Trust, Security and Privacy in Global Computing

A thesis submitted to the
University of Dublin, Trinity College,
in fulfilment of the requirements for the degree of
Doctor of Philosophy (Computer Science)

2005

Jean-Marc Seigneur

DECLARATION

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

Jean-Marc Seigneur,

The 30st of March 2005.

PERMISSION TO LEND AND/OR COPY

I, the undersigned, agree that the Trinity College Library may lend and/or copy this thesis upon request.

____

Jean-Marc Seigneur,

The 30st of March 2005.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my whole family, especially my parents, for their love and constant support. A special thought to my friends from my village and from the world.

I also express my gratitude to my supervisor, Dr. Christian Damsgaard Jensen of the Technical University of Denmark.

A few colleagues of Trinity College Dublin (TCD) have to be acknowledged for the fruitful collaborations we have had: Dr. Joerg Abendroth, Mr. Alan Gray, Dr. Fergal Shevlin, Dr. Donal Lafferty, Mr. Pierpaolo Dondio, Mr. Daniel Solis, Mr. Stephen Farrell, Mr. Yong Chen, Mr. Colin Harris, Mr. Patroklos Argyroudis, Mr. David O'Callaghan, Dr. Gregory Biegel, Ms. Elizabeth Gray, Dr. Tilman Schaëfer, Prof. Vinny Cahill, Dr. Jim Dowling, Mr. Peter Barron, Mr. Alexis Donnelly, Mr. Anselm Lambert and Dr. Margaret Woods.

My appreciation to the members of DSG, the TCD Security Interest Group (TCDSIG), http://www.trustcomp.org and the EU SECURE project, especially Prof. Ciarán Bryce and the members of the OSG of the University of Geneva, Dr. Waleed Wagealla and the members of the Smartlab of the University of Strathclyde, Mr. Nathan Dimmock of the Opera Group of the University of Cambridge, and last but not least the members of BRICS of the University of Aarhus for their formal view on computational trust.

Finally, I would like to acknowledge my two examiners, Dr. Stefan Weber and Dr. Stephen Marsh. They accepted my thesis *as it stands* but I took the opportunity to integrate the result of the argumentation that we had during my viva voce in this final version of the text.

“We shall not cease from exploration,
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.”

T.S. Eliot, “Little Gidding” ***[53]***

ABSTRACT

During the past thirty years, the world of computing has evolved from large centralised computing centres to an increasingly distributed computing environment, where computation and communication capabilities are being embedded in artefacts of everyday life. Billions of computational entities will interact in systems with ever changing configurations determined by local and global context, for example, the location of the user. In such dynamic environments, users would be overwhelmed if involved in computing-related decisions every time the context changes. Due to the number of decisions required to sustain continuous service, most decisions will have to be made by the computing entities themselves. Moreover, due to the global scale of the environment and the potential risk of disconnected operations, the computing entities may have to make these decisions autonomously, without relying on a given fixed infrastructure. Knowledge, especially about the context of the interaction, is vital for the accuracy of these decisions. However, keeping information on a global scale is unfeasible for resource-constrained entities, so some degree of uncertainty must be assumed.

This peer-to-peer type of interaction in an uncertain world where interactions are needed to go forward resembles what occurs in human social networks. The notion of trust has emerged in human society to allow humans to make decisions under such circumstances. It has been proposed that computing entities can make decisions based on a computational model of trust. The trust engine run by each entity distributes and gathers pieces of evidence, that is, knowledge about the interacting entities: direct observations, recommendations or reputation.

Since the trust engines collaborate and malicious collaborating entities exist, security through collaboration must be considered. As the real world does not have a unique legitimate authority, computing entities are owned by multiple authorities and operated from multiple jurisdictions. As in real life, no administrator can be perpetually present to manage the interactions. The trust engine can adapt security in a peer-to-peer way.

A crucial element for the use of trust is to know with whom the entities interact, which corresponds to authentication in traditional computer security. However, this element has been disregarded in computational trust: this is ill-fated given that virtual identities are the means for a number of attacks that are less possible in face-to-face settings. This thesis sets up a framework, called *entification*, which encompasses both computational trust and identity aspects, and whose goal is to be applicable to global computing. For this purpose, this thesis

draws another parallel with human social networks, namely the notion of *entity recognition (ER)*. When someone is introduced by a trustworthy recommender, the identity card of the recommended person is not used and it is sufficient to recognise this person. It provides dynamic enrolment and, in doing so, ad-hoc interactions are possible. It also underlines that the full curriculum vitae of the recommender is not required, which translates to a privacy improvement over trust engines that link all interactions to a real-world identity. The entification framework follows an ER approach: the virtual identities are, by default, pseudonyms – recognised, but without link to the real-world identities. It is sufficient to recognise a virtual identity in order to build trust based on the above list of pieces of evidence. The link to the real-world identity may be considered to be useful for security decisions and our framework does not forbid the use of this link. However, in global computing, the possibility to sue the real-world identities behind the virtual identities is not guaranteed since the jurisdictions of the interacting entities may be contradictory. In addition, most authentication schemes linking a real-world identity with a virtual identity do not achieve dynamic enrolment and their usability compromises security.

Still, our framework takes into account the attacks at the level of virtual identity. Instead of authentication, a novel ER process is carried out. The outcome of ER is associated with a level of confidence in recognition rather than a binary authentication outcome: in doing so, weaker recognition schemes can be used. Recognition is a basis for trust computation, which starts the *end-to-end trust*, which emphasises that the trust in the technical infrastructure must be taken into account when the trust in the virtual identity is computed. In addition, *trust transfer* is introduced to encourage self-recommendations without attacks based on the creation and use of a large number of virtual identities owned by the same real-world identity. Since privacy expectations vary, a privacy-trust trade model is introduced for real-world identities to disclose explicit links between their virtual identities. Once these links are unveiled, *fusionym* is carried out to compute an overall trust value.

An empirical approach to evaluation is taken to understand the impact and limitations of the entification framework. Thus, different application scenarios that have challenging characteristics expected in global computing environments have been chosen. The framework has been implemented and peer-reviewed in these application domains including message-based and vision-based recognition. Simulation has been carried out to evaluate performance. The peer reviews also covered threat analysis and the privacy aspects.

PUBLICATIONS RELATED TO THIS PH.D.

Journal/Magazine papers:

1. "The Claim Tool Kit for Ad-hoc Recognition of Peer Entities", J.-M. Seigneur and C. D. Jensen, in *the Journal of Science of Computer Programming*, Elsevier, 2004.
2. "Using Trust for Secure Collaboration in Uncertain Environments", V. Cahill, E. Gray, J.-M. Seigneur, C. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. d. M. Serugendo, C. Bryce, M. Carbone, K. Krukow, and M. Nielsen, in *Pervasive Computing Mobile and Ubiquitous Computing*, vol. 2(3), July-September, IEEE, 2003.
3. "Fostering Sustainability via Trust Engines", J.-M. Seigneur, in *Technology and Society,* vol. 24(1), IEEE, 2005.
4. "Privacy Recovery with Disposable Email Addresses", J.-M. Seigneur and C. D. Jensen, in *Security&Privacy, special issue on Understanding Privacy*, November-December, IEEE, 2003.
5. "Security in Exotic Wireless Networks", S. Farrell, J-M. Seigneur and C. D. Jensen, NATO Computer and Systems Sciences Series III, vol.193 "Security and Privacy in Advanced Networking Technologies", ISBN 1 58603 430 8, ISSN, 1 387-6694, IOS Press, 2004.

Conference papers:

1. "Trust Transfer: Encouraging Self-Recommendations without Sybil Attack", J.-M. Seigneur, A. Gray and C. D. Jensen, LNCS, in *Proceedings of iTrust'05 the Third International Conference on Trust Management*, Springer-Verlag, 2005.
2. "Towards An Evaluation Methodology for Computational Trust Systems", C. Bryce, J.-M. Seigneur, N. Dimmock, W. Wagealla, K. Krukow and V. Cahill, in *Proceedings of iTrust'05 the Third International Conference on Trust Management*, LNCS, Springer-Verlag, 2005.
3. "A Case Study Implementation of a Trust Engine", C. Bryce, P. Couderc, J.-M. Seigneur, and V. Cahill, in *Proceedings of iTrust'05 the Third International Conference on Trust Management*, LNCS, Springer-Verlag, 2005.
4. "Combating Spam with TEA (Trustworthy Email Addresses)", J.-M. Seigneur, N. Dimmock, C. Bryce and C. D. Jensen, in *Proceedings of the 2nd Conference on Privacy, Security and Trust*, Canada, 2004.
5. "Default Free Introduction, Rare Self-Introduction Fee, Costly Spoofing: No Profitable Spam", J.-M. Seigneur and A. Gray, EUROPRIX Scholars Conference, Tampere, Finland, 2004.

6. "Trading Privacy for Trust", J.-M. Seigneur and C. D. Jensen, in *Proceedings of iTrust'04 the Second International Conference on Trust Management*, LNCS, Springer-Verlag, 2004.
7. "Interaction with Trust in Ambient Intelligence", J.-M. Seigneur, in *Proceedings of UbiMob*, Nice, France, ACM, 2004.
8. "Ambient Intelligence through Image Retrieval", J.-M. Seigneur, D. Solis and F. Shevlin, in *Proceedings of the 3rd International Conference on Image and Video Retrieval*, LNCS, Springer-Verlag, 2004.
9. "Trust Enhanced Ubiquitous Payment without Too Much Privacy Loss", J.-M. Seigneur and C. D. Jensen, in *Proceedings of the 19th Symposium on Applied Computing*, Nicosia, Cyprus, ACM, 2004.
10. "Towards Security Auto-Configuration for Smart Appliances", J.-M. Seigneur, C. D. Jensen, S. Farrell, E. Gray, and Y. Chen, in *Proceedings of the Smart Objects Conference*, Grenoble, France, 2003.
11. "End-to-end Trust Starts with Recognition", J.-M. Seigneur, S. Farrell, C. D. Jensen, E. Gray, and Y. Chen, in *Proceedings of the First International Conference on Security in Pervasive Computing*, Boppard, Germany, LNCS, Springer-Verlag, 2003.
12. "P2P with JXTA-Java Pipes", J.-M. Seigneur, G. Biegel, C. D. Jensen, in *Proceedings of the 2nd International Conference on the Principles and Practice of Programming in Java*, ACM, 2003.
13. "Trust Propagation in Small Worlds", E. Gray, J.-M. Seigneur, Y. Chen, and C. D. Jensen, in *Proceedings of the First International Conference on Trust Management*, LNCS, Springer-Verlag, 2003.

Workshop papers:

1. "The REL Project: Mobile-based Reliable Relations", J.-M. Seigneur, P. G. Argyroudis, D. O'Callaghan, and J. Abendroth, in *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, W3C, 2004.
2. "Towards Trustworthy Eco Computing", J.-M. Seigneur, in *Proceedings of the Sustainable Pervasive Computing Workshop of the Second International Conference on Pervasive Computing*, 2004.
3. "The Role of Identity in Pervasive Computational Trust", J.-M. Seigneur and C. D. Jensen, in *Proceedings of the Security and Privacy in Pervasive Computing Workshop of the Second International Conference on Pervasive Computing*, Kluwer, 2004.
4. "Risk Probability Estimating Based on Clustering", Y. Chen, C. D. Jensen, E. Gray and J.-M. Seigneur, in *Proceedings of the 4th Annual IEEE Information Assurance Workshop*, 2003.
5. "Bank Accounting and Ubiquitous Brokering of Trustos", J.-M. Seigneur, J. Abendroth, C. D. Jensen, in *7th CaberNet Radicals Workshop,* Bertinoro, Italy, 2002.
6. "Secure Ubiquitous Computing Based on Entity Recognition", J.-M. Seigneur, S. Farrell, C. D. Jensen, in *Ubicomp2002 Security Workshop,* Göteborg, Sweden, 2002.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

During the past three decades, the computing environment has changed from centralised stationary computers to distributed and mobile computing. This evolution has profound implications for the security models, policies and mechanisms needed to protect users' information and resources in an increasingly globally interconnected computing infrastructure. In centralised stationary computer systems, security is typically based on the authenticated identity of other parties. Strong authentication mechanisms, such as Kerberos [106] or Public Key Infrastructures (PKIs) [80], have allowed this model to be extended to distributed systems within a single administrative domain or within a few closely collaborating domains. However, small mobile devices are increasingly being equipped with wireless network capabilities that allow ubiquitous access to corporate resources and allow users with similar devices to collaborate while on the move. Traditional, identity-based security mechanism cannot authorise an operation without authenticating the claiming entity. This means that no interaction can take place unless both parties are known to each others' authentication framework. Spontaneous interactions would therefore require that a single, or a few trusted Certificate Authorities (CAs) emerge, which, based on the inability of a PKI to emerge over the past decade, seems highly unlikely in the foreseeable future. In the current environment, a user who wishes to partake in spontaneous collaboration with another party has the choice between enabling security and thereby disabling spontaneous collaboration or disabling security and thereby enabling spontaneous collaboration. The state-of-the-art is clearly unsatisfactory, instead, mobile users and devices need the ability to autonomously authenticate and authorise other parties that they encounter on their way, without relying on a common authentication infrastructure.

In order to address this problem, we first need to examine the challenges introduced by "global computing" [84], which is a term coined by the EU for the future of the global information society, and identify their impact on security. We then examine how trust management and computational trust engines have been proposed to tackle security in these global computing environments. Finally, we summarise the aims, objectives and contribution of this thesis and present its organisation.

1.1 Towards Global Computing Systems

A decade ago, Weiser [187] envisioned computing capabilities to be woven into the fabric of every day life, indistinguishable from it – global computing is the next evolution of computing environments. The scale of the proliferation of computing entities can therefore be expected to be huge. Consequently, scalability issues appear for this increasing base of computing devices, especially the tiny ones, which are resource-constrained in many aspects, for example, memory, power or communication range. In addition, even powerful servers may be challenged to deal with worldwide data and to serve a worldwide base of clients. Decentralisation is also needed due to the potential failures of these servers.

According to Weiser's principle of "calm technology" [188], it is a great challenge not to overload humans with pervasive computing tasks. Ideally, it should go beyond usability, up to transparency. Global computing must be considered from a user acceptance and technology adoption point of view.

Another user aspect to take into account is that global computing functionalities experienced in the user's home environment will be expected anywhere anytime. The user's mobility implies that resources left in the home environment must be accessed via interconnected third-parties. When the user moves to a foreign place for the first time, it is highly probable that the third-parties of this place are a priori unknown, strangers. However, to interact with these strangers is still necessary, for example, to access his/her remote home environment.

1.2 Security in Global Computing Systems

It is a reality that users can move to potentially harmful places, for example, by lack of information or due to uncertainty, there is a probability that previously unknown computing third-parties used to provide global computing in foreign places are malicious. The assumption of a known and closed computing environment held for fixed, centralised and distributed computers until the advent of the Internet and more recently global computing.

Legacy security models and mechanisms rely on the assumption of closed computing environments, where it is possible to identify and fortify a security perimeter, which protects against potentially malicious entities. However, in these models, there is no room for anytime anywhere mobility. Moreover, it is supposed that inside the security perimeter, there is a common security infrastructure, a common security policy or a common jurisdiction where the notion of identity is globally meaningful. The absence of this assumption creates unaddressed issues in four surfacing dimensions, which intermingle and overlap, *Adaptability, Security, Usability and Privacy* (*ASUP*):

- Adaptability: There is a risk of scalability issues, for example, the memory of a tiny device cannot store the information about all its peers. Some information cannot be retrieved fast enough, for example, due to computationally expensive security operations so the window of opportunity is missed. An application must be able to adapt to an entity's capabilities and to the capabilities offered by the environmental context, for example, as the user roams. Ideally, it must be adaptable to all environments without requiring the availability of known parties in a decentralised peer-to-peer (P2P) way.
- Security: When known third-parties are unreachable, which is even more often the case in mobile ad-hoc networks than in traditional fixed networks, security mechanisms requiring access to these third-parties cannot be applied. In this case, security must be created from scratch with reachable unknown entities. A fundamental requirement for global computing environments is to allow for potential interaction and collaboration with unknown entities because they are interdependent.
- Usability: Due to the potentially large number of previously unknown entities and simple economic reasons, it makes no sense to assume the presence of a human administrator who configures and maintains the security framework. This means that individual entities must decide about each of these potential interactions themselves. It applies to security decisions too, for example, concerning the enrolment of a large number of unknown entities.
- Privacy: Some entities will be ambient/invisible but listening – so, it becomes even harder to guarantee private spaces. However, there are privacy protection laws and use of computing technologies must comply with these laws. Even if we assume that

computing entities are willing to respect privacy, the burden of countless privacy decisions to be made as the context changes goes against usability.

Therefore, traditional security approaches are challenged in global computing and means are needed to address issues in four dimensions, the ASUP dimensions.

1.3 Trust Engines for Security in Global Computing

There is an inherent element of risk whenever a computing entity ventures into collaboration with a previously unknown party. One way to manage that risk is to develop models, policies and mechanisms that allow the local entity to assess the risk of the proposed collaboration and to explicitly reason about the trustworthiness of the other party in order to determine if the other party is trustworthy enough to mitigate the risk of collaboration. Formation of trust may be based on previous experience, recommendations from reachable peers or the perceived reputation of the other party. Reputation, for example, could be obtained through a reputation system such as the one used on eBay.

Explicit trust management has additional advantages: it allows an entity to select the service provider that is most likely to provide the required service whenever it is faced with a number of previously unknown service providers. We therefore believe that a general trust management middleware will provide the enabling technology for secure and reliable collaboration in highly dynamic (mobile) computing environments.

In these potential dynamic environments of global computing, users would be overloaded by making decisions every time the context changes. The usability requirement implies that most decisions will have to be made by the computing entities themselves, probably autonomously. Trust engines, based on computational models of the human notion of trust, have been proposed to make security decisions on behalf of their owner. For example, this thesis has been carried out as part of the SECURE project [27, 151, 154], which has built a generic and reusable trust engine. More generally, each computing entity would run a trust engine. These trust engines allow these entities to compute levels of trust based on sources of trust evidence, that is, knowledge about the interacting entities: local observations of interaction outcomes or recommendations. Based on the computed trust value and given a trust policy, the trust engine can decide to grant or deny access to a requesting entity. Then, if access is given to an entity, the actions of the granted entity are monitored and the outcomes, positive or negative, are used to refine the trust value.

Another source for trust in human networks consists of real-world recourse mechanisms, such as insurance or legal actions. Traditionally, it is assumed that if the actions made by a computing entity are bound to a real-world identity, the owner of the faulty computing entity can be brought to court and reparations are possible. In open environment with no unique authority, such as in global computing, the feasibility of this approach is questionable. An example where prosecution is ineffective occurs when email spammers do not mind to move operations abroad (at the time of writing Brazil and China [137]) to escape any risk of prosecution. It is a fact that there are multiple different jurisdictions in global computing. Therefore, security based on the authenticated identity may be superfluous. Furthermore, in the first place, there is the question of which authority is in charge of certifying the binding with the real-world identity, since there are no unique global authorities. "Who, after all, can authenticate US citizens abroad? The UN? Or thousands of pair wise national cross-certifications?" [101].

More importantly, is authentication of the real-world identity necessary to be able to use the human notion of trust? This thesis focuses on this question, especially with regard to the SECURE project. Indeed, a critical element for the use of trust is to retrieve trust evidence on the interacting entities but trust evidence does not necessarily consist of information about the real-world identity of the owner: trust evidence may simply be the count of positive interactions with a virtual identity. In order to retrieve trust evidence, in this thesis, we draw another parallel with human social networks, namely the notion of entity recognition. When a new person is introduced by a trustworthy recommender, the identity card of the recommended person is not used and it is sufficient to recognise this person. It serves to improve the spontaneity of the interactions. It also highlights a potential benefit from a privacy point of view. Still, as long as the interacting computing entities can be recognised, direct observations and recommendations can be exchanged in order to build trust, interaction after interaction. This level of trust can be used for trusting decisions. Thus, trust engines can provide dynamic protection without the assumption that real-world recourse mechanisms, such as legal recourse, are available in case of harm.

1.4 Aims and Objectives

The general objective of this thesis is to study the role of identity in pervasive computational trust and develop novel mechanisms for managing and authenticating identities in a trust-based security framework. The previous section has already introduced the notion of entity recognition, which seems to be sufficient to make trusting decisions based on trust engines. This thesis must demonstrate how this new computing paradigm, extracted from a parallel with human social networks, can be effectively used within trust engines.

In order to do so, first, a computational process based on this notion of entity recognition must be designed and implemented. Then, this process must be integrated into a trust engine. Therefore, a framework integrating both entity recognition and computational trust is needed.

Then, it must be validated that recognition can effectively be used. Since there are a number of attacks that can happen at the level of identity and challenge the use of trust engines, the framework must be discussed from a security point of view. A threat analysis is required to evaluate the use of entity recognition in trust engines. Although concrete global computing environments do not exist yet, tests in related existing environments are needed.

In addition to the security dimension, the framework must be discussed according to the other ASUP dimensions of global computing environments: to what extent the framework improves adaptability, security, usability and privacy.

1.5 Contribution of this Thesis

Within the context of this thesis we have designed, implemented and evaluated the first framework that combines the human notion of entity recognition and computational trust. This framework is called the entification framework. It fills the gap between identities and their level of trust, which is one of the eight "major issues" [43] in developing identity management for the next generation of distributed applications.

The state-of-the-art highlights two application domains that can be tested to study global computing aspects: the email domain, where any honest sender, worldwide, should be allowed to send emails without a priori information about their honesty; and the smart home, where administrators are not economically viable.

The contributions are to the ASUP dimensions in the following way.

Firstly, the entification framework improves the adaptability dimension thanks to:

- the management of the large number of recognition clues based on context to cope with scalability; tested with vision-based recognition, which indeed generates a lot of recognition clues;
- the possibility to plug and use a number of different ER schemes depending on the application domain and the context.

Secondly, the entification framework also contributes to the security dimension by:

- the identification of layers of trust, technical trust and trust in the entity, which are explicitly taken into account in the framework; trust in the entity, which is normally formed and maintained by the trust engine; static and dynamic evidence-based means to compute the level of technical trust, when it is abstracted to ER schemes; technical trust, which is useful to mitigate identity usurpation attacks, especially in message-based applications; new anti-spoofing techniques, peer-reviewed in the email domain to fight spam; and a generic tool kit for message-based recognition implemented in Java;
- the use of ER performance results in cost/benefit security analysis and decision-making, which is especially important for the risk analysis component of the SECURE trust engine (the SECURE project was the work context of the thesis);
- a computational model of entity recognition as a replacement of the authentication process, which can work without centralised authentication trusted-third-parties;
- the mitigation of attacks due to the control of many virtual identities by the same real-world identity when the level of trust is based on counts of interaction outcomes; it has shed light on novel networked engineered attacks using a priori knowledge about the network to select and attack the most well-connected entities.

Thirdly, the contributions to the usability dimension are made thanks to:

- the possible removal of any binding between the real-world identity and the virtual identity when *pure ER schemes*, which do not require the link between the virtual identity and the real-world identity, are used; it means than the enrolment can be fully automated and software-based;

- the support in the tool kit of the automatic selection of the most appropriate virtual identity depending on the current context, including the threat level.

Finally, contributions to the privacy dimension correspond to:

- the identification of the inherent conflict between privacy and trust: the impact that computational trust, based on knowledge about the trustee, has on privacy;
- the possibility to encourage privacy protection by the use of pure ER schemes and the support of multiple pseudonyms per person in the message-based tool kit;
- the new functionality to negotiate privacy for trust between the interacting entities.

Therefore, this thesis contributes to the four ASUP dimensions.

1.6 Organisation of this Thesis

After this introduction to global computing, security, trust engines and entification, we structure the remainder of this thesis as follows. In Chapter 2, we delve into the details of the evolution from traditional computer security to trust engines. Chapter 3 details the SECURE trust engine (result of a three-year five-partner joint-work), which is assumed to be the reference trust engine for the thesis. In Chapter 4, it is brought to light that the current approach to identity in trust engines is too simplistic, and the reasons why trust engines do not fulfil the ASUP requirements are discussed. Chapter 5 describes the entification framework, whose goal is to fill this gap between identity and trust engines to achieve security in global computing. Chapter 6 presents the platform for instantiation of, and experimentation with the entification framework, especially in the two application domains identified in the state-of-the-art as suitable approximations of global computing to evaluate the ASUP requirements. In Chapter 7, we validate our work by presenting qualitative and quantitative results in these selected application scenarios with regard to the ASUP requirements. Finally, Chapter 8 concludes this thesis by summarising the presented work and outlining issues that remain open for future work.

CHAPTER 2: TOWARDS TRUST ENGINES FOR SECURITY IN GLOBAL COMPUTING

This chapter begins by presenting a survey of legacy computer security and how it is challenged in global computing. Then, trust management is surveyed, which leads to an examination of computational trust engines and how they can be used to address security in global computing.

2.1 Computer Security

Although computer security influenced many early computing systems, the discipline really started to be studied in the early 1970s. The traditional goal of computer security is to ensure the Confidentiality, Integrity and Availability (these are often referred to as the CIA properties) of information and resources in the system [72]:

- Confidentiality: the goal is that only authorised persons have the access to information and resources;
- Integrity: the goal is that the information cannot be tampered with, neither during communication nor in its fixed storage;
- Availability: the goal is that both information and resources are available when the authorised persons need them.

There are two main approaches to tackle these goals: information flow and access control.

Information flow control models describe the information flow within an information system and how flows can be prohibited [7, 105]. Information flow security tackles covert channel attacks. Non-interference models are related but focus on what the entities know about the state of the system and prevent access if there is interference [91, 105]. In the Chinese Wall

Model [33], imaginary walls are built as entities access objects in conflict-of-interest domains.

In this thesis, we focus on the access control method to computer security. In the access control model, confidentiality and integrity of stored data is ensured by restricting access to information and resources to entities who are trusted (this fails in global computing where traditional implicit trust relationships do not exist). The method is to decide whether or not a requesting entity is allowed to access a resource or execute an action. There are two main approaches:

- identity-based access control: in order to make this decision, the identification of the requesting entity is followed by authentication, which consists of verifying that the requesting entity has the identity it claims to have; then, once authentication is successfully passed, further access control operations are carried out in order to reject/grant authorisations and privileges: "authentication is one problem and access control is a completely different one" [168];
- credential-based access control: in this case, instead of relying on the identity of the requester, the requester exhibits credentials different than identity; however, the credentials must still be obtained and certified in the first place.

The authorisation work is usually done by a component called the reference monitor. Abendroth [7] presents the various access control mechanisms of distributed systems in a unified view. When an access matrix model [110] is used, an object may be associated with an Access Control List (ACL [72]), which lists the access right of the entities. A second implementation may be that the access rights are given to the entities in the form of capabilities. An extension of the second implementation may be that instead of a system protected capability, certificates with a standard format are issued [7]. In Role-Based Access Control (RBAC) [58], roles are given to entities, they can choose to play/use one or more of these roles. Usually a role combines all permissions necessary to perform a duty in the application environment.

Confidentiality and integrity of communications are normally achieved through cryptography. There are two main approaches concerning the secret used for encryption. With symmetric cryptography algorithms, the secret must be known locally for encryption or decryption. The symmetric encryption algorithms fall in two main categories [168]: block ciphers (DES, AES, IDEA, Blowfish, CAST…) and stream ciphers (RC4, SEAL…). With asymmetric cryptography, there is no need for shared secrets (e.g., previously exchanged during an a priori secure session) since a public key (*Pub*) can be publicly distributed while a private key (*Pri*) is kept local to sign or decrypt. The encryption is done with the public key of the entity who is supposed to decrypt. It is also possible to verify a signature made with a private key using the corresponding public key. Asymmetric cryptography is based on hard mathematical problems, for example, RSA [143] uses large enough prime numbers to make the trial-and-error factoring attacks too difficult to be practical. Another example is the Digital Signature Standard (DSS [52]), which uses another hard problem other than factorisation, which is the discrete algorithm problem. Challenge/Responses (C/Rs) consist of sending a challenge to be encrypted with the private key and verified using the public key. Public key certificates (or credentials) are used to bind/link attributes to public keys. There are certificate standards to improve their interoperability: for example, globally unique X.509 distinguished names [83].

2.2 Legacy Security is Challenged by Global Computing

Gollman [72] emphasises that “decisions about technical security matters cannot be made without reference to the context they are applied in”. In this section, the evolving landscape of computer security is presented from traditional managed computing environments to global computing environments. At the time of the writing of this thesis, security has moved from “easy” [72] computer security in a single centralised security enforcing system to a collection of computers linked via some network, that is, distributed systems. Security for systems connected to the Internet – the World Wide Web – is considered even more difficult, because it spans a plethora of independent (possibly conflicting) authorities. Global computing goes even further because collaboration becomes open to any of the potential interacting computing entities.

Figure 1 depicts the scenario where legacy security mechanisms are applied to a global computing environment. In this figure (as well as in Figure 2), the security perimeters or domains of different parties are delimited by dashed lines; the flows of requests made by an entity are represented by large white arrows (from identification to authentication to authorisation/reference monitor to the resource); if at some stage manual intervention is required to set some sort of trust relation, a blue arrow is drawn. The blue colour is also used to indicate that the administrators must be highly trusted. Figure 1 focuses on the domain *A*, which has a security perimeter administered by an administrator. The basic scenario is when an entity identifies itself as a known local entity of *A*'s domain. After this identification, authentication is used to make sure that the entity really possesses the identity it claims to have. This is usually possible because the trusted administrator of *A*'s domain will have created an account (or related authentication material) for this entity, which usually involves manual tasks to enable this permission. Then, the reference monitor is consulted to find out whether the authenticated entity is allowed to access the specific resource. Again, it usually involved that the administrator decided that the entity is allowed access and carried out manual tasks. A variant is when credentials are used directly instead of identity. In this case, identity authentication is not done. However, the credentials, which authorise access, still have to be manually configured at some stage and the validity of these credentials must be verified anyway. To summarise the access inside the security perimeter is restricted thanks to some manual trust set up by the administrator.

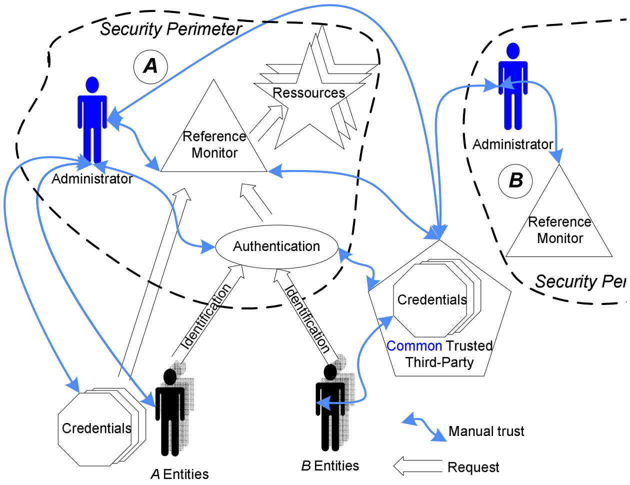


Figure 1. Security Administration Burden of Legacy Mechanisms in Global Computing

When it is required to allow collaboration between two domains or security perimeters (A and B in Figure 1), the administrators of both domains must agree on how to merge their security domains. It can be facilitated if they have a trusted-third-party in common as depicted in Figure 1 with the pentagon, although manual tasks are needed to associate the credentials of the third-party to the entities of each domain. In global computing, due to the scale of the number of entities, such an approach where manual administrative tasks are constantly needed would most likely overwhelm the people playing the role of administrators. In fact, it may even be the case that there is no administrator available, who is knowledgeable about security related issues, because it is too expensive to pay a dedicated administrator. For example, the home environment is different than the corporate environment because most households cannot employ an administrator. This is a real problem because it is expected that intelligent appliances and home networks will become standard home assets. In the email domain, the cost of involving humans in security decisions concerning spam has underlined that even quick security decisions made by humans are very costly when they have to be made in a great number [98].

Due to the global scale of the environment, as depicted in Figure 2, a foreign entity may have credentials from a trusted-third-party but they are useless because this trusted-third-party means nothing to the domain under consideration (that is, the domain *A* in our example). The authentication may be certified by the foreign trusted-third-party but the reference monitor cannot take this information into account: it is represented by the red circle and question marks. Global authentication would require a single or few commonly trusted CAs to emerge, which, based on the inability of a global PKI to emerge over the past decade, seems very unlikely in the foreseeable future.

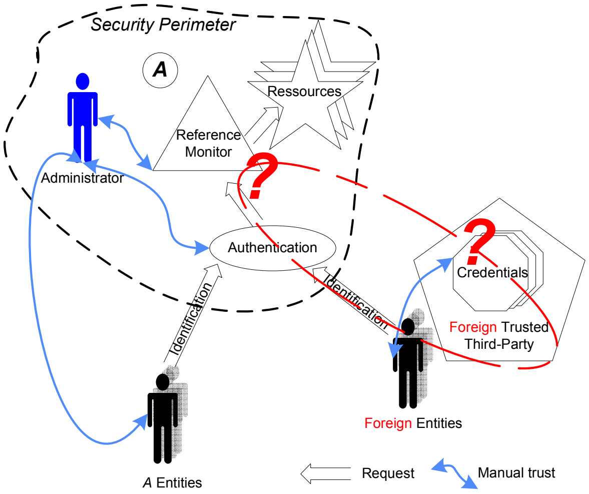


Figure 2. Shortcoming of Legacy Security Approach in Global Computing

2.3 Solving Security via Trust Engines in Global Computing

We start with an explanation of why implicit and credential-based trust approaches are inadequate in global computing. Then, we present collaborative computational trust engines.

2.3.1 Inadequacy of Implicit Trust and Credential-based Trust

The terms trust/trusted/trustworthy, which appear in the traditional computer science literature, are not grounded on social science and often correspond to an implicit element of trust. For example, we have already mentioned above the use of trusted-third-parties, called CAs, which are common in PKI infrastructures. Another example is the Trusted Computed Platform Alliance (TCPA) [174], which makes it clear that we can speak of trust in technical components. Its goal is to create enhanced hardware – by using cost effective security hardware (more or less comparable to a smart card chip) that acts as the "root of trust". *They are trusted* means that they are assumed to make use of some (strong) security protection mechanisms. Therefore they can/must implicitly be blindly trusted and cannot fail. This cannot address security when it is not known who or whether or not to blindly trust.

The term "trust management" has been introduced in computer security by Blaze et al. [21] with the following definition: "specifying and interpreting security policies, credentials, and relationships that allows direct authorization of security-critical actions". Some of the trust management systems have been very effective in connected, administered, distributed, rather static environments (for example, Keynote [21, 100]). Privileges can be delegated to nodes thanks to credentials or certificates. A chain of credentials may be created to represent the propagation of trust between nodes. Many extensions have been added, such as the possibility to negotiate credentials. The specification of application security policies and credentials is standardised. Thus, the policies can easily be distributed. Entities that, after identification, request to carry out actions can do so if the compliance checker agrees given the policy and the set of presented credentials. They differ from the previous system security approach, where trust policies are implicit, by using security credentials (or certificates) that must be held for authorisation. The trust management system tries to prove that a request and a list of credentials comply with a specific policy. Others have argued that the model still relies on an implicit notion of trust because it only describes "a way of exploiting established trust relationships for distributed security policy management without determining how these relationships are formed” [177]. Automated Trust Negotiation (ATN) is argued to improve trust management systems such as Keynote, which all “support delegation of authority, but are not helpful for establishing trust between strangers using general-purpose credentials” [190]. However, ATN “does not address the client’s need for trust before it requests service” [189]. Furthermore, because it is required to show credentials in order to start the collaboration, this type of system may have a bootstrapping problem when no credential is already obtained, or cannot be discovered, which may often be the case in global computing settings [177]. It also does not help if no third-party is accepted as a common credential provider by both parties. There is a need for trust formation mechanisms from scratch between the two strangers. Credentials may be exchanged or accepted after this trust formation phase but not before it is achieved.

2.3.2 Collaborative Trust Computation to Bootstrap Trust in Strangers

The use of an explicit notion of trust based on the human notion of trust has yielded to a new class of trust management system, called *evidence-based trust management* where the level of trust is explicitly computed by a trust engine. There are many definitions of trust in a wide range of domains (please refer to [120, 139, 140, 177]). In this thesis, the human notion of trust is Romano’s one [145]:

> *"Trust is a subjective assessment of another’s influence in terms of the extent of one’s perceptions about the quality and significance of another’s impact over one’s outcomes in a given situation, such that one’s expectation of, openness to, and inclination toward such influence provide a sense of control over the potential outcomes of the situation." [145]*

We call the trust in a given situation, the *trust context*. In social research, there are three main types of trust: interpersonal trust, based on past interactions with the trustee; dispositional trust, provided by the trustor’s general disposition towards trust, independent of the trustee; and system trust, provided by external means such as insurance or laws [125, 140]. Depending on the situation, a high level of trust in one of these types can become sufficient for the trustor to make the decision to trust. When there is insurance against a negative outcome, or when the legal system acts as a credible deterrent against undesirable behaviour, it means that the level of system trust is high and the level of risk is negligible – therefore the levels of interpersonal and dispositional trust are less important. It is usually assumed that by knowing the link to the real-world identity, there is insurance against harm that may be done by this entity: in essence, this is security based on authenticated identity and legal recourse. In this case, the level of system trust seems to be high but one may argue that in practice the legal system does not provide a credible deterrent against undesirable behaviour, e.g., it makes no sense to sue someone for a single spam email, as the effort expended to gain redress outweighs the benefit. We have already strengthened that due to multiple jurisdictions in global computing, legal recourse is questionable.

An interesting case might be to consider the real-world recourse mechanism as an entity and the level of trust is explicitly computed based on the number of times the real-world recourse has successfully played its role (which is indeed evidence-based computation of the level of trust). Of course, scenarios where the level of system trust is low make interpersonal trust more important. The level of dispositional trust may be set due to two main facts. First, the user manually sets a general level of trust, which is used in the application to get the level of trust in entities, independently of the entities. Secondly, the current balance of gains and losses is very positive and the risk policy allows any new interactions as long as the balance is kept positive. Marsh uses the term "basic trust" [122] for dispositional trust; it may also be called self-trust.

Interpersonal trust is represented as a computed *trust value*. Thus, in this thesis:

> *a trust value, that is the digital representation of the trustworthiness or level of trust in the entity under consideration, is seen as a non-enforceable estimate of the entity's future behaviour in a given context based on past evidence.*

"Trust essentially is and should be based on knowledge" [90] – knowledge is brought by evidence. Computational trust is an innovative mechanism towards the prediction of behaviour [176, 177]. When a user is involved in the application scenario and manually sets trust values in specific virtual identities (for example, an email address of a known important contact is whitelisted even if the email address is spoofed from time to time or banned due to the same reason), it must be considered as *manual trust values*. Manual trust values are difficult to be integrated in trust engines if not converted in an equivalent of event outcomes. For example, if the user gives a value of *18* positive outcomes and *2* negative ones, it seems fine, if he/she gives a value of *0.75* on a *[0,1]* scale, it does not seem compatible.

Based on the trust value, security decisions can be made according to trust policies. For example, the resource is granted to any entities who are associated with a greater trust value than a threshold. The bootstrapping with unknown entities, strangers beyond the security perimeter, can now be carried out without a priori knowledge. The trust value can be formed after an interaction and can be further refined during subsequent interactions. Previous direct interactions may not be obligatory if trustworthy recommenders recommend the newcomer.

2.4 Summary

This chapters presented challenges to legacy approaches for computer security introduced by global computing. The challenges arise because the assumptions of the availability of a dedicated technologically-aware administrator and mutually trusted-third-parties may not be viable.

Trust engines have been researched to decrease the number of decisions that would otherwise involve humans. Another advantage of trust engines is that trust can be built from scratch without the need of system trust and a priori knowledge – strangers beyond the security perimeter can slowly be granted more resources: interaction after interaction their trustworthiness is formed.

CHAPTER 3: THE SECURE TRUST ENGINE

This chapter presents an overview of the SECURE trust engine and highlights the work context of the thesis with regard to computational trust.

3.1 High-level View of the SECURE Trust Engine

Figure 3 depicts the high-level view of the SECURE trust engine. The goal of the SECURE project [27, 151, 154] was to achieve an advanced trust engine formally grounded and usable. A pentagonal decision-making component is called when a requested entity has to decide what action should be taken due to a request made by another entity, the requesting entity.

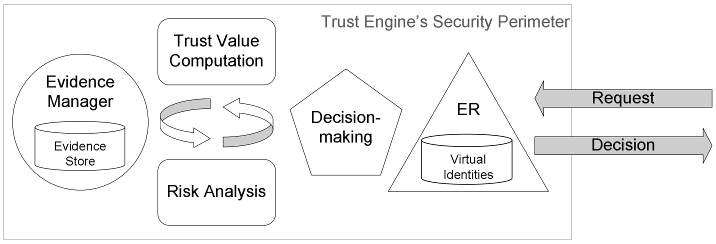


Figure 3. High-level View of the SECURE Trust Engine

In order to take this decision, two sub-components are used:

- one that can dynamically compute the trust value, that is, the trustworthiness of the requesting entity based on pieces of evidence (for example, direct observations[1] or recommendations [183]);
- a risk module that can dynamically evaluate the risk involved in the interaction [47, 48]; risk evidence is also needed.

[1] By direct observations, we mean that the entity has directly interacted with the requesting entity and personally experienced the observation. Another type of observation is when a third-party observes an interaction between two parties and infer the type of outcome.

The chosen action should maintain the appropriate cost/benefit ratio [47, 48]. Depending on dispositional trust and system trust, the weight of the trust value in the final decision may be small. In the background, another component is in charge of gathering evidence: recommendations, comparisons between expected outcomes of the chosen actions and real outcomes… This evidence is used to update risk and trust information. Thus, trust and risk follow a managed life-cycle [177]. A trust engine may be called Trust/risk-based Security Framework (TSF) due to the presence of this risk element.

The Entity Recognition (ER [160]) module deals with digital virtual identities and is in charge of dynamically recognising them. This thesis motivated and permitted the integration of this crucial module into the SECURE trust engine: in fact, this module fills the gap between the notion of identities and trust. Without this ER module (i.e., without this thesis), the SECURE trust engine would not address any identity or privacy attacks. The ER module is also supposed (within the consortium view of the SECURE trust engine) to participate in the population of risk data to be used by the risk analysis component. Information of interest corresponds to: on one hand, performance and overhead of ER schemes; on the other hand, an estimation of the security strength and threat context at the ER level. ER may also be used when evidence is received.

Since some recommenders are more or less likely to produce good recommendations, even malicious ones, the notion of *recommending trustworthiness* has been added to advanced trust engines [6]. Intuitively, recommendations must only be accepted from senders that the local entity trusts to make judgements close to those that it would have made about others. Assuming the user has a metric for measuring the accuracy of another sender's recommendations, Abdul-Rahman and Hailes [6] and Jøsang [91] have suggested models for incorporating that information into the local trust computation. In many cases, the final trust value, which is used locally, may be different than the recommended one. For example, a recommender with trust value of *0.6* on a *[0,1]* scale giving a recommendation of *0.8* provides the adjusted trust value: $0.6 \times 0.8 = 0.48$. However, different trust value formats are possible and some formats are more suitable for evidence-based trust than others. For example, the previous format, a value on a *[0,1]* scale, may be intuitive for humans in order for them to manually set a value but it does not give enough detail on the evidence used to choose this value. It is the reason that the standard SECURE trust value format [127] is a tree of *(s,i,c)*-triples, corresponding to a mathematical event structure [128, 129]: an event outcome count is represented as a *(s,i,c)*-triple, where *s* is the number of events that supports the outcome, *i* is the number of events that have no information or are inconclusive about the outcome and *c* is the number of events that contradict the expected outcome. This format takes into account the element of uncertainty via *i*.

3.2 High-level Activities of the SECURE Trust Engine

There are two high-level processes (as depicted in Figure 4), which are run in parallel: the decision-making process and evidence processing. The directed black arrows represent messages sent from one process to another. When the arrows point outside the rounded rectangle representing the local trust engine, it means that they point to the processes of other trust engines. Activities related to these processes occur because a virtual identity can be: a requester (of an action), a decision-maker, an observer (of first-hand pieces of evidence, e.g., on its own interactions or captured from its own sensors), a receiver (of indirect pieces of evidence; it highlights issues of overcounting due to second-hand pieces of evidence), the subject (of a piece of evidence) or a recommender (of a piece of evidence). Full delegation, that occurs when the local trust engine leaves the decision to an external trust engine, is still possible. However, this is not fully in line with a scenario composed of autonomous entities. A *trust reference* request may also be used to obtain the trust value that a specific entity has in another entity.

