Towards a framework for connection anonymity

HEIKO TILLWICK AND MARTIN OLIVIER
University of Pretoria

Anonymising services have evolved from simple proxies to complex systems. Numerous techniques have been developed to thwart and confuse attackers, thereby improving the degree of anonymity. These techniques are often presented as additional advantages of specific anonymising services. Comparisons of anonymity services exist, however, there is a need for a more structured approach towards the understanding of the various techniques employed by these services.

This paper takes a meta-level look at connection anonymity, how it has evolved and how and why certain design choices are made. A conceptual framework describing what we consider to be important connection anonymity factors will be proposed. We consider design factors, fundamental connection anonymity functions and objectives. The framework aims to provide for a more structured and formal view of current anonymising strategies and techniques. It should thereby set the stage for further advances in connection anonymity.

Categories and Subject Descriptors: C.2.2 [**Network Protocols**]: Applications; H.3.5 [**Online Information Sharing**]: Data Sharing—*Security*; K.4.1 [**Public Policy Issues**]: Privacy

General Terms: Privacy, Anonymity, Anonymising Technologies

Additional Key Words and Phrases: Unlinkability, Connection Anonymity, Framework

1. INTRODUCTION

A large body of research has been dedicated to building and analysing various forms of anonymising technologies. The types of anonymity provided by these technologies can be split into two categories: data anonymity and connection anonymity. Data anonymity deals with identifying information in the data itself, whereas connection anonymity protects the communication channel between the sender and receiver. Data anonymity has benefited from the relative maturity of cryptographic techniques. Connection anonymity, although not as mature as data anonymity, has received a fair amount of attention. However, the approaches and effectiveness of the proposed solutions vary considerably.

Absolute connection anonymity has proven to be a difficult objective. A large number of identity disclosure attacks are known rendering many of the simpler solutions inadequate for certain users or for certain application domains. This has prompted designers to employ numerous techniques to thwart and confuse attackers. Anonymising technologies have evolved from simple proxies to complex solutions that offer better degrees of anonymity.

However, stronger anonymity often comes at the price of decreased efficiency. This has resulted in the majority of the research focusing on anonymous remailers; mainly because email is a high latency and low volume communication. More recent research has seen increased interest in the use of adequate anonymising technologies for use in other application domains such as web browsing and other interactive or even real-time applications. Efficiency and performance play a more significant role in these systems.

Other criteria besides the degree of anonymity have therefore emerged. These criteria are often presented as additional advantages of specific anonymising systems. However, comparisons often disregard these criteria and solely focus on the degree of anonymity. A clearer understanding of the requirements and objectives of such technologies is required if further advances in connection anonymity are to be made.

Our previous research includes work done on anonymous web browsing technologies. These technologies are called Polar [Tillwick et al. 2005] and Flocks [Olivier 2004]. There is a need to critically analyse their usefulness and effectiveness within the broader context of connection anonymity. The designs of Polar and Flocks do not

Author Addresses:
H. Tillwick, Information and Computer Security Architectures (ICSA) Research Group, Department of Computer Science, University of Pretoria, Pretoria, 0002, South Africa; htillwick@cs.up.ac.za
M.S. Olivier, Information and Computer Security Architectures (ICSA) Research Group, Department of Computer Science, University of Pretoria, Pretoria, 0002, South Africa; molivier@cs.up.ac.za

offer the same level of anonymity as some more complex solutions. However, they attempt to bridge the gap between simple anonymising proxies and inefficient mixes (that will be discussed next). Both Polar and Flocks have slightly different objectives and offers various other advantages besides anonymity. Comparisons based on the degree of anonymity alone do not suffice. A framework modelling the fundamental functions and objectives of connection anonymity technologies will be useful. Such a framework is, according to our knowledge, not available. An attempt at such a framework will be presented in this paper.

This paper takes a meta-level look at connection anonymity, how it has evolved and how and why certain design choices are made. A conceptual framework describing what we consider to be important connection anonymity factors will be proposed. We consider design factors, fundamental connection anonymity functions and objectives. This paper introduces continual anonymity as a new perspective to connection anonymity and considers it in context of our framework. We take note of the fact that research on connection anonymity will see increased interest in interactive or real-time applications. The framework will thus aim to provide structure to the understanding of existing connection anonymity techniques so as to set the stage for future research.

This paper is structured as follows. Section 2 will provide a brief overview of current connection anonymity technologies. Section 3 discusses two perspectives on anonymity on the Internet and introduces continual anonymity. The framework is introduced in section 4. The design factors, the connection anonymity functions and the objectives are discussed in sections 5, 6 and 7 respectively. Finally we conclude with a brief discussion in section 8 and the conclusion in section 9.

2. BACKGROUND

This section provides a brief overview of current connection anonymity technologies. We merely wish to highlight some important architectural differences in order to set the scene for the rest of the paper. A more complete overview can be found elsewhere [Oppliger 2000; Goldberg et al. 1997; Goldberg 2002; Diaz et al. 2002].

