Analysis of Anonymity Services from a Tunable Perspective

Reine Lundin, Stefan Lindskog, and Anna Brunstrom

Department of Computer Science
Karlstad University
Karlstad, Sweden
{reine.lundin| stefan.lindskog| anna.brunstrom}@kau.se

Abstract. In this paper, we investigate the tunable features provided by Mix-Nets and Crowds using a conceptual model for tunable security services. A tunable security service is defined as a service that has been explicitly designed to offer various security levels that can be selected at run-time. Normally, Mix-Nets and Crowds are considered to be static anonymity services, since they were not explicitly designed to provide tunability. However, as discussed in this paper, they both contain dynamic elements that can be used to achieve a tradeoff between anonymity and performance.

1 Introduction

Today, many security services are rather static. That is, once designed the security configuration provided by the service is fixed and cannot be changed during run-time. Furthermore, the security configurations of security services are often set to achieve the highest possible level of security [4], which may affect the performance of the system. Hence, in situations where we have heterogeneous devices with varying and/or limited computing and energy resources, tunable security services that can change the security configuration at run-time to create a tradeoff between security and performance are desirable.

One important component of security is privacy [2], which has anonymity as one of its major goals. In [8] anonymity is defined as "the state of not being identifiable within a set of subjects, the anonymity set". That is, anonymity ensures that a user may use a resource without disclosing his or her identity. Two commonly used anonymity services are Mix-Nets [1] and Crowds [9]. The major difference between these two services is that Mix-Nets provides anonymity by hiding the relation between incoming and outgoing messages and Crowds provides anonymity by hiding one user's actions within the actions of many others.

In this paper, Mix-Nets and Crowds are analyzed using a conceptual model for tunable security services. Even though the two anonymity services were not initially explicitly designed as tunable anonymity services, they contain dynamic elements that can be used to achieve a tradeoff between anonymity and performance.

The rest of the paper is organized as follows. In Section 2, the conceptual model for tunable security services, used in the analysis, is presented. Using the conceptual model, Section 3 investigates Mix-Nets, and Section 4 investigates Crowds. Finally, Section 5 concludes the paper.

2 Conceptual Model

In [6], a conceptual model of tunable security services is presented. It describes in a formal way the requirements for tunable security services, and can thus be used to examine the construction and previous work of tunable security services. The model is described by the three sets:

- $T = \{\text{Tuner preferences}\}$
- $E = \{\text{Environmental descriptors}\}$
- $S = \{\text{Security configurations}\}$

and the mapping

$$TS : T \times E \rightarrow S \tag{1}$$

The TS function illustrates the mapping from tuner preferences, T, and environmental descriptors, E, to a particular security configuration, S. Hence, the TS mapping gives under which conditions the security configuration should be changed for the service. For example, when a device reaches a threshold in battery level the TS function makes a decision if the security configuration of the device should be changed to increase the remaining time of the battery. Note that, for a security service to be a tunable security service S must contain at least two security configurations, otherwise the service will be static. The same will happen if both T and E are singular sets, since then $T \times E$ is a singular set.

Through the elements in T, the tuner preferences, a tuner entity can affect the security configurations in order to achieve desired tradeoffs between security and performance. The tuner entities that set the tuner preferences of the security services typically exist on several layers, or phases of the system life cycle, such as system owner and/or end user. For example, a system owner might assign some tuner preferences for the provided service so that it fulfills the security policy of the company, while the end users in the same company are free to affect the rest of the preferences. The elements in T can be expressed at various abstraction levels, for example as low, mid, or high security, or by specifying frames or layers to encrypt in MPEG movies. T might also be constructed from several parameters, each representing a different security objective such as confidentiality and integrity. In E, the environment and applications descriptors that may influence the selection of security configurations are described. Possible elements in E include characteristics of equipment, type of attacker, energy consumption, and network load. The elements in S represent the possible security configurations of the tunable security service, such as encryption algorithm, MAC algorithm, key length(s), and key establishment algorithm.

In previous work, the above described conceptual model has successfully been used to examine the tunable features provided by seven different security services. Four services were analyzed in [6], the paper that introduced the model, and three additional services were evaluated in [5]. In this paper, we apply the conceptual model when analyzing two anonymity services, i.e., Mix-Nets and Crowds.

