Anonymous Communications in Wireless Access Networks

Alisdair McDiarmid

A thesis submitted for the degree of Doctor of Philosophy to
Institute for Communications and Signal Processing
University of Strathclyde

October 2006

Abstract

The growth of mobile communications systems over the past decade indicates a trend towards an always-on, ubiquitously networked society. This increase in communications availability leads to a corresponding increase in information gathered, processed, and transmitted over these networks. Some of this information is loosely considered by users of these systems to be private, and in some cases, even the general pattern of use of network services could be regarded as revealing personal information. Therefore, along with growth in communications comes growth in privacy concerns.

One approach to protecting users' privacy is to offer anonymity: the ability to blend into a crowd, such that any communications cannot be attributed to a particular real identity. This research study investigates two aspects of providing anonymity in mobile networks: in services, and for network access.

Anonymous mobile service provision is approached by analysing several fixed-network approaches to anonymous communications, and examining how they can be reapplied to mobile systems. A set of conclusions and recommendations for future implementations are contributed, along with a case study of providing anonymous location-based services for mobile systems.

Being able to connect to a network while remaining truly anonymous is a novel concept, only made possible by the untethered nature of mobile communications. Analysis of the practical requirements for achieving such service is presented, and a solution is proposed, based on a new approach to mobile network service provision called the Digital Marketplace. To support this approach to network access, the fair and reliable operation of the market is ensured by securing its protocol operation. Further modifications to this scheme are proposed, in order to enable fully anonymous network access.

Acknowledgements

I would like to thank several people, without whose support I would not have been able to complete this work: James Irvine, Victoria Catterson, John Bush, Colin Arthur, and Robert Atkinson.

Contents

List of Figures

List of Tables

List of Publications

[1] Alisdair McDiarmid and James Irvine. Anonymous Location-Based Services. *Proceedings of 60th IEEE Vehicular Technology Conference*, September 2004.

[2] Alisdair McDiarmid and James Irvine. Security Requirements for the Digital Marketplace. *Proceedings of 61st IEEE Vehicular Technology Conference*, May 2005.

[3] Alisdair McDiarmid and James Irvine. Securing the Digital Marketplace. *Proceedings of 62nd IEEE Vehicular Technology Conference*, September 2005.

[4] Alisdair McDiarmid and James Irvine. Anonymous Network Access using a Secure Digital Marketplace. In *Proceedings of Wireless World Research Forum Meeting 15*, December 2005.

[5] Alisdair McDiarmid and James Irvine. Commitment-Aware Reputation System for the Digital Marketplace. *IEE Electronics Letters*, 42(2):102–103, January 2006.

[6] Alisdair McDiarmid and James Irvine. Commitment-Aware Reputation System for the Digital Marketplace. *Proceedings of 63rd IEEE Vehicular Technology Conference*, May 2006.

[7] Alisdair McDiarmid and James Irvine. Anonymous Network Access using the Digital Marketplace. *to be published in Proceedings of 64th IEEE Vehicular Technology Conference*, September 2006.

Chapter 1

Introduction

Contents

Improvements in wireless communications technology over the last decade have led to huge growth in mobile network usage. Over two billion people worldwide own and use mobile phones, and the market is continuing to grow[3]. This near-ubiquity of cellular communications has opened a vast market for network services, which allow users to use handheld devices in new ways.

Another trend in the last decade is the increase in threats to privacy. Demographic data has become a highly valued commodity, and personal information is now being traded in ways previously unimagined. Whole industries are dedicated to collecting, analysing, and selling sensitive data that individuals once viewed as private. This is worryingly easy to do: many people are happy to exchange information like shopping habits, credit history, and lists of friends for discounts on products or services.

In a functional democracy, the government is allowed only the information about its citizens which is required to do its job; giving the government more power than it needs can lead to corruption, and an increasingly authoritarian state. In the modern era, large corporations wield so much lobbying leverage and financial resources that they are almost as powerful as governments. and so keeping personal information out of their hands is just as important.

Therefore, the consequences of privacy loss are extremely concerning, whether the privacy invasion comes from the state or a corporation. Never before has it been so vital to keep personal information private. At the same time, the increase in privacy-invading technologies makes this task extremely challenging.

Privacy invasion need not happen on a large scale for it to have a great impact. A particular type of fraud known as 'identity theft' can devastate an individual's life, leaving them in financial or legal trouble through no fault of their own. This occurs when a criminal uses false identification to impersonate an innocent member of the public; the consequences of any illegal actions they then take fall upon the victim instead.

Many organisations and individuals recognise these new privacy problems, and are working to raise public awareness. This in turn could lead to a new mobile services market: providing guaranteed privacy and anonymity to users who desire it. There are many technical challenges to overcome on the way to achieving true mobile network anonymity, and this thesis aims to identify and address these.

1.1 Privacy

As in all areas of security, achieving privacy is only possible with respect to a specific threat model. Threats to mobile communications systems security have traditionally been from third parties; that is, the network service provider is trusted, as is the user. This security model continued into much of the privacy research undertaken in the past few years, which leads to the situation that there is little work which does not trust the network service provider.

However, this assumption of trust is not necessarily justified. Even if users feel able to trust the corporation as a whole with private data, individuals within those corporations can be bribed or otherwise coerced into revealing sensitive information. Social engineering is an extremely powerful technique, which can involve using confidence tricks to convince key employees to compromise security measures. Some particularly privacy-aware consumers would prefer to invalidate such attacks, by removing the necessity to trust their network services provider.

A fresh approach to mobile communications security is needed to solve this growing problem. Starting with the assumption that the network service provider is not to be fully trusted can make it more difficult to achieve some goals, but it is also then possible to attain full privacy even if the provider is an attacker.

1.2 Anonymity

Pfitzmann defines anonymity as "the state of being not identifiable within a set of subjects, the anonymity set"[99]. This definition is often used as a starting point for anonymity research, and schemes are proposed to maintain the anonymity set against a series of attacks. The threats which these schemes defend against tend to be statistical analyses of data or traffic, used to reduce the anonymity set towards unity.

Defending against such attacks is extremely challenging. It is a never-ending task to keep up with new statistical attacks, which often bring in side information which was thought to be irrelevant by the system designers. However, in many situations, this is the only feasible solution: research into preserving anonymity against data mining attacks is one example of this[122].

Another approach is to set aside the concept of a set of possible subjects, and simply concentrate on unlinkability. If the attacker is unable to determine a relationship between two or more discrete events, it is impossible to build a user activity profile from observation. If this is the case, and the user never explicitly reveals his or her identity, anonymity is maintained. In network systems, methods for achieving this clearly require hiding of the address of the user, and sometimes more sophisticated defences against traffic analysis.

Whichever approach is taken, technology providing some form of anonymity for mobile communications systems could assist greatly in combating privacy invasion. This work presents security research leading towards this goal in two areas: mobile services, and mobile network access.

1.3 Outline

The remainder of this thesis is organised into eight chapters. Following this introduction, chapter 2 presents a background in the concepts of privacy and anonymity. A definition of the meaning of anonymity in the context of this study is given here.

Chapter 3 provides a basic background to relevant areas of cryptographic technologies. The core principles of secrecy and authenticity are covered, including both symmetric and asymmetric ciphers, message authentication codes, and digital signatures. An introduction to the field of certification and public-key infrastructure is also presented. Finally, the cryptographic techniques and security implications of digital cash are discussed.

Building on this, chapter 4 presents the state-of-the-art in anonymous communications protocols. The concept of a MIX network is explained, along with examples of current applications of this technology. Derivatives and alternatives, such as onion routing and crowds, are also presented. Following this is discussion of the challenges facing adoption of these protocols in a mobile environment. The final contribution in the chapter is a case study of protocol design for anonymous mobile communications, with location-based services as the application.

Chapter 5 introduces the concept of privacy-enhanced network access. Discussion of its increasing importance leads to examples of problems with current identified network access systems. An examination of current technologies which could be used for a limited form of anonymous access is presented, and from this a series of requirements for an improved scheme are derived. Following this, the Digital Marketplace (DMP) is introduced; a next-generation call management architecture, the DMP has the potential to be used as the basis for a privacy-enhanced network system. Discussion of the match between the requirements and the DMP as it stands leads to the conclusion that it is not currently suitable, but this could be altered.

One of the key obstacles to adoption of the DMP as an anonymous network access scheme is that it has not been securely designed. Therefore, chapter 6 contributes a comprehensive security analysis of the Digital Marketplace protocols. From this analysis, a definition of a secure Digital Marketplace is derived. Then, design changes and modifications to the protocol are presented which meet these requirements. To conclude the chapter, the newly-secured protocol is presented, along with an argument that all known security threats have been countered.

Another major issue with Digital Marketplace security is reputation. Chapter 7 examines the DMP-specific needs for a reputation system in detail. A novel reputation function is contributed, and compared with other relevant work in the field. The function is tested by simulation in a number of scenarios, and results are presented to demonstrate its appropriateness to the Digital Marketplace.

With the security and reliability of the Digital Marketplace dealt with, chapter 8 examines its application as a privacy-enhanced network access scheme. The issues which make the DMP immediately unsuitable for anonymity are presented and discussed. Building on the work of the previous two chapters, augmentations to the protocol to enable anonymity are contributed. A discussion of tertiary threats to anonymity follows, along with counter-measures for those threats.

The final chapter summarises the work presented in this thesis, and reemphasises the importance of anonymity in mobile communications systems. The major achievements and contributions from this study are identified, and an outline of possible future work is given.

Chapter 2

Privacy and Anonymity

Contents

2.1 Mobile Communications and Privacy

Recent enhancements in mobile network technology has led to almost ubiquitous communications availability, and accordingly high usage levels. In 2005, 80% of UK adults owned mobile phones, and this figure is growing every year[94]. The ability to keep in touch at any time has great benefits, both for personal communications and for business. However, the same technology can be misused to invade personal privacy.

As the general population spends more and more time in communication, there is a corresponding increase in personal information transferred across networks. This privacy issue may go unnoticed by many users; the information released is perhaps not obviously private. For example, a user switching on a mobile phone allows their network provider to find their identity, and record their location along with the time of day. This in itself is perhaps not highly private information, but the ability of the provider to record the movement of the user throughout the day is more invasive.

Privacy invasions do not only come from network service providers. With the availability of 2.5G and 3G data services, mobile communications is moving away from circuit-switched communications towards packet-based data transfer. This gives the mobile phone the ability to directly access Internet services, and even receive information pushed to the device as appropriate, which has great potential for new mobile applications. At the same time, accessing these services requires the user to reveal their network address, which may be used by Internet providers to find the user's network provider, location, and potentially other identifying information.

2.1.1 Demand

A 2004 study by the UK Information Commisioner's Office reports that 70% of respondents indicated high or very high concern for protecting people's personal information[62]. Privacy was rated as more important than equal rights, freedom of speech, the environment, and unemployment; this indicates that it is not a fringe issue.

Similarly rapid growth in usage of the Internet for electronic commerce had a similar effect on interest in privacy, as is shown by several Internet-related privacy surveys. The 2001 UCLA Internet Report showed that 94.5% of surveyed respondents are concerned about privacy of their personal information when participating in online commerce[25]. Furthermore, 77% of world-wide web users responding to a 1998 Georgia Tech survey indicated that they value privacy more than convenience[56]. The importance of private communications is also shown by a 2001 Culnan-Milne report, which showed that 64% of respondents chose not to purchase online because of privacy concerns[29].

These data show that the level of public interest in privacy is high. Awareness of personal information issues in mobile communications is uncertain, but with this level of concern for privacy in general, it is fair to assume that mobile privacy issues would garner the same interest if well-understood. Therefore, technological solutions to offer privacy to those who wish it would be of benefit to a significant proportion of the communicating population.

2.1.2 Summary

The aim of this chapter is to provide an understanding of what privacy protection and anonymity in communications systems means. To achieve this, related material in the area of privacy protection and anonymity provision is presented and summarised. Several properties of anonymity are shown, and an approach to classifying level of anonymity is discussed.

2.2 Privacy Protection

Control of personal information is a fundamental concept which is the basis of much privacy research. With current communications systems, the end-user often has little or no control of much of this information; instead, the network or service provider is trusted not to act against the interests of the user. A common theme in privacy research is proposals which allow the user to express privacy preferences as a policy, which the network or service provider must then implement.

These policies must be written in a standardised way, to allow unlimited interchange of privacy-controlling requirements. For example, the W3C's Platform for Privacy Preferences (P3P)[27] allows web sites and web services providers to specify their privacy practices in an XML format. These privacy policies can then be interpreted by users' browsers or web service clients, and appropriate action taken depending on the user's locally-specified privacy preferences.

Gunter et al. give an example of the application of P3P to mobile systems, for privacy of location data[58]. This work combines P3P into a digital rights language to formalise users' privacy demands into contracts, which are agreed by the service providers before service usage. This language allows users to specify what the providers may do with their data, how long it may be retained, and the consequences for non-compliance.

Agrawal et al. proposed an implementation of P3P which is more suitable for mobile systems than the commonly-used client-based filtering[1]. This proposal uses a database server to process privacy policies on behalf of users. Among other benefits, this reduces the processing requirement on the client device, which is of prime concern for mobile systems with low-powered handsets.

One important problem with the policy approach is the trust requirement. Users must trust that the service providers both accurately represent their privacy policies, and ensure compliance to those policies by their employees. This trust is extended still further for centralised policy checks, where the user must also trust the policy database provider to operate fairly. In practice, policy-based approaches are only useful when all participants in the system are generally trustworthy, and the user has a mild preference for how their information is processed. When privacy is more important, and other parties are not so trusted, other approaches should be taken.

2.3 Anonymity

In everyday life, anonymity is generally thought of as the state in which a person cannot be identified by name. In communications, the term is more nuanced than this, and the person's name is rarely of immediate relevance.

Within the field of privacy, anonymity provision is concerned with ***information which could be used to identify the user.*** Research in the field has taken a number of different approaches to controlling such information, and there are several different properties of anonymity to achieve. These were defined by Pfitzmann and Kohntopp's terminology paper[99], and are discussed below.

These properties of anonymity lead to a more formal concept of degree of anonymity. This was proposed by Reiter and Rubin in their work on the Crowds anonymity scheme[104]. Degree of anonymity is described in section 2.3.5.

2.3.1 Anonymity Set

Formal classifications of anonymity normally do not claim that the user is completely unidentifiable. Instead, they restrict the attacker to being able to claim that a user is a member of a given *anonymity set*. If known, the size of this set can be used as an estimate of how anonymous a user is, for a given situation. In some situations, the size of the set is not clearly defined: an anonymity scheme which only depends on the total number of users n can be said to have an anonymity set of size n.

The idea of the anonymity set was used by Sweeney to provide a particular level of privacy for data-rich reports[122]. By programmtically restricting the scope of the information retrieved, a set of k indistinguishable records is created, thus providing anonymity for the people described by those records. Similar techniques can be applied to high-latency communications systems; see the description of MIX networks in section 4.1 for an example.

2.3.2 Pseudonymity

One simple form of hiding identity is to use a false name, or *pseudonym*. If every communicant has a pseudonym for identification, then communication remains possible, and real identities of the communicants are not necessarily revealed.

This is a very straightforward method of achieving some level of anonymity. Many protocols already use some form of pseudonym to address users: for example, email addresses are not necessarily directly identifying. So, for email pseudonyms, all that is required is an address which reveals nothing about the identity of the user. For example, this could be a random number at a fixed public domain, like `73961629@anonymail.com`. The address itself does not contain the name of the user; it only has meaning once an association is somehow made with the real email recipient.

The main problem with using a fixed pseudonym for anonymity is that it is often possible to build up a usage profile based on this false identity. In itself, this can reveal identifying information about the user: for example, the contents of an email message could contain the user's name; or, the user could regularly receive email from a set of contacts which give information about who the sender is likely to be. Even if this profiling does not immediately reveal the identity of the user, other possibly-sensitive information could be revealed and stored; then, if the identity associated with a pseudonym is later found, all of this information can be tied to that person.

2.3.3 Unlinkability

To prevent profiling attacks, detectable links between subsequent uses of a communication system must be removed. This property is called *unlinkability*, and provides a higher level of privacy protection than pseudonyms alone.

If a user's messages are unlinkable, then an attacker cannot distinguish between two messages from the same user, and two messages from different users. This is analogous to using a different pseudonym for each message; if the pseudonym is fully anonymous, so that there is no way to determine who sent an individual message, anonymity is achieved even against statistical attacks. This is a much more useful result, as it allows users to send or receive several messages without undue risk of identification.

2.3.4 Unobservability

A final, and less practical property of anonymity, is one which allows users to send or receive messages without detection: *unobservability*. This means that an observer is not only unable to determine who sent or received a message, but also unable to determine that a message was sent at all.

Chaum published a protocol which achieves this in his Dining Cryptographers paper[32]; the resulting scheme is known as DC-Net. Clearly, this provides even more privacy than a successful unlinkable communications system. However, the protocol is not particularly practical. It is highly inefficient, requiring constant transmission of data even when none needs to be sent. It also assumes the availability of a reliable broadcast network, which does not match well either with the fixed Internet or with cellular mobile systems. For these reasons, and because no significantly more efficient methods have been found, unobservability is generally not a feature of pragmatic privacy schemes.

2.3.5 Degree of Anonymity

The above three properties of anonymity have been presented on a scale. Pseudonymity offers some identity protection, but is susceptible to profiling over multiple messages. Unlinkability counters this by removing links between several sessions, removing the ability to correlate between them. Finally, unobservability ensures that the communications session itself cannot even be detected.

Reiter and Rubin's scale takes a related but slightly different approach. The degree of anonymity[104] is a measure of the anonymity offered by a given scheme, against a given attack. It ranges from absolute privacy, where communication is undetectable, to being provably exposed, where the attacker can demonstrate the identity of the user to others. The full scale is shown in Figure 2.1.

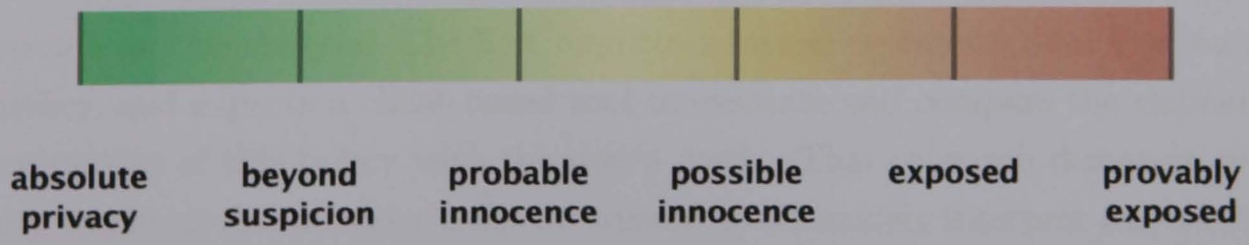


Figure 2.1: Reiter and Rubin's *degree of anonymity* scale

The points at the extremes of the access are of less interest than those towards the centre. Absolute privacy is not practical to obtain in most circumstances, as it requires that the anonymity scheme provide full unobservability; for reasons discussed above, this is difficult to achieve. Similarly, an anonymity scheme which allows its users to be exposed (that is, identified), or even provably so, has failed.

Therefore, *beyond suspicion* is the best achievable level of anonymity. Consider a scheme which provides anonymity to the sender of messages. Then, the sender of a given message is beyond suspicion if all possible senders appear to be equally likely to have sent it.

Further along the scale, *probable innocence* is a state of lesser anonymity. For the same example, this means that the sender of a message appears to be equally likely to have sent or not sent the message. Clearly this will generally create a reduced anonymity set compared with being beyond suspicion, but the sender cannot be detected as being the most likely sender of the message.

Further still, *possible innocence* could also be termed 'plausible deniability'. This means that, from the perspective of the attacker, there is a reasonable chance that a participant other than the real sender could have actually sent the message.

This scale can be a useful tool for measuring the properties of anonymity schemes against various attacker models. The Crowds paper uses it immediately to show the degree of anonymity provided by the scheme, considering a variety of anonymity properties and opponent models.

2.4 Summary

Two approaches to privacy protection for communications systems are presented by this chapter. The first requires a formal representation of privacy policy, and expects a client-based tool to evaluate and compare the claimed properties of this policy with the user's needs. This approach demands every single service provider must be trusted to accurately interpret the user's privacy policy, and strictly adhere to the rules contained therein.

Another option, which may be used in conjunction with a policy approach, is to provide some level of anonymity to users. At one extreme, this can mean merely providing a pseudonym which is unconnected to the user's real identity; at the other, the scheme could hide the fact that communication is taking place at all. Between these extremes is a more common goal: a scheme which ensures that the identity of the user is hidden, and also that there are no detectable links between messages which enable correlation.

Finally, the degree of anonymity measure is presented. This is a scale of identity protection, ranging between fully hidden communications and having identity completely and provably exposed. This gives a series of reference points against which some anonymity schemes can be compared.

Chapter 3

Cryptographic Background

Contents

Communications systems depend on cryptography to provide security. The field of cryptography is very broad, and there are thousands of protocols, schemes, and algorithms which aim to solve many different problems.

This chapter introduces some fundamentals of cryptography, selecting interesting ideas from the field which are relevant to this thesis. The areas of cryptography covered include secrecy, authenticity, certification, and digital cash.

3.1 Secrecy

In secure communications, secrecy is the ability to hide information from third parties. This is the most well-known area of cryptography, and it has been explored for hundreds of years[67]. The process of altering information so that it is obscured except to the intended recipient is known as *encryption*; the reverse process is *decryption*; and an algorithm which performs these tasks is known as a *cipher*. The input to a cipher is known as the *plaintext*, and the output is the *ciphertext*.

In the late 19th century, Kerckhoffs listed six desirable properties of a cipher[68], many of which still apply today. The most well-known of these properties is often restated as: "the security of the cipher must not be compromised by the publication of the algorithm". Therefore, the secrecy provided by the cipher must come from another source: this is a parameter called the *key*.

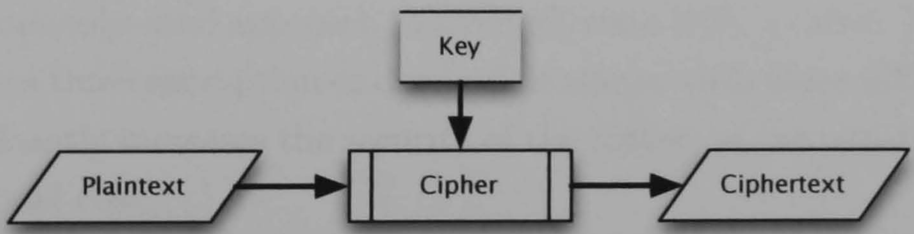


Figure 3.1: Flow chart of cipher operation

Therefore, a generic cipher algorithm has two secret inputs and one publishable output (see Figure 3.1). There are two main classes of cipher: symmetric (also known as secret-key) and asymmetric (or public-key), each of which are discussed in further detail below.

3.1.1 Symmetric Cryptography

Symmetric ciphers require a *secret key* to be shared between the sender and the recipient. The key is combined with the plaintext by the cipher to give the ciphertext; this is done in such a way that it should be unfeasible for anyone without the key to perform the reverse operation.

Symmetric ciphers can be broken into two groups: *stream ciphers*, which process variable-length data by encrypting a byte or bit at a time; and *block ciphers*, which work with a number of fixed-length groups of bits. This thesis is concerned only with block ciphers, which are discussed in more detail below.

3.1.1.1 Block Ciphers

The first important block cipher was the Data Encryption Standard[85]. DES is a modified form of Lucifer, which was created at IBM by a group of researchers led by Horst Feistel. The United States National Security Agency suggested the modifications, which were later found to protect against a sophisicated form of cryptanalysis. However, due to its key size of 56 bits, DES is now obsolete. In 1999, a distributed effort used brute force to break a DES key in 22 hours and 15 minutes, clearly demonstrating the obsolesence of the industry standard cipher.

One commonly-used approach to strengthening DES is called Triple DES. This involves three encryption or decryption stages, with three different keys. This significantly increases the security of the cipher, at the expense of extra computational cost.

The official replacement for DES is an algorithm submitted to the standards institute as Rijndael, and now known as the Advanced Encryption Standard (AES)[90]. It is expected that this cipher will be secure for at least several more decades, due in part to its variable key size of up to 256 bits. At current estimates of processing power, a brute force attack against a 256-bit key would take hundreds of trillions of years to complete. In new systems, AES is the preferred cipher for symmetric cryptography, due to its balance of robustness against cryptanalysis with hardware and software implementation efficiency.

3.1.1.2 Block Cipher Modes

Block ciphers are defined as operating on fixed-length blocks of bits. When encrypting a message which is not exactly one block in length, the block cipher must be used as part of another encryption function. These functions are known as block cipher modes. Some commonly-discussed cipher modes are Electronic Codebook (ECB), Counter (CTR), and Cipher Block Chaining (CBC)[41].

Each of these modes requires padding of the input data until its length is an exact multiple of the block size. There are several secure padding schemes for block ciphers, many of which are extremely simple. One common method is to count the number of bytes to the next block boundary (n), then append n bytes of value n to the plaintext[107].

ECB is the simplest possible block cipher mode. The plaintext is padded, split into blocks, and each block is encrypted separately. In practice, ECB is never used to encrypt multiple blocks: because identical plaintext blocks encrypt to identical ciphertext blocks, it does not hide large-scale data patterns.

Both CTR and CBC take a third input to the cipher: a block-sized, message-specific number known as an initialisation vector (IV) or nonce (a contraction of 'number used once'). As the name indicates, the number must only be used once with the key and cipher, but it need not be kept secret.

CBC combines the nonce with the plaintext using XOR, then encrypts the result. This block is output as ciphertext, and used instead of the nonce in the next block: each subsequent block of plaintext is XORed with the previous block of ciphertext.

In CTR mode, the nonce is encrypted with the cipher, and the output is XORed with the plaintext to generate the ciphertext. The nonce is incremented after each encryption step, so two identical plaintext blocks will not encrypt to the same ciphertext.

Each mode has advantages and disadvantages. CTR encryption can be parallelised to improve performance; clearly, in CBC mode the blocks must be encrypted in order. Another advantage of CTR is lower memory usage when used as a stream cipher: it can process a plaintext block into a ciphertext block, send the ciphertext, then reuse the memory allocated for it. In comparison, CBC mode clearly must keep the previous ciphertext block around to continue encryption. However, if a nonce value is ever used twice, CTR mode can reveal some information about the entire message, whereas CBC only leaks information about the first block. Therefore, the most appropriate mode depends on the threat model of the security system.

3.1.2 Asymmetric Cryptography

The most difficult problem in cryptography has traditionally been secret-sharing. Defeating the best ciphers available has always been difficult: the Vigenère cipher remained unbroken for nearly 300 years; DES, while now obsolete, can still only be broken using brute force; Shannon presented a 'one-time pad' method of encryption which is theoretically impossible to break. However, every symmetric cipher requires the secure transfer of some secret—the keyword, the 56-bit key, or the random pad—and it is this stage which is often most difficult. Worse, the problem is exacerbated as the number of communicators increases.

A group of 10 people wishing to securely communicate with each other, pair-wise, needs a total of $\sum_{i=1}^{9} i = 45$ keys. This number might be manageable, but it grows quadratically: a group of 100 people needs 4950 keys; a group of 10,000 people needs 49,995,000 keys. Before communication can take place, each of these keys must be agreed between the participants via some secure channel; for example, by meeting in person. This is obviously impractical with a large number of participants, especially for long-distance communications.

3.1.2.1 Diffie-Hellman Key Exchange

In 1976, Diffie and Hellman proposed a key-exchange protocol which allowed the secure agreement of a key over an insecure channel[36]. First, two participants (traditionally called Alice and Bob) agree on a public key. This consists of a large prime modulus n, and a generator[1] g. These values can be shared among many participants, and can be agreed over the insecure channel.

Then, Alice chooses some large number x, and sends Bob $X = g^x \mod n$. In turn, Bob chooses some large number y, and sends Alice $Y = g^y \mod n$. Then, Alice computes $k = Y^x \mod n$, and Bob computes $k' = X^y \mod n$. Both k and k' are now equal to $g^{xy} \mod n$, and so the shared key k is reached. Note that an eavesdropper on the insecure channel cannot calculate k without knowing one of x and y, which Alice and Bob keep secret at all times. Diffie-Hellman key exchange is secure unless g^{xy} can be computed given g, X, and Y; solving this computation efficiently is known as the Diffie-Hellman problem. The most efficient currently-known approach is to solve the discrete logarithm problem; calculating x given g^x.

This concept is now known as public-key or asymmetric cryptography, due to the use of a public key with an asymmetric algorithm. The asymmetry in the algorithm demands the use of a one-way function E, such that given only a *public key* it is trivial to compute $E(x)$, but extremely difficult to reverse this process $E^{-1}(E(x))$. To be a practical cipher, the one-way function must also have a 'trapdoor', which (given a *private key*) enables the reverse process $E^{-1}(E(x))$. The public and private keys are known as a *key pair*; the public key should be disseminated as widely as possible, and the private key kept absolutely secret. There are many asymmetric ciphers; two of direct relevance to this thesis, RSA and ElGamal, are presented below.

[1] A generator is a number such that, for every i from 1 to $n-1$, there exists some j such that $g^j = i \mod n$.

3.1.2.2 RSA

In 1978, Rivest, Shamir, and Adleman were first to publish a public-key cipher algorithm[106]. Named after the initials of its inventors, RSA is now the most widely used public-key algorithm in the world. Its popularity is in part due to being released first: the three letters RSA have become synonymous with public-key cryptography. The algorithm has been studied extensively, and has not yet been broken.

An RSA public key consists of n and e; the private key consists of n and d. n is a product of two primes p and q, which must also remain secret; e is a number relatively prime to $\phi = (p-1)(q-1)$. d is calculated as $e^{-1} \mod \phi$, which means that d and n are also relatively prime.

Given the public key data, n and e, and a message m, it is trivial to compute the ciphertext $c = m^e \mod n$, but extremely difficult to compute m from c. However, if d is known, the inverse computation becomes easy: $m = c^d \mod n$. In RSA, this property is used as the trapdoor back through the one-way function.

Therefore, RSA's security is dependent on the difficulty of factoring large composite numbers. The variable element of the public key, n, is the multiple of two large primes; if n is factored, the algorithm is broken. Factoring algorithms have improved dramatically over the last 25 years: a 256-bit public key can be factored in a few hours on a desktop PC; a 512-bit key was broken by a distributed effort of several hundred PCs in 1999. A theoretical device described by Shamir and Tromer in 2003[113] would be capable of factoring 1024-bit numbers in less than a year, at a cost of a few dozen million US dollars. At time of writing, the current recommendation for RSA key size is 2048 bits.

3.1.2.3 ElGamal

ElGamal published his eponymous public-key cipher in 1984, as an alternative to RSA[42][43]. Unlike RSA, it is unpatented, although it was arguably covered by the Diffie-Hellman patent. This expired three years before the RSA patent, which for a brief period made ElGamal the first generally useful public-key algorithm which was unencumbered by patents in the US. While its popularity is still far less than RSA, there are some novel properties which make it interesting in some use-cases.

An ElGamal public key consists of p, g, and y; the private key consists of p and x. p is a chosen large prime, and g and x are two random numbers both less than p; finally, $y = g^x \mod p$. Both g and p can be reused; only x (and therefore y) must be different for each user to maintain secrecy.

ElGamal encryption requires another parameter, k, a random number which is relatively prime to $p - 1$. Then, to encrypt message m, calculate $a = g^k \mod p$ and $b = y^k m \mod p$ to find the ciphertext $c = a, b$[2]. Note that this ciphertext is twice the size of the message. Decryption is simply $b/a^x \mod p$, because $a^x = g^{kx} \mod p$, and $b/a^x = y^k m/a^x = g^{xk} m/g^{kx} = m \pmod{p}$.

Clearly, the decryption process requires that x must be known; calculating x from y, g, and p requires calculating the discrete logarithm of $y \mod p$. The complexity of this calculation is equivalent to factoring an integer n of the same size as p, where n is the product of two primes of roughly equal length[72].

One property of ElGamal which makes it particularly useful is due to its use of finite fields. To encrypt a message to multiple recipients, their public keys $y_0, y_1, \ldots, y_n$ can be multiplied to create a group key, y_N. Then, this key can be used in the normal process of ElGamal encryption, and the ciphertext c must be decrypted by every recipient in order to find the message. See section 4.1.2.1 for an example use of this quirk of the algorithm.

3.1.2.4 Practical Implementations

Both of these asymmetric encryption algorithms have some nuances which must be taken into account when applying them in practice. Most important is a class of problems with small values of m, which reduces the security of both RSA and ElGamal[16]. For this reason, small messages must be securely padded up to the size of the modules (n for RSA, or p for ElGamal). Random

[2] The , operator indicates concatenation of the two surrounding terms

secure padding is required to ensure that the same plaintext encrypts to a different ciphertext each time it is encrypted.

Still, the public-key algorithms described above are not normally used to encrypt data, for two reasons. First of all, the algorithm can only be used to encrypt a message which is at most the equal to the modulus: n for RSA, and p for ElGamal. Messages significantly smaller than this value must be padded, as described above; and larger messages must be broken up into segments no larger than the modulus. Secondly, public-key encryption algorithms are significantly slower than their symmetric counterparts. This is particularly noticeable with large messages, as each segment requires a completely new run of the encryption algorithm.

One solution to this is to use both asymmetric and symmetric algorithms together, to create 'hybrid' cryptography, also known as a digital envelope. The asymmetric cipher is used only to encrypt the randomly-generated symmetric cipher key; this key is likely to be smaller than the asymmetric cipher's modulus, so encryption is as fast as possible. Then, the symmetric cipher is used to encrypt the message, and the two ciphertexts are combined.

The de-facto standard method for hybrid encryption is published by RSA Labs as PKCS #1[108]. This was later built upon by a group led by Victor Shoup, in order to create an ISO standard for public-key encryption[64]. ISO 18033 defines the two halves of hybrid cryptography—key encapsulation and data encapsulation—and provides limited security proofs for several schemes for each. Of the schemes for key encapsulation, RSA-KEM has the most efficient proof, and is therefore regarded as probably being most secure.

3.1.2.5 Forward Secrecy

When using asymmetric or hybrid cryptography, the private key must clearly be kept secret. However, under realistic threat models of many situations, there is the possibility that the private key could be found by an attacker; for example, a laptop could be stolen, or some form of virus or other malware loaded onto the system. In this case, with the standard digital envelope, all previously-transmitted messages could be decrypted by the attacker.

The ability of the cipher to cope with this situation is known *perfect forward secrecy*, or more simply *forward secrecy*[3]. This can be achieved by using a key agreement scheme, such as Diffie-Hellman (see section 3.1.2.1), and regularly changing keys and destroying all copies of the previous key. In this case, even if the current key is revealed, all previously-recorded messages can no longer be decrypted. One example of such a scheme is Off-The-Record messaging[17], which uses Diffie-Hellman, AES, and HMAC to provide forward secrecy and message repudiation for casual online conversation, such as instant messaging and email.

3.2 Message Authentication

One aspect of secure communication which is often overlooked is message authentication. Message authentication provides two important capabilities: determining who sent a particular message, and detection of in-transit modification. While it may seem at first counter-intuitive, secrecy is often less important than message authentication. If an attacker controls a communications channel, the ability to modify message contents without detection can be far more dangerous than merely reading their contents.

Message authentication is achieved in different ways for symmetric and asymmetric ciphers. When a symmetric shared key is used, message authentication is normally given using a Message Authentication Code (MAC), which is calculated and transmitted along with the message. The MAC is dependent on the algorithm chosen, the message, and the shared key.

This allows each communicant to verify that a message was not modified in transit. The recipient can take the message, and apply the MAC algorithm with the shared key. This value can then be compared with the transmitted MAC: if the two are equal, the message has not been modified in transit.

There is an important difference between the type of authentication offered by MACs and digital signatures. In a MAC scheme, any participant

[3] Perfect secrecy is the property of a cipher that the ciphertext reveals no information about the plaintext; this does not apply to all systems which provide forward secrecy, so the simpler term is more accurate.

with the shared key could have created or modified the message. With digital signatures, the message can be traced directly to the holder of a private key: since these are not shared, this generally means to one individual.

Digital signature schemes and some MACs require a message digest function, more commonly known as a hash. Hashes, MACs, and digital signatures are all described in more detail below.

3.2.1 Hash Functions

A hash function reduces an arbitrarily-long message m to a fixed-size output $h(m)$, known as the message hash, digest, or fingerprint. A hash must have three properties to be considered cryptographically secure:

One-way Given $h(m)$, it must be hard to find m

Second pre-image resistance Given m, it must be hard to find another message m' such that $h(m) = h(m')$

Collision resistance It must be hard to find two messages m and m' such that $h(m) = h(m')$

An ideal hash function with output of length n bits requires 2^n work for a second pre-image attack (where m is fixed), and $2^{n/2}$ work for a collision (where m is variable). Any function which requires significantly less work than this is considered flawed and should not be used for cryptography.

The NIST standard hash function family is known as the Secure Hash Algorithm (SHA)[87]. The first member of the family, now known as SHA-0, was found to be flawed by Chabaud and Joux in 1997[20]; collisions can be found with complexity 2^{61}. In 2005, real collisions in SHA-0 were found by Wang et al. with only 2^{39} hash operations[128]. SHA-0 was made obsolete in 1995 by SHA-1[88]; however, SHA-1 has been found subject to similar attacks by Wang et al.[129], with expected 2^{69} operations.

Three more variants have been published since SHA-1, all with increased output size compared to the original 160 bits: SHA-256, SHA-384, and SHA-512, collectively known as SHA-2[91]. The flaws found in SHA-1 have not yet been found in SHA-2, and the increased digest size should reduce the practicality of any such attack. As there are currently no clearly-superior alternatives to SHA available, the current NIST recommendation is to migrate to SHA-2 for now, and continue work on a new hash standard[18].

3.2.2 Message Authentication Codes

A message authentication code (MAC) takes two parameters: the message to be authenticated, and an authentication key. The output is a tag which can be used to demonstrate that the message was not altered in transit, and that the creator of the message had the authentication key. The same authentication key is used to verify the MAC, so third-party verification is not possible.

There are many MAC schemes in the literature. Below are brief discussions of CBC-MAC, HMAC, UMAC, and Poly1305-AES. CBC-MAC is an early standard MAC algorithm, while HMAC is another widely-deployed scheme; the more recent UMAC and Poly1305-AES compete for highest speed. The choice of MAC can be made based on the availability of cryptographic primitives: if a block cipher is available, use a CBC-MAC or Poly1305; if a hash function is available, use HMAC or UMAC.

3.2.2.1 CBC-MAC

CBC-MAC is a class of MAC, and can be based on any block cipher; for example, DES-CBC-MAC is a widely-used NIST standard which uses DES[86]. The principle is simple: encrypt the message using the block cipher and the authentication key, in CBC mode, and use the final block as the authentication tag. It has been proved as secure as the underlying cipher, but only for messages of fixed-length[9]. For messages with variable length, it is well-known that CBC-MAC is not secure. Black and Rogaway corrected this, with several schemes using multiple keys to ensure that messages of any length can be authenticated securely[15].

3.2.2.2 HMAC

HMAC is also a generic MAC, and can use any cryptographically-secure hash function[8]; for example, HMAC-SHA-256 uses SHA-256. The method for computing the tag is as follows. First, the key is XORed against a constant with length equal to the hash function block size: this is known as padding, because the result has fixed size. The padded key has the message appended, and the joined result is then hashed. Then, the key is XORed and padded again, with a different constant. This second modified key has the output of the first hash appended, and the joined result is hashed.

This double-hashing means that HMAC can be slow, particularly noticeably for small messages; for larger messages, the execution time of the first hash dwarfs that of the second, as its input is so much larger. The overall performance is dictated by the hash function; HMAC-MD5 is much faster than HMAC-SHA-256, but not as resistant to attack. HMAC can be used with messages of any length, and the security analysis in [8] shows that it is as secure as the underlying hash function.

3.2.2.3 UMAC and Poly1305-AES

Another hash-based MAC was proposed by Black et al. in 2000, which aimed to address the slow performance of HMAC[14]. The scheme is complex, but was shown to be an order of magnitude faster than HMAC, for large messages. This improvement comes from the use of a very fast hash function 'NH', which is used to shorten the input message; UMAC still requires the use of a cryptographic hash to provide a pseudo-random mapping between the output of NH and the tag. The paper provides a security reduction of the NH function which demonstrates that UMAC is as secure as the cryptographic hash employed.

UMAC has disadvantages, however. Every message requires a large nonce, and there are significant memory requirements to achieve the increase in throughput. Furthermore, UMAC has not been widely deployed nor extensively studied; HMAC's greater popularity and visibility implies that it is the safe choice when speed is not so important.

Another high-performance hashing scheme was published by Bernstein in 2005[12]. This scheme uses AES, although it could be used with other algorithms, and is shown to be as secure as the underlying cipher. Its speed is comparable to UMAC, but it uses less memory. This means that it scales better with a large number of keys, leading to higher throughput with a realistic network load.

3.2.3 Digital Signatures

Some asymmetric cryptosystems are used to provide digital signatures; there are even some which are designed for signatures only. A signature is calculated and transmitted along with the message, similarly to a MAC tag. However, digital signatures have several properties which MACs cannot provide.

Most importantly, the digital signature identifies exactly who signed the message. Only the owner of the private key could have signed it; in a normal environment, only one entity should have this private key, and so the signature can be used for identification.

This property also means that digital signatures can be used for non-repudiation. Once a message has been signed, the signer cannot feasibly make the claim that someone else signed it. The only way this could have happened is if the private key was lost before the message was signed. Even then, there are online non-repudiation protocols which handle such situations; a summary of the field is given by Kremer et al.[70]

Of course, digital signatures also provide message integrity. A signed message cannot be modified without detection, or the signature would be worthless.

There are numerous schemes which can create digital signatures. Both RSA and ElGamal can be used to sign messages as well as encrypt, and a third algorithm called DSA was specifically designed as a signature-only algorithm. In practice, RSA and DSA are most commonly used, and those signature schemes are discussed here.