Proxies were originally used for caching to conserve the use of external bandwidth. However, they also provide a limited means of identity hiding and thus feature significantly in connection anonymity. To our knowledge, dc-nets [Chaum 1988] and variations thereof are the only non-rerouting based connection anonymity technology. Instead broadcasts are used to facilitate a multi-party computation. Although secure, a dc-net is impractical and easy to disrupt; a single corrupt member can disrupt the service.

Two of the more well-known anonymising proxies include `anon.penet.fi` and Anonymizer. `Anon.penet.fi` was one of the first anonymous remailers. Replies to an anonymously sent email is made possible through the use of pseudonyms. The remailer keeps a correspondence table mapping real email addresses to pseudonymous addresses. Similarly, Anonymizer pursues anonymity in web browsing by relaying web requests through a single proxy.

These solutions, although simple and efficient, have numerous drawbacks. Because the service is centrally managed, it is susceptible to a large number of attacks, most notably denial of service and compulsion (legal) attacks. Legal pressure forced `anon.penet.fi` to reveal the identities of certain users [Helsingius 1996]. The service has since shut down fearing further legal actions. A further threat is the possibility of a compromised system. This could result in all users' identities being revealed including a history of their web or mail usage (in those cases where logs are kept). There is also the possibility that (corrupt) system administrators release or even sell information. Thus, a considerable amount of trust in the integrity of the system and the administrators is required.

Anonymizer is still a popular service. This can be attributed to a number of reasons: it is simple to use and requires no additional software, there are few suitable replacements and it caters for people who only require mild protection. Anonymizer and other similar solutions have instead introduced a number of additional, value-adding features. These are mostly data anonymity techniques such as encryption of the communication channel, the removal of identifying information embedded in the request and the provision of pseudonyms for use with specific web sites [Gabber et al. 1999].

Cypherpunk remailers adopt a more distributed architecture, allowing for multiple proxies to be chained together. Reply blocks are used instead of correspondence tables. Reply blocks accompany the email and contain the encrypted return address. Only the remailer can decrypt it. Nevertheless, it is still an improvement over previous versions since return addresses are no longer held with the remailer. These remailer are known as Cypherpunk or type I remailers (as opposed to type 0 remailers such as `anon.penet.fi`).

Type I remailers are still vulnerable to a large number of attacks. These include message coding attacks, timing attacks and message length attacks; each is performed by correlating a remailer's incoming and outgoing messages using their coding, time and length respectively.

A solution to these attacks was presented by Chaum [1981] and forms the bases for the type II remailers. Chaum proposes the concept of a mix that hides the correspondence between incoming and outgoing messages.

This is achieved by imposing strict size-invariance through slicing and padding, performing cryptographic transformations on the message, batching and reordering messages and providing cover traffic which (to an outsider) is indistinguishable to real traffic. These obfuscation techniques as well as the method of routing between the mixes has received a considerable amount of attention. Because of their performance and resource implications, they are mostly used for low-latency, store-and-forward mediums such as email. Section 6 will take a closer look at some of the current mix strategies.

Cryptography forms an important part of anonymising services. This includes end-to-end as well as link-to-link encryption. However, a number of colluding proxies can still compromise the system. Goldschlag et al. [1996] propose the use of onions to address this threat. Each message is successively encoded with layers of encryption corresponding to each node in a chain of onion routers. Each onion router removes his respective layer of encryption before forwarding to the next router. Decryption thus occurs in reverse order as the encryption process. This approach requires only one honest router to preserve connection anonymity and is used in most mix implementations.

Crowds, another solution to connection anonymity, is presented by Reiter and Rubin [1998]. Crowds attempts to achieve anonymous web browsing by collecting users into a group. Members of this group collaborate by forwarding web requests among themselves before passing them to the specified web server. The choice of forwarding to another proxy instead of to the end server is a random decision based on some system-wide parameter $p_f > \frac{1}{2}$. Crowds effectively introduces the use of volunteers as opposed to dedicated servers.

Other solutions have since adopted the Crowds concept. Our previous work on Polar [Tillwick et al. 2005] and Flocks [Olivier 2004] are examples thereof. Flocks considers caching and logging and allows identity disclosure in case of a forensic investigation. It differs slightly to Crowds because it groups the proxies into one administrative domain and only allows for external requests. Polar considers the use of a structured peer-to-peer overlay to improve performance. Tarzan [Freedman and Morris 2002] is another example of a solution using a structured peer-to-peer overlay. Polar uses a topology-aware routing algorithm whilst Tarzan operates at the IP-layer (as opposed to the application layer as most anonymising technologies). Tarzan also uses mixing strategies. The use of a peer-to-peer model allows for a number of advantages over the traditional client-server model, but also introduces new vulnerabilities.