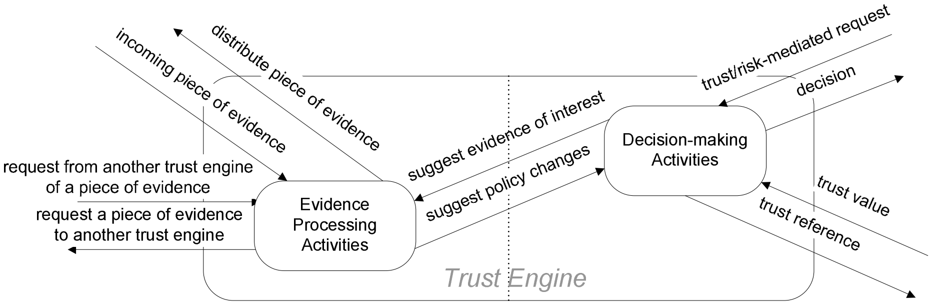

incoming piece of evidence
distribute piece of evidence
request from another trust engine of a piece of evidence
request a piece of evidence to another trust engine
Evidence Processing Activities
suggest evidence of interest
suggest policy changes
Decision-making Activities
trust/risk-mediated request
decision
trust value
trust reference
Trust Engine

Figure 4. High-level Activities of the SECURE Trust Engine

The requester (that is, the requesting virtual identity) of an action and the decision-maker (i.e., the trust engine to whom the request is made) may collaborate during the decision-making. The requester may specify authorisation hints to facilitate the decision-making process. In email settings, the evaluation work (please refer to Chapter 7) has shown that it could be used to some extent to increase the cost on the sender side, and that it makes more sense to put that cost on the sender side than the receiver. The decision-maker may return a decision. In the case of negative decisions, the decision may suggest authorisation hints to be specified in subsequent requests. The different request/decision exchanges form the trust negotiation. The decision-making process might be faster. Further optimisation may consist of attaching the recommendations themselves in authorisation hints. However, care should be taken when including authorisation hints given by the requester because it creates opportunities for attacks on the decision-making process (for example, if authorisation hints consist of a list of recommenders and a restricted number of recommenders are queried during trust formation [177], the requester can force the decision-maker to query only colluding recommenders).

The decision-making process should be able to suggest what kind of evidence is of interest to improve scalability, performance and context adaptation (for example, authorisation hints specify evidence of interest to the evidence manager). In return, the evidence process should be able to suggest changes in the policies driving the decision-making process. The ER module developed in this thesis can indeed be tuned and contributes to the update of the environmental context. Context plays an important role in both processes. Context should affect the way in which the trust engines behave. There are different granularities of context: environmental context concerns the context in which the decision-maker seems to be (e.g., under-attack) and is built from various pieces of evidence; an action request inherently carries further context. The action context, that is, the trust context, allows the trust value to be contextualised to the dimension of trustworthiness of interest. For example, if the request is about the access of privacy information, the trustworthiness of the requesting entity in respecting privacy expectation is of greater importance than the trustworthiness in paying money in time.

3.3 Summary

This chapter presented the SECURE trust engine, especially to emphasise the work context of this thesis and which de facto assumptions were expected with regard to computational trust. Beyond the trust aspect, there is a risk component and it is expected that the ER module feeds this risk analysis with cost/benefit data, such as, ER performance and security strength.

CHAPTER 4: THE IDENTITY GAP IN PREVIOUS TRUST ENGINES

This chapter emphasises that the current approach to identity in trust engines is too simplistic and cannot achieve security in global computing. First attacks due to identity on trust engines are surveyed. Then, remaining issues of the current state-of-the-art due to authentication with regard to the ASUP requirements are presented. The terminology used for identity in this thesis is introduced in Section 4.4. Finally, related work on identity and trust frameworks is discussed, compared and contrasted according to the ASUP dimensions.

4.1 Flawed Trust Engines due to Simple Identity Approach

This section starts by describing the general attacks that are possible on trust engines due to weak identity approach. Then, the well-known Sybil attack [50] is detailed.

4.1.1 Trust Metrics under Identity Usurpation and Multiplicity Attacks

Complete lists of attacks at the authentication level can be found in [8, 22]. In these lists of attacks, the ones that disturb the computing service, such as Denial-of-Service (DoS), are considered too general and computing security problems beyond the scope of this thesis. A number of other attacks can be categorised as identity usurpation attacks: meaning that legitimate trustworthy virtual identities are compromised and become under the control of the attacker, especially with regard to their statements. In addition, we underline that they include the *Security Breach Attacks* (SBAs), which mean if successful that the attacker has been able to compromise a specific security perimeter and gain control over all the resources inside this security perimeter. SBAs are a general challenge to computer security, especially because compromising the personal computer of a user is at time of writing feasible due to the security vulnerabilities present in common operating systems. A *trust metric* consists of the different computations and communications which are carried out by the trustor (and his/her network) to compute a trust value in the trustee. According to Twigg and Dimmock [180], a trust metric is γ-resistant if more than γ nodes must be compromised for the attacker to successfully drive the trust value. For example, the trust metrics used in Rahman's explicit computational trust framework (compared in Section 4.5) is not γ-resistant for $\gamma>1$ (i.e., a successful attack needs only one victim).

Indeed, there are a number of specific attacks due to collaboration, for example, in order to compromise a majority vote. We do not consider real-world identities, which form an alliance, and use their vote to undermine other entities. On one hand, this may be seen as collusion. On the other hand, one may argue that real-world identities are free to vote as they wish. Instead, we focus on attacks based on vulnerabilities in the identity approach and subsequent use of these vulnerabilities. The vulnerabilities may have different origins, for example, technical weaknesses or topological ones (for example, we evaluate novel topologically engineered attacks in Section 7.6.1). However, these attacks commonly rely on the possibility of identity multiplicity: meaning that a real-world identity uses many virtual identities.

Ziegler and Lausen [194] discuss global group trust metrics, which compute a global trust value without taking into account personal bias but require the complete trust network information. For example, Google's PageRank [24] can be considered as one of them, where virtual identities and their contacts are replaced by pages and their hyperlinks. Another type of trust metric takes into account personal bias and is called local trust metric [194]. Local trust metrics have two sub-types [194]: local group metrics and local scalar metrics. The local group metrics, such as Appleseed [194] or Levien's trust metric [117] return a subset of the most trustworthy peers from the point of view of the local trustor over a partial view of the trust network, given the amount of trustworthiness desired. Only Appleseed is compared in Section 4.5 since it appears to be more flexible with attack resistance similar to Levien's one, which in addition needs a number of virtual identities assumed to be trustworthy as an input to the trust metric. Local scalar metrics compute the trust value of a specific virtual identity from the point of view of the local trustor "tracking recommender chains from source to target" [194]. Finally, the computation may be centralised or distributed, meaning that the recommendation received is evaluated before being passed to the successor in the recommender chain.

4.1.2 The Sybil Attack

A very well-known identity multiplicity attack in the field of computational trust due to collaboration is Douceur's Sybil attack [50]. Douceur argues that in large scale networks where a centralised identity authority cannot be used to control the creation of virtual identities, a powerful real-world entity may create as many virtual identities as it wishes and in doing so challenge the use of a majority vote and flaw trust metrics. This is especially important in scenarios where the possibility to use many pseudonyms is facilitated and provided by the trust engine. In fact, a sole real-world entity can create many pseudonyms who blindly recommend one of these pseudonyms in order to fool the trust engine. The level of trust in the latter virtual identity increases and eventually passes above a threshold which makes the decision to trust (the semantics of this depend on the application).

In his draft PhD thesis [117], Levien says that a trust metric is attack resistant if the number of *faked virtual identities*, owned by the same real-world identity, that can be introduced is bounded. Levien argues that to mitigate the problem of Sybil-like attacks it is required to compute "a trust value for all the nodes in the graph at once, rather than calculating independently the trust value independently for each node". Another approach proposed to protect against the Sybil attack is the use of mandatory "entry fees" associated with the creation of each pseudonym [3, 62]. This approach raises some issues about its feasibility in a fully decentralised way and the choice of the minimal fee that guarantees protection. Also, "more generally, the optimal fee will often exclude some players yet still be insufficient to deter the wealthiest players from defecting" [62]. An alternative to entry fees may be the use of once in a lifetime (1L [62]) pseudonyms, a.k.a. pseudonym commitment, where an elected party per "arena" of application is responsible to certify only 1L to any real-world entity, which possesses a key pair bound to this entity's real-world identity. The technique of blind signature [30] is used to keep the link between the real-world identity and its chosen pseudonym in the arena unknown to the elected party. However, there are still three unresolved questions about this approach: how the elected party is chosen; what happens if the elected party becomes unreachable; and how much the users would agree to pay for this approach. More importantly, a Sybil attack is possible during the voting phase, so the concept of electing a trusted entity to stop Sybil attacks does not seem practical.

Bouchegger and Le Boudec envisage the use of expensive pseudonyms [62], cryptographically generated unique identifiers (e.g., CBIDs [1, 130]) and secure hardware modules (e.g., TCPA [174], which is discussed in Section 2.3.1) to counter the Sybil attack. This may overcome the Sybil attack, but at the same time it may exclude poor users as said above according to [62]. Similarly, Kinateder et al.'s workarounds [102] are two-fold. Firstly, some of the risks of pseudonymity are alleviated via TCPA trusted hardware including a trusted CA that would certify the pseudonym without disclosing the real-world identity until legal bodies want to retrieve the link. Secondly, the trust engine should be combined with electronic payment systems, which allow the creation of an originality statement during the payment process which can be included in a recommendation. However, relying on real money turns the trust mechanism into a type of system trust, which is high enough to make the use of interpersonal trust (that is, the trust value) almost superfluous. In the real world, tax authorities are likely to require traceability of money transfers, which would completely break privacy in a money-based system. In this thesis, another solution is adopted (and described in Section 5.5).

A real-world application where the Sybil attack occurs is the email system. The success of spammers has proven it is still cheap enough to create (or spoof) text email addresses, which act as pseudonyms in order to carry out profitable, large-scale spam attacks. It is for this reason that we evaluate our solution in the email and anti-spam domain.

4.2 Remaining ASUP Issues due to Authentication

One of the foundations of security is authentication. Stajano [171] emphasises that without being sure with whom an entity interacts, the CIA security properties can be trivially violated. It underlines the importance of authentication from the security point of view of the ASUP dimensions. Creese et al. [37] agree with Stajano's view [171]: authentication is at the core of security issues in pervasive computing. In this section, we first explain the impacts of naming in distributed collaborative settings and authentication with regard to the usability dimension. It is followed by a discussion on the adaptability of authentication mechanisms.

4.2.1 Naming Issues in Distributed and Collaborative Settings

When only one computer is used, it is straightforward to use unique identifiers to refer to software entities (such as objects). In distributed computing settings, the issue of naming of entities is more complicated, especially when there is no central naming authority. At any rate, collaboration may be required in global computing environments, thus the entities should be able to refer to other entities.

The deployment and management costs of global name/identifier hierarchies, such as X509 [83] distinguished names, are not always viable. Consequently, the first issue related to naming is that it is often required for two different parts of the distributed system to understand that they speak about the same entity: if the two parts have generated local identifiers for the entity under scrutiny, it is very likely that the two local identifiers are different and they fail to realise that they talk about the same entity. Simple Distributed Security Infrastructure (SDSI) [2, 118, 142] and Simple Public Key Infrastructure (SPKI) [54, 118] solve the latter issue through the use of linked local name spaces. "Each principal has its own name space" [118], and can introduce names of its own and issue certificates to define local names. "Name resolution is the process of mapping a principal expression to a global identifier" [2]. Roughly, in SDSI, p_i may be a global identifier, a local name or a compound name. For example, a compound name represented in Abadi's logic [2] is *(Self:*$p_1 \ldots p_n$*)*, which intuitively means p_1*'s…'s* p_n (there are other representations, e.g., [118]).

A second issue related to naming is when the same identifier is picked in two different parts of the system for two different entities. A mathematically random generation of identifiers may decrease this kind of naming collision. For example, public keys may be considered as global identifiers [2] because they are assumed to be chosen with a very high degree of randomness and collisions are highly improbable.

Another issue concerning naming is yet again due to usability. Lampson et al. [111] underline that "when users refer to principals they must do so by names that make sense to people, since users can't understand alternatives like unique identifiers or keys". For example, text email address can be exchanged orally, digital keys cannot. It is even more problematic when a real-world identity has many virtual identities. Depending on context, the appropriate pseudonym must be selected. This may be facilitated in context-aware pervasive computing, for example, if computing entities are instrumented with various sensors. Previous work on identity management in ubiquitous computing environments [86, 115] demonstrated that the model of switching identities according to context is appealing and meaningful for users. However, this gain in privacy protection thanks to multiple pseudonyms [102, 105, 163] is undermined by a cost in terms of usability due to the increase of complexity in managing many identities in many contexts.

4.2.2 Usability and Administrative Tasks at the Authentication Level

Generally, authentication schemes start with enrolment of entities. This task is often time consuming and requires explicit human intervention, such as setting up an account by a system administrator. Enrolling new users may involve considerable work and resources: a random initial secret may be sealed in an envelope and sent to the new user; it can be even more expensive with smart tokens, which can involve two separate activities – token programming and user management [168]. There are already six steps needed for token programming. In the biometrics layer model [124], the enrolment, that is, the first measurement of the biometric characteristics, is crucial and "should be guided by a professional who explains the use of the biometric reader" in order to decrease the number of "fail to enrol" users. The biometric samples are often processed and features templates are extracted. Then, further authentication is usually fully automated.

In fact, once enrolment is complete, authentication often consists of two steps: the requester claims an identity (a.k.a., identification) and the claimed identity is verified (a.k.a, verification). As Smith chose for the title of his book on authentication, authentication techniques have evolved from "passwords to public keys" [168]. There are three main authentication factors: something that you know, e.g., a password; something that you have, e.g., a smart card token; and something that you are, e.g., biometrics. It may also be based on other factors, for example, where you are. Thanks to the context-awareness available in global computing, the "where you are" authentication factor is easier than ever to obtain (for example, based on satellites localisation such as GPS or Galileo).

There are roughly two scenarios of use: either it deals with real humans or it is a remote access. The usability issue of authentication mechanisms is still challenging, for example, on average people can remember five to seven items but they are told to use passwords of more than nine meaningless characters. Most of the time, the enrolment (or bootstrapping step at first meeting) is costly in terms of administrative tasks, for example, a public key cannot be exchanged orally or kept in mind. The cost/benefit of security protection at the authentication level is very important. Smith [168] notes that "people are more worried about having their computers available and usable than they are about password cracking". A real challenge concerning the ease of authentication is to facilitate authentication across multiple and different authority domains: this may be achieved by single sign-on or federated identity management [81, 119, 173], where a trusted-third-party is in charge of managing identity information and ensuring authentication.

The protection by passwords (something that you know) has evolved a lot. However, there are two main categories: cultural history (the rest of the world is unlikely to know but it is not necessarily secret) and random secrets. The usability of Personal Information Numbers (PINs) and passwords undermines their security: on one hand, they need to be changed from time to time and users are asked to use complex passwords; on the other hand, it is difficult for humans to achieve that.

There are protection cost issues in the deployment of authentication tokens (what you have): expensive tokens may be lost or there are difficulties to span different sites with different authorities.

Biometrics (what you are) techniques measure the behaviour or physical traits to authenticate people. In real applications, the False Rejection Rate (FRR) is often greater than *10%*, which is a usability issue for legitimate users [124]. "Sometimes biometric authentication systems replace traditional authentication systems not because of higher security but because of higher comfort and ease of use" [124].

The usability issue of managing and using public keys on a large scale has already shown the limits of approaches based on the exchange of keys that must be bound to real-world identities [64]. For example, in the email domain, previous attempts based on asymmetric encryption and binding of public keys to the identity of the owner of the private key have failed to gain large acceptance and to solve the spam problem. Authentication systems that are designed to run on top of the legacy email system suffer from many usability issues in deployment, use and management. In web of trust style systems [67, 195], the users should carefully check (ideally using an out-of-band channel such as a phone call or attending a key-signing party) that the public key received is really the one sent by the sender due to potential Man-In-the-Middle (MIM[2]) attacks. This is the first example of an identity usurpation attack.

[2] In this thesis, any attack where an attacker can act between the two legitimate entities is called a MIM attack.

CA schemes, such as in Privacy Enhanced Mail (PEM) [99] or based on S/MIME [146], replace the onerous need for individual users to check identities, but the charges imposed by the CA act as a barrier to adoption. In all cases, the public key of senders and receivers must be acquired and validated [64]. There is the problem of bootstrapping/first meeting without a computer at hand, which is a significant feature for the wide-spread adoption of any new solution.

Bootstrapping with the real-world identity might not be needed. Stajano coins the term "anonymous authentication" [171] where "globally unique X.509 'distinguished names' are unnecessary luggage". For example, in SDSI or SPKI, the entity identity can simply be the public key. The public key itself makes declarations by issuing verifiable signed statements that can be certificates or requests for service. In an open and global world, it should be possible to introduce completely unknown authorities: it creates a kind of bootstrapping/newcomer attack, which this thesis tackles in the following chapters. Then, if an a priori infrastructure is not mandatory, personal certification may be envisaged combined with reputation or web of trust [195], which is related to the idea of majority vote and indeed computational trust. Before delving in more detail into the adaptation to a no a priori infrastructure based on computational trust, there are other aspects on the adaptability of authentication.

4.2.3 Adaptability of Authentication

In addition to MIM and bootstrapping/newcomer attacks, there are many other attacks that can flaw authentication because it is a reality that authentication products are not perfect. Ideally, an accurate authentication mechanism consistently rejects authentication attempts by people who are not who they claimed to be during identification, while not rejecting authentication attempts by the true people specified during identification. However, the real-world technical constraints may lead to a trade-off between False Acceptance Rate (FAR) and False Rejection Rate (FRR). For example, in biometrics systems, the yes/no decision is based on a threshold, which must be chosen by the administrator. The protection is manually adapted to the application domain.

A biometric system may operate in verification/authentication mode (when a claimed identity must be verified) or identification mode (when there is no claim of identity and search through all the previously known identities). Jain et al. [85] use the word recognition to encompass the verification mode and the identification mode. In verification mode, positive recognition has the aim to prevent multiple people from using the same identity. In identification mode, negative recognition has the aim to prevent a single person from using multiple identities [85]: this is the first example of identity multiplicity attacks in this thesis.

Tackling Attacks at the Authentication Level

The pattern of combination of authentication mechanisms in order to increase security has been proposed in order to adapt the protection to the growing number of attacks at the identity level and the inherent shortcomings of specific authentication mechanisms. For example, a knowledgeable attacker can generally carry out successful probing attacks on authentication tokens and extract critical data from them [168]. In order to solve the problem of lost tokens, a PIN may be added. It follows multimodal authentication, a.k.a. n-factor authentication (e.g., something you know and something you have). Another example is multimodal biometrics [18, 85], which address the issue of non-universality of the biometric characteristic and decrease the risk of successful usurpation attacks thanks to the combination of many biometrics techniques [85, 124, 168].

Usurpation attacks are most likely to succeed when a lot of information about the users is known. A habitual target for attackers is then the user's personal computer because it contains the knowledge required to carry out successful usurpation attacks. If SBA is possible, the ideal counter-measure may be that "no secret information is stored anywhere, including on the host being protected" [78]. For example, S/Key authentication [78], based on Lamport's key-chains, uses secure hash functions (meaning that it is easy to compute $f(x)=y$ but hard to retrieve x by only knowing y and f) and a sequence of hashes starting with a user-chosen password (plus a seed). The last hash of the sequence becomes the first one-time password stored on the server. Each time authentication is required the server checks that the stored hash is equal to the secure hash of the hash provided by the client. Then, the new hash provided by the client becomes the hash stored by the server. A new sequence is generated when the sequence of hashes has been processed. In doing so, only secure hashes are stored.

In this context, certificate revocations are necessary because private keys can be stolen or broken. However, due to the offline nature of asymmetric cryptography, certificate revocation becomes difficult [171]. This aspect of adaptability may be addressed by: revocation lists, online revocation (although this contradicts the offline philosophy), and certificates Time-To-Live (TTL).

The environmental context may change due to an attack. For example, an attacker starts a trial-and-error attack on the authentication scheme used by the application. There is an implicit trust in the level of protection given by this attacked authentication scheme but the point is that it may fall under attack. In the intrusion tolerance approach [181], some intrusions may be allowed, but tolerated: "the system triggers mechanisms that prevent the intrusion from generating a system security failure". There is a need to a dynamic adaptation to the environmental context.

Making Use of Weak Authentication

An example where explicit (but static) risk analysis is used to tolerate a number of attacks occurs in the domain of authentication and is called "weak authentication" [14, 182]. Weak authentication is not really suitable for applications requiring a link/binding to real-world identities or traditional authentication frameworks such as "PKI or the IETF Authentication, Authorisation and Accounting (AAA) [82]" [14]. However, it may provide means for some form of authentication (that is part of recognition, which is the term used in the framework of this thesis), without pre-shared secrets, previous enrolment to an infrastructure or manual configuration (especially by a technology-aware administrator). There are four main mechanisms [14]: "spatial separation" (e.g., checking whether or not communication can be made on specific paths or channels); "temporal separation" (for example, it is assumed that no MIM is present at first encounter and common information of previous encounters is verified); "asymmetric costs" (that is, the attack is made more costly to the attacker, e.g., the rich targets can only be found at random); and "application semantics" (for example, the identifier is derived from the public key). Finally, these mechanisms can be combined or orchestrated to increase the level of implicit trust in authentication/recognition, which is in line with multimodal authentication. The distinction between authentication and recognition is also made by Weimerskirch and Westhoff [186]. They emphasise that in pervasive scenarios, it is likely that there is no pre-shared secrets or common trusted-third-parties between two complete strangers. They introduce new schemes, called Zero Common-Knowledge (ZCK) schemes, to "recognise" previously encountered virtual identities based on spatial separation and temporal separation. They claim that in such ad-hoc scenarios "recognition is the best we can achieve" [186]. Arkko and Nikander [14] argue that it may be sufficient if the link/binding with the real-world identity is not needed and some attacks are assumed to remain very unlikely (such as the MIM attack). They mention it “cannot achieve as much as other authentication schemes” [186] and it is one of the reasons it consists of an implicit and static level of trust. Generally, a ZCK uses a C/R based on a public part of a secret, which is stipulated at first interaction. It is based on a standard asymmetric encryption public key followed by crypto-based nonce challenges or a new type of public key based on Lamport’s key-chains, which is more efficient for resource-constrained devices. Although the new schemes implemented in this thesis for recognition make use of C/R and hashes, they are both differently combined and orchestrated. In addition, our schemes allow any entity to share their recognition clues in order to relate shared experiences about any entities.

Weak authentication may not always achieve perfect security but may still significantly increase practical security, or is otherwise good enough for the requirements under consideration. Overall, weak authentication may allow for economic trade-offs between security and usability. So, the economic analysis and threat scenarios of the application domains under consideration give a static implicit initial level of trust, which may be regarded as a level of system trust. For example, the targets are attacked at random among a large community (such as, random large-scale spam attacks without marketed database of email addresses). However, if a dedicated attacker targets a specific entity, the level of trust in these assumptions vanishes. If the attack is detected, mechanisms are needed to adapt the state of the entity. As in intrusion tolerance, the system adapts itself and reacts to the detection of an attack. It is said that with zero risk, no trust is needed. Because in weak authentication, the risk of a successful attack “is also quite small, but not zero” [14] (and it may increase when some events are detected) a dynamic explicit level of trust is needed, even if it is a trust level in a technical component of the underlying technical infrastructure.

A final adaptability aspect is to cope with scalability and adapt to the great number of entities expected in global computing environments. For example, a hierarchy of CA can be put in place. The notion of certificate chain implies that the authentication fails if the certificate is not validated by a trusted CA on the chain. A centralised hierarchy is when any chain leads to one single root CA. The other approach is to use of a web of trust, where users issue their own certificates and set levels of trust in certificates sent by other users. Concerning scalability and adaptability to resource-constrained devices, computationally intensive (for example, based on asymmetric cryptography) or high overhead security mechanisms (e.g., using XML-encoding instead of compressed binary or requiring many round-trips for zero-knowledge protocols) may be impossible to be run. This is the reason that the evaluation sections of this thesis contain performance and security overhead results. This thesis also revises the requirements at the authentication level and presents novel approaches for the usability requirement, via dynamic enrolment, in Section 5.1 and the adaptability requirement in Section 5.3.

4.3 Other Remaining Issues Relating to Privacy

In computing security terms, privacy is close to confidentiality or secrecy [72], where only authorised persons have the access to certain information or how information flows [7, 105]. Privacy is linked to intellectual and philosophical ideas [25]. It is difficult (Langheinrich even claims it is impossible [113]) to provide an all encompassing definition of privacy. Privacy can be seen as a fundamental human right "to enjoy life and be let alone" [34] or a basic need (according to Maslow's hierarchy of needs [123]) for a private sphere protected against others. The most sensitive personal information, called Personally Identifiable Information (PII), is directly associated with the real-world end-user identity. According to Tobias and Olsen [121], "personal data is defined in the EC Directive on Data Protection as any information relating to an identified or identifiable natural person. An 'identifiable person' is one who can be identified, 'directly or indirectly' within a reasonable time, considering the necessary effort taking account of all the means likely reasonably to be used".

4.3.1 Privacy Threats Introduced by Global Computing

For example, personal data can be used to build accurate user profiles for marketing and selling purposes. New privacy vulnerabilities have come along with the creation of the Internet, the World Wide Web and the electronic mail system. To some extent, personal information has become a commodity that can be traded in online commerce.

Some invasions of privacy cause annoyance and waste time [192]. For example, information can be used to contact the person with the email address even though the person does not want this specific contact. We see email addresses as PII having the property of easily (i.e., for a very low cost) and effectively (i.e., an email delivered in the Inbox will surely obtain human attention, even if it is only for a fraction of time) making contact with the related human. The privacy violation that occurs when a third-party obtains the email addresses of some people without their consent, is usually followed by unsolicited messages – known as spam – sent on the open communication channel associated with the email address, that is, the standard electronic mail system. The cost of email privacy violation varies a great deal between users, but the overall cost is known to be very high [40].

With the advance of online worldwide social networking services, such as Friend-Of-A-Friend (FOAF) [59], it is even possible to build the social network topology between the persons (as it is done in this thesis, which also evaluates related network engineered attacks due to collaboration and identity multiplicity).

Biometric authentication techniques bring their own privacy threats fear. Biometrics data contain a lot of covert channel information [85, 124], for example, preposition to disease in DNA biometrics. The identification mode is the worst from a privacy point of view because the system does not take into account whether the user tries explicitly to hide his/her identity. However, this mode is more convenient than the verification mode since the user has not to claim an identity.

In ubiquitous computing environments, sensors will possibly sense private information anytime anywhere. Later on – possibly after a very long time – parties that stored this information will be able to provide access to this information anytime anywhere to probably anyone who is able to pay a small fee as the cost of the technology decreases. Data-mining and complex computation will then enable the correlation of data from different sources, locations or periods of times by identifying hidden patterns, allowing the building of fine-grained profiles. The installation of a plethora of invisible sentient sensors creates privacy issues that are considered to be impossible to solve technologically, especially when faced with determined attackers [112, 113]. In fact, the state-of-the-art in ubiquitous computing security and privacy [28, 163] assumes that limited schemes are used to recognise the users. For example, the assumption made for the privacy protection mechanisms developed for Gaia smart spaces follows: “We also assume that the spaces supporting our privacy system would not contain surveillance cameras or voice recognition devices, otherwise, users will have to take additional physical precautions to protect their privacy, like wearing masks or staying silent!” [13].

Therefore, in global computing both individual's private information and the private information of the members of his/her network of collaborators are threatened. Langheinrich has proposed six useful principles for guiding the design of privacy protecting mechanisms in ubiquitous computing [113]: notice; choice and consent; anonymity and pseudonymity (that should be provided by default and support the fact that means should be left to the users to be in control of their private data); proximity and locality (it may be sufficient in some applications to rely on the locality of the real-world parties to grant or deny access); access and recourse; adequate security (which emphasises that the security solution should be adapted to the risk involved and the resource constraints).

4.3.2 Adaptability to Privacy Changes and their Usability Issues

We first consider legislative means for privacy protection: going from country-specific legislations to higher level legislations (for example, European Union (EU) 95/46/CE [56], which deals with the quite worldwide principle of collection limited to the mandatory required data for the purpose [121]). However, legislation has shown its limitations: there may be multiple contradicting jurisdictions and, even in the same jurisdiction, privacy protective laws [121] can be reversed to excessive data retention [44, 57]. Non-physically enforced approaches, based on privacy policies and the good will of the contractor [36], have also shown their limitations: for example, there are known cases where millions of airline passengers information was disclosed in violation of the airline stated privacy policy in order to support research in data mining and screening systems [55].

An important aspect of privacy is that people have dynamic privacy expectations: "our privacy needs change almost constantly in response to our desire to interact with one another and social moral and institutions affect privacy expectations" [25]. The change of the EU position regarding privacy protection [44, 57, 121] is one example of this dynamic aspect. Privacy is a trade-off "with efficiency, convenience, safety, accountability, business, marketing, and usability" [87]. Privacy is a constant interaction where information flows between parties [87, 131]. Privacy expectations vary based on context changes [163]. Adaptability is another time required.

This great number of changes would require changes to privacy policies. However, to set up and tune policies takes time and effort. Agrawal [11] has questioned if users will be able to tune adequately their policies or simply if, busy as they are, they will make the effort to set up their policies. Frequent and time consuming configurations for privacy protection run counter to the requirement that ubiquitous computing should not monopolise the attention of the user [188]. User centric security is a requirement in ubiquitous computing environments [144]; security mechanisms should distract the user as little as possible. This concerns usability.

Currently, there is active research on privacy enhancing technologies (PET) [15, 16, 30, 31, 141]. The main technological line of defence, used in this thesis, is to use virtual identities – pseudonyms. The ordinary definition of a pseudonym is "a fictitious name used when the person performs a particular social role"[3]. Others [71, 79, 105], in other domains than computational trust, have presented how pseudonyms can be used for privacy protection and shown that different levels of pseudonymity and configurations exist. Their work is valuable in the decision to choose the appropriate type of configuration and pseudonymity for the purpose of trust engines.

4.4 Identity Terminology

From the survey of the literature, there are three main categories of terms related to identity and the authentication process: the entity, the virtual identity and the real-world identity.

The real-world identity spans a broad panel of resources: software objects, software credentials, agents, files, file systems, processes, printers, machines, nodes, network resources, people, organisations [19, 95]. However, when a computing entity is requested by a real-world identity, the link to the real-world identity is initially unknown and the authentication process is started. Based on distinguishing characteristics, the requesting entity is associated with the virtual identity. The link with the real-world entity is usually either known or unknown.

The virtual identity is often called a "principal" [49, 66, 147] and may be abstracted to pseudonym [43, 105, 136]. The real-world identity generally corresponds to "subject" [109, 136]. In the Java Authentication and Authorisation Service (JAAS) [109], based on Pluggable

[3] Definition from WordNet Dictionary:
http://www.hyperdictionary.com/search.aspx?define=pseudonym

Authentication Module (PAM) [148], once the source of a request is authenticated, the subject is associated with identities or principals (for example, a name principal "John Doe" or a SSN principal "123-45-6789"). In JAAS, the authentication implies a true or false association. A JAAS subject may also own security-related attributes – a set of credentials – split into private (more protected and access restricted) credentials and public credentials.

In fact, from explicitly disclosed credentials or implicit correlation of information [43, 189], such as usage data [105], it is possible to form profiles and define virtual identities that can possibly be linked to real-world identities. Ian Goldberg [71] underlines that any transaction engaged in by a person reveals meta-content, especially information about the identity of the person. He defines "the nymity of a transaction to be the amount of information about the identity of the participants that is revealed" and gives a continuum, called the "Nymity Slider", with different levels of nymity: verynimity (e.g., government issued id), persistent pseudonymity (e.g., pen names), linkable anonymity (e.g., prepaid phone cards), and unlinkable anonymity (e.g., based on Chaum's mix technique [31]).

Figure 5 summarises the different terms found in the literature and organises them in the three main levels. In Figure 5 (as in Figure 7, which is a revised version) the red left circle represents the first level of terms, summarised as entity; then, after identification and authentication, the terms in the orange middle circle are used, summarised as virtual identity; the green right circle contains the terms summarised as real-world identity. The white arrows depict the flow from entity to virtual identity to real-world identity. The second white arrow mentions Yes/No to underline that a virtual identity may or may not be linked to a real-world identity. The first arrow can be passed thanks to authenticated distinguishing characteristics.

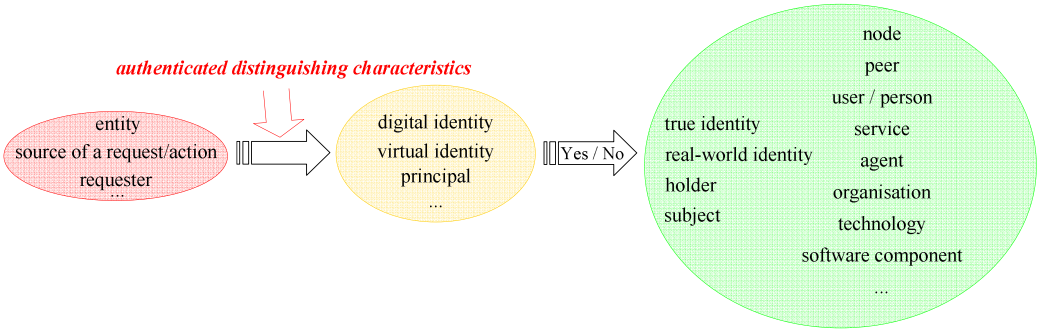


Figure 5. Identity Terms Categorisation

From Figure 5, it becomes clear that a unique real-world identity may use different pseudonyms *to "speak for"* [2, 111] it without revealing the link, i.e., which is unknown to other entities. In doing so, different profiles may be used in different contexts. As it is hard to link different profiles, it is much more difficult to create an accurate global profile of the owner. Previous work on unlinkability [30, 107] and anonymity [9, 10, 136] provides means to evaluate the level of privacy protection reached. For example, JAAS [109] specifies that a subject may have several principals. Therefore, in this thesis, a real-world identity may indeed have multiple pseudonyms or virtual identities.

4.5 Identity and Trust Frameworks Comparison

We first describe the identity and trust frameworks that are most related to this thesis. Then, we compare them with regard to the ASUP requirements.

4.5.1 From Proven Related Legacy Frameworks to Pervasive Ones

The first two frameworks can be categorised as adaptable authentication and authorisation frameworks, especially because they are based on standards describing how to plug different authentication schemes. JAAS is a framework and programming interface that enforces in the Java^{TM} platform access control based on the identity of the user who runs the Java application code. The Extensible Authentication Protocol (EAP) [8] is a P2P authentication framework (described in a RFC [8]), which supports multiple authentication methods and takes into account their security strength according to a list of attacks at the authentication level and a static risk analysis.

Then, we review three credentials-based authentication and authorisation frameworks. It starts with the very influential identity-based framework, PGP [195]. The next framework is Lampson et al.'s framework [111, 191], which relies on "speaks for" relations between a large panel of types of virtual identities (for example, communication channels or roles). They evaluated their framework thanks to the implementation of a few prototypes and performance evaluation, as we do in this thesis. They note that the performance of their framework depends on the cost of the cryptographic operations (in the evaluation of this thesis, we also discuss this cost). They define a convenient API, whose parts are given below in a way closer to the terminology used for the design in this thesis:

```
void send(Address dest, VirtualIdentity vI, Message m)
Channel getChannel(Address dest)
SubChannel getSubChannel(Channel c, VirtualIdentity vI)
VirtualIdentity getVirtualIdentity(SubChannel subC)
boolean grantAccess(ACL acl, VirtualIdentity vI)
Credential sign(Credential cred, VirtualIdentity vI)
```

The third advanced credentials-based framework is ATN [190]. In addition to what has been said in Section 2.3.1, an other interesting element of this framework is that it takes into account the privacy aspect of credentials released thanks to negotiation.

The comparison continues with two ubiquitous computing frameworks. Al-Muhtadi et al.'s framework [12], which we call *Jini/SESAME*, tackles the security issues in a smart home with the plug and play technology based on Java called Java Intelligent Network Infrastructure (JINI), which is extended with a light-version of the Secure European System for Applications in a Multi-vendor Environment (SESAME) [166]. SESAME is itself grounded on Kerberos. Kerberos has been very influential (it has been turned into an Internet standard RFC1510). It provides a mechanism to authenticate and share temporary secret keys between cooperating processes. Forgery or replay attacks are defeated if a protocol such as Needham-Shroeder [149] is used thanks to nonces (or timestamps introduced to solve the reuse of a session key) and C/Rs. The second ubiquitous computing framework is Gaia. The environment of Gaia [28] goes beyond the previous environment of a single smart home. For example, it may be a smart campus. However, there is still the assumption that an infrastructure is present and that technology-aware and dedicated administrators manage this infrastructure. So, it is not yet Weiser's vision of calm ubiquitous computing [188]. An interesting point is that the administrator must set the "confidence" in the security protection given by the different possible authentication schemes. For example, active badge authentication is given *0.6* on a *[0,1]* scale. Depending on how many authentication means the user uses to authenticate his/her pseudonym, the final confidence p_n increases or decreases according to the following formula, where n authentication schemes are used and p_i is the confidence of a successfully passed authentication scheme:

$$p_{final} = 1 - (1 - p_1)\ldots(1 - p_n)$$

Another innovative element is to use this final confidence and other context information in the rules used for authorisation decisions.

A related framework using biometrics is Shakhnarovich et al.'s framework [167], where face and gait recognition are integrated. They empirically found based on experiments on twelve users that multi-modal recognition increases the level of confidence in recognition. Since some recognition schemes may be more or less accurate, they first empirically evaluated the success rates of face and gait classifiers. They found they were similar but face recognition may simply be impossible due to not enough captured images containing the face. Therefore, they computed multi-modal confidence vector according to the following formula given the observed sequence of multi-views *x*:

$$p_{combined}(x) = \left\{ \begin{array}{c} p_{gait}(x), no\_face \\ \left(\dfrac{p_{gait}(x) + p_{face}(x)}{2} \right), otherwise \end{array} \right\}$$

4.5.2 Evidence-based Frameworks

Concerning explicit evidence-based computational trust frameworks, the six most relevant ones, present in three surveys [120, 140, 177] published at the time of writing this thesis, have been selected to be compared according to the same specific points. They are called with the following short names: *YKBB* for [17, 193]; *Rahman* for [4, 6, 140]; *Marsh* for [122]; *SULTAN* for [74-76]; *Jøsang* for [89-91] and *Kinateder* for [102, 103].

In Yahalom et al's framework [17, 193] (YKBB), entities communicate via channels and have a unique identifier and a secret, which can be used for authentication. Some of these entities are supposed to have the role of authentication servers. Each virtual identity is assigned several trust values based on probabilities, which are computed from a number of counters in different trust contexts. These counters are incremented or decremented each time there is an action and its associated outcome (positive or negative) related to the trust context.

Marsh's PhD thesis framework [122] is one of the first computational models of trust based on social research. Each trust context is assigned an importance value in the range *[0,1]* and utility value in the range *[-1,1]*. Any trust value is in the range *[-1,1)*. In addition, each virtual identity is assigned a general trust value, which is based on all the trust values with this virtual identity in all the trust contexts. Dispositional trust appears in the model as the basic trust value: it is the total trust values in all contexts in all virtual identities with whom the trustor has interacted so far. Risk is used in a threshold for trusting decision making.

The Simple Universal Logic-oriented Trust Analysis Notation (SULTAN) [73-76] framework provides a simple notation for the writing of policies dealing with trust and recommendation concepts. A tool kit is available to specify, analyse and monitor trust specifications. It is dedicated to Internet applications. The risk evaluation service allows the trust policy to take into account risk information.

Rahman's PhD thesis framework [4-6, 140] has evolved a lot and the final version is not finished at the time of writing. It consists of a decentralised trust model based on social research. It focuses on the formation and evolution of trust values based on recommendations and direct observations rather than the use of trust values for decision making. It is one of the reasons that no risk component is present in the framework. There are different levels of trust from very untrustworthy to very trustworthy. There are two main contexts for the trust values [138]: "direct", which is about the properties of the trustee; and "recommend", which is the equivalent to recommending trustworthiness. Trust contexts for direct trust values may be possible. Recommending trustworthiness is based on consistency on the "semantic distance" between the real outcomes and the recommendations that have been made. The default metric for consistency is the standard deviation based on the frequency of specific semantic distance values: the higher the consistency, the smaller the standard deviation and the higher the trust value in recommending trustworthiness. However, different rather arbitrary choices had to be made to map the consistency to trust values. In case of unknown virtual identities, the initial trust value may be based on other means, such as, related to dispositional trust (as in Marsh's framework above). The "recommendation protocol" [6] is used by a trustor to find recommender chains about the trustee. The default recommendation search scheme is related to a depth-first search directed by the recommending trustworthiness of the recommenders if they do not know the subject.

Jøsang's framework [89, 90] is called "subjective logic" and integrates the element of ignorance and uncertainty, which cannot be reflected by mere probabilities but is part of the human aspect of trust. In order to represent imperfect knowledge, an opinion is considered to be a triplet, whose elements are belief (*b*), disbelief (*d*) and uncertainty (*u*), such that:

$$b + d + u = 1 \qquad \{b, d, u\} \in [0,1]^3$$

The relation with trust evidence comes from the fact that an opinion about a binary event can be based on statistical evidence. Information on posterior probabilities of binary events are converted in the *b*, *d* and *u* elements in a value in the range *[0,1]*. The trust value (*w)* in the virtual identity (*S*) of the virtual identity (*T)* concerning the trust context *p* is:

$$w_{p(S)}^{T} = \{b, d, u\}$$

The subjective logic provides more than ten operators to combine opinions. For example, a conjunction ($\wedge$) of two opinions about two distinct propositions determines from the two opinions a new opinion reflecting the conjunctive truth of both propositions. The consensus of independent opinions is equivalent to two different virtual identities using the total of their independent direct observations to generate a new opinion. The recommendation ($\otimes$) operator corresponds to use the recommending trustworthiness (*RT*) to adjust his/her recommended opinion. If a recommender chain is used, Jøsang strengthens that opinion independence must be assumed and transitivity is allowed only if the recommenders do not recommend different trust values about the same virtual identity depending on the source of the request for recommendation. Noticeably, there is no risk component. Jøsang's approach can be used in many applications since the trust context is open. One of his applications [91] is especially relevant to this thesis since it deals with authentication using public keys in open environments. He argues that most of previous related work based on trust values does not take into account the binding between the real-world identity and the public key and/or the element of uncertainty. Although recommender chains are possible, we only give the formula that he proposes to compute the trustworthiness of the binding between the real-world identity (*S*) and the public key based on a recommendation from a trustworthy recommender (*R*):

$$w_{Binding(S)}^{T} = \left(w_{BindingRT(R)}^{T} \wedge w_{Binding(R)}^{T}\right) \otimes w_{Binding(S)}^{R}$$

In the case of a recommender chain, a recommendation must specify the recommending trustworthiness and the binding trustworthiness. Only direct observations must be passed in order to avoid opinion dependence due to the overcounting of evidence. In case of multiple recommender chains, the consensus operator should be used. However, Twigg and Dimmock [180] explain that the consensus operator, which combines the opinions as if they were observed independently, allows the trust metric to be driven to a trust value chosen by an attacker due to an identity multiplicity attack.

In the framework that we call Kinateder [102], the trust value can be contextualised to the "trust category" (trust context) of concern of the current request (for example, with regard to book security expertise). Their recommendations can be considered as trust values contextualised to a particular trust context. A view rarely discussed in other frameworks, which is also argued in this thesis, is that "trust categories are not strictly independent but they are influencing each other" [102]. Another view, which has also been overlooked by other frameworks, is that privacy is required (as argued in this thesis) and users must be allowed to use virtual identities to minimise the risk of profiling them due to the history of their transactions, which are pieces of evidence required to compute their trust value. However, in Kinateder and in contrast to this thesis, one virtual identity should be used per trust context. Their approach works fine within their infrastructure where management tools are provided but they do not explained how authentication schemes different than key pairs would fit in their framework. There is no risk analysis in their trust model, especially with regard to the technical trustworthiness of their mechanism. Their system is still prone to the Sybil attack, since a real-world identity could create an arbitrarily large number of faked virtual identities, then copy recommendations from recommenders with high trust values and present these as recommendations from the just created faked virtual identities. As said in the above section on the Sybil attack, their workarounds (e.g., [103]) involve system trust. Trusted-third-parties are likely to be needed if real-world identities must be bound to the keys, if trusted hardware is needed or when they introduce payment.