3.2.3.1 RSA

To understand how RSA signatures work, note that the RSA algorithm is commutative (with one key pair). Define RSA encryption of a message m with key pair k as $E_k(m)$, and corresponding decryption as $D_k(m)$. Then, $D_k(E_k(m)) = E_k(D_k(m))$; that is, encryption and decryption can be performed in either order.

Clearly, only the holder of the private key can decrypt; therefore, that is the signing operation. Anyone with the signer's public key can verify the signature by encrypting it with the public key, and comparing the result with the purported message. The signature is valid if and only if the two are identical.

However, RSA signatures are not calculated quite like this in practice. Sending another full copy of the entire message as the signature is wasteful, as is encrypting such a large message using an asymmetric cipher. Therefore, the normal RSA signature scheme is to first apply a cryptographic hash function to the message, then pad and sign the hash. This common practice was formalised in PKCS #1[108], along with a list of hash algorithms for use with the signature scheme.

In 1996, Bellare and Rogaway published their Probabilistic Signature Scheme (PSS)[10], which is proven to be more secure. PKCS #1 signatures depend on both the security of RSA, and the security of the hash function used. PSS effectively randomises the input to the hash, which allows a security proof such that PSS is only related to the security of the RSA function. This has been integrated into the latest PKCS #1 document, and is the recommended signature scheme for use with RSA.

3.2.3.2 DSA

The Digital Signature Algorithm (DSA) was developed by the NSA, and published by NIST in 1994; the most recent version of the standard was published in January 2000[89]. It is derived from the ElGamal signature scheme, and also depends on the difficulty of calculating discrete logarithms.

DSA had two advantages over RSA signatures at time of publication:

it was claimed to be unencumbered by patents, and the signature size is much smaller. Twenty years later, the RSA patent has expired, so the first point is no longer relevant. The signature size advantage has increased: DSA signatures are 320 bits, but RSA signatures are the size of the modulus. An RSA modulus of 1024 bits was recommended in 1994, and 2048 bits is more common now.

However, DSA is specified only to use SHA-1. With the recent break of SHA-1 (see section 3.2.1), DSA is no longer safe from a very well-funded attacker. SHA-1 could be replaced by SHA-2, or another has function; however, the resulting signature scheme would not be compatible with DSA. For this reason, future systems are recommended to use RSA-PSS for signatures.

3.3 Certification

While asymmetric ciphers help to solve the shared-key distribution system problem, participants' public keys must be distributed somehow. Anyone can encrypt a message with a given public key, using one of the schemes described above, and be certain that only the holder of the corresponding private key will be able to read it. However, an attacker could publish a false public key, claiming to be someone else, and then read any messages encrypted to it.

Public-key infrastructure (PKI) schemes aim to solve this problem. Instead of publishing only a public key, users publish a certificate: this includes identifying information, a public key, and a signature. Then, anyone who wishes to communicate securely with a user retrieves the certificate, checks the identifying information matches the user, verifies the signature, and uses the public key. If the signature is invalid, the certificate cannot be trusted, and secure communications cannot be guaranteed.

The security offered by such PKI schemes depends on all three parts of the certificate described above. The identifying information must be accurate, and sufficient to verify the purported identity of the certificate holder. Obviously, the underlying asymmetric cryptography must be secure for the system to work. Finally, the entity which signs the certificate must ensure that the identifying information matches the owner of the key.

3.3.1 Transport Layer Security

One example of a very widely-used application based on certification is part of the world-wide web. Secure web transactions use the HTTPS protocol, which uses Transport Layer Security (TLS)[35]. The original security layer for the web was called Secure Sockets Layer, or SSL; TLS is an evolution of this, and the current standard. This scheme uses X.509 certificates[65], which are part of the ITU-T/ISO X.500 directory series of specifications.

Web browsers (or operating systems) must hold a list of trusted Certification Authorities (CAs), which are trusted to verify the link between the identifying information and the public key. However, these CAs can delegate this responsibility to other parties, by signing their certificates. This leads to a certification hierarchy.

When a certificate is retrieved from a web server, its identity is checked against the server, and the signature is verified with the CA's public key. If the CA is in the browser's trusted list, this concludes the verification process; otherwise, the CA's certificate is verified in the same way. This process continues until a trusted CA is found, or a signature is invalid.

3.3.2 Certificate Revocation

PKI schemes must cope with the case where a private key is compromised. This is achieved by allowing certificate holders to publicly revoke their certificate, and encouraging users to check for revoked certificates. Therefore, CAs must publish a certificate revocation list (CRL), and users must check all certificates against this whenever they are used.

3.4 Digital Cash

The field of cryptographic electronic cash was started by Chaum's blind signature scheme[22]. A blind signature is one which allows a participant to have a message signed by another party, without revealing any information about the message contents. Chaum later gave an example of a digital cash scheme which uses RSA blind signatures[23], and then another scheme which prevents double-spending without requiring online verification[24].

3.4.1 Blind Signature Schemes

Digital cash requires a bank to relate the value of an electronic coin (eCoin) to real currency. This is done by signing a blinded message; the bank's signature on any message is agreed to be worth some fixed sum, say \$1. To create coins, a consumer must contact the bank and pay for such a signature. To receive coins, the merchant must only verify the signature of the coin, and check that it has not been spent twice. Since the bank's signature is blind, it cannot determine which of its users paid for the coin, and therefore payment is anonymous.

The RSA scheme is quite simple. Assume that the bank has an RSA key pair: n and e are the public key, and n and d are the private key. The consumer generates some random value r, such that r is relatively prime to n, and then creates $x = r^e m \mod n$. This message is 'blinded' by r, and the bank cannot remove this blind without knowing the value of r. The bank signs x in the normal way to give $y = x^d \mod n$. Finally, the consumer can generate the coin by removing the blind. Because $x^d = (r^e m)^d = rm^d \pmod{n}$, dividing by r gives an unblinded signature: $s = r^{-1} y = m^d \pmod{n}$.

At the end of this coin generation process, the consumer has a signature which is agreed to be worth a fixed value; say, \$1. This can then be presented to a merchant in exchange for goods or services, just as a normal coin would. The merchant can verify the signature to ensure that the coin is authentic, check with the bank to ensure that it has not already been spent, and then offer goods or services in exchange for the cash. As noted above, the bank cannot trace which of its users created the coin as long as the asymmetric cipher is secure.

3.4.2 Micropayments

Blind-signature-based digital cash schemes use asymmetric cryptography extensively. In some applications, these operations are very expensive, and the computational requirements are too great to be feasible. For example, a mobile cellular handset is general low-powered, and so computing an RSA signature may take a significant proportion of its battery life.

This would not be a problem if payments were infrequent. However, there has recently been significant interest in micropayment schemes: those in which a great number of small transactions are made, for low-value services. For such a situation, there exist digital cash systems which do not require so many asymmetric cipher operations. Rivest and Shamir proposed two schemes, Payword and Micromint, which require relatively few signature operations, using hashes instead[105]. Payword requires only one signature operation per vendor, and subsequent transactions require only hash computation. Furthermore, it is a credit-based system, and so can operate offline.

3.4.3 Commercial Systems

There are many online payment systems, but few offer anonymous digital cash. Chaum's work led to the founding of the DigiCash corporation in 1990, which failed to take off, and ceased trading in 1998. A renewed interest in anonymous micropayment schemes may lead to further commercial ventures; recent start-ups include InternetCash[126] and ePoint[31]. The security and efficiency of digital cash protocols is no longer a limiting factor: if demand for anonymous cash-like electronic payment schemes increases, companies like these will be able to meet that demand.

Chapter 4

Anonymous Service Provision

Contents

Several schemes exist in the literature which aim to provide anonymity at service level. This chapter discusses three of the most important of these: MIX Networks, Onion Routing, and Crowds. The threat model, operation, and security properties of each of these is presented, followed by a summary of service-level anonymity in mobile networking. Finally, a case study of applying an anonymity scheme in a mobile environment is contributed.

4.1 MIXes and MIX Networks

Many anonymity schemes are based on a 1981 paper by David Chaum[21]. In this seminal work, Chaum introduces the concept of a 'MIX'[1], a server which aggregates, reorders, and dispatches messages to disguise their origin.

[1] Chaum's paper refers to this as a 'mix', but most later work capitalises the noun to distinguish it from the verb.

4.1.1 Chaum's MIXes

Chaum's original work has been modified and corrected in the years since its publication. However, research has gone in several directions, and it is easier to assess the latest MIX networks with a clear understanding of the original paper. This section presents the original MIX network specification, and examines some of the unstated properties of this proposal.

4.1.1.1 Threat Model

Two assumptions about the capabilities of the modelled attacker are stated in Chaum's original paper:

1. No attacker can determine any relationship between a set of ciphertexts and the corresponding set of plaintexts.
2. An attacker can see the sender, receiver, and content of all messages in the network, and may also inject, remove, or modify messages.

In addition to these, the following implicit assumptions are evident:

1. An attacker can compromise a number of MIXes in the network, without detection.
2. An attacker is sufficiently well-funded to record and analyse all traffic in the MIX network for a duration at least equal to the maximum end-to-end latency.

We also assume that the goal of the attacker is to learn any information about the communicating parties. Examples of such attacks are linking senders and receivers, determining the sender or receiver of any given message, or determining which messages were sent or received by any given participant.

Notable security properties that Chaum's MIX network does not aim to provide include:

End-to-end secrecy The MIX network only encrypts the message while it is in transport. At the last MIX node, it is decrypted. For end-to-end secrecy, the sender must encrypt the message to the recipient.

Denial-of-service resistance To create a denial-of-service for users of a MIX network, an attacker need only disable one node. Because the MIX cascade is fixed (see section 4.1.1.3), this means that the network will be unusable until either the node recovers, or the cascade is redefined.

Communication anonymity MIX networks do not aim to disguise the fact that a given user is sending or receiving messages, only the origin and destination of communication.

4.1.1.2 MIX Operation

Each MIX requires an asymmetric key pair: a public key (K) and a private key (K^{-1}). Chaum denotes encryption of a message M using K by $K(R, M)$, where R is some random string of padding. This simplistic form of padding is now known to be insecure: see section 4.1.1.7 for an attack specifically on MIX networks.

To send a message, the public keys of a MIX (K_1) and of the recipient (K_a) are required. The sender "seals" a message, and sends it to the MIX for processing. To send a message M to a recipient with address A, the sender generates the sealed message X as follows:

$$X = K_1(R_1, K_a(R_0, M), A) \tag{4.1}$$

After gathering a fixed-size batch of sealed messages, the MIX unseals each to get R_1, $K_a(R_0, M)$ and A, and forwards the message $K_a(R_0, M)$ to the recipient A. The random padding R_1 is discarded. The order in which messages are processed is random; Chaum suggests sorting the messages by their ciphertext.

If the underlying cryptosystem functions correctly, only the MIX may unseal the message, and only the recipient may read its contents. Additionally, by adding the random delay between the input and output stages and re-ordering the messages, a passive observer is unable to determine which output message corresponds to the input. This provides correspondence anonymity: an attacker is unable to determine who is communicating with whom.

4.1.1.3 Cascade of MIXes

The output of a MIX can be sent to another MIX, instead of directly to the recipient. Chaum's paper proposes a fixed-route cascade, where the sender of a message repeatedly seals the message to several MIXes in a pre-defined order. For n MIXes, with sending order $n \rightarrow 1$, the sealed message is:

$$X = K_n(R_n, K_{n-1}(R_{n-1}, \ldots K_2(R_2, K_1(R_1, K_a(R_0, M), A)) \ldots)) \qquad (4.2)$$

Each MIX decrypts the outermost sealing and forwards on to the next MIX in the cascade. Clearly, each MIX must know its position in the cascade. The decryption process removes one layer of random padding. Therefore, to preserve message length, the output must be re-padded after each decryption stage.

A simplified illustration of message flow through a mix network is given in Figure 4.1. Note that although the messages (M_1 to M_4) are colour-coded for clarity, a passive observer could not correlate them at the input and output of each stage.

The advantage of using multiple MIXes is that any single MIX can provide anonymity for correspondence passing through the entire cascade. This means that for n MIXes we require only 1 to be honest, in order that anonymity is guaranteed.

For $n = 1$, clearly this is the case: if the only MIX is dishonest, we cannot have anonymity. For $n = 2$, we have an input MIX and an output MIX. The input MIX can correlate the senders to the unsealed messages, and the output MIX can correlate the unsealed messages to the recipients. Both MIXes would need to be dishonest, and collaborate, in order to relate the senders and the recipients.

For $n = 3$, in addition to input and output MIXes, we have an intermedi-

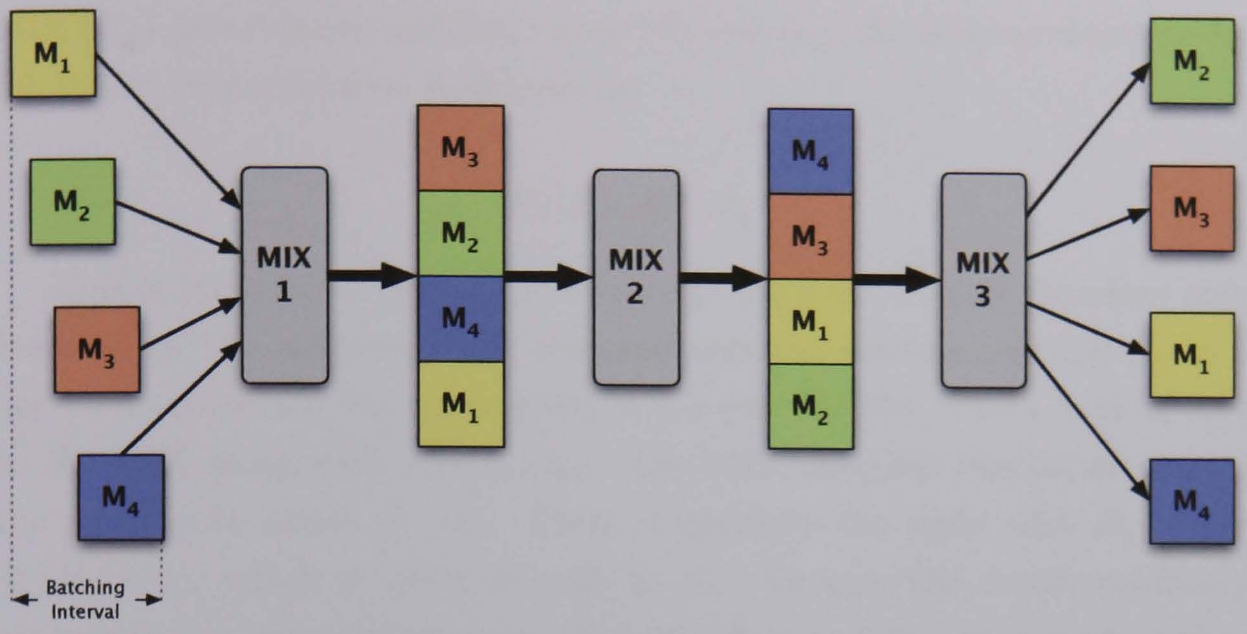


Figure 4.1: Message flow through a three-layer MIX cascade

ary MIX. In this case, the input MIX can correlate the senders to the first-stage unsealed messages, the intermediary can correlate the first-stage unsealed messages to the second-stage unsealed messages, and the output MIX can correlate the second-stage unsealed messages to the recipients. Again, to relate the senders to the recipients, all three MIXes need to collaborate.

Clearly, $n > 3$ is equivalent to $n = 3$: each intermediary node can correlate only between its inputs and outputs. Therefore, if any of the n MIXes does not participate in the collusion, and the underlying cryptosystem holds, there can be no relation determined between the sender and the receiver. This is a very valuable property, as it allows for a significantly compromised network to continue to provide anonymity to its users.

4.1.1.4 Return Addresses

The previous two sections describe a method which allows a sender to transmit a message to a receiver, without the receiver (or a third party) knowing who sent the message. Another technique is used to allow the receiver to reply to the sender, still without removing anonymity protection.

The case of a single MIX is easiest to understand. The sender of a message creates an untraceable return address, which requires the public key of the

MIX (K_1), two random padding values (R_0 and R_1), the address of the sender (A_s). The return address is defined as:

$$K_1(R_1, A_s), R_0 \tag{4.3}$$

This is included as part of the message payload M, and therefore only readable by the recipient. The recipient uses the random padding R_0 as a key to a symmetric cipher to encrypt a reply $R_0(M\prime)$, which is then sent to the MIX along with $K_1(R_1, A_s)$. The MIX decrypts this latter part of the message to attain R_1, A_s. Then, it encrypts the reply with R_1 to get $R_1(R_0(M\prime))$, which it sends directly to A_s. Because the sender originally generated the two random keys (R_0 and R_1), and no other party knows both, only it can decrypt the message. Note that this is an example of hybrid cryptography, as discussed in section 3.1.2.4.

Applying this to a MIX network requires $n+1$ random keys for n MIXes. Before exiting each MIX, the reply must be encrypted with the associated random key. This leads to a final message encrypted $n+1$ times:

$$R_n(R_{n-1}(\ldots R_2(R_1(R_0(M)))\ldots)) \tag{4.4}$$

4.1.1.5 Receipts

One obvious problem with a network of MIXes is that failure at a single node will kill the entire chain. Chaum discusses the possibility of a malicious node failing to forward its full input, and proposes a mechanism to discourage such behaviour. Each sender is provided with a receipt Y for each message input:

$$Y = K_1^{-1}(C, K_1(R_1, K_a(R_0, M), A)) \tag{4.5}$$

Note that this is simply a digital signature (K_1^{-1}) of a random value C and the sealed message. Now, if a message is not forwarded by a MIX, the user can prove this by revealing the receipt Y, the sealed message X, and the random padding R_1. Only the original sender of the message has all three of these messages. Because MIXes are bound to output each message received in the next batch, it is a trivial task for the sender to verify that his or her message has been forwarded, assuming that they can watch the output of the MIX. This principle can be used by each mix in a chain to verify that all of its input is represented in an output, so that the malicious MIX can be discovered.

4.1.1.6 Security Properties

MIX networks are designed to provide two properties of anonymous communication: unobservability, and sender anonymity. Unobservability means that, for any given message, an attacker should not be able to determine any relationship between the sender and receiver. Sender anonymity is achieved when the receiver is unable to determine the identity of the sender.

However, the fact that an individual is communicating is not hidden. A passive observer at the entry or exit of a MIX network can easily see who is sending or receiving messages. This can be countered at the sender side by continuously sending messages with a fixed time interval, using dummy data when no real message is queued.

Information about the sender and receiver can be leaked in other ways. Message length, for example, can be used to correlate between encrypted and decrypted messages. For this reason, Chaum recommends a fixed message length, with longer messages split into several parts. Additionally, the contents of the message could also reveal information about the identity of the sender to the recipient. Ensuring that this is not the case is the responsibility of the user.

Identity protection is achieved by hiding the addresses of the participants. Clearly, the initiator of a sequence of correspondence must know both the address and the public key of the intended recipient. The intention is not to keep all addresses secret, only to prevent an apparent association between the addresses of the sender and the receiver.

4.1.1.7 Security Issues

Chaum's original paper was susceptible to an active attack, due to oversimplistic RSA padding. This was noted and corrected by Pfitzmann and

Pfitzmann[100]. Later, Möller published a security proof for a related method of secure encryption, aimed at length-invariant protocols such as MIXes[83].

As detailed above, MIX networks provide unobservability even if $n-1$ of n MIXes are colluding against the sender. However, the anonymity provided is only within the set of j honest senders and k receivers. Therefore, for small values of j and k, or where some side information about likely sender-receiver pairs is known, an attacker can make a probabilistic association for a given message. Also, the j and k sets may be reduced in size by an attacker, who could flood the network with enough false traffic to reduce the anonymity set to one person.

Another problem with many MIX implementations is that forward messages and replies are distinguishable from each other. This allows an attacker to divide the overall anonymity set into two parts: initiators of correspondence, and everyone else. This is particularly important when replies are infrequent compared to forward messages: in this case, an attacker with control of part of the MIX network can more easily trace replies[30].

Chaum proposes signed receipts as a mechanism for detecting MIXes which fail to forward messages. In theory, this is a valid solution; however, in practice it is not particularly useful. The purpose of a MIX network is to allow anonymous communication, so it seems ridiculous to require a user to reveal their identity and their sealed message to prove that a node is malfunctioning. This immediately removes any anonymity that the MIX network offers them, and so users have a clear disincentive for helping to detect failure of the system.

Finally, all MIX network proposals have a fundamental trust issue. All users of the MIX network must trust the definer and announcer of the cascade. If the cascade includes only compromised MIXes, no anonymity is provided. Therefore, in the initial MIX network proposal, the cascade announcer is a single point of failure for anonymity.

4.1.2 Modified MIX Networks

There has been a great deal of research around the idea of a MIX, leading to a number of modifications to Chaum's original design. Some of these changes are generally incremental improvements, not offering radically new applications for MIX networks.

4.1.2.1 Efficiency

Chaum's original proposal is inefficient with regard to network throughput. Multi-level nested encryption using RSA with secure padding results in a large expansion in message length, proportional to n for n MIXes. This is due to the multiple instances of random padding, which are required for each encryption. Park et al. proposed a clever modification of Chaum's scheme[96], which increases efficiency by using ElGamal encryption (see section 3.1.2.3 for further detail). The message is encrypted once, using the product of all public keys. Decryption is then performed incrementally, with each MIX removing its own key from the product, until all keys are removed and the plaintext is found. The properties of ElGamal encryption ensure that the fully-sealed message remains fixed in length for any value of n.

4.1.2.2 Cascade versus Free Routing

Another practical issue arises due to the fixed cascade of MIXes. If one of the MIXes in the cascade fails, the entire network is useless until another cascade route is created and distributed to users. Allowing the user to choose the route through the network can help with this problem, and also has other advantages.

Within such a free-routing network, the chosen path can be of any length; because the anonymity set at each MIX is equal to the union of the anonymity sets of its inputs, this leads to an increase in the overall anonymity set size. Additionally, a user community can report on the trustworthiness of individual MIXes, and choose to route around them as appropriate. Berthold et. al. provide an analysis of free routes compared to cascades, which shows that free-routing MIX networks provide only slightly poorer anonymity than cascades in the worst case[13].

4.1.3 Anonymous Remailers

The most common practical application of MIX technology is in anonymous remailers. An anonymous remailer is a service which receives messages, and forwards them on to their intended recipient[54]. The message is modified and forwarded with the intent that the receiver cannot determine the original sender of the message.

4.1.3.1 Penet Remailer

Anonymous remailer technology effectively began with the *Penet* remailer service[60]. This server stripped identifying RFC822[28] headers from incoming messages, and forwarded the output to the intended recipient. It also kept a table mapping a generated pseudonym to the email address of the sender, allowing for anonymous replies. This scheme has several flaws, the most worrying being that the administrator of the server has to be fully trusted by all the users. The server owner is capable of tracking messages on input and output, and recording their origin and destination. Worse, the pseudonym table could be inspected, either due to seizure of the server by the government, or simply if the server was exploited. Therefore, there can be no promise of forward anonymity to users of the service.

4.1.3.2 Cypherpunk Remailers

These problems led to the development of *Cypherpunk* remailer schemes, which offer three major security enhancements over the Penet service. First, generated pseudonyms are not supported, as the lookup table was seen to be a major target for attacks. Secondly, and perhaps most importantly, these remailers support input and output of encrypted messages, countering passive eavesdropping of message contents. Finally, Cypherpunk remailers can forward messages to other remailers on the way to the final recipient. The latter two enhancements imply similarity to a free-routed MIX network; however, there are some important differences. Although supported, there are no guarantees for link encryption or message re-ordering. Additionally, output messages are variable in length, which enables some passive correlation attacks.

4.1.3.3 Mixmaster

A third attempt at secure anonymous email forwarding was made with *Mixmaster*[84]. Again, some key flaws of previous anonymous remailer designs were addressed.

Mixmaster embraces several elements of the Chaumian MIX network. The scheme enforces multiple-server chaining, with link encryption between servers; the output of each Mixmaster node is always fixed-length, to defeat correlation attacks based on message length; messages are re-ordered between input and output, to provide security against wide-observation timing attacks. However, Mixmaster also has some disadvantages when compared to previous schemes. Most crucially, there is no support for anonymous replies, which has hindered migration from Cypherpunk remailer network.

4.1.3.4 Mixminion

The most recent step in the evolution of anonymous remailer technology is *Mixminion*[30]. Arguably its most major practical advantage over Mixmaster is support for anonymous replies. Another important improvement is guaranteed forward anonymity: instead of using SMTP[101], link-to-link communication is done using TLS[35] over TCP. Once the ephemeral TLS keys are destroyed, it is impossible for a node to reveal messages transmitted in the past. The Mixminion specification has been adopted by the Mixmaster team as the basis of the next generation Mixmaster network, and is likely to be adopted by most anonymous email users.

4.1.4 Conclusions

Due to the batch-gathering delays at each stage, and the multiple decryption operations, MIX networks are not well-suited to real-time communication. Chaum's paper is a proposal for an electronic mail system, and indeed the most common real-world use for MIX networks is in anonymous remailers. However, other applications are possible, if a transmission delay of a few seconds to a few minutes is acceptable.

4.2 Onion Routing

In the mid 1990s, Goldschlag et al. designed and published a real-time routing protocol which attempts to anonymise connections, using methods somewhat related to MIXes[55]. This is known as *Onion Routing*, due to the multilayered structure used to set up anonymous connections. Further ideas and analyses were later published: on its evolving design, as it was experimentally implemented and deployed[103]; a discussion of the access mechanisms to the onion routed network for various participant classes[123]; and an adversary and security analysis[124].

4.2.1 Operation

The scheme uses a distributed network of onion routers, which is used to obscure the senders and receivers of real-time communications. Anonymous communication requires three stages: connection setup, data transfer, and teardown. The scheme is designed to ensure that the connection setup adds as little latency as possible, and that the data transfer stage requires as little message size overhead as possible.

4.2.1.1 Connection Setup

Connection setup is controlled by the initiator of communications, by creating a data structure which defines the path taken through the network of onion routers. This is known as an *onion*, and is a recursively layered structure of parameters for communication. Each onion is encrypted using the public key of the router for which it is intended, and consists of two parts: control information for the router, and a payload. The payload may either be another onion, or for the final stage, random padding[2].

Each onion's control information consists of the address of the next router (A), a symmetric key to communicate with the previous router (K_{j-1}), a symmetric key to communicate with the next router (K_j), and a timeout parameter for the hop (T_j). The address of the previous router is implicitly known, as it forwarded the onion to the current router. Timeout is specified so that the connection can be dropped automatically if the initiator stops responding.

Upon receiving an onion, the router decrypts it, and records a copy of it for future use in communications. It then pads the payload to a fixed size, and forwards it to the next router, using the address given in the control information. This process can be seen in Figure 4.2.

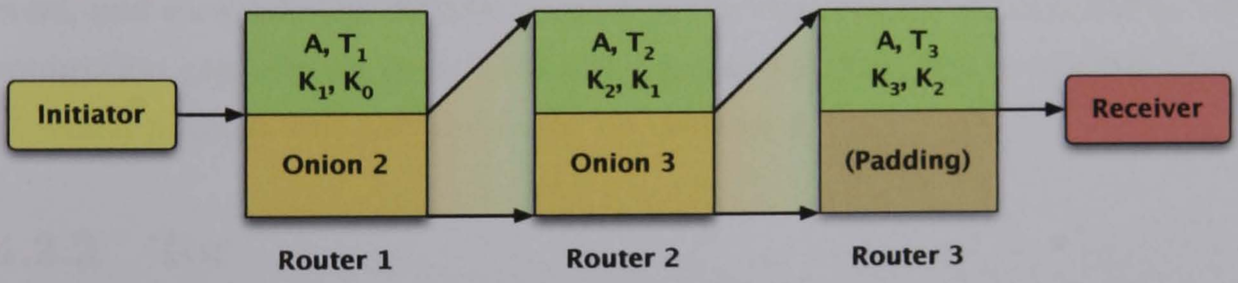


Figure 4.2: Connection setup for a three-hop onion route

The first and last hops of the route must be *entry* and *exit funnels* respectively; these are special routers which are able to accept incoming and create outgoing connections to nodes outside the onion network. When the onion reaches the exit funnel, connection setup is finished.

4.2.1.2 Data Transfer

Once the setup phase is complete, data transfer can begin. To begin this, the initiator repeatedly encrypts the payload using the symmetric keys generated

[2] The final onion contains random padding so that only the exit node knows that it is the final step in the route.

for the original onion (K_i). This is done in reverse order: the last hop's key is used first, then the second-last, and so on.

Data is then sent to the first hop on the route, which decrypts its contents using the symmetric key for the next router (K_j), as received in the onion. It then forwards the message to the address specified as the next hop on the route (A), which follows exactly the same process. At the exit funnel, the message is unencrypted and sent to the intended recipient.

For return data, the opposite process is followed. Each hop encrypts the data with the key for the previous router, and it arrives at the initiator encrypted with all keys. Clearly, only the initiator can decrypt this message.

4.2.1.3 Teardown

When a connection is no longer needed, the initiator can send a *destroy* control message to begin the teardown process. Each node in the route receives the message in turn, closes all connections, destroys all cryptographic parameters, and forwards the destroy request to the next router. Alternatively, the connection can remain open until the timeout for each hop is reached (Tj), at which point it will automatically be destroyed.

4.2.2 Tor

The original onion routing specifications never left the experimental stage. However, the protocol design was later improved and deployed in a widely-used anonymous communications scheme: Tor[38]. Initially an Electronic Frontier Foundation funded internal project, Tor is now used to anonymise a wide variety of Internet traffic: there are currently over 500 routers, handling average traffic of around 60MB/s[125].

Tor is available as a free download for all major operating systems, and acts as a SOCKS proxy[79]. This allows it to interoperate with most standard Internet client software, the most commonly-used being web browsers and instant messaging protocols.

While the above principles of onion routing still apply to Tor, there are some significant changes to the onion routing design, to counter issues discovered in the experimental deployment stage. Some of these are implementation issues: for example, in contrast to previous deployments, Tor does not remove private information from the application layer. Instead, a secondary privacy-enhancing proxy is required to protect privacy at higher layers; for example, the web privacy tool Privoxy[34].

Most significantly, the onion structure above is not used in its original form. Instead, a route is built incrementally; this is done to achieve forward secrecy (see section 3.1.2.5). This means that the initiator agrees session keys individually with each hop in the route; at the route teardown, these keys are destroyed, and they cannot be recovered.

The data transfer and teardown steps of the protocol are broadly the same as for previous onion routing proposals. More details of the exact operation of the Tor initialisation protocol can be found in the current protocol specification[37].

4.2.3 Threat Model

Both Tor and the original onion routing scheme are pragmatic in their threat model. Instead of countering a global passive adversary, the designers assume a more realistic and less powerful attacker.

Tor does not make any attempt to hide the fact that communication is taking place, only to obscure the eventual recipient of each outgoing message. Similarly, it is not secure against end-to-end timing or correlation attacks. The assumed adversary has control over some portion of the network, and the expected attack is traffic analysis: trawling for interesting information, rather than attacking a specific user.

4.2.4 Security

Each router only knows which two nodes are before and after it in the route, and has no knowledge of the rest of the connection. Indeed, a router could be looped through twice in a single route, without being able to distinguish between such a case and two different connections. This property means that an onion routed connection is secure unless the entire network is compromised.

While Tor is based on the ideas behind MIX networks, in practice no batching and re-ordering is performed by routers. This is one of the trade-offs made to improve end-to-end latency, and enables timing attacks to be performed. Therefore, the only anonymity advantage given by a fully-operating Tor network is obscuring communications metadata from trawling adversaries; it cannot defeat a concerted attack against a particular user.

Another problem with onion routing is that it is centralised. A list of routers is kept on several "trusted" servers, to allow initiators to build an onion. If these servers are compromised, the entire network can no longer be trusted. Such a single point of failure makes the distributed nature of the network less important. The ability to handle a majority of malicious routers is irrelevant if there is an easier target for a powerful attacker, and in this case that target is the directory servers.

4.2.5 Conclusions

The Tor implementation of onion routing is a successful low-latency anonymity service. It is readily available and well-tested, in part because it is free and open source software. The anonymity properties provided are a useful subset of those given by classical MIX networks: sender-receiver unlinkability against a limited attacker. Despite this, it has not yet gained mainstream use.

The most major blocker to widespread adoption of Tor is the trade-off between perceived utility and performance. Onion routers are bandwidth-limited, and the route setup time is in the order of one or two seconds[38]. This leads to a noticeable drop in web browsing performance, for little obvious value in most cases. Unless the user is defending against a specific attacker, there is little reason to accept this performance degradation and use Tor regularly. If an anonymity scheme is to be adopted widely, it must provide some useful privacy enhancement with little or no degradation in perceived performance.

4.3 Crowds

A different approach to service anonymity was proposed by Reiter and Rubin in 1998: Crowds[104]. The scheme was named appropriately: every message travels through a large group of peers, or a crowd, before reaching its intended destination. Crowds is superficially similar to a free-routed MIX network, the major difference being that the route is not sender-defined, but generated randomly as the message passes through the network.

One of the most interesting aspects of this work is the introduction of the *degree of anonymity* metric; see section 2.3.5 for an explanation of this term. Crowds aims to provide varying degrees of probabilistic anonymity against various attackers, but it does not achieve sender-receiver unlinkability.

4.3.1 Operation

A Crowd is composed of a number of members (n), each of which runs a piece of software called a *jondo*[3]. The jondo receives and sends messages to other members of the crowd; membership is managed by a central server, known as a *blender.*

Upon receiving a message, a jondo has two options: it may either ***submit*** it directly to the recipient, or *forward* it to another jondo in the crowd. The choice between forward and submit is made with a random "coin toss", weighted such that the probability of forwarding is p_f. This probability is fixed for the entire crowd, and received by each member upon joining. This value determines the trade-off between performance and degree of anonymity; this is discussed further in section 4.3.3.

An example of three possible message routes for a simple crowd is given in Figure 4.3. For each route, the message originates at member j_a. In the first route, the message follows $j_a \rightarrow j_b \rightarrow j_c \rightarrow r$; the second route is $j_a \rightarrow j_c \rightarrow j_b \rightarrow r$; the third is simply $j_a \rightarrow r$. As can be seen, any path through the crowd is possible, and therefore the recipient cannot be sure who was the original sender of the message, even in the case where it was sent

[3] Pronounced "John Doe": a US term for an anonymous or unknown person

directly from the first jondo.

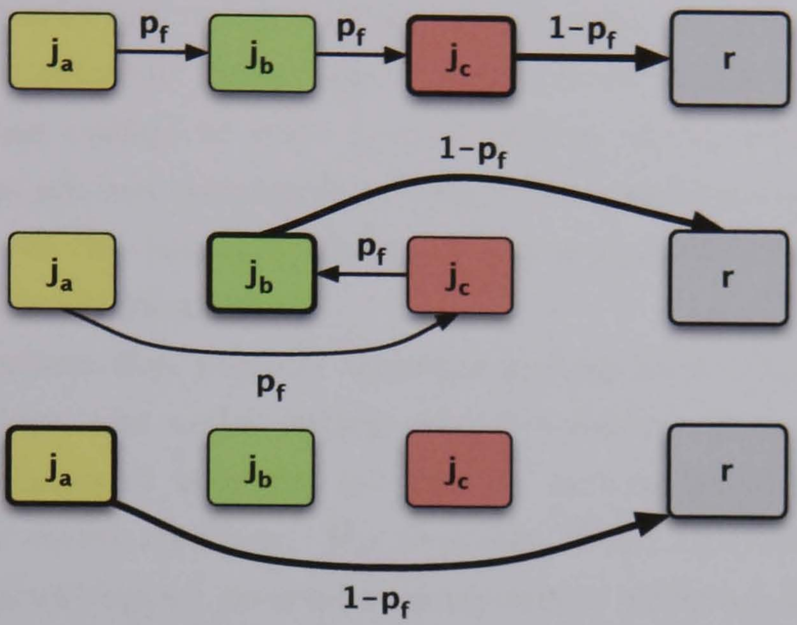


Figure 4.3: Three possible routes through a Crowd of jondos j_i to recipient r: the recipient cannot determine which jondo was the original sender

Replies are transmitted back through the request path, in reverse order. This is achieved by recording path IDs in a local store for each message. When forwarding or submitting a message, the jondo waits for a response from the next jondo or the end server, as appropriate. Upon receiving a response, it simply sends it back to the jondo from which it received the request.

Once a route through the network is established, it remains in place until one of two conditions is met. First, if a connection error occurs such that one jondo fails to communicate with the next, the remainder of the path is randomly re-routed by the last functioning jondo. Secondly, all paths are dropped after a fixed period of time. This allows newly-joined members of the crowd to add new paths at the same time as other members; otherwise, in a near-saturated crowd, any new paths can be reasonably assumed by an attacker to be originated by new members.

4.3.2 Threat Model

Like MIX networks and onion routing, Crowds does not hide the fact that communication is taking place. Like Tor, the threat model also states that the attacker has control of some limited portion of the network. Global passive attacks are not countered; in fact, even a local eavesdropper, who can only observe the network activity of one jondo, can determine which messages that jondo initiated.

Crowds assumes that possible attackers include local eavesdroppers. the message recipient, and collaborating crowd members. It is assumed that the first two classes of attackers are passive, and do not collaborate with each other, or crowd members. For example, if the recipient collaborates with a local eavesdropper, no sender anonymity is achieved; if the recipient collaborates with malicious crowd members, the sender anonymity is reduced.

4.3.3 Security

Crowds provides some degree of sender anonymity against collaborating crowd members and the message recipient. This differs from MIX networks and onion routing, which only provides sender-receiver unlinkability against collaborating MIX nodes; sender anonymity is not achieved.

The most important anonymity provision Crowds offers is *beyond suspicion* sender anonymity against the receiver. This means that the receiver of a message has no reason to suspect one particular crowd member is the originator of the message, as all members of the crowd have similar probability of being the original sender. This provision is guaranteed for any size of crowd, although clearly the value of this anonymity increases with the size of the crowd, along with the anonymity set.

Against a certain number of collaborating members, Crowds provides *probable innocence* sender anonymity. For this to be true, the following condition must be met:

$$n \geq \frac{p_f}{p_f - 1/2}(c+1), \tag{4.6}$$

for n crowd members, p_f forwarding probability, and c collaborating members. When this is the case, the sender of any message appears no more likely to be the originator than to merely be forwarding it. Further, as the size of the crowd increases, and c remains fixed, the probability of the collaborating members receiving any message tends towards zero, and so sender anonymity tends towards *absolute privacy.*

Another property which Crowds provides is receiver anonymity, again to some degree. As the crowd increases in size, a local eavesdropper is increasingly unlikely to be able to determine who the recipient of any given message is; this probability is equal to the chance of the observed jondo being the final hop on the route. Similarly, as crowd size increases, collaborating members are less likely to receive any given message, and so receiver anonymity tends towards *absolute privacy.*

4.3.4 Further Work

Two security analyses of Crowds are of particular interest here. The first describes the "predecessor attack", which was first noted as an issue in the original Crowds paper[104]. Another work applies a probabilistic model checker, PRISM, to the routing protocol used in Crowds.

The impact of the predecessor attack on Crowds and other related anonymity schemes was considered in detail by Wright et al. in [133]. This attack assumes that there are c collaborating jondos in the crowd, which share information in order to try to determine the initiator of communications.

When a path initiated by a non-collaborator reaches a collaborator, the collaborator can assume that the most likely initiator of the path is its immediate predecessor. For some values of c, p_f, and n, this assumption can have confidence greater than 1/2, in which case *probable innocence* sender anonymity is lost. This was noted by Reiter and Rubin in [104], but Wright et al. demonstrate that a more concerted attack can lead to this situation within $O(\frac{n}{c})$ path destruction/reformation iterations. This is an important attack: recall that paths are destroyed and reformed periodically, and whenever the crowd membership changes.

Shmatikov conducted a formal analysis of the performance of Crowds against the predecessor and other attacks, using a model checking tool[114]. This work concentrated on small crowds, due to the computational complexity of the technique used. Regardless, it showed that the sender anonymity provided by Crowds against collaborating jondos actually decreases as n increases, so long as n/c remains constant. This is a somewhat surprising result, although the original work indicated that many collaboration-related threats could only be solved by increasing n while keeping c constant.

4.3.5 Conclusions

Crowds is an alternative approach to low-latency service anonymity provision. Its main advantages over MIX networks and onion routing are performance and scalability. First, because no public-key cryptography is required, Crowds is much less demanding of server resources. Somewhat related to this, the lack of a separate path setup stage, combined with the temporary preservation of routes, leads to a probable reduction in perceived latency.

Scalability is even more important. Every member of the crowd must run a jondo, and must contribute to the network by forwarding messages. However, as the crowd increases in size, the likelihood of carrying traffic decreases, and so the network load on each jondo remains roughly the same. MIX networks must have many more users than MIX nodes, and onion routers must handle a significant traffic load. These properties make Crowds a more suitable choice for broad acceptance.

Unfortunately, the degree of anonymity provided is lower in some circumstances. If we reduce the purpose of the crowd to obscuring which member contacted a given recipient, where the attacker is the recipient itself, Crowds performs admirably. As detailed above, the protocol guarantees *beyond suspicion* sender anonymity. However, against in-network attackers, Crowds performs poorly, as shown by [133] and [114].

4.4 Mobile Service-Level Anonymity

There is some discussion in the literature of mobile communications anonymity. Asokan describes a protocol for authentication for mobile services which preserves some anonymity, hiding the mobile user's identity from a passive adversary but not the service provider or the mobile operator[5]. Samfat extended this work to allow authentication under many different threat models, including hiding almost all information about the user from the service provider[110].

Federrath, Jerichow, and Pfitzmann apply MIX networks at the signalling level to hide the cell location of a mobile user from a third-party passive observer, but not from the network operator[46]. Reed describes the use of Onion Routing for mobile services[102], but without considering the significant differences in capabilities and costs when comparing mobile and fixed networks.