The rest of the paper will take a closer look at some of the more subtle differences between the various connection anonymity technologies. More specifically, we are interested in the different routing and obfuscation techniques with respect to each technology's unique objectives. However, we will first consider the motivation for anonymity on the Internet.

3. MOTIVATION

The need for anonymity on the Internet is two-fold. Firstly, in society, anonymity is often seen as an acceptable form of privacy protection. Anonymous activities include voting, counselling, whistle-blowing, refereeing and voicing political or other dissent. These activities should ideally also be possible online.

The second need arises from the transparent and distributed nature of the Internet. The ease with which information can be collected and redistributed has put users at risk of being profiled. Online privacy is threatened by long-term data storage and eventual disclosure as well as any unauthorised monitoring of network traffic. These threats have raised many privacy concerns. Anonymity, although not suitable for all cases, could be one solution to avoid profiling.

One should consider the case of online search engines. Brandi and Olivier [2004] have shown that inference attacks, based on the search queries submitted by a user over a considerable amount of time, are possible. Analysing a user's search queries could reveal a substantial amount of information about the user's interests or browsing habits. The collection of such information could be used for targeted marketing purposes or worse, could reveal incriminating or sensitive information. This not only applies to search queries but can be generalised to include any personally identifying information recorded by web or other logs.

Anonymity on the Internet is thus required on two levels. In the first case, absolute (or close to absolute) anonymity is needed for specific uses. Identity disclosure could have detrimental consequences for the individual. In the second case, emphasis is placed on *continual* identity protection. Identity disclosure of a single event might not be severe, but *continual* identity disclosure leads to profiling. This difference is often overlooked. In fact, we know of no other work that explicitly makes this distinction. We therefore introduce *continual anonymity* as a different perspective of anonymity on the Internet.

Making this distinction is beneficial for several reasons. The degree of anonymity as well as any performance or reliability expectations will differ for solutions that cater for either one of the cases. One could argue that any technology that offers absolute anonymity will be useful in both cases. This is true, only if such a technology can be used transparently, without side-affects. This is unlikely because the degree of anonymity inversely affects

the performance of the system; this is not a proven fact but seems likely because many obfuscation techniques require a time delay. Section 6.2 will look at this statement more closely. As an example, one could consider how practical a slow but secure mix is for use in everyday browsing. We believe that *continual anonymity* will become more and more important as Internet privacy becomes an increasingly controversial topic.

We wish to define our framework by taking specific cognisance of these two perspectives on anonymity. Moreover, we wish to introduce other objectives in addition to the degree of anonymity. We hope that by providing a more structured and formal approach, further advances in connection anonymity can be facilitated that not only consider absolute anonymity but also continual anonymity.

4. FRAMEWORK

This section introduces our framework, allowing the rest of the paper to elaborate on it. Later sections will discuss individual issues in more detail.

The framework is depicted in figure 1. It consists of three parts namely the design factors, the connection anonymity functions and the objectives. These will be covered in sections 5, 6 and 7 respectively. Although, no further reference is made to the numbering used in the framework (e.g. A.2. - receiver unlinkability), we believe it allows for easy identification and allows for specific references to individual items in the framework.

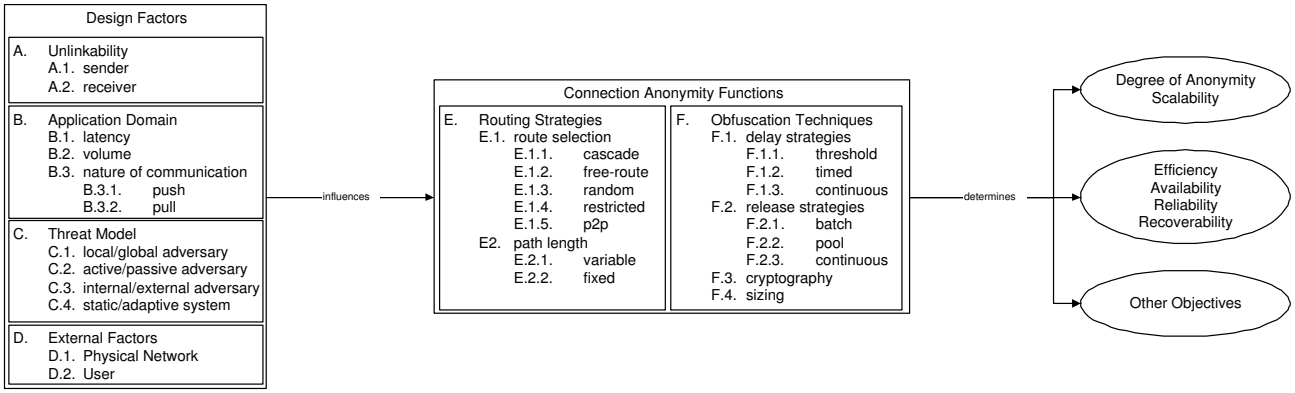

Figure 1. A conceptual framework for connection anonymity

We consider four design factors: the unlinkability, the application domain, the threat model and the external factors. Each has a number of sub-items. The design factors are heuristic measures beneficial to the design (or also the evaluation) of a connection anonymity service. Many anonymity solutions already explicitly or implicitly address some of these. However, a more formal and extensive approach is attempted here.