4.5.3 Real-World Social Networks-based Frameworks

The next frameworks are based on real-world social networks, where users set manual trust values to other users. None of the frameworks explain how the manual trust value could be converted into a trust value based on event outcomes.

For example, the Friend-Of-A-Friend (FOAF) [59] initiative can be described as a text format for the online description of the profile of a user (name, contact information, interests…) and links to the profiles of users that he/she knows. Generally, social networks bring interesting properties since they exhibit the "small-world" [77, 179, 184] phenomena, whereby the diameter (meaning the greatest number of hops between any two users) of the network increases logarithmically with the network size and users can be reached in few hops.

Milgram's experiment [179], which ended up in the small-world theory or at maximum six degrees of separation between any two persons in the world is well-known.

In Golbeck et al. [68, 70]'s framework, that we call Golbeck, the FOAF schema has been extended to include "trust assertions", that is, manual trust values in the individuals targeted by these assertions. They use their network for a kind of anti-spam tool. TrustMail is not supposed to be a spam filter but a layer used to "provide higher ratings to emails that come from non-spam senders" [69]. Emails from known, trustworthy senders are given higher priority in the user's inbox whereas emails from unknown or disreputable senders are given lower priorities. Email senders are given a manual trust value in the range *[1,10]*; with *1* meaning that the recipient has little or no trust in the sender and *10* meaning that the recipient trusts the sender maximally. In case an email is received from an unknown sender, the recipient attempts to infer the equivalent of a manual trust value for the sender based on recommender chains starting with the recipient's known contacts. Using a Breadth First Search (BFS) in the whole network stored on the TrustMail centralised server, a path from the recipient to the sender is searched for and manual trust values on the path are combined.

Ziegler and Lausen's trust metric, called Appleseed [194], is possible since it is assumed that all users make their manual trust values publicly available. This constitutes a threat to privacy since a clear view of the network of acquaintances can be obtained. The manual trust value ranges from *0* (lack of trust) to *1* (blind trust). Recommending trustworthiness does not explicitly appear but it is reflected in the choice of the "spreading factor" [194], which is recommended to be set to *0.85* and "may also be seen as the ratio between direct trust […] and trust in the ability […] to recommend others as trustworthy peers". Appleseed spreading factor highlights that trustworthiness gained by a virtual identity may be passed to another virtual identity. However, this factor is not set by the recommender but by the computing trustor and similar for all virtual identities in the network. In fact, Appleseed is a local centralised group metric. They evaluate their work by simulations based on real social networks extracted from an online community Web site and discuss the attack resistance of their metric (as it is done in the evaluation of this thesis).

4.5.4 Peer-to-peer Frameworks

The last type of frameworks covered are ones based on decentralised P2P techniques [46, 126]. They can be built based on two approaches. First, there are unstructured networks, where no index information is maintained, messages with search information (such as, TTL, message identifier or list of already contacted peers) flood the network. Their performance is roughly [45]: search latency; storage and update costs low; good resilience to failures; and high messages bandwidth. Secondly, there are structured networks, where index information is distributed and maintained among the peers according to different solutions (for example, based on a distributed hash table or a binary search tree). Their performance is roughly [45]: a logarithmic search but higher storage and update costs (for example, due to routing tables or replication). From a privacy point of view, the second approach is likely to imply that trust evidence of a user would be maintained by other users, who cannot be chosen by the user. When the network consists of a social network (as in the frameworks of the previous section), a variant of the first approach can be used. Thanks to the assumed small-world properties of the social network, the search may be optimised by directing the search according to properties of the target of the search and properties of the direct peers [77]. For example, if an email about movies has been sent by a previously unknown sender, the search for information about the sender would start with the contacts that are known to have an interest in movies. In this section, three peer-to-peer frameworks with adjunct computational trust are reviewed.

According to Ziegler and Lausen's trust metric classification [194] (explained above), the trust metric used in the Eigentrust [96] framework is a global distributed group metric. The trust values based on the number of positive and negative outcomes are normalised according to the following formula (trust value *c* of virtual identity *i* in virtual identity *j*):

$$c_{ij} = \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)}, \quad s_{ij} = NumberOfPositiveObservations - NumberOfNegativeObservations$$

They depend on one (or several) structured P2P networks for the trust value computation. They use the global trust value to increase the quality of P2P file sharing systems (based on an unstructured P2P network). They evaluate their work on a simulated network constructed according to power law and have a threat analysis, where different strategies are used by a number of malicious peers.

Damiani et al. [41, 42] (the Damiani framework) add a computational trust metric on top of a P2P unstructured network. The searching follows Gnutella's flooding technique [60], which consists of sending an identified search request message to a number of direct contacts with a TTL. Their first application domain is file sharing. The trust metric is used to choose the most trustworthy peer among the peers who claim to have the sought-after file. In order to minimise the risk of Sybil attack, recommendations coming from a clique of IP addresses are discarded. Similarly, a number of recommenders are re-contacted to check that they really meant the recommendation and it is supposed to increase the cost of running faked virtual identities by the same real-world identity. It is also an example of weak authentication (detailed in Section 4.2.3) and related to the schemes developed in this thesis. They evaluate their work by a discussion on the communication overhead (as we do) introduced by the collaboration for computational trust on top of the file sharing system (since "usually, the limiting resource in P2P networks is network bandwidth rather storage" [41]). The second application [42] reuses their Gnutella-based computational trust to fight spam in email settings. In order to protect the privacy of the users of a mail server, only the mail servers are considered to be peer in the unstructured network. The mail server aggregates direct observations of its email users about spam emails. Since it is common that spam emails are slightly modified, a fuzzy hash mechanism is used to give the same hash for slightly different spam emails. The peers send updated collection of hashes of spam emails, without reference to the involved email users, to another type of peers, called super-peers. The super-peers maintain a distributed collection of spam hashes and peers can query information about unknown emails. The result of the query is a number of recommendations that are used to compute the final trust value based on the recommenders trustworthiness and a trust metric, whose choice is left to the future users.

Sierra is the implementation of the OpenPrivacy computational trust management framework [26, 133]. Sierra is composed of: the “Nym Manager”, which creates, manages and authenticates the pseudonymous certificates; the “Reputation”, which is signed by the current local virtual identity and used as recommendation or observation; the “Reputation Calculation Engine (RCE)”, which implements the trust metric, computes and maintains Reputations; the “Query” package to query and index data; the “Communication” interface for transparent communication with peers (the type of P2P network can be plugged with this interface); and the “Storage Manager”. Any trust metric could be used as long as there is an RCE implementation of the trust metric. Noticeably, there is no risk component in their framework. The “Nym Manager” has many interesting features although it is limited to public keys for authentication. According to the trust context, different virtual identities can be used. It is possible that a parent virtual identity generates different child virtual identities [108]. It is not clear how they would implement the automatic selection of the appropriate virtual identity according to the current context. Generally, the specifics of their framework are left undefined. They underline that long-lived virtual identities are preferable in order to be granted interactions requiring a great trust value. In this thesis, we introduce how to combine trust values of different virtual identities once a link has been proven between them. The final contribution of their framework is in the use of certificates of recommendations, called “gifts” [26], carried by the trustee. It is useful in scenarios where the recommender might become unreachable. It is supposed to work in a fully decentralised manner (but this will depend on the communication type chosen and the trust metrics).

4.5.5 ASUP Qualitative Frameworks Comparison

From the previous chapters, the ideal framework based on identity and trust addresses the four ASUP dimensions. In order to get a finer grained comparison of the related frameworks, each ASUP dimension is given different points of comparison, listed below, including the types of questions that are related to the points. These points of comparison are extracted from the previous chapters of this thesis and correspond to the main points addressed by the framework in this thesis. The purpose is to show that the framework in this thesis covers points that other frameworks only cover in part, as evident in Table 1. These points are not exhaustive and may overlap because the ASUP requirements intermingle. It is why each point and associated questions must be considered according to its heading ASUP requirement.

Concerning the *adaptability* points of comparison:

1. *to the available technical infrastructure and scalability:* can more or less resource consuming authentication and authorisation schemes be used?; can it handle a large number of virtual identities?;
2. *to the trade-off between security protection and its tolerable cost*: are there means to limit privileges when weaker technical infrastructure is available?; is there dynamic risk (cost/benefit) analysis?;
3. *to the context*: beyond adapting the decision based on virtual identity, can it be adapted to slightly different application context, for example, from trustworthiness for the driving of a car to the driving of a motorcycle?; is the system able to adapt its response to the environmental context, for example, under attack or when normal conditions are perceived?;
4. *to the use of many virtual identities*: is there a problem or a difference when multiple identities per real-world identity are used?; can it handle it without failing?
5. *to new security domains*: are human administrators needed when new security domains are encountered?; can new application domains be used?

Concerning the *security* points of comparison:

1. *cost of the management overhead to be done by the user:* is the user supposed to administer the system?; how often is manual reconfiguration needed?; how often is explicit user intervention needed?;
2. *openness to newcomer*: do newcomers need to be enrolled by a human administrator?; do they have to pay first?; are they considered untrustworthy by default?; what do they have to do to increase their privileges?; is there a mechanism to build trust from scratch?;
3. *through collaboration*: is collaboration and computational trust used to decide about security decisions?; is it only for mere selection of the most trustworthy virtual identities (that is, collaborative filtering)?;

4. *against identity multiplicity attacks*: how does the framework behave against this type of attack?; is the Sybil attack possible?; how is the Sybil attack mitigated?;
5. *against identity usurpation attacks*: how does it behave against these attacks?; what is done against spoofing?; are SBA and MIM attacks discussed?;
6. *uncertainty consideration*: is it assumed that everything is perfect and binary?; what may be uncertain?;
7. *explicit evidence-based trust levels*: are there trust levels?; are they explicitly computed based on evidence?;
8. *based on standards*: has the solution been peer-reviewed?; is it part of a large consortium process?; are old standards reused or new ones accepted?;
9. *possibility of full decentralisation or presence of trusted-third-parties:* are trusted-third-parties mandatory?; what changes when they are available?; is it supposed to work in a fully decentralised manner (although it might fail under special circumstances, such as attacks)?;
10. *clear separation between authentication and authorisation:* is there clear separation between authentication and authorisation?; does identity matter?*;*
11. *mandatory assumption of effective system trust:* does it rely on the accountability of the real-world identity and successful prosecution?; is there the assumption that the link to the real-world identity guarantees a successful prosecution?; does it rely on real-world recourse such as insurance?

Concerning the *usability* points of comparison:

1. *of bootstrapping/enrolment:* how dynamic are the possible enrolment schemes, especially from an authentication point of view?;
2. *of management of multiple identities*: are there features to ease the management of multiple identities?;
3. *of the specification of privacy policies*: how easy is it for a user to specify privacy policies?;
4. *of the specification of trust policies*: how easy is it for a user to specify trust policies?

Concerning the *privacy* points of comparison:

1. *pseudonymity*: is it possible to use pseudonymity or many virtual identities?;
2. *link to the real-world identity*: is it mandatory?; how hard is it to infer the link?;
3. *negotiation*: is it possible to negotiate the amount of private information to be disclosed?;
4. *user-centric and in control*: is the user in control of his/her privacy?

The points of comparison and their number are summarised in Figure 6. The following Table 1 presents the comparison of the different frameworks according to the points of comparison determined in this section.

In addition, for all the compared frameworks, there are a number of shared facts. None of the surveyed frameworks really solves the issue of SBAs. SBAs are indeed very harmful in security through collaboration since compromising one entity may impact its collaborating entities. Further trust dynamics research is needed to overcome this type of attacks. Concerning usurpation attacks, if addressed, the general workaround is to use asymmetric cryptography and to sign transactions and messages. Finally, at time of writing, there is no computational trust standard.

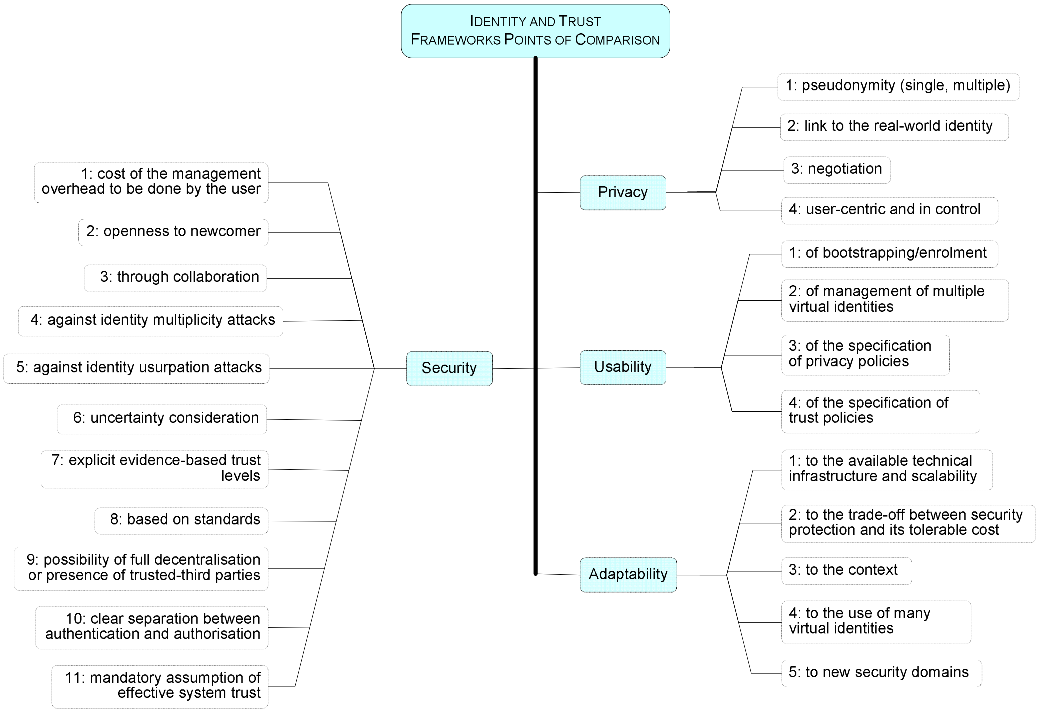

Identity and Trust Frameworks Points of Comparison
Privacy
1: pseudonymity (single, multiple)
2: link to the real-world identity
3: negotiation
4: user-centric and in control
Usability
1: of bootstrapping/enrolment
2: of management of multiple virtual identities
3: of the specification of privacy policies
4: of the specification of trust policies
Adaptability
1: to the available technical infrastructure and scalability
2: to the trade-off between security protection and its tolerable cost
3: to the context
4: to the use of many virtual identities
5: to new security domains
Security
1: cost of the management overhead to be done by the user
2: openness to newcomer
3: through collaboration
4: against identity multiplicity attacks
5: against identity usurpation attacks
6: uncertainty consideration
7: explicit evidence-based trust levels
8: based on standards
9: possibility of full decentralisation or presence of trusted-third parties
10: clear separation between authentication and authorisation
11: mandatory assumption of effective system trust

Figure 6. Identity and Trust Frameworks Points of Comparison

Point of Comparison: strongly addressed (●) approached (○)		JAAS	EAP	PGP	Lampson et al.	ATN	Jini/SESAME	Gaia	Shakhnarovich et al.	YKBB	Marsh	SULTAN	Rahman	Jøsang	Kinateder	Golbeck	Appleseed	Eigentrust	Damiani	OpenPrivacy
Adaptability	1	○	●				○	○	○		○			○			○			○
	2		○		○					○	●	●					○			
	3	○	○		○		○	○	○	○	○	○	○	○		○			○	○
	4	●			○		○	○							○			○		
	5	○	○	○	○	○							○			○	○			
Security	1							○	○			○	○	○		○	○			○
	2			○					○	○	○		○	○	○	○		○	○	
	3			●	○					●	○	●	●	●	●	○	●	●	●	●
	4													○	○		○	○	○	
	5	○		○	○	○							○	○	○				●	○
	6	○	○	○				○				○		●						
	7		○	●		○		○	●	●	●	●	●	●	●	●	●	●	●	●
	8	●	●	○			○	○								○				
	9		○	○						○			○	○			○	○	○	○
	10	●			●		●	●	○					○	●					
	11									○	○	○	○			○	○	○	○	○
Usability	1	○	○		○			○	○		●	○			○					
	2	○			○										○					○
	3					○				○	○	○	○							
	4			●	○	○				○	○	●	○	○		○	○			○
Privacy	1	○			○	○		○				○			●				○	●
	2	○	○		○	○				○	○	○	○		●				○	●
	3		○			●		○												
	4			○		○		○					○		○				●	●

Table 1. ASUP Comparison of Previous Frameworks

4.6 Summary

The current approaches to identity in trust engines are too simplistic and contribute to why trust engines do not fulfil the ASUP requirements. Authentication is partly responsible for the need of an administrator, especially due to enrolment, which is usually costly in terms of manual tasks. In addition, the fact that the notion of real-world identity has been central for security based on authenticated identity and legal recourse contributed to the administrative overhead of authentication due to the difficulty to link a real-world identity with a virtual identity. From the requesting entity, authentication is used to point to the virtual identity, which may be linked to a real-world identity. A real-world identity may have many different virtual identities. The link between a real-world entity and a virtual identity varies from unknown to known to other external entities. Privacy suffers from this potential link between the real-world identity and the virtual identity. Privacy protection also contributes to the administrative overhead, especially in global computing where the context keeps changing and the number of entities is so huge. Usability is further challenged due to naming issues both for humans and between distributed computing entities, which need to collaborate.

Trust engines must make sure that these collaborations are not flawed due to a number of attacks that can be carried out by potential attackers. There are two main high-level attacks at the level of identity: identity usurpation (meaning that a legitimate trustworthy virtual identity is compromised and the attacker can make false statements under the name of this compromised virtual identity) and identity multiplicity (meaning that a real-world identity uses many virtual identities).

The frameworks comparison showed that a few evidence-based computational trust frameworks approach the requirement of privacy but do not reach (or clearly give) a solution for greater adaptability and usability. A few others are more usable and adaptable but they do not take into account their technical security strength, attacks and threat analysis. Legacy adaptable frameworks, at present, do not integrate explicit evidence-based trust engines. The frameworks are generally evaluated according to performance and security analysis. Although there is no global computing environment available for empirical evaluation, two main application domains are usually used: the smart home and email anti-spam, with an emphasis on the network of email users. The next chapter describes the entification framework, which fills this gap between identity and computational trust.

CHAPTER 5: THE ENTIFICATION FRAMEWORK

In this chapter, the high-level view of the framework developed in this thesis, called entification, which combines identity and computational trust approaches, is given. Firstly, the authentication process is revised based on the notion of recognition to increase dynamic enrolment and auto-configuration. This is followed by the description of the integration of recognition into a trust engine and the means to increase adaptability. Then, we detail how the framework explicitly supports the possibility to use multiple virtual identities per user to protect privacy and to trade privacy for trust. Finally, means to mitigate flaws and attacks at the level of identity are presented.

5.1 Recognition rather than Authentication

In Chapter 4, it is explained that more usable authentication is required and enrolment is especially important to achieve this goal. How enrolment is done in current computing systems is not satisfactory for global computing because it requires a lot of manual administrative work and global computing aims at Weiser's "calm technology" [188].

To allow for dynamic enrolment of strangers and unknown entities, we propose an entity recognition process. Table 2 compares the current Authentication Process (AP) with our Entity Recognition (ER) [157] process.

Authentication Process (AP)	**Entity Recognition (ER)**
A.1. Enrolment: generally involves an administrator or human intervention	
A.2. Triggering: e.g., someone clicks on a Web link to a resource that requires authentication to be downloaded	E.1. Triggering (passive and active sense): mainly triggering (as in A.2), with the idea that the recognising entity can trigger itself
A.3. Detective Work: the main task is to verify that the entity's claimed identity is the peer's	E.2. Detective Work: to recognise the entity to be recognised using the negotiated and available recognition scheme(s)
	E.3. Discriminative Retention (optional): "preservation of the after effects of experience and learning that makes recall or recognition possible"[4]
A.4. Action: the identification is subsequently used in some ways. Actually, the claim of the identity may be done in steps 2 or 3 depending on the authentication solution (loop to A.2)	E.4. Upper-level Action (optional): the outcome of the recognition is subsequently used in some ways (loop to E.1)

Table 2. Authentication and Entity Recognition Side-by-side

There is no initial enrolment step at the beginning of the entity recognition process but this does not mean that enrolment cannot be done. Actually, in step E.3, if the entity to be recognised has never been met before, what will be retained is going to be reused the next time this entity is going to be recognised. Depending on the recognition scheme, it should be more or less transparent, that is, more or less like the enrolment step in A.1. Thus, by moving down the enrolment step in the process, we emphasise that the door is still open for interacting with strangers and unknown entities. An authentication process may be seen as an ER scheme by doing enrolment at step E.3. In the implementation chapter, an example of a message-based ER scheme, called "A Peer Entity Recognition" (APER) [157], is described as well as one based on vision techniques, called Vision Entity Recognition (VER) [164].

A number of different sensing, recognition and retention strategies can be envisaged for entity recognition schemes. In VER, the context at time of retrieval is used to optimise the responsiveness. The detective work depends on which recognition scheme is used. For example, in the APER recognition scheme, it may consist of sending a challenge/response or signature verification. Although Jain et al. [85] use the word recognition, which encompasses verification/authentication and identification, they do not present a full process where enrolment is postponed and not mandatory done as a separate initial step. When they mention that negative recognition cannot be done in non-biometric systems, they fail to present the

[4] http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=retention&x=0&y=0

generic potential of the recognition process. The ER process of this thesis is designed to also be applicable to authentication schemes which are not based on human biometrics. It is the reason that the evaluation is more focused on message-based recognition but also covers a biometrics scheme to demonstrate the generic process.

By self-triggering (step E.1), we mean that the entity takes the initiative to start the recognition process in order to recognise potential surrounding entities, for example it may be starting the recognition scheme that involves the recogniser monitoring the network and selectively carrying out detective work on (some of) the entities that are observed. Step E.4 is optional since it is not required if the only objective is to gather recognition clues. Step E.3 is also optional but the reason is different: recognition clues need not be retained – say if the entity has been seen before.

In our approach, when an entity wants to refer to another virtual identity, it provides recognition clues for other virtual identities to recognise the virtual identity under scrutiny. Each local computing entity interprets these clues in its own way. For example, in the VER scheme, the main clues consist of sequences of images of people passing in front of cameras. In this case, the smart concierge software carries out its own local detective work based on the provided images.

To cope with scalability, we propose to *forget* about entities based on context, for example, that we have not collaborated with, after a certain time. Actually, the tremendous number of entities expected in a pervasive computing environment raises the question of how to scale entity recognition (or authentication) to billions of entities with potentially different distinguishing characteristics. We do not specify how the forgetting mechanism is implemented at this stage and do not mean that all recognition information is deleted: it may be stored in a storage with less efficiency for retrieval. In the evaluation chapter, retrieval is driven by context information and shows optimisation in the retrieval. There are also scalable peer-to-peer schemes that would allow the trustworthy entities to share the load of recognition clues.

5.2 The End-to-End Trust

Since an authentication scheme can follow the entity recognition process explained above, we already support a considerable set of legacy entity recognition schemes: symmetric and asymmetric keys, biometrics… Moreover, the openness required for enrolment suggests many more schemes to come, for example, our APER scheme (detailed in Section 6.2). However, it is known that different authentication schemes are more or less difficult to compromise. As in the Gaia framework (reviewed in Chapter 4), a level of confidence may be associated to the authentication scheme used. Differences in the strength of recognition schemes obviously raise the question of trust in the underlying technical infrastructure. Dynamic enrolment allows previously unknown virtual identities to become acquaintances but what guarantees about the security of this mechanism are offered. Trust in a virtual identity cannot be accurate if the information used at the recognition level is imprecise or simply invalid (for example, due to a successful usurpation attack). Therefore, technical trust in the infrastructure must be explicitly taken into account. Furthermore, it is impossible to expect more than what the technical infrastructure provides: applications requiring strong security should not be run with weak ER schemes.

So, there are layers of trust and the two main categories are trust in the underlying technical infrastructure and trust in the requesting or interacting entity. The point is that these layers form an end-to-end trust, a chain of layers of trust. It has been reported that "information security measures reside in the physical layer of the trust model and have interaction with the personal layer" [120]. Similarly, Golbeck et al. [70] notice that "a security measure builds trust about the authenticity of data contained in the network, but does not describe trust between people in the network". The overall level of trust is the result of how much trust is found at each level. Whether the overall level of trust is acceptable or not is a separate issue. Some benefits of autonomous applications make it worth relying on not-so-trustworthy underlying technologies. There is a trade-off between what can be obtained and what can be lost. This trade-off has to be acknowledged and specified. To get the full potential of autonomous computing, the risks of using not-really-trustworthy environments have to be considered explicitly, as it is indeed done in the risk analysis component of trust engines. Thus, we have the following generic function for the calculation of the overall trust value, the *end-to-end trust value* [157]:

$$EndtoEndTrustValue = f\left(TechnicalTrustValue, VirtualIdentityTrustValue\right)$$

There are different functions that can be used to compute the final end-to-end trust value. For example, in Jøsang's framework (reviewed above), the conjunction operator could be used. Another example may be that the two trust values are on a scale between *0* and *1*, where trust may be interpreted as the probability that an entity behaves in the expected manner for the intended purpose. Assuming these trust values are independent, their multiplication would limit the overall trust value. Beyond a simple level of confidence in recognition, manually set by an expert, the recognition scheme can be associated with a technical trust value, which can be based on direct observations and recommendations. The recognition scheme is seen as an entity, whose trust value varies dynamically interaction after interaction, which is an improvement compared to static confidence values. In this remainder of this thesis, the underlying technical infrastructure is abstracted to the technical trust of the recognition scheme. However, recognition is only one piece of the underlying technical infrastructure. Other technical elements could be considered, for example, secure communication over networks after authentication.

In fact, Jøsang, in his metric for public keys web of trust [91], did not consider the technical trust at the level of the recognition scheme, such as differences between the size of keys. Instead, he focused on the link between the real-world identity and the public key, which corresponds to the virtual identity. In this case, the trust value is set manually by the user. It reminds us that there is also a level of confidence in the association between the real-world identity and the virtual identity. Figure 7 presents a revised version of the identity terms and their relations. The authenticated distinguishing characteristics are replaced by recognition clues. A notion of uncertainty has been added to the arrows by the means of the +/- characters. Damiani et al. [43] also consider that the binding between a virtual identity and a true digital identity goes from unbound to weak to strong but they do not underline that it depends on the authentication scheme used. A variation with [43] is that a partial identity is more or less bound to the user's real-world identity, going from an explicit authenticated binding or implicitly guessed (for example, based on data mining): the binding is not binary.

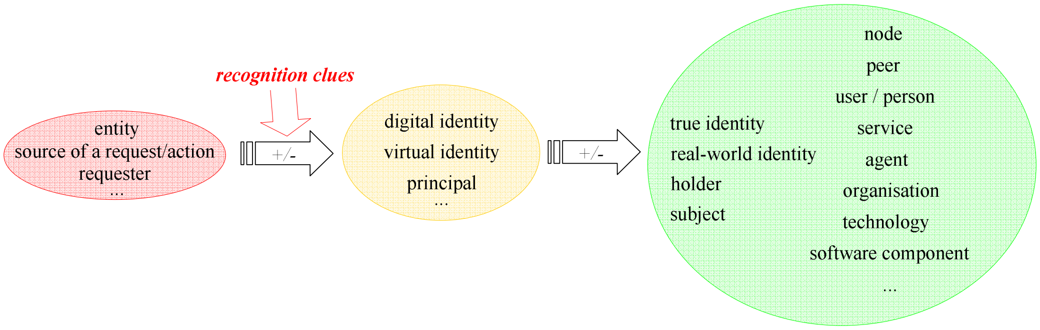


Figure 7. Revised Identity Terms Categorisation

To summarise, the ER process in the light of end-to-end trust consists of four steps:

1. *Triggering* of the recognition mechanism;
2. *Detective Work* to recognise the entity using the available recognition scheme(s); this provides the level of confidence (or technical trust) in recognition;
3. *Discriminative Retention* of information relevant for possible recall or improved future recognition; this is the equivalent of enrolment and constitutes the first main difference with authentication;
4. *Upper-level Action* based on the outcome of recognition, which includes technical trust; this constitutes the second main difference with authentication.

The next section further discusses the use of available recognition schemes mentioned in the step 2.

As said above, in some global computing scenarios, the possibility of prosecution of the real-world identity behind the virtual identity may be low. When the level of system trust is high, the need of a trust value in the entity is less motivated. The link between the real-world identity and the virtual identity seems not mandatory to be able to compute the trust value in the entity. The link has its impact on the trusting decisions since more interactions may be allowed if recourse in the real-world is provided, which is equivalent to a high level of system trust. If the decisions change according to the level of system trust, the outcomes of the different interactions also change and therefore the interpersonal trust value in the entity is changed. Still, a strong level of system trust is not mandatory. It means that recognition schemes, which do not link the real-world identity and the virtual identity, are sufficient for the computation of trust value in the entity. This is good news from a privacy point of view, especially when means to mitigate attacks at the level of identity are provided.

A parallel can be drawn between intrusion tolerance [181] and the need of dynamic enrolment provided by the ER module, which is followed by the formation and evolution of the level of trust in the entity. The door must be open to strangers but if they behave badly, their level of trust decreases and forbid them to generate major security failures. In intrusion tolerance, another mechanism is used to react to the attack but both approaches provide adaptability.

5.3 Means for Recognition Adaptation

According to the identified ASUP dimensions, the framework must be adaptable. Firstly, the recognition module of the entification framework is pluggable with the broad panel of recognition schemes. The outcomes of the different ER schemes can be combined to define the set of recognised virtual identities, including their level of confidence in recognition. Secondly, the environmental context can be used to tune the Pluggable Recognition Module (PRM) [160], which in turn generates information about the recognition state.

5.3.1 Pluggable Recognition Module: Recognised Virtual Identities Set

Global computing encompasses both Internet-based distributed networks and mobile ad-hoc environments. This means that we must be able to reuse existing authentication schemes. It also requires that we can adapt the scheme used to the resources available in the underlying platform. Resource-constrained nodes may need more dynamic enrolment at the expense of stronger security. Choosing a weak recognition scheme, perhaps one allowing for highly dynamic enrolment, is possible but this impacts upon the end-to-end trust. The highest level of trust possible is as high as the level of trust in the underlying technology.

The recognition module should be pluggable as PAM (mentioned in Section 4.4) allows for the use of different legacy authentication schemes. So, we should aim to develop a PRM where auto-configuration is present and a large spectrum of recognition schemes can be used. Adaptability to an entity's capabilities and to legacy authentication solutions is required. The design of that PRM is leveraged from PAM. The main difference is the use of the level of confidence in recognition in the outcome of the recognition process. Beyond the static technical confidence level used in Gaia's pluggable authentication module, the level of confidence may consist of a dynamic trust value in the technical trustworthiness of the ER scheme, based on a flow of evidence.

In JAAS, many authentication schemes can be specified and used and this results in a set of authenticated virtual identities. In the PRM, many recognition schemes can be used. A set of different virtual identities can be recognised with an associated level of confidence in recognition with each of them. Furthermore, due to the use of different ER schemes, with varying strengths, the outcome of recognition carries uncertainty. The uncertainty in the outcome of recognition may be so high that a number of virtual identities may be confused. For example, the VER scheme (detailed in Section 6.6) is proactive: it triggers itself and uses a range of vision techniques which give evidence to compute a probability distribution of recognised entities.

Therefore, the outcome of the ER process can be a set of *n* virtual identities (*vi*) associated with a level of confidence in recognition *lcr*. A range of methods can be used to compute the distribution of the recognised virtual identities, e.g., fuzzy logic or Bayes. In addition, the combination between the level of confidence in recognition in the entity and the technical trust of the ER scheme used can be done in different ways, for example, Jøsang's conjunction operator could be used. The general approach should be to follow the pattern of combination of authentication mechanisms in order to increase security, a.k.a., n-factor (or multimodal) authentication. Jain et al. [85] underline that there are different approaches for the calculation of the output of recognition of biometric techniques, for example: serial mode, the outcome of each ER scheme is used to narrow down the set of identities; or parallel mode, information from multiple traits is used simultaneously. At the difference of the ER process, which provides a set of recognised entities with a level of confidence in recognition, in their identification mode, the identity of the best matched user is returned and the verification mode returns "accept" or "reject". In most combination approaches, it is important to know whether or not the ER schemes use independent recognition mechanisms. For example, one person among *n* previously recognised enters a room which is equipped with a biometric ER scheme (such as VER). The outcome of recognition hesitates between two people: vi_2 and vi_3, which it believes it recognises with respective levels of confidence *8%* and *92%*. Other persons are recognised with a *lcr* of *0%*. The outcome of recognition corresponds to:

$$\sum_{i=1}^{n} lcr_i \times vi_i = 0 \times vi_1 + 0.08 \times vi_2 + 0.92 \times vi_3 + 0 \times vi_4 + \ldots + 0 \times vi_n$$

Technical trust (*tt*) is associated with each ER scheme, such as face template matching in VER. Each technique provides a level of recognition (*lr*) for each entity. If the sum of (*lr*) is too low, this suggests that we need to create a new virtual identity, as long as there are enough recognition clues to distinguish the potential new virtual identity. Assuming that we have *m* ER schemes and that each technique is weighted (with *w*) compared to the other ER schemes used, we have:

$$lcr = \sum_{j=1}^{m} lr_j \times tt_j \times w_j$$

We have also developed recognition in message-based applications provided by the *Claim Tool Kit (CTK)* [160], which is one of the main implementation contribution (detailed in the next chapter): the APER scheme uses cryptographic keys, hashes of previous messages and challenge/responses. Since public keys are used for recognition, at first glance, it is less uncertain than with vision techniques. However, depending on the key size, the symmetric cryptographic algorithm used and the time since creation, technical trust may also vary. The more time since the key has been generated, the more time attackers have to break the key.

5.3.2 A Tuneable/Talkative ER Module

In the security intrusion-tolerant architectures [181] (as said in Section 4.2.3), it is assumed that security faults remain, that is, that security is not perfect. However, faults are mitigated by the presence of error detection mechanisms, which are triggered when a fault is detected, and error handling mechanisms, whose goal is to avoid failures (e.g., network connection is closed if a remote fault source is detected). False alarms are a burden for the administrator, especially if the administrator has no time or skills to be an administrator (e.g., the busy tenant of a smart home, who cannot pay a real administrator). Adaptability makes the life of the users easier.

Assuming that the ER module can be used in a plethora of contexts varying from one extreme context to another (for example, due to different types of hardware), in addition to the absence of skilled administrators and requiring minimal user intervention (if a user is present at all), an “autonomic” [32] model of the ER module is needed. The basic pattern of an “autonomic element” [32] consists of a management unit and a functional unit. When we apply this pattern to our ER process, its four steps (namely Triggering, Detective Work, Discriminative Retention and Upper-level Action) become parts of the functional unit as depicted in Figure 8. The management unit is in charge of monitoring and tuning the ER module.

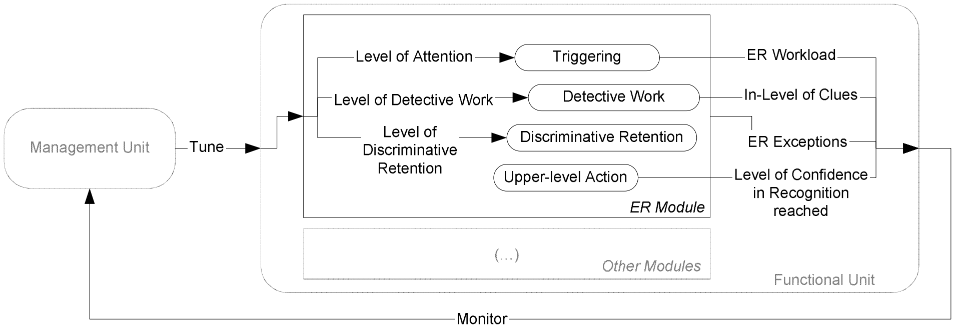


Figure 8. The ER Autonomic Functional Unit[5]

The type of management unit can vary a great deal: "unlike conventional computing systems, which behave as they do simply because they are explicitly programmed that way, the management unit of an autonomic element will often have a wide range of possible strategies" [32]. The management unit is open to a broad panel of policies and decision-making mechanisms. Of course, in the entification framework, the decision-making is based on computational trust.

The access control of the autonomic element pattern can be enforced by tuning the level of attention: the lower the level of attention, the fewer recognition rounds are processed. When the level of attention changes, the triggering of the ER module becomes more or less sensitive. In return, more or less computation is spent for entity recognition. This is useful when management of computation resources is required. A greater or lower level of detective work means that the ER module spends more or less time and applies more or fewer mechanisms to recognise current entities. A greater or lower level of discriminative retention means that the ER module retains more or less recognition clues for later recognition.

Exceptions occurring during the entity recognition process should be used to update the environmental context in order to react to potential attacks at the recognition level (for example, denial-of-service due to too many triggerings, or trial-and-error attack over a set of possible observable attributes). Each time there is a triggering and not enough recognition clues provided, the ER module can log an optional piece of evidence summarising this

[5] Both management and functional units are represented as rounded rectangles. The black directed arrows represent information from one element to another. The functional unit rounded rectangle contains two module rectangles: the ER module and another one for potential other modules. The ER steps are drawn as rounded rectangles in the ER module rectangle.

exception. Other kinds of ER exceptions may be envisaged. This logging is represented by the loop called Monitor back to the management unit at the bottom of Figure 8. Other optional useful pieces of evidence to be given back to the management unit consist of the level of confidence in recognition reached after each recognition, the in-level of clues (that is, an estimation of the level of clues used by other virtual identities) present and used for reaching these levels of confidence in recognition and the ER current workload. All these pieces of evidence can be used by the management unit, in addition to external information, to choose the most appropriate level inputs for the ER Module. For example, by knowing that the system is under DoS attack, the triggering of the ER process can be made less sensitive in order to decrease resources used for recognition. If more security is required, the level of detective work may be increased in spite of spending more resources. The final element that the management unit can tune is the level of clues to be exhibited to other entities, which may include the selection of the local pseudonym to use as encouraged in the next section.

5.4 Encouraging Privacy and Still Supporting Trust

In our view, privacy is a human right or a need that must remain in the hands of each individual. Therefore, we encourage the use of multiple pseudonyms to protect privacy. Still, the entification framework allows the users to trade privacy for increased trust if they wish.

5.4.1 Encouraging the Use of Multiple Pseudonyms for Privacy’s Sake

The ER process is presented above as a more general replacement for authentication that does not necessarily bind an identity to the recognised virtual identity. We consider authentication as a special case of recognition that binds a real-world identity to the recognised virtual identity. We argue that the ability to recognise another entity, possibly using any of its observable attributes, is sufficient to establish trust in that entity based on past experience. Our end-to-end trust model starts with recognition where the link with the real-world identity in absolute terms is not needed. Therefore recognition intrinsically favours privacy by divorcing the recognition and representational aspects of identity. Our expectation is that entities are in general virtually anonymous to the extent that the link to the real-world identity alone conveys little information about likely behaviour. What is important as a prerequisite is not really "Who exactly does this entity represent?" but "Do I recognise this entity as a trustworthy collaborator, whomever it represents?" The real-world identity may bring system trust but it is not mandatory for the computation of interpersonal trust values. We assume virtual anonymity and therefore we do not require (but do allow) the ability to establish the real-world identity of a given entity in absolute terms, for example, through globally unique and meaningful certified X.509 "distinguished names" assigned to real-world identities. As already said, in global computing settings, it may be not feasible to enforce penalty mechanisms on the real-world entities due to no unique legitimate authority.

Information becomes personal when it can be linked back to an individual or when it, in some way, allows two individuals to be linked together. This means that control of the dissemination of personal information can be exercised through preventing, or at least limiting, linkability of information to individuals. This is illustrated in Figure 9, where a user Alice performs some transactions with another user Bob (neither Alice nor Bob needs to be actual users, but could be clients, servers or part of the computing infrastructure).

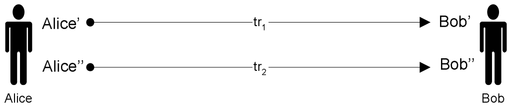


Figure 9: Linkability of Transactions

In Figure 9, Alice performs two transactions tr_1 and tr_2 with Bob. In order to protect the privacy of Alice[6], it is important that Bob, or anyone who eavesdrops on their communication, is unable to link either transaction tr_1 or tr_2 directly to Alice's real-world identity. However, it is equally important to prevent Bob from linking the two transactions to each other, since this would allow him to compile a comprehensive profile of the other party, which could eventually identify Alice. Moreover, the violation of Alice's privacy would be increased dramatically if any future transaction tr_x can be linked to Alice, since this would allow Bob to link the full profile to Alice and not just tr_x. However, trust is based on knowledge about the other party [90], which directly contradicts the prevention of linkability of information to users, so perfect privacy protection, i.e., preventing actions to be linked to users, prevents the formation, evolution and exploitation of trust in the online world.

[6] The rights/needs to privacy of Alice and Bob are symmetrical, so it may be equally important to prevent Alice from knowing that the two transactions were performed with the same entity.

Recalling the process of trust formation makes apparent the fact that privacy is at stake in trust-based systems. Computational trust is built by linking interactions over time and recommendations between entities. In order to be able to make the decision to trust another entity, the first step is to establish the level of trust in that entity, which is the result of an analysis of the existing knowledge and evidence. If full knowledge is available, it is true that the need of trust vanishes because there is then no uncertainty and no risk. To establish the level of trust in the entity requires ways to relate the entity with its trust value. The common way is to use real-world identities to be able to achieve this relation. First of all, privacy is really in danger when identities point to real world users. In this case, they become Personally Identifiable Information (PII). For example, two entities interact and information about the outcome of their interactions is recorded. Depending on the outcome, the trust between each entity is increased or decreased. Further, this trust information may be forwarded to other entities as recommendations. The drawback of this approach is that an entity may disclose arguably private information and compromise the privacy of the targeted entity [132]. Even if trust has only been built with direct observations, PII information stored in another entity may still have to conform to directives, for example, the Fair Information Practices (FIP) [39]. Secondly, when trust built due to direct observations is used for further recommendations or reputation, the new trust values created in other entities are by no means part of a common interaction, removing any privacy legitimacy of their source. Trust relies on profiling, where more information is better, because it allows the likely behaviour of the other entity to be more accurately estimated. The trust engines are fuelled with information which aims at building more and more accurate profiles over time. Any link with the real end-user would change this information into sensitive PII. There is an inherent conflict between trust and privacy because both depend on knowledge about an entity, but in opposite ways. There must be a mechanism that can dissociate users from their actions [108]. However, when privacy protection is high, the need of trust is far greater that when full knowledge is available.

From a privacy protection point of view, we argue for the use of multiple virtual identities, acting as pseudonyms as a first technological line of defence. In Kobsa and Schreck's classification, transaction pseudonyms (such as a pseudonym used for only one transaction), and anonymity cannot be effectively used because they do not allow linkability between transactions as required when building trust. Pseudonyms appear to be the appropriate solution for protecting privacy in trust-based systems and achieving some level of privacy and trust.

The minimum requirement is a local reference for the formation of trust, which is in turn managed by other components in the trust engine. According to the privacy protection principle of "collection limitation" [113], data collection should be strictly restricted to mandatory required data for the purpose of the collection. Since trustworthiness estimation accuracy increases as information increases, it is not inbuilt for trust engines to minimise collection of personal information. Our requirement is to establish the trustworthiness of entities and not their real-world identity. This is why pseudonymity, the level of indirection between trust and the real-world entity, is sufficient. Giving users the option to conceal their identities seems a viable way to alleviate users' privacy concerns, whilst preserving the benefits of trusted interactions.

It is known that "pseudonymization is effective only if identity cannot be easily inferred from user behaviour" [87]. Ian Goldberg [71] underlined that any transaction engaged by a person reveals meta-content, especially information about the identity of the person. Traffic analysis, data triangulation and data-mining, with some effort, may also associate a pseudonym with the real user. That is why it is important that we provide multiple pseudonyms that are levels of indirection between trust and real-world identity. Others have proposed to change pseudonyms, for example, based on temporary pseudonyms [15, 108, 134] or random selection of one in a set of pseudonyms [134]. We implemented a prototype (detailed in Section 6.5) where pseudonyms are selected according to context, especially location.