All three approaches discussed in the previous section, and many other examples from the literature, are designed from a fixed network perspective. MIX networks have been used with many fixed-network protocols; Tor is more specifically concerned with Internet protocols such as HTTP and IRC; and Crowds is even more specialised, as it is intended only for World-Wide Web transactions.

However, there are some inherent differences between fixed and mobile networks which undermine the security assumptions made for these schemes when applied to mobile network anonymity. To achieve service-level anonymity with mobile communications, these differences must be examined. This section discusses how anonymity schemes apply to currently-available technologies, and goes on to make recommendations on how best to achieve anonymity in mobile systems.

4.4.1 Fixed versus Mobile Networks

One obvious difference between fixed and mobile communications is availability. A fixed network connection is always-on; if the network connection is unavailable, it is generally due to a fault. For a variety of reasons, this is not true for mobile systems. No mobile communications technology has 100% geographic coverage, so if the mobile device is actually moving, it may eventually move out of range of the network. Battery life is limited, especially when actively transferring data, so usage tends to be bursty. Since mobile systems are not always available, they cannot be relied upon to act as peers in an anonymity network.

Additionally, current mobile communications systems tend to have poorer network performance than their fixed equivalents, in several areas. The most obvious is latency: enhanced second-generation cellular systems, like EDGE, have very poor round-trip times. Studies have shown that latency in such systems is in the order of seconds[111]; this is orders of magnitude higher than that of fixed access networks, which tend to be around 30ms[111]. High end-to-end latency has both first- and second-order effects. Most obviously, the delay added to each step of a Tor or Crowds route would lead to extremely high total round-trip times, which would appear to the user as unacceptable latency at the application layer.

A second-order effect of high latency is reduced effective throughput with small file sizes. Due in part to high round-trip times, enhanced second-generation cellular systems have an average TCP throughput of around 40kbit/s for 100kB HTTP transfers[111]. In comparison, fixed broadband Internet connections have much lower latency, and much higher throughput of around 600kbit/s for equivalent transfers[59]. For smaller transfers, the gap will only widen further. This further hinders the ability of a mobile device to act in a peer-to-peer network: the network is only as fast as its slowest link, and this low capability would make anonymous service usage seem very sluggish.

Another difference between fixed and mobile networks is cost of usage. Along with slower transfer speeds, mobile communications systems are also more expensive. In the UK, mobile data is currently charged on a per megabyte basis, at around £1 per megabyte transferred (see section 5.3.2 for details). This level of expense makes data throughput a scarce resource for users, which has two consequences for mobile anonymous services.

First, there is a need for efficiency: the overheads added by anonymity schemes must be kept to a minimum, or the cost will be prohibitive. More importantly, this cost model is yet another barrier to the adoption of peer-to-peer anonymity schemes. While in fixed networks there is effectively no cost associated with being a good citizen in a peer-to-peer network, in mobile networks this cost potentially overrides any benefit that could be gained from taking part.

In summary, there are several major differences between fixed and mobile networks, which affect the peer-to-peer nature of anonymity schemes such as Tor and Crowds. These include sporadic availability due to limited coverage and battery life, high end-to-end latency, lower throughput, and cost of sending and receiving data. For these reasons, peer-to-peer anonymity schemes are practically unworkable on mobile networks, until costs are lowered to approach those of fixed networks.

4.4.2 Distributed Anonymity Schemes and Trust

If mobile terminals are unable to take part in peer-to-peer anonymity systems, another approach to anonymity must be found. The most straightforward option would be to take part in a Tor or Crowds network, but only as a simple, non-contributing client. However, this reduces the level of anonymity offered by both protocols.

A mobile user could use Tor as a client, without running a local onion router. This is not recommended, as it means that the first router in the path used will know for certain that the user originated the connection.

Similarly, a user could connect through a multi-user jondo to a Crowds network. This case is even worse, as the jondo will know that you originated the connection and who the destination server is. In this case, sender anonymity is therefore only offered against the destination server.

This leads to a fundamental trust problem. Because a mobile terminal cannot be a first-class citizen in an anonymity network, even more trust than usual must be placed in the other network participants. There is little justification for this trust: the user would have no legal recourse if the service is dropped, or anonymity is removed. Further, the user would have to place trust in those responsible for providing the anonymity service without necessarily knowing who they are.

One way around this problem is to offer a commercial anonymity service. Instead of placing trust in unknown peers, the user need only be concerned with the commercial entity with which he/she has a contractual agreement. Any breach of privacy can be punished by prosecution under contract law, at a minimum, and information privacy laws as applicable (depending on the appropriate country's legal system).

When this option is considered, solutions to other mobile anonymity problems fall into place. With a single trusted entity providing anonymity, there is no need for a sophisticated multi-hop protocol. Instead, a simple single-layer protocol could be used, and this would also have the benefit of having lower overheads in terms of latency and message size. With a single-hop anonymity protocol, performance should not be noticeably degraded over a non-anonymous system; any additional latency will be due to extra routing through fixed networks, which are so much lower-latency than mobile access networks that the extra latency will be in the noise.

4.4.3 Threat Model

As with any security system, the threat model applied to an anonymity scheme must be clearly defined. In this section, potential attackers to mobile service-level anonymity are examined, and a pragmatic threat model is outlined.

There are four expected potential attacker classes: the anonymity provider, the service provider, the network provider, and a global passive attacker. The first three of these are real entities, whereas the latter is the more general attacker class dealt with in the anonymity schemes in the literature.

4.4.3.1 Anonymity Provider

As discussed above, with Crowds and Tor in mobile systems, the anonymity provider is the operator of the network user list. In a commercial anonymity service, the company running the service clearly takes the anonymity provider role instead.

For any practical anonymity system to work, users must place trust in the anonymity provider. Crowds and Tor restrict this trust to fairly presenting the list of routers, possibly also extending this responsibility to include verification of fair router operation. In comparison, using the proposed commercial anonymity service implies much greater trust in the anonymity provider. However, this is justified due to identification and contractual obligation: there are severe consequences for breaching the agreement and trust.

Therefore, this threat model defines that the user must trust the anonymity provider. Any unfair behaviour on its part is likely to be difficult to detect, but in the case of a commercial provider, there is a route for recourse.

4.4.3.2 Service Provider

Remembering that this is service-level anonymity, we define the service provider as the operator of the system which provides the high-level service to the user. For example, a messaging server, web site operator, or location-based services provider.

The service provider is the most obvious attacker to defend against. A user might wish to make use of the services being offered, but not trust the service provider with their identity or any identifying information. Service providers are likely to receive multiple requests from the same user, and may be able to build up a profile based on the similarity of these requests, especially if they can be tied to the same network address.

The service provider is expected to want to find the user's identity. It is capable of analysing multiple received messages to mine for identifying information, and any anonymity system must work against this to make it difficult for the service provider to find the user's identity.

4.4.3.3 Network Provider

It is likely that the mobile network operator has full control of the network. It has the ability to read, modify, delete, and inject data into the network, without detection by the user. Furthermore, it has a great deal of identifying information about the user, and it is likely that it would benefit from gaining more data; this could be used for marketing purposes, or sold to other companies for direct profit.

For these reasons, the network provider must be considered as a threat to anonymity. Its capabilities are high, and it has some motive for gathering data about its users.

4.4.3.4 Global Passive Attacker

While relevant in fixed networks, this threat is somewhat unlikely to exist in mobile systems. The mobile access network is private and fairly well-secured. Before any network traffic reaches a wider network, such as the Internet, it is only likely to be readable by the network provider. An attacker which has the ability to read data on both the access network and the wider network would have to be extremely powerful, and presumably also have control of the network provider.

Note that in GSM and GPRS networks, this would not be the case; communications between base stations are not encrypted in any way, and an attacker could easily intercept at this weak spot. The UMTS third generation mobile architecture corrects this, so with future mobile systems such an attack will not be possible. Only third generation and future systems are considered for this analysis.

A subset of the global passive attacker would be one which has the ability to read any traffic passing over the wider network, but not the access network. This is still unlikely, but equivalent to global attackers in anonymity schemes in fixed networks. It is therefore fair to assume that such an attacker exists, and that the scheme should cope with it.

4.4.4 Summary and Recommendations

Anonymity schemes can theoretically be applied to mobile networks exactly as they are to fixed networks. However, with current technology, the differences between mobile and fixed networks cause some major practical problems with this simple approach.

Mobile systems have several properties, described above, which make it unfeasible to use a mobile device as a peer in a distributed anonymity scheme. Therefore, mobile service-level anonymity requires that the mobile device act as a client to an anonymity-providing server.

Such a configuration reduces the anonymity capability of the device with some protocols, but less so with others. More importantly, it represents a fundamental shift in trust, which is perhaps not immediately obvious. As noted in the threat model above, the use of any practical anonymity scheme in mobile networks demands that a great deal of trust is placed in the operator of the server used to access the anonymity scheme.

There are two options for the mobile user who wishes to achieve anonymity. They can either use a generally-available system, such as Tor or Crowds, and trust that the central server fairly administers the list of servers, and that the server they interact with is trustworthy. Alternatively, and more pragmatically, the user can pay for an anonymity service from a commercial provider, who contractually agrees not to reveal the user's identity.

It is this latter option which is recommended by this study, as it is the most practical in today's third-generation mobile networks. The disadvantage, of greater trust placed in the anonymity provider, is outweighed by the many advantages. Trust is placed in a known entity; there is an avenue for compensation should the anonymity provider act unfairly; and performance degradation due to the anonymity scheme is likely to be significantly lower than other options.

There remains the question of exactly how the commercial anonymity provider should operate. This varies according to the precise requirements of the user, the service, and the threat model applied. The following section presents a case study of this recommended scheme, for anonymous location-based services.

4.5 Case Study: Anonymous Location-Based Services

Many new high-end mobile phones are now equipped with Global Positioning System (GPS) capabilities, which allow precise measurement of the handset's current location. The trend in incorporating GPS functionality into handsets is partially due to location measurement requirements of the recent 'E911' legislation in the United States. This law requires that new handsets must be able to report their location when making 911 emergency calls, in order to help first responders to find people in danger more quickly[44].

Service providers aim to take advantage of the capabilities offered by GPS, and are preparing to offer sophisticated location-based services (LBS). For example, a user could find nearby stores with special offers, or request suggestions of nightclubs to visit when leaving a bar, or find online reviews of nearby Italian restaurants. There is potential for a very wide range of such services, and current services may be augmented by the addition of location information.

A high degree of positioning accuracy is necessary for these new services to operate. Using GPS meets this requirement, but this raises new privacy concerns for users. GPS receivers can compute position to within a radius of 5–20 metres, and such precise location can be sensitive information. Some privacy-conscious customers may be concerned about the possibility of being tracked in everyday situations. Therefore, a major privacy problem facing LBS is how to anonymously receive location-related information.

4.5.1 Location Measurement

Many methods of measuring mobile phone location exist, and the simplest are already possible with today's networks. The most obvious method in cellular networks is to use the known centre and fixed radius of the current cell as the location area. However, this gives a highly variable precision, which can be as poor as 35km[4], so other methods have been developed.

[4] The maximum diameter of a standard GSM cell is 70km.

The best of these, Uplink Time Difference of Arrival (U-TDOA) provides far better precision than the cell-ID method, varying from 50m to 200m depending on the number of measuring base stations[19]. However, it requires significant investment, and does not function well in rural areas, due to the sparsity of base stations. As a result, it has as yet been unsuccessful in the marketplace, and many operators in the US have chosen GPS as the solution to the E911 requirements.

GPS is a satellite-based location measurement system, which allows handheld devices to attain precise time and position information[53]. The system consists of a constellation of 24 satellites, many controlling ground stations across the world, and any number of end-user receivers. Each satellite contains an atomic clock for precise timing, and it continuously broadcasts a time signal downwards, towards the planet surface.

The receiver computes its position using trilateration, using the difference in time of arrival of signals sent from four of the satellites. Errors in measurement are introduced due to varying atmospheric conditions and multipath effects; standard GPS yields location precision of around 20 metres. Some of these errors can be countered using fixed ground-based reference stations, which improve this figure to around 5 metres. For mobile systems, further enhancements have been made to improve the time taken to make the measurements[39].

4.5.2 Previous Work

Prior work in the area of location privacy includes policy specification and adherence, and operator-provided perturbation of data in time and precision. These two approaches, discussed below, are complementary; they share the assumption that the mobile network operator is a trusted party which can be employed to provide privacy to the untrusted external world.

The IETF's Geographic Location/Privacy (GeoPriv) working group[81] has defined a policy framework which facilitates the transfer of location information while retaining privacy[112]. The user is able to specify exactly what can be done with the private information, using a well-defined XML-based language. This language allows location information to be tailored to the viewer, and even allows rules specifying propagation and retention time of a location object. However, it is obviously only useful if the location-based service provider supports the framework, and honours the policy specification given by the user.

In contrast to a policy-based approach, other researchers have proposed schemes which reduce the information given to the LBS providers. Beresford and Stajano propose a MIX-like approach to location anonymity[11]. This was implemented for the Active Badge position-tracking scheme[130], but is equally applicable to mobile location-based services. Users create temporary pseudonyms to support return messages, and outbound requests enter a location-specific MIX node to counter timing attacks. The conclusion reached from analysis of simulations of the scheme was that the scheme provides some form of anonymity, but only at the level of *possible innocence* (see section 2.3.5). This is supposed to be due to the high accuracy of the Active Badge location data, but no evidence is presented to directly support the conjecture that the scheme would improve for a lower-accuracy system.

Enhancing this approach, Gruteser and Grunwald describe a scheme in which the network operator collates and degrades location data, to create a set of k indistinguishable users[57]. With this scheme implemented correctly, it should be impossible for any external party to determine the identity of a user given only their location. However, quality of service is significantly reduced: much like a MIX, the scheme requires several similar in-flight requests to be grouped together before forwarding them, but also each of these will have the location data degraded to make them seem alike. The consequences are twofold: first, there will be significant delays while sufficient requests are collated; most importantly, the user's position will be misrepresented and precision reduced.

4.5.3 Threat Model

The approaches described above all assume that the network operator is to be trusted with the location data, in order to mix or distort it to preserve anonymity. This trust is not necessarily well-placed. As described in section 4.4.3, the network provider is a powerful attacker, which already collects identifying information about the user. The ability to gather further data about the user's exact location could be of benefit to the network operator, and of concern to the user. Therefore, the threat model for the scheme proposed here assumes that the network operator is an attacker which should not be given the user's exact location.

However, it is clearly also unwise to trust the location-based services provider with the user's location *and* identity. The user should be free to use any LBS provider, without concern for revealing their identity along with their precise location. The second attacker in the system is assumed to be the LBS provider, and its goal is to identify the user.

Therefore, this threat model assumes that the mobile network provider knows the user's identity, and wants to know the exact location of the user. Similarly, the LBS provider wants to know the user's identity, and giving away the user's network address would lead to this. It is also assumed that the mobile network and the LBS provider are not in collusion against the user.

Clearly there must be some party which provides anonymity to the user. Again as described in section 4.4.3, this party must be trusted by the user to provide anonymity. It is assumed that the anonymity provider does not collude with the mobile network provider, or with the LBS provider. It is also provisionally assumed that the LBS provider is not a global attacker; it cannot read the network traffic entering and exiting the anonymity provider.

4.5.4 Degree of Location Privacy

There are several techniques which can be used to find the location of a cellular phone (see section 4.5.1). Most of these are network-measured, based on measurements of uplink or downlink transmissions; currently, only those using GPS are entirely handset-measured. Clearly, only handset-based techniques may be used to measure location anonymously, if the network operator is assumed to be an attacker.

While the network operator could theoretically obtain an approximate location of the user, this is likely to be via the cell-ID method, which is extremely imprecise. The current best method of locating mobile users without their co-operation, U-TDOA, is still an order of magnitude less precise than GPS[19]. Furthermore, any network-centric location-tracking system other than cell-ID would require significant investment to operate, and so such an attack would be unlikely to go unnoticed.

If the network operator knows the cell location of the user, the operator may be able to estimate which city or town the user is in. If given GPS measurements, they will know which building the user is standing next to. There is a dramatic difference in the precision of these two measurements; the latter gives the network operator much more information about the user.

The very nature of current cellular mobile telephony requires users to give up some privacy in order to make and receive calls. Withholding any further location information is in the privacy interests of the user. Therefore, there is significant reason for the user to withhold the more accurate GPS measurements from the mobile network operator, in order to retain some degree of location privacy.

4.5.5 System Overview

There are two main problems with previous approaches to anonymous location-based services. The first is in the threat model: the network provider is assumed to be trustworthy, and the anonymity scheme is expected to be run by the network operator itself. As explained by the threat model above, this is not necessarily a valid assumption. Secondly, distorting the location measurement value, in order to create an anonymity set, leads to lower quality of service. The exact location of the user cannot be known by the LBS provider, and so the utility of LBS may be significantly reduced.

The solution presented here does not require the user to trust the network operator, nor reduce the precision of their location measurements. Anonymity is granted against the LBS provider, and the network operator is given no further information about the user's location. This is achieved by using a single anonymity-providing server, operated by a user-selected third-party.

Users of this scheme must trust this anonymity provider, but only in a very limited sense. It must not collaborate with the LBS provider, or anonymity will be lost. This constraint means that the relationship between the user and the anonymity provider is simple. Instead of a long-term contract, payment for this service could be made in advance, using a digital-cash micropayments system (see section 3.4). Then, the anonymity provider would not know the identity of its users, only their network addresses, thus mitigating the privacy loss if the anonymity provider colludes with the LBS provider.

The scheme defines a short customised protocol, which is optimised for the network characteristics of mobile systems. At the same time, it places low requirements on the anonymity provider, which should keep running costs low. The scheme's protocol is described below.

4.5.6 Protocol Definition

The protocol uses an anonymity router, which blindly forwards LBS requests and responses between mobile users and a third-party LBS server. These messages consist of an encrypted payload (unreadable by the router) and an encrypted return address (unreadable by the LBS server). This is achieved by using hybrid cryptography (see section 3.1.2.4). The full protocol operation is given in Figure 4.4. Table 4.1 lists the notation used in this section.

Term	Explanation
MT	Mobile terminal, the user of location-based services
AR	Anonymity router, provided by the network operator
LS	LBS server, external to the mobile network
K_x	Symmetric encryption with key x
K_x^{-1}	Symmetric decryption with key x
E_y	Asymmetric encryption with the public key of y
D_y	Asymmetric decryption by y
RA	Return address of the mobile terminal

Table 4.1: Notation for anonymous LBS scheme description

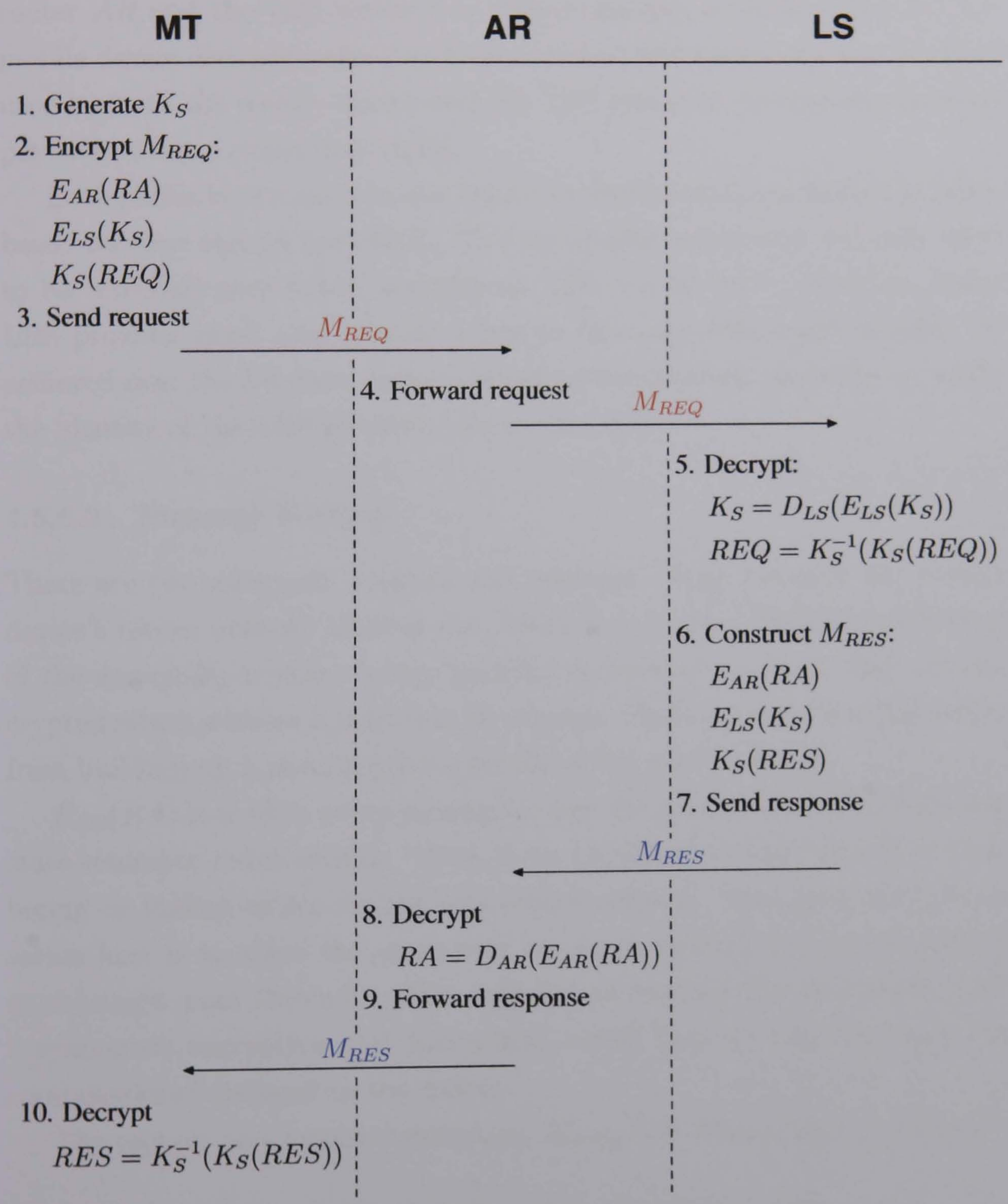


Figure 4.4: Protocol flow for anonymous LBS scheme

4.5.6.1 Key Requirements

The system requires two public/private key-pairs: one each for the anonymity router AR and the LBS server LS. When starting a protocol session, the mobile device also generates a random session-valid secret key K_S, which is used by both the mobile device and the LBS server to encrypt the message payloads using a symmetric cipher.

Both public keys must be made known to the mobile device before location-based services can be requested. The anonymity router key will only need to be retrieved once before anonymous LBS can be used. However, every LBS provider must also provide a key to the user; this could possibly be achieved over the Internet, using a standard certification hierarchy to verify the identity of the LBS provider (see section 3.3).

4.5.6.2 Message Format

There are two messages: request and response. Each contains the mobile device's return network address RA, which is encrypted with the public key of the anonymity router. Secure padding is required to ensure that the encrypted return address is unique to the session, which prevents the LBS server from building up a pseudonymous profile of the user's location.

$E_{AR}(RA)$ is used in every message to free the anonymity router from any state-retaining requirements: there is no need for message sequence numbering or lookup tables for the real return address. One potential optimisation here is to allow the anonymity router to encrypt the return address as messages pass through. This replaces an asymmetric decryption with a symmetric encryption and decryption, which may or may not place less computational demand on the router.

The request and response messages, M_{REQ} and M_{RES}, have the format:

$$M_{REQ} = E_{AR}(RA).E_{LS}(K_S).K_S(REQ) \quad \text{and}$$
$$M_{RES} = E_{AR}(RA).E_{LS}(K_S).K_S(RES),$$

where REQ and RES are the request and response payloads. The request payload consists of the location co-ordinates, and the service required; the response contains the result of the service requested. A mobile device can determine which LBS request resulted in an incoming response by comparing the two encrypted session keys.

4.5.6.3 Forwarding

The original request message is forwarded, by the anonymity router. to the LBS provider. The address of the LBS provider is specified at the network layer; at this stage, the only purpose of the anonymity router is to strip any trace of the mobile terminal's address from the message it forwards.

At step 8 of the protocol, the anonymity router must decrypt the message to find the return address of the mobile terminal. The complete response is then forwarded to this address, where the mobile terminal can read the LBS response.

4.5.6.4 Cipher Requirements

To maintain the user's anonymity, the protocol must conceal the address of the user from the LBS provider, and the location request from the network operator. At the same time, the LBS provider must be able to read the location request, and the anonymity router must be able to forward the response back to the mobile terminal.

The address must be unique for each exchange of messages, to prevent the LBS provider from building a profile of the user's location. This can be achieved with RSA by using the secure padding scheme described in section 3.1.2.4.

Encrypting the location information has further requirements. Because the network operator can measure the approximate location of the mobile user, it is possible to guess parts of the plaintext associated with the enciphered request. It is therefore vital that the encryption scheme used is robust against known-plaintext analysis. Using the Cipher Block Chaining (CBC) mode of the symmetric cipher provides this security; CBC is described in section 3.1.1.2.

We also require that message integrity is verifiable. This allows the mobile user to verify that the received LBS response originated at the LBS provider, and that it has not been altered in transit by the mobile network operator.

The final requirement is that secure ciphers and key lengths are used. This scheme recommends RSA with 2048 bit keys for the public-key cipher, and AES-128 for the symmetric cipher. These parameters can be increased if extra security is desirable, or if more efficient attacks are found against the ciphers used.

4.5.7 Protocol Analysis

4.5.7.1 Security

The protocol's simplicity makes a security analysis relatively straightforward. The request and response messages have a similar format: they consist of two instances of an RSA-KEM-like key transport, followed by symmetrically enciphered data with the same key. Therefore, if implemented correctly, the security is equivalent to that of RSA-KEM+DEM1 (which has a formal security analysis by Shoup[115]), and of the underlying ciphers. RSA-KEM is proven to be secure against chosen-plaintext attacks, and DEM1 is secure against known-plaintext attacks.

Additionally, DEM1 includes a Message Authentication Code, which provides message integrity. This scheme suggests using HMAC-SHA-256 as the MAC, but any of the algorithms specified in ISO/IEC 9797-2[66] are suitable.

A major strength of the protocol is that no public/private key pair is required for the mobile user. This removes the need for complex public-key infrastructure, which would be particularly difficult to implement with the rapidly-changing subscriber base common in mobile networks.

4.5.7.2 Anonymity

Anonymity is provided to the user under the above threat model. The mobile network and anonymity provider cannot find the user's location without colluding with the LBS provider; the LBS provider cannot find the user's true network address without colluding with the anonymity provider. The scheme is somewhat similar to a mobile optimised single-hop onion router, and provides much the same anonymity with lower overheads.

Note that the scheme described above does not provide anonymity against the LBS provider if a global passive attack can be mounted. To counter this, a MIX-like batch-and-reorder phase could trivially be added. The disadvantage to this is increased latency between request and response, which is proportional to the number of LBS requests being handled. This does not seem to be a likely attack, and so the default of this scheme is to assume it is unimportant.

In practice, the anonymity router could be run by the mobile network, as it cannot read the messages passing through it. This would be disadvantageous if the mobile network operator also offers location-based services; then, it would be trivial to collude within the organisation to trace messages. For this reason, an external anonymity provider would be a safer option.

4.5.7.3 Overheads

To estimate the percentage overhead due to encryption, we must first estimate the size of the payload. The request and response payloads consist of the location data, and the location request or response.

We assume that location is transmitted in latitude and longitude degrees (-90 to 90 and -180 to 180), minutes, seconds, and hundredths of seconds. This gives a precision of around 30cm, and latitude and longitude can be stored in 32 bits each. Therefore, a full location can be represented in 64 bits (8 octets).

Payload data are more variable in size. Location-based service responses can vary from a few characters of text to an annotated map; therefore, their size can vary from hundreds of bytes to tens of kilobytes. We will consider two cases, with payloads of 1024 bytes and 64 kilobytes, representing a series of directions, and a detailed graphical map, respectively.

The protocol overhead is defined as that of two RSA-KEM+DEM1 messages. RSA-KEM's output length is equal to the length of the RSA public key; for this analysis, we will choose this as 1024 bits (128 octets). DEM1's output length is affected by the symmetric cipher mode and the MAC chosen; for AES-128-CBC and HMAC-SHA-256, this will again be 128 octets in the worst case. Therefore, the total overhead per message due to encryption is $(128 + 128) \times 2 = 512$ octets.

In terms of percentage overhead, the protocol encryption therefore adds $512/(1024 + 8) = 49.6\%$ to a request with a 1024 byte LBS payload, and $512/(65536 + 8) = 0.8\%$ to a request with a 64 kilobyte payload. This will clearly introduce some delay into the transmission of the message, but it is so little as to be negligible in comparison to other inherent latencies in the overall communications system.

Most importantly, the protocol requires no additional messages or wait-states over a non-anonymous but otherwise equivalent solution. This makes it approximately equal in performance, while adding the benefit of fully anonymous service usage.

4.5.8 Summary

This section presents a protocol design which allows users to anonymously receive location-based information, within certain restrictions. The protocol can be used to guarantee user anonymity for user-requested location-based services with a location measurement system which is handset-based.

The cryptographic techniques used have security proofs in the literature, and do not seriously reduce the performance of the system compared to a non-anonymous scheme. This is important, as it means that the overhead due to providing anonymous services is low, which will encourage uptake.

4.6 Conclusions

The anonymity literature describes several schemes which aim to provide service-level privacy enhancements to users. Three of the most major of these are MIX networks, Onion Routing, and Crowds. Each of these schemes is described in detail in this chapter.

However, much of this work has been aimed at fixed networks, particularly the Internet. Mobile networks have very different characteristics from those assumed by these schemes. Latency is much higher, expected throughput is lower, and cost is a significant issue. For these reasons, service-level anonymity schemes cannot trivially be applied to mobile systems.

This work recommends a close assessment of the threats against anonymity in mobile systems. Some previous research into mobile anonymity has not considered the privacy danger which exists when the network service provider is trusted. Some proposals assume that the service provider can be trusted to anonymise data. Section 4.4.3 describes a generic threat model for mobile anonymity which argues that this is not a safe assumption to make.

Also presented in this chapter is a case study of a mobile anonymity scheme, applied to the new market of location-based services. This is related to previous service-level anonymity schemes, but optimised for mobile systems. A security and anonymity analysis is provided, along with an estimate of the overheads required. Against the pragmatic threat model determined to be appropriate for mobile systems, this scheme provides anonymity to LBS users without significantly impacting performance.

In summary, the previous work on service-level anonymity must be closely considered when providing anonymous services to mobile systems. The threat model presented in this chapter can be used as a starting point, which must be extended and further analysed for each specific situation. A further recommendation is to take into account the network performance of mobile systems when designing anonymous systems, as this has a significant impact on the anonymity which can be provided. Uncomplicated schemes, such as the anonymous LBS protocol described above, can offer a better anonymity-to-performance compromise than traditional, more complicated systems.

Chapter 5

Privacy-Enhanced Network Access

Contents

5.1 Introduction

Access to any communications network requires a network service provider, an organisation which offers local-link connectivity to a wider network. Examples include dial-up or broadband Internet Service Providers, and GSM or CDMA mobile network operators such as T-Mobile or Verizon. By definition, the role of the network operator is to allow a local node to send data to (and receive data from) many remote nodes. All data transferred to and from the local device must pass through this network operator.

From a privacy standpoint, this raises an immediate problem. The network operator can inspect, record, modify, and inject data transferred to each local device. The ability to inspect and record data allows the network operator access to any personal or sensitive information which its users communicate; modification and injection of data could change the behaviour of application-level protocols, leading to personal information leaking.

It is therefore placed in a position of absolute trust by default, regardless of whether or not this trust is deserved. With ever-decreasing storage costs, each transmitted message can be stored indefinitely, creating a long-term record of who each user communicates with, and what was said. Along with the message, or instead of the full message, the network operator can record relevant metadata, such as time of transmission, or in mobile systems, the user's location. These data can be analysed to find the frequency of communication, and then to group sets of discrete messages into conversations. Furthermore, long-term profiling can be used to track the user's location and communication habits, revealing still more information which many people feel should be private.

To counter this, cryptographic techniques can be used to provide secrecy and authenticity of communications for all transactions made via the network operator: secrecy prevents inspection and recording of data, and authenticity prevents modification and injection. Additionally, the service-level anonymity schemes discussed in the previous chapter can be used to provide communications privacy at the application level. However, the communication metadata is still revealed to the network provider, and it may analyse it to find usage patterns. Therefore, the network provider must be trusted by its users. Recent events have shown that this is not be a safe position to take.

5.1.1 US Cell Phone Records Released

In mid-2005, a demonstration of the privacy problems inherent with communications systems entered the public eye. Several companies in the United States were found to offer a cell phone record retrieval service: given any cell phone number, and a small fee, the company would provide a list of time-stamped inbound and outbound calls[71]. Despite the obvious personal privacy invasion, the legality of the companies' actions was a matter of dispute. These services employ a technique known as "pretexting": impersonating a customer to retrieve customer information. While this sounds to the layman like a simple case of fraud, this appears to be legal unless information is retrieved from a financial institution[45]. Although pretexting has reportedly been used by private investigators for many years, the new wider availability of the information caused a great outcry from the general public, and led to several proposals[69] of laws specifically targeted at the cell phone record retrieval companies.

This example shows that, whether through incompetence or malice, any communications provider is able to reveal private information about all of its customers. An effective solution would be to remove this capability: prevent communications providers from obtaining, recording, and therefore releasing personal information. Without the ability to abuse identity information, the motive or punishment for doing so become irrelevant. This can be achieved by enabling anonymous network access.

5.2 Towards Anonymous Network Access

With the above in mind, the potential for privacy invasion by network operators is clear. While not universal, there are a growing number of mobile network users who are uncomfortable with this possibility. Some of these users would prefer to use an alternative method of network access, and even pay a premium for it, if they could be certain that it would allow them to do so anonymously.

In order to achieve this anonymous network access, its exact meaning must be made clear. Therefore, let *anonymous network access* be defined as **a scheme for wide-area network access in which users are unidentifiable by the network provider**. We therefore explicitly distrust the network access provider, and must assume that it is attempting to determine its users' identities.

Creating an all-new anonymous network access system would require massive investment, so much as to be unfeasible. This thesis therefore aims to take current network infrastructure, along with previously-discussed future network architectures, and enhance them so that these privacy-conscious users can subscribe to a service which offers them complete privacy from their network operator.

5.2.1 Wireless Networks

The recent increase in wireless network usage is a mixed blessing for personal privacy. On one hand, the trend towards ubiquitous, always-on connections implies greater network usage, which in turn makes statistical profiling of an individual much more feasible. However, some properties of wireless networks open new possibilities for privacy protection.

Unlike in fixed networks, wireless network users have no physical connection to their network access provider. This fundamental difference means that users are able to change service provider whenever they desire: for example, by changing the SIM card in a GSM/UMTS handset. From this, the possibility arises of continually changing service provider to mitigate long-term profiling.

While this is a privacy-enhancing concept, it would only mitigate the ability to detect users' usage patterns. If the user must still use a SIM card to authenticate to the network provider, pseudonymity is all that is possible. The SIM is a constant identifier which can be matched to customer records to find the identity of a user for any given connection. Therefore, the user's identity must be revealed to the network provider in order to gain access to the network. This obviously prevents user anonymity.

5.2.2 Wireless Hotspots

One example of potentially-anonymous network access is using open 802.11 "WiFi" hotspots. All 802.11-compliant wireless base stations have at least some security functionality in WEP[73]; however, it is not universally used. A 2004 world-wide survey of 228,000 access points[61] revealed that 61.6% had no access restrictions at all[40]. Many of these can be used to anonymously access the Internet, with or without the consent of the owner.

There have been several cases of legal proceedings taken against people who use unsecured wireless access points without permission[80][78]. However, there are also many wireless access points which are advertised as being free for all to use; in pubs, restaurants, cafés, and public parks. In many countries, there are ongoing trials of publicly-funded free wireless access in urban areas[131][6].

5.2.2.1 Anonymity Provided

Because the wireless networks are unsecured, no credentials whatsoever are required to access them. Therefore, access is granted to anyone with a capable WiFi device. This is an important step towards anonymous network access: the ability to connect to a network without revealing any identifying information.

In these circumstances, the user's identity is not necessarily explicitly revealed at any point during network usage. There remain application-level mistakes that the user can make, which would accidentally reveal their identity. For example, the user should avoid the usage of authenticated services, such as webmail or VPN clients. Alternatively, such usage could go through a service-level anonymity scheme. If this is done, and the network is used for purposes which could be attributed to anyone, anonymous usage is possible.

5.2.2.2 Disadvantages

However, there is still the potential to perform long-term profiling of users, if they use the same (or affiliated) access points on a regular basis. All 802.11 network interfaces have a fixed IEEE 802 MAC address, which can be used as a key to relate two separate sessions to one user. Practically speaking, this is less of a problem than SIM-based session linking. Open wireless hotspots tend to be independent and run by small companies or individuals. Therefore, there are fewer resources available to carry out tracking, and in many cases tracking over multiple different locations is impossible. However, looking forward, we may see more and more government-run free wireless networks, at which point session tracking will clearly become more of an issue.

There are other problems with depending on free WiFi hotspots for anonymous network access. Perhaps most important is the lack of coverage. At the moment, WiFi hotspots are obviously more common in highly-populated urban areas than in rural locations. Access points also have highly limited range: ordinarily around 100 feet, which degrades rapidly with physical obstacles such as walls. This means that hotspots tend only to be usable indoors, within the same building as the access point. The final technical concern is that there is currently no way to seamlessly handover between two independent WiFi networks, so client mobility is very much restricted.

From an economic standpoint, there is one further issue. Users can have no guarantee or expectation of reliability from a free wireless access point. This is important, because 802.11b performance rapidly degrades as network utilisation increases[48]. This factor is directly related to the lack of payment for service: there is no economic incentive for the operator to improve the network to increase reliability, as long as it works most of the time.

5.2.2.3 Summary

Open wireless hotspots are one possible method of achieving anonymous network access, with technology that is widely available today. There are several advantages to this method of network access. These include:

Cost Open WiFi hotspots are free to use;

Anonymity No identification is required to access the network;

Throughput 802.11b provides enough local throughput to match most broadband Internet connections.

There are also disadvantages to relying on wireless hotspots for anonymous network access. The most important of these are:

Linkability Client hardware address can be used to link multiple sessions, removing some anonymity;

Coverage Hotspot range is limited, and geographic coverage is therefore fairly low;

Mobility No ability for inter-network handover means mobility is restricted;

Reliability Users cannot expect service to be fully available, as it is free.

This example highlights desirable properties of a sustainable anonymous network access system. While initially attractive, open wireless hotspots do not meet all of these requirements, and therefore we should seek a better solution.

5.2.3 Requirements for Anonymous Network Access

From the example above, we can derive three key requirements of any system which aims to provide anonymous network access. These are *reliability*, *unidentifiability*, and *unlinkability*. If all of these requirements are met, there would exist an anonymous network access system which is a viable functional replacement for current mobile access networks.

5.2.3.1 Reliability

The most crucial failure of wireless hotspots as an anonymous network access mechanism is their availability. Each hotspot tends to be isolated: access points are fairly sparsely distributed; there are very few public wireless networks outside built-up urban areas. Additionally, there is no support for mobility: moving between multiple networks is not supported, and most free access points are not part of a single federated network.

Further, neither availability nor quality of service are guaranteed. If the free hotspot stops working, there's nothing that a user can do but go elsewhere and try to find another network connection. As there is no direct revenue stream from providing the hotspot, it may not be a high priority for investment from the provider. Therefore, reliability is always going to be an issue.

But to provide viable anonymous network access, the network must be reliable. Coverage, mobility, and quality of service must be of a level which is acceptable to potential users, or they will not use it. Without some source of revenue, there is no way to invest in the service provision, and widespread reliability then becomes difficult to ensure. An obvious solution is to charge users for access at the point of usage.

In summary, for a network access mechanism to be reliable, it must have three features. These are:

Coverage Network access must be available over a wide area, and in a majority of urban locations;

Mobility Users must be able to use the network service while travelling;

Quality of Service Sustained availability of the network connection must be guaranteed.

These features all create expense for the network operators, which means that users must be able to pay for network service. This leads on to the next requirement, as payment must be made while maintaining unidentifiability.

5.2.3.2 Unidentifiability

There are two main properties of anonymity that an anonymous network access scheme must have. This first requirement states that the network service provider should not be able to identify its users. Simply put, anyone should be able to access the network without the provider knowing who they are. This has several meanings.

Traditionally, a user must authenticate to their service provider to access a service. This is normally achieved using a shared secret, such as a user name and password, or simply a unique subscriber number. The service provider validates the secret given by the user against its records: if successful, this will identify a unique user account. This is one form of identification: determining which one of all known users is using a particular network connection.

Another form of identification is determining who this unique user is in the real world. For paid services, this can easily be achieved using billing information, such as a credit card, or contact information, such as a home address. This thesis defines personally identifying information as anything which can be used to reasonably distinguish a person from the set of all other people.

Therefore, an anonymous network access scheme requires an anonymous payment scheme, which does not allow the service provider to identify its customers. It must also be usable without any other form of identifying information, such as a unique user account.

5.2.3.3 Unlinkability

The final requirement is another property of anonymity, unlinkability. Specifically, this means that the network provider must not be able to correlate any two user sessions. A user should be able to use the network without revealing any constant information that allows this to happen.

We must exclude anything outside the immediate domain of network access here. That is, while the user should be able to access the network without revealing any linkable information, this does not mean that they must retain anonymity when sending and receiving data. Application-level anonymity schemes must be employed if the user wishes to make use of identifying services without revealing this information. The choice of which scheme is most appropriate is outside the scope of the network access scheme.

There are many possible data which would create links between sessions. For example, as mentioned in the previous section, a requirement for a unique user pseudonym would create a very clear link. Effectively, any data which leaves the user's mobile terminal must be subject to scrutiny to ensure that it does not leave a trail over multiple sessions.

5.2.3.4 Summary

To summarise, there are three key requirements for any anonymous network access scheme. These are:

Reliability Network access must be paid for to ensure availability and quality of service;

Unidentifiability Users must be able to pay for and use network service without being identified;

Unlinkability Network providers must not be able to correlate two sessions to one user.