The connection anonymity functions are fundamental tasks that a connection anonymity service performs. The framework considers two independent functions: routing strategies and obfuscation techniques. Each has an associated number of alternatives. These alternatives are not all-inclusive but present most of the techniques currently employed by connection anonymity services. These techniques are therefore a summary of techniques invented by designers of current services.

Our contribution lies in the classification and categorisation of these techniques. More specifically, we are interested in the alternatives and their associated advantages and disadvantages with respect to the given objectives. These advantages and disadvantages will be discussed in in the following sections but do not feature in the framework itself. This is because they often depend on a number of implementation-specific issues, making any meta-level approach difficult if not impossible. Instead, the advantages and disadvantages are related to the objectives of a connection anonymity service.

Our framework considers three classes of objectives. The first is related to the degree of anonymity offered by a system. The second features efficiency and reliability issues. The first emphasises absolute anonymity whereas the second allows for continual anonymity. The trade-off between the degree of anonymity and the performance of the service will feature significantly in the the rest of the paper. Finally, the third class of objectives are any service-specific objectives that are not directly related to the degree of anonymity or efficiency. This class will not be discusses further. It merely allows for additional objectives that are not fundamental connection anonymity objectives.

Before elaborating on the framework we would like to draw attention to the intended purpose and usefulness of the framework. The objectives of the framework are two-fold: 1) it should provide for a more structured and

formal view of current connection anonymity strategies and techniques and 2) should thereby facilitate further advances in connection anonymity.

Following is a discussion on the framework starting with the design factors.

5. DESIGN FACTORS

When designing an anonymity service a number of factors need to be considered. These factors will define various requirements that the the anonymising technology needs to address. We draw from experiences made by previous technologies in order to identify and explicitly define these factors. Anonymity versus unlinkability, the intended application domain and the expected threat model will be discussed. These factors present the design considerations in our conceptual framework.

5.1 Anonymity and unlinkability

We adopt the definition given by Pfitzmann and Köhntopp [2001]: *Anonymity is the state of being not identifiable within a set of subjects, the anonymity set.* We also take note of the difference between anonymity and unlinkability. Pfitzmann and Köhntopp define unlinkability as *two or more items (e.g. subjects, messages, events, actions, . . .) that within a system are no more no less related than they are related concerning the a-priori knowledge.*

The anonymising technologies mentioned in section 2 do in fact not achieve anonymity (as defined by Pfitzmann and Köhntopp) but rather attempt unlinkability. On the Internet, participants have an IP address making them clearly distinguishable from from each other. Instead, these technologies protect the communication channel by attempting to hide the relation between a message (an email, web request or other) and the sender of the message. We will use the terms anonymity and unlinkability interchangeably, but note that in all cases (except where mentioned) we are indeed referring to unlinkability.

A distinction is made between sender and receiver anonymity. In this paper we do not consider responder anonymity. A responder is essentially both a receiver and a sender. We note that further research is required to determine if responder anonymity warrants having its own category; the fact that a response is generated in reply to a request could potentially threaten the unlinkability of the request and/or response.

Publisher anonymity, as offered by other systems [Goldberg and Wagner 1997; Marc Waldman and Cranor 2000; Clarke et al. 2000], is also not discussed in this paper. These systems offer a combination of data and connection anonymity services and offer censorship-resistance in addition to anonymity. Incorporating publisher anonymity into our framework is left as future work.

The framework will thus provide for sender and receiver anonymity. Together they will form the first design consideration under the heading of Unlinkability.

5.2 Application domain

Different applications will often have different communication properties. This becomes evident when comparing email, web browsing and Internet conferencing. Each of these has different performance expectations. This difference will ultimately also have an affect on the respective anonymising technologies.

We consider the following: *store-and-forward*, *interactive* and *real-time* mediums. Each has different *latency* and *volume* properties. They can also be categorised as either a *push* or a *pull* technology. Email is a push technology whereas HTML is a pull technology (a pull technology implicitly implies that one receiver is also a responder).

These communication properties will affect the routing strategies (discussed in section 6.1). The framework will group these factors under the heading of Application Domain.

5.3 Threat model

A threat model deals with the capabilities of an adversary. The capabilities will differ depending on whether the adversary is an individual, a corporation with large resources or a national entity with large resources as well as authority over the legal system. In context of the Internet, this will have consequences on the following:

—*Local-Global Adversary*: A global adversary has an overview of the whole system whereas a local adversary can only observe a part of it. On the Internet, it depends on how many participating nodes are located in the fraction of the network that an adversary can observe.