5.4.2 Trading Privacy for Trust

Although trust allows us to accept risk and engage in actions with potentially harmful outcome, a computational trust engine must take into account that humans need (or have the right to) privacy. However, depending on what benefits can be reaped through trustworthiness, people may be willing to trade part of their privacy for increased trustworthiness: hence, contextual privacy/trust trade is needed. Due to the division of trust evidence between many pseudonyms, it takes more time for the entities behind these pseudonyms to reach the same trustworthiness than for a unique virtual identity. In Section 4.3, we have seen that the privacy expectations of a user vary across time and depend on contexts. Users can get benefits from the knowledge of profiles, preferences and identity information thanks to customised services [105]. Depending on what they can get, they may be willing to divulge some of their private data. One may argue that this is not right [11]. However, business must also be considered [116] along with technology, legislation and social norms. "Social norms are cultural phenomena that prescribe and proscribe behaviour in specific environments, the emergence of which are key to trust formation and privacy concerns" [87]. There is definitely an intrinsic relationship between trust, privacy, legislation, technology, social norms and markets.

Therefore, we introduce a model for privacy/trust trade based on linkability of pieces of evidence. We start by an informal summary of the model. When true knowledge[7] about an entity increases:

- The evaluation of its trustworthiness is more accurate and if this entity is indeed truly trustworthy, its trustworthiness increases;
- Its privacy decreases and it is almost a one-way function[8] because privacy recovery is hard to achieve [71, 161].

Knowledge is composed of evidence [90]. A piece of evidence *ev* may be any statement about some entity(ies), especially: a transaction *tr*, a direct observation[9,1] *obs* (i.e., evaluated outcome of a transaction [88]), or a recommendation *rec*. The *nymity* of evidence is the amount of information about the identity of the entity that is revealed. The *trustworthiness assessment impact*, called *tai* of evidence, is the amount of information that can be used for assessing the trustworthiness of the entity, which is represented as a trust value.

[7] By true knowledge, we mean knowledge which cannot be refuted (i.e., it cannot be a lie, noise information or revised).

[8] On Goldberg's Nymity Slider, it is "easy to change the transaction to have a higher position on the slider" and "extremely difficult to move a transaction down the slider (towards unlinkable anonymity)".

[9] It is sometime difficult to find out when the observation should be made because it is not clear whether the action is finished or not. It may be solved by having a kind of dynamic observation, i.e., a piece of evidence which varies through time as well.

There are different levels of nymity. So we assume that there is a partial order between nymity levels, called *Privacy Asset Order (PAO).* Goldberg's Nymity Slider [71] is one example of such ordering. We present another example of PAO below:

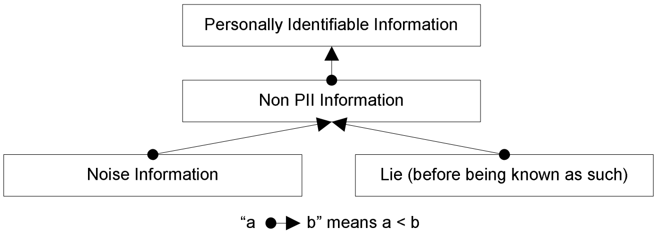

Personally Identifiable Information
Non PII Information
Noise Information
Lie (before being known as such)
"a ●▶ b" means a < b

Figure 10: Privacy Asset Order Example[10]

Similarly, evidence may be more or less useful for trustworthiness assessment. So we assume that there is a partial order between tai levels, called *Trustworthiness Assessment Impact Order (TAIO).* An example of TAIO is:

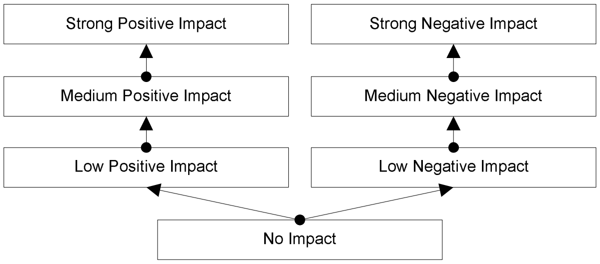


Figure 11: Trustworthiness Assessment Impact Order Example[10]

A piece of evidence of PII nymity is more likely to have a strong positive impact tai, especially when it is assumed that the real-world identity can be sued. However, one non-PII evidence may have low positive impact and another one strong positive impact.

We provide a mechanism that can link n pieces of evidence ev_i for $i=1,\dots,n$ and represented by:

$$link(ev_1, ev_2, \dots, ev_n)$$

[10] In all figures of this section representing partial orders, rectangles represent a level in the partial order and a directed black arrows between two levels indicates which level is greater than another one in the partial order.

The result of the link mechanism is a new piece of evidence with a new tai level as well as a new nymity level. Sometimes, linking of evidence is implicit (i.e., the requesting entity cannot keep secret that two pieces of evidence are linked) and it is redundant to make it explicit (i.e., the requesting entity discloses to other entities that two pieces of evidence are indeed linked). For example, if two events ev_2 and ev_3 are implicitly linked, then explicitly linking ev_1 and ev_2 is equivalent to explicitly linking ev_1, ev_2 and ev_3: $link(ev_1,ev_2) = link(ev_1,ev_2,ev_3)$.

For example, after the first transaction, the requested entity links the transaction with the pseudonym virtual identity *vi*: $link(tr_1,vi)$. Then, after the second transaction, the requested entity does: $link(tr_1, vi, tr_2)$ and so on. Thus, the pseudonym links a set of pieces of evidence together. If each transaction is non-PII/low positive impact and the recognition clues used to recognise *vi* are considered as non-PII/no impact, the resulting evidence is: two low positive impacts from a tai point of view and three non-PII from a nymity point of view.

If not enough evidence is available under the chosen pseudonym, evidence not linked to this pseudonym may improve trustworthiness and allow the requesting entity to be granted the request. The entity may be willing to disclose further evidence to the requested entity in spite of potential increased privacy loss. So, a protocol for disclosing to the requested entity that some evidence can be linked is needed. We present such a protocol, called the *privacy/trust trade process* [162] (depicted in Figure 12).

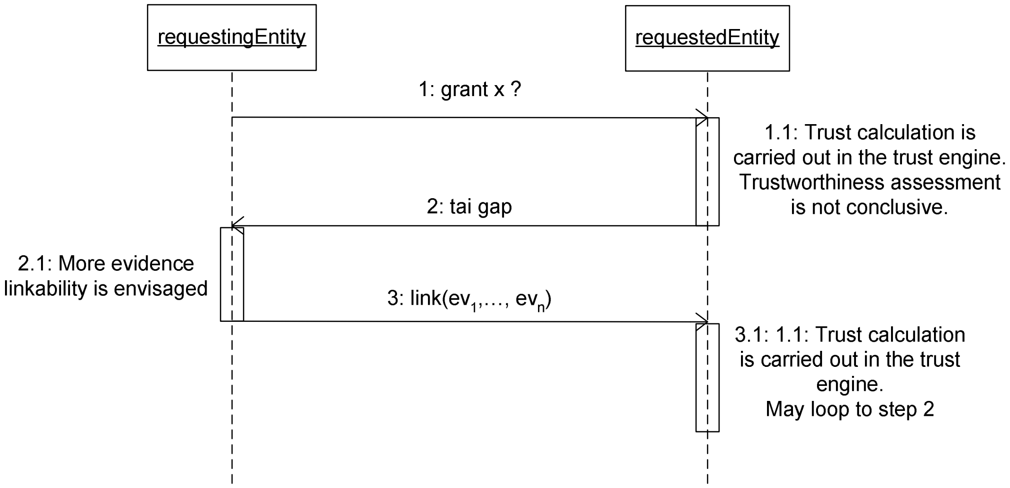

requestingEntity
requestedEntity
1: grant x ?
1.1: Trust calculation is carried out in the trust engine. Trustworthiness assessment is not conclusive.
2: tai gap
2.1: More evidence linkability is envisaged
3: link(ev_1,..., ev_n)
3.1: 1.1: Trust calculation is carried out in the trust engine. May loop to step 2

Figure 12: Privacy/trust Trade Sequence Diagram[11]

In this process, the requested entity makes the decision that not enough evidence is available for granting and this fact should be disclosed to the requesting entity. So, after step 2, the requesting entity knows the tai of evidence that should be obtained.

In step 2.1, different potential evidence can be envisaged to be linked by the requesting entity. The choice of evidence should be based on the following principle:

The Minimal Linkability *principle: No more evidence than needed should be linked.*

The latter principle is a variant of the "Need-To-Know" principle. One of the reasons is that more trust implies more knowledge given out, thus less chance for privacy. Some thresholds should be set concerning the acceptable evidence that should be disclosed in step 3. Without such thresholds, an attacker may ask to retrieve all evidence (i.e., knowledge), which is what we want to prevent by using pseudonyms. If the user must confirm that some evidence can be linked, more care has to be taken into account. It is known that users can easily agree to sell privacy in stressed circumstances without thinking of the consequences [170], which are often irrevocable since privacy recovery is hard [161]. For example, Alice, in order to get quick access to a large video display, may regret to present her full profile to the video club due to this small benefit compared to life-long spam messages sent by a malicious video club.

[11] In any sequence diagram of this thesis, the vertical dashed lines represent the lifelines of sequence diagrams; the rectangle above each lifeline indicates the name of the entity targeted by the lifeline; the directed arrows represent which message is sent from one entity to another; and the numbers indicate the chronological order of the messages.

One way to prevent such abuse may be the existence of a broker where reasonable trades are listed (this also reduces interoperability issues). In practice, it may require an exchange of messages with trusted-third-parties to decide whether the trade is fair (within the current market price) or not. We propose to introduce another partial order to cope with such abusive trade attack. The utility of a transaction is represented on a *utility partial order (UO)*. An example[10] UO may be:

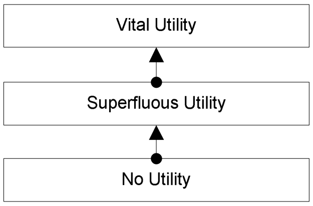


Figure 13: Utility Order Example[10]

During a trade process, tai, nymity and utility must be balanced. Alice under the pseudonym *vi* requests Bob to grant the transaction tr_x of utility *u* from Alice's point of view. In step 1.1, if *vi* had done two previous transactions tr_1 and tr_2 with Bob, Bob's trust engine checks if the trustworthiness given by this previous evidence is enough to grant tr_x. In this case, the trustworthiness assessment is not conclusive, so the trust engine computes the *z* tai of evidence missing, called *tai gap*. Alice's trust engine is noticed that *z* tai of evidence is missing. In step 2.1, Alice's trust engine does the following 2-step algorithm, called *link selection engagement (liseng) algorithm*:

1. *Search link of evidence expected to fill the tai gap but minimise nymity*: As an example, we assume that the trust engine cannot guarantee that all recommenders of *vi* can exhaustively be found and queried in a timely manner. All transactions directly done between Alice and Bob should have been taken into account by Bob's trust engine. However, Alice has done 2 transactions with Charles, tr_{1r} and tr_{2r}. We assume that these two transactions may not have been recommended by Charles to Bob in the first round. We end up with one set: *link*(tr_{1r}, tr_{2r}, *vi*). Alice has done transactions with other people than Charles and Bob but tr_{1r} and tr_{2r} fills the tai gap and adding more transactions would increase nymity.
2. Check that nymity of the selected link of evide*nce is reasonable compared to the utility*: if yes, engage in further trade steps; else abort the trade. We assume that each utility level is associated with a maximum nymity threshold. This check corresponds to a cost/benefit analysis. So, the risk module of the trust engine should be responsible for carrying out this analysis.

By allowing any entity to make recommendations, we directly support a change of identity, where evidence can be transferred and linked to the new identity through a recommendation, without explicitly linking the two identities. It may indeed consists of *self-recommendations*, that is, recommendations from virtual identities belonging to the same real-world identity. This limits the extent of the profile that can be built for a given virtual identity, thereby reducing the violation of privacy resulting from a single transaction being linked to the real-world identity of a user. So, in step 3 of the privacy/trust trade process, a list of pseudonyms owned by the requesting entity could be sent back as potential new recommenders. If the requested entity has not already used these pseudonyms as recommenders, it would do so. However, the tai of evidence provided by these entities would be discounted by the recommendation process. This is why it may be more beneficial to make the link between some pseudonyms explicit: an operation that we call *fusionym* [159].

Indeed, the link mechanism can be used to link different virtual identities if there is evidence that the virtual identities correspond to the same real-world identity. The linkage can be more or less strong. Anyway, the result of linking evidence is another piece of evidence with its own nymity and tai. In our example implementation in the next chapter, evidence is linked through digital signature validation. For example, we may have $link(vi', vi)$. It is worth noticing that we also implicitly link all m transactions tr'_j linked to vi' and n transactions tr_i linked to vi: $link(tr'_1, \ldots, tr'_j, \ldots, tr'_m, vi', tr_1, \ldots, tr_i, \ldots, tr_n, vi) = link(vi', vi)$.

Fusionym is when two or more pseudonyms are linked together, so that they can be regarded as one pseudonym whose overall trust value is calculated based on the pieces of evidence of each composing pseudonym. Even though perfect unlinkability is hard to achieve[8], we need to include the possibility of partitioning a set of pieces of evidence and associating each subset with a new pseudonym. *Partitionym* is when from the set of pieces of evidence of one pseudonym, new pseudonyms materialise. It includes the case when only one new pseudonym is created. Depending on the feasibility of unlinkability, partitionyms are still possible but provide limited privacy protection. If we assume that it is feasible to unlink evidence, after partitioning, the new pseudonyms provide as much privacy protection as pseudonyms created from scratch. However, most of the time, partitionyms cannot guarantee that the link will not be discovered because of the difficulty of perfect unlinkability. For example, four transactions tr_1,tr_2,tr_3 *and* tr_4 between Alice and Bob are linked to a piece of evidence ev. So, Bob knows $link(tr_1,tr_2,tr_3,tr_4,ev)$. If Alice interacts with Charles and she is able to unlink the transactions from ev, she can present $link(tr_1,tr_2,tr_3,tr_4,ev')$, a link with another piece of evidence ev', to Charles. However, the link is still left present and as soon as Charles communicates with Bob, the link to ev can be re-established. There is a window of time when unlinkability is concealed. It is this window of time between Alice's transaction with Charles and Bob's transaction with Charles that allows the evidence to be linked. In this case, linkability remains implicit even though partitionym is attempted.

This new prospect for linking evidence allows us to envisage new linked evidence in step 2.1 of Figure 12. So, in step 3, a list of pseudonyms owned by the requesting entity could be sent back as potential new evidence in the form: *link*(vi_1,..., vi_i,...,vi_n) given that linkage evidence is provided. In step 1 of the liseng algorithm (using the example we presented above when describing this algorithm), another choice may be to use two transactions, tr_3 and tr_4, that Alice under the pseudonym *vi'* did with Bob: the resulting link can be specified with more or less explicit linked evidence depending on what can be implicitly linked. For example, if the trust engine does not guarantee that all transactions done under a specific pseudonym can be available in a timely manner (especially for recommendations), the explicit link should be longer: *link*(tr_3, tr_4, *vi', vi*). If any transaction is guaranteed to be known by all entities[12], it would be sufficient with a link of this type: *link(vi', vi)*. The link between two virtual identities is permanent and cannot be easily undone (for example, when we link two keys, we use the fact that an entity cryptographically shows the ownership of both private keys of the two pseudonyms). It is important to note that transactions are often temporary, while linking transaction and/or virtual identities is permanent. This must be taken into account when estimating the utility of a given transaction.

5.5 Accuracy and Attack-Resistance of the Trust Values

Usurpation attacks can be mitigated with the level of confidence in recognition or the technical trust in the recognition scheme used. Chapter 7 details how we do that for dynamic email anti-spoofing techniques. Still, there are remaining issues. First, the issues concerning overcounting trust evidence are detailed. Then, a solution to get safe fusionym is given. When using pseudonyms, a means must be present to prevent users from taking advantage of the fact that they can create as many virtual identities as they wish [79]. This section explains how identity multiplicity issues are mitigated.

[12] It is a strong assumption to guarantee global propagation of information. This assumption is not realistic in most scenarios (e.g., when random disconnection is possible).

5.5.1 Issues Surrounding Accurate Trustworthiness Assessment

A difficult aspect of the liseng algorithm is to take into account the sequencing of interactions. Pieces of evidence revealed before the current interaction can impact the selection as well as future pieces of evidence due to the combination of pieces of evidence. For example, for two candidates ev_1 and ev_2 with same tai but different nymity ($nymity_1 < nymity_2$), in the scope of this specific interaction, ev_1 should be chosen. However, if a future interaction links ev_3 with $nymity_{link(ev1,ev3)} > nymity_{link(ev2,ev3)}$, the choice becomes more difficult.

We emphasise that care should be taken when linked evidence on multiple virtual identities is assessed. The most important requirement is to avoid counting the same evidence twice when it is presented as part of two different pseudonyms or overcounting overlapping evidence. In some cases, passing recommendations in the form of a simple trust value, instead of all supporting information[13], does not fulfil the later requirement. Assessing evidence may require analysis and comparison of each piece of evidence to other pieces of evidence. For example, let us assume that we have the relation depicted in Figure 14 and we know the trust values of two virtual identities vi_1 and vi_2, tvi_1 and tvi_2 respectively. The X axis of Figure 14 represents the number of good observations and Y axis the resulting trust value.

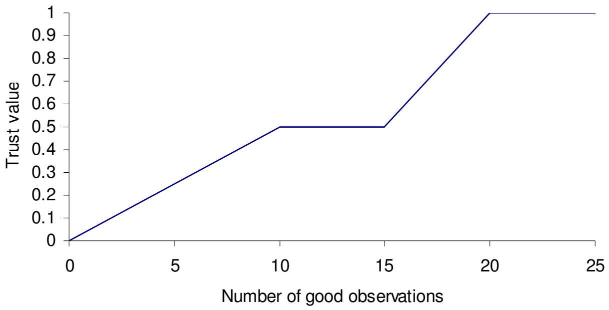


Figure 14: Example Relation between Observations and Trust Values

If $tvi_1 = 0.5$, whatever value tvi_2 is, we cannot compute the combined trust value without knowing the number of good observations, which is at a level of evidence deeper[14] than the

[13] We agree that only passing the trust value may improve performance and may be better from a privacy point of view than all evidence information. However, it may also decrease interoperability as highlighted here, and may show how another entity computes trust from evidence. This may help to mount attacks and may reveal feelings towards other entities, which may not be welcome.

[14] With this case of relation, it is also insufficient to only transfer the trust value in recommendations.

level of trust values. In fact, assessing linked evidence requires great care and implementations may vary depending on the complexity of trust-lifecycle [183] and trust dynamics [88]. When recommendations are used, previous self-recommendations are also not easy to take into account. If this is part of a low cost mechanism for introducing new pseudonyms, it may be tolerated to simply discard the recommendations in the calculation. However, this must be part of a risk analysis decision. If this is the case, then it is difficult to determine how to fairly incorporate this into a trust engine based on count of event outcomes, as the self-recommendation is not based on a real history of interactions. In addition, recommendations must be stored separately from direct observations and identified with the recommender name. Another choice might be to consider such recommendations as evidence of untrustworthiness. Let vi_1 and vi_2 be two pseudonyms of the same entity. At the first interaction with the requested entity, vi_2 is used as a recommender for vi_1 due to the recommendation $rec_{2,1}$[15]. So, the entity has now $link(vi_1,tr_1,rec_{2,1})$ for trustworthiness assessment of vi_1. At the second interaction, vi_1 discloses $link(vi_1,vi_2)$. Logically, the tai of $rec_{2,1}$ needs to be revised, for example, by discarding rec_{21} in the tai of the resulting evidence. However, if no linkage is done, then we cannot know if there has been a self-recommendation. Permission to make self-recommendations at will, without cost paves the way for a Sybil attack, unless addressed. We introduce *trust transfer* (explained in Section 5.5.3) to address this issue.

5.5.2 Safe Fusionym

In our solution for safe fusionym, we assume that accurate trustworthiness assessment of the link between some pseudonyms is possible if:

- the trust value is based on direct observations or recommendations of the count of event outcomes from one specific entity (reputation from a number of unidentified entities and credentials as in trust management [20, 189] are not covered);
- the link function is only used for linkage at the level of virtual identities: $link(vi_1,\ldots,vi_n)$ for n linked pseudonyms; the linkage is supposed perfect, unconditionally true and proven by some means such as cryptographic signatures;

[15] In $rec_{2,1}$, the $_{2,1}$ means that vi_2 makes a recommendation about vi_1.

- a pseudonym can neither be compromised nor spoofed; an attacker can neither take control of a pseudonym nor send spoofed recommendations; however, everyone is free to introduce as many pseudonyms as they wish;
- all messages are assumed to be signed and time stamped.

Different formats can be used to represent the trust value. A simple scenario is when there is one type of outcome and the trust value is represented as a tuple *(g,b)* counting the number of good (*g*) outcomes and bad (*b*) outcomes. Another format compliant to our assumption is the SECURE trust value format (as detailed in Section 3.1).

With such a trust value format and a perfect linkage, it is sufficient to add each element. For example, the fusionym trust value of *link*($vi_1, \ldots, vi_n$) is:

$$\sum_{i=1}^{n}(g_i, b_i) \quad \textit{or} \quad \sum_{i=1}^{n}(s_i, i_i, c_i) \quad \textit{in the whole event structure.}$$

It is crucial to note that the above fusionym is only safe based on direct observations. Due to the possibility of self-recommendations, it is not safe to simply add the recommended trust values. The next section gives an extended solution where recommendations can be used without the threat of a Sybil attack.

5.5.3 Trust Transfer to Mitigate Identity Multiplicity

In a system where there are pseudonyms that can potentially belong to the same real-world entity, a transitive trust process is open to abuse. Even if there is a high discounting factor due to recommending trustworthiness, the real-world entity can diminish the impact of this discounting factor by sending a huge number of recommendations from his/her army of pseudonyms in a Sybil attack. Additionally, obtaining a measure of recommending trustworthiness is rather difficult, for example, the "semantic distance" [6] between recommendations and local outcomes has to be calculated and that involves the (rather arbitrary) choice of a number of parameters.

It has been noted in the literature that there are issues "with trusting recommenders to recommend arbitrarily deep chains" [6]. They argue that trust at level *n* is independent of trust at level *n+1*. However, this contradicts Romano's view (and the view adopted in this thesis) of trust as a single construct that varies across contexts: there is a dependency between trust contexts as they are not independent multiple constructs. Romano also notes that trust is "functional, such that trusting behaviours are attempts to attain desirable outcomes by protecting one's interests through actions that either increase or decrease influence in accordance with one's assessment of such influence" [145]. When someone recommends another person, he/she has influence over the potential outcome of interaction between this person and the trustor. The inclination of the trustor with regard to this influence "provides a goal-oriented sense of control to attain desirable outcomes" [145]. So, the trustor should also be able to increase/decrease the influence of the recommenders according to his/her goals. Moreover, according to Romano, trust is not multiple constructs that vary in meaning across contexts but a single construct that varies in level across contexts. We conclude that the overall trustworthiness depends on the complete set of different domains of trustworthiness. This overall trustworthiness must be put in context: it is not sufficient to strictly limit the domain of trustworthiness to the current trust context and the trustee; if recommenders are involved, the decision and the outcome should impact their overall trustworthiness according to the influence they had. Kinateder et al. [102] also take the position that there is a dependence between different trust contexts. For example, a chef known to have both won cooking awards and murdered people may not be a trustworthy chef after all. In this thesis, we introduce the possibility of a dependence between trustworthiness and recommending trustworthiness in the same trust context.

In addition, as La Rochefoucauld [114] wrote[16] a long time ago, recommending is also a trusting behaviour. It has not only an impact on the recommender's overall trustworthiness (meaning it goes beyond recommending trustworthiness) but also on the overall level of trust in the network of the involved parties. This social network formed by trusting behaviours is intricate and a model assuming independence of any of its parts appears to be unlikely to result favourably. La Rochefoucauld highlighted that when one recommends another, they should be aware that the outcome of their recommendation will reflect upon their trustworthiness and reputation since they are partly responsible for this outcome. Benjamin Franklin noted about recommendations that each time he made a recommendation, his

[16] Original quotation in French: La confiance ne nous laisse pas tant de liberté, ses règles sont plus étroites, elle demande plus de prudence et de retenue, et nous ne sommes pas toujours libres d'en disposer: il ne s'agit pas de nous uniquement, et nos intérêts sont mêlés d'ordinaire avec les intérêts des autres. Elle a besoin d'une grande justesse pour ne livrer pas nos amis en nous livrant nous-mêmes, et pour ne faire pas des présents de leur bien dans la vue d'augmenter le prix de ce que nous donnons.

recommending trustworthiness was impacted: “in consequence of my crediting such recommendations, my own are out of credit” [61]. However, his letter underlines that still he had to make recommendations about not very well-known parties because they made the request and not making recommendations could have upset them. This is in line with Covey’s “Emotional Bank Account” [35, 152], where any interaction modifies the amount of trust between the interacting parties and can be seen as favour or disfavour – deposit or withdrawal. In Covey’s model, there is one bank account, which is similar to Romano’s single construct, and any type of interaction has an impact on this single construct.

As said above, the trustor should be able to increase/decrease the influence of the recommenders according to his/her goals. The goal in our case is to build a trust engine, which allows recommendations without being vulnerable to the Sybil attack, so the mechanism used to control the recommender’s influence must achieve this goal. We call this mechanism, *trust transfer* [159], which relies on the same assumptions as in Section 5.5.2.

Trust transfer implies that recommendations cause trust on the trustor (*T*) side to be transferred from the recommender (*R*) to the subject (*S*) of the recommendation. A second effect is that the trust on the recommender side for the subject is reduced by the amount of transferred trustworthiness. If it is a self-recommendation, then the second effect is moot, as it does not make sense for a real-world entity to reduce trust in his/her own pseudonyms. Even if there are different trust contexts (such as trustworthiness in delivering on time or recommending trustworthiness), each trust context has its impact on the single construct trust value: they cannot be taken separately for the calculation of the single construct trust value. A transfer of trust is carried out if the exchange of communications depicted in Figure 15 is successful. A local entity’s *Recommender Search Policy (RSP)* dictates which contacts can be used as potential recommenders. Its *Recommendation Policy (RP)* decides which of its contacts it is willing to recommend to other entities, and how much trust it is willing to transfer to an entity.

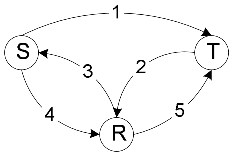


Figure 15. Trust Transfer Process[17]

Trust Transfer (in its simplest form) can be decomposed into 5 steps:

1. The subject requests an action, requiring a total amount of trustworthiness *TA* in the subject, in order for the request to be accepted by the trustor; the actual value of *TA* is contingent upon the risk acceptable to the user, as well as dispositional trust and the context of the request; so the risk module of the trust engine plays a role in the calculation of *TA*;
2. The trustor queries its contacts, which pass the *RSP*, in order to find recommenders willing to transfer some of their positive event outcomes count to the subject. Recall that trustworthiness is based on event outcomes count in trust transfer;
3. If the contact has directly interacted with the subject and the contact's *RP* allows it to permit the trustor to transfer an amount *(A≤TA)* of the recommender's trustworthiness to the subject, the contact agrees to recommend the subject. It queries the subject whether it agrees to lose *A* of trustworthiness on the recommender side;
4. The subject returns a signed statement, indicating whether it agrees or not;
5. The recommender sends back a signed recommendation to the trustor, indicating the trust value it is prepared to transfer to the subject. This message includes the signed agreement of the subject.

Both the *RSP* and *RP* can be as simple or complex as the application environment demands. For now, we limit the policies to simple ones based on trust values. For example, a more complicated *RSP* could be based upon privacy considerations (as is highlighted in the email anti-spam application in Section 7.6.2). An advanced *RP* could be based upon level of participation in the collaborative process and risk analysis.

[17] In this type of figure, the circles represent the different involved entities: *S* corresponds to the sender, which is the subject of the recommendation and the requester; *T* is the trustor, which is also the target; and *R* is the recommender. The directed black arrows indicate a message sent from one entity to another. The arrows are chronologically ordered by their number.

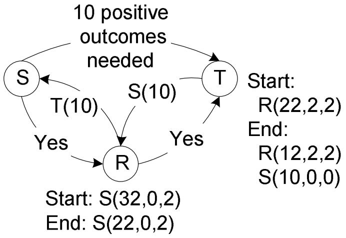


Figure 16. Trust Transfer Process Example[17,18]

The trust transfer process is illustrated in Figure 16 where the subject requests an action, which requires *10* positive outcomes (recall that the system uses interpersonal trust based on the outcome of past events). The *RSP* of the trustor is to query a contact to propose to transfer trust if the *balance (s-i-c)* is strictly greater than *2TA*. This is because it is sensible to require that the recommender remains more trustworthy than the subject after the recommendation. The contact, having a balance passing the *RSP* (*s-i-c=32-0-2=30),* is asked by the trustor whether he/she wants to recommend *10* good outcomes. The contact's *RP* is to agree to the transfer if the subject has a trust value greater than *TA*. The balance of the subject on the recommender's side is greater than *10 (s-i-c=22-2-2=18)*. The subject is asked by the recommender whether he/she agrees *10* good outcomes to be transferred. Trustor *T* reduces its trust in recommender *R* by *10* and increases its trust in subject *S* by *10*. Finally, the recommender reduces her/his trust in the subject by *10*.

The recommender could make requests to a number of recommenders until the total amount of trust value is reached (the search requests to find the recommenders are not represented in the figures but the issue is further discussed in the evaluation in Section 7.6.2). For instance, in the previous example, two different recommenders could be contacted, with one recommending *3* good outcomes and the other one *7*.

A recommender chain in trust transfer is not explicitly known to the trustor. The trustor only needs to know his/her contacts who agree to transfer some of their trustworthiness. This is useful from a privacy point of view since the full chain of recommenders is not disclosed. This is in contrast to other recommender chains such as public keys web of trust [91]. Because we assume that the entities cannot be compromised, we leave the issue surrounding the independence of recommender chains in order to increase the attack resistance of the trust

[18] In this figure, an entity *E* associated with a SECURE triple *(s,i,c)* is indicated by *E(s,i,c)*.

metric for future work. The reason for searching more than one path is that it decreases the chance of a faulty path (either due to malicious intermediaries or unreliable ones). If the full list of recommenders must be detailed in order to be able to check the independence of recommender chains, the privacy protection is lost. This can be an application-specific design decision.

Thanks to trust transfer, although a real-world identity has many pseudonyms, the Sybil attack cannot happen because the number of direct observations (and hence, total amount of trust) remains the same on the trustor side. Still, local newcomers can be introduced thanks to collaboration. In the previous example, if the subject and the recommender are pseudonyms of the same real-world entity, they remain unlinked. If the proof is given that they can be linked, the fusionym trust value can be calculated, as in Section 5.5.2 with a guarantee of no overcounting of overlapping evidence or self-recommendations. One may argue that it is unfair for the recommender to lose the same amount of trustworthiness as specified in his/her recommendation, moreover if the outcome is ultimately good. It is envisaged that a more complex sequence of messages can be put in place in order to revise the decrease of trustworthiness after a successful outcome. This is left for future work, because it can lead to vulnerabilities (for example, based on Sybil attacks with careful cost/benefit analysis). The current approach is still limited to scenarios where there are many interactions between the recommenders and where the overall trustworthiness in the network (that is, the global number of good outcomes) is large enough that there is no major impact to entities when they agree to transfer some of their trust (such as in the email example in Section 7.6.2). Ultimately, without sacrificing the flexibility and privacy enhancing potential of limitless pseudonym creation, Sybil attacks are guaranteed to be avoided, which is a clear contribution to the field of decentralised, computational trust.

5.6 Summary

Recognition schemes postpone the enrolment phase. In doing so, more dynamic interactions are possible but they may be less secure. The link with the real-world identity may be absent but recognition is sufficient to build trust in a virtual identity based on pieces of evidence. The link with the real-world identity may bring greater system trust but this is not mandatory. In addition, if the link is not mandatory, it enhances the privacy of the users. Furthermore, the users can use multiple virtual identities, which further enhance privacy protection.

However, depending on what benefits can be reaped through trustworthiness, people may be willing to trade part of their privacy for increased trustworthiness: hence, contextual privacy/trust trade is needed. We propose a model for privacy/trust trade based on linkability of pieces of evidence. If insufficient evidence is available under the chosen pseudonym, more evidence may be linked to this pseudonym in order to improve trustworthiness and grant the request. We present a protocol for explicitly disclosing to the requested entity that some evidence can be linked. Some thresholds should be set concerning the acceptable evidence that should be disclosed. This is why we introduce the liseng algorithm to ensure that the Minimal Linkability principle is taken into account. During a privacy/trust trade process, tai, nymity and utility must be balanced.

More generally, depending on which ASUP requirements are more important, the entification framework can be tuned to better fulfil these most important requirements, possibly at the expense of the other requirements. For example, pure ER schemes may be better for privacy and usability. Adaptability is greater when the range of allowed ER schemes is broader. However, pure ER schemes provide less security since system trust based on the real-world identity cannot be enforced.

To entify[19] means to reify: "to regard something abstract as a material or concrete thing". Entification allows us to regard a set of pieces of evidence as an entity on its own. Two main aspects have to be taken into account to compute the overall trust value, which is the result of entification: the technical trust, which is part of the end-to-end trust; and the fusionym, which links the evidence of multiple virtual identities proven to form a single virtual identity. We emphasise that care should be taken when linked evidence on multiple virtual identities is assessed: the main requirement is to avoid overcounting overlapping trust pieces of evidence. Safe fusionym is possible concerning direct observations and trust values based on the count of event outcomes. Due to the possibility of self-recommendations and attacks due to identity multiplicity, fusionym is more difficult when recommendations are used. The technique of trust transfer mitigates these fusionym issues due to self-recommendations and identity multiplicity attacks, such as the Sybil attack. Trust transfer is still limited to scenarios where the number of interactions is important and transferring trust does not significantly undermine the recommenders. According to the Appleseed's trust metric classification (detailed in

[19] Merriam-Webster's thesaurus: http://www.m-w.com/cgi-bin/thesaurus?book=Thesaurus&va=entify

Section 4.1), the trust transfer metric corresponds to a new type: a local decentralised scalar metric. The other main attacks occurring at the identity level based on identity usurpation are alleviated by the use of the level of confidence in recognition and technical trust. Finally, the adaptability to the environmental context is possible thanks to a pluggable recognition module, which is tuneable and logs evidence.

CHAPTER 6: ENTIFICATION/ASUP TEST BED

The entification framework combines computational trust and identity aspects to investigate a solution to the ASUP requirements in global computing. The assumption stated in Chapter 3 is that the computational trust part corresponds to the SECURE trust engine, which has been designed and implemented during the SECURE project. The novel identity approach of the entification framework can be added to the SECURE trust engine thanks to its ER module. Section 6.1 recalls which application domains provide an interesting approximated global computing environment, with regard to which ASUP requirements. Then, instantiations to carry out experiments in these application domains are detailed.

6.1 Approximated Global Computing For ASUP Testing

As found in Chapter 4, because there is no global computing environment available for empirical evaluation, recurrent application domains exhibiting characteristics of global computing environments are usually used. In this thesis, the same approach is used in order to be able to achieve empirical evaluation. The first selected experiment is in the email environment, where anyone should be able to create and use email addresses in a decentralised way but without receiving spam. Then, a location-aware application is selected to study the privacy and context-awareness aspects. Another experiment occurs in the smart home, where auto-configuration is crucial since busy inhabitants cannot be expected to function as full-time administrators.

In order to have a basis to evaluate pure recognition, meaning that neither an a priori trusted-third-party nor the link to the real-world identity are required, we consider the email environment, which corresponds to message-based applications. The first section describes a pure message-based ER scheme. The second section presents an API tool kit, which allows the evaluation of these pure message-based ER schemes and demonstrates how the design of the PRM can be implemented. Then, a first implementation example is done in the email application domain. Section 6.4 shows how the privacy/trust trade can be implemented with the previous message-based ER schemes. Finally, another type of ER scheme, which focuses on the usability and transparency issues for the tenants, is implemented. The final section presents vision-based ER schemes.

6.2 Pure ER Scheme Experiment

The A Peer Entity Recognition (APER) scheme is used for recognising peers on a network, without link to a real-world identity certified by a trusted-third-party. APER assumes that the network supports some form of broadcast or multicast messaging, for example using IP broadcast or multicast addresses, or adopting an application layer broadcast approach. In P2P systems, the implemented propagation scheme may be used (for example, the JXTA propagate pipe [153]). There are two roles distinguished in APER, the recogniser and claimant (though any party can take on any role). The basic approach is for the claimant to broadcast a digitally signed packet (a claim) and for the recogniser to be able to challenge the claimant as desired or simply to recognise the peer on the basis of correctly signed claims. When a challenge is issued, producing a correct response to the challenge requires the claimant to possess the private key used to sign some previous claims. The claimant may include some context information (e.g., time, network address, application layer naming) in claims. There is one further trick used in order to increase the recogniser's level of confidence in recognition. In order to provide evidence that the claim is fresh, and not replayed or copied from some other broadcast network, the claimant is required to (where possible) include within its claim the hashes over the last *n* claims which were seen on the network (by the claimant). If the recogniser has also seen one or more of these (the recogniser is assumed to record its own set of recently received claim hashes) then the recogniser can treat the claim as being fresh. Each level will have some associated parameters (e.g., the number of claims seen), which may also impact on how the recognition is treated. The levels are:

- APERL1: claimants signature verified over a set of recently seen claims;
- APERL2: level 1 and claimants recent claims are fresh, based on the last-n-hashes mechanism;
- APERL3: level 2 and the claimant successfully responded to a challenge.

A claim *c* is composed of the following fields detailed in Table 3, where the first column corresponds to the short name of the field in the claim, e.g., *ctxt*, and the second column gives the description of the purpose of this field:

$$c = \{n, [ctxt], fresh, [this, that]\}$$

Regarding a distribution of recognised peers, an extension to APER is to say that a claim may be signed by *n* different keys, which can be seen as recognition clues. For example, it may be because both keys are indeed owned by the same peer. The following example is when an APER claim is signed by two keys and both signatures are valid:

$$\sum_{i=1}^{n} (lcr_i, vi_i), e.g., \{(APERL1, PublicKey1), (APERL2, PublicKey2)\}$$

It is worth mentioning that, depending on the key length and the cryptographic algorithm used, the clue given by signing claims is more or less strong.

Item	Description
{x}y	A digitally signed form of *x*, verifiable using (and containing) public key *y*
a,b	The comma is used for catenation. *a,b* is the catenation of a and b
[x]	An optional field is enclosed in square brackets
C	Claimant
R	Recogniser
n, n',n''	Nonces, i.e., a long (say 128 bits) random values
Pub	A public key claimed by *C*
Pri	A private key that ought to be *C*'s
ctxt	(optional) Context information, e.g., time, network address, application scope
fresh	A value which provides evidence that the claim is *fresh*, in this case, this contains the last-n-hashes value (during a bootstrapping sequence this may be empty)
this, that	Identifiers for claims used when linking claims together
c	A claim, *c={n,[ctxt],fresh,[this,that]}Pub*
chal	A challenge to *C chal=n'*
Resp	A response to *chal*. *Resp={n'',hash(chal)}Pub*

Table 3. APER Claim Format

6.3 Java-based Claim Tool Kit (CTK) Experiment

The ER process has first been implemented in message-based recognition as the Claim Tool Kit (CTK) in order to be able to use and empirically evaluate the above APER scheme. The main application domain has been email anti-spam. However, it has also been used to sign and link payment transactions and public-key pseudonyms in location-aware scenarios (please refer to the next experiment in Section 6.4).

The core of the tool kit consists of *30* abstract classes and *20* concrete classes in *3,500* lines of code[20]. The `Claim`[21] class is more abstract than an `APERClaim` in order to be able to plug message-based ER schemes different than the APER scheme. In doing so, the CTK is closer to a Pluggable Recognition Module. The APER implementation is made of more than *3,000* lines of code distributed across *20* concrete classes, which are mainly concrete subclasses of the core abstract classes and a few other abstract classes. The abstract APER classes are subclassed in order to give application examples of the full set of classes. With the application specific classes, more than *22,000* lines of Java code have been written and tested for the CTK. All storage classes are abstract and have been implemented to be stored in memory. A secure persistent version of these storage classes could be implemented, for example, as a database or encrypted file storage.

The first step to obtain a CTK compliant to the ER process is to add the claiming functionality to the previous Functional Unit, which already includes the ER module, as depicted in Figure 17, which is similar to Figure 8 with regard to how the Management Unit, Functional Unit and information propagation are represented. Another rectangle of Figure 17 contains the actions of the claiming module. There is also the element of context, which sends information to select the appropriate pseudonym depending on context.

[20] The count of lines of code includes standard and Javadoc comments.

[21] In this section, most of the words starting with upper-case letters are Java classes or interfaces part of the CTK.

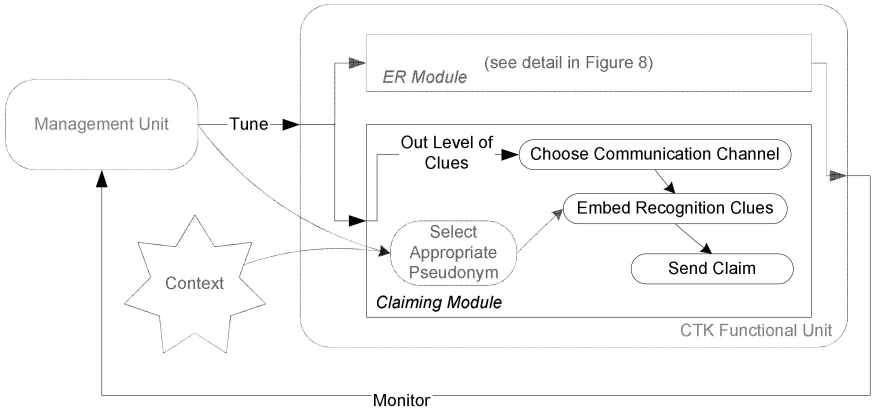


Figure 17. The CTK Functional Unit[5]

The CTK approach is about recognition in a P2P way. In a CTK style of interaction between peers, peers claim statements by sending `Claims` over communication channels. An example of such a scheme is the APER protocol. Practical investigations in the email domain have shown that it is possible and useful to use `Claims` in unicast communication channels. APER focuses on broadcast communication channels. However, the CTK goes beyond a broadcast communication channel (as assumed in APER) and the specific format of `APERClaims`. Generally, a `Claim` carries both `ClaimContent` and `RecognitionClues` (used for recognition during the Detective Work of the ER process).

A CTK has two main responsibilities:

1. it must allow a peer to recognise what peer made a `Claim` (thanks to its `RecognitionClues`);
2. it must allow a peer to make `Claims` over communication channels; any communication channel object can be used as long as it implements the interface `ClaimSendable`.

In order to obtain an internal maintainable CTK, we spread the functionalities of the CTK across different high-level classes based on Class-Responsibility-Collaboration (CRC) cards [150]. We give an example of a CRC card in Table 4 (even though we use a textual representation thereafter). The number of responsibilities *n* should be around three in order to keep a comprehensive tool kit [150]. A responsibility must be implemented if not specified otherwise.

Class Name	
Responsibility 1 of this Class	*This Class collaborates with such and such other Classes to fulfil Responsibility 1 …*
Responsibility 2 of this Class	*Collaborate with … to fulfil Responsibility 2*
…	…
Responsibility n of this Class	*Collaborate with … to fulfil Responsibility n*

Table 4. A CRC Card Example

The `ClaimSender` has the responsibilities: to register the possible communication channels, called `ClaimSendables` (for example, a JXTA pipe or SMTP have been used), needed by the application; to send new `Claims` over a `ClaimSendable`; optionally to select the appropriate `ClaimSendable` based on the `OutCluesLevel`. The last responsibility is closer to a Management Unit concern than the other ones: given the level of threat or the trade-off between performance and security (please refer to the next chapter for evaluation results), the trust engine can tune the `RecognitionClues` and select more secure/dynamic communication channels that are used to send Claims.

The `OutCluesLevel` is a subclass of the abstract `TuningLevel` class, which is Comparable and also a superclass of: `AttentionLevel`, `DetectiveWorkLevel` and `DiscriminativeRetentionLevel`. The following is the pseudo-code to create a new instance of a local CTK, which can send/receive email and recognise the sender of the email according to the APER scheme, and set the `DetectiveWorkLevel` to APERL3:

```
EmailAPERCTK ctk = new EmailAPERCTK(“Bob’s CTK”, “bob@trustcomp.org”);

ctk.setLevelOfDetectiveWork(APERDetectiveWorkLevel.APER_L3);
```

It means that the CTK will try to recognise any sender of emails up to the level 3 of APER confidence in recognition, which implies a signature check of the email, the presence of the hash of past emails embedded in the email and a cryptographic challenge/response. If APERL1 had been used, the ER process would have only carried out the signature check of the email. So, the `Claim` or `APERClaim` carries specific `RecognitionClues` and is assessed with regard to the `DetectiveWorkLevel` that is reached after the Detective Work.