3 Analyzing Mix-Nets

To achieve untraceable electronic mail David Chaum introduced the idea of Mix-Nets [1]. A Mix-Nets is a network of special network stations called mixes, where each mix has the task of hiding the relation between incoming and outgoing messages. Hence, a Mix-Nets basically attains sender anonymity and unlinkability between sender and recipient. In Fig. 1, a Mix-Nets chain, which is a ordered sequence of mixes, is illustrated.

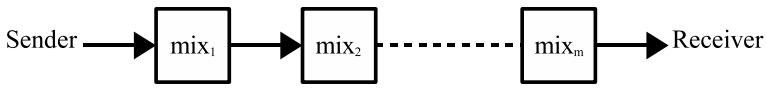

Fig. 1. A Mix-Nets chain.

The major work for a single mix is to collect messages in a pool, decide when a subset of messages should be flushed from the pool, and decide which subset of the messages in the pool to flush. The flushing conditions divide the mixes into two types, timed mixes and threshold mixes [3]. Timed mixes flushes on certain predefined time intervals and threshold mixes flushes when they have collected a certain amount of messages. A combination of the two types also exists [7]. The subset of messages to flush is determined by the pool flushing algorithm. Below we will analyze mixes that have a deterministic pool flushing algorithm [3], using the conceptual model for tunable security services, which was presented in the previous section.

3.1 Security Configurations (S)

A deterministic pool flushing algorithm uses the number of messages in the pool, n, to determine the number of messages to send out, s. For such mixes, we can write $s = nP$, where P is the fraction of sent messages, obviously $1 \leq s \leq n$.

Note, however, that the subset of sent messages are still randomly chosen from the pool, even if the number of sent messages is deterministic. See Fig. 2 for an illustration of a mix with a deterministic flushing algorithm.

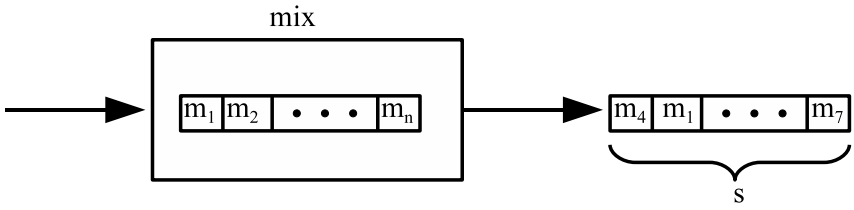

Fig. 2. A mix having n messages in the pool and flushing s messages.

The cycle of collecting and flushing messages is called one round. Furthermore, since Mix-Nets often consist of several mixes, we write s_{ij} to denote the number of sent messages in round i at mix j, $i \geq 1$ and $1 \leq j \leq m$. In a similar way, n_{ij} denotes the number of messages in the pool in round i at mix j. Now, since the only parameters that affect the security configurations (and also the performance) of Mix-Nets, with a deterministic pool flushing algorithm, are s_{ij} and n_{ij}, we get the following security configurations $S = \prod_{j=1}^{m}(s_{ij} \times n_{ij})$.

3.2 Tuner Preferences (T)

For deterministic mixes the security configurations are directly controlled by the tuners, since no abstraction is offered through the system of the security configurations. Furthermore, for Mix-Nets there might be a tuner for each mix, which is the system owner of the mix. Hence, the set of tuner preferences is in this case equal to the set of possible security configurations, $T = \prod_{j=1}^{m}(s_{ij} \times n_{ij})$. Although not explicitly expressed in T, the selection of a security configuration represents a tradeoff between the level of anonymity and the resulting overhead in delay of the system. Note that the end users have no possibilities to control the anonymity level in the system and are therefore not tuners.

3.3 Environmental Descriptors (E)

Since the set of tuner preferences and the set of security configurations are equal, the security configuration is directly controlled by the tuner. Hence, the set of environmental descriptors is in this case the empty set, i.e., $E = \emptyset$. However, the tuner can take the environment into account when selecting a security configuration.

3.4 The TS Mapping

The TS function is in this case the identity mapping, $TS(t, \emptyset) = t$, where $t \in T$. The simplicity of the TS function is an effect of the direct tuner control of the security configurations. It is thus up to the tuner to select an appropriate security configuration and to investigate the tradeoff between anonymity and performance.