—*Active-Passive Adversary*: A passive adversary only listens whilst an active adversary adds, removes or modifies messages or directly attacks participants of a system.

—*Internal-External Adversary*: An internal adversary has access to or knowledge of the internal workings of a system in contrast to the communication channel only.

The final property is more applicable to the anonymity system since adversaries are generally considered to be adaptive - if an attack has been performed once it is often repeatable unless the system changes. In contrast, anonymity system are not always adaptive.

—*Static-Adaptive System*: An adaptive system is able to recover from an incident; an attack will only leave the system compromised for a limited amount of time. In encryption, periodic key renewals will prevent any compromised key from compromising all future transactions. Such a system is also described as being *forward-secure*.

We will not consider individual attack techniques as a separate factor in our framework. This is because attacks affect the vulnerability of a system which directly affects the degree of anonymity (this is discussed in section 7.1). Many of the known attacks are particular to specific design choices or to the implementation of the technologies. Message coding and timing, denial of service and compulsion attacks are some of the attacks already mentioned. Raymond [2000] present a more detailed discussion on the different anonymity attacks.

5.4 External Factors

Two external factors are considered: the physical network and the user. These are deployment issues which can indirectly affect the effectiveness of an anonymising technology (including the degree of anonymity).

These issues generally receive limited attention, however, some examples do exist. Reiter and Rubin [1998] note that users sitting behind a firewall might not able to effectively participate in Crowds. This is because Crowds requires incoming connections as well as outgoing connections. Feamster and Dingledine [2004] discuss network diversity and the impact it has on the level of anonymity offered by a system. As for the user, knowledge of a user's habits or preferences can give an adversary a substantial amount of information. This could subsequently lead to an inference attack [Farkas and Jajodia 2002].

More research is required to further analyse the impact that these external factors could have on an anonymising technology. However, this is beyond the scope of the paper.

6. CONNECTION ANONYMITY FUNCTIONS

The previous section identified unlinkability as the true objective of a connection anonymity technology, considered the application domain, the threat model as well as external factors. By keeping these issues in mind, a more formal approach at identifying fundamental connection anonymity functions can be made.

This section considers the basic functions of a connection anonymity technology. A review of the background work in section 2 has revealed that most connection anonymity technologies are routing based and thus use one or more proxies. It was also observed that various obfuscation techniques are required to increase the level of anonymity. We therefore identify routing and obfuscation as the two fundamental functions of a connection anonymity service. The functions and the techniques used are discussed in more detail. Although, we do claim that all possible variations are covered, we do intend to cover a good proportion of the known techniques. References to existing solutions that employ the respective techniques will be made where appropriate.

6.1 Routing strategies

We have identified two aspects of routing: route selection and path length. These aspects affect various properties of a system including the scalability, availability, efficiency as well the degree of anonymity.

6.1.1 *Route selection*

Routing forms an integral part of connection anonymity. Routing algorithms are networking problems, however, this sections will consider various strategies that are relevant to connection anonymity. More specifically we observe how and why certain selection strategies are chosen above others.

—*Single proxies* as used by Anonymizer and `anon.penet.fi` allow for a simple and efficient implementation that is vulnerable to a large number of attacks. It is a centralised architecture that has a single point of failure. It suffers from trust issues and also scales poorly. This architecture offers the weakest protection against known attacks.

—*Cascades* solve a number of issues by chaining proxies. Several mix technologies adopt this approach. These are employed in some real-time systems [Pfitzmann et al. 1991] and are also for anonymous web browsing [Berthold et al. 2000]. Berthold et al. [2000] argue that cascade mixes provide better protection against large scale adversaries than free-route mixes. The disadvantages of cascades is that the cascade consists of default proxies which have to be used. Also, the failure of a single proxy renders the whole system inoperable.

—*Free-route* solutions allow users to choose their own path based on reliability statistics. This assumes a fully connected network and requires some configuration on the client side. Mixmaster [Möller et al. 2003] employs this approach.

—*Restricted routes* were first proposed by Danezis [2003] and is a combination of cascades and free-route networks. It attempts to incorporate advantages offered by both approaches.

—*Random* routing deals with plausible deniability. A system-wide routing parameter determines the probability that the previous proxy is the original sender. Crowds [Reiter and Rubin 1998] is the best known technology that employs this technique. The expected path length depends on the system-wide parameter as well as the number of participants. A trade-off is made between the average path length and the resultant degree of anonymity. Olivier [2004] explores this relation in more detail.

—*Structured peer-to-peer* routing is used in more recent systems [Tillwick et al. 2005; Freedman and Morris 2002]. As with Crowds, volunteers are used instead instead of dedicated server. This makes it also very scalable. It is more resilient against legal attacks because no single entity can be held accountable. However, attacks using knowledge of the routing algorithms are possible, making these less secure than some traditional mixes. Also, peer-to-peer systems often suffer from reliability and trust issues resulting from the lack of (or distributed) control.