5.3 Digital Marketplace

5.3.1 Introduction

The first anonymous network access requirement, reliability, can be met by existing cellular telephony networks. In the UK, cross-operator mobile cell coverage exceeds 95% of land area, and mobile handsets are capable of seamless handoff between cells. There is a robust and well-tested mechanism for paying for network access, and so quality of service guarantees can be offered.

Unfortunately, the other two requirements cannot be met. Current-generation cellular telephony systems, such as UMTS and CDMA2000, are founded on the principle of uniquely-identified users. Authorisation is achieved using a shared secret protocol for mutual authentication: the user can verify the network's identity, as the network can verify the user's identity. This principle of identification-for-authorisation cannot easily be changed within present call management architectures.

Another key problem is that the current system assumes a long-term contract with one network provider. A unique account is set up with one network operator for the period of the telephony contract, normally one year or eighteen months. While this does not (in itself) make linkability trivial, it does make it easier: the size of the anonymity set is reduced from all possible users of any network, to all possible users of one network. Ideally, the user's network provider should change frequently, to make it more difficult for any given network operator to correlate multiple user sessions over time.

One proposal for a next-generation call architecture retains the strengths of cellular telephony, while opening the possibility to solve the problems of anonymity. This is known as the Digital Marketplace, and was first proposed by Le Bodic et al. in 2000[77]. The original motive behind the Digital Marketplace proposal was entirely unrelated to privacy: its primary goal was to increase efficiency.

5.3.2 Future Directions in Service Provision

As the mobile telephony market slowly migrates to the third-generation architectures, new business challenges are becoming evident. Customers are used to fairly constant prices, and are on the whole uninterested in paying extra for new services. While text messaging is still extremely popular, WAP, picture messaging, video calling, and downloadable content have all dramatically failed to meet sales expectations. For example, in the UK in 2005, approximately 550 messages were sent by each active mobile customer on average; of these, only 0.8% were multimedia messages[93].

At the same time, mobile service providers are expected to maintain quarterly growth, as are all publicly-traded companies. The mobile telephony market is in saturation: in 2005, the UK had 59.6 million residents[120] and 62.5 million active mobile phone subscriptions, with around 80% of UK adults using mobile phones[94]. There is therefore an urgent need for new revenue streams if service providers are to continue to grow. This has led to the rapid development of still more advanced services: for example, location-based maps and directions, music downloads, and streaming video.

One problem with these services is that they all require a significant amount of data to be transferred. As of June 2006, the average end-user

3G data fee on a UK network is around £1 per megabyte (based on prices from Vodafone[127], O2[92], and Orange[95]). Services which require multiple megabytes per session are therefore going to be prohibitively expensive to use, without even considering the cost of the service on top of the data transferred. This will clearly limit the uptake of new services, unless the cost of throughput can be decreased.

5.3.3 Improving Network Efficiency

In Scottish economist Adam Smith's seminal work, *The Wealth of Nations*, he argued that a freely-operating market is the most efficient way to allocate scarce resources while satisfying both producers and consumers[117]. Currently, cellular networks are more accurately described as an oligopoly: competition is extremely limited, the ability to choose supplier is restricted, and the barrier to entry is high. If Smith is correct, this means that the available radio resources are not being used as effectively as possible.

It is upon this principle that the Digital Marketplace (DMP) was designed. The DMP aims to provide a freely-operating market for network service. This should increase resource usage efficiency, and therefore lower costs. This is done by providing a market environment in which auctions can be carried out for network service on a per-call basis. With correct design and implementation, this market will be freely-operating, and the result should be positive for both network service providers and their customers.

5.3.4 Requirements for a Freely-Operating Market

Creating a dynamic market for network service is not difficult, but ensuring that it operates freely is extremely challenging. If this is not achieved, the Digital Marketplace cannot achieve its goal of increasing efficiency, and it will not be able to offer anything useful. There are three core requirements for a marketplace to be freely-operating:

1. Accessibility: Producers and consumers must be able to buy and sell freely;

2. Transparency: Service must be accurately described;

3. Conformance: Consumers must be able to trust that service will be provided as agreed.

5.3.4.1 Accessibility

The first requirement is perhaps the most obvious one in any market environment. For a given marketplace, every producer must be able to offer service to every consumer, and every consumer must be able to purchase service from any producer. Another consequence of this requirement is that the market must have a low barrier to entry. This is to encourage more providers to offer services, creating competition, and lowering prices.

This is fundamentally different from how mobile network access works at present. Currently, a user has a contract with a network operator, who provides service throughout the country. Where the network operator has no coverage, the user cannot access the network. Lowering the barrier to entry, and allowing the user to use different providers, means that availability of network access will increase.

5.3.4.2 Transparency

For one network provider's service to be substitutable for another's, there must be an agreed ontology for describing the parameters of the service provided. The consumer must offer a specific description of the service required, which must be general enough to cope with the possibility of different network providers using different underlying network types. This description of service is the basis of what the Digital Marketplace calls a 'flow contract', and it is these contracts which are the target of bids from network operators.

The Digital Marketplace proposal for flow contracts includes three core parameters which define the required connection quality: bit rate, bit error rate (BER), and delay. However, these terms alone are insufficient to describe a contract for mobile wireless network service: they do not take into account the variability of the environment, and resulting loss of quality. Three additional parameters are used to specifiy this: degradation allowance, sampling rate, and monitoring period.

With these parameters specified, the network providers are able to estimate their commitments, and offer a price to the user. As all network operators are bidding on the same contract, the user may select the best bid accordingly.

5.3.4.3 Conformance

One of the key initial problems with the Digital Marketplace is that consumers are expected to purchase services from producers with whom they have no prior relationship. While there can be some punishment or compensation expected for breach of contract by a network provider, this is of little interest to the user, who simply wants to use a reliable network.

For this reason, the Digital Marketplace uses a reputation system to keep track of network provider reliability. This information can then be used by the user's negotiator as appropriate, to offer some expected probability of conformance. By introducing reputation into the bid selection process, we automatically give reputation a value, and create another incentive for network operators to meet their contractual agreements. This also offers users peace of mind that they are dealing with a reputable provider.

5.3.5 Anonymous Digital Marketplace Access

Several properties of the Digital Marketplace make it an attractive option for enabling anonymous network access. The open marketplace raises the possibility of a user negotiating directly with access network operators. By using different network providers for each call, the user could potentially achieve unlinkability. At the same time, the flexible free-market environment creates a sustainable environment for users to purchase network service without revealing their identities.

While not all of the requirements specified in section 5.2.3.4 can be met with the original proposal, there is potential for some changes to be made to the DMP to correct this. Each of the three requirements raises particular issues, which are addressed in turn in the remaining chapters of this thesis.

The first concern is for reliability. One of the most important aspects of the Digital Marketplace is that it benefits from reusing the pre-existing mobile access network infrastructure. This certainly makes reliability possible; in aid of this, the original proposals include support for quality of service measurement, and a network provider reputation scheme to give a financial incentive for reliability. However, one concern with this aspect of the Digital Marketplace is that there has previously been no examination of the security issues involved in such a market environment. Until these are examined, presented, and addressed, there can be no guarantee that the market will operate securely. Securing the Digital Marketplace is crucial to ensuring that it can feasibly be used for reliable network access.

Both of the remaining requirements deal more specifically with anonymity itself. The current Digital Marketplace design does not explicitly support unidentified access, and therefore modifications are required to enable this. Such modifications must of course take into account the security problems that arise from providing anonymity. The final requirement, unlinkability, has also not been considered in depth. Possible long-term linking attacks against anonymity must be examined to ensure that the modified DMP can indeed provide anonymous network access.

5.4 Summary

Network service providers are placed in the ideal position to gather private information about their customers. Recent events have shown that this position of trust may be unwarranted in some cases. Some privacy-conscious customers would prefer an alternative solution, in which there is no single entity with such power over their personal information. Such a solution would be known as *anonymous network access.*

To achieve sustainable network access with anonymity, three requirements must be met: *reliability*, *unidentifiability*, and *unlinkability*. Traditional mobile network access mechanisms, such as GSM and UMTS, can provide only reliability; the other two requirements would require major changes to call management systems. A possible alternative is open WiFi hotspots, which can meet the second and third requirements; however, the requirement of reliability cannot be met without a secure, scalable, and universal payment system.

The Digital Marketplace is a future market-based call-management architecture. Although the original scheme cannot necessarily meet any of these requirements, it has the potential to meet all three. By providing a means of payment, it enables reliability; by separating the service provider from the network provider, it allows unidentifiability; by opening a wide marketplace, it permits unlinkability.

Meeting these requirements in full demands two areas of original work. First, to ensure that sustainable, reliable service is possible, it is necessary to examine and counter the security issues in the Digital Marketplace protocols. Second, to enable anonymous usage of the Digital Marketplace, all threats to privacy-conscious users must be determined, and modifications to the operation of the DMP must be proposed, such that the market still operates fairly while allowing customers to retain anonymity. These two areas of work are described in the remainder of this thesis.

Chapter 6

Securing the Digital Marketplace

Contents

6.1 Introduction

The Digital Marketplace proposal opens network access provision to a wider number of sellers, in order to increase competition and improve efficiency. Instead of being tied to one access provider, mobile network users can use the one most appropriate for their connection requirements.

This fundamental change in operation also changes the organisation of mobile network access. Instead of having one constant network service provider, the ordinary user may interact with many different service and network providers. Additionally, these interactions will often be fleeting: a network operator may only provide network service to a user once, and never interact with them again.

With the increase in number of participants, and the limited duration of the relationships, it is expected that new security issues will arise. If the marketplace is to be secure, these must be identified and addressed. These steps are necessary for the market to operate successfully. If a serious flaw in the security of the marketplace exists, the free operation of the market can be distorted, and the Digital Marketplace will not be attractive to users, service providers, or network operators.

This chapter first presents the organisation of the Digital Marketplace, as previously proposed. This is followed by a trust analysis, from the perspective of each participant, and a model of potential adversaries. There follows a description of the three key Digital Marketplace protocol components: session initiation, contract negotiation, and session termination. Each of these is analysed in context for possible security threats, from which are derived a series of requirements for secure operation. Finally, a series of security measures to meet the requirements are proposed, thus securing the Digital Marketplace.

6.2 Digital Marketplace Organisation

The Digital Marketplace proposal envisages a number of individual marketplaces, each serving a limited geographical area. Each marketplace consists of several participants: an administrator and several network operators remain engaged permanently, and mobile users and their service providers join to negotiate and leave when finished.

One of the main principles of the Digital Marketplace is this division of the traditional mobile network service provider role. Currently, a mobile network user holds a contract with a service provider, which ordinarily has its own access network. More recently, the concept of Mobile Virtual Network Operators has arisen: a service provider without an access network of its own, but with contractual agreements with network providers to use their access networks. The DMP takes this a step further, and divides the role of the network service provider into two parts: the user-focused service provider, and the access network operator.

In marketplace interactions, the service provider negotiates with the network operator on behalf of its users. While not strictly necessary, the service provider does have a useful purpose: it can aggregate demand, charge users a fixed rate, and benefit from dynamic pricing from network operators. On the other hand, users may choose to represent themselves in the market place, trading off potential savings against variable cost to use the network.

6.2.1 Agents

The main practical difference between current call management strategies and the Digital Marketplace is real-time negotiation. Because connections must be established quickly, auctions for network service cannot be handled manually. To negotiate quickly, the DMP proposal therefore requires some representative for each of the actors in the transaction. The recommended technology in previous Digital Marketplace work is the intelligent agent[132].

6.2.1.1 Agent Overview

An agent is an encapsulated piece of software which communicates using standard grammars and protocols. In practice, an agent is often used as a representative, acting on behalf of a person or a corporation. However, agents are really just another level of abstraction in software design. Instead of thinking primarily in terms of sequential instructions (imperative design), or data structures and relationships (object-oriented design), agent design focuses on environmental perception, goals, and using available abilities to achieve those goals.

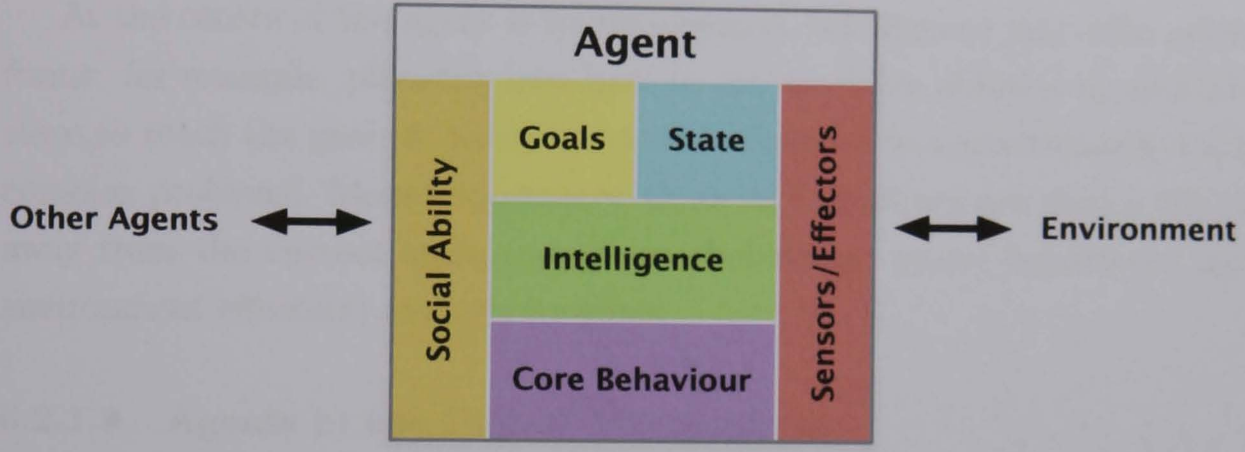


Figure 6.1: Conceptual components of an intelligent agent

A clear distinction between agents and intelligent agents is made by [132]. An agent is simply an autonomous actor in an environment; in contrast, an intelligent agent displays flexibility. This flexibility can be achieved through reactivity, proactiveness, and social ability; intelligent agents have some mixture of each of these properties.

Intelligent agents can be broken down into six main conceptual components (see Figure 6.1). At the base of the agent is its core behaviour: this is the task-specific functionality which allows the agent to achieve its purpose. At the highest level of abstraction are two things: the goals, and the current state. The goals of the agent may be quite simple, or very complex; there may be multiple possible states in which the agent is meeting its goals. State encapsulates both internal properties of the agent, and information gathered from the environment.

Every agent has an environment, and many have sensors (to retrieve information from the surroundings) and effectors (to modify the environment). These are linked most closely to the state component and the core behaviour of the agent. Additionally, agents within multi-agent systems can make use of social interaction to achieve their own goals with the assistance of their peers. Agents can communicate to share their capabilities, present their goals, request assistance, or negotiate with competing agents. This messaging is done through the social interface to other agents, and uses standardised message types, protocols, and content languages.

At the centre of the agent is its intelligence. Intelligence may take many forms: for example, planning how best to use available abilities in multiple steps to reach the goal; or using heuristic techniques to approximately solve complex problems. Most importantly, these AI techniques are used to move away from the current state (using core behaviour, social behaviour, and environment effectors) towards its goals.

6.2.1.2 Agents in the Digital Marketplace

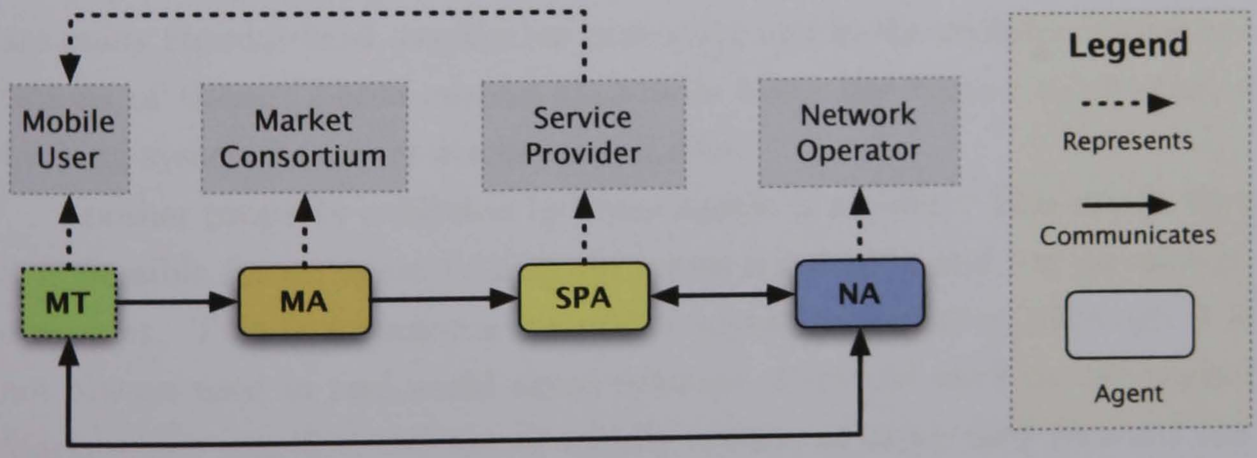


Figure 6.2: Agent organisation in the Digital Marketplace

Each actor in the Digital Marketplace has an associated agent: mobile users, service providers, and network operators (see Figure 6.2). In addition, a facilitator agent is required for administrative purposes; this agent represents an impartial Digital Marketplace Consortium, known as the Market Operator. Therefore, the acting agents in a single marketplace scenario are the user's Mobile Terminal (MT), the Market Agent (MA), the Service Provider Agent (SPA), and the Network Agents (NA).

Agents are a particularly appropriate technology for the Digital Marketplace for several reasons. The six components of an intelligent agent are a close fit for the needs of any user of a freely-operating market. Additionally, some features of agent execution platforms and current middleware implementations match other requirements which are more specific to the Digital Marketplace.

The abilities of an agent which are of most importance here are social communication and intelligent goal-achieving. All agents must send and receive messages in the marketplace, and each agent must employ other agents' abilities to achieve its goals. Each agent must act in the interest of its owner; in the market context, this means that the service provider agent and the network agents must come to an agreement which maximises utility for all parties.

Therefore, it is important to note that agents are an excellent choice for performing negotiations among several competing participants. There are many standardised negotiation protocols, and freely-available implementations of these. These current standards lower the barrier to creating a working system based on market negotiation.

Another property exhibited by some agents is mobility. This means that it is possible for an agent to migrate across a network, and run on another computer. This is a common feature of agent architectures, although it is not always used in real-world agent systems. Previous work on the Digital Marketplace specified the use of mobile agents, in order that national service providers are able to send representative agents to local marketplaces. Although not experimentally verified, this choice is expected to reduce negotiations latency when compared to long-distance messaging.

6.2.1.3 Standards

If an agent is to be able to interact with other agents in a social context, there must be some common language for such interaction. Clearly, standardisation is a key issue in the success of multi-agent systems. The Foundation for Intelligent Physical Agents (FIPA) is a Standards Committee of the IEEE Computer Society, which publishes specifications for many aspects of agent behaviour and communications.

FIPA specifications are available for free, from the committee web site[51]. The standards are grouped into five categories: agent communication, agent transport, agent management, abstract architecture, and applications. Of most direct interest to this thesis are the communication and agent management sections.

Communication specifications cover messages, protocols, communicative acts, and content languages. An Agent Communication Language (ACL) is specified, which allows any two FIPA-compliant agents to send messages to each other in a commonly-understood format. Also of direct interest are the FIPA standard protocols, which enable interoperability between agents in common interaction scenarios.

Two of the most important parts of the agent management specifications are the Directory Facilitator (DF) and the Agent Management System (AMS). The DF can be used to allow agents to locate and communicate with others in the same environment. Agents should register their available services with the DF; then, clients can search the DF for agents which offer particular services. In response to such queries, the DF returns the name of an agent; the AMS is used to resolve this name into an agent address, which can then be used to directly communicate.

6.2.2 Protocol Overview

There are essentially five stages in the contract negotiation procedure, as shown in Figure 6.3. These are:

1. User's Mobile Terminal (MT) sends a connection request to the local

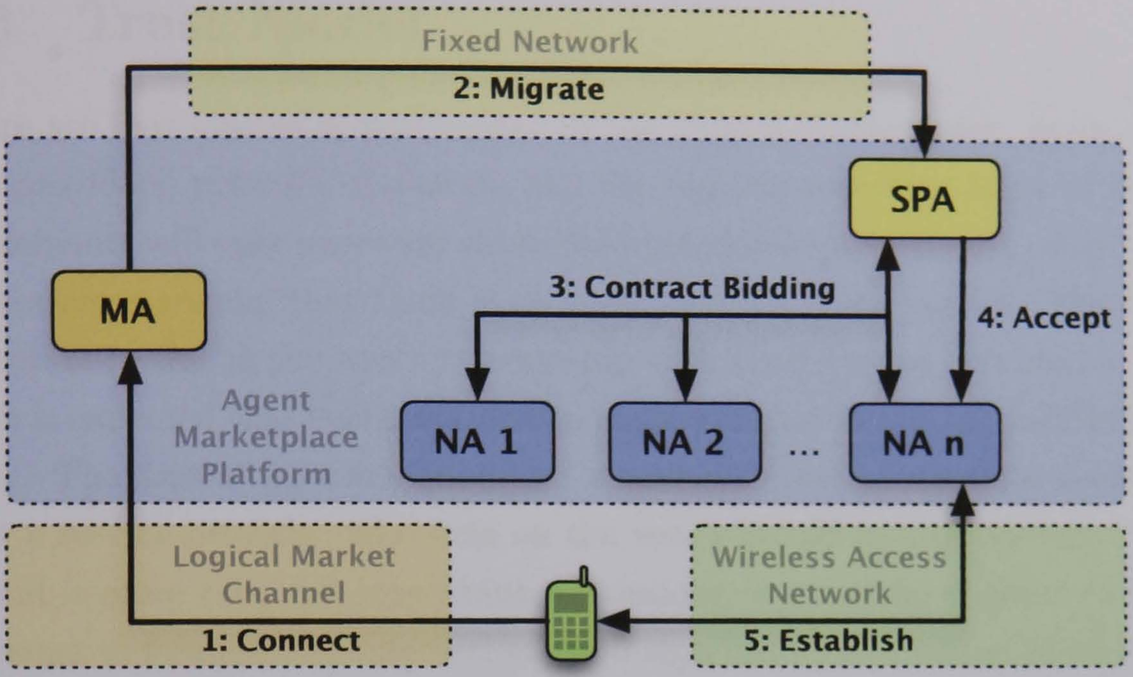


Figure 6.3: Simplified negotiation protocols for the Digital Marketplace

marketplace environment using a publicly-accessible Logical Market Channel

2. Market operator (MA) forwards this request to the addressed service provider, which migrates a Service Provider Agent (SPA) into the agent marketplace environment

3. SPA distributes the user's flow contract requirements to all network operator agents (NAs), which respond with bids

4. Bid selection is made by the SPA, and the winning NA is informed of contract acceptance

5. Winning NA establishes a network connection to the user over the local wireless network

Once the session is established, the user is able to use the network for whatever purpose they desire. The same procedure could be used to set up a registration and paging contract, which would allow incoming calls to be received by the user.

6.3 Trust Model

There are four classes of participant in the Digital Marketplace: users, service providers, network operators, and the market operator. Most of these participants will only have very short-term relationships with each other, and we therefore assume that there is no trust between these parties. The only exception to this is the user's relationship with their service provider.

It is expected that there will be two classes of user in the Digital Marketplace. The first is the most common: a user who has a long-term contract with a service provider, who acts on the user's behalf in negotiations. The second is more rare, but important to consider: a user who chooses to self-represent in marketplace interactions, and thus has no interaction at all with a service provider.

The first class of user clearly has a relationship with a service provider, the terms of which are set out in a legally-binding contract. It is then fair to assume that there is some trust between the user and the service provider. More explicitly, the user trusts that the service provider will provide service to the best of its ability, and bill fairly and accurately, just as is the case in traditional telephony contracts. In the other direction, the service provider only trusts that the user will pay for service as billed.

Until the market interaction protocols are demonstrated to provide security features, we cannot assume any other trusting relationships exist. This is the safest possible position to take, from a security perspective: being so paranoid as to believe that all other actors in the market are trying to win at all costs. With this trust model in mind, the adversary model becomes clearer.

6.4 Adversary Model

In this section, each participant class is characterised according to the model proposed by Salter et al.[109]: in terms of their resources, risk tolerance, and objectives. Resources restrict what an adversary is able to do; risk tolerance determines what an adversary is willing to do; and objectives describe what the adversary wants to achieve. Without a model of the expected adversaries, it is impossible to build a solid threat model. Once adversaries are assessed and prioritised, the most important threats are more easily dealt with.

The adversary model is closely tied to the trust model. With the exception of the user who has a service provider, each participant must view all other participants as potential adversaries. Users with service provider only trust their service provider, and offer no trust to any other party.

Adversary	Resources	Risk	Objectives
Market Operator	High	Very Low	None known
Service Provider	Medium	Medium	Increase revenue
Network Operator	Medium/Low	Low	Harm competitors
Mobile User	Low	High	Reduce costs

Table 6.1: Adversary resources, risk tolerance, and objectives characterisation

As can be seen from Table 6.1, the resource and risk characteristics are defined as relative to other adversaries. The primary goal here is not to determine exactly which attacks are possible, but to examine which of the participants are most likely to be attackers. Therefore, knowing that the network operator has far fewer resources available than the market operator, but equal risk tolerance, implies that the market operator is overall a more important adversary to consider.

6.4.1 Resources

At first glance, it might seem that the market operator should be considered to be a neutral party. The DMP Consortium would be a non-profit entity, which has no financial interest in marketplace transactions. Therefore, it has nothing to gain from distorting the market operation.

However, it is important to recognise that the market operator should not be completely disregarded because of this seeming lack of motive. While the likelihood is very low, it still may act in collusion with another party, and it has exceptional power over the marketplace. The market operator has control of the agent environment in which negotiations take place. If it is allowed to abuse this position, other participants will not trust the market. Therefore, we must consider the market operator to be an unlikely but very powerful adversary.

Attack resources are otherwise based on the estimated financial resources of each entity. This in turn is expected to be related to the number of each class. There are likely to be relatively few service providers, and they are likely to have large financial resources (due to the relatively high cost of marketing to attract customers). Comparatively, there will be a large number of small network operators, as the barrier to market entry is designed to be low; each of these will have significantly less revenue to work with than the service provider. Finally, there will be a large number of individual mobile users, most of whom will have almost no capital to invest in attacks. Therefore, the service providers are likely to have the most financial resources, followed by the smaller network operators, and the individual mobile users.

6.4.2 Risk Tolerance

The market operator's risk tolerance is extremely low. If any participant has reason to suspect that the supposedly-neutral consortium is actually acting unfairly, the market will be undermined. This is why we characterise it as a very unlikely adversary.

As the service provider is entirely dependent on its users for revenue, it has a great deal to lose if the user detects any unfair behaviour. However, this could plausibly be blamed on a billing error, and as long as the user is compensated, little is likely to be lost. In comparison, the network operator has only a very fleeting relationship with the user. If they are caught cheating, the Digital Marketplace's reputation system will come into play, and the network operator will subsequently find it difficult to gain service contracts. Finally, other mobile users have nothing to lose from manipulating the marketplace to their advantage, so their risk tolerance is high.

6.4.3 Objectives

The most likely objectives of each attacker are obviously financial. Service providers and network providers both want to increase profit; both could do so by padding the true cost of the call, over-billing the user or the service provider, respectively.

Network operators also have the objective of increasing revenue. This is achieved by being selected by service providers more often than their competitors in the marketplace.

Malicious mobile users would prefer to make use of network service for reduced cost, or even free. Another potential objective could be to act in collusion with a network operator to damage its competitors, or in collusion with a service provider to defraud a network operator.

6.4.4 Conclusion

It is clear from this model that the least dangerous adversaries are likely to be mobile users. Larger network operators and service providers have the most to gain, and the most resources with which to launch attacks. The remaining medium-scale attackers have higher risk tolerance to counter their lower resources. Finally, we can almost ignore the market operator as a practical adversary, but must remember that it will not be completely trusted by the other participants.

6.5 Detailed Protocol Operation

Earlier work outlined the Digital Marketplace protocol operation for outbound communications sessions[63]. The overall protocol flow is shown in Figure 6.4. Other versions of the protocol exist, for example for registration and paging contracts, but they are equivalent for our security analysis: the only difference being the wireless network activity between steps 7 and 8. For clarity of analysis, the protocol can be separated into three logical components: session initiation, contract negotiation, and establishment to termination.

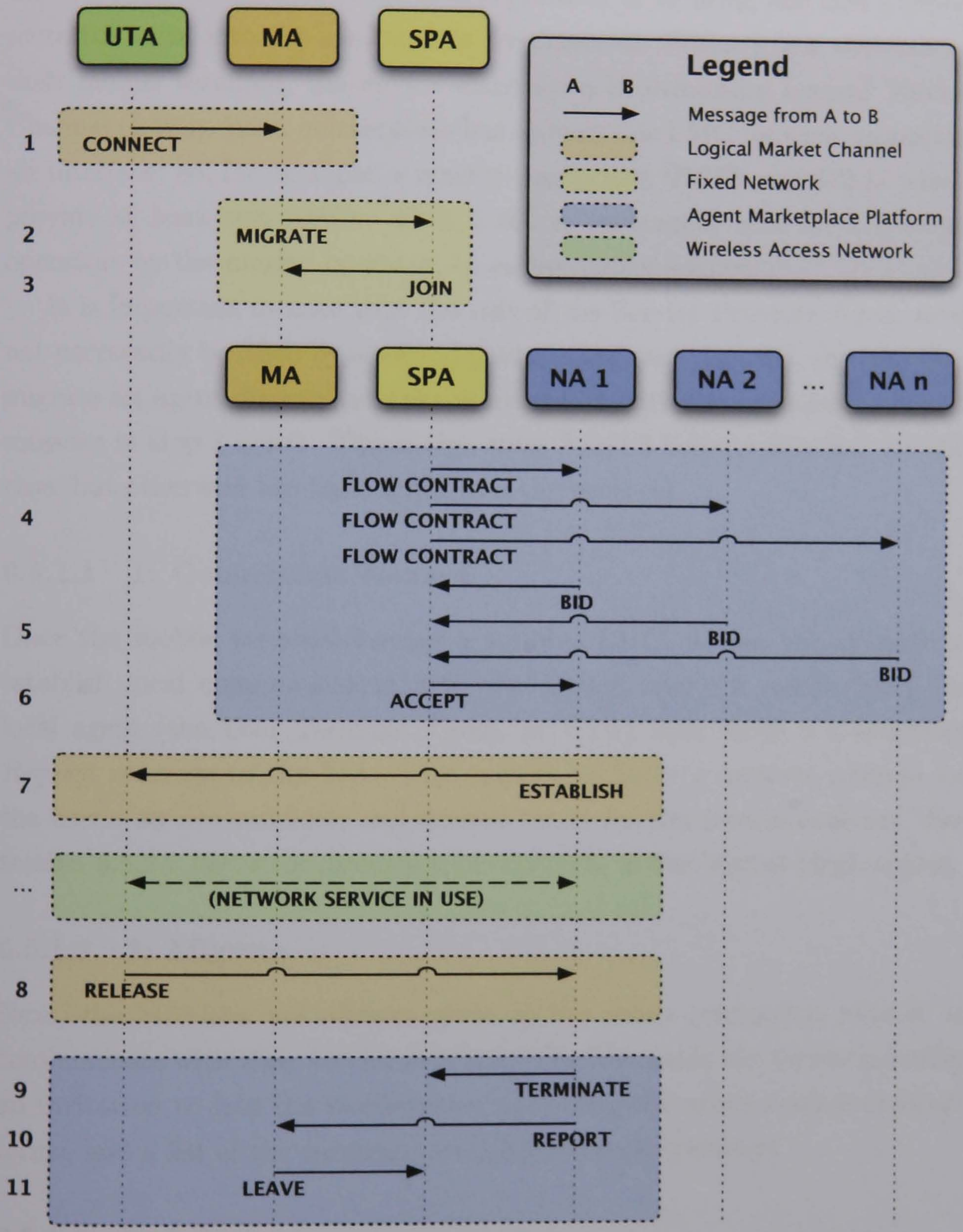


Figure 6.4: Digital Marketplace protocol operation as published in [63]

6.5.1 Session Initiation

The purpose of this first stage of the protocol is to bring the user's representative agent into the marketplace environment. When a user switches on their mobile terminal, the device scans for a broadcasting Logical Market Channel (LMC). Each marketplace has at least one LMC for each supported air interface: so, for example, a market supporting UMTS and 802.11 would provide at least two LMCs. Each LMC is contracted with local network operators by the market operator, to ensure independence.

It is important to note that the role of the Service Provider Agent need not necessarily be filled by a second-party to the user. Instead, the user may migrate an agent directly into the marketplace with the connection request message in step 1 below. This makes steps 2 and 3 below redundant for this case, but otherwise has little impact on the protocol.

6.5.1.1 1: Connection Request

Once the mobile terminal locates a suitable LMC, it uses the channel to establish local communication with the market operator agent (MA). Its local agent (the User Terminal Agent, or UTA) then sends a *Connection Request* message to the MA. This request includes a network address for the user's service provider, and desired terms for the session contract. See section 6.5.2.2 below for discussion of contracts in the Digital Marketplace.

6.5.1.2 2: Migrate

Next, the MA uses the address given in the user's connection request to communicate with their service provider. The MA sends the service provider an invitation to join the marketplace, including the user's session contract terms, and a list of the currently available network operators.

6.5.1.3 3: Join

In response, the service provider migrates an appropriate negotiating agent to the marketplace. This agent is called the Service Provider Agent, or SPA.

It establishes itself in the agent environment by the transfer of compatible agent code across the network, which is then executed in the local agent container.

6.5.2 Contract Negotiation

Once the SPA has migrated to the marketplace, negotiation can begin. Multiple negotiations can take place at one time, even to fulfil only one session contract. Negotiations consist of an *auction* for *flow contracts*: each of these terms is described below.

6.5.2.1 Auction Format

The auction format chosen for the Digital Marketplace is the *sealed-bid, first price* scheme[75]. This was selected over three other major auction types: *sealed-bid, second price*; *open-outcry*; and *descending price.*

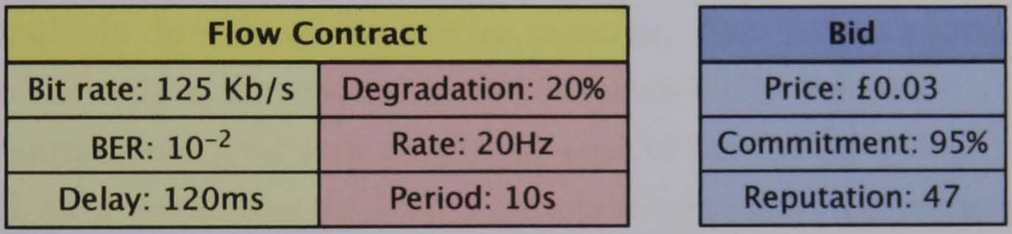
A *sealed-bid, first price* auction is very simple: all bidders place a bid for the price they are willing to pay, and the best bid wins. Bids are sealed, which means that other bidders cannot read them and respond with their own bid. This auction type is most suitable when a quick auction is important, as the only delay is due to bidders deciding on a bid value. The *sealed-bid, second price* auction is identical, except for the amount paid. The bidder with the best bid still wins, but only pays the second-highest bidder's price.

An *open outcry* auction (also known as an English auction) starts at a minimum price, known as a reserve. Bidders then openly announce the maximum price they are willing to pay for the goods, and are permitted to place a new bid in response to competition. Similarly, a *descending price* auction (also known as a Dutch auction) works in reverse: the goods start at a high price, which is lowered in small increments until the price is acceptable to one bidder.

These latter two auction types have several iterations, are therefore unsuitable for the Digital Marketplace due to the resulting extra latency during call set-up. The *second price* auction is also inappropriate, but for a different reason. The bids placed by network operators can be weighted in many ways: favouring reputation over price, or price over reputation. Therefore, the second-highest price depends on how the bid parameters were weighted, and cannot be decided by anyone other than the SPA. For this reason, the only fair price to pay is the first price: the price offered by the winning bidder.

6.5.2.2 Contract Types

Auctions are carried out to agree upon network service contracts. The Digital Marketplace has two fundamental contract types: the session contract, and the flow contract. One flow contract is the subject of multiple bids in each auction. An example flow contract and a single bid are given in Figure 6.5.



Flow Contract	
Bit rate: 125 Kb/s	Degradation: 20%
BER: 10^{-2}	Rate: 20Hz
Delay: 120ms	Period: 10s

Bid
Price: £0.03
Commitment: 95%
Reputation: 47

Figure 6.5: Sample values for a flow contract and associated bid

A session contract is a formal agreement between the user and their service provider, while a flow contract is an agreement between a service provider and a network operator. The service provider may decompose a session contract into multiple flow contracts, if the user's mobile terminal supports multiple concurrent sessions (see Figure 6.6).

Each contract consists of the six parameters discussed in section 5.3.4.2. Performance requirements are specified by bit rate, BER, and delay, and conformance requirements are degradation allowance, sampling rate, and monitoring period. An operator can be said to have failed to meet the contracted commitments if the measured performance is lower than that allowed by the degradation allowance.

6.5.2.3 Bid Format

Bids in the service auctions are placed by network operators. A bid for a flow contract consists of price, commitment level, and reputation. Price and

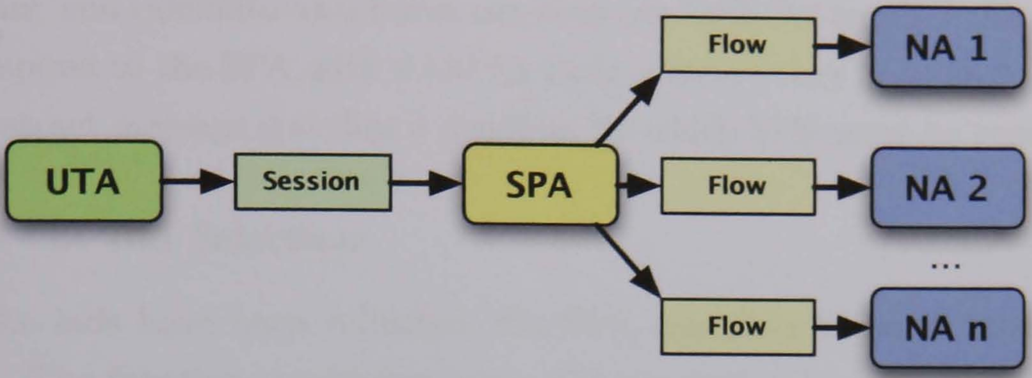


Figure 6.6: Decomposition of single session into muiltiple flows

commitment level can be dynamically altered by the network operator for each auction, whereas reputation is calculated by the market operator and made available directly to the service provider. The service provider must combine these three parameters to decide which bid is the best.

A commitment level is a statistical goal of service provision, which is intended as a parameter to keep reputation accurate. For example, one provider may agree to a 95% commitment level for all calls, and another may offer 90% commitment at a lower cost. The user's agent may choose between the more reliable provider, or the cheaper one. To exactly meet their commitment level, the first provider must consistently fail to provide the agreed quality of service to exactly 5% of calls, and the second must do the same for exactly 10% of calls.

6.5.2.4 4: Flow Contract

Returning to the protocol: once in the marketplace, the SPA uses the session contract to form a number of flow contracts. It sends these contracts to the Network Operator Agents (NAs), requesting bids. Note that the number of flow contracts could simply be one, for some cases.

6.5.2.5 5: Bidding

Each Network Operator Agent (NA) receives the proposed contracts from the SPA. They interpret the parameters, combine with their current commitments, and calculate whether or not they can fulfil the requirements. If so, they respond to the SPA with a bid for each contract they hope to win. The flow contract message specifies a deadline by which bids must be received.

6.5.2.6 6: Bid Selection

Once the bids have been collected, the SPA compares them to choose the winner. The function combining price and reputation is determined by the service provider, and may change to reflect their users' preferences for cost over reliability.

The winning NA is then informed that a flow contract has been awarded to them at the price they bid. At this point, there exists a legally-binding contract between the service provider and the network operator.

6.5.3 Establishment to Termination

The final stage of the negotiation protocol wraps directly around the use of network service. The user establishes the actual network flows purchased by their service provider, and uses them for whatever purpose they desire. At the end of the network session, there is a tear-down and reporting procedure, which concludes the Digital Marketplace protocol.

6.5.3.1 7: Flow Establishment

Before the user can access the network, they must establish a communications session with a network operator. The SPA gives the UTA's address to the selected NAs, each of which instructs the UTA to establish a flow. The UTA can then contact the mobile terminal, which establishes an out-going communications session as normal.

6.5.3.2 8: Flow Release

While the flow is established, the user communicates for whatever purpose they want. At the end of the communications session, the UTA indicates that the flow should be terminated to each of the network operators concerned.

This marks the end of communication between the user's mobile terminal and the network operator.

6.5.3.3 9: Terminate Session

After flow termination, the network operator contacts the service provider to indicate session termination. This marks the end of the flow contract period, and if it is the final flow in the user's session, also the end of the user's communications session.

6.5.3.4 10: Report

The terminating NA also communicates with the Market Operator Agent at this stage. In order to keep its reputation accurate, the network operator is required to measure its commitment fulfilment according to the terms of the flow contract. These measurements are reported to the Market Operator, which applies the appropriate changes to the network operator's reputation.

6.5.3.5 11: Leave

After all flow contracts have been terminated, the session contract is also terminated. At this stage, the SPA leaves the marketplace, and updates the user's billing record.

6.6 Threat Analysis

While the DMP protocol has previously been thoroughly analysed, both for negotiation overhead[76] and free-market feasibility[77], so far there has been no focus on security. The first step towards ensuring that the Digital Marketplace protocol can operate securely must be to perform a comprehensive threat analysis.