In fact, any `Claim` implements the `RecognitionInformation` interface, described in pseudo-code as follows:

```
Interface RecognitionInformation{

Object retrieveRecognitionInformation(RecognitionClueType) throws
                          ctk.core.clues.NoSuchRecognitionClueException;

boolean hasRecognitionClue(RecognitionClueType);

updateRecognitionInformation(RecognitionClue);}
```

Ideally, from any `RecognitionInformation`, it should be possible to search for specific `RecognitionClues` based on `RecognitionClueTypes`. However, this is not suitable for an object-oriented approach, where a class hierarchy is the common approach. This is an example of implementation trade-off between the genericity of the interface and the object-oriented approach where a detailed class hierarchy is usually expected. The approach used in this implementation is to use a `NoSuchRecognitionClueException`, which is a `RuntimeException` and thus an unchecked exception since it would be too inconvenient for the developer to use `try{}catch{}` elements each time a `RecognitionClue` has to be retrieved. Still, in case it is unknown if a `RecognitionClue` is present, the `hasRecognitionClue` can be used. In addition, the use of a `RecognitionClueType` parameter rather than a text `String` removes misspelling mistakes and minimises the risk of a wrong casting of the object corresponding to the `RecognitionClue` of interest. In fact, each application may have to define its own types of `RecognitionClues`. For example, for `APERClaims`, the clues, represented in the following pseudo-code, had to be created and used.

```
class ctk.aper.clues.CurrentPubRecognitionClue extends
                                 ctk.clues.PublicKeyRecognitionClue{

  public static final RecognitionClueType TYPE =
                              new RecognitionClueType("currentPublicKey",
                              PublicKey.class);

    public CurrentPubRecognitionClue(PublicKey pub) {super(pub, TYPE);}}

                                         …

(PublicKey) theClaim.retrieveRecognitionInformation(
                                        CurrentPubRecognitionClue.TYPE)));
```

This is the reason that different packages have a `.clues`, which contains the set of `RecognitionClues` that could be needed in the application domain of interest.

In addition to `Claim`, a few other classes implement the `RecognitionInformation`. Indeed, a virtual identity, which has previously sent a `Claim`, corresponding to the `Claimant` interface can be recognised. A local virtual identity controlled by the owner of the CTK, that is, a pseudonym, can also be recognised and is called a `MasteredClaimant` (for example, the control can be represented by the ownership of the private key associated with the public key). A CTK should allow a peer to manage several `MasteredClaimants` (i.e., to make `Claims` under the name of different `MasteredClaimants`), especially due to privacy protection, which is encouraged in the entification framework where multiple pseudonyms per real-world user are recommended. In Section 6.5, we demonstrate an implementation of the CTK where the peer automatically sets the `MasteredClaimant` based on location (it corresponds to the greyed *Select Appropriate Pseudonym* in Figure 17). The CTK may provide the functionality to link `Claimants`: to claim that one or more `Claimants` are indeed originated from the same peer, which owns these different `MasteredClaimants`. However, the `Linkage` strength and verification is application dependent (an implementation example is given in Section 6.5). In broadcast settings, a CTK should allow a peer to suppose what other peers heard previous `Claims` or are in-the-know of them. For example, it may be helpful for weak authentication (please refer to Section 5.3) schemes based on spatial separation to detect MIM attacks. In this case, the class representing the peer is called `ClaimHearer`. `Claims` are said to be heard by `ClaimHearers`. When `Claimants` send `Claims`, they can specify what `ClaimHearer` is supposed to hear (i.e., receive the `Claim`) and so `ClaimHearer` is `Serializable` in order to be sent over communication channels. When a `Claim` is received, the local receiver peer can associate as an assessment result what

ClaimHearer is supposed to hear it during the Discriminative Retention phase. The default implementation just copies the ClaimHearers, which are specified with the Claim, if the LevelOfConfidenceInRecognition reaches the current specified DetectiveWorkLevel. Other implementations may try to carefully assess which ClaimHearer should be listed, for example, a subset of the ClaimHearers explicitly specified in the Claim.

In fact, a Claim, after its assessment, especially after the Detective Work, is stored as an AssessedClaim. The ClaimHearers are part of the ClaimAssessmentResults and useful to compute the hashes of the Claims that should have been heard by the target of a new Claim, that is, to carry out a recognition based on shared history of past Claims, such as APERL2. For Freshness (the mechanism for checking that different peers have a degree of common past Claims, e.g., the fresh field in APER), there are different possibilities for combining the past Claims. The number of past Claims to use can be specified. The default combination consists of the calculation of the hash of each Claim and the resulting hashes form a List. Then, the Management Unit can tune when a Claim is considered to be fresh by setting how many hashes should be found in common locally and inside the new Claim. An example of an alternative combination of hash may be to use the hash of the previous hash (i.e., hash chaining). The ClaimAssessmentResults also contain the EROutcome, which lists the RecognisedClaimants (pairs of Claimant and LevelOfConfidenceInRecognition) and InformationForUpperLevelActions, which is an Object depending on the application domain.

The AssessedClaimStore has the responsibilities: to store (in collaboration with the ClaimDiscriminativeRecognitionRetentor) and retrieve AssessedClaims in common with a given ClaimHearer or supposed to have been sent by a given Claimant; it should garbage collect useless AssessedClaims in collaboration with the ClaimManager optionally based on context (this responsibility is closer to a Management Unit concern than the other ones).

The ClaimManager has the responsibilities: to manage the Claims (for example, to retrieve/store AssessedClaims, especially in common with the target ClaimHearer, in collaboration with the AssessedClaimStore; to know what Claims are sent/received and how in collaboration with the ClaimSender); it should enforce policies concerning whether to send `Claims` or not (e.g., for privacy protection reasons) optionally based on context (this responsibility is closer to a Management Unit concern than the other ones).

The `ClaimantManager` has the responsibilities: to manage and forget the `Claimants` and `MasteredClaimants` (e.g., to create them according to the correct cryptographic algorithms); to return the long local identifiers of the `Claimants,` which are recognised according to the given `RecognitionInformation`; it should enforce policies regarding the choice of the `MasteredClaimant` to be selected to send `Claims` (e.g., for privacy protection reasons) optionally based on context (this responsibility is closer to a Management Unit concern than the other ones).

The `ClaimHearerManager` has the responsibilities: to manage the `ClaimHearers` including to store new ones in the `ClaimHearerStore` and optionally to garbage collect useless `ClaimHearers` optionally based on context (this responsibility is closer to a Management Unit concern than the previous one).

It may happen that during the ER process, when previously unknown `Claimants` are being processed, temporary `Claimants`, representing these unknown `Claimants`, must be created. They will only be managed by the `ClaimantManager` if it is decided so during the Discriminative Retention step, which depends on the implementation of the abstract `ClaimDiscriminativeRecognitionRetentor` class for the application under use. For example, the default implementation logs a warning message if APERL1 is not reached, which means that the signature check failed, and manages any recognised `Claimant`, even newcomers. A `Claimant` can be managed or unmanaged. Any unmanaged `Claimant` object has its long identifier set to `Claimant.UNMANAGED_LOCAL_ID` (i.e., *-1*).

The `ClaimantStore` (which is an interface) has the responsibilities: to store and retrieve `Claimants` and `MasteredClaimants` based on the given long identifier; it should garbage collect (that is, forget) useless `Claimants` in collaboration with the `ClaimantManager` (for scalability reasons) optionally based on context (this responsibility is closer to a Management Unit concern than the other ones).

In order to facilitate the sending of new Claims, two wrappers class are provided. SendingCluesContentBundle wraps the RecognitionInformation and ClaimContent of the new Claim. SendingConfiguration contains the necessary information to send a Claim over specific communication channels (given the number of the registered ClaimSendables to use and their required parameters) to a specific ClaimHearer, optionally under the name of a specific MasteredClaimant (given its long identifier).

The ClaimBuilder has the responsibilities: to build new Claims given the SendingCluesContentBundle and SendingConfiguration; to add more or less strong RecognitionClues depending on the OutCluesLevel given by the Management Unit.

The ClaimListener has the responsibilities: to discard/filter Claims according to the current AttentionLevel; to start the Triggering step of the ER process for accepted Claims; it should provide the ER Workload (that is, information about what and how often the ER process is triggered). It also implements the ImplicitRecognitionClueAble interface because implicit RecognitionClues can be carried by specific incoming communication channels.

The ClaimDetective has the responsibilities: to compute the EROutcome from a new received Claim; to do more or less Detective Work based on the current DetectiveWorkLevel; optionally to update the average InCluesLevel. An implementation of the ClaimDetective may provide a helper method to create a new Claimant, which is not yet managed but contains the RecognitionInformation sufficient to the application domain.

The ClaimDiscriminativeRecognitionRetentor has the responsibilities: to discriminately manage Claimants and ClaimHearers found in new AssessedClaims; to discriminately retain the AssessedClaim including the associated ClaimHearers (for example, Claims due to APERL3 C/R are not stored); it may depend on the DiscriminativeRetentionLevel. This class indeed carries out the Discriminative Retention step of the ER process. It may be where ER exceptions are caught (e.g., an exception is logged if a Claim with an invalid signature is received).

The `ClaimActioner` has the responsibilities: to process the last step of the ER process, which consists of using the `EROutcome` for Upper-level Actions related to the application domain; to pass the `EROutcome` to registered `EROutcomeListenable` (the objects to be notified when new `Claims` come in); to reply to a challenge, and more generally to carry out automated actions related to `Claims` specific to the ER scheme (for example, the implementation of the `APERClaimActioner` carries out the actions expected when the `Claim` is related to an APERL3 C/R).

All the above classes and interfaces are used to carry out the ER process. However, in order to achieve a more cohesive implementation of the logic and decrease coupling between the different objects, we apply the Mediator pattern, which uses an object to coordinate state changes between other objects instead of distributing the logic over the other objects. So, the `ERProcessMediator` class implements the flow between the main methods of the ER process, which are listed in the `ERProcessable` interface: `triggering`, `detectiveWork`, `discriminativeRetention`, `passForUpperLevelAction`, optional `forget` (that is, garbage collection of what becomes useless to increase scalability, for example, resources spent to store `AssessedClaims` relating to `Claimants` that are unlikely to be met again).

As said above, for each new application domain, the implementation classes must be written from the above abstract classes and their implementation examples. For example, an `APERClaimant`, which extends `Claimant`, is mainly recognised by a `PublicKey`, therefore the `CurrentPubRecognitionClue` has been created. The `APERMasteredClaimant`, which extends the `AbstractMasteredClaimant`, has access to the `PrivateKey` associated with the `PublicKey`. The `APERClaimBuilder` has the responsibilities: to build the `APERClaim`, which includes the `APERFreshness`, to be sent (in collaboration with the `ClaimManager`) according to the specified cryptographic algorithms and number of hashes specified in the `APERFreshnessType`; to sign a new `APERClaim` according to the specified cryptographic algorithms. In order to implement the `AntiSpamAPERClaimBuilder`, which extends the `APERClaimBuilder`, the only part of the code that had to be written, consisted in retrieving and adding specific `RecognitionClues` in the following method (in pseudo-code).

```
buildNewClaimWithMoreOrLessClues(sendingCluesContentBundle,
                                sendingConfiguration ){

APERFreshnessType freshType = (APERFreshnessType)
                           sendingCluesContentBundle.getRecognitionClues().
                           retrieveRecognitionInformation(
                           FreshnessTypeRecognitionClue.TYPE);

                                            …

newAPERClaim.updateRecognitionInformation(
                           new SenderEmailAddressRecognitionClue(
                           "Bob@trustcomp.org"));}
```

The APERClaimDetective class is in charge of carrying out the Detective Work on APERClaims. Due to the fact that the APERL3 relies on APERChallengeResponse included in APERClaims between the Claimant and the local peer, the issue of no response from the Claimant demonstrates that the Detective Work can take more or less time and potentially never terminate. This is a general property of the Detective Work step; more Detective Work may lead to a better LevelOfConfidenceInRecognition but the counterpart is that it also takes more time. This trade-off between the LevelOfConfidenceInRecognition reached and time is related to the performance evaluation results in Section 7.4. Performance results help the Management Unit to choose the best trade-off, for example, in case of a trust engine as the Management Unit, the risk analysis component is used for the decision. In order for the CTK to handle other Claims whilst in the process of a C/R, the CTK uses a pool of APERClaimDetectives, which extends PoolAPERClaimDetectives. There is a maximum time that can be set to decide that the C/R has failed (even though no response Claim has been received).

Finally, the classes are organised in a comprehensive set of packages, for example: ctk.core; ctk.core.clues; ctk.core.util; ctk.aper; ctk.aper.clues; ctk.email; ctk.email.clues; ctk.memstore; appexamples.privacytrustmaps; appexamples.secure…

6.4 CTK/SECURE Email Anti-spam Experiment

The SECURE trust engine, implemented as an email proxy, has been used to fight spam. The use of the ER process implemented with the CTK and APER-related schemes has proven to be valuable in increasing the security protection in email settings, without too much inconvenience for the users. In fact, it provides new anti-spoofing techniques. This is an example where the assumptions that “the adversary can eavesdrop on all messages that are sent and received” and “if this [previous] possibility can be discounted then there is probably no need to apply security at all” [22] is too strict. It is evaluated thanks to an economic threat analysis including the performance cost of the anti-spoofing mechanism in the next chapter.

Firstly, we explain how a trust engine can be used as an email proxy to prioritise emails according to the trustworthiness of their sender. Then, we present how we obtained real networks of email users. Finally, the new anti-spoofing techniques are detailed.

6.4.1 The CTK/SECURE Email Proxy

The SECURE Trust/risk-based Security Framework (TSF) [154] is implemented in Java: its kernel and API is application neutral, and contains around *7,000* lines of code. In our email settings, we use one SECURE triple and we map *(s,i,c)* to *(non-spam emails, yet to be read emails, spam emails)*. For instance, if sender Alice has been spoofed once by spammer Malory and receiver Bob has read *26* emails (including the spoofed spam email) from the *30* emails received so far with Alice’s email address, then Bob’s trust value for Alice is *(25, 4, 1)*. Note that we assume that Bob forms his opinion on the quality of an email only after it has been read.

In order to be able to use the CTK in email settings to enforce anti-spoofing, both receivers and senders simply need to point their email client to a proxy, called the CTK/TSF proxy [156, 158] (please refer to Figure 18), which can be run either locally on the user’s machine, integrated in their standard mail server or managed by a service provider (as depicted in Figure 31). *3,000* further lines of code have been added to the SECURE kernel to obtain the running proxy.

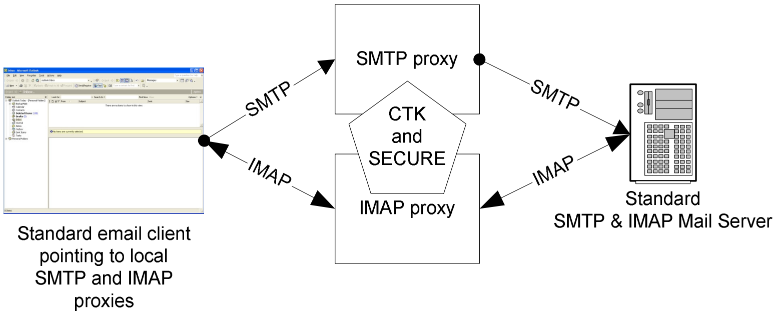


Figure 18. CTK/TSF Email Proxy

6.4.2 Trust value-based Email Prioritisation

We assume that each email user uses our CTK/TSF proxy (already depicted above in Figure 18). The trust value is composed of one triple per email sender, which represents the number of emails considered of good quality received from the sender from the point of view of the recipient, i.e., the email user. The emails in the Inbox are prioritised according to the trust value of the sender thanks to a column in the email client graphical user interface ordered by:

$$\frac{s}{s+i+c}$$

In this situation, *s* and *c* correspond respectively to the number of good and bad quality emails from the sender. The *i* element of the triple corresponds to unread email in the Inbox. Several unread emails from the same sender are prioritised from the oldest received email to the most recent. Many email addresses can be *pre-trusted* by external means, for example, the email addresses in the to:, cc: and bcc: fields and those appearing in the address book. Pre-trusted email addresses get the value *(1,0,0)*.

If the email is considered too highly prioritised by the user at time of reading, the user can specify in one click that the email prioritisation is wrong and *c* is increased by *1*; otherwise the sender's *s* value is increased by one after the email is closed. The trust values are recomputed and emails are reorganised in the folders after each user reading and feedback cycle.

If a spammer sends an email with a disposable email address that will never be reused, the email will end up with the lowest prioritisation, which is *0*. Collaboration between email users is used to prioritise emails from legitimate users. Thanks to the user's feedback with regard to the quality of the emails received, the trust value may also be used as the recommending trustworthiness of the sender since senders with higher trust values are likely to prioritise senders of emails of the same good quality.

6.4.3 A Real Social Network of Email Users

As the above survey of related identity and trust frameworks in Chapter 4 shows, the evaluation of trust metrics often requires to take into account the network typology of collaborating peers (e.g., [69, 194]). One may argue that even though some algorithms generate networks with small-world or scale-free properties, these networks are not similar to real-world networks of email users [158]. Therefore, we mined different online databases such as linked FOAF profiles and newsgroups to extract different real-world networks of email users. More than *8,000* FOAF profiles were retrieved from one person, which shows the privacy issue behind these unprotected networks of detailed profiles and their contact. Google's archive of the main usenet and newsgroup are very useful to do so. The following archives have been mined: *rec.arts.books.hist-fiction* from the 16th of September 2004 to the 24th of February 2001; and then the previous 1,000 threads from the 16th of September 2004 of *rec.music.artists.springsteen*, *alt.music.bruce-springsteen*, *rec.arts.movies.current-films*, *alt.fan.tolkien, alt.movies* and *rec.arts.books.tolkien*. Each contributor to the same thread is considered a contact of the others contributors to the thread. The threads with only one contributor are discarded. Two contributors appearing in two different threads are not considered as contacts if they have never contributed to the same unique thread. For example, the mining of *rec.arts.books.hist-fiction* provides a network of *909* different email contributors, which are connected according to Figure 19. The Java JUNG [92] library is used to generate the graphs according to the Fruchterman-Reingold algorithm for node layout [63], where the email users that are connected attract each other and unrelated email users repel each other. The privacy of each email user is protected by changing the address to a numbered vertex associated with edges to his/her contact email addresses appearing in his/her address book (as depicted in the zoom of the network view in Figure 20).

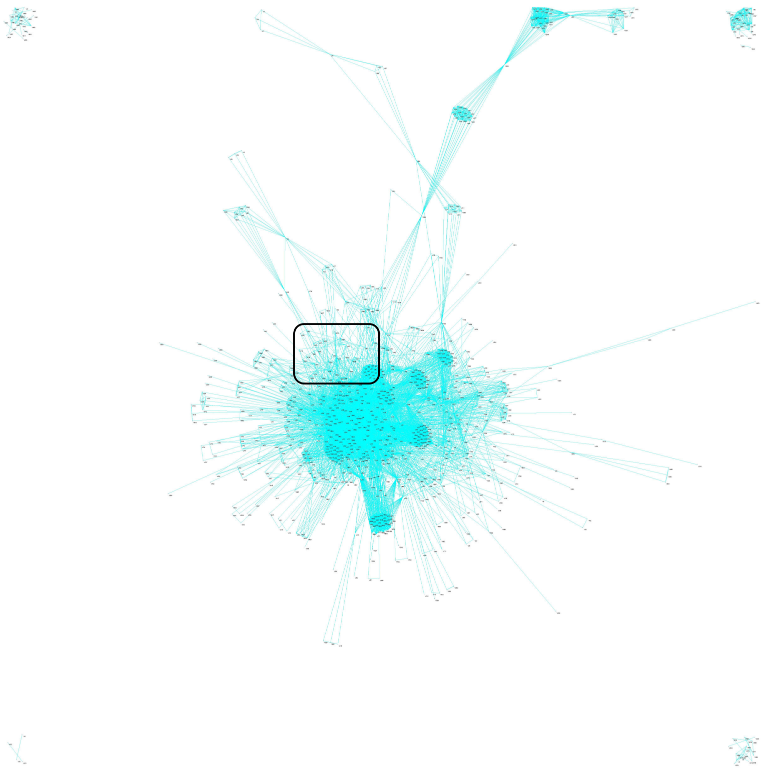


Figure 19. The Real-World Network of Email Users and Zoomed Area Position

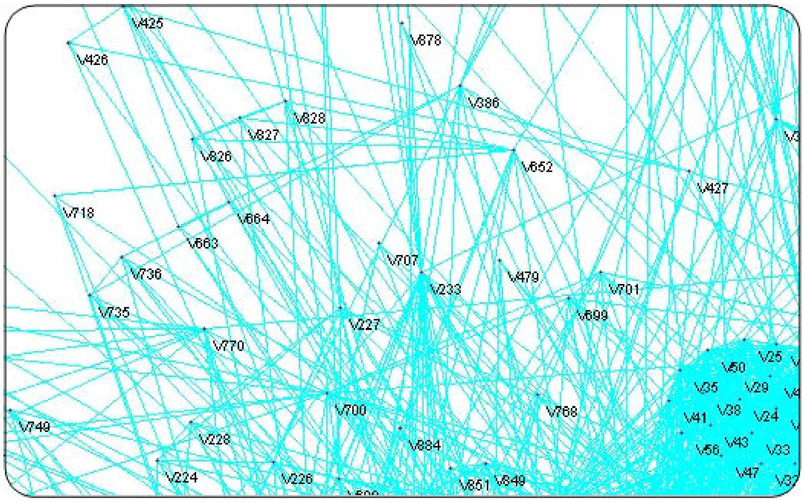


Figure 20. Zoomed Network View

6.4.4 New Decentralised Anti-spoofing

To prevent spoofing without making any changes to the core of the legacy email system, we use the combination of two new techniques [156] (available in the CTK): proof of knowledge of a shared message history and an automated proxy-based challenge-response system.

In this instantiation of the CTK, a claim is simply a MIME multipart email that can be sent over (and without changes to) SMTP. One of the MIME parts is a serialised Java Claim object.

Anti-spoofing based on Shared History

The first CTK ER scheme is based on past and shared history/knowledge between the email sender and receiver. Both should be more or less aware of the content of previous messages (please see Figure 21). So, we keep hashes of previous messages and offer the possibility to send some of these hashes with the emails in order to prevent spoofing. The email address is not considered spoofed if the previous history is known, that is, by verifying that some of these hashes are also found on the receiver's side. It may be misleading to require finding all previous hashes due to the fact that SMTP does not guarantee the delivery of an email. This is a new approach to embed common hashes between the sender and the specified receivers at time of sending. Different strategies are possible to decide what and how many hashes should be found.

The second technique that we provide is to send a challenge to the sender in order to check that he/she is the real initiator of the email and owns the email account bound to the email address. The C/R may consist of a cryptographic challenge but it may also be based on the ability to send a hash of the last email received including some random data (also know as "salt", which is depicted in Figure 21).

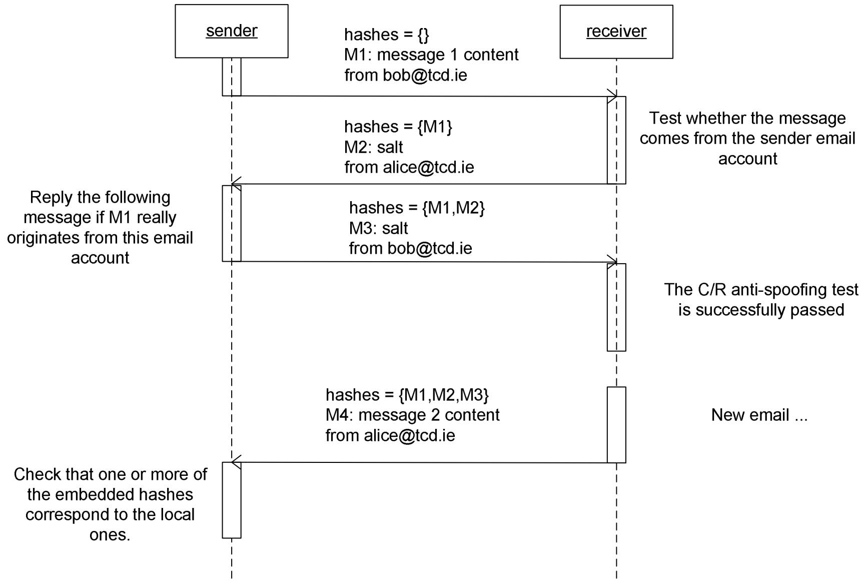

sender
receiver
hashes = {}
M1: message 1 content
from bob@tcd.ie
hashes = {M1}
M2: salt
from alice@tcd.ie
Test whether the message comes from the sender email account
Reply the following message if M1 really originates from this email account
hashes = {M1,M2}
M3: salt
from bob@tcd.ie
The C/R anti-spoofing test is successfully passed
hashes = {M1,M2,M3}
M4: message 2 content
from alice@tcd.ie
New email ...
Check that one or more of the embedded hashes correspond to the local ones.

Figure 21. Typical Newcomer Bootstrapping Sequence[11]

Many different C/R systems have been proposed [175], but we believe ours is fundamentally different from previous systems, because the challenge is usually used to confirm that the email was sent by a human rather than an automated spammer. A second class of C/R systems are those which attempt to charge a fee to the sender by, for example, asking them to carry out a lengthy computation before their email can be delivered. In contrast to both of these types of system, our technique relies on a fully automated proxy-based C/R, which does not involve humans. Indeed, we only verify that the sent email was really sent from the email account associated with the email address, using shared knowledge of previously exchanged emails (although our system could be combined with the concept of bankable postage [3]).

[175] also lists some common bugs in C/R systems (mistakenly categorised as unworkable flaws by others [165]) and explains how to counter them. For example, the sending of unintelligible messages to users who do not use our system, for example due to automated challenges sent by our system, cannot happen. The reason is that it is possible to check whether the sender of an email uses our system or not based on the email parts. Special proxy-related emails are never delivered in the receiver's *Inbox*. If the user does not participate, our system does not send C/Rs or proxy-related emails. The protocol is also designed to prevent the occurrence of an infinite loop of challenges between proxies.

One bug which is difficult to address is preventing malicious senders using the C/R system to distribute spam via the challenge. In our system, the text body of the challenge is under our control so it is not possible to advertise anything by this means, and therefore it cannot be a profitable spam attack. However, a challenge might be sent to a non-participating sender by this means which is irritating to the recipient of the challenge, even if not useful to the spammer. This is not a new attack since "most SMTP servers can [already] be made to respond with a 'bounce' to a faked address" [175]. To mitigate this annoyance, the body of the challenge explains to the receiver that they should not have seen this email and that it is possible to discard any such email by using the special header flag that we embed in all emails generated by our proxy. Since this flag is well-known, it may be provided in advance to the most widespread email client filters, even if they do not implement our system.

[165] also raises other issues related to the use of C/R systems for email which we believe are effectively countered in [175]. As stated above, our method places no additional burden on the sender of email since the protocol is conducted by automated proxies, and as with bankable postage, known addresses may be whitelisted in advance – Templeton [175] presents a useful list for this purpose. For example, all email addresses present in his/her address book are automatically whitelisted. We generalise this approach by calling it pre-trusted. Since no user intervention is needed, the C/R emails are exchanged at the speed of the standard email system.

No change is necessary for senders who do not use the proxy, although their emails are intrinsically less trustworthy due to no anti-spoofing protection, for example, their message might end up being assigned a low priority (due to lower end-to-end trust) by receivers who use our proxy. Unfortunately for the user of our system, it is not easy to know whether an intended recipient who the user has not dealt with before is a user of the system or not, and therefore whether to send a normal email or a one with a claim attached. Since if an unknown MIME part is received, it is simply added as text at the end of the body of the email (or as an attachment), it is perfectly acceptable to speculatively include a claim in the initial email, then if no challenge C/R is ever received back from the new receiver, it is considered that the receiver does not run our type of proxy and the next emails sent will just be normal emails.

Email-based identification and authentication [65] has shown that successful C/Rs sent to an email address provide a proof of ownership, which usually involves the user's intervention to manually confirm. It has been used for a variety of tasks (for example, password resets) "because it combines ease of use with a limited challenge-response system that is not trivial to defeat" [65]. In our approach, the confirmation is transparent, without human confirmation, because the response is automatically computed and sent back. It is related to [14, 182] combined temporal and spatial separations weak authentication and less straightforwardly to the "application semantics" and "asymmetric costs" of the email system. The use of past hashes relates to cultural history-based passwords [168]: the rest of the entities are unlikely to know the shared history of exchanged emails but it is not really secret (since it is not encrypted by default). However, in contrast to pure cultural-based password, the fact that once the attacker has learnt them then they cannot be changed, is mitigated by the fact that old hashes may become obsolete since the list of hashes changes as emails are exchanged.

Adjunct Asymmetric Cryptography Protection

Our CTK also supports traditional asymmetric (public-key) cryptographic signatures as yet another possible technique to address recognition. Note that, unlike in traditional signature methods, there is no need to bind the key to a real-world identity – the key needs only to be bound to the email address account. A CTK bootstrapping protocol using C/R, which this time can be based on a cryptographic nonce challenge signed by the receiver's private key, is a means for this binding. The response must be signed by the sender's private key and once the bootstrapping is complete, it may be sufficient to rely on local checks of shared hashes of past messages and not use challenge/response each time an email is received. The extended sequence is described in Figure 22.

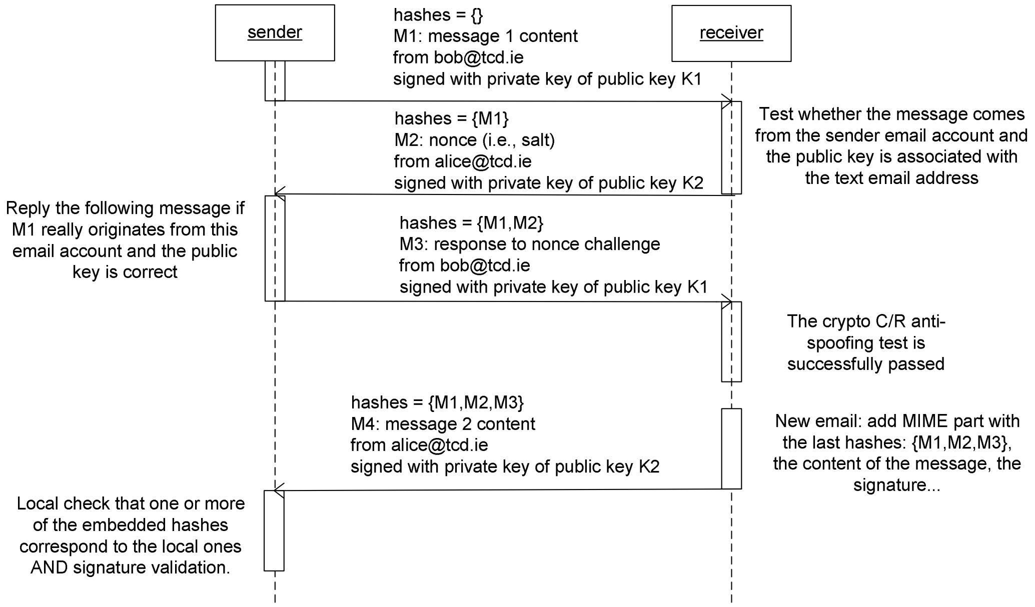

sender
receiver
hashes = {}
M1: message 1 content
from bob@tcd.ie
signed with private key of public key K1
hashes = {M1}
M2: nonce (i.e., salt)
from alice@tcd.ie
signed with private key of public key K2
Test whether the message comes from the sender email account and the public key is associated with the text email address
Reply the following message if M1 really originates from this email account and the public key is correct
hashes = {M1,M2}
M3: response to nonce challenge
from bob@tcd.ie
signed with private key of public key K1
The crypto C/R anti-spoofing test is successfully passed
hashes = {M1,M2,M3}
M4: message 2 content
from alice@tcd.ie
signed with private key of public key K2
New email: add MIME part with the last hashes: {M1,M2,M3}, the content of the message, the signature...
Local check that one or more of the embedded hashes correspond to the local ones AND signature validation.

*Figure 22. Extended Newcomer Bootstrapping Sequ*ence[11]

Effectively, the requirement is changed from the need to authenticate a real-world identity to the ability to recognise a triggering entity for whom trust information can then be accessed. Generally, in order to increase the level of confidence in whether it is a spoofing attack or not, challenge/response, check of common hashes and signature verification as well as other recognition/authentication schemes may be combined.

We need anti-spoofing techniques in order to be able to recognise email addresses, which become trustworthy thanks to the use of a trust engine. Obviously, our techniques differ regarding their level of confidence in recognition and technical trust. However, there is no exact way to say that one technique is weaker than another one. For example, it is not straightforward to choose which of the following offers the higher level of confidence: a valid signature with a very short asymmetric key, which has been used for years, or the ability to show that the sender is able to receive emails sent to a specific email address. In Section 7.3, static and dynamics means to estimate the technical trust are evaluated.

By using either our proxy-assisted C/R anti-spoofing technique or our verification of common hashes technique, we get a level of confidence in the binding between the text email address and the ownership of the email account. The technique based on hashes has the advantage of local verification. It may also be more feasible to be applied to resource-constrained devices than resource-consuming asymmetric encryption. However, it cannot be used for the very first exchange of email because the sequence contains no previous email (or if all the hashes have been lost). Fortunately, the C/R technique allows the sender to bootstrap with the receiver. After C/R bootstrapping, common hashes comparison is used. However, once the bootstrapping is done, in order to minimise the overhead of emails sent due to our approach, the possibility to check whether the correct hashes are present or not is valuable because the check can be done locally. This overhead is also evaluated in the next chapter.

6.5 Context-aware Privacy/Trust Trade Experiment

The link mechanism has been implemented in a context-aware pervasive computing environment based on the CTK [163]. We consider a mobile commerce scenario, where anonymous digital cash resides in an electronic purse on the customer's mobile phone. The anonymous digital cash can be used for payment of small amounts, for example, public transportation, snacks or groceries at the local corner shop. Associated with every purse is a unique identifier that cannot be traced back to the customer and which the customer can change at will, e.g., different identifiers may be used with different merchants. This identifier allows the merchant to recognise returning customers, without violating customer privacy. Because of the inherent problems of double spending in anonymous offline digital cash, merchants may only accept small amounts from previously unknown customers, but if the digital cash is redeemed by his/her bank larger amounts may subsequently be accepted. If the customer uses the same virtual identifier in all shops, the local council of commerce will eventually be able to establish a full spending profile for all customers, which they may use for direct marketing or for credit approval. This would be a violation of the customer's privacy. In this scenario, computational trust is used to reduce the inherent problem of double spending in anonymous digital cash systems, while virtual identities preserve the privacy of customers.

The e-purse identifier corresponds to the public keys of key pairs generated locally by the e-purse. The default implemented privacy disclosure policy is to automatically create and select the correct public key based on location and squared areas. The user can set up squared privacy areas starting from the home location. It is also possible to change the size of the squares. For example, users can reduce the area to *50m* which would allow them to use different pseudonyms in different shops (or *500m* to allow different pseudonyms in different parts of town). The user may also select another mode, called *One-Time*, which creates a new public key each time a new claimant is met. In fact, public keys correspond to pseudonyms. In order to ease pseudonym management, we provide two maps. The maps represent Europe and can be zoomed in and out. For now, the user's location is changed by moving a pink circle on the map; a GPS/Galileo module would dynamically change the position of this circle. The user's home is displayed as a green rectangle. The first map is the privacy map [163], where privacy areas covered by pseudonyms are displayed. By clicking on each zone, information about claimants (that is, vendors) bootstrapped with the pseudonym associated with the zone and their associated information (such as, content of past transactions) can be displayed. The goal is also that the user, by using queries or clicking on the map, can easily retrieve any information about any claimants or mastered claimant (time of bootstrapping, trustworthiness, mastered claimants disclosed to claimants, textual information entered by the user…) as well as sets of entities (for example, all claimants bootstrapped in a specific area). In Figure 23, the zones covered by the user's pseudonyms are represented in the Graphical User Interface (GUI) as 6 rosy squares. The second map is called the *trust map* [163], where all vendors bootstrapped so far are displayed as small rectangles. The trick is to change their colour according to their trustworthiness [163] in order to quickly understand the trustworthiness of vendors (as depicted in Figure 24) as well as areas.

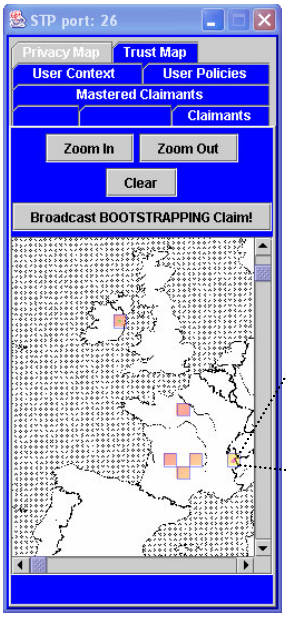


Figure 23. Privacy Map

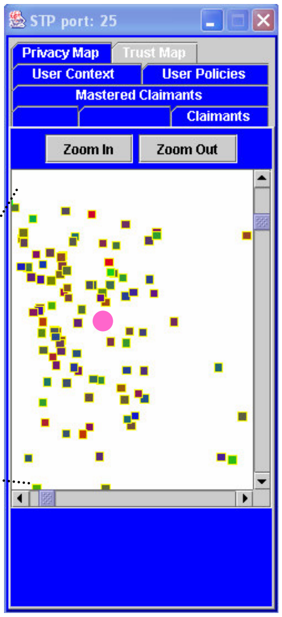


Figure 24. Trust Map

Linkability of different transactions with a specific virtual identity is achieved by using the CTK for transactions between the two virtual identities. The approach is for the claimant to send claims, i.e., digitally signed messages, and for the recogniser to be able to recognise the claimant on the basis of correctly signed claims. So, transactions are linked through asymmetric key digital signature validation using the same key, which provides the APERL1 level of confidence in recognition. The requested entity can refer to a specific virtual identity (e.g., in order to get recommendations about a specific virtual identity) by specifying the public key, which is contained in the recognition clues of the claim sent by the requesting virtual identity.

As an example, the following figure depicts the scenario where Alice plans to spend her holidays in SunnyVillage. Normally Alice works and lives in RainyTown. She will take the plane and relax for two weeks in this village where she has never been but that some of her friends recommended. She will have to pay to enjoy some of her leisure activities, which could be enhanced if collaboration with other local entities is allowed. We assume that Alice uses an e-purse. So, an e-purse is associated with public key (*Pub*) / private key (*Pri*) pairs: a public key becoming a pseudonym for Alice. An e-purse has also an embedded trust engine, which takes care of trust decision-making and management. Similarly, a vendor's cashier-machine can be recognised with a public key and runs a trust engine. For example, exchange of Alice's trustworthiness in being a good payer in the neighbourhood would let her rent a large video display without being asked for real-world credentials (for example, a passport that she has forgotten at the hotel); credit may also become viable. Vendors would also benefit from computational trust adjunct. The video shop of SunnyVillage, having to deal with passing customers, would be reassured to take a lower risk if payment with electronic coins is combined with the level of trust in the customer. Nevertheless, Alice also wishes to protect her privacy and have different social profiles in different places. Alice has indeed two pseudonyms automatically selected according to location: one in RainyTown (*PubAliceRainyTown*) and one in SunnyVillage (*PubAliceSunnyVillage*). This offers better protection for her privacy than having one pseudonym. Even though the video club holding spans both domains, SunnyVillage's video club cannot obviously link *PubAliceRainyTown* and *PubAliceSunnyVillage* by comparing keys known by RainyTown's video club. The latter would not be true with a unique public key for Alice's e-purse.

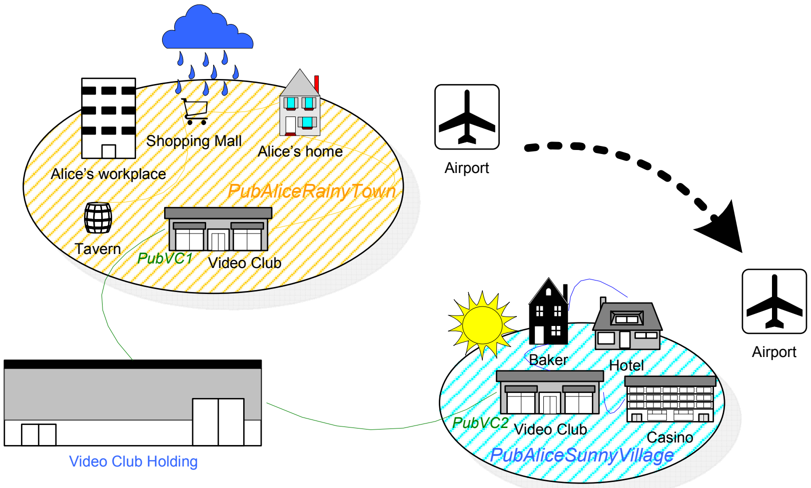


Figure 25: Alice's Smart World[22]

However, trust, as with privacy (as it is explained in Section 4.3.2), is dynamic and evolves interaction after interaction. Depending on what people can get based on their trustworthiness, they may be willing to disclose more of their private data in order to increase trust. There is a need for contextual privacy/trust trade. Let us assume that the trustworthiness of people for being good payers is managed by the trust engine of the vendor's cashier-machine. Recalling the scenario in Figure 25, if Alice arrives in SunnyVillage's video club for the first time, her e-purse will exhibit *PubAliceSunnyVillage* when she wants to pay for the large video display that she wants to rent. Since no direct observation, that is, a previous experience with *PubAliceSunnyVillage*, is available, *PubVC2* (the SunnyVillage video club cashier's public key) will ask for recommendations from its neighbours (for example, *PubBaker*). However, Alice's trust obtained through recommendations is not enough to commit the renting transaction because she has made too few transactions in SunnyVillage and cannot present her passport left at the hotel. Alice really wants the display, so she is now disposed to give up some of her privacy in order to exhibit enough trust. In fact, SunnyVillage's video club is held by a holding of video clubs, which has a video club in RainyTown. The following example of contextual privacy/trust trade is started. The list of public keys owned by the holding is sent to Alice's e-purse, which finds that *PubVC1* of

[22] The dashed black arrow represents Alice flying to her holidays location. The ovals indicate the different geographical regions covered by Alice's different public keys. The lines connecting the different symbols (for example, the blue line between hotel and baker) represent trust relationships and indicate that recommendations are exchanged between the entities depicted by the symbols.

RainyTown's video club is a known virtual identity. Alice has noticed that she could link *PubAliceRainyTown* and *PubAliceSunnyVillage* in order to reach the necessary level of trust. Although Alice now knows that what she has done in RainyTown is potentially exposed to both areas, that is, RainyTown and SunnyVillage, she agrees to present herself as the owner of both keys/pseudonyms.

Therefore, the outcome of the ER process can be a set of *n* virtual identities *vi* associated with a level of confidence in recognition *lcr*:

$$\sum_{i=1}^{n}(vi_i, lcr_i), e.g.\{(Pub_1, APERLevel1), (Pub_2, APERLevel1)\}$$

The above example is when an APER claim is signed by two keys[23] and both signatures are valid. The following sequence of interactions carries out the privacy/trust trade process when pseudonyms are linked. Let vi_1 be the requesting entity and vi_2 the requested entity, they exchange the following APER claims with special keywords in the *ctxt* field[24].

1: $vi_1 \rightarrow vi_2$: [GRANTX]vi_1

2: $vi_2 \rightarrow vi_1$: [TAIGAP,HINT]vi_2

3: $vi_1 \rightarrow vi_2$: [LINK]vi_1,…,vi_i,…

In step 2, *HINT* is optional and may contain hints for optimising the liseng on the requesting entity's side. In fact, it may say which recommenders have been used for the first round of the trustworthiness assessment. It would then be known that it is useless to send back a link for the same recommenders. In our scenario, the *HINT* consists of a list of other virtual identities (video clubs) owned by the video club holding company. Then, on Alice's side, the liseng should try to link evidence to these virtual identities. In step 3, the *LINK* lists other public keys that are linked to vi_1 and the claim must be signed by the private key of each listed public key.

[23] We refrain from using other technical trust pieces of evidence (for example, key length and type of algorithm used). The next chapter demonstrates how they could be used.