6.1.2 *Path length*

The framework considers the path length as the second routing factor.

—*Fixed* path lengths are used in cascades and in some implementation of free-route mixes. The known path length is used in a number of attacks to reduce the anonymity set (i.e. the set of possible senders or receivers).

—*Variable* path lengths are considered to be superior to fixed path lengths [Guan et al. 2002]. Random and peer-to-peer routing algorithms use variable path lengths. Some free-route mixes also have variable path lengths. However, these often impose a maximum path length.

Other routing issues besides route selection and path length do exist. This includes synchronisation (as opposed to asynchronous communication) as well as any other protocol specific issues. We will however only consider those routing issues which we consider to be the most relevant to connection anonymity i.e. those that have a direct impact on the degree of anonymity as well as any efficiency or reliability issues.

6.2 Obfuscation techniques

Obfuscation techniques are issues independent of the underlying routing strategies. The appropriateness of these techniques depend considerably on the application domain and its associated communication properties.

Anonymising technologies are often evaluated against a very strong threat model. A large number of attacks that could potentially lead to identity disclosure are known. This makes it difficult to find a solution that offers absolute anonymity. Solutions that provide a 'good' level of anonymity employ numerous techniques to thwart or confuse an attacker. These techniques have considerable resource and time requirements and therefore mostly target non-interactive high latency environments such as email.

Research on remailers has led to many advancements in obfuscation techniques. Many of these techniques were originally proposed by Chaum [1981] who subsequently named it mixing. A number of enhancements or additions to mix techniques have hence been made. A number of these enhancements will be considered next.

This section will attempt to summarise many of these techniques. We will not provide a detailed discussion on each but will instead focus on the benefits offered by each technique.

6.2.1 *Delay strategies*

Delays are used to prevent timing attacks. Timing attacks correlate incoming and outgoing messages using the arrival and departure time. Delays thus allow for the collection of multiple messages within a time-frame. These can subsequently be reordered before being forwarded.

—*Threshold* mixes collect a fixed number of messages before releasing them. Henceforth, low volume communications suffer from long delays.

—*Timed* mixes restrict the amount of time the messages can be delayed. Mixes thus flush their messages periodically. This is more suitable for a low latency communication medium but is less secure for low volume mediums.

—*Continuous* mixes [Kesdogan et al. 1998] treat messages individually instead of in batches. Every message is assigned a random delay time from an exponential distribution. They were originally called stop-and-go-mixes

and attempted to improve on threshold and timed mixes. The trade-off between performance and degree of anonymity depends on the chosen distribution function.

6.2.2 *Flushing strategies*

Flushing strategies imply the use of a delay strategy. Both strategies are thus closely related.

—*Batch* flushing was originally used. Here all messages are released at once.

—*Pool* mixes achieve better degrees of anonymity than batch mixes. In each round, only a random number of messages are released. The other messages are kept for the next round. This is repeated continually allowing for messages to remain in the mix for a number of iterations. Pool mixes do not only mix all messages in a particular batch but extend this to include multiple batches. Mixmaster [Möller et al. 2003] is one system that employs this technique. The disadvantage is that a message is not guaranteed to leave the mix after a round. It could potentially stay in the mix for a number of rounds.

—*Continuous* flushing implies the use of a continuous delay strategy.

Other obfuscation techniques include *cover traffic*, *sizing* and *cryptographic transformations*. These were originally also proposed by Chaum and therefore also attempt to hide the correlation between input and output message. Sizing and cryptographic transformation prevent message length and coding[1] attacks. While sizing and cryptographic transformations impose minimal overhead, cover traffic is resource intensive.

7. OBJECTIVES

Degree of anonymity and efficiency issues features significantly in the previous discussion on the design considerations and the connection anonymity functions. These issues will therefore be considered as separate objectives in our framework.

7.1 Degree of anonymity

Different anonymity services provide different degrees of anonymity. However, measuring this degree is not a trivial task. Reiter and Rubin [1998] present a qualitative scale of anonymity. This scale ranges from *absolute anonymity* on the one side, where one cannot perceive the presence of a communication, to *exposed* on the other. *Beyond suspicion* is applicable when there is evidence of a communication, but all potential senders are equally likely to have sent the message. *Probable innocence* implies that the probability of having sent a message is less than the probability of not having sent the message; and inversely so for *possible innocence*.

Chaum [1988] realises the benefit of anonymity set sizes; the degree of anonymity is directly related to the cardinality of the anonymity set. This allows for a quantitative approach and is thus used by a number of solutions. The cardinality of the anonymity set is dependant on the *scalability* of a system. We therefore consider scalability as an additional objective in our framework.

Another approach to measuring the degree of anonymity is offered by Serjantov and Danezis [2002] and Diaz et al. [2002]. They propose that the probabilities of users sending messages should be taken into account. This places more emphasis on the routing and obfuscation strategies rather than on the number of participants.