The method used to perform this threat analysis is described here. For each step of the protocol, I consider the capabilities and motivations of all potential adversaries (see section 6.4). Then, an enumeration of possible security threats which result from this analysis can be presented. A security threat is defined here as an unfair action which can be taken by a market participant, in order to benefit the participant, to the detriment of other parties in the market.

The output of this Digital Marketplace threat analysis is the discovery of a great number of security threats applicable to the session negotiation protocol described in section 6.5. Each of these threats is associated with one or more steps in the protocol, and can be assigned to an overall class of security threat. These threat classes include: impersonation, bid repudiation, bid inspection, reputation manipulation, and collusion.

6.6.1 Impersonation

This class of threat is one in which the attacker sends messages purporting to be from someone else. With little trust among the many participants in the open marketplace, clearly this is an important issue to consider.

Many of these attacks could be performed without the ability to modify the contents of messages. All that is required is the ability to record and re-send previous transactions. This is commonly known as a *replay attack*.

Another difference from the modification threat class is that many of these attacks are time-sensitive. If the attacker is only able to modify a request while in-transit, they are unable to carry out timing-related attacks: for example, ending a session before the user requests it to end. The ability to carry out such attacks makes impersonation a much more pervasive security problem.

6.6.1.1 1: Connection Request

Any observer of the LMC could observe a mobile user's identifying information as the connection request is transmitted to the market operator. It could later use this information to send another connection request to the original user's SPA. The perpetrator of this attack could receive network service for free, while the victim pays for it.

A similar attack with a more malicious motive also exists. Instead of using only the identifying information, the attacker could resend a previous connection request verbatim. This would result in the original user being supplied unwanted service, and again having to pay for it. Clearly this could be of benefit to the winning network operator, which gives a motive for the attack.

6.6.1.2 5: Bidding

Some of the most critical threats due to impersonation arise in the auction stages. First consider the bidding stage: if identity can be forged, an attacker could place a contract bid in the name of any network operator. The SPA would not be able to distinguish this from a real bid, which causes several problems.

Since the auction format demands a single bid from each participant, this would distort the results of negotiations whatever the outcome. Following this attack, there are effectively two possible states after the auctions ends: either the false bid is the only one associated with the victim network operator, or there are multiple bids from the victim network operator.

In the first case, if the bid wins, the network operator would be held to a bid it did not place. This could cause it to either suffer a reputation penalty (if it cannot meet the commitments) or lose revenue (if the price is lower than it would have bid).

If two bids are received, the SPA must then make a binary decision: drop both bids, or select one of them. If both bids are dropped, then this attack becomes a denial-of-service problem: an attacker can stop a network operator from participating in auctions by consistently placing bids on its behalf. Alternatively, if one bid is selected, the consequences are equivalent to the single bid case above. In both cases, the most likely attackers are other network providers, as they stand to gain from the losses of the victim.

6.6.1.3 6: Bid Selection

Other security threats in the auction process exist in the bid selection stage. Then, an attacker could send a message to a losing NA, purporting to be a bid acceptance notification from the SPA. As with the reverse case, the victim would have no way of telling that this message is fake.

The consequence in this case would be that the victim network operator would expect to provide network service to the relevant user. If the remaining steps of the protocol are carried out, the user could then receive network service, and the network provider would be refused payment from the service provider.

Depending on the attack profile, the user may be either aware or unaware of the illegitimate flow contract. If the user were involved in the attack, they would be aware of the situation and therefore refuse payment to the service provider; if not, the service provider is the most likely attacker, and could bill the user and keep all funds as profit.

This outcome is therefore detrimental to the network operator, and at the same time beneficial to the user or their service provider. Therefore, any of the actors in the marketplace—other network operators, users, and service providers—are potential attackers.

6.6.1.4 8: Flow Release

An attacker could send a flow release message to the currently-serving NA, purportedly from the mobile user. This would stop the flow, leading to a loss of connectivity for the user. There are two possible motivations for such an attacker, with the attacker either being the serving NA or another NA.

In the first case, the advantage for the NA would be that the attack frees up network resource by dropping a connection. If the market has shifted such that the price of the current flow contract is under market value, it would be of advantage to the NA to drop the connection and start a new one.

On the other hand, there may be penalties applied for the connection dropping. The network operator's reputation could suffer, because the user would be unable to distinguish this attack from a dropped connection due to network conditions. Furthermore, if the contract specifies a price per unit time or throughput, the network operator could lose out on revenue due to the contract being terminated prematurely. This would then be of benefit to competing NAs.

6.6.1.5 9: Terminate Session

Similarly to the previous threat, an attacker could send a session terminate message to the SPA, pretending to be from the NA. This would mark the end of the billing period, which would therefore reduce the network operator's revenue. Therefore, this attack is again most likely to be executed by a competing network operator.

6.6.1.6 10: Report

The final threat which stems directly from impersonation is in the commitment reporting stage. A competing network operator could wait until session termination, then submit a falsified report message. The report could incorrectly indicate that a network operator did not meet the agreed commitments. Clearly, this would damage the reputation of the network operator unfairly. As before, the motive would be to damage a competing network operator, and in doing so raise the relative reputation of the attacker.

6.6.2 Bid Repudiation

With a single round sealed-bid auction, it is vital that bidders be kept to their promises. If a network operator can reasonably claim that their winning bid was sent by someone else, this enables threats based on bid repudiation.

One possible attack would be for a network operator to place an aggressive bid in **step 5**, on a flow contract which has quality of service levels that it cannot meet. Its goal here would be to deny other network operators the opportunity to sell this level of service. If it wins the auction, it could then claim that the bid was placed by an attacker, and refuse to supply service; if this claim is justifiable, it would be unfair to apply de-commitment penalties.

An alternative strategy would be to place a bid at a given price in **step 5**, this time on a flow contract which it expects to meet. If the flow commitments are not met, the network operator could then claim that the bid was actually for a higher price. Unless the true price can be verified, the contract is impossible to enforce, and failure to meet commitments again cannot be punished.

Attacks like these would damage the overall operation of the market. Users must be confident that paying for network service will lead to one of two outcomes: either the service will be provided, or the failing operator will be punished. If this is not the case, the worth of calls in the marketplace will be continuously lowered, until the value is equal to that of the lowest quality provider.

If these strategies were possible, they would also quickly become the only winning bidding strategy. The result of this would be that every bid on a broken contract would be repudiated, de-commitment penalties would never be applied, and all providers' reputations would tend toward the maximum possible value. The reputation system would then become worthless.

6.6.3 Bid Inspection

Another important issue around **step 5** of the protocol is the secrecy of bids. If a network operator can learn the value of its competitors' bids, it can use this information to change bidding strategy. For example, the NA could very slightly improve upon the best bid so far, maximising its revenue if it wins.

If bid inspection were possible, it could also lead to an increase in the latency overhead due to negotiations. Network operators would be given an incentive to delay bidding until other bids are placed, increasing the maximum overall delay. It is therefore extremely important that the bids placed are held secret from everyone but the service provider.

6.6.4 Reputation Manipulation

The proposed protocol leaves open the possibility of fraudulent reputation reports from the network operator. At **step 10** of the protocol, the NA is asked to report on its commitments. There is no incentive at all for the network operator to always report truthfully here.

If commitments were met, of course the NA would report this. However, if not, the network operator would lose nothing from claiming that the service was provided successfully. Therefore, it would always be in the best interests of the operator to claim that commitments were met.

This would lead to all NAs having a reputation tending toward the maximum possible. If the reputation score was unlimited, it would then become a reflection of the number of calls serviced, rather than the success rate of those calls. This would vastly reduce its utility in the bid selection process.

6.6.5 Collusion

A great number of threats exist only when two or more parties agree to purposely pervert the market operation. These all fall under the class of collusion attacks, despite their disparate nature.

The most likely participants in collusion are network operators, as they stand to gain the most from manipulating the market. However, their capabilities are limited, and so partnering with a service provider or the market operator is a logical choice. The most likely motive for the service provider or market operator would be receiving financial compensation, taken from the increased revenue of the colluding network operators.

6.6.5.1 Threat Categorisation

There are three classes of collusion-related security threats: discrimination, differential messaging, and message modification. Discrimination attacks simply apply different standards to different parties, depending on whether or not they are part of the collusion. Similarly, differential messaging changes messages to be received by non-colluding parties, to disadvantage them.

In contrast, message modification covers two types of attack, both of which require interception of a message. This may either be followed by replacing its contents and forwarding it to the original destination, or simply destroying the message silently.

As the negotiation protocols take place in an agent environment, the only possible perpetrator of message modification attacks is the controller of the platform: the market operator. A correctly-operating agent platform would not allow messages to be modified in transit; therefore, the only way to modify messages is to distort the operation of agent environment. From the adversary analysis above, it is clear that the market operator by default has little incentive to attack any party in the marketplace. However, it is possible that it could be corrupted, and collude with other parties. It is in such a circumstance that these attacks become possible.

6.6.5.2 1: Connect

The mobile user sends the session contract to the market operator, to be forwarded to its service provider. An attacking MA could modify the session contract before forwarding it, in order to favour network operators it is colluding with. The service provider would not be able to detect that the session contract differed from the user's desired parameters, and would negotiate in good faith.

For example, the market operator may know that its colluders can provide better quality-of-service than other network operators. Therefore, it would increase the QoS requirements in the contract to reach beyond non-colluding parties, but still within the reach of its partners. The converse case is also possible: reducing the QoS requirements to allow bids from poorer-performing colluding network operators.

6.6.5.3 2: Migrate

Along with the session contract, and the market location, the MA includes the list of present NAs in the *migrate* message. At this stage, a corrupt MA could exclude non-colluding NAs, effectively removing them from negotiations.

If such an NA never receives contract tenders, it would eventually detect this attack, and report it to a higher authority. In turn, this could be countered by only probabilistically removing NAs from the list, ensuring that they are involved in enough auctions to lower their suspicions. Then, the attack would be impossible for a network operator to detect.

6.6.5.4 4: Flow Contract

Similarly to the previous attack, an SPA could remove NAs from the list of recipients of the flow contract. Those with whom it has an offline agreement would always receive the contract tenders; others would receive them less often, or not at all. Clearly this would benefit the colluding network operators.

The SPA is also directly responsible for sending out identical copies of the flow contract to each of the NAs in the marketplace. However, it could selectively slacken the contract terms for colluding network operators, or increase them for non-colluding NAs. This would unbalance the auction in favour of its colluders, while still sending flow contracts to all competing NAs.

The market operator is also capable of the same two attacks at this stage. By deleting messages, flow contracts could be prevented from reaching non-colluding network operators. Messages could also be modified in-transit, to favour colluding network operators.

6.6.5.5 5: Bidding

The bidding stage is one of the most crucial points in the protocol for message interception and modification. An attacker could change the price given in the bid, either upward or downward, to influence the bid selection process.

If the bid is increased, the network operator in question is made less likely to win the auction. This is clearly of advantage only to other network operators. If the bid is decreased, and the network operator wins, it may be contracted to supply service at reduced profit, or even at a loss. This second case is of benefit to other network operators, if the lower revenue harms the victim; and also beneficial to the service provider, because the cost of network service is reduced.

Another possible attack is deletion of bids. The controller of the agent environment could selectively drop or delay bids from non-colluding parties, effectively removing them from the auction process. As there is no requirement for a bid from all network operators, the service provider could not tell that this attack took place. Likewise, there is no requirement for a bid acknowledgement from the SPA to non-winning bidders, the network operators would be unaware of the attack.

6.6.5.6 6: Bid Selection

Again similarly to threats present in **step 4** and **step 5**, the service provider could abuse the bid selection process. A colluding SPA could ignore bids from non-colluding NAs, or simply slightly favour bids from colluding NAs. Because the bids are sealed, other actors in the marketplace would be unaware of this collusion, even if a clearly inferior bid is declared the winner of the auction.

6.6.5.7 7: Flow Establishment

After winning the auction, the NA must contact the UTA to begin the communications flow. This message could be modified or dropped by the MA, in order to prevent the flow from starting. The user would then not receive service, and the network operator would be penalised for failing to meet its commitments. This would be of benefit to colluding network operators.

6.6.5.8 10: Report

MA could collude with NAs to maintain a high reputation, even if commitments are not met. For example, the MA could disregard any commitment reports which would adversely affect the NAs' reputations.

Commitment reports are another possibly-modifiable message type, which could be exploited to the benefit of some actors in the marketplace. As the protocol stands, the only motivated attack would be to change a 'success' report to read 'failure': the relevant network operator would then be unfairly penalised for failing to meet commitments. Again, this would benefit other network operators by comparison.

6.6.6 Summary

The comprehensive list of threats presented in this analysis are broken into seven classes and sub-classes. The main classes are impersonation, bid repudiation, bid inspection, reputation manipulation, and collusion. This last class includes message modification, differential messaging, and discrimination attacks. Each threat is associated most closely with a single protocol step, and therefore each step may have many potential associated threats. This information is summarised in Table 6.2.

	Protocol Step									
Threat Class	1	2	3	4	5	6	7	8	9	10
Impersonation	✓				✓	✓		✓	✓	✓
Bid Repudiation					✓					
Bid Inspection					✓					
Reputation Manipulation										✓
Modification	✓				✓		✓			✓
Differential Messaging				✓						
Discrimination		✓		✓		✓				

Table 6.2: Security threats by class and protocol step

6.7 Security Requirements

From this comprehensive threat analysis, patterns of security problems emerge. The next step in securing the Digital Marketplace is to define a series of security requirements, each related to a group of threats. Once specified, these requirements determine exactly which aspects of the system need to be improved in order to secure its operation.

Security requirements are implementation-independent. The purpose of this stage of analysis is to create a series of security targets, which can be reached via many paths. Once met, regardless of the methods used, the system can then said to be secure with regard to the threat analysis. This enables different security measures to be applied, each with differing optimisation characteristics. Some solutions may target ease of implementation, while others could focus on reducing communications overheads; as long as the requirements are shown to be met, the approach is valid and the system is secure.

The necessary changes to counter the threat classes above can be broken into four requirements: authenticity, secrecy, collusion detection, and reputation. These are each presented in more detail below.

6.7.1 Authenticity

The majority of security threats identified above are related to message authentication. All impersonation, bid repudiation, and message modification threats require the subversion of message sender or contents. Ensuring that messages can be authenticated is therefore of great importance, as it would counter many possible attacks.

Message authenticity as discussed here actually demands several security properties. Most obvious is the ability to determine who the original sender of a message was; but it is also important to ensure that the message cannot be recorded and replayed by an attacker. Another property is message integrity: it must not be possible to modify the contents of a message without detection by the recipient. Finally, all of these properties must be demonstrable, to ensure that bids cannot be repudiated.

Requirement 1 All parties must be able to verify authenticity of received messages.

6.7.2 Secrecy

Message contents only need be kept secret in the bidding stage of the protocol. As discussed in section 6.6.3, the ability to read other network operators' bids would damage the auction process. In the worst case, every participant in the marketplace would be able to inspect every message transferred. Therefore, the auction protocol must ensure that each bid's content can only be read by the service provider who initiated the auction.

Requirement 2 Bids must be readable only by the recipient SPA.

6.7.3 Collusion Detection

To gain support, the Digital Marketplace must be seen to operate fairly. Perverting the free operation of the market is possible with the published protocol design, even when the two previous requirements are met. Arguably the main threat to correct operation is collusion between two or more actors.

The simplest technical approach to preventing market abuse is to ensure that it can be detected after the protocol is complete, allowing the appropriate punishment to be applied later, offline. This is a less invasive and more practical method than radically altering the protocol's operation, which could in itself damage the operation of the market by introducing delay or other overheads.

All participants in the marketplace must be able to detect and demonstrate unfair behaviour. There is a limited set of colluding actors to detect. The market operator may collude with the service provider, or one or more network providers; or, the service provider may collude with one or more network providers. Each of these cases must be handled.

Requirement 3 All marketplace participants must be able to detect and demonstrate collusion between two or more parties.

6.7.4 Reputation

The final issue with the current DMP protocol is the reputation system. Giving control of reputation updates to the network operator inevitably leads to abuse, as argued above. The system must be changed to ensure that failing to meet commitments always leads to a decrease in reputation, and that there is no incentive for the network operator to lie about its commitment satisfaction.

In addition to changing the protocol, the reputation calculation mechanism may also need to be changed. To enable truly substitutable services, a network operator's reputation must reflect two aspects of its recent performance: the difficulty of the commitments taken on, and how well those commitments were met. The previously-proposed reputation system does not achieve this goal; see chapter 7 for more detail.

Requirement 4 Reputation must fairly reflect network operator behaviour.

6.8 Security Measures

Meeting the security requirements described above, without dramatically changing the way the Digital Marketplace works, is a challenging balancing act. If too many modifications to the protocol or system are made, the operation of the marketplace could be damaged. Conversely, if the modifications are not fully effective, the DMP will remain insecure.

This section discusses the solutions to the security challenges facing the Digital Marketplace. First, some general security issues with the Digital Marketplace are addressed: how to punish dishonest participants, and what is required to allow mobile agents to operate safely. Then, each of the security requirements are dealt with in turn: authenticity, secrecy, collusion detection, and reputation. Finally, the changes made are summarised, justified, and a modified Digital Marketplace protocol is presented.

6.8.1 Judicial System

It is unfeasible to prevent all security breaches in any non-trivial system; the Digital Marketplace is no exception to this rule, and its real-time negotiation requirements make it fairly complex. Therefore, it is almost inevitable that the security of the market protocols will be breached. In this case, it is most important that such breaches can be caught, and that there is a deterrent to those who would carry out attacks on the market.

This requires a judicial system to be in place as part of the Digital Marketplace organisation. Previous work has discussed a Digital Marketplace consortium, which would be responsible for the day-to-day running of each marketplace environment[63]. It would be prudent to expand its role to include handling claims of unfair behaviour, and pursuing punishment for those parties who are thought to be guilty.

Many of the security threats described above would be prosecutable under common law as fraud: deception intended to achieve financial gain. Of course, whether such prosecution would be successful or not depends on the individual case, but the presence of an independent body which would initiate the prosecution should be of some deterrence to would-be disruptors.

6.8.2 Secure Agent Execution Environment

As described in section 6.2, the Digital Marketplace is based upon an agent architecture. One of the less common requirements of the marketplace environment is support for mobile agents: those which migrate over a network and run with others in a single container.

This is clearly a security concern: running code from untrusted parties is always potentially dangerous. If the agent execution environment is not secure, the migrated agents could be able to exploit this to alter the behaviour of other agents in the marketplace.

For an agent platform to be considered secure enough for the Digital Marketplace, it need meet only one simply-stated requirement: no agent should be able to view or modify another agent's code or data. Many previously-published secure execution environments exist which aim to meet this goal[121][7][119][74].

However, this would not be enough to convince all actors that the marketplace is secure. Almost all secure agent platforms assume that the host must be trusted; whereas, we have seen that the Market Operator is a formidable adversary, and should not be given absolute trust. To demonstrate the security of the agent platform, the market operator must be able to prove that the execution environment has not been tampered with in any way.

To demonstrate to other actors that this is the case, we require first that the environment must run on a trusted-computing platform[116]. Secondly, the source code to the agent execution environment must be published by the Digital Marketplace Consortium, in order to allow any interested party to inspect its operation. Finally, the trusted-computing platform must respond to a *Remote Attestation* request[118], to allow a client to verify that the code being executed matches that which was published.

A similar mechanism was used for another recent security project, the Reusable Proof-of-Work scheme[47]. This ran on an IBM 4758 tamper-proof cryptographic co-processor, which could be queried to supply a digitally-signed hash of the code it was running. This digest, known as an *Outbound Authentication* in IBM terminology, could then be compared to a hash of a locally-compiled version of the published source code by any third party, thus verifying that the code actually running is as-published. Then, a security audit of the published source code can be done to ensure that there are no back-doors or other security problems.

With such a scheme in place, the agent execution platform can be considered secure. Then, the only security issues that remain are as a result of communications between agents, as it can be fairly assumed that none of the actors in the marketplace are otherwise able to interfere with each other.

6.8.3 Authenticity and Secrecy

The first two requirements, for message authenticity and communications secrecy, initially seem solvable with standard cryptographic techniques. As described in section 3.2, message authentication with non-repudiation can be achieved with digital signatures; and as noted in section 3.1, communications secrecy requires the use of ciphers.

There are well-respected standards for each of these techniques, which are useful aids in choosing secure algorithms. The current best-practice signature scheme for RSA is the Probabilistic Signature Scheme (PSS)[10], which is in the standardisation process at time of writing. ISO 18033 presents schemes for secure use of asymmetric ciphers[64], including the widely-used RSA-KEM. However, in order to use these asymmetric algorithms, we need to securely distribute keys.

6.8.3.1 Public-Key Infrastructure

Certification (see section 3.3) is often a problem when designing a system which uses asymmetric cryptography. The best authenticity and secrecy techniques in the world are clearly useless if the public keys used are compromised. In the Digital Marketplace, where most of the parties involved have only very short relationships with each other, how public keys are distributed is clearly of great importance.

Therefore, it is vital to define a secure infrastructure for sharing public keys among market participants. Below, this problem is examined from the individual perspective of each class of actor involved in negotiations: the user, the service provider, and the network operator.

User Perspective

The user must be able to communicate securely with all other participants in the marketplace. In order of protocol flow, these are the market operator, the service provider, and winning network operators. However, the security requirements for communicating with each are different.

To be able to trust the marketplace security at all, the user must be able to verify that the market operator is approved by the Digital Marketplace consortium. Therefore, the market operator must broadcast a certificate over the Logical Market Channel. The certificate must be signed by a DMP-wide trusted third-party, as part of the market operator verification process.

If the user trusts the DMP consortium to act fairly, this certification should assure the user that the LMC corresponds to a valid Digital Marketplace. Therefore, it is then reasonable to send the connection request over the LMC, to initiate the protocol.

As noted above, it is important that the connection request message be authenticated, to ensure that it is not modified in-transit by a corrupt market operator. Third generation mobile systems, such as UMTS, already include support for mutual authentication using asymmetric cryptography[4]. Therefore we can assume a secure offline key-pair exchange has taken place between the user and the service provider; then, each can authenticate signed messages sent by the other.

The next part of the protocol which directly involves the user is step 7: flow establishment. At this point, there are two issues to address. How does the network operator securely communicate with the user? How does the user know that the network operator actually won the auction? Both of these issues can be addressed by more in-depth specification of protocol step 6 (bid selection) and step 7.

In step 6, a winning NA must receive a notice of bid acceptance; this must be signed by the service provider to be regarded as a valid contract. The signed portion of the message must include the communicating address of the network operator, to ensure that it cannot be re-used by another network operator. Finally, the bid acceptance notice must include the public key and address of the user, to allow a secure communications channel to be set up (see section 3.1.2.4).

Then, when creating the flow establishment message, the network operator must include a copy of the signed bid acceptance. The user can verify the service provider's signature to ensure that the network operator is authorised to provide network service. This concludes the user's requirements for secrecy and authentication in the marketplace.

Service Provider Perspective

The service provider has similar requirements to the user. Secure communication with the user is dealt with above, using current mutual authentication schemes; likewise, verifying the market operator's validity can be done by checking a copy of the certificate sent in the migration request. However, there are different requirements for communicating with network operators.

One of the core principles behind the operation of the Digital Marketplace is that there must be a low barrier to entry; without this, the market would fail to operate freely. Therefore, the number of possible network operators is unlimited, and may be constantly changing. Engaging in an offline key exchange with every possible network operator is clearly unfeasible; therefore, a trusted third-party (TTP) must be used to certify the identity of each network operator.

A clear candidate for this position is the market operator. Network operators must already communicate offline with the market operator to be granted access to the marketplace. As part of this authorisation process, there must be a requirement for the market operator to verify the legal identity of each network operator. The market operator must then sign a certificate for the network operator to later demonstrate that this verification step has taken place.

Verification of the network operator's identity is important for several reasons. First of all, it provides a real-world target for the offline judicial process, as described above. Perhaps more importantly, it prevents network operators from covertly rejoining the marketplace after punishment which would normally prevent them from trading.

These certifications can then be passed on to the migrating service provider agent. To check that all network operators are valid, the service provider need only verify the market operator's signature in the certificate. If the market operator is trusted to verify and sign honestly, and its own identity can be verified, this process enables secure identification of and communication with the network operators.

Network Operator Perspective

In comparison to the user and the service provider, the network operator has much simpler requirements of the certification and key exchange process. Identification and key exchange with the market operator is achieved offline, as part of the marketplace induction process described in the previous section. Therefore, the only parties for which online key exchange is an issue are the service provider and the user.

Because the network operator agrees delayed-payment flow contracts with the service provider, it clearly must have some knowledge of the SPA's real-world identity. The service provider's certificate therefore must be signed by the DMP-wide trusted third-party, just as the market operator's is. As with the network operator's certificate, this would provide a legal identity, for billing or judicial purposes.

In the case where the user has a self-representing service provider agent, payment must be made using an electronic payment scheme. This implies up-front payment for service, which in turn means that identification is not necessary. The user cannot use service and later fail to pay, so no legal identity is required.

Similarly to the user's authentication requirement for step 7 (flow establishment), the network operator must authenticate the flow release message in step 8. However, this can be achieved by simply continuing to use the secure channel set up in step 7, as described above under 'User Perspective'. Then, the only possible sender of the flow release message is the real user.

6.8.3.2 Certificate Revocation

All asymmetric cryptography schemes must deal with the possibility of compromised or lost keys. One commonly-used method of handling such a case is to maintain certificate revocation lists (CRLs) for invalid but not-yet-expired certificates. This means that these revocation lists must be regularly checked by all client software, to prevent malicious certificated parties from continuing to use an invalid certificate.

For the network operators' certificates, this is clearly not an issue. The SPA is given these certificates upon entry to the market by the certification authority. Therefore, no CRL check needs to be made.

However, for the other two certificated parties (the market operator and the service provider), these checks must be made. This is simple to do in the general case, as the verifying parties have fixed network connections to retrieve the CRL: the service provider can check the market operator's certificate, and the network operators can check the service provider's certificate.

The more complex case is that of the self-representing user, who must verify the certificate validity of the market operator without a network connection. Nothing can be done to improve this situation, except encourage the user to retrieve and cache the Digital Marketplace CRL at every opportunity. The list is likely to be extremely small, and infrequently changing, so this is not too onerous a demand to make.

Owner	Certifier	Users
Market Operator	Trusted Third-Party	User, Service Provider
Network Operator	Market Operator	Service Provider
Service Provider	Trusted Third-Party	Network Operator

Table 6.3: Certification in the Digital Marketplace

6.8.3.3 Certification Trust Model

As discussed above, three of the Digital Marketplace participant classes must have certificates to prove their identities. The certificate owner, certifying party, and the users of the certificate are enumerated in Table 6.3.

Introducing this certification scheme raises the implication that the parties involved must trust each other to some degree. There are four classes of actor in this trust system: the service provider, the market operator, the network operators, and the trusted third-party: see Figure 6.7.

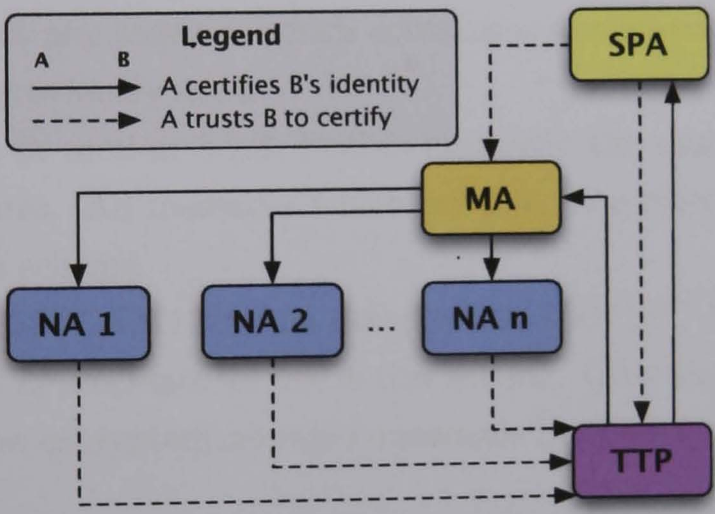


Figure 6.7: Certification and trust in the Digital Marketplace

As discussed above, the trusted third-party's role is to certify the identities of the service provider and the market operator. Similarly, the market operator is required to certify the identities of all network operators in its market. These certifications are only required to ensure that cheating parties are identifiable and can be held accountable for their actions.

Trust is limited to confidence that the trusted party will identify and certify parties with due diligence. In this respect, the service provider trusts the trusted third-party to certify the market operator, and trusts the market operator to certify the network operators. The network operators only rely on the trusted third-party to certify the service provider.

6.8.3.4 Secure Messaging

All communicating parties in the marketplace will have one or more public keys in their certificates; secure distribution of these certificates is described above. For secure messaging, we require that many messages are signed, and that some are encrypted. There are many possible schemes to achieve this: an example, using RSA as the cryptosystem, is given below.

Digital signatures must be used to counter modification, replay, and repudiation threats. Modification and repudiation threats are countered by a standard digital signature scheme. To counter replay threats, we require that each message contain a unique incrementing sequence number, and that participants reject any message which contains a sequence number less than or equal to one previously used.

As described in section 3.2.3, PSS is currently the most secure scheme for RSA signatures. All messages which need to be authenticated must be signed using this scheme.

The RSA-KEM+DEM1 scheme described in ISO/IEC 18033[64], and in section 3.1.2.4, is appropriate for use in this setting. Only the *connect* and *bid* messages must be encrypted; no other messages have secrecy requirements.

6.8.4 Collusion Detection

Even with the authentication and secrecy measures described above, collusion is still a problem. Of all the collusion-related threats listed in Table 6.2, only the message modification attacks are solved. Messages can only be secretly dropped if the agent platform is unfair; as specified in section 6.8.2, all DMP agents will run in a verifiably-secure agent execution environment, which prevents this. The remaining modification and impersonation attacks are prevented by the use of message authentication, as described in the previous section.

However, differential messaging and other discriminatory behaviour both remain possible. There is no way to modify the protocol to prevent such attacks from ever taking place. Therefore, as noted in the security requirements, the best approach to this problem of collusion is rapid detection. Every user of the marketplace should be able to detect unfair actions due to collusion, and report them to the judicial system. This should act as sufficient deterrent to prevent these attacks from occurring, or at least reduce their frequency.

Therefore, the remaining open collusion-related threats are:

- MA modifying network operator list (section 6.6.5.3)
- SPA modifying network operator list (section 6.6.5.4)
- SPA sending differing contracts (section 6.6.5.4)
- SPA making an unfair bid selection (section 6.6.5.6)
- MA unfairly updating reputation (section 6.6.5.8)

The first three of these can be grouped into an overall flow contract distribution problem; the last two must be considered individually. Each of these three classes is addressed below.

6.8.4.1 Flow Contract Distribution

It is vital that every network operator receives an identical flow contract at the start of the auction; otherwise, the market becomes distorted. As previously described, the procedure for distributing flow contracts is as follows:

1. MA provides list of NAs to SPA
2. SPA creates flow contracts from session contract
3. SPA sends flow contracts to each NA

Each of these stages corresponds to an attack above: the MA can modify the list of NAs; the SPA can create different flow contracts for different NAs; the SPA can send flow contracts only to some NAs. These attacks are possible even with the security measures described above in place. The invitation message is sent outside the marketplace, so cannot be countered by using a secure agent platform; the SPA is an untrusted agent, so its behaviour in this regard cannot be regulated.

There is therefore a need for some secure mechanism for distributing flow contracts, which counters the three threats described above. This can be achieved by fundamentally altering the flow contract distribution procedure, to route through a trusted Logging Agent (LA). See Figure 6.8 for a graphical representation of the consequences of this change.

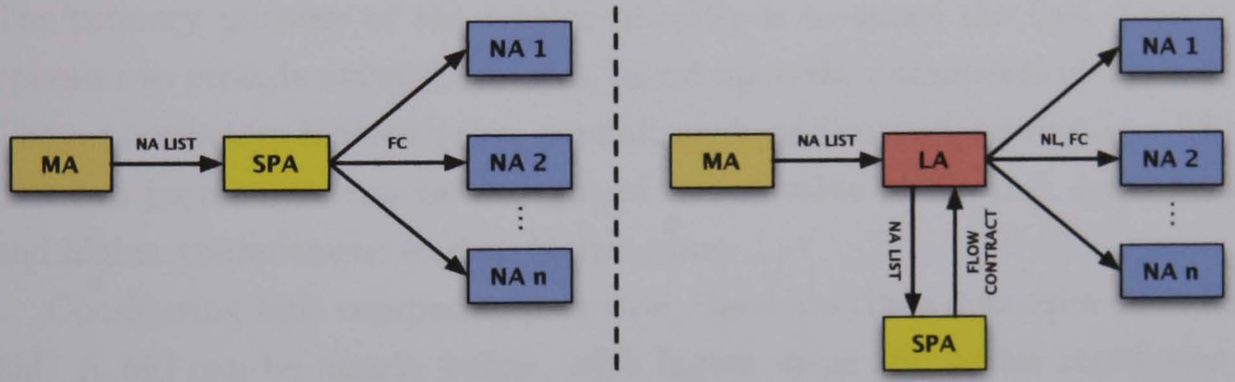


Figure 6.8: Previous flow contract distribution (left) compared with distribution via secure logging agent (right)

The MA no longer includes the list of NAs, reputations, and certificates with the invite message. Instead, the MA sends any changes in the NA list to the LA. The SPA needs the NAs' certificates (for bid signature verification) and reputations (for bid comparison). Therefore, the SPA requests these data from the LA as required. Similarly, to distribute a flow contract, the SPA sends it to the LA.

Every NA in the marketplace subscribes to all messages leaving the LA. Therefore, whenever the NA certificate list changes, or a flow contract is tendered by a service provider, every NA will receive a copy of the message. This is true even if the MA modifies the list of certificates: as long as the NA exists in the secure agent environment, it will be able to receive messages from the logging agent.

This therefore solves the flow distribution problem: the MA cannot modify the list of NAs without being detected, and the SPA cannot unfairly distribute contracts. Of course, this is only true if the logging agent is fair, so there must be good reason for the network operators to trust it. This can be solved in exactly the same way as the verifiably secure agent environment: simply publish the source code to the agent, and run it in the secure agent platform. This will allow any user to verify that the code running matches the public source, and therefore that its behaviour is trustworthy.

6.8.4.2 Bid Selection Decision

The primary purpose of the service provider is to select the best network operator to provide network services, based upon the parameters of the bids. These parameters include price, commitment, and market reputation. The first two parameters can be thought of as the value of the bid: lower cost and higher commitment lead to higher value.

Considering bids competing pair-wise, there are three outcomes for each bid. A bid can be clearly better, with higher value and better reputation; conversely, it can be clearly worse, with a lower value and poorer reputation. Any other combination of value and reputation leads to an unknown state, where the bids cannot be ordered without using some weighting function. It is the service provider's responsibility to develop an algorithm to choose between bids in such a state.

Clearly, if a bid is overall better than any other, winning on both value and reputation, it must be the winner. If such a bid is not selected, the service provider can be determined to have unfairly chosen another bid from a colluding network operator. Similarly, if there exists a bid which exceeds the winning bid on both value and reputation, the service provider has unfairly chosen between bids.

To allow interested parties to verify this, the Logging Agent can again be used. This is achieved by requiring that the SPA decrypt and publish the sealed bids at the conclusion of the flow, after step 8. With this *auction report* data, any interested party can evaluate the bids, and verify that there are no losing bids which are clearly better than the winning bids.

Network operators must also be able to demonstrate that they actually placed a bid; otherwise, the SPA could simply drop the bid from the auction report and claim it was never received. Again, the LA can be used to store the encrypted bids until the end of the auction. This can also be used to demonstrate that the bid was placed before the deadline.

6.8.4.3 Reputation Update

Reputation is updated at the end of each flow contract, using a function with two parameters: the contract terms, and the commitment report. This function must be published so that both network operators and service providers can understand what the reputation value actually means.

Unfair modification of reputation scores is the final collusion-dependent threat. This means any manipulation of reputation by the market operator which does not match the designated reputation function. For example, increasing or decreasing the reputation by too much or too little, or even not updating the reputation at all.

Again, the Logging Agent is an appropriate tool for detection of such attacks. There are two additional message logging requirements: both the commitment report and the reputation update must be sent to the LA. These can then be retrieved by any interested party.

All network operators subscribe to reputation update messages, and verify that reputations are updated appropriately. This verification can be done using the published reputation update function, and three data sets from the LA: auction report, commitment reports, and reputation updates. The network operator simply applies the reputation function to the first two parameters, and checks that the resulting reputation change is correct.

Each network operator can choose to monitor its own reputation, all reputations, or a selection of its peers. The distributed nature of these checks discourage any unfairness on the part of the market operator, as it can always be detected and punished for such actions.

6.8.4.4 Logging Agent Summary

Five classes of message are handled by the LA: the NA list, flow contracts, encrypted bids, auction reports, and commitment reports. The NA list consists of certificates and reputations for each NA, and is updated only by the MA. A flow contract is sent by the SPA, forwarded by the LA to the NAs, and encrypted bids are sent by NAs in response. Auction reports and commitment reports are submitted at the end of the flow contract, following step 8. Any user of the marketplace may subscribe to these messages.

This data flow is a common pattern of agent systems, and is standardised by FIPA as the Subscribe Interaction Protocol[49]. An interested marketplace actor (the initiator) sends a *subscribe* message to the LA (the participant). This message includes a query, which specifies which data the initiator is interested in receiving. The participant responds with either *refuse* or *agree*, depending on whether or not it can handle the subscription. If agreed, there then follows a series of *inform-result* messages from the participant, once initially and then again after every change. To stop receiving these updates, the initiator can *cancel* the subscription. The full protocol is shown in Figure 6.9.

Because this is a standardised protocol, and nothing else is done by the Logging Agent, it is extremely simple to implement. Depending on the features of the agent platform used, it may only be a few tens of lines of code: only FIPA standard technology is necessary to implement it.

The argument for using a separate secure Logging Agent is based on this simplicity. As a system grows more complicated, it becomes more difficult to secure; the more complex the code, the more must be tested. The LA is almost trivial in its operation, and so it can be implemented and security-audited very easily.

A potential alternative would be to require that the Market Operator Agent be remotely-verifiable, but this would be a much more difficult task. The MA must interact with non-agent code to handle the LMC and agent

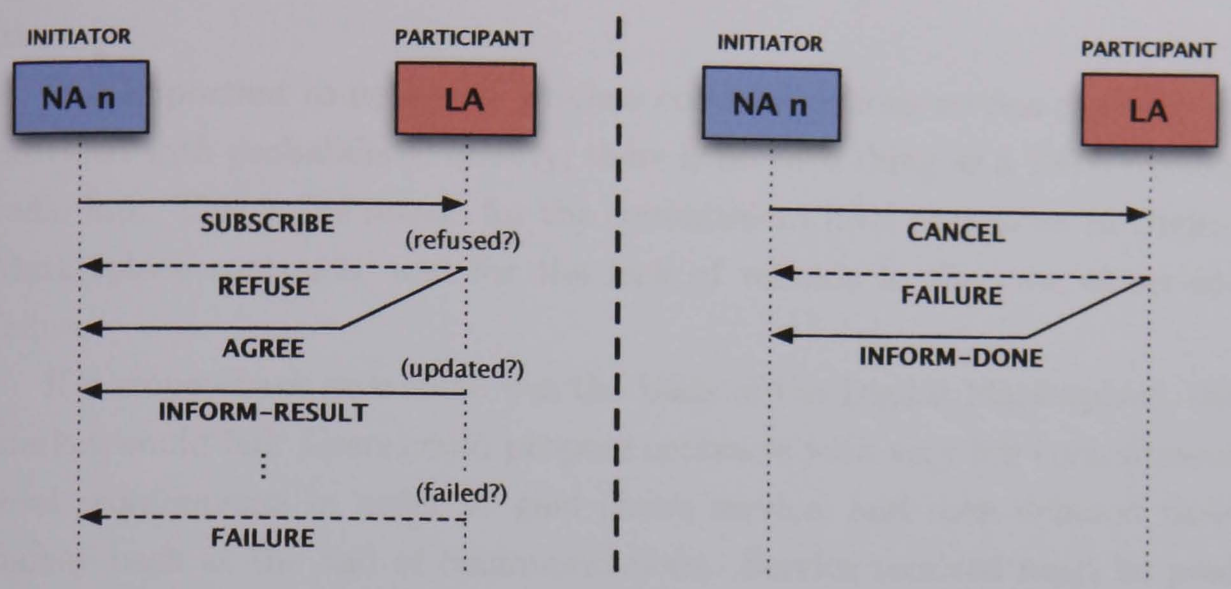


Figure 6.9: Message flow for FIPA Subscribe Interaction Protocol (left) and FIPA Cancel Meta-Protocol (right)

migration, and so its implementation would be more difficult to verify as secure.

6.8.5 Reputation

A fair reputation system is vital to the success of the market. Without an accurate reflection of past behaviour, services from many sellers cannot easily be compared. Two aspects of the current reputation system must be addressed: the reporting mechanism, and the reputation updating function.

6.8.5.1 Commitment Reports

The most important issue with the reputation system as described in the current protocol is in the reporting stage. As shown in section 6.6.4, the network operator is placed in a position of trust to report its failings. Economically, there is no incentive for it to act honestly, as doing so would only harm its reputation. Therefore, we must find another way of judging the performance of a network operator during a flow contract period.

Of all actors involved in the marketplace, only the mobile user and the network operator are able to report on commitments. Clearly, the network operator cannot be trusted, and so the only possible reporter is the mobile user.

It is important to note that wireless communications service can only be provided with probabilistic quality; there is no such thing as a 100% reliable radio link. This is the reason for the commitment level parameter in Digital Marketplace contracts, and for the lack of refunds in the case of service failure.

If a money-back guarantee was the basis of the Digital Marketplace, the market would fail. Users could propose contracts with very low commitment level requirements in order to gain cheap service, and then demand their money back at the end of communications. Service received must be paid for, regardless of whether the quality commitments were met; therefore, the user has no incentive to make inaccurate reports.