[24] We use the notation: *X* is the special keyword used in the *ctxt* field of an APER claim, *vi* is a virtual identity; $vi_1 \rightarrow vi_2$ means that an APER Claim is sent from vi_1 to vi_2; $[X]vi_1,\ldots,vi_i,\ldots,vi_n$ means that *X* is signed by several private keys, for example, *vi_i's private key*.

For example, in Alice’s scenario, we have:

1: $p_1 \rightarrow p_2$: [GRANTX(“rent large video display”)]PubAliceSunnyVillage

2: $p_2 \rightarrow p_1$: [TAIGAP(“strong positive impact”),HINT(“PubVC1”)]PubVC2

3: $p_1 \rightarrow p_2$: [LINK(“PubAliceSunnyVillage,PubAliceRainyTown”)]PubAliceSunnyVillage,PubAliceRainyTown

Concerning the liseng, the provided hint allows the requesting entity’s trust engine to immediately search for evidence that can be linked to *PubVC1* and find the link with *PubAliceRainyTown*.

6.6 Vision-based Entity Recognition (VER) Experiment

In order to evaluate the generic aspect of the entification framework, another domain of application is studied, namely the smart home. In order to increase the level of auto-configuration in smart homes, a trust engine is integrated into the smart appliance access control mechanism to manage interaction between previously unknown users or smart appliances [155]. Rather than having to set up an access control list for individual entities – other users or devices – or groups of entities, the owner of a device only has to set up the level of trust required before interaction with an encountered entity can take place. The trust engine takes care of trust management – evolution of the trust value based on new observations, recommendations and reputations – on behalf of the user. In doing so, the access control at the entity level, which may be overwhelming for home users if done manually, is implicitly managed.

There is still the issue of dynamic enrolment of strangers (and initially the tenants of the home) without too much human intervention. The solution may come from a “concierge” process aware of what happens in the space [164], which can recognise strangers, acquaintances, friends or foes. For this purpose, the ER process is based on images captured with low-cost but widespread webcams and easy-to-deploy image processing techniques (in order to minimise configuration tasks).

Vision is an obvious mechanism for the recognition of people in spaces. It has been used for authentication based on visual biometrics (such as fingerprint, face or gait recognition [85, 167]). Generally, these techniques are used in controlled environments, where enrolment is mandatory (i.e., persons to be enrolled have their visual biometrics entered into the security system in advance). In the home, enrolment cannot always require human intervention, e.g., from a system administrator. A smart space is not an improvement if it makes busy householders even busier. In public environments, there is no list of known people to be enrolled. People roam from one space to another as they wish. The VER scheme [164] addresses the requirement for smooth dynamic enrolment, i.e., the door should not be closed to strangers, but instead any stranger presenting themselves might become an acquaintance. From an evaluation point of view, the VER scheme provides empirical technical trust of different vision techniques. In addition, it investigates ER forget-related and scalability issues, especially how to improve indexing and retrieval of previously recorded imagery based on its context (e.g., time and weekday) in addition to its content. Please refer to the next chapter for detailed results.

In our current prototype, we assume a room with one door (see Figure 26) and the following equipment. The low-cost CCD camera with USB interface to a conventional laptop (*Pentium III mobile CPU 866MHz with 256MB RAM*) is used in a mode which provides 320×240 pixel *8 bit* colour imagery at *15Hz* for each channel. Actual resolution and sensitivity are lower due to a colour filter over the CCD and the poor-quality analogue-to-digital converter used for quantisation. The camera's focal length is *30mm*. Its lens faces the door.

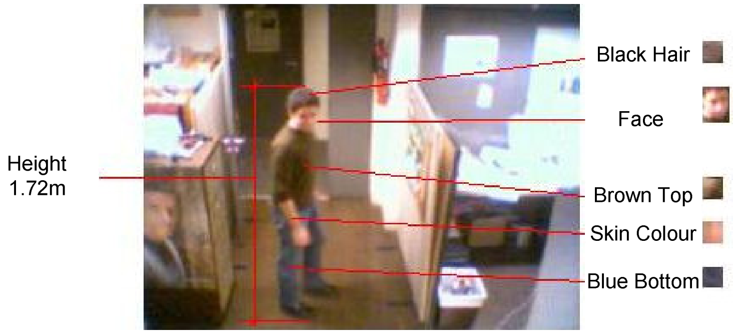
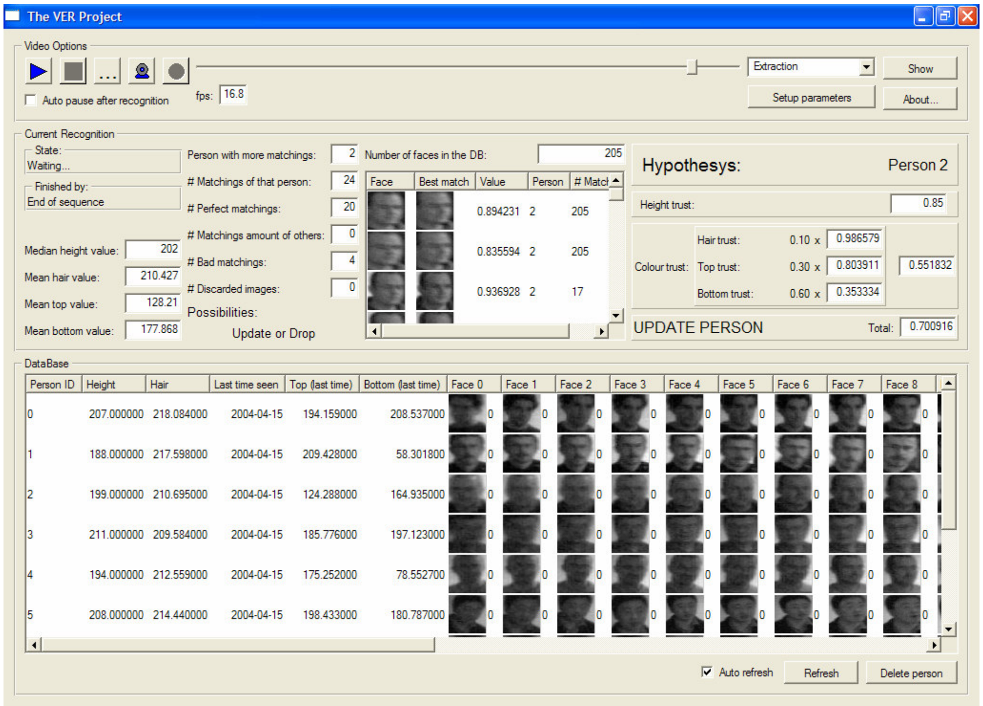


Figure 26. VER Environment and Segmented Features

The software is written in C++ and uses a MySQL database. The GUI is presented in Figure 27. Few configuration tasks are required in the GUI before being able to run the VER scheme to recognise people passing in front of the door. The bottom of the GUI lists the supposed different persons that have been recognised: one person per row; the final columns of each row contain the different face snapshots and how many times they have been matched so far. The top of the GUI provides information that was used to test the software during the development as well as buttons to set up the video and camera configuration.



□ Auto pause after recognition

fps: 16.8

Extraction

State: Waiting...

Finished by: End of sequence

Median height value: 202

Mean hair value: 210.427

Mean top value: 128.21

Mean bottom value: 177.868

Person with more matchings: 2

Matchings of that person: 24

Perfect matchings: 20

Matchings amount of others: 0

Bad matchings: 4

Discarded images: 0

Possibilities:

Update or Drop

Number of faces in the DB: 205

Face	Best match	Value	Person	# Match
		0.894231	2	205
		0.835594	2	205
		0.936928	2	17

Hypothesys: Person 2

Height trust: 0.85

Colour trust:

Hair trust: 0.10 x 0.986579

Top trust: 0.30 x 0.803911

Bottom trust: 0.60 x 0.353334

0.551832

UPDATE PERSON

Total: 0.700916

Person ID	Height	Hair	Last time seen	Top (last time)	Bottom (last time)	Face 0	Face 1	Face 2	Face 3	Face 4	Face 5	Face 6	Face 7	Face 8
0	207.000000	218.084000	2004-04-15	194.159000	208.537000	0	0	0	0	0	0	0	0	0
1	188.000000	217.598000	2004-04-15	209.428000	58.301800	0	0	0	0	0	0	0	0	0
2	199.000000	210.695000	2004-04-15	124.288000	164.935000	0	0	0	0	0	0	0	0	0
3	211.000000	209.584000	2004-04-15	185.776000	197.123000	0	0	0	0	0	0	0	0	0
4	194.000000	212.559000	2004-04-15	175.252000	78.552700	0	0	0	0	0	0	0	0	0
5	208.000000	214.440000	2004-04-15	198.433000	180.787000	0	0	0	0	0	0	0	0	0

☑ Auto refresh

Figure 27. VER Software GUI

We combine different image processing and retrieval techniques to recognise people entering and leaving the room. The ER process allows recognition of previously observed/encountered entities based on visual recognition clues, that is, imagery. There is a PRM where different vision schemes can be implemented (for example, face matching or clothes colour). Each time someone moves in front of the camera, the ER process (depicted in Figure 28) is triggered: we call this self-triggering because the system itself takes the initiative to start the recognition process in order to recognise potential surrounding entities. In step 2 of the ER process, the Detective Work consists of carrying out a variety of visual analyses to obtain a level of confidence of each recognition. Retrieval of previous imagery is based on content as well as context. Step 3 is closely related to step 2 because Discriminative Retention of recognition must be based on previously stored imagery. A difficult question is to define when the person who enters the room is new and converge to the real number of different persons monitored so far. In the ER process, there is no initial mandatory enrolment but enrolment is moved down in the process and occurs at step 3 when recognition information on a new entity is stored for the first time and for later recall. A person is digitally represented by a virtual identity (*vi*). The indexing of stored imagery for future retrieval at the end of step 3 also makes use of context. Step 4 of the ER process concerns further actions to be taken according to what person is recognised. The trust value in the recognised person is not covered in this implementation. However, context is used to tune the level of Detective Work. For example, if a new person is recognised at *2am*, the concierge should increase its level of suspicion (and maybe send a warning message to the security guards) as well as increase the level of Detective Work and Discriminative Retention (which may augment the chance to later recognise the potential thief).

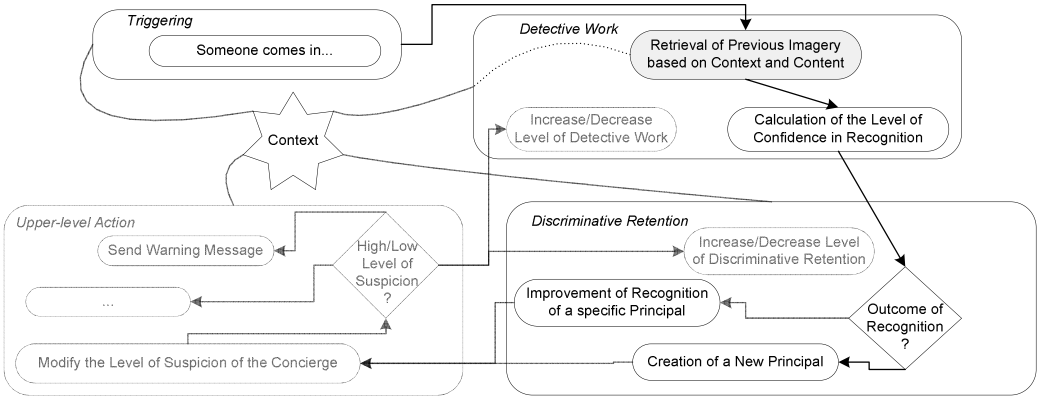

Triggering
Someone comes in...
Detective Work
Retrieval of Previous Imagery based on Context and Content
Calculation of the Level of Confidence in Recognition
Increase/Decrease Level of Detective Work
Context
Upper-level Action
Send Warning Message
High/Low Level of Suspicion ?
...
Modify the Level of Suspicion of the Concierge
Discriminative Retention
Increase/Decrease Level of Discriminative Retention
Improvement of Recognition of a specific Principal
Outcome of Recognition ?
Creation of a New Principal

Figure 28. VER Process Diagram[25]

[25] The actions done during each ER step are contained in the four rounded rectangles. The directed black arrows represent the flow between the actions started after each ER triggering. The Upper-level Action step is slightly transparent because it is not the focus of this implementation part, which is not dedicated to evaluate how trust in entities is used but ER technical trust and ER process scalability and adaptability to context. The use of the element of context is represented by a shape called Context and its connections to other shapes.

Due to the important requirement that the system needs as little as possible set-up or calibration by the owner of the space, the techniques used for image segmentation and analysis are necessarily simple. Additionally, the near-real time performance requirements of the system preclude the analysis of complex biometric characteristics such as gait, but we have designed our indexing and retrieval scheme to allow the inclusion of such characteristics should sufficient computational power exist.

Firstly, feature segmentation is done. Simple inter-frame image subtraction allows motion to be identified. If the motion blob area exceeds a certain threshold then it is considered a potential person. The region-merging via boundary-melting algorithm [169] is applied to segment the blob into distinct regions of significance for recognition.

The significant regions, as shown in Figure 26, are:

1. Skin. Using the approach to skin segmentation suggested by Perez et al. [135], we transform from (R,G,B) colour space into the normalised (RN,GN) model and classify a pixel as skin if its values lie between certain upper and lower thresholds in RN and GN.

2. Face. The uppermost region of skin exceeding a certain area threshold and with appropriate elongation is considered to be the face. Its bounding rectangular region is extracted. If the region is larger or smaller than 40×40 pixels then it is sub- or super-sampled as appropriate to facilitate inter-image comparison.

3. Clothes. Non-skin regions exceeding a certain area are considered clothes. There are typically two such regions found: top and bottom.

4. Hair. In theory it should be relatively straightforward to segment hair, using its colour as another feature to facilitate recognition. However in our environment was insufficient contrast between the hair and the background for it to be segmented reliably.

5. Height. Relative height can be approximated as the difference between the highest and lowest segmented pixel. Any height comparison must take into account the position of the feet.

Secondly, feature analysis is carried out. The face is the only feature that can be used for recognition with any reasonably high degree of confidence (as confirmed in the evaluation in Chapter 7). Simple template-matching (normalised cross-correlation) is used to match segmented faces.

Due to the real-time requirement of the application domain, the storage of virtual identities and their recognition clues had to be carefully implemented. It gave an opportunity to evaluate issues related to scalability and forget aspects of the ER process. The chosen approach consists of context-based image retrieval. Each time a face is segmented from the real-time video sequence, it is appended to a list. When the sequence is finished, each face of the list is compared to the set of different segmented faces stored previously. If there is no match above a minimal level of confidence, or no faces have been stored previously, it is added to the list of observed faces. Details of the other segmented features (for example, clothes colour) are associated with the face, as are temporal attributes such as date, time, and day of week. If a face matches above the minimal level of confidence, then the other details are retrieved and used in the recognition process. In our approach, there is no training data or database of known users per se due to the requirement of dynamic enrolment. This differs from related work on real-time vision-based multi-modal recognition [167].

The advantage of pervasive computing environments is that computing entities are context-aware – environmental information that is part of an application's operating environment can be sensed by the application. Castro and Muntz [29] pioneered the use of context for multimedia object retrieval. We apply the concept within our ER process, which enables the concierge to adapt retrieval and recognition based on context and level of suspicion without the help of an administrator. We especially make use of time and date to index and retrieve imagery. Concerning indexing, the first time the VER scheme is started in a new space, the list of faces and associated visual and temporal attributes is empty. As soon as someone comes in front of the camera, a sequence of faces is extracted from the video. Associated with each sequence is a structure storing the other elements of specific context. Our proof-of-concept implementation consists of storing the time and the day of the week. For each sequence, height and colour information is also computed.

A database is used to store the recognition clues extracted from each sequence. These recognition clues are indexed in specific rows and each row consists of a supposed different person. We can then dynamically index the different rows based on context similarity. For example, we can order the rows decreasingly from the row which contains images the most often seen on Monday mornings around 8am. For performance reasons, each sequence of images is processed for face template matching after the end of the sequence when nobody is moving in front of the camera. The face template matching process is too expensive to be run in parallel during the capture of the sequence.

There are four parameters used in our algorithm: `TimeAndDayOfWeek`; `PerfectFaceRecognition` (that is, the percentage threshold above which the recognition match is considered perfect: empirically from the reading of several sequence processing samples say *92%*), `UnknownFaceRecognition` (that is, the percentage threshold below which the recognition match is considered either a new person or a very different face profile of a known person: again empirically say *85%*) and `BogusFaceDiscarded` (that is, the percentage threshold below which the recognition match indicates that the image does not correspond to a face and is discarded: we empirically chose *30%*). Once all the images of the sequence are compared, we obtain a probability distribution of the virtual identities of the following form:

$$\begin{aligned}&(N_{PFR1}+N_{FR1})\times vi_1+\ldots+(N_{PFRi}+N_{FRi})\times vi_i+\ldots+(N_{PFRn}+N_{FRn})\times vi_n+\\&N_{unknown}\times vi_{unknown}+N_{discarded}\times vi_{discarded}\end{aligned}$$

where vi_i is the supposed different person i among n previously seen persons, N_{PFRi} is the number of perfect face recognitions (match above `PerfectFaceRecognition`) of person i, N_{FRi} is the number of face recognitions (match below `PerfectFaceRecognition` but above `UnknownFaceRecognition`) for the person i, $N_{unknown}$ is the number of faces either of a new person or a very different face profile of a known person (match below `UnknownFaceRecognition` but above `BogusFaceDiscarded`) and $N_{discarded}$ is the number of images considered to be of bad quality (below `BogusFaceDiscarded`).

From this distribution, a choice has to be made. Is it a new person or should it update the recognition clues of a previously known person? The update only consists of faces that are considered different enough to previous images (that is, between `PerfectFaceRecognition` and `BogusFaceDiscarded`) in order to improve scalability. In cases where face recognition confidence is borderline, we use the other visual attributes to help in the decision-making process. We have followed an empirical solution, which has given encouraging results in real settings. However, we have chosen to discard sequences which might pollute the database with poor quality face images. The following simplified pseudo-code presents the algorithm.

```
Pick the person i with the greatest ($N_{PFRi}$+$N_{FRi}$)
if($N_{PFRi}$>(10%×TotalOfNotDiscardedImages)) UpdateFacesOfPerson$_i$
if(NoPerfectMatch){
if($N_{FRi}$>50%×$N_{unknown}$)
      if (((HeightMatching×50%)+(ColourMatching×50%)) >= 50%)
         UpdateFacesOfPerson$_i$
if($N_{unknown}$>50%×$N_{FRi}$)
      if (((HeightMatching×50%)+(ColourMatching×50%)) < 50%)
            CreateNewPerson}
```

Concerning retrieval, thanks to our indexing, we can prioritise the retrieval based on context (for example, time and day of the week). In order to benefit from a probabilistic approach and the fact that the images of a same sequence correspond to the same person (who is entering the room), at most *30* faces are extracted from the sequence and compared to all previous faces stored in the database. In order to speed up the process, the comparison is stopped if `PerfectFaceRecognition` is reached and then the images stored in the database are reordered. The reordering consists of presenting the images of the previously recognised person first, ordered by their number of previous matches. This retrieval approach is evaluated in the next chapter.

6.7 Summary

A test bed is needed to carry out empirical evaluation of the entification framework since no global computing is readily available. In fact, different experiments in different application domains are needed to test all the ASUP requirements expected in global computing.

The email domain is useful to test security without the availability of centralised trusted-third-parties where any newcomer may be a legitimate sender. In order to be able to test different schemes, a tool kit to rapidly implement message-based recognition has been developed. Indeed, new pure message-based ER schemes have been created and tailored to email anti-spam.

Then, a context-aware privacy/trust trade experiment, built on top of this message-based kit, is introduced to evaluate the privacy aspects of the entification framework.

The last experiment is carried out in the smart home to go beyond software simulation to evaluate technical trust, adaptability to context and scalability to a large number of recognition clues.

CHAPTER 7: ENTIFICATION/ASUP EVALUATION

First, the goals of the evaluation based on the experiments detailed in the previous chapter and expected for each ASUP requirements are presented as well as the methodologies and hypotheses. Then, the evaluation results are detailed. Finally, the entification framework is compared to the reviewed frameworks based on the same points of comparison identified in the background chapters at the beginning of this thesis.

7.1 Evaluation Goals, Methodology and Hypotheses

The goal of the evaluation is to evaluate how the entification framework addresses the ASUP requirements in global computing. The methodology and hypotheses vary depending on the application domain and the ASUP requirement under evaluation. However, the approximated test beds developed in the previous chapter must be used because there is no such global computing environment. Although usability is part of the ASUP requirements, further Human Computer Interaction (HCI) evaluation based on the test beds and large-scale user trials is beyond the scope of this thesis.

Concerning the adaptability requirement, the goals are:

- the scalability of the use of recognition clues: the methodology consists of empirical evaluation in a scenario where the scale of recognition clues is great; the first hypothesis is that the scale of recognition clues in the VER experiment is challenging enough; the second hypothesis is that the context-awareness in the application can be used to effectively manage the recognition clues;
- the generic aspect of the ER process: the methodology consists of verification via realistic prototype validation in different application domains; the successful instantiations and experiments of the ER process in both message-based and vision-based recognition validate this goal.

Concerning the security requirement, the goals are:

- the demonstration of static and dynamic technical trust: the methodology is to provide estimates of technical trust based both on theory and on the experiments using the test bed;
- the provision of ER performance for security cost/benefit analysis: the methodology is to provide real performance results in the real experiments of the test bed and to carry out security protection cost/benefit analysis; the operational analysis is conducted on the basis of time and space complexity and verified via experimentation and performance results; this goal is especially important for the SECURE project because such results are needed to feed the risk analysis component engineered by one of the partners of the project;
- the viability of pure ER schemes via threat analysis: the methodology is to provide message-based ER schemes of deployable quality in real email settings based on the threat analysis of the CTK/SECURE email anti-spam experiment; threat analysis is common to evaluate security mechanisms; it is common practice to evaluate reliable identity-based anti-spam techniques from an economic and risk analysis point of view [185]; the VER experiment cannot make a strong case because the prototype is rather rough and could not be deployed at this stage;
- the mitigation of identity multiplicity attacks: the methodology is to carry out experiments with random and engineered attacks based on the knowledge of the network topology of the peers and compare their impact; the hypothesis is that engineered attacks are more harmful; the second hypothesis is that trust transfer can be applied to mitigate these attacks; it is also discussed how safe fusionym is achieved in Section 5.5.2; trust transfer against identity multiplicity attacks is evaluated below in the email domain.

Concerning the usability requirement, the goals are:

- the possibility of dynamic and automated enrolment: in fact, the security goal of viability of pure ER schemes validates this goal as well because no step involving the link between the real-world identity and the virtual identity is mandatory and no mandatory link simply facilitates the process;

- the selection of the most appropriate virtual identity based on context: the methodology is once again via real prototype validation; this goal has been reached by the instantiation of the functionality in the CTK and used in the context-aware privacy/trust trade experiment.

Finally, the privacy goals for the evaluation are:

- the argumentation behind the inherent conflict between privacy and trust: the methodology consists of the discussion of the relation between privacy and trust; this discussion is done in Section 5.4.1;
- the support of multiple pseudonyms per user: the methodology is again via real prototype validation; the functionality is implemented in the CTK and used in the context-aware privacy/trust trade experiment;
- the support of privacy/trade negotiation: the methodology is based on the discussion of the privacy/trust trade model in Section 5.4 and the context/aware privacy/trust trade experiment detailed in 6.5.

Therefore, the goals cover the four ASUP dimensions.

7.2 Scalability of the Use of Recognition Clues

To evaluate the scalability of the use of recognition clues, the VER experiment (described in the above Section 6.6) is used to compare random-based and context-enhanced retrieval and indexing of virtual identities and their recognition clues.

One of the reasons we privileged our context-based retrieval and indexing rather than retrieval with a random order of images is to obtain a faster retrieval scheme. It is related to the management of virtual identities, their recognition clues according to the context of interest and the forget aspect of the ER scheme. It is worth mentioning than stopping the retrieval and not assessing all stored images for each new image is faster but we lose the opportunity to detect a recognition result greater than `PerfectFaceRecognition`. However, this allows us to compare if context-based retrieval is really faster than random-based retrieval.

For this assessment, videos of *9* different persons (including Europeans, Chinese and Indians) entering the room *4* to *5* times were recorded. The database was populated with the same sequence for each person: these *9* sequences resulted in *205* faces stored in the database. Then, the remaining sequences of each person were processed (although no update/creation was applied) and resulted in the extraction of *757* faces. Using a random approach, this corresponds to *155,185* matches (*757* × *205*). Thanks to our `PerfectFaceRecognition` bound and context reordering (explained in Section 6.6), assuming that each match takes the same time, the process was roughly *1.4* faster than the random-based retrieval (that is, *44,780* fewer matches were needed).

7.3 Demonstration of Static and Dynamic Technical Trust

First, two evaluations of static ER technical trust are presented. Then, a means to dynamically evaluate ER technical trust is given.

7.3.1 AAS Technical Trust

A base secret is hard to guess if it has high entropy: the entropy of a base secret is given by the number *N* of possible values it can have. A number's *bit space* is the number of binary bits in that number and given by the logarithm relative to the binary base 2 [168].

The Average Attack Space (AAS) is based on the number of trial-and-error attempts imposed by an authentication technique on the attacker and corresponds to the number of guesses, on average, the attacker must make to find the base secret [168]. The AAS is represented as a bit space, where *N* is the number of possible base secrets. When all possible base secrets are equally likely to be chosen, the attacker must, on average, try half of the base secrets:

$$AAS = \log_2\left(\frac{N}{2}\right)$$

If the likelihood to be chosen differs between the different base secrets (for example, calendar dates for luggage locks [168]), the bias should be reflected in the calculation of the AAS. Another requirement is that the AAS of a secret that can be attacked offline must be greater than when only online attacks are possible. Once the difference between offline and online attacks is considered, the AAS can be used to compare a broad panel of authentication techniques. For example, it can also be applied to cryptographic techniques (hence, offline attacks are possible): the AAS of *56*-bit DES is *54* bits.

Each recognition scheme will have to be assessed concerning its technical trustworthiness. A static value between *0* and *1* for each recognition scheme based on the AAS may be used. For example, the company RSA Security still recommends at time of writing to use public keys of at least *1024* bits for corporate use and "*2048* bits for extremely valuable keys like the root key pair used by a certifying authority"[26]. *2048*-bit public keys correspond to an AAS of *116* bits [168] that can be attacked off-line: this is our reference for a trust value close to 1. Imprinting with strong key (that is, *128*-bit AES, which gives an AAS of *127* bits [168]) in the Resurrecting Duckling scheme [172] would get a technical trust value near to *1* because it respects our criteria when off-line attacks are possible (*127* bits > *116* bits reference). A well-designed biometrics system is not vulnerable to offline attacks, so biometrics systems can rely on a smaller AAS than for systems where on-line attacks are possible and imply greater AAS. Well-designed biometrics, those that can only be attacked interactively, are considered strong when the false acceptance rate (FAR) is around *1* in *1,000,000* [168]. So, we consider that schemes respecting at least the latter criterion would get a technical trust value near to *1*. Biometrics with higher FAR would get a trust value in proportion with the criteria for strong biometrics (e.g. a FAR of *1/100,000* would get *0.1*). With higher FAR, enrolment can be achieved more dynamically because the learning phase is simpler.

7.3.2 Empirical Assessment of VER Technical Trust

Each recognition scheme has to be assessed concerning its technical trustworthiness. The number of people we used for this assessment (that is, *9*) is in line with the assessment done in previous related work [167] (that is, *12*). Practically, for each different vision recognition technique (face, height and colour), we populated the database used in the experiment detailed in Section 6.6 with *9* persons and then for each remaining sequence (*4* different sequences for each person: *36* sequences in total), we counted how many times each scheme makes the right decision (that is, if the sequence corresponds to person *i*, the scheme should recognise person *i*). We obtained a technical trust of: *0.94* for face template matching (*34/36*), *0.39* for height matching (*14/36*) and *0.53* for colour matching (*19/36*).

[26] http://www.rsasecurity.com/rsalabs/node.asp?id=2218.

7.3.3 Dynamic ER Technical Trust

Our end-to-end trust model [157] says that the decision-making process should use a function of trust in the entity and trust in the underlying technical infrastructure (which is the level of confidence in recognition, called *lcr*). Instead of the static choice of a level of trust for a specific message-based ER scheme (as presented above), the ER scheme can be considered as an entity and a trust value is explicitly computed based on direct observations and recommendations. In this section, another experiment is used. Technical trust can be dynamically computed in the CTK/SECURE email anti-spam experiment. We introduce a *lcr* with two components: one that is global to all senders and one that is related to the current sender. We use the SECURE trust value format with the following event structure. The event *0* is triggered each time the ER scheme is used, that is, if Bob or another sender sends an email, the counter is incremented. The event *U0* is triggered each time the ER scheme is used for a specific sender, for example Bob. There is an event called *cor*, which records that the ER outcome has been confirmed correct by a human. The confirmation can be implicit, for example, no complaint is made about a received and delivered message. Another event, called *spo*, records that this sender has been spoofed with this type of ER scheme. We illustrate this by way of the example in Figure 29 – this ER scheme has been used *175* times so far and *100* recognitions were confirmed right by Alice (the receiver), *25* messages have not been read by Alice yet and *50* spoofings occurred due to this scheme. Then for the current sender, called Bob, Bob has been recognised *28* times by this ER scheme. Bob sent *25* emails that were correct, two messages that are still unread and Bob has been spoofed one time with this ER scheme.

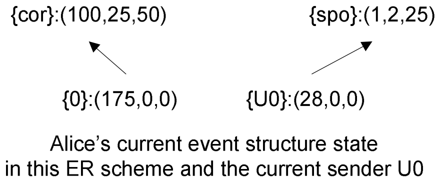

{cor}:(100,25,50)
{spo}:(1,2,25)
{0}:(175,0,0)
{U0}:(28,0,0)
Alice's current event structure state
in this ER scheme and the current sender U0

Figure 29. lcr Event Structure Example[27]

[27] A → B means that event A is necessary for event B. We use the following representation: *{eventname}:(s,i,c)*, which corresponds to a standard SECURE triple.

The final *lcr* consists of:

$$lcr = \frac{s_{cor}}{s_{cor} + i_{cor} + c_{cor}} \times \frac{c_{spo}}{s_{spo} + i_{spo} + c_{spo}}$$

The use of i_{cor} should become negligible over time compared to *s* and *c* because messages do not stay a very long time without human inspection – a user is likely to check his/her email each week. However, this value can also be useful to detect DoS attacks. Ideally, *cor* values should be global, which means that users should make their results regarding an ER scheme's strength publicly available. This can be done by recommendations distributed via the trust engines. In doing so, we obtain a dynamic evaluation of the technical trust of the ER schemes used. The trust engine manages at least two trust contexts: the trust value of the ER scheme and the trust value of the interacting virtual identities.

7.4 ER Performance for Security Cost/Benefit Analysis

When it comes to performance evaluation, the real setting and the application domain play a central role. This section gives two examples of performance results. The first results concern the overhead of the security of JXTA communication channels that can be registered in the CTK. The second results have to do with the CTK/SECURE email anti-spam experiment.

7.4.1 Ad-hoc Peer-to-Peer Performance/Security Trade-off Evaluation

Performance plays a key role for resource-constrained peers, which use the CTK, (as Recognisers or as `Claimants`) when they attempt to tune their `AttentionLevel`, `DetectiveWorkLevel`, `DiscriminativeRetentionLevel` and `OutCluesLevel`. The SECURE trust engine requires such information to be fed into the risk component to be able to make meaningful decisions.

Motivation By Example

We take the example of peers that can only communicate for short periods of time called "contacts", (e.g., in Delay Tolerant Networks [51]) with few of these periods, which become real communication opportunities. Further, the peers that can be encountered during these contacts are not all known and their work consists of sending `Claims` to specific peers. Some of the peers present at time of contact are malicious and try to spoof other peers. There are two communication channels for peers sending `Claims` to specific peers: a weakly secure communication channel and a more secure communication channel. The more secure communication channel takes *50%* more time to send a `Claim` than the other one. The contact time allows each peer to send *100* Claims over the weakly secure communication channel and *50* `Claims` over the more secure communication channel. The peer's mission is to optimise the number of successfully sent `Claims` to specific peers during each contact. If the peer knew that the success rate with the weakly secure communication channel is *20%* and *100%* with the more secure communication channel, the Management Unit should set the `OutCluesLevel` that commands the CTK to use the more secure communication channel.

Assessment of JXTA-Java pipes

Our previous example requires that a peer can use different communication channels with different security strengths and easily switch from one communication channel to another one. This is easily achievable for a JXTA peer [94] thanks to one of the main abstractions in JXTA, which is the concept of *pipe*. A pipe describes a connection between a sending endpoint encapsulation of the native network interfaces provided by a peer – and one or more receiving endpoints. A pipe – a kind of virtual communication channel – is used to conveniently connect peers and to send messages between them because a network transport can be accessed without interacting directly with the endpoint abstraction. Any transport capable of unidirectional asynchronous unreliable communication can be used. JXTA implements TLS [178] to secure the communication through pipes. The following keywords are used to easily switch between the different kinds of pipes: *JxtaUnicast* (that is, unicast, unreliable and insecure pipe); *JxtaUnicastSecure* (that is, unicast and secure pipe); *JxtaPropagate* (that is, propagated, unreliable and insecure pipe). Thus, in our previous example, if a peer wants to send a `Claim` to another peer, two pipes must be registered as `ClaimSendables` in the CTK: one pipe specified with the JxtaUnicast keyword and a second one with the *JxtaUnicastSecure* keyword. From this point, the Management Unit can specify what communication channel to use because knowing that TLS is used increases the `OutCluesLevel` and `InCluesLevel`.

However, we had to define how to obtain performance results about each communication channel (for example, the type of pipe). Such results are needed for the Management Unit to choose the best strategy. Our testing method [153] is based on the "performance assessment framework for distributed object architectures" [93]. Nevertheless, some changes have been made to get results more appropriate for the JXTA pipe paradigm. The JXTA model is not a distributed object model as such. Our testing method may be used to assess the performance of pipes connecting two peers separated by a context-dependent number of peers with context-dependent types of transport between them. We present results for one specific configuration called *TCP only*, which consists of a direct physical connection between two peers using TCP.

Among the different criteria for quantitative evaluation of performance, three criteria are relevant for our case: Round Trip Time (RTT), throughput, and data throughput. The definition of each criterion had to be adapted, as follows:

- RTT – measures the time needed from the point when the `Claimant` peer sends a message (that is, a `Claim`) to the target peer to the point when the `Claimant` receives the message back. Any other processing needed for sending and receiving the message on both peers is minimised as much as possible.
- Throughput – measures the number of message round trips in a given time interval. Throughput and RTT are inversely proportional. This is the reason that only RTT results are presented.
- Data throughput – measures the efficiency of data transfer. The data are transferred as elements in the messages sent and returned. Our results are for data as strings of text.

The same Java source code was used for the different tests. The local results are for a *PentiumIV 1.7 GHz with 256 MB RAM* and the *LAN* results are for this computer and a *PentiumII 450 MHz with 128 MB RAM* connected to a *10 Mb Ethernet hub*. Time was measured with the *System.currentTimeMilli()* method. The precision of this method is *1 ms* under Linux and *10 ms* under Windows. Different issues arose when we assessed the accuracy of our results. The most important issue concerned the calculation of the necessary number of observations. Another issue involved the analysis of the steady state. We had to define how many observations were needed to get an appropriate level of confidence.

There are two requirements on each occurrence of what is observed, X_i, to allow reliable data analysis [97]:

1. Each X_i should be normally distributed. To satisfy this first requirement, we carried out experiments to check that the arithmetic mean of a sufficiently large batch of individual observations is close to normal. This is the reason that we measure the time of one thousand round trips.
2. All X_i should be independent. We also carried out experiments to study the issues related to the steady state. Ideally, any data obtained from the transient period must be discarded, but practically this is not easy to detect [97]. Nevertheless, in the case of JXTA pipes, it appears that the first one thousand round trips of messages are processed slightly more slowly than the rest of the round trips. The standard deviation is also higher, showing that it is not a steady state. The slow start may be due to the initial creation of objects needed for processing the requests. However, we found that this is not really significant compared to the total number of round trips and thanks to the fact that we ran three time the same test in a row.

Since the two requirements are fulfilled according to our experiments, assuming that $\underline{X}$ is the mean, s is the standard deviation and z is the inverse of the standard normal cumulative distribution with a probability of *(1-a)*, we can now calculate n, the number of observations necessary to obtain an accuracy of $r = \pm 1\%$ on loops of one thousand round trips of messages with a confidence level of $100(1- a) = 99\%$ for the whole range of string sizes. We ran *400* repetitions of one thousand round trips of message in order to be able to apply the following formula to find the necessary number of observations:

$$n = \left(\frac{100 \times z \times s}{r\underline{X}} \right)^2$$

Hence, we validated that fifty repetitions of one thousand round trips is enough to get this level of confidence in our results for the whole range of string sizes. In all cases (see Figure 30), the RTT looks linearly dependent on the string size. It can be approximated with a linear function in the form:

$$RTT(StringSize) = k \times StringSize + Offset$$

In Table 5, there is an estimation of the RTT function for each configuration calculated by linear regression analysis by using the least squares method to fit a line through the set of observations.

TCP Only	offset (ms)	k (ms/character)
Local, Windows, JXTA build 49b	7.6718	0.0003
Local, Linux, JXTA build 49b	8.1387	0.0003
LAN, Windows, JXTA build 49b	12.3042	0.0021
LAN, Windows, JXTA build 65e	17.7732	0.0020
LAN, Windows, JXTA build 49b, Secure	30.1358	0.0092
LAN, Windows, JXTA build 65e, Secure	39.8799	0.0095

Table 5. RTT(Stringsize) Linear Functions

Figure 30 also shows that to use secure communication via JXTA secure unicast pipes implies an overhead as expected. Of course, secure communication is more useful under a LAN configuration. Roughly speaking, the security overhead grows from *125%* for a string of one character to *300%* for a string of *30,000* characters.

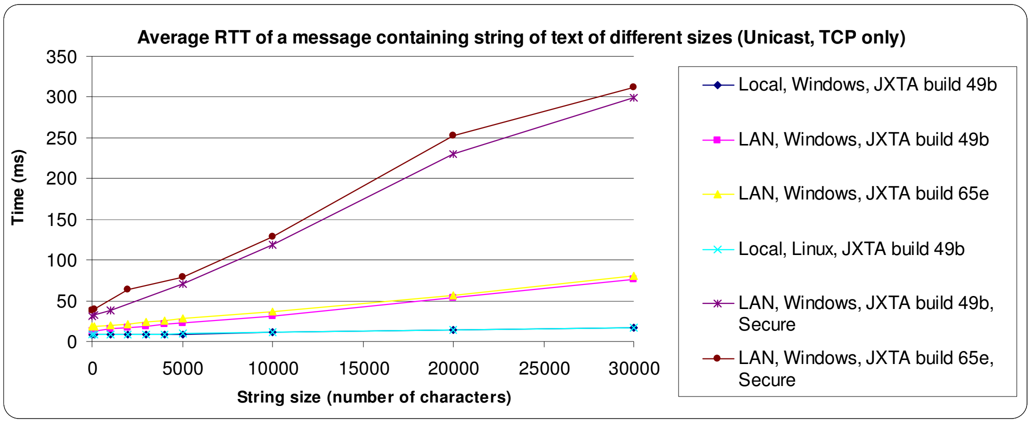


Figure 30. Data Throughput

After all this performance assessment work, we obtain that *k* for secure communication is approximately *4.6* times bigger than without security. For example, this information is crucial for the risk analysis made by the SECURE trust engine.

7.4.2 CTK Performance and Cost/Benefit in Email Anti-spam

As is presented below, in addition to a greater computation cost for CTK anti-spoofing when the ER technique also uses signature validation based on asymmetric cryptography, the CTK protection introduces overhead due to collaboration. However, this computation and communication overhead can be considered affordable since the main cost seems to come from human distraction due to spam. The following threat analysis discusses this economic aspect further.

Since our techniques involve sending additional emails to confirm the identity of the sender, we will first of all evaluate the resulting overhead this causes.

In the default combination of the C/R and hashes techniques, there is a C/R for each newcomer followed by local checks of hashes. To make the analysis tractable, we make the following assumptions: every email sent reliably reaches the receiver; only one receiver is specified by each email sent; all users participate (run our system); and no loss of state can happen due to failures. We examine the overhead of proxy-based emails after a period of time with regard to the whole network (it may also be useful for mail server overhead, where all counted email addresses would be from the same email server). At this stage, we do not introduce spammers as they will be considered in the next section on threat analysis.

A case without spammer is the worst case from a protection cost/benefit point of view because the cost of protection is (ultimately) useless. Let us say that: *N* is the number of involved email addresses (all legitimate for now); *UE* is the number of emails sent in the unprotected case; *PE* is the number of emails due to protection; NCF_i is the final number of newcomers seen by a legitimate email address *i*. For each newcomer, the C/R adds two proxy-related emails, even for pre-trusted ones (otherwise it opens a window of time during which a spammer can send the first email before the legitimate sender). If we do not use *friends* (pre-trusted recommenders that are allowed to introduce their trustworthy contacts) for collaboration, we obtain:

$$PE = UE + 2 \times \sum_{i=1}^{N} NCF_i$$

The worst case happens in environments where there is a high percentage of newcomers, for example, if one-time disposable email addresses [161] are common for privacy reasons or for a new online shop. However, there cannot be more newcomers than the number of emails sent without protection.

Therefore, at most, we have:

$$\left(PE = UE + 2 \times \sum_{i=1}^{N} NCF_i \right) \leq 3 \times UE$$

In a closed community, where everybody knows everybody else, *PE* is close to *UE*. Based on a small size survey (of the email accounts of the computer science department of Trinity College Dublin, which consists of more than *750* users), it seems that in personal email settings, the number of newcomers per day is negligible compared to the number of emails processed (say on average one newcomer and *50* emails exchanged per day per user: *PE=50N+2N*; an overhead in traffic of only *4%*). Therefore, the introduction of our hashes technique is very useful to considerably reduce the overhead in most personal settings, which otherwise reaches *200%* of the load without protection if C/Rs are done for each email.

It is worth considering a scenario with collaboration with some friends' email addresses. Let us consider that each user has a total number of friends, who are sequentially polled in case of a newcomer *j* in order to see whether it is a trustworthy email address or not. However, a polled friend only checks his/her local trust value and does not contact his/her friends in case the local trust value is *(0,0,0)*. As soon as a friend says it has already encountered it, the remaining pollings are not processed and the number of real pollings is recorded as FRC_j. We have:

$$PE = UE + 2 \times \sum_{i=1}^{N} \left(NCF_i + \sum_{j=1}^{NCF_i} FRC_j \right)$$

The best case is when only one friend is polled for any newcomer. Let us say that FRC_j is constant. As previously, the worst case is when there are only newcomers: $NCF_i=1$ and *N=UE*:

$$PE \leq \left(UE \times (3 + FRC) \right)$$

Therefore, from a network traffic view, as soon as the collaboration requires polling more than three friends, the traffic of the worst case scenario without collaboration doubles. The search overhead of potential friends knowing newcomers is further studied in Section 7.6.2, where real networks of email users, identity multiplicity attacks on them and tailored trust transfer are evaluated.

Once we approximately know the number of additional emails to be processed, it is interesting to evaluate the increase in terms of memory space and computation time. We have not considered the number of hashes so far. From a memory point of view, experiments on a corpus of *1,000* emails showed that the serialised Java MIME email of a message of *1,000* characters takes on average *2,000* bytes. The serialised CTK version of this email with signature (which is the worst case overhead; Java-based RSA asymmetric encryption with *2,048* bits) but without hashes is *11,025* bytes. A serialised CTK hash object takes only *8* bytes. Therefore, we assume that the adjunct of a few hashes is negligible (for example, *10* hashes should be sufficient). The overhead of CTK claims (especially signed ones) may be significant, especially when it is combined with the overhead of proxy-related only emails. However, means may be found to optimise the CTK claims serialisation. From a computation point of view, an external provider's proxy-based service server should carefully study the computation power needed, especially at the opening of the service. In fact, due to the overhead in number of messages due to newcomers, who will be plenty at the beginning, and the non-negligible computing time required with public keys of a secure number of bits (please refer to Table 6, which gives the average time of signing based on batches of *1,000* claim signing tasks, done on a *Pentium 4 1.7GHz*, for messages of *10,000* characters, four SHA-1 hashes and Java-based RSA asymmetric encryption), the computation power needed might be challenging for a single server.