4 Analyzing Crowds

The basic idea of Crowds [9] is to provide anonymous web browsing by hiding one user's web actions within the web actions of many others. The Crowds system consists of two main components. The Jondo proxy application, which the browser requests must be set to go through, and the Blender server for managing memberships. See Fig. 3 for an illustration of the Crowds system. Below the Crowds system is analyzed, using the conceptual model.

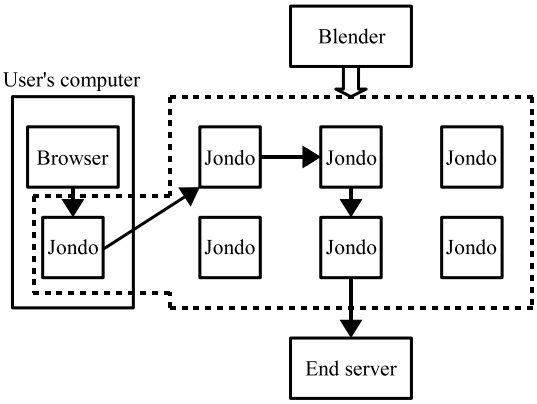

Fig. 3. The Crowds system.

4.1 Security Configurations (S)

One important parameter in Crowds is p_f, $0 \leq p_f < 1$. It gives the probability of forwarding a message, in the path creation process. When the first request arrives at a local Jondo, it forwards the request to another random Jondo in Crowds, possibly itself. The next Jondo, on the path, chooses to forward the request to another random Jondo in the Crowds system with probability p_f, or to submit the request to the end server with probability $1 - p_f$. This decision

process, forward or submit, continues until a Jondo submits the request to the end server. Since the only parameter that gives the security configuration in Crowds is p_f, we get that $S = p_f$.

4.2 Tuner Preferences (T)

In [9], the authors defined six anonymity levels (AL), based on the certainty that the sender is the real originator from the attacker's point of view.

- A sender has absolute privacy (AP), if an observation gives the attacker no additional information.
- A sender is beyond suspicion (BS), if though the attacker can see evidence of a sent message, the sender appears no more likely to be the originator of that message than any other potential sender in the system.
- A sender is probable innocence (PrI), if from the attacker's point of view, the sender appears no more likely to be the originator than to not be the originator.
- A sender is possible innocence (PoI), if from the attacker's point of view, there is a non trivial probability $\delta > 0$ that the real sender is someone else.
- A sender is exposed (Ex), if the attacker can identify the sender, but not necessary prove it to others.
- A sender is provably exposed (PE), if the attacker can identify the sender and also prove the identity to others.

In this paper, we treat the ALs Ex and PE as equal. Furthermore, as we will see later, BS and AP are only possible to achieve under probabilistic assumptions. Thus, $T = \{PrI, PoI, Ex\}$.

4.3 Environmental Descriptors (E)

Except from the p_f parameter, two further parameters, n and A, are needed to describe a Crowds system. The total number of Jondos in the Crowds system is represented by n, $n > 1$. A represents one of the following attacker types in a Crowds system.

1. A local eavesdropper (LE) is an attacker who can observe all communication to and from a specific Crowds member's computer.
2. Collaborating members (CM) are attackers in the form of Crowds members that can pool their information and even deviate from the prescribed protocol.
3. The end servers (ES) are attackers to which web requests are directed.

Thus, $A = \{LE, CM, ES\}$, and $E = n \times A$. When $A = CM$ we set $A = c$, the number of collaborating Jondos.

4.4 The TS Mapping

The sender/receiver ALs that are achieved by Crowds are given in [9]. However, it is only possible to tune sender anonymity when $A = c$, since this is the only occasion where the value of p_f affects the AL. When $A = c$, three cases can occur.

- If the path does not contain any CM sender anonymity is AP, which occurs with probability $p(AP) = 1 - \frac{c}{n-p_f(n-c)}$.
- If the path initiator does not have a CM as an immediate predecessor sender anonymity is BS, which occurs with probability $p(BS) = \frac{p_f c(n-c-1)}{n(n-p_f(n-c))}$.
- If the path initiator is the first CM immediate predecessor sender anonymity is X anonymity, which occurs with probability $p(X) = \frac{c(n-p_f(n-c-1))}{n(n-p_f(n-c))}$.