It is conceivable that a user could act in collusion with a network operator to reduce the reputation of other network operators. This class of security threat was noted in previous work, and it was conjectured that this must be countered using a bilateral reputation system[2]. However, such an attack would be impractical: to make any significant impact on reputation, a large number of calls would have to be made, and each would have a real financial cost to the colluders. This cost would cancel out any advantage gained by manipulating reputation in this way; it would be more productive to use this financial resource to undercut competitors instead. Therefore, there is no reason to distrust the user's reports, and they can be taken as accurate on the whole.

6.8.5.2 Reputation Function

Previous work assumed an extremely simple reputation function, which had fixed unit penalty and reward for failure and success. The problem with such a function is that it does not reflect the difference in services provided. Contracts with different commitment levels have different associated costs to fulfil: a higher commitment level is worth more than a lower one.

Therefore, some reputation function must be implemented which takes into account both the commitment report, and the flow contract's commitment level. This is a complex problem in itself, and is outside the scope of these overall security requirements. For this reason, it is dealt with in chapter 7.

6.8.6 Summary

The first security measure is non-technical: there must be some judicial system for handling complaints about market unfairness, which has sufficient power to punish participants who act unfairly (section 6.8.1). Secondly, and also not directly related to the protocol, the marketplace must run in a secure agent execution environment; further, all users of the marketplace must be able to remotely verify the agent platform is secure, as discussed in section 6.8.2.

All messages in the protocol must be authenticated, and many must be secret, which leads to the following set of security measures. There is a need for a public-key infrastructure, wherein all marketplace participants have a signed certificate to prove their identity and enable secure communications (section 6.8.3.1). The market operator must certify network operators, and there is also a need for a trusted third-party certifier; exact details of certification are given in Table 6.3. Requirements for message encryption and signing are given in section 6.9 below.

To counter the collusion-related security threats, we require the use of a trusted Logging Agent, which must also be verifiably secure (section 6.8.4). This agent holds a secure public record of auction events, allowing peer-verification of fair behaviour: for details on its operation, see section 6.8.4.4.

Finally, there are two changes to make to the reputation system, as described in section 6.8.5. First, the mobile user must report on commitment fulfilment, rather than the network operator; otherwise, the reports will be extremely unreliable. Secondly, a new reputation function must be devised, which fairly reflects the difference in difficulty of providing flow contracts with different commitment levels.

6.9 Modified Protocol

Incorporating these security measures significantly changes the protocol message flow. This section presents the DMP protocol with the measures in place; the original protocol description can be found in section 6.5 on page 104.

For clarity of comparison, the same stage numbering scheme is used as in the previous description. Some protocol stages now have additional messages defined, but the purpose of each stage remains the same. A graphical overview of the protocol flow is given in Figure 6.10.

6.9.1 Session Initiation

To use the Digital Marketplace, the user's terminal must locate a Logical Market Channel to communicate with the market operator. It must then indicate the location of its service provider, and desired service parameters, in order for the service provider to join the marketplace and begin negotiations.

6.9.1.1 1: Connection Request

The User Terminal Agent (UTA) sends a *Connection Request* message to the Market Agent (MA). This request includes the network address of the user's service provider, and desired terms for the session contract. Both the user's authentication parameters and the session contract must be encrypted and signed, to prevent reading or tampering by the market operator.

6.9.1.2 2: Migrate

Next, the Market Agent forwards the encrypted part of the *Connection Request* to the user's service provider, as an invitation to join the marketplace. Also in this *Migrate* message is the market operator's certificate, used by the service provider to verify the authenticity of the marketplace.

At this stage, the service provider validates the signature in the market operator's certificate, also checking that it has not been revoked. It then sends a *Remote Attestation* request to verify the integrity of the Agent

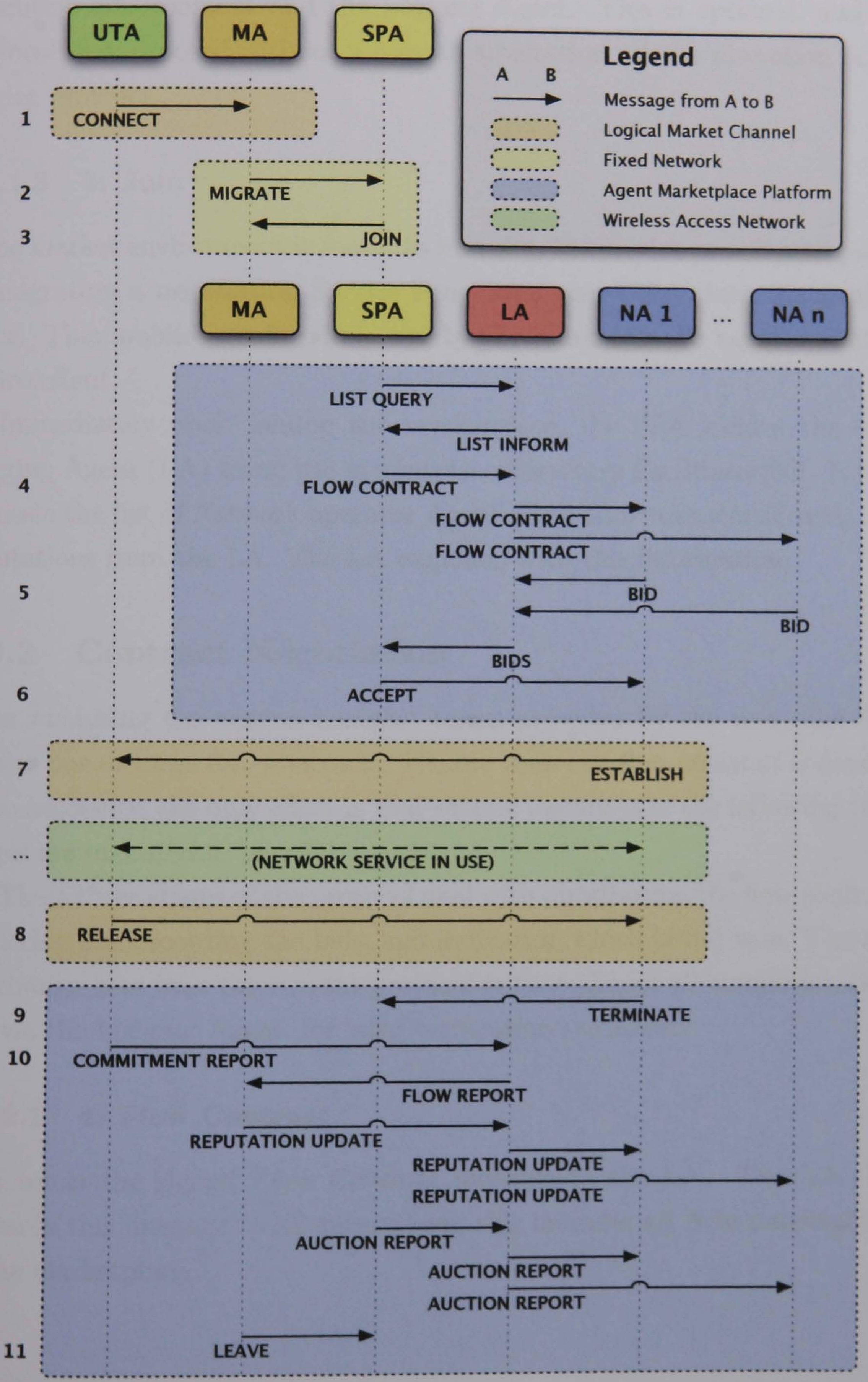


Figure 6.10: Modified Digital Marketplace protocol operation

Execution Environment and the Logging Agent. This is optional, and the protocol may proceed without a remote attestation, at the discretion of the service provider.

6.9.1.3 3: Join

If the market environment is found to be valid, the service provider responds by migrating a negotiating Service Provider Agent (SPA) into the marketplace. This mobile agent code is run by the marketplace's agent execution environment.

Immediately upon joining the marketplace, the SPA locates the local Logging Agent (LA) using the marketplace Directory Facilitator[50]. It then requests the list of Network operator Agent (NA) addresses, certificates, and reputations from the LA. The LA responds with this information.

6.9.2 Contract Negotiation

After analysing the session contract terms provided by the user, the SPA creates one or more flow contracts. If more than one flow contract is deemed to be necessary, the only effect is that several instances of the following three stages are executed in parallel.

These three stages of the protocol deal with distributing the flow contract, gathering and recording the bids, and indicating which bid(s) won. The major change here over the current protocol is that almost all communications are via the Logging Agent, for later verification purposes.

6.9.2.1 4: Flow Contract

SPA sends the signed *Flow Contract* message to the LA. The LA then forwards this message to all subscribers; this includes all NAs participating in the marketplace.

6.9.2.2 5: Bidding

Each NA either ignores the contract, or responds with an encrypted and signed *Bid* message. All bids are sent to the LA, and must be received by the fixed deadline specified in the flow contract. When all NAs have responded, or the deadline has been reached, the LA forwards all bids to the SPA.

6.9.2.3 6: Bid Selection

The SPA combines the reputations and bids of each participating NA, and selects a winner. The winning NA is notified by the SPA, using a signed *Accept* message. The accept message includes a flow contract reference, bid reference, and the address and public key of the UTA.

6.9.3 Establishment to Termination

With the winning NA or NAs for the session notified, the end-user communications session can start. Once every flow is concluded, the session is terminated, and the reporting process begins.

6.9.3.1 7: Flow Establishment

Each winning NA sends a signed *Establish* message to the UTA. This includes information on how to use the network operator's services, and a copy of the *Accept* message to demonstrate that the NA is entitled to provide service. The user's mobile terminal then starts the communications flow, and makes use of the network services as normal.

6.9.3.2 8: Flow Release

At the end of the communications session, the UTA indicates that it wishes to terminate the flow using a signed *Release* message. This marks the end of the communications flow.

6.9.3.3 9: Terminate Session

The NA then contacts the SPA to indicate the end of the flow, using a signed *Terminate* message, including a copy of the *Release* message. This defines the end of the billing period, and the start of the auction report process.

6.9.3.4 10: Report

Reporting is a multi-step stage. First, the UTA sends a signed *Commitment Report* to the LA, which includes a reference to the NA and flow contract, and a binary 'passed/failed' flag. The LA forwards this report along with the flow contract to any interested parties, most importantly the MA, as a *Flow Report*.

Next, the MA updates the relevant NA's reputation, using the reputation function parameterised with the flow contract and the commitment report. The MA then sends a *Reputation Update* message to the LA, which forwards it to interested NAs.

Finally, the SPA decrypts and sends all bids to the LA as the *Auction Report*. Again, this is forwarded to NAs for verification.

6.9.3.5 11: Leave

At the end of the reporting stage, the SPA is instructed by the MA to leave the marketplace. This concludes the DMP protocol session.

6.9.4 Message Signing and Encryption

Many messages in the protocol flow must be signed, and some must be encrypted. For clarity, all messages are listed in Table 6.4, along with their cryptographic requirements.

6.10 Analysis

The security measures described above are aimed to directly counter the threats found in section 6.6. This section justifies each measure, demonstrat-

Step	Message	Signed	Encrypted
1	Connect	✓	✓
2	Migrate	✓	
3	Join		
	List Query		
	List Inform		
4	Flow Contract	✓	
5	Bid	✓	✓
6	Accept	✓	
7	Establish	✓	
8	Release	✓	
9	Terminate	✓	
10	Commitment Report	✓	
	Flow Report		
	Reputation Update		
	Auction Report	✓	
11	Leave		

Table 6.4: Encrypted and signed messages in the Digital Marketplace

ing that every security threat is dealt with. By doing so, it is shown that the measures described here will secure the Digital Marketplace under the threat model given in this chapter.

The security threats are again collected into the three components of the protocol: session initiation, contract negotation, and establishment to termination. Protocol steps with no specific security threats are omitted from this analysis: these are the *join* and *leave* steps.

6.10.1 Session Initiation

1: Connection Request

Authentication Replay By inspecting and saving the contents of the connection request, a malicious market operator could reuse the authentication parameters to pretend to be the mobile user (see section 6.6.1.1). This was solved by specifying that the connection request message must be encrypted such that only the service provider can read it (see section 6.8.3).

Message Replay Alternatively, the market operator could simply record the entire connection request message, and later resend it to the service provider (see section 6.6.1.1). With the security measures in place, this is not possible due to the replay-proof authentication scheme, again specified in section 6.8.3.

Modifying Connect Finally, the market operator could alter the connection request message before forwarding it to the service provider, in order to favour colluders (see section 6.6.5.2). The message authentication scheme specified in section 6.8.3 is employed to prevent such attacks; modification of the request message would be detected by the service provider.

2: Migrate

Selective NA List In the previous protocol design, the list of Network Operator Agents was transmitted with the migration request. This meant that the market operator could modify this list to the advantage of its colluders (see section 6.6.5.3). The counter-measure to this attack is to change the distribution of the NA list, as described in section 6.8.4.1.

6.10.2 Contract Negotiation

4: Flow Contract

Selective Contracts A dishonest service provider could previously send contracts only to certain network operators, excluding those not in collusion with it (see section 6.6.5.4). To counter this, flow contracts are distributed by a secure logging agent, as described in section 6.8.4.1. This ensures that every registered NA receives every flow contract.

Differing Contracts Similarly, a service provider could send different contracts to colluding network operators, effectively excluding others from winning the auction due to unfair terms (see section 6.6.5.4). Again, the centralised flow contract distribution described in section 6.8.4.1 ensures that this is not possible. The service provider only sends one contract, which is duplicated by the secure logging agent.

5: Bidding

Fake Bid In the bidding stage, a malicious network operator could place a bid on behalf of another network operator, in order to hinder its ability to compete in auctions (see section 6.6.1.2). This threat is countered by the message authentication mechanism described in section 6.8.3. The service provider can verify the signature on each bid to ensure that it was really placed by the appropriate network operator.

Bid Repudiation A network operator could place an overly aggressive bid, and later claim that it was placed by someone else (see section 6.6.2). This is also solved by the signature scheme described in section 6.8.3, as the non-repudiation feature ensures that only the real network operator could have placed the bid.

Bid Inspection While bidding is taking place, a network operator could wait for others to bid, and place a minimally better bid, in order to win with maximum profit (see section 6.6.3). The counter-measure to this attack is that all bids are encrypted to be readable only by the service provider, as described in section 6.8.3. Therefore, network operators cannot read each others' bids, and this threat is removed.

Modified Bid An attacker could modify the contents of a bid, in order to damage the network operator who placed it. This could cause them to win the auction at too low a price, or to lose the auction unfairly (see section 6.6.5.5). Again, digitally signing bids as described in section 6.8.3 ensures that they cannot be modified without detection, and counters this threat.

Deleted Bid Finally, the market operator could delete bids as they are placed, before reaching the service provider (see section 6.6.5.5). However, with the secure agent execution environment described in section 6.8.2, deletion of bids is not possible, and this threat is countered.

6: Bid Selection

Fake Acceptance An attacker could send a message indicating acceptance of a bid to a network operator, when in fact the bid was not accepted. This could lead to provision of service where it was not paid for (see section 6.6.1.3). The counter to this threat is the message authentication scheme described in section 6.8.3, which would allow the network operator to verify that the acceptance came from the service provider.

Unfair Selection The service provider could unfairly select the winning bid from all those presented to it (see section 6.6.5.6). This is countered by enforcing a public auction reporting process as described in section 6.8.4.2, which allows any interested party to verify that the winning bid was not clearly worse than any other bid, and therefore the bid selection was fair.

6.10.3 Establishment to Termination

7: Flow Establishment

Deleted Establish A malicious market operator could delete the establish message before it reaches the user, preventing service provision from taking place. This would damage the network operator's reputation (see section 6.6.5.7). This is again countered by using the secure agent execution environment described in section 6.8.2, which prevents deletion of messages in transit.

8: Flow Release

Fake Release An attacker could send a release message, purportedly from the user, which would end the communications flow. This would deny service to the user, and reduce the billing period of the network operator (see section 6.6.1.4. This is prevented by requiring that the flow release message is signed by the user, as described in section 6.8.3. Then, the network operator must only accept verified release messages.

9: Terminate Session

Fake Termination Similarly to the previous thereat, an attacker could send a false session termination request, reducing the network operator's billing period (see section 6.6.1.5). Again, this is countered by requiring that the terminate message is signed, as described in section 6.8.3.

10: Report

Fake Report Another threat due to impersonation exists in the reporting stage. An attacker could submit a session report in order to manipulate the reputation system (see section 6.6.1.6). Once more, authenticating the message counters this threat, as described in section 6.8.3.

Falsified Report The previous reputation system required that the network operator submit reports on its own performance, which clearly encourages inflated reports (see section 6.6.4). This is not an issue with the secured protocol, as the reputation reports are required to come from the mobile user, as described in section 6.8.5.

Ignored Report Commitment reports were previously sent to the market operator in order to update the relevant network operator's reputation. A colluding MA could ignore negative reports, to falsely inflate its colluders' reputations (see section 6.6.5.8). This is countered by requiring a public reputation update phase, as described in section 6.8.4.3. This allows interested parties to ensure that the network operator's reputation was updated fairly.

Modified Report Similarly, the user's commitment report could be modified by the MA to favour its colluders (see section 6.6.5.8). To counter this, it is required that the user's report be signed, as described in section 6.8.3.

Step	Threat	Section	Measure	Section
1	Authentication replay	6.6.1.1	Encryption	6.8.3
	Message replay	6.6.1.1	Authentication	6.8.3
	Modifying connect	6.6.5.2	Authentication	6.8.3
2	Selective NA list	6.6.5.3	Contract distribution	6.8.4.1
4	Selective contracts	6.6.5.4	Contract distribution	6.8.4.1
	Differing contracts	6.6.5.4	Contract distribution	6.8.4.1
5	Fake bid	6.6.1.2	Authentication	6.8.3
	Bid repudiation	6.6.2	Authentication	6.8.3
	Bid inspection	6.6.3	Encryption	6.8.3
	Modified bid	6.6.5.5	Authentication	6.8.3
	Deleted bid	6.6.5.5	Secure platform	6.8.2
6	Fake acceptance	6.6.1.3	Authentication	6.8.3
	Unfair selection	6.6.5.6	Auction report	6.8.4.2
7	Deleted establish	6.6.5.7	Secure platform	6.8.2
8	Fake release	6.6.1.4	Authentication	6.8.3
9	Fake termination	6.6.1.5	Authentication	6.8.3
10	Fake report	6.6.1.6	Authentication	6.8.3
	Falsified report	6.6.4	User reporting	6.8.5
	Ignored report	6.6.5.8	Public update	6.8.4.3
	Modified report	6.6.5.8	Public update	6.8.4.3

Table 6.5: DMP security threats and counter-measures

Then, the public update phase described in section 6.8.4.3 would allow any interested party to verify that the report was not modified.

6.10.4 Summary

In this section, each security threat has been reiterated and summarised, and linked to a countering security measure. Every threat has been countered by the work presented in this chapter, and therefore the Digital Marketplace protocol is secured under this threat model. A summary of the threats and counter-measures, along with references to the relevant sections of this chapter, is given in Table 6.5.

6.11 Conclusions

The Digital Marketplace has the potential to revolutionise mobile network service provision. By creating a freely-operating market, and lowering the barrier to entry, competition will be increased and costs lowered. This in turn will encourage greater uptake of next-generation mobile services, and the benefits to the mobile user can only increase.

However, without a securely designed protocol, the market cannot be relied upon to operate freely. This is a major obstacle to the adoption of the DMP: until it can be demonstrated that the marketplace will work fairly for all, service providers and network operators will not be interested.

This chapter has presented a thorough security analysis of the existing Digital Marketplace design. From the threats found in the analysis stage, a set of security requirements were derived. A detailed series of countermeasures to these threats were presented and justified, and a modified and secured protocol was described. Finally, each security threat was taken in turn and shown to be countered by a matching security measure.

At this stage, two areas of work must be covered before the Digital Marketplace can be used as an anonymous network access scheme. As noted above, a reputation system which is appropriate for the specific needs of the Digital Marketplace must be designed. This is a complex problem, and is dealt with in detail in chapter 7.

With a reputation system, and the above secured protocol, the DMP can be considered secure. However, anonymity has not been considered at this stage. This requires a re-examination of the security properties of the Digital Marketplace, and the invention of some method for enabling anonymous network access. This is covered in chapter 8.

Chapter 7

Reputation in the Digital Marketplace

Contents

As described in section 5.3, the Digital Marketplace allows users and their service providers to negotiate for network service in real-time. Users can connect to a mobile network which is owned by any participant in the market, and do so at the best price and performance available to them. One potential problem raised by this approach is that the user has no long-term relationship with the network operator, and therefore does not necessarily trust them to provide service as contracted.

In traditional markets, the customer has ample time to assess the trustworthiness of available sellers. However, each DMP transaction must complete quickly and without interaction, or the interruption to the user will be noticeable. Therefore, the marketplace must provide some way to allow a user's agent to select a bid based on the estimated reliability of each network operator. The best way to do this is to employ a marketplace-wide reputation system, which represents past behaviour of each network operator, and provides it to bid-selecting agents.

7.1 Introduction

The Digital Marketplace reputation system has a number of constraints. It must be simple enough to be understood, compared, and updated rapidly, to ensure that the marketplace protocol completes rapidly. Each operator's reputation must be updated based on exactly three variables: the previous reputation, a flow contract, and a binary success/failure report. Finally, the reputation score must reflect a fairly complex system, in such a way that a service provider can quickly distinguish between reliable and unreliable network operators.

7.1.1 Reputation in the Digital Marketplace

Section 6.8.5 describes how a reputation system should integrate with the Digital Marketplace protocol. The reputation of each Network Operator Agent (NA) is calculated by the Market Operator Agent (MA), and recorded by a secure Logging Agent (LA). All reputations are forwarded to the service provider agent (SPA), along with the NAs' addresses and public keys, to allow the SPA to consider the reputation along with the bids. NAs also receive copies of the reputations, to allow them to reflect on the current market status, and also to ensure that their reputation is updated fairly.

The SPA reports back to the MA on behalf of the user on the quality-of-service received for the call; this report consists of a copy of the flow contract c and a binary success/failure indicator σ. The MA then calculates the new reputation value r' for the relevant NA, based on the previous reputation r and the other two parameters: that is, $r' = f(r, c, \sigma)$, for some reputation update function f.

A reputation update message is then sent to the Logging Agent, which updates its stored reputation value for the NA and forwards the report to interested parties. See Figure 7.1 for a graphical representation of this messaging process.

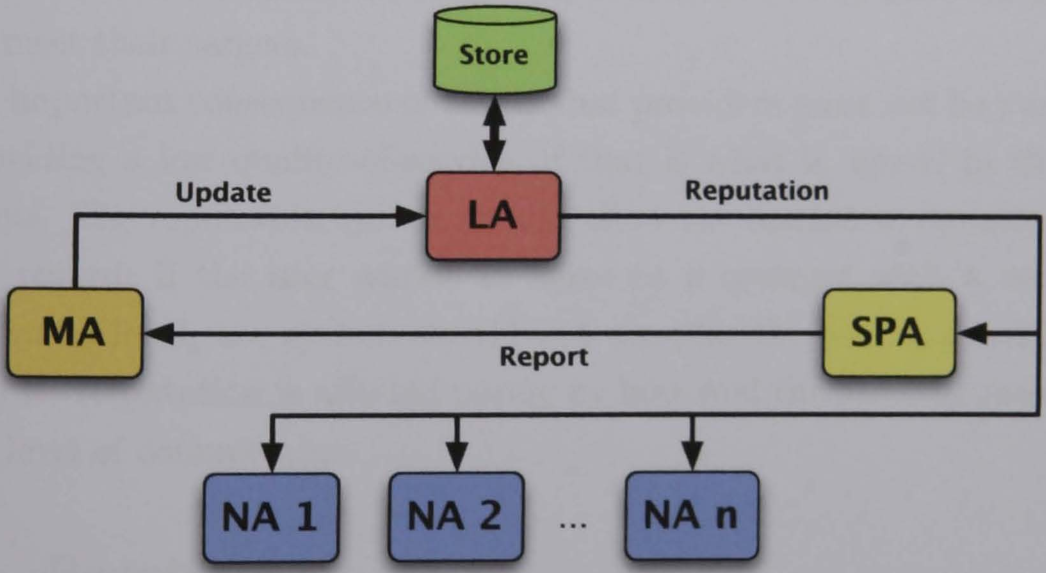


Figure 7.1: Flow of reputation-related messages

7.1.2 Commitment Levels

Flow contracts in the DMP include commitment levels, and the reputation system must reflect this. As described in section 6.5.2.3, commitment level is a statistical goal of service provision. One provider may agree to a 95% commitment level, while another may offer 90% commitment at a lower cost. The user's agent may choose between the more reliable provider, or the cheaper one. To exactly meet their commitments, the first provider must provide the agreed service for 95% of all calls, but the second only needs to achieve 90% success rate. If service is provided more often than this, commitments are exceeded, and the operator is considered more reliable. Otherwise, if failure occurs more often, the operator is considered less reliable.

The goal of the reputation system is to differentiate between varying providers by their past reliability. Providers who achieve higher quality-of-service levels should have better reputation scores, and those who do not reach their commitments should have poorer reputation scores. Therefore, it is reasonable to require that if the two providers above exactly achieve their respective commitment levels, their reputations should be both equal and at the centre of the scale. This allows room for providers to demonstrate reliability above their commitment, and to catch those providers who consistently fail to meet their targets.

An important consequence of this is that providers must not be punished for providing a low quality-of-service, if that is what is agreed in the flow contracts. The reputation system should allow the market to operate freely in this regard; if the user wishes to agree to a contract with a very low commitment level, the system should not penalise the service provider for offering it. Reputation is affected purely by how well the provider meets the agreed level of commitment.

7.1.3 Requirements

There are four requirements which must be met for a function to be deemed suitable for the Digital Marketplace reputation system. These are:

Simple Output Reputation should be a scalar value, to simplify comparison

Success Dependent Change in reputation should relate to success and commitment, such that a network provider is rewarded or punished statistically and symmetrically based on success/failure and the commitment level agreed

Commitment Scaling Contracts with higher commitment levels must result in a larger change in reputation than those with a lower commitment level

Responsive Reputation must reflect recent behaviour, rather than the full lifetime of the network operator

7.2 Related Work

The most widely-used electronic marketplace reputation system is on eBay, the internet auction site. Analytical work by Dellarocas has shown that binary reputation systems such as eBay's are capable of being well-functioning, if sellers and buyers assess feedback scores correctly[33]. Empirical examinations of the system by Resnick et al. have shown the system to function adequately, despite its imperfections[97][98].

However, while lessons can be learned from the eBay system, it cannot be used in the Digital Marketplace. Two differences in the marketplaces are of particular importance. First, eBay does not differentiate between buyers and sellers, but the DMP separates these two classes of actor; also, unlike eBay, the buyer/seller roles are never reversed. The second difference is that the DMP enforces at least half-completion of the transaction. Bidders can win auctions on eBay, and fail to pay; this cannot happen in the DMP, and so recording non-payment is not an issue.

Initial research into DMP security had suggested need for reputations for all users[2]. This was presented as a counter-measure to reputation attacks, where users would maliciously file negative reports against certain network operators. Further analysis of the situation shows this scenario to be unlikely. The DMP enforces payment, by blacklisting non-paying service providers; therefore, such an attack would be expensive. Additionally, with a well-designed reputation function, which is responsive to good network operator performance, an attack against the system would have to be of such immense scale as to be unfeasible. The money spent by the network operator on this attack would be better spent undercutting its competitors in the market.

From these analyses, it is clear that the DMP does not require a bilateral reputation system. The sets of buyers and sellers do not intersect, and reneging on payment is not an issue. Additionally, false-report attacks on the system are made economically non-viable by the payment structure. Therefore, a DMP reputation system need not record the reputation of users, only those of network operators.

7.3 Previous Reputation Function

Initial work on the Digital Marketplace described a simple linear reputation function[77]. The function outputs a network operator's *penalty*, which is calculated from a sequence of commitment reports. Each call has a commitment report c_i, which is recorded as 0 (for failure) or 1 (for success). A network operator's penalty p for depth d is then calculated by the following equation:

$$p = \frac{1}{d}\sum_{i=0}^{d} 1 - c_i$$

This means that a penalty of 1 is applied for a failure, and no penalty for success; then, the penalty score is simply the average of these penalties over the most recent d commitment reports. The depth parameter is used to ensure that the network operator is not permanently punished for poor performance; only recent calls are taken into account.

However, this function does not take into account the commitment level for a flow contract, which makes it unsuitable for use in a secure Digital Marketplace. This is demonstrated, by comparison with the reputation function proposed here, in simulation results presented in section 7.5.

7.4 Proposed Reputation Function

As the previously-published reputation systems are inappropriate for the Digital Marketplace, a new function must be designed. This chapter proposes a reputation update function which fits with the previously-described reputation system. The reputation update function takes three parameters: the current reputation r, user success report σ, and a commitment level c.

Reputation r is restricted to lie between 0 and 100, where 0 is the worst possible reputation, and 100 is the best. The success report σ is 1 if the call completed successfully, and 0 if the call failed to complete. The commitment level $0 < c < 1$ represents the expected probability of success.

Reputation is updated using the formula:

$$r' = \begin{cases} \min(r+1, 100) & \text{if } \sigma = 1 \\ \max(r - \frac{c}{1-c}, 0) & \text{if } \sigma = 0 \end{cases}$$

For example, with $c = 0.8$, and a 5-call sequence of $\sigma = (1,1,1,1,0)$, the reputation updates will be $(1,1,1,1,-4) = 0$. The function ensures that failure causes reputation loss to a degree proportional to commitment: failure with $c = 0.8$ leads to a reputation update of $r' = r - 4$, because we would expect four of five calls to succeed.

This expected change in reputation can be expressed more formally. If $\sigma = 1$, then the change in reputation $r_\delta = 1$; if $\sigma = 0$, then $r_\delta = -\frac{c}{1-c}$. Let s be the probability of success; that is, the probability that $\sigma = 1$ for every call. Then, the expected value of r_δ is given as

$$E(r_\delta) = s \times 1 + (1-s)(-\frac{c}{1-c}).$$

If the probability of success exactly meets the commitment level, then $s = c$, and

$$\begin{aligned} E(r_\delta) &= c \times 1 + (1-c)(-\tfrac{c}{1-c}) \\ &= c - c = 0. \end{aligned}$$

This property means that, over the long term, an operator with success rate exactly equal to commitment level will see no change in their reputation. Those who fail to meet commitment levels will suffer reputation loss, and those who consistently exceed requirements will increase their reputation. The magnitude of change increases with the deviation from the agreed commitment level, asymptotically towards the upper and lower limit for increased and decreased reliability, respectively.

Initial reputation value for new network operators is set at 50, the centre of the range. This value is chosen to ensure that newly-joining operators are not given any advantage nor any penalty. However, this also means that a poorly-performing operator (one with a reputation below 50) will have an incentive to leave the market and rejoin to reset their reputation. However, external constraints to the reputation system ensure that this will not happen. Entering the market requires an offline key exchange with the market operator, which implies a long delay and presents an opportunity for the market operator to detect such fraudulent activity.

This function meets the four requirements specified in section 7.1.3. The output of the function is a scalar value, which makes comparison between network operators very simple. Each reputation update is dependent on the success or failure of the call, and negative results are also dependent on the commitment. The magnitude of reputation degradation scales with commitment value, such that higher-commitment calls can result in greater level of punishment for failure. Finally, the reputation value is limited between 0 and 100, which ensures that the reputation reflects only a limited number of contracts.

7.5 Simulation and Analysis

The reputation function was simulated for a number of representative situations. Results are averaged over 250 runs, and displayed with a 95% confidence interval. Simulated network operators are paramaterised with initial reputation, call success probability, and a function for temporal variation in success probability.

Several simulation scenarios were tried, with varying levels of agreed commitment, success, and over a variable number of calls. Note that call success probability is independent of commitment level: a network operator with $s = 0.95$ will have 5% of its calls fail, regardless of commitment. Also, no negotiations took place for these simulations, as only the reputation function is under test; therefore, network operators are entirely independent.

Other values of the constants for this function were examined: the limit of 0–100 for r and the weighting of r_δ can be changed to any arbitrary numbers. These were chosen for two reasons. First, a scale between 0 and 100 is easily understood as analogous to a percentage. More importantly, the response and stability of the function with these chosen values was found to be appropriate for the requirements of the system.

Along with the proposed reputation function, the previously-published penalty-based system[77] was simulated for comparison purposes. As described in section 7.3, the penalty depth can be configured; for these simulations, a depth of 200 was chosen, in accordance with [77].

Five sets of simulation results are presented below. First is a simple example of several network operators, each with a different fixed success level which is close to the agreed fixed commitment. Then, the opposite approach is taken: several network operators with fixed success rate, but each responding to flow contracts with different commitment levels. This is followed by an examination of the response to an instantaneous degradation of success rate, while commitment remains fixed. Next, the functions are compared for recovery time from minimum reputation, for various success rates. Finally, the effects of a temporary degradation followed by recovery are presented.

7.5.1 Success Level Spanning Commitment

Figure 7.2 shows reputation for five network operators over 3000 calls with 90% commitment level. The operators have success levels spanning 2% either side of the commitment level, with one operator exactly matching the requirement.

The operator who matches the required commitment has a reputation of around 50%, while those who exceed and fail have appropriately higher and

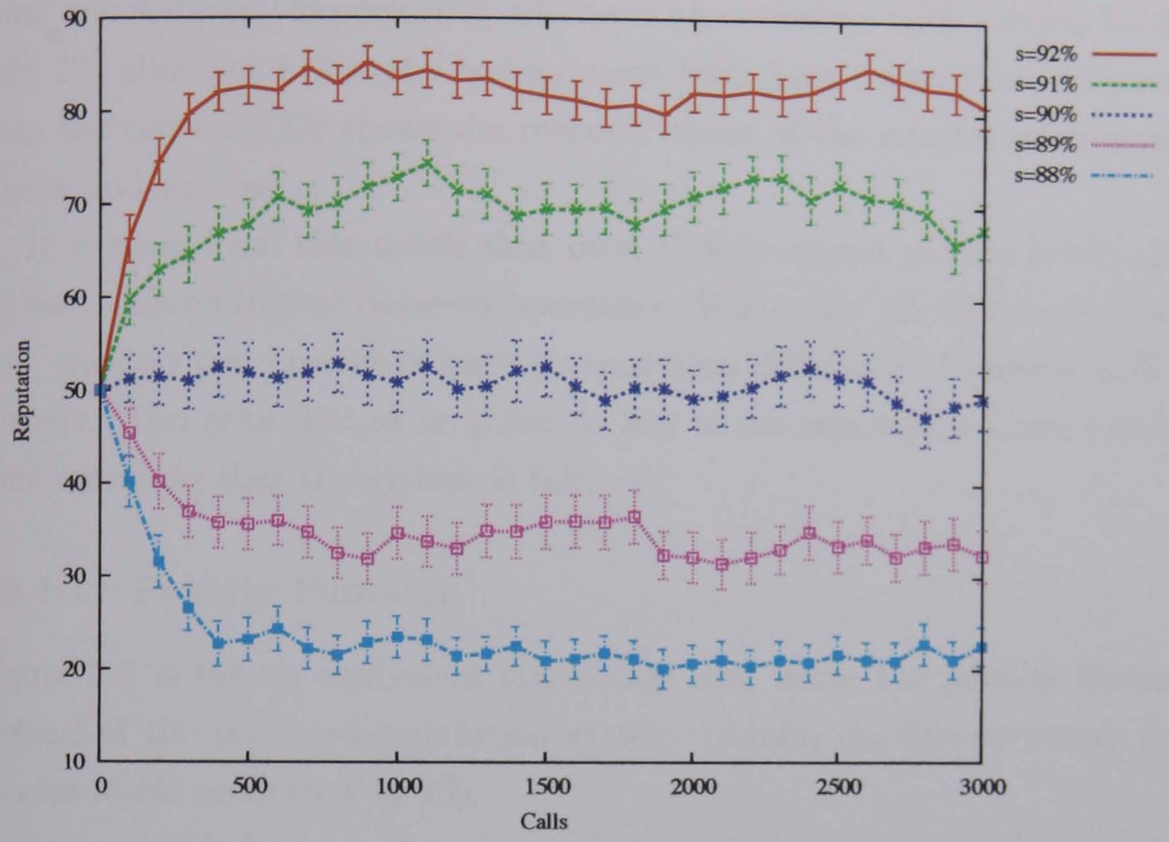


Figure 7.2: Success level spanning commitment

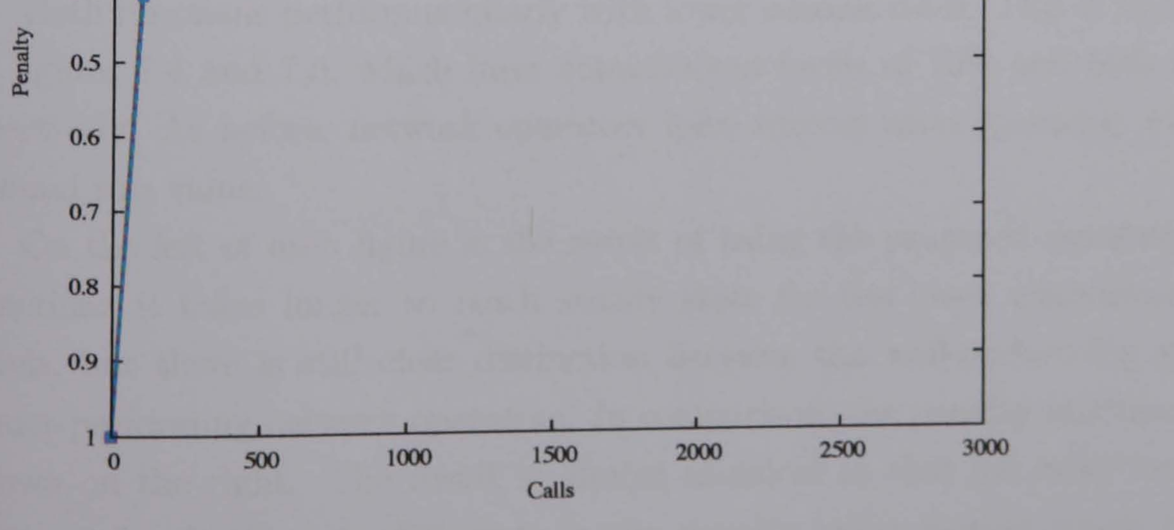


Figure 7.3: Penalty function for success level spanning commitment

lower reputations. Importantly, the network operators with success level of only 2% above or below the commitment level have reputations 30% away from the centre. This shows the responsiveness of the reputation system to failure and success.

It is clear from this result that even tightly-spaced success levels allow for easy differentiation between operators. With only 1% difference in success, the top two operators have a reputation difference of around 12% on average. Also note that at no point do any of the reputation scores overlap, demonstrating that the system is fair.

7.5.1.1 Penalty Function

Figure 7.3 is for an equivalent simulation run, using the penalty function instead of the proposed reputation system. Again, the five operators have success levels separated by 1%.

As can easily be seen, there is very little separation between the operators for this function: all five network operators have penalty values ranging from 0.075 to 0.125. The penalty values do not overlap at any point. However, very little of the possible penalty range is used; it would be desirable that those operators whose success is marginally below the commitment level have a reputation which is clearly distinguishable from those who exactly meet their commitments. This property is given by this chapter's proposed function, but not by the penalty system.

Both functions perform similarly with lower success rates. This is shown in figures 7.4 and 7.5, which have commitment levels of 70% and 50% respectively. As before, network operators have success rates spanning ±2% around this value.

On the left of each figure is the result of using the proposed reputation function: it takes longer to reach steady state for the lower commitment levels, but there is still clear distinction between the well-performing and under-performing network operators. In comparison, the penalty function is shown on the right. The result is almost identical to that for every commitment level; the only difference is the penalty value around which the

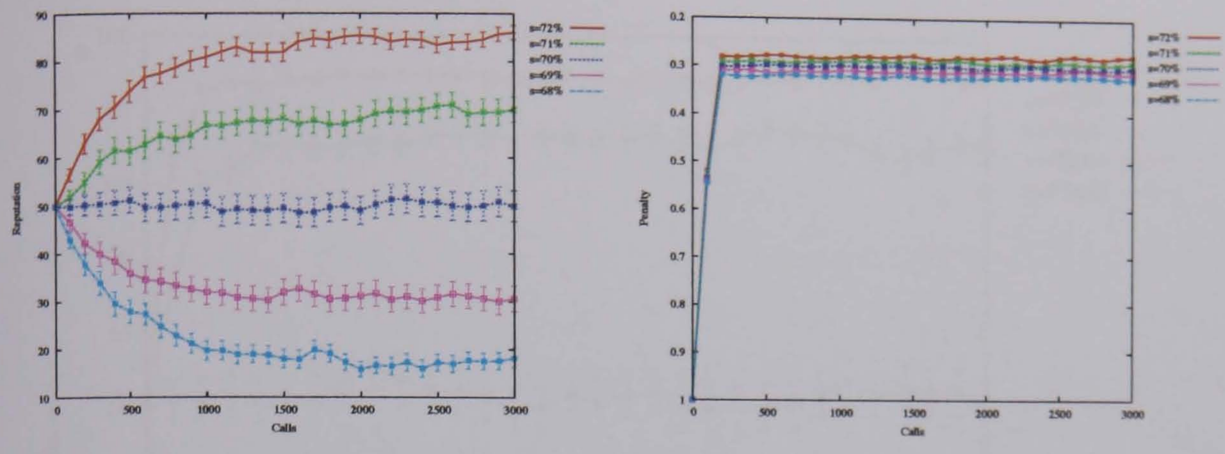


Figure 7.4: Proposed (left) and previous (right) reputation functions are simulated, with success level spanning a fixed commitment level of 70%

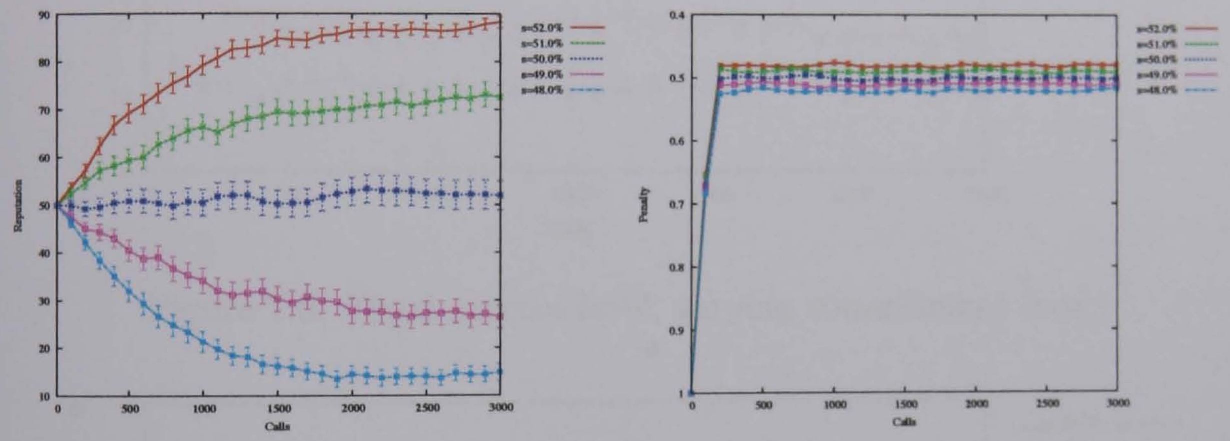


Figure 7.5: Proposed (left) and previous (right) reputation functions are simulated, with success level spanning a fixed commitment level of 50%

operators are centred.