Absolute anonymity is a desirable but often an unrealistic goal. A considerable amount of research has been dedicated towards exposing or degrading anonymising protocols [Wright et al. 2002; 2004]. A large number of attacks are known that decrease the degree of anonymity even in the more complex solutions. Probabilistic attacks might not necessarily result in identity disclosure, however, they are (by definition) attacks that increase the probability of a sender being exposed.

This has resulted in anonymising services employing numerous techniques to confuse and thwart an attacker. Anonymising services therefore impose a considerable overhead (routing and obfuscation techniques) in order to improve the degree of anonymity. This overhead affects the usability of the system, requiring the user to have sufficient motivation for the use thereof.

7.2 Efficiency

Continuous anonymity places increased emphasis on efficiency with an acceptable (as opposed to absolute) level of anonymity. What constitutes acceptable depends on the user, the application domain and the intended uses. As far as continuous anonymity is concerned, an acceptable degree of anonymity will make it difficult (or unfeasible) for an attacker to continuously perform the attack.

Any solution that is inefficient and time consuming will not gain widespread acceptance for use in daily activities. This means that solutions that have considerable overheads or impose detracting delays will only

[1]Coding attacks directly correlate the message code or signature of two or more messages.

cater for a select few users i.e. those users who require strong anonymity for use in very specific tasks. These solutions will probably not be used by individuals who require weak protection. This will inevitably result in users refraining from using the anonymity service.

We have therefore identified efficiency as an additional objective in addition to the degree of anonymity. We foresee increased interest in continuous anonymity for use in daily activities. Availability, reliability and recoverability are additional objectives more commonly associated with continuous anonymity than with absolute anonymity.

8. DISCUSSION

The framework covers some of the fundamental aspects of connection anonymity. Future research might reveal additional factors or issues that merit being included. We believe the framework provides a basis which will allow for additions or possibly some modifications. Connection anonymity is, compared to other research fields in security such as cryptography or access control, far less mature, especially for inter-active and real-time communication mediums. We foresee increased interest in anonymity for use in these low latency communication mediums.

The framework should provide a basis with which current strategies and techniques can be evaluated and compared against each other, thereby facilitating a more structured approach to future connection anonymity solutions.

9. CONCLUSION

This paper considered connection anonymity. An analysis of the progression of anonymising technologies revealed that a number of different approaches are employed by current anonymising technologies. Each has different benefits and drawbacks. A need for a more structured approach towards the understanding of these approaches was identified. The technologies were traditionally evaluated against a very strong attacker model and therefore attempt absolute anonymity. Continual anonymity was introduced as a new perspective to connection anonymity. It places increasing emphasis on efficiency and usefulness. Strategies and techniques employed in current anonymising technologies were analysed and considered with respect to absolute and continual anonymity.

This paper took a meta-level look at connection anonymity and proposed a framework. The framework aims to provide for a more structured and formal view of current connection anonymity strategies and techniques. It should thereby also facilitate further advances in connection anonymity.

The framework considered design factors, fundamental connection anonymity functions and objectives. The bulk of the paper explored each of these in more detail. Four design factors were considered namely: unlinkability, the application domain, the threat model and external factors. These factors serve as heuristic measures beneficial to the design. Two independent connection anonymity functions were discussed: routing strategies and obfuscation techniques. An analysis of current connection anonymity services revealed a number of alternative techniques for each function. Finally, three separate objectives were identified. These include the degree of anonymity, efficiency and other objectives.

Further research is required to extend the framework to include publisher anonymity. This could reveal a whole new set of objectives and possibly some new anonymity functions. However, we believe the framework is a step in the right direction and hope that future research will use it and extend on it.

REFERENCES

Berthold, O., Federrath, H., and Köpsell, S. 2000. Web MIXes: A system for anonymous and unobservable Internet access. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, 115–129.

Berthold, O., Pfitzmann, A., and Standtke, R. 2000. The disadvantages of free MIX routes and how to overcome them. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, 30–45.

Brandi, W. A. and Olivier, M. S. 2004. On privacy and the Web. In *Proceedings of the Fourth Annual Information Security South Africa Conference (ISSA2004)*. Information Security South Africa (ISSA), Midrand, South Africa. Published electronically.

Chaum, D. L. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM 24,* 2, 84–90.

Chaum, D. L. 1988. The dining cryptographers problem: unconditional sender and recipient untraceability. *J. Cryptol. 1,* 1, 65–75.

Clarke, I., Sandberg, O., Wiley, B., and Hong, T. W. 2000. Freenet: A distributed anonymous information storage and retrieval system.

Danezis, G. 2003. Mix-networks with restricted routes. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, R. Dingledine, Ed. Springer-Verlag, LNCS 2760.