Even if p_f affects the probability of the three cases, it is only in the last case that Crowds have the possibility to guarantee an AL. In [9], the authors derived an anonymity measure $P(I|H_{1+})$ for X, where $P(I|H_{1+})$ is the probability that the path initiator is the first collaborator's immediate predecessor, given that there is at least one CM on the path.

$$\begin{aligned} P(I|H_{1+}) &= \frac{p(X)}{1-p(AP)} \\ &= 1 - p_f \frac{n-c-1}{n} \\ &= 1 - p_f N(n,c) \end{aligned} \tag{2}$$

The $N(n,c)$ in equation (3) is the fraction of non-CM in Crowds excluding your own Jondo. Furthermore, by setting $P(I|H_{1+}) \leq \frac{1}{2}$, Reiter and Rubin [9] showed that Crowds offers PrI if

$$n \geqslant \frac{p_f(c+1)}{p_f - \frac{1}{2}} \tag{3}$$

Hence, if we rewrite equation (3) Crowds offers PrI as long as $p_f \geq \frac{1}{2N(n,c)}$. This implies that we must have $N(n,c) \geq \frac{1}{2}$. Similarly, by setting $P(I|H_{1+}) \leq 1-\delta$, Crowds offers PoI as long as $p_f \geq \frac{\delta}{N(n,c)}$.

Now, assume that we would like to minimize the delay in Crowds, under a given security constraint. Then, since the expected path length is $L = \frac{2-p_f}{1-p_f}$, [9], the smallest value of p_f minimizes the delay. We thus get the following TS function.

$$\begin{aligned} TS(PrI,n,c) &= \tfrac{1}{2N(n,c)} \\ TS(PoI,n,c) &= \tfrac{\delta}{N(n,c)} \\ TS(Ex,n,c) &= 0 \end{aligned} \tag{4}$$

We have assumed that $N(n,c) \neq 0$, otherwise Crowds only offers Ex. In Fig. 4, we have plotted X with respect to p_f for $N = 1$ ($n \to \infty$, c fixed) and $N = 1/2$ when $\delta = \frac{1}{6}$. Note that it is not possible for the system to achieve PrI as N becomes one half.

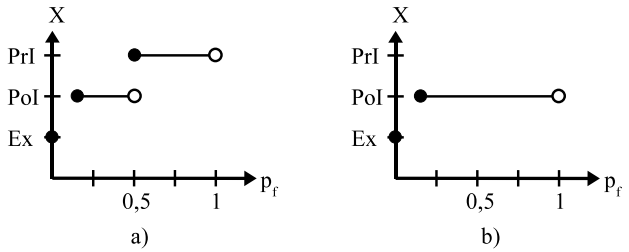

Fig. 4. X when a) $N = 1$ and b) $N = 1/2$.

5 Concluding Remarks

In this paper, the tunable features of Mix-Nets and Crowds have been analyzed, using a previously proposed conceptual model. Both tuner preferences (T) and environmental characteristics (E) that influence the choice of a specific security configuration (S) have been identified. In addition, the mapping to a particular security configuration has been described through a mapping function, which is referred to as the TS function. This implies that the dynamic elements of each service have been identified. A continuation of this work would be to further investigate the dynamic features of Mix-Nets and Crowds, and to examine if additional tunability can be added to the systems.

References

1. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981. `http://www.eskimo.com/~weidai/mix-net.txt`.
2. Common Criteria Implementation Board. Common criteria for information technology security evaluation, version 3.1. http://www.commoncriteriaportal.org/, September 2006.
3. C. Díaz. *Anonymity and Privacy in Electronic Services*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, December 2005.
4. S. Lindskog. *Modeling and Tuning Security from a Quality of Service Perspective*. PhD thesis, Chalmers University of Technology,, Gteborg, Sweden, April 2005.
5. S. Lindskog, A. Brunstrom, and Z. Faigl. Analyzing tunable security services. In *Proceedings of the Third Swedish National Computer Networking Workshop (SNCNW 2006)*, October 26–27, 2006.
6. S. Lindskog, A. Brunstrom, R. Lundin, and Z. Faigl. A conceptual model of tunable security services. In *Proceedings of the 3rd International Symposium on Wireless Communication Systems (ISWCS 2006)*, September 5–8, 2006.
7. U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster Protocol — Version 2, July 2003.
8. A. Pfitzmann and M. Hansen. Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology. Draft, July 2000.
9. M. K. Reiter and A. D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.