7.5.2 Differing Commitment Level

The success of the proposed reputation function is most clear in situations with different commitment levels for each network operator. Figure 7.6 is the result of a simulation run where all of the five network operators had a success rate of 90%, but each was handling contracts of a different commitment level. The commitment level varies from 80% to 95%, in steps of 2.5%.

As expected, the result for the proposed function is very similar to the situation where commitment is fixed and success varies. Operators whose commitment exactly matches their success level have a reputation near the

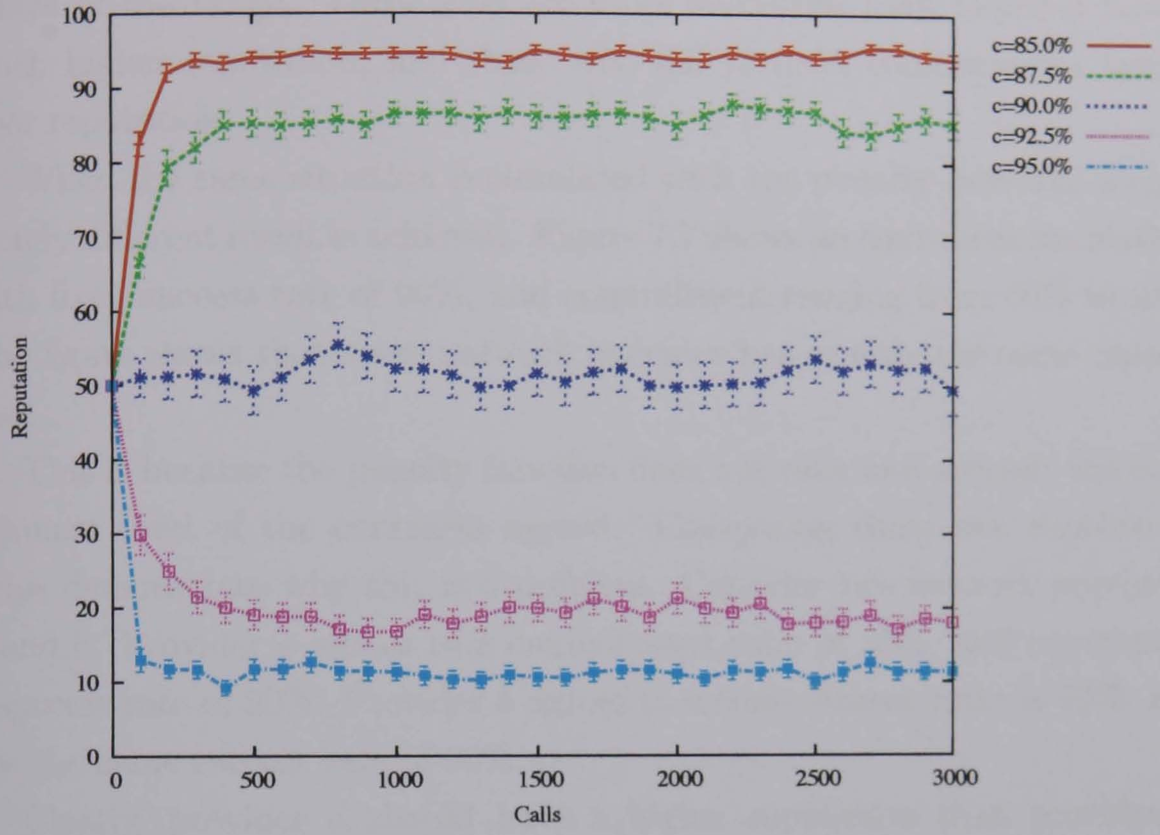


Figure 7.6: Fixed success level, varying commitment level

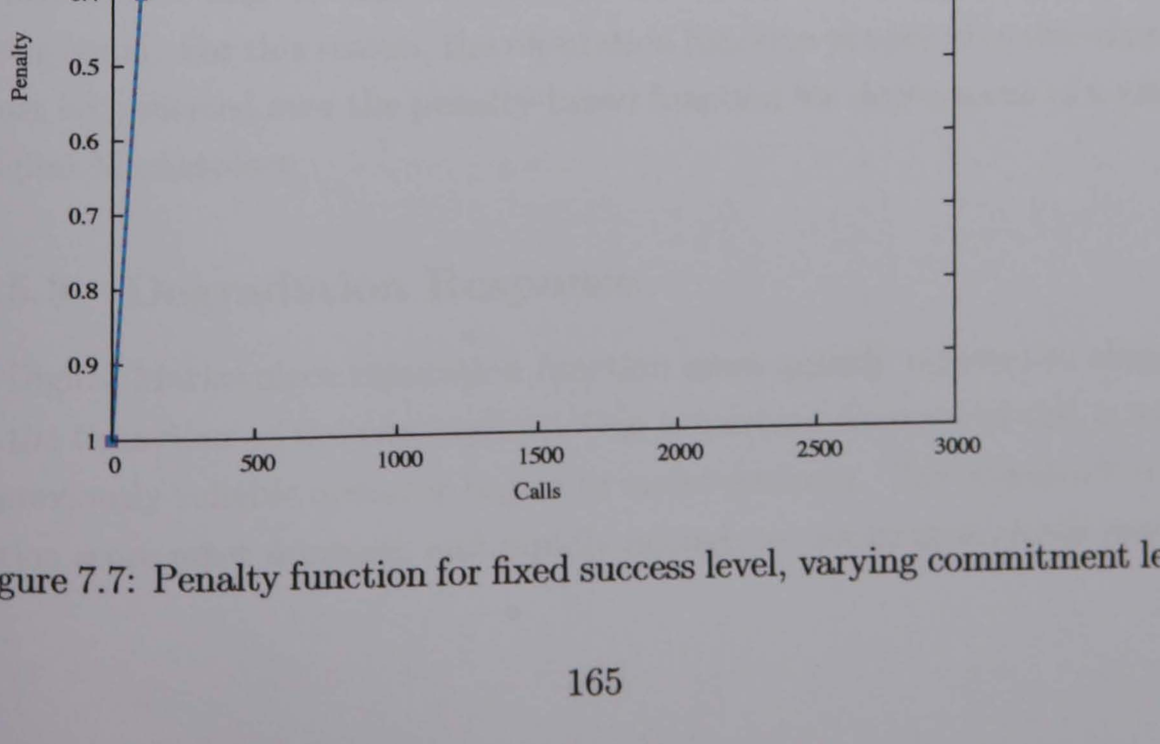


Figure 7.7: Penalty function for fixed success level, varying commitment level

centre of the range. Those who are more successful than required have a much higher reputation, and those who fail to meet commitments have a poor reputation.

When the same situation is simulated with the penalty function, a completely different result is achieved. Figure 7.7 shows an equivalent simulation, with fixed success rate of 90%, and commitment ranging from 80% to 95%. The figure shows that every network operator has exactly the same reputation.

This is because the penalty function does not take into account the commitment level of the contracts agreed. Comparing these two simulations helps demonstrate why this is a problem. Consider two network providers, *a* and *b*. Provider *a* agrees to a commitment ratio of 85%, and experiences a success rate of 90%. Provider *b* agrees to a commitment ratio of 95%, and has the same success rate of 90%.

Clearly, provider *a* should have a higher reputation than provider *b*. Provider *a* has not only met its contractual agreements, but exceeded them; provider *b* has failed to yield the service level it agreed to. The reputation function proposed in this chapter meets this requirement, as is clear from Figure 7.6.

However, the penalty function gives all network providers with equal success level the same reputation score, as shown in Figure 7.7. This makes it impossible for a service provider to distinguish between a network operator which is exceeding its commitments, and one which is failing, all other things being equal. For this reason, the reputation function proposed in this chapter must be preferred over the penalty-based function for deployment in a viable Digital Marketplace.

7.5.3 Degradation Response

A Digital Marketplace reputation function must quickly respond to changes in the behaviour of the participants. One important example of this is when a previously-reliable operator begins to under-perform. The operator's reputation score must decrease, and rapidly enough to ensure that recent market behaviour is reflected.

Figures 7.8 and 7.9 display the simulation results for a scenario where a sudden drop in success level occurs. The commitment level is fixed at 85%, with operator success levels initially set to 85%, 90%, and 95%. After 3000 calls, these success levels drop by 10 points, to 75%, 80%, and 85%, respectively.

This chapter's proposed reputation function is shown in Figure 7.8, and the initial results are as expected: the two operators with success level above commitment have high reputations, and the operator with success equal to commitment has a centre-valued reputation. Likewise, the steady-state results after success levels are lowered are desired: only the most successful operator, at 85%, retains a good reputation score.

One interesting feature of the results is that success level affects response rates. The network operator with poorest performance (75% success level) suffers a rapid drop in reputation. Within 100 calls, the operator's reputation drops from 50 to 10, then within another 100 calls lowers slightly to its steady-state reputation of around 5.

The other two network operators have a less rapid loss of reputation. The network operator with final success level of 80% drops to a reputation of 56 within 100 calls, and takes a total of 500 calls to reach a steady-state value of 8. The third network operator drops only 16 points within 100 calls, from 98 to 82; its steady-state value of 50 is reached after 1000 calls.

These results show that the proposed reputation function responds very rapidly when the success rate is significantly below the agreed commitment rate; less rapidly when success is marginally below the commitment; and slowly when success is equal to commitment. Most important is the fact that operators which perform very poorly have their reputation scores updated very quickly to reflect their change in behaviour.

Simulation results for the penalty-based system in the same circumstances are shown in Figure 7.9. All three operators take around 200 calls to drop from their initial reputation score to their steady lower score. Therefore, the penalty function's response delay is approximately equal to that of the proposed function, when considering network operators with success level 10

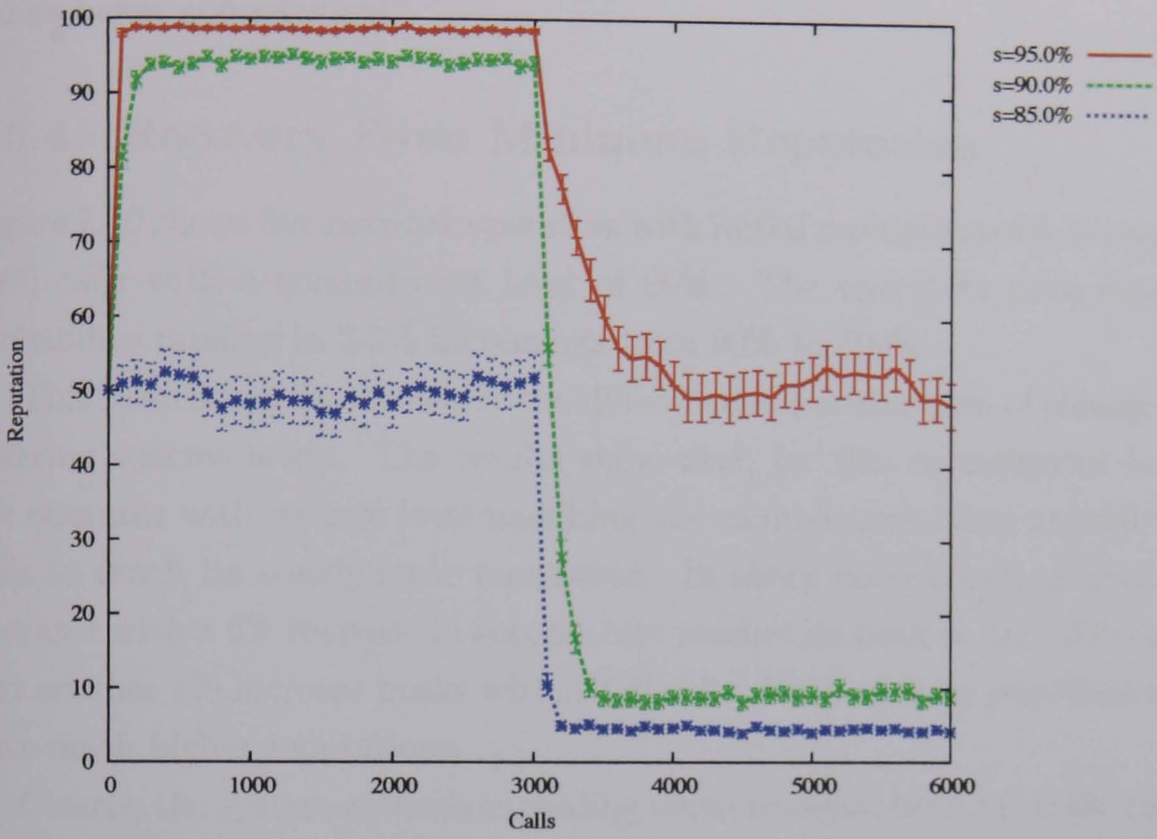


Figure 7.8: Success degradation by 10 points at 3000 calls

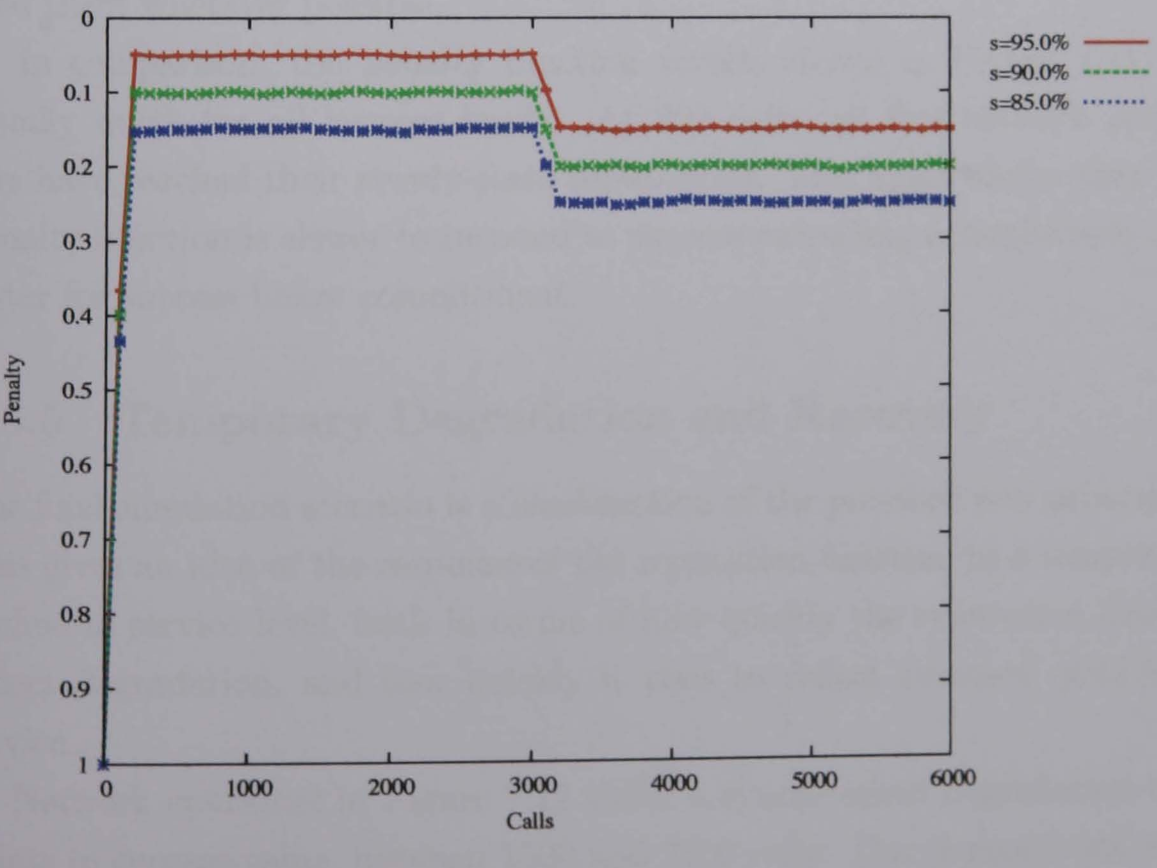


Figure 7.9: Penalty function for success degradation scenario

points below commitment.

7.5.4 Recovery From Minimum Reputation

Figure 7.10 shows five network operators with initial reputation of 0, servicing 1600 calls with a commitment level of 90%. The operators have success probability ranging in 2.5% increments from 90% to 100%.

This simulation shows clearly the difference in positive rate of change for differing success levels. The results show that, for this commitment level, the operator with success level matching the commitment takes around 600 calls to reach its steady-state reputation. In sharp comparison, a network operator with a 5% increase in success rate reaches its peak at only 300 calls, and another 5% increase peaks within 100 calls. Both of these providers also have much higher reputations.

Clearly, the system rewards exceeding commitments, both in stable reputation value, and in the rate at which this is achieved. This would encourage operators to realistically assess their acheivable commitment levels, and exceed them wherever possible.

In comparison, the penalty function results shown in Figure 7.11 are equally quick for all success levels. At 200 calls, all five network operators have reached their steady-state reputations. This again shows that the penalty function is slower to respond to success exceeding commitment, and faster for success below commitment.

7.5.5 Temporary Degradation and Recovery

The final simulation scenario is a combination of the previous two situations. This gives an idea of the response of the reputation function to a temporary decline in service level, both in terms of how quickly the reputation falls to reflect degradation, and how quickly it rises to reflect resumed quality of service.

Network operators in Figure 7.12 suffer a synchronised degradation of 5 points in success value, between 1500 and 2500 calls. The commitment level is fixed at 90%, and the network operators' initial success values are 100%,

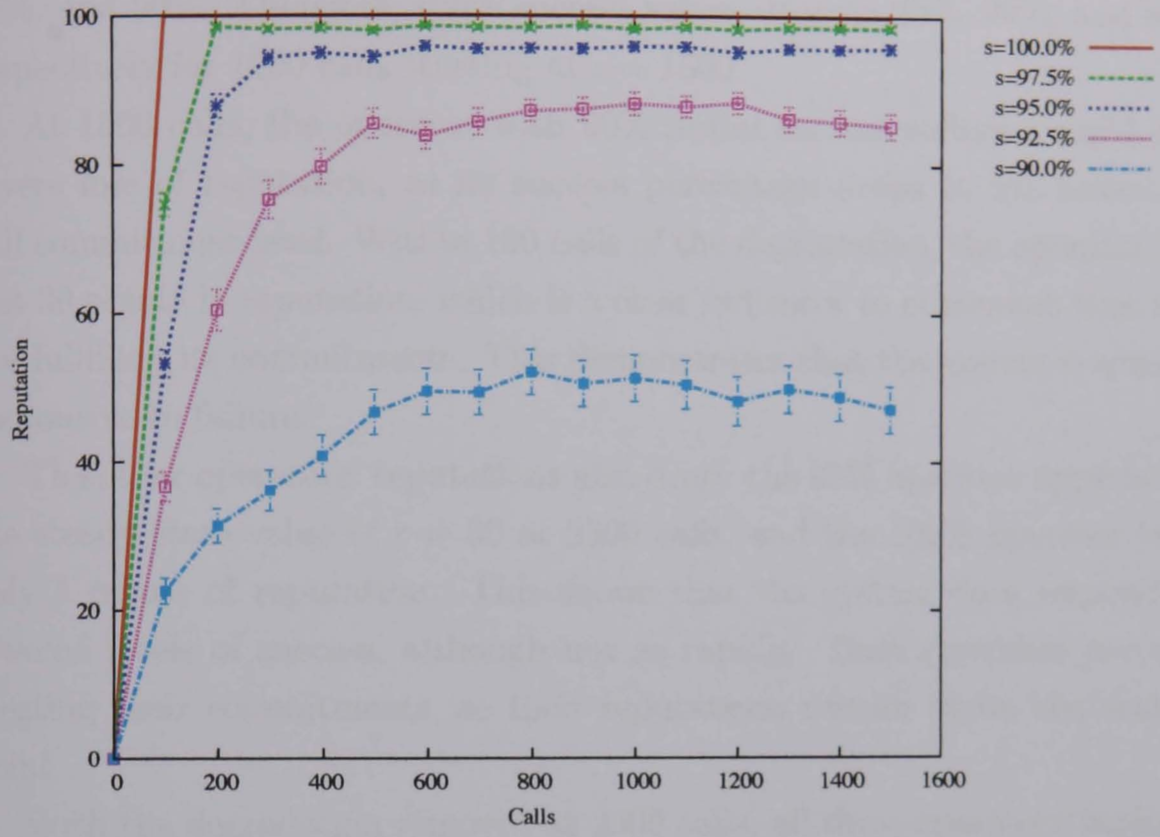


Figure 7.10: Recovery from minimum reputation

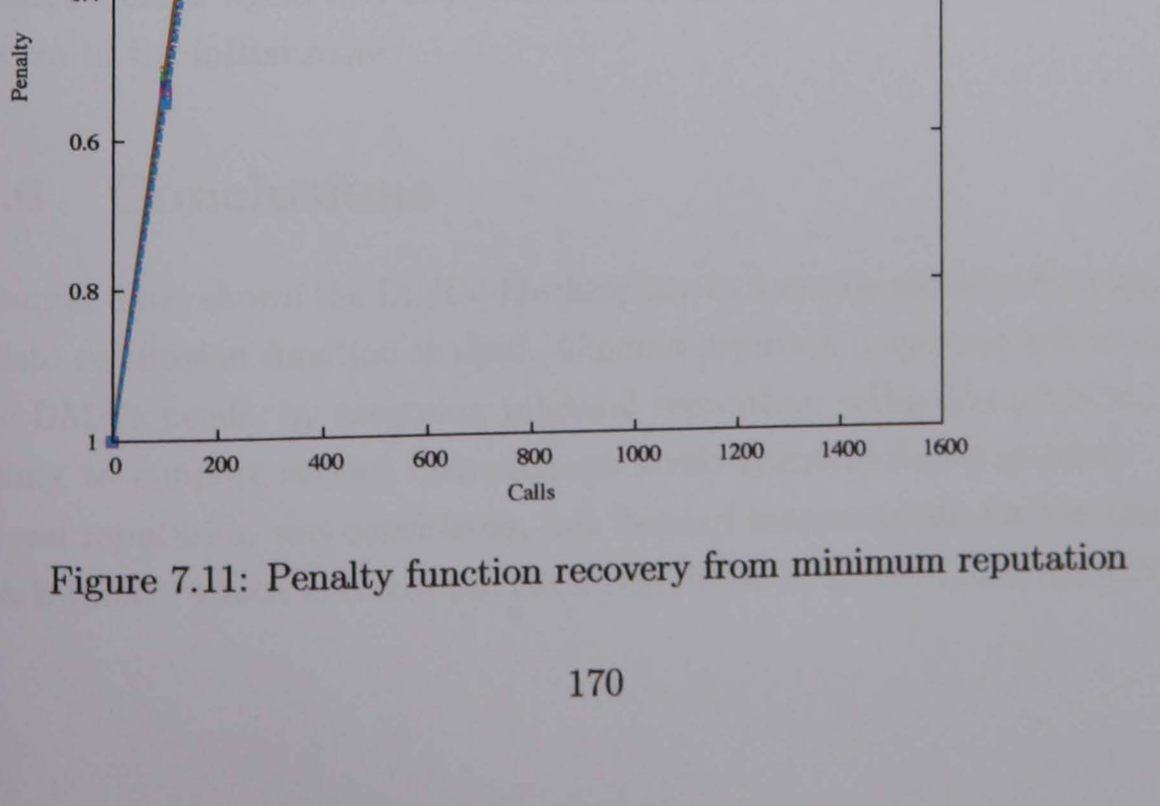


Figure 7.11: Penalty function recovery from minimum reputation

95%, and 90%. Therefore, these success values drop to 95%, 90%, and 85% respectively for 1000 calls starting at $c = 1500$.

At 1500 calls, the operator with 90% initial success suffers a rapid and severe loss of reputation, as its success percentage drops to 5% below the call commitment level. Within 100 calls of the degradation, the operator has lost 30 points in reputation, which is a clear indicator to customers that it is not fulfilling its commitments. This demonstrates that the system is quickly responsive to failure.

The other operators' reputations also drop: the 95% operator approaches the steady-state value of $r = 50$ at 2300 calls, and the 100% operator loses only 5 points of reputation. This shows that the system does respond to lowered levels of success, although not as rapidly. Both providers are still meeting their commitments, so their reputations remain above the central point.

With the degradation removed at 2500 calls, all three operators begin to recover. The recovery rate is clearly proportional to the difference between success and commitment: the 90% operator has the slowest recovery, reaching its initial reputation around 1000 calls later. The 95% operator recovers within 300 calls, and the 100% operator within 100.

Figure 7.13 shows the results of the same scenario under the penalty function. As before, it is notable that all providers suffer equal reputation loss, as commitment level is not taken into account. Also, the reputation change time is again 200 calls, both to reach the lower reputation, and to return to the initial score.

7.6 Conclusions

Chapter 6 has shown the Digital Marketplace to function securely if an appropriate reputation function is used. Current reputation systems fail to meet the DMP's needs, by assuming bilateral reputation (eBay-like systems), or failing to support service commitment levels (penalty-depth system). Bilateral reputation was considered, but deemed inappropriate for the Digital Marketplace, which enforces payment regardless of success; and recognising

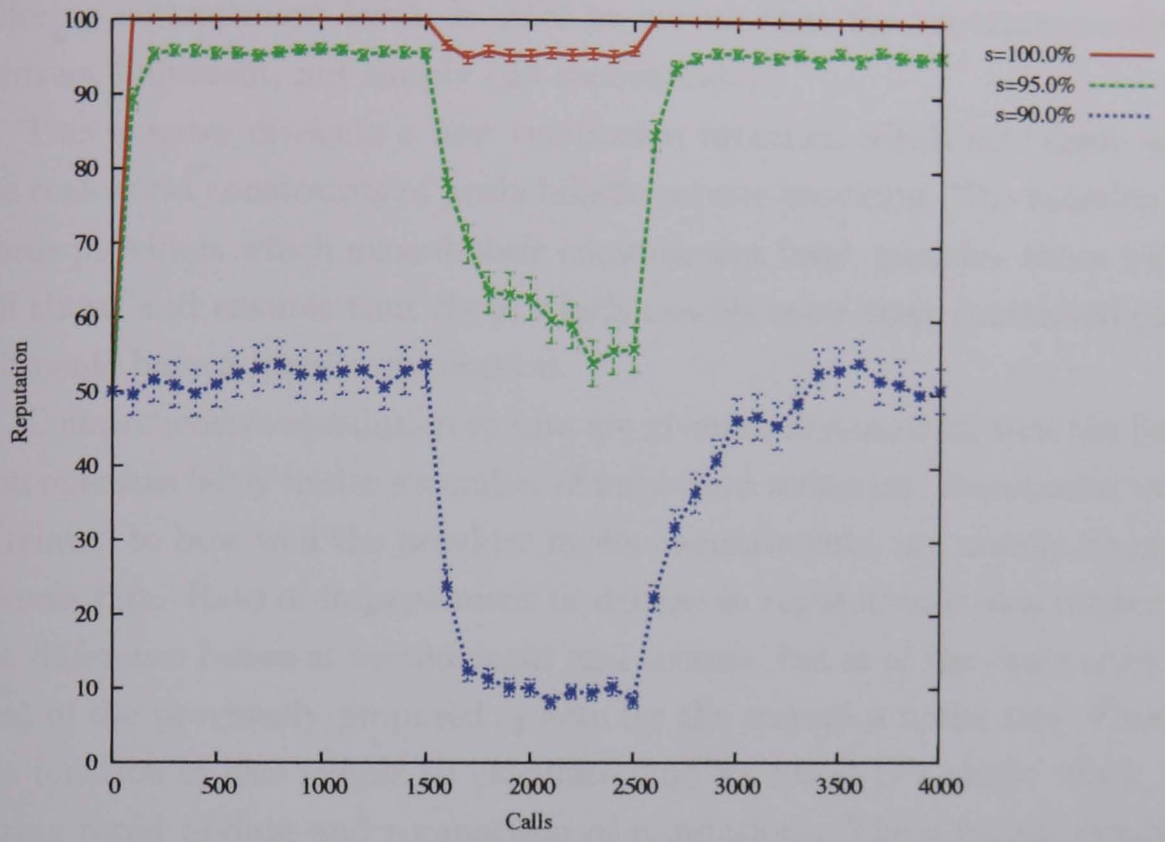


Figure 7.12: Temporary degradation between 1500 and 2500 calls

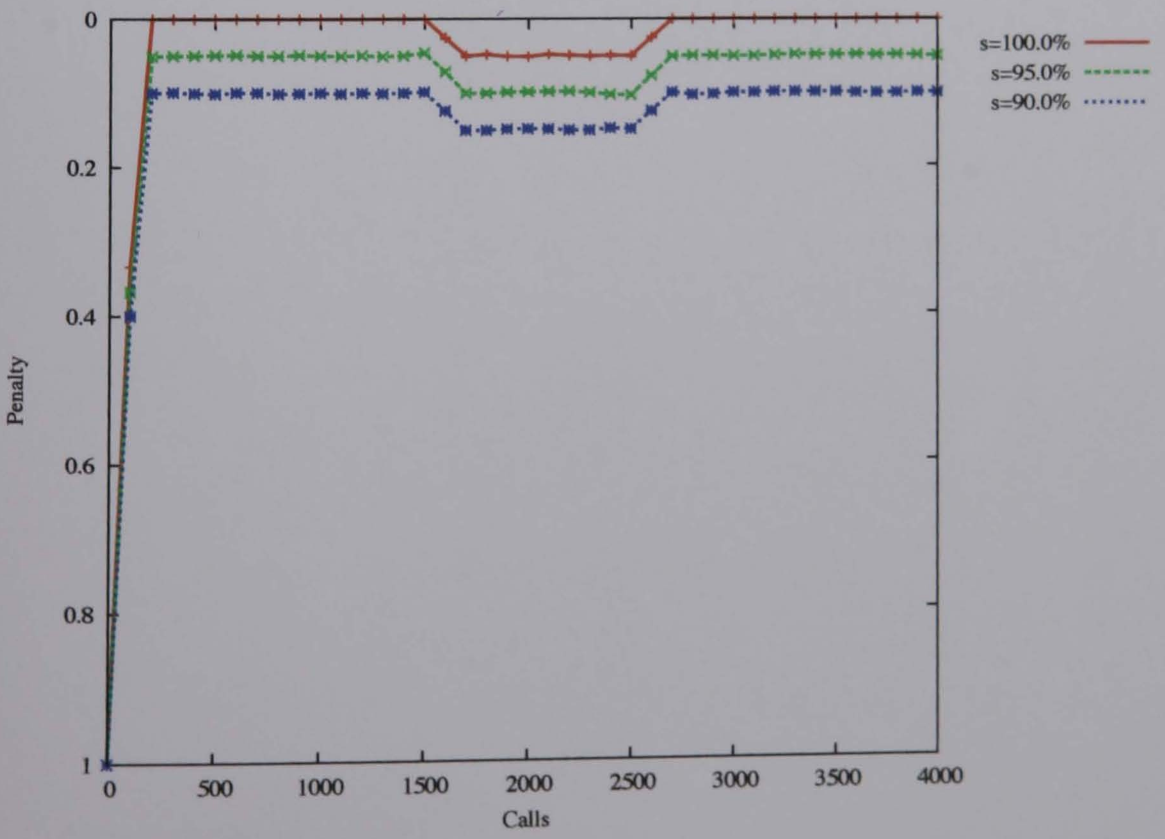


Figure 7.13: Penalty function degradation between 1500 and 2500 calls

differing commitment levels is vital to ensure that the reputation reflects contract fulfilment, not simply call success rate.

This chapter presents a new reputation function, which acts fairly with the real-world constraints of probabilistic service provision. The function rewards providers which exceed their commitment level, punishes those which fall short, and ensures that those which exactly meet their contracted commitments have a median reputation.

Comprehensive simulation results are given to demonstrate that the function operates fairly under a number of important scenarios. Reputation value is related to how well the provider meets commitments, not merely its gross success rate. Rate of improvement or decline in reputation is also related to the difference between commitment and success, but is of the same order as that of the previously-proposed system for the scenarios under test. Finally, the function is also simple to calculate, and its result is a single value, ensuring rapid update and comparison of reputations. These factors combine to enable true substitutability of services for the Digital Marketplace.

Chapter 8

Truly Anonymous Network Access

Contents

As discussed in chapter 5, three requirements must be met to enable sustainable anonymous network access: *reliability*, *unidentifiability*, and *unlinkability*. The work presented in chapters 6 and 7 creates a secure Digital Marketplace environment, which in turn fulfils the reliability requirement. The DMP as it stands provides neither unidentifiability nor unlinkability, and so anonymous network access is not immediately available.

However, this chapter aims to show that both of these requirements can be met. By altering some aspects of the Digital Marketplace operation, users can continue to take advantage of reliable and economically viable network access services, without giving away identifying or linking information about themselves.

Unidentifiability can be achieved if no other party in the DMP is able to derive a user's identity from information that the user must share to engage in the market. Unlinkability can be achieved if no actor in the DMP can observe any relationship between two subsequent calls made by one user. If these goals are met without compromising the Digital Marketplace's security and reliability, it can then be used to provide anonymous network access.

It is important to recognise at this point that some countries currently have legislature enforcing "legal interception" of communications. It is possible that such laws may hinder the adoption of the system described in this chapter. However, the legal and political issues associated with the proposed technical solution are outside the scope of this work.

8.1 Unidentifiability

The Digital Marketplace does not fundamentally change the traditional relationship between the mobile terminal user and the service provider. As described in section 6.2, the service provider is still the user's representative in the market. It therefore knows the identity of each user, and participates in what is likely to be a long-term relationship. Clearly, we cannot provide unidentifiability while this relationship exists.

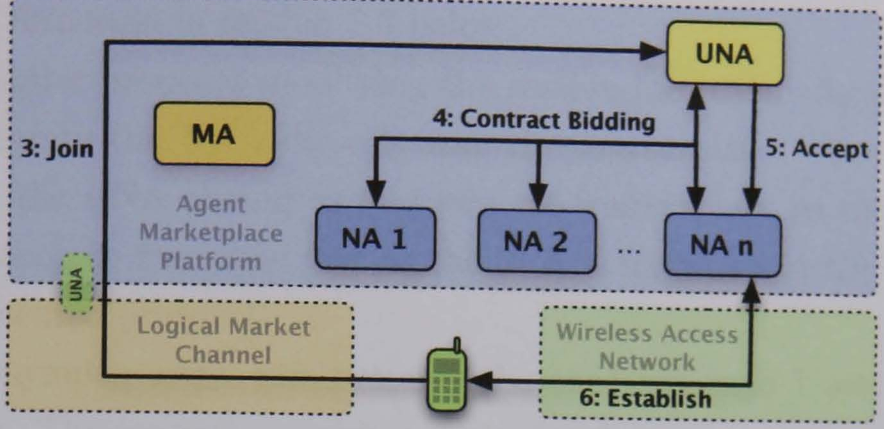


Figure 8.1: Integration of UNA into the Digital Marketplace: the UNA is migrated in the Join message (step 3 of the standard secured protocol)

8.1.1 User Negotiation Agent

However, the DMP demands that an agent is present in the marketplace to negotiate for network service. Therefore, we propose modifying the secured DMP protocol specification to permit user-submitted agents to take the role of the SPA. These User Negotiation Agents (UNAs) would have to comply with the DMP protocols, and be compatible with the DMP Agent Environment, but may otherwise operate however the user desires. With a UNA in place, the SPA is not needed, and so the anonymity-desiring user cannot be identified by the service provider.

One possibility is a standard free software UNA, which would allow the user to deploy their own agent without requiring that they create an agent from scratch. Free software[52] is a legal term, which means that the software is licensed to enable anyone to inspect and modify the source code, and distribute the results, without fee. This step would encourage technically adept users of the DMP to audit the UNA code, ensuring that it does not harm the privacy of users. Then, other users could choose to trust the consensus, or inspect the source themselves.

Making this change also creates other possibilities for the Digital Marketplace, unrelated to privacy concerns: for example, users could modify the negotiation behaviour of their agent to exactly suit their requirements. However, it is not expected to completely remove the need for service providers, for reasons discussed in section 8.3 below.

This chapter proposes modifying the session initiation stage of the protocol to migrate the UNA directly into the marketplace. As indicated in Figure 8.1, the UNA should simply join the marketplace as the first stage of the protocol, in the same way as the SPA is migrated in the previously-described secured protocol.

For anonymous users, this removes the need for steps 1 and 2, *connect* and *migrate*. The connect message was previously used to transmit the user's contract requirements to the SPA; since the negotiating agent is on the mobile terminal, this is not necessary. Likewise, the migration request from the MA is no longer needed, as the user's mobile terminal already knows that it should migrate the agent into the marketplace environment. See Figure 8.2 for the full anonymous DMP protocol flow; compare with the standard secured protocol shown in Figure 6.10 on page 142.

8.1.2 Trust and Payment

The SPA has its certificate signed by a Trusted Third Party (TTP) in the DMP; however, it is not in a position of great trust. As described in section 6.8.3.1, certification is only necessary to give network operators a course of action if a service provider fails to pay for network connectivity provided to its users. This facilitates the standard telephony model of "payment in arrears": service is used, and paid for in aggregate at the end of the billing period.

The UNA cannot be expected to present a TTP-signed certificate, as this would enable linking of multiple calls, thus defeating anonymity. Therefore, to allow a UNA to function in place of an SPA, we require that it generate a session-valid self-signed certificate. This can be used almost exactly as the SPA certificate is; to allow encryption and signature procedures. The only difference is that the association between the public key and the UNA's identifier is not certified.

This leaves the network operator with no avenue for recourse, should the UNA fail to pay for service used. We therefore require pre-payment.

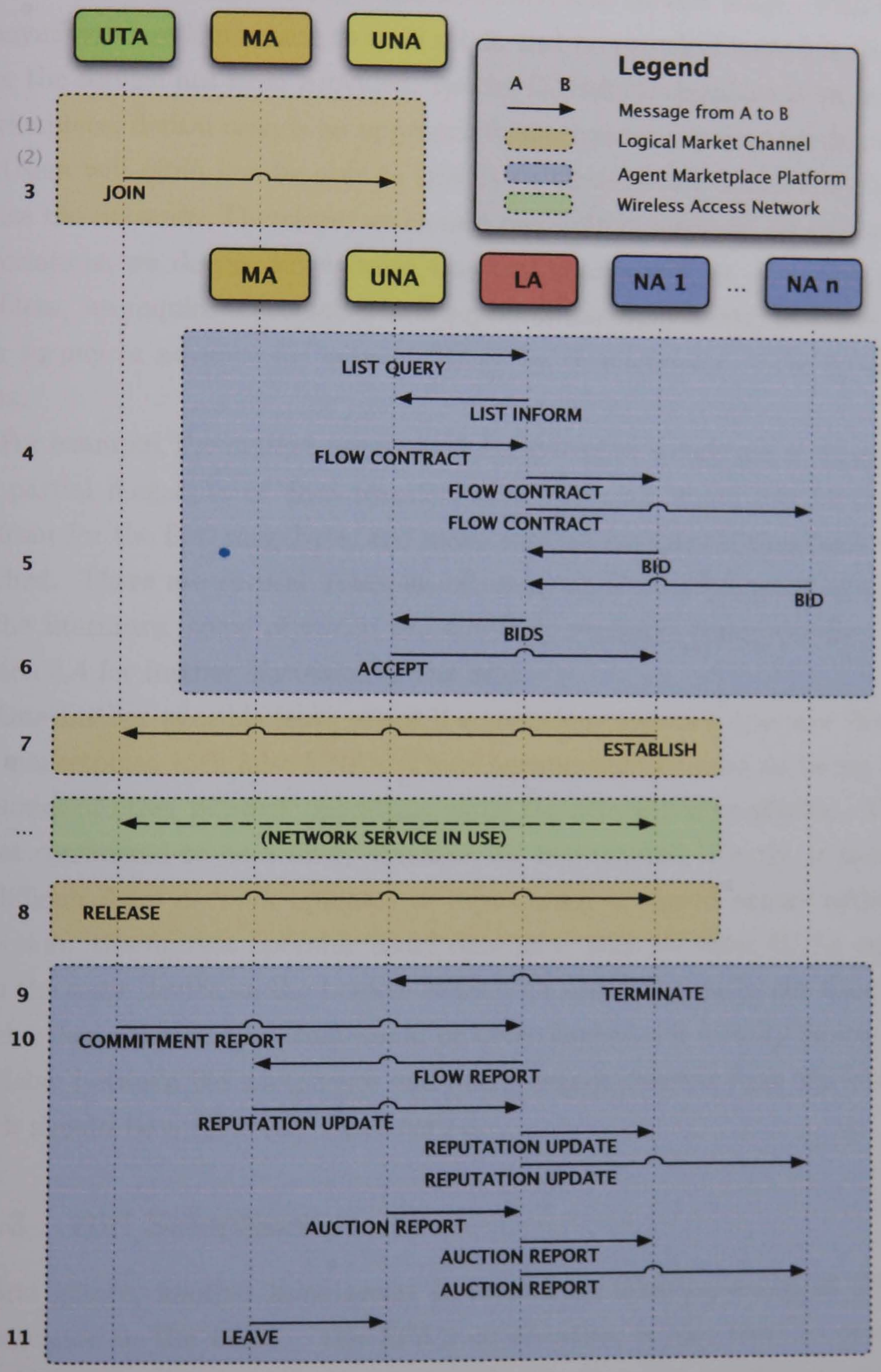


Figure 8.2: Anonymous Digital Marketplace protocol operation

This is an unavoidable consequence of anonymous service usage: with user anonymity, there is nowhere to send a bill, and no practical incentive to pay after the service has been provided. As the Digital Marketplace is an online marketplace, digital cash is an appropriate anonymous payment mechanism.