Key length (bits)	Mean time to sign 1 Claim (ms)
512	7
1,024	37
2,048	234

Table 6. CTK Claim Signing Computation Time

7.5 Viability of Pure ER Schemes via Threat Analysis

The evaluation of this goal is application dependent: indeed the threat analysis (peer-reviewed [156]) corresponds to the CTK/SECURE email anti-spam experiment.

The root cause of spam is ultimately the same property of email that makes it so attractive and useful: the low cost of open communication with a large number of people all over of the world. Moreover, the near-zero cost of creating and spoofing an email identity ensures that even when the sending of unsolicited bulk messages is prohibited by law or ISP policy, tracing and punishing the offender is not easy because the underpinnings of current email systems were not designed with authorisation and secure authentication in mind. Proposed solutions that attempt to remedy this oversight have been dismissed as infeasible in the short term as transitioning all of the world's email users to a new system is a monumental task [98, 161].

7.5.1 Adaptability, Security, Usability and Privacy Discussion

The points of comparison used to compare the frameworks related to this thesis underline that there are convoluted issues surroundings adaptability, security, usability and privacy. Users will not adopt the CTK/TSF proxy protection if there is not a favourable cost-benefit ratio to spending resources for its use compared to its success against spam.

The goal of our techniques is to prevent spoofing attacks on a sufficient large-scale (that is, a large number of plain-text email addresses owned by non-spammer users) for spamming to be profitable, without compromising the usability present in the legacy email system for user acceptance of our solution. A solution requiring the binding of a key with a real-world identity is too inconvenient. Hence, our solution keeps chosen user-friendly text email addresses due to two reasons: they are viable to be easily remembered and exchanged (for example, by voice); and they are part of the legacy email system. A first advantage of the CTK is that it increases the level of authentication to legacy plain-text email addresses without too much inconvenience. Our approach does not require that all users switch to our system at the same time. We have already explained that they are not bothered by annoying automated emails and that non-participating users may see only a small meaningless attachment in the first email sent by the user.

An important aspect for the convenience of our solution is the ability to process feedback from the user to improve its future decision making. Explicit feedback (for example, by the mandatory input of a quality percentage before closing the email reading window) might be considering too costly. It is said that the sacrifice of usability for more security may sacrifice both. Hence, our solution uses an implicit (although less fine-grained than with a percentage) feedback from the receivers, which is detected as they move emails between folders. All is transparent for the users because IMAP and SMTP proxies are used between the email client and the real mail server and this means our solution works with any IMAP/SMTP-compliant email client. At any time, the receiver can pre-trust a new email address (for example, the email address of the new mailing list of interest). Email addresses to be pre-trusted may also be automatically extracted from software (for example, the user's Outlook address book or any email addresses appearing in the to:, cc: and bcc: fields of the emails sent by the user). The CTK/TSF proxies take care of storing hashes of previous emails, signature validation and challenging each other as depicted in Figure 21 and Figure 22 based on common hashes found in the emails and cryptographic C/R.

If the proxy is not run on the user's local machine then there is a risk to their privacy, since this requires copies of all their emails to be kept on the server. However many users do this anyway for the convenience of remote access (for instance, by using the IMAP protocol, or webmail services such as Hotmail), so this is of low concern, although the problem can also be mitigated by storing only the hashes of the emails instead of the full content. The advantage of pointing to a service provider (as depicted in Figure 32) is that the scheme is guaranteed to work 24 hours a day and without maintenance burden. A proxy service may also be useful for resource-constrained mobile devices. Another advantage of an external proxy is that it forces the attacker to go to a more protected zone, if we make the reasonable assumption that an expert administrator takes care of the security of the service. The direct benefit for users of such a CTK/TSF proxy is that their text email address cannot be spoofed (to other participating users) for large-scale spam attacks. They may also prioritise incoming emails from other trustworthy senders since they have proven that they are indeed more trustworthy.

There are still possibilities for DoS attacks. Because our approach makes use of challenge/responses and recommendations, the previous section estimates the overhead of emails sent over SMTP. We envision that it should be feasible to optimise and limit congestion due to the number of extra emails sent. This will depend on the evidence propagation scheme of the trust engine. A specific trust transfer scheme detailed and evaluated in Section 7.6.2 limits malicious overhead due to collaboration by forcing the sender to participating actively in the search activities. On a general note, DoS attacks are an open issue for any networking software.

Our approach, in addition to be more than a simple human-involved C/R scheme, addresses the "techno-economic underpinnings of spam" said to be overlooked in other C/R-based approaches [165]. The next subsection strengthens this economic aspect.

7.5.2 Defeating Profitable Attacks

The primary threat that our model aims to nullify is a spammer who sends a large number of emails with forged legitimate sender email address, thereby defeating simple anti-spam filters and hiding its true source from casual inspection, protecting the spammer from possible retaliatory action or prosecution under their ISPs terms and conditions.

Because our model depends on knowledge of a user's emails, the fact that the vast majority of email is sent over the Internet in the clear leads to the possibility of another attack, one in which a spammer may eavesdrop on a sufficient number of a user's emails to forge the hashes or C/R response. However, while this attack may be feasible on one user's email account, the reason for spam is that despite the very low response rate, the per-message cost is sufficiently small for it to remain profitable. Obtaining access to enough points on the Internet to eavesdrop on a large number of users against whom to use this attack would raise the per-message cost to prohibitive levels. Furthermore, the use of our asymmetric cryptography extension mitigates this type of attack because the emails are signed anyway.

There is currently a trend for spammers to use compromised desktop machines as distribution points. Since these machines have a compromised operating system, we have to assume that the attacker has full access to the user's email store and may make full use of their programs to send email as if they were the user, thereby side-stepping the protection offered by our system. However, because our system allows the recipient to know from which trusted address the spam came, they can easily tell which user's computer has been compromised and inform them or setup a temporary filter until the machine is fixed. For example, the receiver can manually set a temporary trust value for the compromised sender to be *(0,0,1)*. Recommendations can then be used to propagate this information to friends of the receiver to protect them from this sender. As a result, a spammer who compromises one trusted sender's machine is easily detected and shut out of the network before they can send a sufficient volume of spam to make breaking the security of the machine worth their while.

We shall now consider a final class of attacks, the SBAs, in more detail.

7.5.3 Unprofitable Security Breach Attacks (SBA)

Security breach attacks can occur at different places in the email system as described in the following figures.

An attack on a user's local machine, shown in Figure 31, has already been covered in the previous section. The result is likely to be the same even if the CTK/TSF proxy is run externally (Figure 32) as the compromised machine may sniff the login details of the proxy when the legitimate user accesses it.

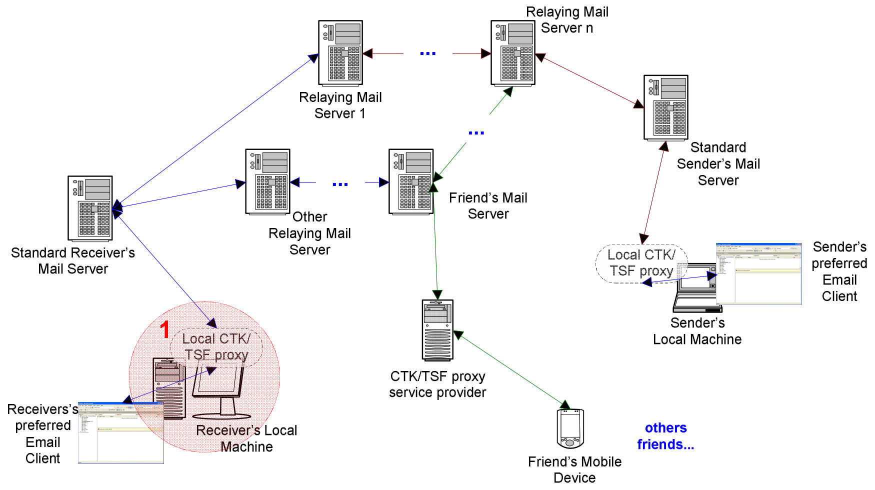


Figure 31. SBA Type 1 – User's local machine is compromised

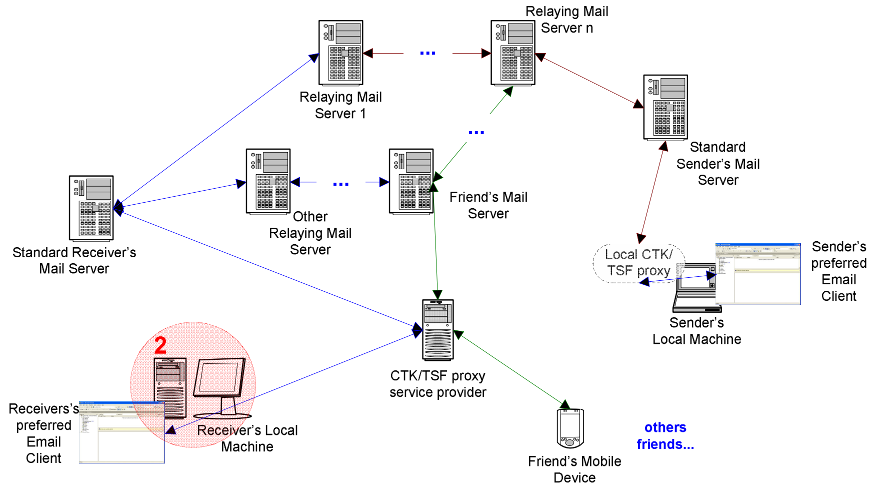

Relaying Mail Server 1
...
Relaying Mail Server n
...
Standard Sender's Mail Server
Other Relaying Mail Server
...
Friend's Mail Server
Standard Receiver's Mail Server
Local CTK/ TSF proxy
Sender's preferred Email Client
Sender's Local Machine
2
CTK/TSF proxy service provider
Receivers's preferred Email Client
Receiver's Local Machine
others friends...
Friend's Mobile Device

Figure 32. SBA Type 2 – User's local machine is compromised but with the proxy hosted externally

An attacker could also compromise the user's mail server, as shown in Figure 33. This would permit them to eavesdrop and intercept all the communications made by the users of that server, and then later use that information to spoof the trustworthy email address recognition information. Should an attack of this type succeed then the ability to impersonate all the users of that server would clearly be very beneficial to the spammer, but equally it should be possible to assume that a professionally administered server is significantly harder to hack than a desktop machine. Therefore, it is expected that the cost of compromising the server would outweigh the benefit gained in the short time before the compromise was detected and shut down. A similar analysis applies whether the proxy is run on the user's desktop or on the server as it is the attacker's ability to eavesdrop on and intercept messages before they reach the proxy that is important here.

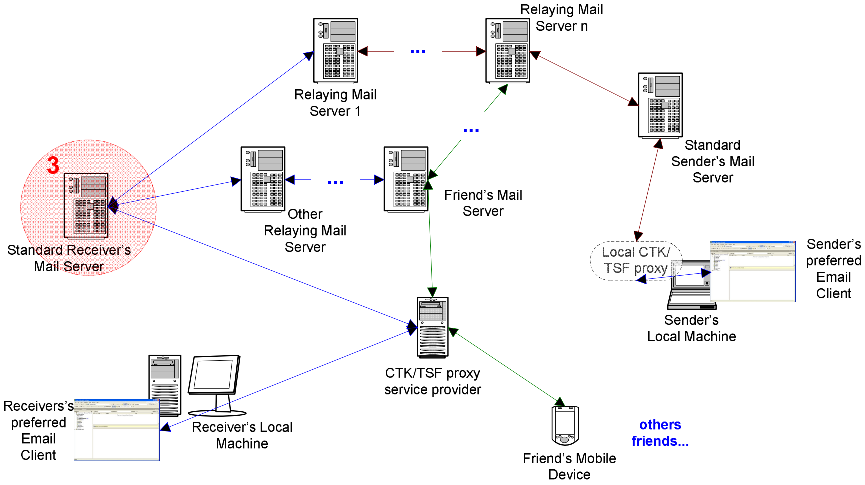

Relaying Mail Server n
Relaying Mail Server 1
...
3
Standard Sender's Mail Server
Friend's Mail Server
Other Relaying Mail Server
Standard Receiver's Mail Server
Local CTK/ TSF proxy
Sender's preferred Email Client
Sender's Local Machine
CTK/TSF proxy service provider
Receivers's preferred Email Client
Receiver's Local Machine
Friend's Mobile Device
others friends...

Figure 33. SBA Type 3 – Mail server compromised

A subtype of the previous attack is where a relaying SMTP server on the path between two users is compromised, as shown in Figure 34. The benefits to the spammer in this case are even fewer than in the previous case as only a subset of the communications can be observed making it much harder to reliably use that information in an attack on trustworthy email addresses.

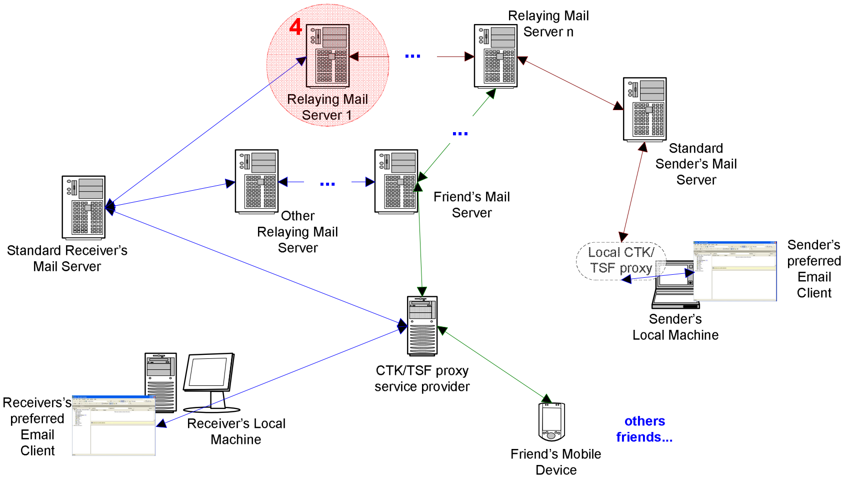

4
Relaying Mail Server 1
...
Relaying Mail Server n
Standard Sender's Mail Server
...
Other Relaying Mail Server
...
Friend's Mail Server
Standard Receiver's Mail Server
Local CTK/ TSF proxy
Sender's preferred Email Client
Sender's Local Machine
CTK/TSF proxy service provider
Receivers's preferred Email Client
Receiver's Local Machine
Friend's Mobile Device
others friends...

Figure 34. SBA Type 4 – A relaying mail server between sender and receiver is compromised

7.6 Mitigation of Identity Multiplicity Attacks

In order to evaluate trust transfer and its usefulness against identity multiplicity attacks, we applied it in the email domain. First, issues surrounding identity multiplicity in the email domain are explained and empirically estimated in a real-world network of email users. Then, trust transfer is tailored to the email domain in order to mitigate these attacks.

7.6.1 Behaviour Under Attack of A Real-Network of Email Users

One of the simplest anti-spam techniques is whitelisting. In this approach any emails from email addresses manually or implicitly whitelisted (e.g., the email addresses of an email address stored in the user's personal folders are whitelisted) are always delivered. In reality this method is not effective for two reasons: firstly, standard text email addresses are so easy to spoof that many spam emails appear to come from a legitimate address and secondly it makes it harder to establish a communications channel with a new contact (or an old contact using a new address). The former is solved thanks to anti-spoofing provided by our CTK. The latter is more difficult but recently the "bankable postage" [3] (BP) technique has been proposed to allow the sender of an email to attach a proof (or means to point to the remote proof in a secure way) that guarantees that a certain cost has been incurred to obtain this proof.

Unfortunately, while this is a technically feasible approach to solving the underlying problem of spam, namely the near zero-cost of sending it, how to set the minimal fee required guaranteeing protection remains an issue. Additionally, using BPs imposes additional burdens on the sender which make it significantly less attractive to ordinary users than traditional email. Finally, it may exclude poor users [62]. The bankable postage system seems a promising defence against the Sybil attack, but it does involve a significant alteration to the way in which email works that may act as a disincentive to newcomers to the system. The trade-off between usability and security is not acceptable. To counter this, we use the trustworthy collaboration features of trust engines to minimise the number of bankable postages a newcomer must pay before they are accepted into the system. We envisage that since email corresponds to a social network (which is in line with Golbeck and Hendler's work [69] reviewed above) the number of degrees of separation between an unknown sender and a specific receiver should be low and thus the propagation of trust should be fast. The ultimate scheme would guarantee that once a trustworthy complete newcomer, whose only means is to pay one bankable postage, sends one email, they should never have to pay another bankable postage provided they continue to behave in a trustworthy manner. A final reflection is that if users retain user-friendly and permanent text email addresses, which is usually the case for obvious usability reasons, most of the trustworthy email addresses are likely to be pre-trusted somewhere in the trust network. If the trust computation performs well, no bankable postage is needed for all of them. So, we assume that situations with complete newcomers are rare and that a bankable postage is only needed in rare situations or where users wish to create disposable or anonymous addresses with no relation to their previous address. The trust engine allows for a broad range of automated decision delivery policies and more importantly an efficient propagation of trustworthy email addresses, which further decreases the use of bankable postages.

There are three basic means for a spammer to fool collaborative anti-spam solutions at the level of identity:

1. spoofing of email addresses of legitimate email users:
    a. to get spam content through as if it was sent by the spoofed email address;
    b. to propagate false recommendations as if they were sent by the spoofed email address;
2. compromising the email accounts of legitimate email users:
    a. to get spam content through by sending it from the real email address;
    b. to propagate false recommendations by sending it from the real email address;
3. creation of virtual email addresses, which corresponds to identity multiplicity attacks:
    a. to propagate a great number of false recommendations from real email addresses;
    b. to be disposed as soon as they have been used for one spam attack.

It may be interesting to compare the cost of applying a combination of these means to the cost of compromising an email account of a user of interest. For example, if the attack is made to present a spam email to a specific ordinary email user, it would be more expensive to compromise a number of his/her contacts than to compromise his/her personal account straightaway. When collaboration and recommenders are used, it may open new means for attacks based on a group of recommenders.

For example, it may be the following fourth type of attack, called the *collaborative deceptive pleasing attack* [158] (pleasing attack in short):

4. the spammer uses a number of email addresses looking like real honest email users to:
    a. infrequently send spam;
    b. imperceptibly recommend spamming email addresses.

The following simulations show that a spammer can significantly decrease the cost of attacks with a few co-ordinated pleasing attacks on some legitimate recommenders.

A constraint on the search scheme is that it should not overload the network to the point where the quality of service is degraded. From the point of view of our anti-spam application, if a legitimate email user has been considered trustworthy somewhere in the network, this email user should ideally not have to (re)pay a bankable postage when sending emails to any other legitimate users. Also, if the spammer succeeds in a pleasing attack on one legitimate user in the network, if the search scheme is ideal, any spam emails sent after the success of the pleasing attack would be regarded as recommended to be whitelisted. Then, the notion of time becomes important. If the pleasing attack happens at the end of a day and the following spam attacks during the night of that day, any spam will get into the *Inbox* folder until the first email user, who checks his/her emails the earliest that day, will blacklist the pleased recommenders and the email address from which the spam was sent. Of course, given that the search scheme in P2P systems is not ideal, the worst case scenario is that a cluster or clique of email addresses will be fooled in this attack. In actual fact, the spammer may not have been able to obtain the graph of the network topology and hence not know the cluster to which the pleased user belongs in order to attack it.

A number of search schemes can be used. The schemes evaluated in this thesis may be qualified as simple unstructured schemes. Structured P2P schemes could be assessed [126] in the same way. Two search schemes have been evaluated: a search scheme based on a breadth first search (BFS) and another one based on a Random Walk (RW) without back-tracking, both limited in number of hops.

In our real social network described in the previous chapter, a spammer can be added to the network of vertices represented by the email users. This is done either by a spammer joining the network by means of self-introduction, compromising an existing email address in the network, or by means of a pleasing attack. Once the spammer has joined the network, the collaboration emails work for the spammer because they increase the chances of getting more spam through with fewer BPs.

As stated earlier, if the pleasing attack is successful, and a spammer has been automatically whitelisted, a lot of spam can be sent before the spammer is blacklisted by a user. This flaw can be easily remedied by including a human in the process of whitelisting. This way, no email sender is considered to be a legitimate contact without a mandatory human check. The benefit of this is that the spam is caught immediately, and there is zero chance that the second email sent to a second receiver goes in the *Inbox* without having to pay a BP because the first receiver never recommends the sender to be whitelisted. A quick simulation on our network showed that for RW search scheme with *25* users pleased and two hops, *748%* more spam went through when mandatory human check is not applied.

In order to quantify the effect of these random attacks, a set of simulations was carried out, the results of which are presented in Figure 35 and Figure 36. The simulations were averaged out over *1,000* runs, to remove extraneous and spurious results. The parameters were: *5* or *25* emails users are pleased at random; a maximum number of hops just greater than the average diameter of the network (which is *10* since the average diameter of this network is approximately *9.6*) and a maximum of two hops (in order to limit the number of collaboration emails); and using RW and BFS as the search algorithms. Then the number of spam going through without the need of a BP is counted. As can be seen, the cost to the spammer is lowest when a search scheme that gives the highest guarantee of success is used. This is because BFS with a maximum number of hops greater than the diameter of the network guarantees that all connected email addresses in the network are checked.

From a spammer's point of view, the attack should be as cheap as possible in order to maximise the profit. The cost of engineering a *Network Topology Engineered* (NETOPE) attack, which requires building a view of the network, is tricky to estimate. Nowadays, it becomes easier to collect information about the social networks of email users, for example, mining a FOAF network. Thus, an attacker has the resources to engineer attacks beyond random ones. It is already possible to carry out engineered attacks as we demonstrate below. In our case, a real source of social network of email users has easily been mined and attacks engineered based on that information allows any motivated spammer to carry out that type of attacks.

Thus, another set of results is based on simulations where *5* or *25* email users (as above) are pleased, based on the most important email users in the network in a graph relative to all email users according to some importance algorithms. The following set of algorithms has been used: PageRank [24], Betweeness Centrality [23], Degree Distribution Ranker [92] and HITS [104], and are compared to the results for the random attacks in Figure 35 and Figure 36.

In addition to the cost of engineering the attack in terms of mining the network and calculating the importance metrics for each email address in the network, there is a cost involved in orchestrating a pleasing attack, which we will attempt to quantify here. This is composed of two costs: the cost of the bankable postage (*CBP*) when it is cashed in; and the cost of pleasing the email user (*CP*), which is the cost of gaining the trust of the target email user by exchanging a few emails with him/her. Without collaboration the only means to get spam emails in the *Inbox* would be that all spam emails are sent with BPs, which are all cashed in. Thanks to collaboration, some of the emails can be sent without BP because a recommender can be found, using the pleasing attack. At the end of the attack, *NBP* is the number of BPs that have been needed because no recommender was found. *NP* is the number of pleased email users before the sending of one spam to any of the remaining email users.

At this stage, the cost experienced by the spammer is:

$$NP \times CP + NBP \times CBP$$

For simplicity's sake, we assume that *CBP=CP=1 Euro*, although pleasing an email user may cost more than *1* Euro, so the cost to the spammer becomes *NP+NBP* Euro.

Figure 35 highlights that NETOPE attacks based on importance metrics can significantly reduce the cost to the spammer. One of the points underlined by these simulations is that if the search scheme used to find recommenders guarantees that any connected users can be found (as when BFS limited to *10* hops is used), it is less beneficial for the spammer to use NETOPE attacks. It is more important to please users who will give the fewest disconnected users, in other words, to please at least one user in each partition in the network. When the search scheme is not so successful (for example, in the case of BFS limited to two hops or the RW search scheme), it appears that spammers can save a great deal of BPs and cost by using a NETOPE attack and pleasing the most important email users instead of random ones.

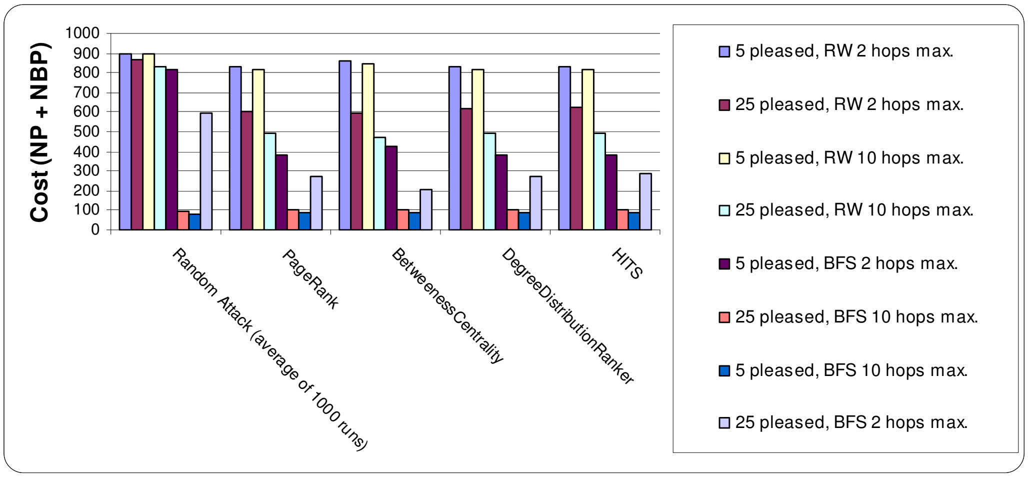


Figure 35. Spammer Cost

The evaluation of the attack resistance of scale-free networks discusses the impact of compromising a ratio of the most important nodes in the network compared to random ones: "for example, when 2% of the nodes fails, the communication between the remaining nodes in the network is unaffected, while, when the 2% of the most connected nodes is removed, then L[28] almost doubles its original value" [38]. In our work, the number of pleased email users can be compared to the resulting total number of fooled email users (spammed and pleased ones). An attack with *5* pleased email users approximately corresponds to *0.55%* of the total email users. An attack with *25* pleased email users approximately corresponds to *2.75%* of the total email users. For *5* pleased email addresses (*0.55%*), the worst case scenario varies from *1.5%* to *91.5%* of fooled email users in random configurations and from *5.8%* to *91.2%* in importance-based configuration. For *25* pleased email addresses (*2.75%*), the worst case scenario varies from *7.4%* to *92.5%* fooled email users in random configurations and from *33.7%* to *91.2%* in importance-based configuration. The greatest benefit between the random attack and the engineered attack happens with the following configuration: *5* pleased email users, the RW search scheme limited to two hops and PageRank. In the latter configuration, *5.9* times more email users fall in the NETOPE attack than in the random attack.

None of the tested importance metrics seem to surpass the others with regard to the increase in profitability of the NETOPE attack – they all have approximately similar costs. All of the importance metrics can be calculated off-line, once the underlying network of connections has been obtained by the spammer. PageRank, DegreeDistributionRanker and HITS all take approximately the same amount of time to compute, as they are $O(n^2)$ computations. Betweeness Centrality takes longer to be calculated, since it is approximately $O(n^3)$.

The final cost associated with the NETOPE attacks is the extra overhead of computation and communication bandwidth due to collaboration. The cost is negligible for the spammer compared to the previous cost, because the cost is mainly borne by other email users. In fact, the collaboration impact does not really concern the spammer since the collaboration is done on behalf of the spammer, in an attempt by the receiving user to find recommenders for the sender. From the point of view of the network, it is bad to spend effort on collaboration for

[28] L in the literature means the characteristic path length of a network. This is the average of the shortest path lengths of all pairs of the *n* nodes in the network.

the purpose of an attack and, as Figure 36 shows, this overhead is much more when engineered attacks are carried out since they tend to require more collaboration.

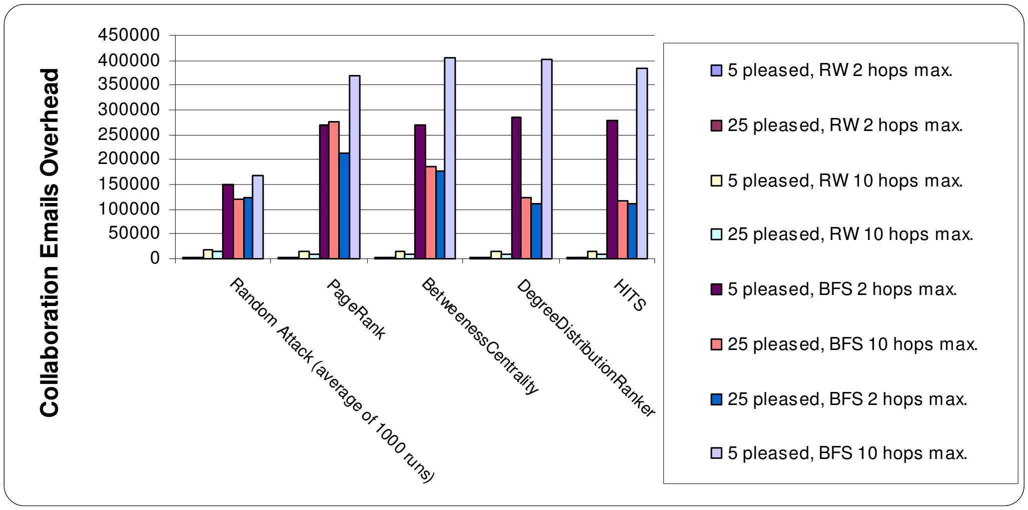


Figure 36. Collaboration Emails Overhead due to Attacks

7.6.2 Proof-of-Friends (PoF) and Trust Transfer

We assume that compromising email accounts and spoofing are not possible. So, the spammers can only rely on the creation of virtual identities that will collude to try to get spam emails through, which corresponds to identity multiplicity attacks. We also assume the trust value-based email prioritisation of the experiment described in Section 6.4.2 and the basic trust transfer assumptions listed in Section 5.5.2.

It is not acceptable to leave the legitimate user proxies to carry out the recommender search on behalf of the spammer, especially since engineered attacks are more harmful and require more recommendation emails (as found in Figure 36). So, we revise the trust transfer process in order to put more work on the spammer side. We slightly modify the search for recommenders needed for trust transfer in order to put more work on the spammer side and take into account privacy considerations. We do not mean it as a strong proof-of-work or bankable postage scheme [3] but it makes more sense to leave the work on the spammer side (when possible), in order to drive up the per-email cost of spamming. The main idea is to return the list of potential recommenders, that is, the contacts of the receiver, to the sender.

Instead of the receiver or recommender contacting further recommenders, the sender will use the list to contact each potential recommender (according to the search algorithm chosen).

There is a potential privacy issue in giving the lists of contacts to be processed by the sender. However, since the sender has no other choice to start with his/her own best friends, the lists of potential recommenders can be adjusted according to the trust value of the sender. If the sender is not trustworthy, no list is returned and it cannot find a path to the receiver. In order to ensure non-repudiation, we assume that all requests and responses are signed. Finally, the receiver has to locally verify the signatures (without having to re-contact the recommenders). We need another protection mechanism related to privacy disclosure. Each time an email is sent to a new receiver, the email sender sets two local values on a *[0,1]* scale. The first value corresponds to the level of privacy information that the receiver is allowed to see from *0* (none) to *1* (full information). The second value, which is specified in the sender's email when the receiver must change it, corresponds to the level of privacy required by the contacts of the receiver to be allowed to use the sender as a recommender. $Contact_{Privacy}(0.8,0.7)$ means that the contact has a privacy level of *0.8* and allows the recommender to disclose their relationship to subjects with privacy level greater than (or equal to) *0.7*.

Thus, the default trust transfer is changed to the one in Figure 37 that we call *PoF trust transfer* [159] (the search requests for the different recommenders are not represented).

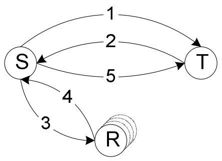


Figure 37. Proof of Friends Trust Transfer[17]

The trust transfer consists of the following steps:

1. The subject requests an action of the trustor;
2. The trustor replies that an amount of trustworthiness, *TA*, is required before it will grant the request;

Until a complete recommender chain is found (or the search's time-to-live expires):

3. The subject starts to query his/her contact email addresses, who pass the *RSP*, to find a recommender chain to the trustor;
4. If the privacy test is passed and the recommender does not know the receiver, it sends back the list of privacy checked contacts to the sender, including a statement signed by the recommender that he/she is willing to recommend the sender as part of a recommender chain, if one can be found;

Once the recommender chain is found, every recommender involved confirms that they have transferred trustworthiness accordingly by signed statement;

5. The subject sends the recommendation to the trustor.

In the example of Figure 38, the sender has only one contact, who does not know the receiver target. However, this contact has one contact who knows the receiver. In this scenario, the *RSP* requires that a potential recommender must have a balance of at least *2TA* on the trustor side. The *RP* is that the subject must have a balance greater than *TA* on the recommender side.

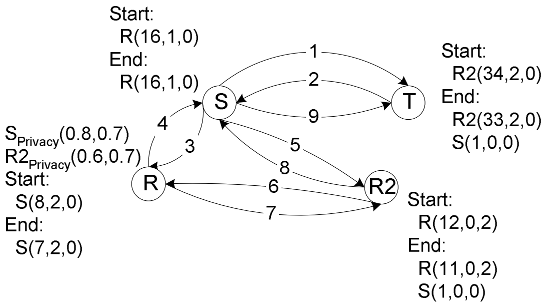


Figure 38. Proof of Friends Trust Transfer Example[17,18]

In our default collaboration scheme, the following emails are exchanged:

1. The sender sends an email to a new receiver;
2. The receiver replies that the proof of having sent one legitimate email is needed to get the email out of the spam folder. In other words, *TA* is *(1,0,0)*;
3. The sender starts contacting his/her list of contacts; in this case, there is only one contact;
4. The sender's only contact is queried; it does not know the receiver, so it starts checking its list of contacts to see if the privacy test is passed; in this case, the sender has a privacy disclosure trust value of *0.8*, which is higher than the threshold specified by the potential recommender *R2*; therefore, the contact is passed back to the sender: more precisely, the contact signs an email stating that it is inclined to recommend *(1,0,0)* for the sender based on a recommender chain including *R2*;
5. The recommender's contact is queried by the sender and is found to be a contact of the receiver; it agrees to recommend *(1,0,0)* for the sender to the receiver as long as it receives a confirmation from *R*. This is because *R* has a balance of *10* on *R2*'s side, which is greater than *TA*. *S* has a balance of *6* on *R*'s side, so the *RP* is passed on all nodes in the recommender chain;
6. An email is sent to *R* in order to confirm that the trustworthiness in the sender has been decreased by *(1,0,0)* on *R*'s side;
7. The trustworthiness in the sender is decreased by one and a confirmation email is sent back to *R2*;
8. *R2* transfers some trustworthiness (one supporting outcome) from *R* to the sender; then, an email confirming that *R2* recommends *(1,0,0)* is sent back to the sender;
9. The recommendation is passed to the receiver, who transfers *(1,0,0)* trustworthiness from *R2*'s trust value to the sender's trust value.

In doing so, the spammer has now to endure most of the recommendation work and his/her attacks are more expensive since many more real emails from the spammer are needed.

At first glance, the email infrastructure may be challenged by trust transfer because of the overhead of emails to find potential recommenders. However, in the modified search we present, the spammer must start the search with his/her own friends. As a rule of thumb [3], increasing the cost to the spammer for an attack is desirable and our new trust transfer further increases this cost when engineered attacks are used. This means that the network load is localised to the spammer's network of honest friends (of whom there are unlikely to be many). In addition to this, recommendations are usually only required for newcomers. Finally, the overhead of collaboration is limited by the *RSP* and more intelligent directed search schemes, for example [68] based on local knowledge about similarity between the sender and the local contacts, do not flood the network.

7.7 ASUP Qualitative Evaluation

In this section, the entification framework is assessed according to the points of comparison defined in Section 4.5.5, which cover the four ASUP dimensions.

Concerning the *adaptability* points of comparison:

1. *to the available technical infrastructure and scalability:* the set of pluggable recognition schemes is large; in order to scale, specific ER schemes can be used and context is useful to forget/garbage collect virtual identities unlikely to be met in the current context of interest; collaboration with trustworthy entities can be used to share the load of data;
2. *to the trade-off between security protection and its tolerable cost*: the risk analysis of the trust engine may maintain the right trade-off; ER schemes are associated with level of confidence in recognition, which can be considered as dynamic technical trust values;
3. *to the context*: the trust value varies across contexts: there are different trust contexts; the ER module generates environmental context evidence and the trust engine can tune the ER module; context can be used to select the appropriate virtual identity;

4. *to the use of many virtual identities*: link, fusionym and trust transfer are means to maintain an accurate trust value;
5. *to new security domains*: thanks to collaboration and dynamic enrolment, the adaptation to new security domains is smoother; trust may be built from scratch, interaction after interaction.

Concerning the *security* points of comparison:

1. *cost of the management overhead to be done by the user:* once the trust policy is written, the trust value may automatically evolve interaction after interaction; some ER schemes may be more dynamic (but less secure);
2. *openness to newcomer*: newcomer virtual identities can dynamically join the community and their trust value is built interaction after interaction; trust transfer facilitates the consideration of trustworthy newcomers; however, newcomers must have trustworthily interacted with other entities at some stage;
3. *through collaboration*: direct observation, recommendations and collaboration are used; security decisions are possible and should be based on the explicit ER technical trust;
4. *against identity multiplicity attacks*: the Sybil attack is mitigated through trust transfer in case the trust values corresponds to the count of event outcomes;
5. *against identity usurpation attacks*: the level of confidence in recognition and technical trust can be used to mitigate spoofing;
6. *uncertainty consideration*: trust values (including for technical trust) may have an element of uncertainty, for example, if the SECURE trust value format is used;
7. *explicit evidence-based trust levels*: the trust values based on event outcomes are explicitly computed based on pieces of evidence;
8. *based on standards*: the different parts of the entification framework have been reviewed in a number of publications;
9. *possible full decentralisation or presence of trusted-third-parties:* a trusted-third-party is not mandatory: it depends on the ER scheme, new message-based ER schemes, which do not require trusted-third-parties, have been created; full

decentralisation may be achieved thanks to trust transfer in some scenarios as presented in this thesis;

10. *clear separation between authentication and authorisation:* there is a clear separation in the end-to-end trust and the notion of virtual identity is at the core of the framework;
11. *mandatory assumption of effective system trust:* the assumption is that it is possible to rely on event outcomes and collaboration to build interpersonal trust from scratch; system trust may be used but it is not mandatory.

Concerning the *usability* points of comparison:

1. *of bootstrapping/enrolment:* dynamic enrolment may be provided by some entity recognition schemes since enrolment is postponed in the ER process; in case of a complete newcomer, the potential element of uncertainty in the trust value may be used;
2. *of management of multiple identities*: the disclosure of the appropriate virtual identity can be done based on the trust values and context, for example, automatically switching identities based on location facilitates the management;
3. *of the specification of privacy policies*: context and trust values can be used;
4. *of the specification of trust policies*: the count of event outcomes appears to be intuitive, especially to include the notion of uncertainty when the outcome of the event is yet unknown.

Concerning the *privacy* points of comparison:

1. *pseudonymity*: many virtual identities per user are supported and encouraged;
2. *link to the real-world identity*: the link is not mandatory; recognition is sufficient; many virtual identities make it harder to infer the link to the real-world identity;
3. *negotiation*: the link mechanism is used to trade privacy for trust;
4. *user-centric and in control*: the user is in control.

7.8 Summary

This chapter completes the evaluation of the entification framework with results in the real settings of the experiments composing the test bed.

The email application domain is clearly flawed by identity multiplicity and usurpation attacks. The entification framework used in this domain has been shown to mitigate both of these types of attacks. It is possible to enhance email anti-spoofing – against usurpation attacks – thanks to new message-based ER schemes. In addition to this security enhancement, evaluated by a peer-reviewed threat analysis, these new techniques can be used transparently by the users without a worldwide change to the legacy email infrastructure. Thus, the usability requirement is taken into account. These schemes do not require a trusted-third-party or CA; they are carried out in a P2P fashion. Performance and protection overhead results are given in order to be able to choose the most appropriate trade-off for the message-based application under consideration. Another choice concerns the deployment of this type of ER proxy because different configurations have different impact on privacy. Concerning identity multiplicity, trust transfer is also evaluated in this email application domain. In fact, the recommender search used in the trust transfer is revised to take into account privacy and economic aspects of the application domain. In order to do so, the cost and overhead of different attacks on real networks of email users have been evaluated. The trust transfer applied in email settings is fully decentralised.

The evaluation of the end-to-end trust and context adaptation is also covered. Means are given to evaluate technical trust, especially at the level of ER. The use of the AAS of the ER scheme gives a static evaluation. Another static evaluation is done in the smart home domain with the VER scheme: this corresponds to an empirical assessment of the technical trust of vision-based ER schemes in real settings. However, since the environmental context and the security of technologies change, a dynamic evaluation of technical trust is introduced. This consists of considering the ER scheme not as a kind of system trust but as an entity, whose interpersonal trust value is managed by trust engines. Adaptability to the context is further evaluated with the VER scheme. Since the VER scheme generates a lot of recognition clues, the evaluation of the use of context to manage this information shows that it can be a means to deal with the potentially huge amount of recognition clues. Another set of results on performance and security is obtained with the deployable JXTA-Java pipes, where adjunct security protection clearly introduces an overhead. These results can be used to select the most appropriate communication channel and tune the CTK. This type of information is crucial for the SECURE trust engine to feed the risk module.

Finally, the entification framework is compared to the reviewed frameworks at the beginning of the thesis based on its answers to the same identified points of comparison. The entification framework clearly covers the four ASUP requirements, which is not the case for these existing frameworks. In future, we hope that further large-scale HCI user trials will be carried out.

CHAPTER 8: CONCLUSIONS AND FUTURE WORK

This thesis presented the entification framework designed to address the requirements of the four ASUP dimensions in global computing environments. This chapter summarises the most significant achievements of the work described in this thesis and outlines its contribution to the state-of-the-art. This thesis is then concluded with a discussion of related research issues that remain open for future work.

8.1 Achievements

Traditional security solutions, based on the assumption of a dedicated administrator, would overwhelm users in global computing environments. This problem implies requirements in four dimensions: adaptability, security, usability and privacy. Trust engines have been proposed to let the computing entities make autonomous decisions based on evidence, which has been experienced by themselves or exchanged between them. The motivation for the work presented in this thesis arose from the observation that a central but weak element in the state-of-the-art research was the notion of identity in computational trust frameworks. The link between the virtual identity and the real-world identity may be ineffective in global computing where real world recourse may not be feasible and largely contributes to the burden of security administrative tasks. Furthermore, the trusted-third-parties or CAs that may alleviate this burden may not be relevant since they may not be trusted or immediately contactable on a global scale.

Within the context of this thesis we have designed, implemented and evaluated the first computational framework, called the entification framework, that integrates the human notions of entity recognition and trust. It fills the gap between identities and their level of trust, which is one of the eight “major issues” [43] in developing identity management for the next generation of distributed applications. The other issue of “linkability” [43] is at the core of the framework. It also addresses other issues presented in [43] and mentioned in the paragraphs below.

In order to do so, a computational model of entity recognition has been developed and integrated in a trust engine as a replacement for the authentication process: this is called the Entity Recognition (ER) process.

Thanks to the possibility to build trust from scratch, interaction after interaction, it is sufficient to recognise the virtual identities and the link to their real-world identity is not mandatory. Since this link to the real-world identity is not mandatory, the enrolment of virtual identities is facilitated and can be dynamic with little human intervention – the usability requirement is addressed in this way. In order to mitigate the fact that more dynamic enrolment may be less secure and the possibility of identity usurpation attacks, the outcome of the ER process includes a level of confidence in recognition.

In fact, the ER process can be more or less dynamic or secure depending on the chosen ER schemes. In order to take into account security considerations, layers of trust have been identified, namely technical trust, that is, trust in the components of the underlying technical infrastructure, and trust in the interacting entities. Both layers are explicitly taken into account in the framework: the end-to-end trust takes into account both the level of technical trust of the ER scheme and the level of trust of the interacting virtual identity, computed by usual computational trust. The framework has successfully been used with the advanced SECURE trust engine [27, 151, 154], which has an explicit risk analysis component. The components of technical infrastructure have been abstracted to entity recognition. Static and dynamic evidence-based means to compute the level of technical trust in ER schemes have been proposed and evaluated. These levels of technical trust can be used for threat analysis, especially concerning identity usurpation attacks. When dynamic evidence-based means are used, technical trust is changed from system trust to interpersonal trust in the technical components. The novel exchange of recognition clues between entities combined with the technical trust of ER schemes mitigates the “cross-domain communication” [43] issue, which is strengthened by the possible absence of global naming and centralised authentication authorities in global computing.