Diaz, C., Seys, S., Claessens, J., and Preneel, B. 2002. Towards measuring anonymity. In *Proceedings of PET 2002*. Springer-Verlag GmbH, San Francisco, United States of America.

Farkas, C. and Jajodia, S. 2002. The inference problem: a survey. *SIGKDD Explor. Newsl. 4,* 2, 6–11.

Feamster, N. and Dingledine, R. 2004. Location diversity in anonymity networks. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004)*. Washington, DC, USA.

Freedman, M. J. and Morris, R. 2002. Tarzan: a peer-to-peer anonymizing network layer. In *CCS '02: Proceedings of the 9th ACM conference on computer and communications security*. ACM Press, Washington, DC, USA, 193–206.

Gabber, E., Gibbons, P. B., Kristol, D. M., Matias, Y., and Mayer, A. 1999. Consistent, yet anonymous, Web access with lpwa. *Commun. ACM 42,* 2, 42–47.

Goldberg, I. 2002. Privacy-enhancing technologies for the Internet, II: Five years later. Workshop on Privacy Enhancing Technologies 2002.

Goldberg, I. and Wagner, D. 1997. Taz servers and the rewebber network enabling anonymous publishing on the World Wide Web. *First Monday electronic journal 3,* 4 (May), 1–14.

Goldberg, I., Wagner, D., and Brewer, E. 1997. Privacy-enhancing technologies for the Internet. In *COMPCON '97: Proceedings of the 42nd IEEE International Computer Conference*. IEEE Computer Society, Washington, DC, USA, 103.

Goldschlag, D. M., Reed, M. G., and Syverson, P. F. 1996. Hiding routing information. In *Proceedings of the First International Workshop on Information Hiding*. Springer-Verlag, London, UK, 137–150.

Guan, Y., Fu, X., Bettati, R., and Zhao, W. 2002. An optimal strategy for anonymous communication protocols. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*. IEEE Computer Society, Washington, DC, USA, 257.

Helsingius, J. 1996. Johan Helsingius closes his Internet remailer. http://www.fitug.de/news/1997/penet.html.

Kesdogan, D., Egner, J., and Büschkes, R. 1998. Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In *Proceedings of Information Hiding Workshop (IH 1998)*. Springer-Verlag, LNCS 1525.

Marc Waldman, A. D. R. and Cranor, L. F. 2000. Publius: A robust, tamper-evident, censorship-resistant, Web publishing system. In *Proc. 9th USENIX Security Symposium*. USENIX, Denver, Colorado, USA, 59–72.

Möller, U., Cottrell, L., Palfrader, P., and Sassaman, L. 2003. Mixmaster Protocol — Version 2. Draft.

Olivier, M. S. 2004. Flocks: Distributed proxies for browsing privacy. In *Proceedings of SAICSIT 2004 — fulfilling the promise of ICT*, G. Marsden, P. Kotzé, and A. Adesina-Ojo, Eds. South African Institute for Computer Scientists and Information Technologists, Stellenbosch, South Africa, 79–88.

Oppliger, R. 2000. Privacy protection and anonymity services for the World Wide Web (WWW). *Future Generation Computer Systems 16*, 379–391.

Pfitzmann, A. and Köhntopp, M. 2001. Anonymity, unobservability, and pseudeonymity - a proposal for terminology. In *International workshop on Designing privacy enhancing technologies*. Springer-Verlag New York, Inc., New York, NY, USA, 1–9.

Pfitzmann, A., Pfitzmann, B., and Waidner, M. 1991. ISDN-mixes: Untraceable communication with very small bandwidth overhead. In *Proceedings of the GI/ITG Conference on Communication in Distributed Systems*. 451–463.

Raymond, J.-F. 2000. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, Berkeley, CA, USA, 10–29.

Reiter, M. K. and Rubin, A. D. 1998. Crowds: anonymity for Web transactions. *ACM Trans. Inf. Syst. Secur. 1,* 1, 66–92.

Serjantov, A. and Danezis, G. 2002. Towards an information theoretic metric for anonymity. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, R. Dingledine and P. Syverson, Eds. Springer-Verlag, LNCS 2482.

Tillwick, H., Neumann, T., Olivier, M., Venter, H., and Eloff, J. 2005. Polar: Proxies collaborating to achieve anonymous Web browsing. In *Proceedings of the Fifth International Network Conference (INC2005)*. Samos, Greece.

Wright, M., Adler, M., Levine, B. N., and Shields, C. 2002. An analysis of the degradation of anonymous protocols. In *Proceedings of the Network and Distributed Security Symposium - NDSS '02*. IEEE.

Wright, M. K., Adler, M., Levine, B. N., and Shields, C. 2004. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Secur. 7,* 4, 489–522.

H Tillwick and MS Olivier, "Towards a framework for connection anonymity," in J Bishop and DG Kourie (eds), *Research for a changing world - Proceedings of SAICSIT 2005*, 113-122, White River, South Africa, September 2005