Users will often not be able to specify in advance how much they want to use the network. Therefore, we cannot pay fully in advance, as even after negotiations we do not know what the final charge will be. To solve this problem, we require a scheme which supports micropayments, to allow the user to pay in advance for network usage by throughput or time in small units.

For example, the market negotiation could lead to a contract of 50 pence per partial megabyte of data transfer; then, the user would pay 50 pence up-front for the first megabyte, and make another payment if that limit was reached. There are several examples of anonymous micropayment systems in the literature, some of which are currently available commercially: see section 3.4 for further discussion of the field.

One further possible issue arises if a malicious network operator floods the marketplace with false UNAs. These agents could be used to tie up the resources of other network operators, while the attacker is unaffected. This is not considered to be a likely scenario, for two reasons. Firstly, it should be difficult for a network operator to inject many unsigned agents without detection: the market operator could determine that all these UNAs come from the same device on the Logical Market Channel. Secondly, the time for which other network operators would be overcommitted is strictly limited to the delay between bid acceptance and non-payment timeout from the UNA, which should be a relatively short period.

8.1.3 Bid Selection

At first glance, another issue seems to arise from allowing unsigned SPAs to negotiate in the DMP. The SPA's certification is also used to ensure that detected unfair bid selection is punishable, to counter collusion. With an unsigned UNA, this cannot be punished, as the identity of the user is unknown to the Digital Marketplace judiciary.

However, this is not expected to be a major problem. Most obviously, individual users have no motive to select anything but the most competitive bid; therefore, when the unsigned UNA chooses between bids, the selection will be fair. In the expected use-case, the UNA will be acting on behalf of a privacy-conscious single user, which makes the scenario of collusion with network operators extremely unlikely.

In addition, it is likely that the free market will adapt to ensure that the less trustworthy anonymous transactions incur a premium. This is due to potentially increased risk on the part of the network operator, due to the potential for unfair bid selection by service providers using unsigned agents. Therefore, network operators will increase prices for unsigned customers, to provide an incentive for customers to use signed (and therefore traceable) agents for negotiation. The freely-operating market should balance the potential cost of anonymous users by increasing revenue to match it.

Therefore, it would not be a sensible business strategy for a service provider to present an unsigned agent to the marketplace, in order to collude with network operators without punishment. The associated increase in call cost should ensure that this approach would lead to higher costs to the user, making the service provider uncompetitive.

8.2 Unlinkability

The modifications proposed in the previous section allow users to gain network access without directly revealing identifying information. However, to fully achieve anonymity, it is necessary to also ensure that there are no detectable links between two separate Digital Marketplace interactions. Without such a property, long-term profiling is possible, which can eventually lead to identification.

To achieve unlinkability, we must stop any leaks of information which allow attackers to correlate any two sessions. For this analysis, it is necessary to exclude any application-level information disclosure; it is assumed that the user will use higher-layer anonymity schemes as described in chapter 4 to counter this problem. Therefore, only interactions during the Digital Marketplace negotiation protocol are discussed here.

8.2.1 Messaging

As shown in section 6.8.2, both the agent execution environment and Logging Agent are verifiably trustworthy, and no other agents have any direct interaction with the UNA. Therefore, it is reasonable to assume that the only adversaries to anonymity are the network operators.

From the eleven-step Digital Marketplace protocol diagram shown in Figure 8.2, it can be seen that the UTA sends three messages, and the UNA sends four. The messages sent by the UTA are *join*, *release*, and *commitment report*; the UNA sends *list query*, *flow contract*, *accept*, and *auction report*.

Of these, many reveal no information about the user. *List query* is a standard protocol message, which is not authenticated, and therefore completely identical for all users. The *release*, *commitment report*, and *auction report* messages are all authenticated; however, the user's authentication key must change for each session, removing the possibility of any link. Again, these messages all have standard, simple formats, and give no information which can be used to profile users.

This leaves the *join*, *flow contract*, and *accept* messages to be examined. Each of these may be different for each user, and therefore provide an opportunity to trace multiple accesses of the Digital Marketplace. These are dealt with in turn below.

8.2.1.1 User Negotiation Agent

The *join* message is transmitted over the Logical Market Channel, where it is clearly readable by many of the competing network operators. To migrate the agent into the marketplace environment, it clearly must contain the agent code for the UNA, which may not be identical for every user.

The secure agent execution environment ensures that the agent code is unreadable by the Network Operator Agents; therefore, the only place that the agent code could be read by the network operators is in the join message, as it is sent over the LMC. Therefore, this join message should padded to a fixed length, and encrypted so that only the secure agent execution environment may read it. Given a functional cipher, an encrypted message of fixed length should be indistinguishable from random data, and there should be no way for a third party to correlate two such messages.

This in turn implies that the UNA code must be of a limited size. However, restrictions on the agent's resource consumption are necessary anyway, to ensure reliable operation of the agent environment. Therefore, it is not unreasonable to constrain its size to some fixed number of bytes. The exact value for the agent's maximum size is an implementation-specific issue, which will depend on the agent technologies used.

8.2.1.2 Flow Contract and Bid Selection

Two other potential sources of profiling information remain in the DMP protocol. These are the *flow contract* and *accept* messages. The flow contract parameters are specified by the user, and may be different for each call. Similarly, the bid selection process results in a choice being made by the UNA, which means that the accept message has some information value to the marketplace participants.

However, since the bid selection behaviour is opaque, and the algorithm used may be complex, it is likely to be extremely difficult to use this data to profile a single UNA. Additionally, the use of a standard UNA, without modification, would make it impossible to distinguish between all anonymous users. It is therefore assumed that it is unfeasible to link two sessions to one unique UNA from bid choice alone.

Similarly, parameters in the flow contract are likely to be similar for all agents. Again, the standard UNA could be used to provide a limited set of flow contract terms, with several profiles for different expected usage requirements. Along with random variation depending on network conditions, this makes all users practically indistinguishable by network operators.

8.2.2 Hardware Addresses

One other possibility for linking several sessions exists. It is assumed that all local wireless networks used by the Digital Marketplace require hardware addresses for clients; for example, the MAC address in 802.11. This is observable by anyone accessing the network, and is normally never changed. It therefore provides a very simple method for tracking users over multiple sessions.

Previous work by Tortonesi and Davoli discussed this problem, and proposed several solutions[82]. They suggest using cryptographic techniques to create dynamic hardware addresses. One proposal requires a loosely-synchronised clock to generate a guaranteed-unique local hardware address. At time of connection to the network, the algorithm encrypts the combination of a local network prefix, the true hardware address, and the current time. The output remains IEEE 802 compliant, and would therefore work with any 802-compatible network interface.

Other access protocols would require similar dynamic hardware address techniques to provide unlinkability. For example, the same method could be applied to GSM or UMTS devices, to replace the unique International Mobile Equipment Identity (IMEI) number. This would again need to be changed at the beginning of every session to achieve unlinkability.

However, this is currently illegal in some countries without the manufacturer's consent; for example, in the United Kingdom, due to the Mobile Telephones (Re-programming) Act 2002[26]. It is noted that this legal restriction is commonly intended to deter theft of handsets; because IMEIs are assumed to be unchangeable, operators can blacklist handsets which are reported as stolen.

A manufacturer of mobile handsets must agree to allow its users to change the network-visible IMEI, if the devices are to be used anonymously. Handsets will require significant software modifications to support the Digital Marketplace, and possibly specific changes to support anonymity. Therefore, manufacturer consent to change the IMEI is a reasonable assumption, although the trade-off between anonymity and theft deterrence is noted.

8.3 Analysis

The two remaining requirements for anonymous network access are met by making further modifications to the secure Digital Marketplace described in chapter 6. Unidentifiability is achieved by the use of a user-provided negotiating agent, and anonymous digital cash for payment. Unlinkability is maintained so long as the agent used acts similarly to others, and the network hardware address can be changed for each session.

8.3.1 Service Provider Role

It is important to note that the use of the UNA will not fundamentally change the Digital Marketplace operation for users who are unconcerned with being identified. Service providers will continue to have a useful role to play in negotiating on behalf of users, and it is not expected that all users of the DMP will choose to be anonymous.

As discussed in section 8.1.3, there is likely to be an increase in cost for anonymous users, to balance the potential for unfair bid selection attacks against the network operators. This increase alone is likely to give an incentive for users to choose to use a service provider to interface with the marketplace.

In addition, other current benefits of service providers can continue within the DMP. For example, fixed-cost contracts with usage allowances and discounted services are made available on the market today. A service provider could continue to offer these contracts to users, who may be happy to budget for a fixed cost each month.

8.3.2 Protocol Changes

The introduction of a UNA requires an optional different path through the protocol. Instead of sending a connect message to the MA, the user skips the first two steps of the protocol and immediately migrates its agent into the marketplace. However, instead of being transmitted over a fixed network, this must be done over the Logical Market Channel, which has restricted throughput.

This is not expected to be a major issue, as the agent migration message must be limited to a fixed size to ensure that it is not recognisable in transit by network operators. While the size of this message is dependent on the agent technology used, the UNA is not a particularly complex piece of software.

If the secure agent execution environment provides FIPA language and protocol support, the agent code transferred could simply represent the specific flow contract terms and bid selection behaviour of the user. The remainder of the agent operation could be supplied by the execution environment, which is remotely-verifiable as secure.

The flow contract consists of only six parameters, each of which could easily be represented in a 32-bit number (see section 5.3.4.2). Therefore, the flow contract could theoretically be represented in around $32 * 6 = 192$ bits, or 24 bytes. The bid selection algorithm is more dependent on the language used, but could most likely be written in a few hundred bytes. Therefore, the transfer of this agent is unlikely to use a significant proportion of the LMC's available bandwidth.

8.4 Summary and Conclusions

The Digital Marketplace is a free-market based call management architecture, which aims to decentralise network service provision to improve market efficiency. As a side-effect of this, the possibility of a new form of communications privacy is opened.

Truly anonymous network access is possible if no external party can determine the identity of a network user. As described in chapter 5, this is only feasible with reliable network service. This is provided by the secured Digital Marketplace, but the secured protocol does not directly support anonymity.

Necessary changes to the protocol to provide anonymity are described and justified above in section 8.1 and section 8.2. They are summarised below.

8.4.1 User Negotiation Agent

The Digital Marketplace must allow a user to self-represent in the marketplace negotiations to provide unidentifiability. This is achieved by permitting a User Negotiation Agent (UNA) to take the place of the SPA in the DMP protocols. This agent is controlled directly by the user, who therefore acts without the use of a separate service provider.

8.4.2 Migration

A user's UNA is migrated into the marketplace at the session initiation stage of the protocol. Steps 1 and 2 are elided for anonymous users; step 3 migrates the agent directly over the Logical Market Channel. The message which performs the migration must be padded to a fixed length, and encrypted to ensure that adversaries cannot determine the operation of the agent.

8.4.3 Payment

The standard Digital Marketplace model expects that that the Service Provider Agent will arrange network service, and pay later in aggregate. This is obviously not possible with anonymous users, who must therefore pay for services beforehand. This can be achieved with an anonymous digital cash scheme. The scheme must support micropayments, so that users can pay for service incrementally as it is used.

8.4.4 Certification

Unlike the SPA, the UNA cannot be certified, as this would lead to identification of the user. Instead, a session-valid asymmetric key pair must be generated by the user's device, to allow secure communications between the UNA and other agents. Lack of certification does not directly affect the protocol, due to pre-payment as described above.

8.4.5 Hardware Addresses

To ensure that network operators cannot trace users over multiple sessions. one final change must be made. The unique hardware address of each device must be modified so that it exists only for a single session. Previous work has described schemes to enable this for 802 MAC protocols. such as WiFi; this could readily be applied to cellular technology.

8.4.6 Conclusions

Several properties arise from the changes described above. First, the user no longer requires a long-term relationship with a service provider in order to gain network access. Therefore, there is no requirement for the user to reveal their identity to anyone in order to gain network access. Finally, two separate network access sessions cannot be linked together to build up a profile of any user. Therefore, this modified Digital Marketplace enables a new form of anonymous communication: truly anonymous network access.

Chapter 9

Conclusions and Further Work

Contents

9.1 Conclusions

Mobile communications technology is now a part of many people's everyday lives. The increasingly ubiquitous availability of network access and the breadth of services on offer lead to usage levels which are higher than ever before. Along with these increases come privacy issues: because more data is being transmitted from personal mobile devices, more information can be revealed about the people using them.

Privacy is a worrying issue for many people using modern communications networks. Some recent studies deal with Internet commerce, where shoppers have been shown to be very cautious about revealing personal information to retailers. More general research has shown privacy to be a concern rated very highly in a list of social issues: even more important than equal rights, the environment, and unemployment, among one set of respondents.

One method of enhancing privacy is to remove identifying information from communications. This allows anonymous usage of services and networks, which can reduce the impact of any privacy-infringing participants. Anonymity is the core issue of this thesis, and two approaches to anonymous communications are inspected: service anonymity, and anonymous network access.

Anonymity at the level of service provision is a field of research which has received much interest over the past two decades. Selected important work from the field of electronic communications anonymity is presented, discussed, and analysed in the context of mobile communications. This thesis examines three key anonymity schemes: MIX networks, Onion Routing, and Crowds. While shown to be useful for fixed networks, these schemes are not directly applicable to current mobile networks. Therefore, the thesis presents a series of conclusions and recommendations for achieving anonymity at the service level using wireless access networking. These recommendations are exemplified with a case study into anonymous location-based services, in which a privacy-enhancing protocol is contributed.

Network service providers are in an excellent position to gather private information from their users' communications, even when service-level anonymity schemes are in place. Some privacy-conscious users may not trust their service provider with this information, and would prefer to gain access to communications networks without identifying themselves. This is only possible with a viable anonymous network access scheme.

This thesis approaches this problem by identifying a series of requirements for sustainable anonymous network access, recognising that the current best effort comes from free open WiFi networks. However, such networks are not reliable, due to the lack of a standardised, widely-deployed, and secure payment system.

A previously-published call management architecture, the Digital Marketplace (DMP), has the potential to meet the requirements of anonymous network access. Its basis is a freely-operating market in network access, which could potentially allow a user to agree new network service contracts for each session. This could remove the long-term privacy threat of the network service provider. However, the published form of the DMP did not quite achieve these goals, and correcting this was a major aim of this thesis.

First, and most importantly, the Digital Marketplace proposals so far had not been subject to a thorough security analysis. This thesis contributes significant work in this area, beginning with a comprehensive threat analysis of the DMP protocols. There follows a series of security requirements, which must be met to ensure that the marketplace can operate securely, reliably, and fairly. A matching set of security measures is also contributed, in order to fulfil the requirements, and a final analysis demonstrates that the security threats are countered by these changes.

One final requirement for a reliable Digital Marketplace design is an appropriate reputation system. A key issue in a freely-operating market is substitutability of goods or services: different competitors' offerings must be comparable, in order for true competition to exist. The DMP attempts to marry this requirement with probabilistic service provision, which is an inherent property of mobile wireless networks; a service provider can only estimate the probability of providing the service required. A reputation system must reflect this uncertainty of service accurately, to ensure that offers from competing network providers can be fairly compared.

This thesis contributes a novel reputation function, which meets this requirement. Its operation is described, and an implementation is simulated for various important scenarios. Results are presented and compared with a previous Digital Marketplace reputation function. This demonstrates that the proposed reputation function out-performs the previous function, while fairly reflecting market operator behaviour as required.

Finally, with the protocol secured, and a functional reputation system in place, using the Digital Marketplace to achieve anonymous network access is discussed. The secured protocol as it stands does not allow anonymity, and so modifications must be made. This thesis proposes the addition of a user-controlled agent into the negotiation process, which removes the need for a representative service provider. Users are expected to pay with digital cash, using micropayments to enable variable length of service provision. Finally, a need for randomised hardware addresses is shown, and a previous scheme to achieve this is noted.

With these measures in place, contributed analysis indicates that anonymous Digital Marketplace users are able to securely obtain network service, leaving other users in the market unaffected. Identity of these users is not revealed at any point, and no information is available to potential attackers to allow profiling over multiple sessions. Therefore, this scheme enables truly anonymous network access.

These contributions in two areas of anonymity in wireless access networks demonstrate that private communication is possible, even in mobile systems. Service-level anonymity techniques can be adapted to mobile systems, allowing current users to access services without revealing personal information. Additionally, future deployment of the Digital Marketplace will grant users fully ground-up private communications. When combined, the privacy enhancements offered by these two approaches are significant for all privacy-aware mobile communications users.

9.2 Future Work

There are several directions in which future Digital Marketplace research can be taken. The contributions in this thesis have enhanced the security and reliability of the market, and argued that it can now function fairly and efficiently. This could be confirmed by in-depth simulation of a representative free market, taking into account the new reputation system and anonymous usage.

Another option would be to create reference implementations of the Digital Marketplace agents, which could then be used in experimental investigations of market operation. Many of these research opportunities are related to efficient operation, and therefore demand the availability of real measurements of the protocol overheads. An implementation of the Digital

Marketplace would offer such measurements and enable this research.

Finally, further research into the security properties of the Digital Marketplace would be of worth. The threat analysis and protocol modifications presented in this thesis are believed to present a secure Digital Marketplace, but secondary validation is always helpful.

9.2.1 Market Simulation

To provide valid results, free market simulations can require an especially careful choice of assumptions, due to the inherent complexity of the system under test. This appears to apply to the Digital Marketplace as well. Particularly important are the design decisions about how to model local and backbone traffic, cost and pricing, and negotiation strategies. All of these parameters can have dramatic effects on how the market performs.

Despite these difficulties, a simulator of this short-term communications contract market would be of benefit to the area. This thesis argues that anonymous Digital Marketplace users would be expected to be subject to some price rise, relative to identified users. This could be verified by simulation, and the average percentage increase in price quantified.

9.2.2 Reference Implementation

As yet, there have been no full implementations of an agent-based Digital Marketplace. The real overheads due to agent technologies can only be fully understood by taking this step, but it requires a significant effort. There are again a great number of open questions to be answered here: for example, which execution environment is most suitable; which agent platform should be used; and how network operator bids should be calculated.

Therefore, there is a great deal of research to be done into how to most efficiently implement a Digital Marketplace environment. However, the real value of a reference implementation is that it opens numerous paths for future research.

9.2.3 Security versus Efficiency

One research direction particularly relevant to this thesis is in the trade-off between security and efficiency. The overheads of the secured protocol can be altered, by changing the parameters of the cryptographic techniques used. Increasing security is likely to lead to increased overheads, both in time taken to create messages, and in the size of the resultant data. To what degree this affects market operation can only be truly quantified once an implementation is created and optimised, so that exact measurements can be made.

Along similar lines, the measured performance of the reference implementation could direct researchers into improving the efficiency of the protocol. If measurements indicate that the negotiation stage completes quickly, then there is no immediate need to alter the protocol design. On the other hand, if the overheads are great in a particular area, this could lead to an improved protocol which counters such problems.

9.2.4 Impact of Digital Cash

Another pertinent example arises from the anonymous Digital Marketplace work presented in this thesis. The practical impact of using Digital Cash as a prepayment mechanism could be measured in a real implementation. Some delay is likely to be introduced by the cash generation and transfer protocol; the difference between the anonymous protocol and its identified counterpart could be quantified.

Different Digital Cash schemes offer varying balances between the overheads of cash generation and funds transfer. With a full agent-based implementation of the market, and an appropriate simulation of the mobile terminal, the measured delay due to each of these areas could be compared for several schemes. This would allow a more optimal trade-off to be taken between the security offered by the cash schemes and the impact on negotiations.

9.2.5 Further Security Work

The method used for the security analysis of the Digital Marketplace in this thesis is somewhat informal. As a first approach to a complex security problem, this is believed to be the best choice. However, in future, more formal methods could be employed to verify the findings of this work.

It is expected by the author to be unfeasible to formally prove the security of a system as complex as the Digital Marketplace. One recommended approach would be to look at the protocol in stages; for example, a verification of the security of the bidding process would be valuable even in isolation from the remainder of the protocol.

Bibliography

[1] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Implementing P3P using database technology. In *Proceedings of the 19th International Conference on Data Engineering*, pages 595–606, March 2003. (cited on page 9)

[2] Alisdair McDiarmid and James Irvine. Securing the Digital Marketplace. *Proceedings of 62nd IEEE Vehicular Technology Conference*, September 2005. (cited on pages 139, 157)

[3] 3G Americas. Two Billion GSM Customers Worldwide. Press Release, `http://www.3gamericas.org/`, June 2006. (cited on page 1)

[4] J. Arkko and H. Haverinen. Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). Internet Engineering Task Force: RFC 4187, January 2006. (cited on page 127)

[5] N. Asokan. Anonymity in mobile computing environment. In *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, 1994. (cited on page 56)

[6] Austin Wireless, Inc. Austin Wireless City Project. `http://www.austinwirelesscity.org/`, June 2006. (cited on page 80)

[7] Christoph Bäumer, Markus Breugst, Sang Choy, and Thomas Magedanz. Grasshopper–a universal agent platform based on OMG MASIF and FIPA standards. In *Proceedings of the First International Workshop on Mobile Agents for Telecommunication Applications (MATA'99)*, pages 1–18, October 1999. (cited on page 125)

[8] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying Hash Functions for Message Authentication. In *Advances in Cryptology - CRYPTO '96 Proceedings*, Lecture Notes in Computer Science, pages 1–15. Springer-Verlag, 1996. (cited on page 27)

[9] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000. (cited on page 26)

[10] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures—How to Sign with RSA and Rabin. In *Advances in Cryptology — Eurocrypt 96 Proceedings*, volume 1070 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996. (cited on pages 29, 126)

[11] Alastair R. Beresford and Frank Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, January–March 2003. (cited on page 65)

[12] Daniel J. Bernstein. The Poly1305-AES Message-Authentication Code. In *Proceedings of Fast Software Encryption: 12th International Workshop*, Lecture Notes in Computer Science, page 32. Springer-Verlag, February 2005. (cited on page 28)

[13] Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The disadvantages of free MIX routes and how to overcome them. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 30–45. Springer-Verlag, LNCS 2009, July 2000. (cited on page 44)

[14] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: Fast and secure message authentication. In *CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pages 216–233, London, UK, 1999. Springer-Verlag. (cited on page 27)

[15] John Black and Phillip Rogaway. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. *Journal of Cryptology*, 18(2):111–131, 2005. (cited on page 26)

[16] Dan Boneh, Antoine Joux, and Phong Q. Nguyen. Why textbook ElGamal and RSA encryption are insecure. *Lecture Notes in Computer Science*, 1976, 2000. (cited on page 22)

[17] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-record communication, or, why not to use pgp. In *WPES '04: Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 77–84, New York, NY, USA, 2004. ACM Press. (cited on page 24)

[18] William Burr. Cryptographic hash standards. *IEEE Security & Privacy*, 4(2):88–91, March/April 2006. (cited on page 26)

[19] BWCS. Making the Most of Legacy Mobiles: Examining the Relationship Between Technology Choices and Location Revenues. *http://www.bwcs.com/whitepapers/Legacy_Mobiles.pdf*, January 2004. (cited on pages 64, 67)

[20] Florent Chabaud and Antoine Joux. Differential Collisions in SHA-0. In *Advances in Cryptology – CRYPTO '98 Proceedings*, pages 56–71, London, UK, 1998. Springer-Verlag. (cited on page 25)

[21] David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981. (cited on page 35)

[22] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology – Proceedings of CRYPTO '82*, pages 199–203. Springer-Verlag, 1983. (cited on page 31)

[23] David Chaum. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985. (cited on page 32)

[24] David Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Advances in Cryptology – CRYPTO '88 Proceedings*, Lecture Notes in Computer Science, 1988. (cited on page 32)

[25] Jeffrey Cole. UCLA Internet Report 2001: Surveying the Digital Future. `http://www.ccp.ucla.edu/`, 2001. (cited on page 7)

[26] Crown Copyright. Mobile Telephones (Re-programming) Act 2002. http://www.opsi.gov.uk/ACTS/acts2002/20020031.htm, July 2002. (cited on page 183)

[27] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. http://www.w3.org/TR/P3P/, April 2002. (cited on page 8)

[28] David H. Crocker. RFC 822: Standard for the format of ARPA Internet text messages. Internet Engineering Task Force: RFC 822, August 1982. (cited on page 44)

[29] Mary Culnan and George Milne. The Culnan-Milne Survey on Consumers & Online Privacy Notices: Summary of Responses. In *Interagency Public Workshop: Get Noticed: Effective Financial Privacy Notices*, 2001. (cited on page 8)

[30] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003. (cited on pages 42, 45)

[31] Daniel Nagy. On digital cash-like payment systems. In *Proceedings of the International Conference on e-Business and Telecommunications 2006*, August 2006. (cited on page 33)

[32] David Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *Journal of Cryptology*, 1(1):65–75, 1988. (cited on page 12)

[33] Chrysanthos Dellarocas. Analyzing the economic efficiency of eBay-like online reputation reporting mechanisms. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 171–179, New York, NY, USA, 2001. ACM Press. (cited on page 157)

[34] Privoxy Developers. Privoxy. http://www.privoxy.org/, July 2006. (cited on page 49)

[35] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. Internet Engineering Task Force: RFC 4346, April 2006. (cited on pages 31, 45)

[36] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976. (cited on page 20)

[37] Roger Dingledine and Nick Mathewson. Tor protocol specification. `http://tor.eff.org/cvs/tor/doc/tor-spec.txt`, June 2006. (cited on page 49)

[38] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004. (cited on pages 48, 50)

[39] Goran M. Djuknic and Robert E. Richton. Geolocation and Assisted GPS. *IEEE Computer*, February 2001. (cited on page 64)

[40] World Wide War Drive. World Wide War Drive 4 Statistics. `https://wigle.net/gps/gps/GPSDB/stats/?eventid=1`, June 2006. (cited on page 80)

[41] Morris Dworkin. Recommendation for block cipher modes of operation: Methods and techniques. NIST Special Publication 800-38A, December 2001. (cited on page 18)

[42] Taher Elgamal. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. In *Advances in Cryptology – CRYPTO '84 Proceedings*, pages 10–18, 1984. (cited on page 21)

[43] Taher Elgamal. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985. (cited on page 21)

[44] Federal Communications Commission. Enhanced 911 - Wireless Services. `http://www.fcc.gov/911/enhanced/`, June 2006. (cited on page 63)

[45] Federal Trade Commission. Pretexting: Your Personal Information Revealed, February 2006. (cited on page 78)

[46] Hannes Federrath, Anja Jerichow, and Andreas Pfitzmann. MIXes in mobile communication systems: Location management with privacy. In *Information Hiding*, pages 121–135, 1996. (cited on page 56)

[47] Hal Finney. RPOW: Reusable Proofs of Work. In *CodeCon 2005*, February 2005. (cited on page 126)

[48] Chuan Heng Foh and Moshe Zukerman. Performance Analysis of the IEEE 802.11 MAC Protocol. In *Proceedings of the European Wireless 2002 Conference*, pages 184–190, February 2002. (cited on page 81)

[49] Foundation for Intelligent Physical Agents. FIPA Subscribe Interaction Protocol Specification, March 2002. Document number SC00035H. (cited on page 137)

[50] Foundation for Intelligent Physical Agents. FIPA Agent Management Specification, March 2004. Document number SC00023K. (cited on page 143)

[51] Foundation for Intelligent Physical Agents. FIPA Specifications. `http://www.fipa.org/specifications/`, June 2006. (cited on page 99)

[52] Free Software Foundation. The Free Software Definition. `http://www.fsf.org/licensing/essays/free-sw.html`, June 2006. (cited on page 176)

[53] I Getting. The Global Positioning System. *IEEE Spectrum*, December 1993. (cited on page 64)

[54] Ian Goldberg, David Wagner, and Eric Brewer. Privacy-enhancing Technologies for the Internet. In *Proceedings of 42nd IEEE Spring COMPCON*. IEEE Computer Society Press, February 1997. (cited on page 44)

[55] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding Routing Information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996. (cited on page 46)

[56] Graphic, Visualization, & Usability Center, Georgia Tech. GVU 10th WWW User Survey. `http://www.gvu.gatech.edu/user_surveys/survey-1998-10/`, 1998. (cited on page 8)

[57] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of MobiSys 2003: The First International Conference on Mobile Systems, Applications, and Services*, May 2003. (cited on page 65)

[58] Carl Gunter, Michael May, and Stuart Stubblebine. A Formal Privacy System and Its Application to Location Based Services. In *Privacy Enhancing Technologies: 4th International Workshop*, May 2004. (cited on page 9)

[59] John Heidemann, Katia Obraczka, and Joe Touch. Modeling the performance of HTTP over several transport protocols. *IEEE/ACM Transactions on Networking*, 5(5):616–630, 1997. (cited on page 57)

[60] Sabine Helmers. A brief history of anon.penet.fi - the legendary anonymous remailer. *CMC Magazine*, September 1997. (cited on page 44)

[61] Chris Hurley. The worldwide wardrive: The myths, the misconceptions, the truth, the future. In *Proceedings of Defcon 11*, August 2003. (cited on page 80)

[62] Information Commissioner's Office. Annual Track Research Findings. `http://www.informationcommissioner.gov.uk/`, 2004. (cited on page 7)

[63] James Irvine. Adam Smith Goes Mobile: Managing Services Beyond 3G with the Digital Marketplace. *Invited Paper to European Wireless 2002*, February 2002. (cited on pages viii, 104, 105, 124)

[64] Information Technology—Security Techniques—Encryption algorithms—Part 2: Asymmetric Ciphers (Draft). ISO/IEC 18033-2, January 2004. (cited on pages 23, 126, 132)

[65] Information Technlogy—Technology—Open Systems Interconnection—The Directory: Public-Key and Attribute Certificate Frameworks. ISO/IEC 9594-8, May 2001. (cited on page 31)

[66] Information Technlogy—Security Techniques—Message Authentication Codes (MACs)—Part 2: Mechanisms Using A Dedicated Hash Function, June 2002. (cited on page 72)

[67] David Kahn. *The Codebreakers*. Scribner, 1996. (cited on page 16)

[68] Auguste Kerckhoffs. La Cryptographic Militaire. *Journal des Sciences Militaire*, pages 161–191, Feburary 1883. (cited on page 16)

[69] Jennifer C. Kerr. Lawmakers to target sale of phone records. The Guardian, Monday 23 January, 2006. (cited on page 78)

[70] Steve Kremer, Olivier Markowitch, and Jianying Zhou. An intensive survey of fair non-repudiation protocols. *Computer Communications*, 25(17):1606–1621, November 2002. (cited on page 28)

[71] Jonathan Krim. Online data gets personal: Cell phone records for sale. Washington Post, Friday 8 July, 2005. (cited on page 78)

[72] B. A. LaMacchia and A. M. Odlyzko. Computation of discrete logarithms in prime fields. *Designs, Codes, and Cryptography*, 1(1):47–62, 1991. (cited on page 22)

[73] LAN/MAN Standards Committee of the IEEE Computer Society. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999. (cited on page 80)

[74] Oscar Lazaro, James Irvine, Demessie Girma, and John Dunlop. Managing Security within Heterogeneous Wireless Communication Systems. In *Proceedings of Communication Systems and Networks 2003*, September 2003. (cited on page 125)

[75] Gwenaël Le Bodic. A Multiagent System for Application of Market Concepts to Emerging Mobile Communication Services, 2000. Thesis for the degree of PhD. (cited on page 107)

[76] Gwenaël Le Bodic, Demessie Girma, James Irvine, and John Dunlop. An Agent-Based Middleware for Enhancing Mobile Communications Infrastructure and Provision of Services in Emerging Systems. In *Proceedings of SOMAS Workshop*, July 2000. (cited on page 111)

[77] Gwenaël Le Bodic, Demessie Girma, James Irvine, and John Dunlop. Dynamic 3G Network Selection for Increasing the Competition in the Mobile

Communications Market. In *Proceedings of 52nd Vehicular Technology Conference*, pages 1064–1071, September 2000. (cited on pages 86, 111, 158, 161)

[78] Alex Leary. Wi-Fi cloaks a new breed of intruder. St. Petersburg Times, http://www.sptimes.com/2005/07/04/State/Wi_Fi_cloaks_a_new_br.shtml, July 2005. (cited on page 80)

[79] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS Protocol Version 5. Internet Engineering Task Force: RFC 1928, March 1996. (cited on page 48)

[80] John Leyden. UK war driver fined £500. The Register, http://www.theregister.co.uk/2005/07/25/uk_war_driver_fined/, July 2005. (cited on page 80)

[81] Allison Mankin, Randall Gellens, and Andrew Newton. Geographic Location/Privacy (geopriv) Charter. *http://www.ietf.org/html.charters/geopriv-charter.html*, June 2006. (cited on page 64)

[82] Mauro Tortonesi and Renzo Davoli. User untraceability in the next-generation Internet: a proposal. In *Proceedings of Communication and Computer Networks*, November 2002. (cited on page 183)

[83] Bodo Möller. Provably Secure Public-Key Encryption for Length-Preserving Chaumian Mixes. In *Proceedings of CT-RSA 2003*. Springer-Verlag, LNCS 2612, April 2003. (cited on page 42)

[84] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. http://www.abditum.com/mixmaster-spec.txt, July 2003. (cited on page 45)

[85] National Institute of Standards and Technology. Data Encryption Standard. *FIPS pub. 46*, January 1977. (cited on page 17)

[86] National Institute of Standards and Technology. Computer Data Authentication, 1985. (cited on page 26)

[87] National Institute of Standards and Technology. Secure Hash Standard. *FIPS pub. 180*, May 1993. (cited on page 25)

[88] National Institute of Standards and Technology. Secure Hash Standard. *FIPS pub. 180-1*, April 1995. (cited on page 25)

[89] National Institute of Standards and Technology. Digital Signature Standard. *FIPS pub. 186-2*, January 2000. (cited on page 29)

[90] National Institute of Standards and Technology. Advanced Encryption Standard. *FIPS pub. 197*, November 2001. (cited on page 17)

[91] National Institute of Standards and Technology. Secure Hash Standard. *FIPS pub. 180-2*, August 2002. (cited on page 25)

[92] O2. Data tariffs. `http://www.o2.co.uk/business/corporate/businesstariffs/datatariffs/`, June 2006. (cited on page 87)

[93] Ofcom. The Telecoms Industry. In *The Communications Market: Interim Report*, page 75, February 2006. (cited on page 86)

[94] Ofcom. The Telecoms User. In *The Communications Market: Interim Report*, page 62, February 2006. (cited on pages 7, 86)

[95] Orange. mobile data: Business Everywhere. `http://www.business.orange.co.uk/`, June 2006. (cited on page 87)

[96] Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient Anonymous Channel and All/Nothing Election Scheme. In *EUROCRYPT '93: Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, pages 248–259, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc. (cited on page 43)

[97] Paul Resnick and Richard Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. *Volume 11 of Advances in Applied Microeconomics*, 2002. (cited on page 157)

[98] Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. The Value of Reputation on eBay: A Controlled Experiment. *Working paper originally presented at the ESA conference*, June 2002. (cited on page 157)

[99] Andreas Pfitzmann and Marit Kohntopp. Anonymity, Unobvservability and Pseudonymity — A Proposal for Terminology. In *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability*, pages 1–9, July 2000. (cited on pages 3, 10)

[100] Birgit Pfitzmann and Andreas Pfitzmann. How to break the direct RSA-implementation of MIXes. In *Proceedings of EUROCRYPT 1989*. Springer-Verlag, LNCS 434, 1990. (cited on page 42)

[101] Jonathan B. Postel. RFC 821: Simple Mail Transfer Protocol. Internet Engineering Task Force: RFC 821, August 1982. (cited on page 45)

[102] M. Reed, P. Syverson, and D. Goldschlag. Protocols using Anonymous Connections: Mobile Applications. In *1997 Security Protocols Workshop*, April 1997. (cited on page 56)

[103] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection*, 1998. (cited on page 46)

[104] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998. (cited on pages 10, 12, 51, 54)

[105] Ronald L. Rivest and Adi Shamir. PayWord and MicroMint – Two Simple Micropayment Schemes. *CryptoBytes*, 2(1), 1996. (cited on page 33)

[106] Ronald L. Rivest, Adi Shamir, and Leonard M. Adelman. A Method For Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978. (cited on page 21)

[107] RSA Data Security, Inc. PKCS #5: Password-Based Cryptography Standard, March 1999. Version 2.0. (cited on page 18)

[108] RSA Laboratories. PKCS #1: RSA Encryption Standard, June 2002. Version 2.1. (cited on pages 23, 29)

[109] Chris Salter, O. Sami Saydjari, Bruce Schneier, and Jim Wallner. Towards a Secure System Engineering Methodology. In *Proceedings of the 1998 Workshop on New Security Paradigms*, pages 2–10, September 1998. (cited on page 101)

[110] Didier Samfat, Refik Molva, and N. Asokan. Untraceability in mobile networks. In *Mobile Computing and Networking*, pages 26–36, 1995. (cited on page 56)

[111] Rafael Sanchez, Julia Martinez, Javier Romero, and Rauli Jarvela. TCP/IP performance over EGPRS network. In *Proceedings of the 56th Vehicular Technology Conference*, pages 1120–1124, September 2002. (cited on page 57)

[112] H. Schulzrinne, H. Tschofenig, J. Morris, J. Cuellar, and J. Polk. A Document Format for Expressing Privacy Preferences for Location Information. `http://www.ietf.org/internet-drafts/draft-ietf-geopriv-policy-08.txt`, February 2006. (cited on page 64)

[113] Adi Shamir and Eran Tromer. Factoring Large Numbers with the TWIRL Device. In *Advances in Cryptology – CRYPTO 2003 Proceedings*, Lecture Notes in Computer Science, 2003. (cited on page 21)

[114] Vitaly Shmatikov. Probabilistic model checking of an anonymity system. *Journal of Computer Security*, 12(3-4):355–377, 2004. (cited on page 55)

[115] Victor Shoup. A Proposal for an ISO Standard for Public Key Encryption (version 2.1). *http://www.shoup.net/papers/*, December 2001. (cited on page 72)

[116] Siani Pearson, editor. *Trusted Computing Platforms: TCPA Technology in Context.* Prentice Hall, September 2002. (cited on page 125)

[117] Adam Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations.* Penguin, 1776. Republished 1982. (cited on page 87)

[118] Sean W. Smith. Outbound Authentication for Programmable Secure Coprocessors. In *Proceedings of 7th European Symposium on Research in Computer Security*, pages 72–89, 2002. (cited on page 125)

[119] Tim Stack, Eric Eide, and Jay Lepreau. Bees: A Secure, Resource-Controlled, Java-Based Execution Environment. In *Proceedings of the 2003 IEEE Conference on Open Architectures and Network Programming (OPENARCH 2003)*, pages 97–106, April 2003. (cited on page 125)

[120] National Statistics. *UK 2005: The Official Yearbook of the United Kingdom of Great Britain and Northern Ireland.* HMSO, 2006. (cited on page 86)

[121] Niranjan Suri, Jeffrey M. Bradshaw, Maggie R. Breedy, Paul T. Groth, Gregory A. Hill, Renia Jeffers, Timothy S. Mitrovich, Brian R. Pouliot, and David S. Smith. Nomads: toward a strong and safe mobile agent system. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, pages 163–164, New York, NY, USA, 2000. ACM Press. (cited on page 125)

[122] Latanya Sweeney. k-anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002. (cited on pages 3, 10)

[123] Paul Syverson, Michael Reed, and David Goldschlag. Onion Routing access configurations. In *DARPA Information Survivability Conference and Exposition (DISCEX 2000)*, volume 1, pages 34–40. IEEE CS Press, 2000. (cited on page 46)

[124] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an Analysis of Onion Routing Security. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000. (cited on page 46)

[125] Tor. Frequently Asked Questions. `http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ`, June 2006. (cited on page 48)

[126] Yiannis Tsiounis. Anonymity & Privacy: The InternetCash Example. `http://www.internetcash.com/fgo/0,1383,white02,00.html`, June 2006. (cited on page 33)

[127] Vodafone. New 3G data card price plans. `http://shop.vodafone.co.uk/index.cfm?fuseaction=home.view3Gdatacards`, June 2006. (cited on page 87)

[128] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Efficient Collision Search Attacks on SHA-0. In *Advances in Cryptology – CRYPTO 2005 Proceedings*, Lecture Notes in Computer Science, August 2005. (cited on page 25)

[129] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Finding Collisions in the Full SHA-1. In *Advances in Cryptology – CRYPTO 2005 Proceedings*, Lecture Notes in Computer Science, August 2005. (cited on page 25)

[130] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The Active Badge Location System. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992. (cited on page 65)

[131] Graeme Wearden. London gets a mile of free Wi-Fi. ZDNet UK, `http://news.zdnet.co.uk/communications/wireless/0,39020348,39195421,00.htm`, April 2005. (cited on page 80)

[132] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, January 1995. (cited on pages 95, 96)

[133] Matthew K. Wright, Micah Adler, Brian Neil Levine, and Clay Shields. The Predecessor Attack: An Analysis of a Threat to Anonymous Communications Systems. *ACM Transactions on Information and Systems Security*, 7(4):489–522, 2004. (cited on pages 54, 55)