These concepts and techniques were realised in prototype implementations. As described in Chapter 6 and 7, a range of application scenarios, which exhibit global computing characteristics, were then selected to conduct a number of evaluation experiments. Pure ER schemes, which do not need the binding between the virtual identity and the real-world identity, have been demonstrated in two application domains: message-based and vision-based. New message-based ER techniques have been demonstrated in the email anti-spam domain. These new techniques have been shown to provide innovative anti-spoofing on top of the legacy email system in a transparent way to the email users, even for users who do not adopt the solution. These novel techniques are significant for global computing because they work in a P2P way, where authentication techniques requiring a priori trusted-third-parties may not be possible. In the vision-based ER scenario, the use of context-awareness (expected in global computing) has tackled the problem of the management of a large number of recognition clues and their retrieval. The adaptability of trust engines is extended by the possibility to use a large range of different ER schemes. It is novel to integrate a pluggable recognition module in a trust engine. The genericity of the approach has been demonstrated thanks to different ER schemes based on: the verification of the knowledge of common past history; signature validation; face template matching; height and colour matching; and challenge/responses. A generic tool kit has been implemented in Java for message-based recognition. This tool kit can be instantiated with a number of message-based ER schemes (as long as they are developed with the given API), that can be tuned based on context and management commands. The CTK is generic enough to instantiate a large range of message-based ER schemes from unicast to multicast settings. Performance results have been obtained: from experiments with hardware and software in real settings; analytic calculation; and simulations of messages exchanged in real-world social networks of thousands of users extracted from online data. More secure JXTA pipes are found to be slower and a trade-off between performance and security may bring better overall results. Performance results can be used in the risk module of the trust engine, which can tune the ER module. In return, the ER module contributes to the risk analysis by informing the trust engine about the current context by logging ER related evidence.

Beyond the fact that privacy decisions would overwhelm the user in pervasive computing environments, this thesis clearly shows the impact that computational trust, which requires knowledge, has in general on privacy. In addition, this thesis identifies the foundations for mitigating this impact. The fact that trust can be built without link to the real-world identity has shed light on the possibility to encourage privacy protection by the use of ER schemes using pseudonyms. The issue of the inherent conflict between privacy and trust has been identified. Actually, both privacy and trust depend on knowledge about the interacting entities albeit in opposite ways. The more evidence is known; the more accurate trustworthiness is reached; the less privacy is left. However, with full knowledge and no privacy, the need of trust vanishes. In high privacy setting, there is a high need of trust: computational trust based on interpersonal trust and trust values rather than system trust is a means for trust in privacy protected environments.

ER schemes, which do not require an explicit link between the virtual identity and the real-world identity, are better from a privacy of view than schemes with compulsory link to the real-world identity. Moreover, for enhanced privacy protection, the entification framework encourages the use of multiple pseudonyms per person because it is harder to infer the link with the real-world identity. From a usability point of view, context can be used to select the appropriate local pseudonym without the user's intervention. The message-based tool kit supports the use of multiple pseudonyms and may facilitate their use, using context-awareness. A ubiquitous message-based payment scenario demonstrates the selection of the appropriate pseudonym based on the location of the user.

However, since privacy expectations change and depend on the user, the framework of this thesis is the first to support the functionality to negotiate privacy for trust between the interacting entities. The privacy/trust trade model allows the requester of an interaction to relinquish some privacy in order to increase trust evidence to be able to reap the benefit of being trustworthy. This corresponds to the "privacy […] negotiation" issue [43]. This is possible thanks to the proposed link functionality. For example, different pseudonyms can be linked in order to increase the amount of trust evidence. The link mechanism is demonstrated with message-based ER schemes based on signature validation. The link mechanism has been implemented with the CTK and demonstrated in the ubiquitous payment scenario. This gives an example of the issue of "lifecycle management" [43] going beyond identity creation, use and deletion. Since virtual identities can now explicitly be linked, the final contribution concerns techniques to mitigate the issues related to the use of multiple pseudonyms, either malicious, as in identity multiplicity attacks, or not. When the level of trust is based on counts of interaction outcomes, the techniques of fusionym and trust transfer address both accurate computation of the level of trust in spite of self-recommendations and identity multiplicity. They address the issue of "identity proliferation" [43]. When the level of trust is based on counts of interaction outcomes, fusionym is introduced to be able to compute the overall trust value of multiple virtual identities, which are proven to be linked. Then, trust transfer mitigates self-recommendations and identity multiplicity attacks. However, trust transfer is still limited to scenarios where there are many interactions with the recommenders. Trust transfer corresponds to a local decentralised scalar metric and is evaluated with simulations of a real social network of email users extracted from online data. Novel networked engineered attacks using a priori knowledge about the network to select the most well-connected entities empirically proved to be more harmful than attacks targeting random entities. Based on this fact, a slightly different trust transfer, which increases the cost to the attacker and better protects the privacy of the network, has been developed. Additionally, this work underlines that the network topology is significant when security through collaboration is used.

8.2 Open Research Issues

Large-scale user trials would address remaining HCI issues. As has been introduced at the end of the previous achievements, an area for future exploration includes the issue of collaboration with entities that are part of a particular network of entities. The notion of groups of entities may be important to carry out a proper risk analysis in the trust engine. Generally, in the identity terminology, the notion of groups is followed by the notion of roles. A virtual identity may be a group or a role. Therefore, the trust engines may have to integrate these notions of roles and groups. An interesting question occurs when the real-world identities behind the group are unknown or it is not known which real-world identity speaks for the group. There are other open research issues related to privacy and trust. In order to adequately negotiate privacy for trust, there is a need of quantification of the trade-off between privacy, trust and utility. This may require the quantification of a piece of evidence from a privacy disclosure point of view or a trust assessment impact point of view. There is also the issue of the sequencing of pieces of evidence: the combination of a new piece of evidence subsequent to the release of a first different piece of evidence may be worse from a privacy point of view than if the initial piece of evidence had been different. Then, partitionym may be useful to be carried out in some situations. A final open research issue concerning privacy concerns the possibility to negotiate the ER schemes without compromising privacy. Actually, more than privacy may be compromised due to ER schemes negotiation.

BIBLIOGRAPHY

[1] "Crypto-ID JXTA", Web site, http://crypto-id.jxta.org/.

[2] M. Abadi, "On SDSI's Linked Local Name Spaces", in *Journal of Computer Security*, vol. 6(1), pp. 3-21, 1998, citeseer.nj.nec.com/abadi98sdsis.html.

[3] M. Abadi, A. Birrell, M. Burrows, F. Dabek, and T. Wobber, "Bankable Postage for Network Services", in *Proceedings of ASIAN*, pp. 72-90, LNCS, Springer, 2003, http://research.microsoft.com/research/sv/PennyBlack/demo/ticketserver.pdf.

[4] F. Abdul-Rahman and S. Hailes, "A Distributed Trust Model", in *Proceedings of the New Security Paradigms Workshop*, pp. 48-60, ACM, 1997, http://citeseer.nj.nec.com/347518.html.

[5] F. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities", in *Proceedings of the Hawaii International Conference on System Science*, Hawaii, 2000, http://citeseer.nj.nec.com/235466.html.

[6] F. Abdul-Rahman and S. Hailes, "Using Recommendations for Managing Trust in Distributed Systems", in *Proceedings of the Malaysia International Conference on Communication*, IEEE, 1997, http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/micc97.ps.

[7] J. Abendroth, "A Unified Access Control Mechanism", PhD Thesis, Trinity College Dublin, 2004.

[8] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible Authentication Protocol (EAP)", in RFC3748, Network Working Group, 2004, http://www.networksorcery.com/enp/rfc/rfc3748.txt.

[9] A. Acquisti, R. Dingledine, and P. Syverson, "On the Economics of Anonymity", in J. Camp and R. Wright, editors, Financial Cryptography, LNCS, Springer Verlag, 2003, http://citeseer.nj.nec.com/acquisti03economics.html.

[10] D. Agrawal and D. Kesdogan, "Measuring Anonymity: The Disclosure Attack", in *IEEE Security&Privacy*, 2003.

[11] R. Agrawal, "Why is P3P Not a PET?" in *W3C Workshop on the Future of P3P*, 2002, http://www.w3.org/2002/p3p-ws/pp/epic.pdf.

[12] J. Al-Muhtadi, M. Anand, M. D. Mickunas, and R. Campbell, "Secure Smart Homes using Jini and UIUC SESAME", 2000, http://www.cs.uiuc.edu/Dienst/Repository/2.0/Body/ncstrl.uiuc_cs/UIUCDCS-R-2000-2184/pdf.

[13] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments", in the *International Conference of Distributed Computing Systems*, Vienna, Austria, pp. 65-74, 2002, http://ciae.cs.uiuc.edu/mist/mist.pdf.

[14] J. Arkko and P. Nikander, "Weak Authentication: How to Authenticate Unknown Principals without Trusted Parties." in *Proceedings of the Security Protocols Workshop*, pp. 5-19, LNCS, Springer, 2002, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2845&spage=5.

[15] T. Aura and C. Ellison, "Privacy and Accountability in Certificate Systems", Research Report A61, Helsinki University of Technology, 2000, http://www.tcs.hut.fi/old/papers/aura/HUT-TCS-A61.pdf.

[16] O. Berthold and H. Langos, "Dummy traffic against long term intersection attacks", http://page.inf.fu-berlin.de/~berthold/publ/BeLa_02.pdf.

[17] T. Beth, M. Borcherding, and B. Klein, "Valuation of Trust in Open Networks", in *Proceedings of the 3rd European Symposium on Research in Computer Security*, 1994, http://citeseer.ist.psu.edu/beth94valuation.html.

[18] J. Bigun, J. Fierrez-Aguilar, J. Ortega-Garcia, and G.-R. J., "Multimodal Biometric Authentication using Quality Signals in Mobile Communications", in *Proceedings of the 12th International Conference on Image Analysis and Processing*, IEEE, 2003, http://csdl.computer.org/comp/proceedings/iciap/2003/1948/00/19480002abs.htm.

[19] A. Birrell, B. Lampson, R. M. Needham, and M. D. Schroeder, "A Global Authentication Service without Global Trust", in *Symposium on Security and Privacy*, pp. 223-230, IEEE, 1986.

[20] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "Keynote: Trust Management for Public-Key Infrastructures", in *Proceedings of the Security Protocols International Workshop*, pp. 59-63, Cambridge, England, 1998, http://citeseer.nj.nec.com/blaze98keynote.html.

[21] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized Trust Management", in *Proceedings of the 17th IEEE Symposium on Security and Privacy*, pp. 164-173, IEEE Computer Society, 1996, http://citeseer.nj.nec.com/blaze96decentralized.html.

[22] C. Boyd and A. Mathuria, "Protocols for Authentication and Key Establishment", Springer, 2003, http://www.springer.de/cgi/svcat/search_book.pl?isbn=3-540-43107-1.

[23] U. Brandes, "A Faster Algorithm for Betweenness Centrality", in *Journal of Mathematical Sociology*, vol. 25(2), pp. 163-177, 2001.

[24] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", in *Computer Networks*, 1998, http://dbpubs.stanford.edu:8090/pub/1998-8.

[25] B. D. Brunk, "Understanding the Privacy Space", in *First Monday*, vol. 7, no. 10, Library of the University of Illinois, Chicago, 2002, http://www.firstmonday.org/issues/issue7_10/brunk/index.html.

[26] K. A. Burton, "Design of the OpenPrivacy Distributed Reputation System", OpenPrivacy.org, 2002, http://www.peerfear.org/papers/openprivacy-reputation.pdf.

[27] V. Cahill, et al., "Using Trust for Secure Collaboration in Uncertain Environments", in *Pervasive Computing*, July-September, vol. 2(3), IEEE, 2003, http://csdl.computer.org/comp/mags/pc/2003/03/b3toc.htm.

[28] R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampermane, and M. D. Mickunas, "Towards Security and Privacy for Pervasive Computing", in *Proceedings of the International Symposium on Software Security*, Keio University, Tokyo, Japan, November 8, 2002, http://citeseer.nj.nec.com/560164.html.

[29] P. Castro and R. Muntz, "Using Context to Assist in Multimedia Object Retrieval", in *Workshop on Multimedia Intelligent Storage and Retrieval Management*, Orlando., ACM, 1999, http://www.info.uqam.ca/~misrm/papers/castro.ps.

[30] D. Chaum, "Achieving Electronic Privacy", in *Scientific American*, vol. August, pp. 96-100, 1992, http://www.chaum.com/articles/Achieving_Electronic_Privacy.htm.

[31] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", in *Communications of the ACM*, vol. 24 (2), 1981, http://world.std.com/~franl/crypto/chaum-acm-1981.html.

[32] D. M. Chess, C. C. Palmer, and W. S. R., "Security in an autonomic computing environment", in *IBM Systems Journal*, vol. 42(1), 2003.

[33] D. Clark and D. Wilson, "A Comparison of Commercial and Military Computer Security Policies", in *Proceedings of the Symposium on Security and Privacy*, pp. 184-194, IEEE, 1987.

[34] T. M. Cooley, "A Treatise on the Law of Torts", Callaghan, Chicago, 1888.

[35] S. R. Covey, "The 7 habits of highly effective people", ISBN 0-684-85839-8, Franklin Covey, 1989.

[36] L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle, "The platform for privacy preferences 1.0 (P3P1.0) specification", W3C Recommendation, 2002, www.w3.org/TR/P3P/.

[37] S. Creese, M. Goldsmith, B. Roscoe, and I. Zakiuddin, "Authentication for Pervasive Computing", in *Proceedings of the First International Conference on Security in Pervasive Computing*, LNCS, Springer, 2003.

[38] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, "Efficiency of Scale-Free Networks: Error and Attack Tolerance", in *Physica,* A n. 320, Elsevier, 2003, http://www.w3.org/People/Massimo/papers/2003/tolerance_physicaA_03.pdf.

[39] CSA, "Model Code for the Protection of Personal Information", vol. CAN/CSA -Q830-96, Canadian Standards Association, 1995, http://www.csa.ca/standards/privacy/code/Default.asp?language=English.

[40] CypherTrust, Web site, http://www.ciphertrust.com/researchcenter/index.php.

[41] E. Damiani, S. D. C. d. Vimercati, S. Paraboschi, and P. Samarati, "Managing and sharing servants' reputations in P2P systems", in *Transactions on Knowledge and Data Engineering*, vol. 15(4), pp. 840-854, IEEE, 2003.

[42] E. Damiani, S. D. C. d. Vimercati, S. Paraboschi, and P. Samarati, "P2P-Based Collaborative Spam Detection and Filtering", in *Proceedings of the Fourth International Conference on Peer-to-Peer Computing*, 2004, http://csdl.computer.org/comp/proceedings/p2p/2004/2156/00/21560176abs.htm.

[43] E. Damiani, S. D. C. d. Vimercati, and P. Samarati, "Managing Multiple and Dependable Identities", in *Internet Computing*, 7(6), pp. 29-37, IEEE, 2003.

[44] Data Protection Working Party, "Opinion 10/2001 on the need for a balanced approach in the fight against terrorism", The European Commission, 2001, http://europa.eu.int/comm/internal_market/en/dataprot/wpdocs/wp53en.pdf.

[45] Z. Despotovic and K. Aberer, "Maximum Likelihood Estimation of Peers' Performance in P2P Networks", in *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004, http://www.eecs.harvard.edu/p2pecon/confman/papers/s2p3.pdf.

[46] Z. Despotovic and K. Aberer, "Trust and Reputation Management in P2P Networks", Tutorial, CEC, 2004, http://lsirpeople.epfl.ch/despotovic/CEC2004-Tutorial.pdf.

[47] N. Dimmock, "How much is 'enough'? Risk in trust-based access control", in *Proceedings of the International Workshops on Enabling Technologies (Special Session on Trust Management)*, pp. 281-282, 2003, http://www.cl.cam.ac.uk/Research/SRG/opera/publications/Papers/ned21-wetice03.ps.

[48] N. Dimmock, J. Bacon, A. Belokosztolszki, D. Eyers, D. Ingram, and K. Moody, "Preliminary Definition of a Trust-based Access Control Model", SECURE Deliverable 3.2, 2004, http://secure.dsg.cs.tcd.ie.

[49] DoD, "Trusted Computer Security Evaluation Criteria", in *DoD 5200.28-STD*, Department of Defense, 1985.

[50] J. R. Douceur, "The Sybil Attack", in *Proceedings of the International Workshop on Peer-to-Peer Systems*, 2002, http://research.microsoft.com/sn/farsite/IPTPS2002.pdf.

[51] DTN, "Delay Tolerant Network", Web site, Internet Research Task Force, http://www.dtnrg.org.

[52] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", in *Transactions on Information Theory*, IT-31(4), pp. 469-472, IEEE, 1985, http://portal.acm.org/citation.cfm?id=19480.

[53] T. T. Eliot, "Little Gidding", in Four Quartets, 1942, http://www.english.uiuc.edu/maps/poets/a_f/eliot/eliot.htm.

[54] C. M. Ellison, "SPKI requirements", in RFC 2692, IETF, 1999, ftp://ftp.isi.edu/in-notes/rfc2692.txt.

[55] EPIC, "Airline privacy violation cases", 2004, http://www.epic.org/open_gov/foiagallery.html.

[56] EU, "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data", 1995, http://europa.eu.int/ISPO/legal/en/dataprot/directiv/directiv.html.

[57] EU, "Directive 2002/58/EC of The European Parliament and the Council of the European Union", in *Official Journal of the European Communities*, 2002, http://www.oftel.gov.uk/ind_info/eu_directives/data0702.pdf.

[58] D. F. Ferraiolo and D. R. Kuhn, "Role Based Access Control", in *Proceedings of the 15th National Computer Security Conference*, NIST, 1992, http://csrc.nist.gov/rbac/ferraiolo-kuhn-92.pdf.

[59] FOAF, "The Friend-of-a-Friend Project", http://www.foaf-project.org/.

[60] J. Frankel and T. Pepper, "Gnutella", Nullsoft, 2000, http://en.wikipedia.org/wiki/Gnutella.

[61] B. Franklin, "The Life and Letters of Benjamin Franklin", E.M. Hale & Co.

[62] E. Friedman and P. Resnick, "The Social Cost of Cheap Pseudonyms", in *Journal of Economics and Management Strategy*, vol. 10(2), pp. 173-199, 2001, http://www.si.umich.edu/~presnick/papers/identifiers/.

[63] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force directed placement", in *Software: Practice and Experience*, 21(11), ACM, 1991, http://portal.acm.org/citation.cfm?id=137557.

[64] J. Galvin, "(In)Security from End to End", 2000, http://infosecuritymag.techtarget.com/articles/march00/features2.shtml.

[65] S. L. Garfinkel, "Email-Based Identification and Authentication: An Alternative to PKI?" in *IEEE Security&Privacy*, 2003, http://csdl.computer.org/comp/mags/sp/2003/06/j6toc.htm.

[66] M. Gasser, A. Goldsteing, C. Kaufman, and B. Lampson, "The Digital Distributed System Security Architecture", in *Proceedings of the National Computer Security Conference*, pp. 305-319, NIST/NCSC, 1989, http://research.microsoft.com/~lampson/41-DigitalDSSA/Acrobat.pdf.

[67] GnuPG, "The GNU Privacy Handbook", The Free Software Foundation, 1999, http://www.gnupg.org/gph/en/manual.html.

[68] J. Golbeck and J. Hendler, "Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks", 2004, http://www.mindswap.org/papers/GolbeckEKAW04.pdf.

[69] J. Golbeck and J. Hendler, "Reputation Network Analysis for Email Filtering", in *Proceedings of the First Conference on Email and Anti-Spam* (CEAS), 2004, http://www.ceas.cc/papers-2004/177.pdf.

[70] J. Golbeck and B. Parsia, "Trusting Claims from Trusted Sources: Trust Network Based Filtering of Aggregated Claims:" in *Proceedings of the 3rd International Semantic Web Conference*, 2004, http://www.mindswap.org/papers/Jen-ISWC04.pdf.

[71] I. Goldberg, "A Pseudonymous Communications Infrastructure for the Internet", PhD Thesis, University of California at Berkeley, 2000, http://www.isaac.cs.berkeley.edu/~iang/thesis-final.pdf.

[72] D. Gollmann, "Computer Security", ISBN 0-471-97844-2, John Wiley&Sons, 1999.

[73] T. Grandison, "Trust Management for Internet Applications", PhD Thesis, Imperial College London, 2003, http://www.doc.ic.ac.uk/~tgrand/PhD_Thesis.pdf.

[74] T. Grandison and M. Sloman, "Specifying and Analysing Trust for Internet Applications", 2002, http://citeseer.nj.nec.com/grandison02specifying.html.

[75] T. Grandison and M. Sloman, "A Survey Of Trust In Internet Applications", in *Communications Surveys*, IEEE, 2000, http://citeseer.nj.nec.com/grandison00survey.html.

[76] T. Grandison and M. Sloman, "Trust Management Tools for Internet Applications", in *Proceedings of iTrust*, LNCS 2693, Springer, 2003, http://www.doc.ic.ac.uk/~mss/Papers/iTrust-03.pdf.

[77] E. Gray, J.-M. Seigneur, Y. Chen, and C. D. Jensen, "Trust Propagation in Small Worlds", in *Proceedings of iTrust*, LNCS 2693, Springer, 2003, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=239.

[78] N. Haller, "The S/KEY One-Time Password System", in *Proceedings of the Symposium on Network and Distributed System Security*, 1994, http://citeseer.ist.psu.edu/haller94skey.html.

[79] R. Hes and J. Borking, "Privacy Enhancing Technologies: The Path to Anonymity", ISBN 90 74087 12 4, 2000, http://www.cbpweb.nl/downloads_av/AV11.PDF.

[80] R. Housley and T. Polk, "Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure", ISBN: 0471397024, John Wiley & Sons, 2001.

[81] IBM/Microsoft, "Federation of Identities in a Web services world", 2003, http://www-106.ibm.com/developerworks/webservices/library/ws-fedworld/.

[82] IETF, "Authentication, Authorization and Accounting Working Group", Web site, IETF, http://www.ietf.org/html.charters/aaa-charter.html.

[83] IETF, "Public-Key Infrastructure (X.509)", http://www.ietf.org/html.charters/pkix-charter.html.

[84] IST, "Global Computing", EU, 2004, http://www.cordis.lu/ist/fet/gc.htm.

[85] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition", in *Transactions on Circuits and Systems for Video Technology*, IEEE, August 2003, 2003.

[86] U. Jendricke, M. Kreutzer, and A. Zugenmaier, "Pervasive Privacy with Identity Management", in *Proceedings of the Workshop on Security in Ubiquitous Computing*, Ubicomp, 2002, http://citeseer.nj.nec.com/544380.html.

[87] X. Jiang, J. I. Hong, and J. A. Landay, "Approximate Information Flows: Socially Based Modeling of Privacy in Ubiquitous Computing", in *Proceedings of the International Conference on Ubiquitous Computing*, LNCS 2498, pp. 176-193, Springer, 2002, http://guir.berkeley.edu/projects/uisper/pubs/ubicomp2002-aif.pdf.

[88] C. M. Jonker and J. Treur, "Formal Analysis of Models for the Dynamics of Trust based on Experiences", in *Proceedings of the Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1999, http://citeseer.nj.nec.com/198191.html.

[89] A. Jøsang, "A Logic for Uncertain Probabilities", in *Fuzziness and Knowledge-Based Systems*, vol. 9(3), 2001.

[90] A. Jøsang, "The right type of trust for distributed systems", in *Proceedings of the New Security Paradigms Workshop*, ACM, 1996, http://citeseer.nj.nec.com/47043.html.

[91] A. Jøsang, "A Subjective Metric of Authentication", in *Proceedings of the European Symposium On Research in Computer Security*, Springer, 1998, http://citeseer.nj.nec.com/josang98subjective.html.

[92] JUNG, "JUNG, the Java Universal Network/Graph Framework", http://jung.sourceforge.net/index.html.

[93] B. M. Juric, T. Welzer, I. Rozman, M. Hericko, B. Brumen, T. Domanjko, and A. Zivkovic, "Performance Assessment Framework for Distributed Object Architectures", in *Proceedings of Advances in Databases and Information Systems*, LNCS, vol. 1691, pp. 349-366, Springer, 1999, http://lisa.uni-mb.si/~juric/PerfAssessmentFrw.pdf.

[94] JXTA, "Project JuXTApose", Web site, www.jxta.org.

[95] L. Kagal, J. L. Undercoffer, F. Perich, A. Joshi, T. Finin, and Y. Yesha, "Vigil: Providing Trust for Enhanced Security in Pervasive Systems", University of Maryland Techical Report, 2002, http://ebiquity.umbc.edu/v2.1/get/a/publication/13.pdf.

[96] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks", in *Proceedings of the International World Wide Web Conference*, 2003, http://www.stanford.edu/~sdkamvar/papers/eigentrust.pdf.

[97] K. Kant, "Introduction to computer system performance evaluation", ISBN 0-07-033586-9, McGraw-Hill, 1992.

[98] R. Kantola, et al., "Peer to Peer and SPAM in the Internet", Technical Report of the Helsinki University of Technology, 2004, http://www.netlab.hut.fi/opetus/s38030/F03/Report-p2p-spam-2003.pdf.

[99] S. Kent, "Privacy Enhanced Mail", IETF Working Group, 1996, http://www.ietf.org/html.charters/OLD/pem-charter.html.

[100] A. D. Keromytis, "The KeyNote Trust-Management System", Web site, 2004, http://www1.cs.columbia.edu/~angelos/keynote.html.

[101] R. Khare, "What's in a Name? Trust", 4K Associates, 1999, http://www.4k-associates.com/IEEE-L7-names-trust.html.

[102] M. Kinateder and K. Rothermel, "Architecture and Algorithms for a Distributed Reputation System", in *Proceedings of iTrust*, LNCS, Springer, 2003, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=1.

[103] M. P. Kinateder, Siani, "A Privacy-Enhanced Peer-to-Peer Reputation System", in *Proceedings of the International Conference on Electronic Commerce and Web Technologies*, 2003, ftp://ftp.informatik.uni-stuttgart.de/pub/library/ncstrl.ustuttgart_fi/INPROC-2003-18/INPROC-2003-18.pdf.

[104] J. Kleinberg, "Authoritative sources in a hyperlinked environment", in *the Journal of the ACM*, 46(5), ACM, 1999, http://doi.acm.org/10.1145/324133.324140.

[105] A. Kobsa and J. Schreck, "Privacy through Pseudonymity in User-Adaptive Systems", in *Transactions on Internet Technology*, vol. 3 (2), pp. 149-183, ACM, 2003, http://www.ics.uci.edu/~kobsa/papers/2003-TOIT-kobsa.pdf.

[106] J. Kohl and B. C. Neuman, "The Kerberos Network Authentication Service (Version 5)", Internet Request for Comments RFC-1510, 1993, ftp://ftp.isi.edu/in-notes/rfc1510.txt.

[107] S. Köpsell and S. Steinbrecher, "Modeling Unlinkability", in *Proceedings of the Third Workshop on Privacy Enhancing Technologies*, 2003, http://petworkshop.org/.

[108] F. Labalme and K. Burton, "Enhancing the Internet with Reputations", 2001, www.openprivacy.org/papers/200103-white.html.

[109] C. Lai, L. Gong, L. Koved, A. Nadalin, and R. Schemers, "User Authentication and Authorization in the Java(TM) Platform", in *Proceedings of the 15th Annual Computer Security Application Conference*, Phoenix, 1999, http://java.sun.com/security/jaas/doc/acsac.html.

[110] B. Lampson, "Protection", in *Proceedings of the 5th Annual Princeton Conference on Information Sciences and Systems*, pp. 437-443, Princeton University, 1971.

[111] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: theory and practice", in *Transactions on Computer Systems*, vol. 10(4), ACM, 1992, http://doi.acm.org/10.1145/138873.138874.

[112] M. Langheinrich, "A Privacy Awareness System for Ubiquitous Computing Environments", in *Proceedings of Ubicomp*, 2002, http://citeseer.nj.nec.com/517334.html.

[113] M. Langheinrich, "Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems", in *Proceedings of Ubicomp*, LNCS 2201, pp. 273-291, Springer, 2001, http://citeseer.ist.psu.edu/491722.html.

[114] La Rochefoucauld, "Réflexions", 1731, http://perso.wanadoo.fr/maliphane/La%20Rochefoucauld2.htm.

[115] S. Lederer, C. Beckmann, A. K. Dey, and J. Mankoff, "Managing Personal Information Disclosure in Ubiquitous Computing Environments", IRB-TR-03-015, Intel Research, 2003, http://www.intel-research.net/Publications/Berkeley/070920030922_139.pdf.

[116] L. Lessig, "The Architecture of Privacy", in *Proceedings of Taiwan Net Conference*, 1998, http://cyberlaw.stanford.edu/lessig/content/articles/works/architecture_priv.pdf.

[117] R. Levien, "Attack Resistant Trust Metrics", PhD Thesis, UC Berkeley, 2004, http://www.levien.com/thesis/compact.pdf.

[118] N. Li, "Local Names In SPKI/SDSI 2.0", in *Proceedings of the 13th Computer Security Foundations Workshop*, 2000, http://citeseer.nj.nec.com/390701.html.

[119] LibertyAlliance, "Liberty Alliance Project", Web site, http://www.projectliberty.org/.

[120] S. Lo Presti, M. Cusack, and C. Booth, "Trust Issues in Pervasive Environments", Trusted Software Agents and Services for Pervasive Information Environments project, 2003, http://www.trustedagents.co.uk/TSA-WP2-01v1.1.pdf.

[121] T. Mahler and T. Olsen, "Reputation Systems and Data Protection Law", *Proceedings of eChallenges*, 2004, http://www.afin.uio.no/forskning/notater/Reputation%20Systems%20and%20Data%20Protection%20Law.pdf.

[122] S. Marsh, "Formalising Trust as a Computational Concept", PhD Thesis, Department of Mathematics and Computer Science, University of Stirling, 1994, http://citeseer.nj.nec.com/marsh94formalising.html.

[123] A. H. Maslow, "Motivation and Personality", Harper, 1954.

[124] V. Matyas and Z. Riha, "Biometric Authentication – Security and Usability", Masaryk University Brno, Czech Republic.

[125] D. H. McKnight and N. L. Chervany, "What is trust? A Conceptual Analysis and an Interdisciplinary Model", in *Proceedings of the Americas Conference on Information Systems*, AIS, 2000, http://aisel.isworld.org/password.asp?Vpath=AMCIS/2000&PDFpath=162.pdf.

[126] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-Peer Computing", Technical Report, HPL-2002-57, Hewlett-Packard, 2002, http://citeseer.nj.nec.com/milojicic02peertopeer.html.

[127] N. Mogens, M. Carbone, and K. Krukow, "An Operational Model of Trust", SECURE Deliverable 1.2, 2004, http://secure.dsg.cs.tcd.ie.

[128] N. Mogens, M. Carbone, and K. Krukow, "Revised Computational Trust Model", SECURE Deliverable 1.3, 2004, http://secure.dsg.cs.tcd.ie.

[129] N. Mogens, G. Plotkin, and G. Winskel, "Petri nets, event structures and domains." in *Theoritical Computer Science*, vol. 13, pp. 85-108, 1981.

[130] G. Montenegro and C. Castelluccia, "Statistically unique and cryptographically verifiable(sucv) identifiers and addresses." in *Proceedings of the Network and Distributed System Security Symposium*, 2002, http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/monten.pdf.

[131] E. M. Noam, "Privacy and Self-Regulation: Markets for Electronic Privacy", 1997, http://www.ntia.doc.gov/reports/privacy/selfreg1.htm#1B.

[132] O. Olsson, "Privacy protection and trust models", in *ERCIM News*, vol. 49, 2002, http://www.ercim.org/publication/Ercim_News/enw49/.

[133] OpenPrivacy, "Sierra", Web site, 2001, http://sierra.openprivacy.org/.

[134] S. Pearson, "Trusted Agents that Enhance User Privacy by Self- Profiling", Technical Report, HPL-2002-196, Hewlett-Packard, 2002, http://www.hpl.hp.com/techreports/2002/HPL-2002-196.html.

[135] C. Perez, M. A. Vicente, C. Fernandez, O. Reinoso, and A. Gil, "Aplicacion de los diferentes espacios de color para deteccion y seguimiento de caras." in *Proceedings of Jornados de Automatica*, Universidad de Le/on, Universidad Miguel Hernandez, 2003.

[136] A. Pfitzmann and M. Köhntopp, "Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology", in *Open Paper for Discussion on the Information Hiding Workshop*, Springer, 2001, http://marit.koehntopp.de/pub/anon/Anon_Terminology_IHW.pdf.

[137] Postini, "Worldwide Map of Origin of Spam", 2004, http://www.postini.com/stats/.

[138] F. Rahman, "Learning to rely on others' opinion on your own", Presentation, Computer Lab (Com Sci Dept), University of Cambridge, 2003, http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/cambridge-trust-mgmt.ppt.

[139] F. Rahman, "Social Trust Survey", 2004, http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/thesis/soctrust.pdf.

[140] F. Rahman, "Trust in Computer Science", 2004, http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/thesis/csreview.pdf.

[141] M. K. Reiter and A. D. Rubin, "Anonymity Loves Company: Anonymous Web Transactions with Crowds", 1999, http://citeseer.nj.nec.com/reiter99anonymity.html.

[142] R. L. Rivest and B. Lampson, "SDSI - A Simple Distributed Security Infrastructure", 1996, http://theory.lcs.mit.edu/~rivest/sdsi11.html.

[143] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", in *Communications*, 21(2), pp. 120-126, ACM, 1978, http://citeseer.ist.psu.edu/rivest78method.html.

[144] P. Robinson, "Outcome of the Ubicomp'02 Workshop on Security in Ubiquitous Computing", in *Proceedings of the Ubicomp'02 Security Workshop*, 2002, http://www.teco.edu/%7Ephilip/ubicomp2002ws/organize/outcome.pdf.

[145] D. M. Romano, "The Nature of Trust: Conceptual and Operational Clarification", PhD Thesis, Louisiana State University, 2003, http://etd02.lnx390.lsu.edu/docs/available/etd-0130103-070613/.

[146] S/MIME, "S/MIME Mail Security (smime)", IETF Working Group, http://www.ietf.org/html.charters/smime-charter.html.

[147] J. H. Saltzer and M. D. Schroeder, "The Protection of Information in Computer Systems", IEEE, 1975, http://www.mediacity.com/~norm/CapTheory/ProtInf/.

[148] V. Samar and C. Lai, "Making Login Services Independent of Authentication Technologies", Sun Microsystems, 1995, http://java.sun.com/security/jaas/doc/pam.html.

[149] M. D. Schroeder and R. M. Needham, "Using encryption for authentication in large networks of computers", in *Communications of the ACM*, vol. 21(12), pp. 993-999, ACM, 1978, http://doi.acm.org/10.1145/359657.359659.

[150] F. Scott, "UML Distilled", Addison Wesley, 2000.

[151] SECURE, "Secure Environments for Collaboration among Ubiquitous Roaming Entities", Web site, http://secure.dsg.cs.tcd.ie.

[152] J.-M. Seigneur, J. Abendroth, and C. D. Jensen, "Bank Accounting and Ubiquitous Brokering of Trustos", in *Proceedigns of the 7th Cabernet Radicals Workshop*, 2002, http://citeseer.nj.nec.com/seigneur02bank.html.

[153] J.-M. Seigneur, G. Biegel, and C. Damsgaard Jensen, "P2p with JXTA-Java pipes", in *Proceedings of the 2nd International Conference on the Principles and Practice of Programming in Java*, ACM, 2003, http://portal.acm.org/citation.cfm?id=957350.

[154] J.-M. Seigneur, V. Cahill, C. D. Jensen, E. Gray, and Y. Chen, "The SECURE Framework Architecture (Beta)", Technical Report, 2004, Trinity College Dublin, http://www.cs.tcd.ie/publications/tech-reports/reports.04/TCD-CS-2004-07.pdf.

[155] J.-M. Seigneur, C. Damsgaard Jensen, S. Farrell, E. Gray, and Y. Chen, "Towards Security Auto-configuration for Smart Appliances", in *Proceedings of the Smart Objects Conference*, 2003, http://www.grenoble-soc.com/proceedings03/Pdf/45-Seigneur.pdf.

[156] J.-M. Seigneur, N. Dimmock, C. Bryce, and C. D. Jensen, "Combating Spam with TEA (Trustworthy Email Addresses)", in *Proceedings of the 2nd Conference on Privacy, Security and Trust*, Canada, 2004.

[157] J.-M. Seigneur, S. Farrell, C. D. Jensen, E. Gray, and Y. Chen, "End-to-end Trust Starts with Recognition", in *Proceedings of the First International Conference on Security in Pervasive Computing*, pp. 130-142, LNCS 2802, Springer-Verlag, 2003, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2802&spage=130.

[158] J.-M. Seigneur and A. Gray, "Default Free Introduction, Rare Self-Introduction Fee, Costly Spoofing: No Profitable Spam?" in *Proceedings of the EUROPRIX'04 Scholars Conference*, Tampere, Finland, 2004, http://www.mindtrek.org/konferenssit/scholarsconference/.

[159] J.-M. Seigneur, A. Gray, and C. D. Jensen, "Trust Transfer: Encouraging Self-Recommendations without Sybil Attack", in *Proceedings of the Third International Conference on Trust Management*, LNCS, Springer, 2005.

[160] J.-M. Seigneur and C. D. Jensen, "The Claim Tool Kit for Ad-hoc Recognition of Peer Entities", in *Journal of Science of Computer Programming*, Elsevier, 2004.

[161] J.-M. Seigneur and C. D. Jensen, "Privacy Recovery with Disposable Email Addresses", in *IEEE Security&Privacy*, vol. 1(6), pp. 35-39, 2003, http://www.computer.org/security/v1n6/j6sei.htm.

[162] J.-M. Seigneur and C. D. Jensen, "Trading Privacy for Trust", in *Proceedings of the Second International Conference on Trust Management*, LNCS 2995, Springer-Verlag, 2004, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=93.

[163] J.-M. Seigneur and C. D. Jensen, "Trust Enhanced Ubiquitous Payment without Too Much Privacy Loss", in *Proceedings of the Symposium on Applied Computing*, ACM, 2004, http://doi.acm.org/10.1145/967900.968218.

[164] J.-M. Seigneur, D. Solis, and F. Shevlin, "Ambient Intelligence through Image Retrieval", in *Proceedings of the 3rd International Conference on Image and Video Retrieval*, pp. 526-534, LNCS 3115, Springer-Verlag, 2004, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3115&spage=526.

[165] K. M. Self, "Challenge-Response Anti-Spam Systems Considered Harmful", Web site, 2004, http://kmself.home.netcom.com/Rants/challenge-response.html.

[166] SESAME, "A Secure European System for Applications in a Multi-vendor Environment", Web site, https://www.cosic.esat.kuleuven.ac.be/sesame/.

[167] G. Shakhnarovich, L. Lee, and T. Darrell, "Integrated Face and Gait Recognition From Multiple Views", in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE, 2001, http://csdl2.computer.org/dl/proceedings/cvpr/2001/1272/01/127210439.pdf.

[168] R. E. Smith, "Authentication: from passwords to public keys", ISBN 0-201-61599-1, Addison Wesley, 2001.

[169] M. Sonka, V. Hlavac, and R. Boyle, "Image Processing, Analysis, and Machine Vision", Second edition, PWS Publishing, 1999.

[170] S. Spiekermann, J. Grossklags, and B. Berendt, "E-privacy in 2nd Generation E-Commerce: Privacy Preferences versus actual Behavior", in *Proceedings of the Conference on Electronic Commerce*, ACM, 2001, http://citeseer.nj.nec.com/spiekermann01eprivacy.html.

[171] F. Stajano, "Security for Ubiquitous Computing", ISBN 0470844930, John Wiley & Sons, 2002.

[172] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", in *Proceedings of the International Security Protocols Workshop*, pp. 172-194, 1999, http://citeseer.nj.nec.com/stajano99resurrecting.html.

[173] Sun Microsystems, "Identity Grid", Whitepaper, Sun Microsystems, 2004, http://wwws.sun.com/software/products/identity/wp_identity_grid.pdf.

[174] TCPA, "Trusted Computing Platform Alliance", Web site, http://www.trustedcomputing.org/.

[175] B. Templetons, "Proper Principles for Challenge/Response Anti-spam Systems", Web site, 2004, http://www.templetons.com/brad/spam/challengeresponse.html.

[176] E. Terzi, Y. Zhong, B. Bhargava, Pankaj., and S. Madria, "An Algorithm for Building User-Role Profiles in a Trust Environment", in *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery*, Springer, 2003, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2454&spage=104.

[177] S. Terzis, W. Wagealla, C. English, A. McGettrick, and P. Nixon, "The SECURE Collaboration Model", SECURE Deliverables D2.1, D.2.2 and D2.3, 2004, http://secure.dsg.cs.tcd.ie.

[178] TLS, "Transport Layer Security", Web site, IETF, http://www.ietf.org/html.charters/tls-charter.html.

[179] J. Travers and S. Milgram, "An experimental study of the small world problem", in *Sociometry*, vol. 32, pp. 425-443, 1969.

[180] A. Twigg and N. Dimmock, "Attack-Resistance of Computational Trust Models", in *Proceedings of the International Workshop on Enabling Technologies*, IEEE, 2003, http://csdl.computer.org/comp/proceedings/wetice/2003/1963/00/19630275abs.htm.

[181] P. E. Verissimo, N. F. Neves, and M. P. Correia, "Intrusion-Tolerant Architectures: Concepts and Design", in *Architecting Dependable Systems*, R. Lemos, C. Gacek, A. Romanovsky (eds.), LNCS, Vol. 2677, Springer, 2003.

[182] R. Volker, "Weak Authentication", Web site, 2004, http://www.igd.fhg.de/~vroth/weakauth.html.

[183] W. Wagealla, M. Carbone, C. English, S. Terzis, and P. Nixon, "A Formal Model of Trust Lifecycle Management", in *Proceedings of the Workshop on Formal Aspects of Security and Trust*, 2003, http://www.iit.cnr.it/FAST2003/fast-proc-final.pdf.

[184] D. Watts, D. P. Sheridan., and M. E. J. Newman, "Identity and search in social networks", in *Science*, vol. 296, 2002, http://arxiv.org/PS_cache/cond-mat/pdf/0205/0205383.pdf.

[185] B. Wattson, "Beyond Identity: Addressing Problems that Persist in an Electronic Mail System with Reliable Sender Identification", in *Proceedings of the First Conference on Email and Anti-Spam*, 2004, http://www.ceas.cc/papers-2004/140.pdf.

[186] A. Weimerskirch and D. Westhoff, "Zero Common-Knowledge Authentication for Pervasive Networks", in *Proceedings of the Annual International Workshop on Selected Areas in Cryptography*, LNCS, Springer, 2004, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3006&spage=73.

[187] M. Weiser, "The Computer for the 21st Century", in *Scientific American*, 1991, http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html.

[188] M. Weiser and J. S. Brown, "Designing Calm Technology", in *PowerGrid Journal*, v. 1.01, 1996, http://powergrid.electricity.com/1.01.

[189] W. H. Winsborough, K. E. Seamons, and V. E. Jones, "Automated Trust Negotiation", in *DARPA Information Survivability Conference and Exposition*, 2000, http://citeseer.ist.psu.edu/winsborough00automated.html.

[190] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu, "Negotiating Trust on the Web", pp. 30-37, IEEE, 2002, http://csdl.computer.org/comp/mags/ic/2002/06/w6030abs.htm.

[191] E. Wobber, M. Abadi, M. Burrows, and B. Lampson, "Authentication in the Taos Operating System", ACM, 1994, http://doi.acm.org/10.1145/174613.174614.

[192] D. Work, "Call for A Social Networking Bill of Rights", in *PlanetWork Journal*, 2004, http://journal.planetwork.net/article.php?lab=work0704.

[193] R. Yahalom, B. Klein, and T. Beth, "Trust Relationships in Secure Systems - A Distributed Authentication Perspective", in *Proceedings of the Symposium on Security and Privacy*, pp. 150, IEEE, 1993, http://csdl.computer.org/comp/proceedings/sp/1993/3370/00/33700150abs.htm.

[194] C.-N. Ziegler and G. Lausen, "Spreading Activation Models for Trust Propagation", in *Proceedings of the International Conference on e-Technology, e-Commerce, and e-Service*, IEEE, 2004, http://www.informatik.uni-freiburg.de/~cziegler/Camera-Ready/EEE-04-CR.pdf.

[195] P. R. Zimmerman, "The Official PGP User's Guide", ISBN 0-262-74017-6, MIT Press, 1995.