Unlinkable Communication

Volker Fusenig Eugen Staab Uli Sorger Thomas Engel
University of Luxembourg
Faculty of Sciences, Technology and Communication
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
Email: {Volker.Fusenig, Eugen.Staab, Ulrich.Sorger, Thomas.Engel}@uni.lu

Abstract

In this paper we present a protocol for unlinkable communication, i.e. where an attacker cannot map the sender and receiver node of a communication. Existing anonymity protocols either do not guarantee unlinkability (e.g. Tor and Mix networks), or produce huge overhead – the dining cryptographers network causes quadratic number of messages. Our protocol needs only a linear number of messages while it still guarantees unlinkability. We introduce a measure of unlinkability and show that our protocol offers the highest possible degree of unlinkability. We show how to use the protocol in practice by adapting it to internet and ad hoc communication.

1 Introduction

A number of techniques were introduced in order to protect the unlinkability of certain data items and users. Mix networks protect the unlinkability of messages entering and leaving a mix [2]. The basic functionality is as follows: The users of a mix send their messages encrypted and with fixed size to the mix. The mix waits until it receives n such messages, decrypts and forwards them in a resorted order to the destination. As result an attacker cannot map the incoming and the outgoing messages only by observing one communication round. However, if he observes a mix for a longer period of time he might gain information that helps to map these communication links. These kinds of attacks are called traffic analysis attacks. In order to complicate traffic analysis Köpsell et al. [9] build paths over several mixes. If a path consists of m mixes where each mix mixes n messages in a round, there are together n^m possible communication partners for a sender node. Still traffic analysis is possible; indeed it is more difficult because the attacker has to identify the correct communication link out of a bigger set of possible communication links.

The more popular protocol TOR [6] builds a path over so called onion routers, which only decrypt the incoming messages and forward them to the next onion router on the path or to the destination. Because there is no mixing at an onion router, a strong passive attacker that is able to observe all the network communication can easily map the communication partners. In fact there have been several successful attacks on TOR (see Bauer et al. [1]).

When using the dining cryptographers network (DC net) technique [3] traffic analysis attacks are not possible. The users of a DC net build groups, where all members of a group share pairwise secret keys that are only used once. Sending of messages is round based, which means that all members have to reserve a time slot anonymously in which the real message is sent. In every round all group members have to publish the sum modulo 2 of all shared secret keys, except node A who is allowed to send a message. He publishes the sum modulo 2 of all secret keys and the message. As result the sum modulo 2 of all published values is the message sent by A. Chaum proved that an attacker does not learn any information by observing the execution of the protocol. The reason why this approach is not used in practice is the high message overhead that is generated by this protocol. To send one message using the DC net protocol, all members of the group have to send one message to all other members of the group. So for a group size of n the protocol generates n^2 messages in order to send one message, where the size of the anonymity set only increases to n.

In this paper we introduce a protocol that also offers provable unlinkability, e.g. where the attacker cannot link the sender and receiver of a communication, but with notably less cost. It bases on layered encryption with fixed path length, where the target node is placed randomly on the path. While using our protocol the message overhead is only linear to the size of the unlinkability set. We use a measure for unlinkability to evaluate the protocol.

The rest of the paper is organized as follows: In Section 2 we give definitions which we will use in the rest of the

978-0-7695-3390-2/08 $25.00 © 2008 IEEE
DOI 10.1109/PST.2008.8

paper. We introduce the protocol in Section 3 and prove the unlinkability. Finally, we conclude our work in Section 4.

2 Prerequisites

In this section we define the terms *anonymity* and *unlinkability* and the attacker model which are used for the evaluation of the protocol.

2.1 Anonymity

In this paper we use the widely accepted definition for *anonymity* of Pfitzmann and Köhntopp [10]:

> Anonymity of a subject means that the subject is not identifiable within a set of subjects, the *anonymity set*.

Anonymity is usually related to an action a. All users that might have performed this action a belong to the corresponding anonymity set, such as the *sender anonymity set* for *sender anonymity* and the *receiver anonymity set* for *receiver anonymity*. Besides from knowing the members of an anonymity set the attacker might have information so that some members are more likely to be the originator of this action than others. These facts lead to the definition of the *degree of anonymity*. We use the information theoretic approach that was independently introduced in [11] and [4]. Both propose to measure the degree of anonymity as entropies whose values depend on the likelihood of the members of the anonymity set for being the originator of an action. The degree of anonymity for a system S with an anonymity set of $A = \{a_0, \cdots, a_n\}$ can be calculated by:

$$H(S) = -\sum_{a_i \in A} P(a_i) \log_2 P(a_i).$$

2.2 Unlinkability

For *unlinkability* we also use the definition of Pfitzmann and Köhntopp [10]:

> Unlinkability of two or more items of interest (IOI) from an attacker's perspective means that within the system, the attacker cannot sufficiently distinguish whether these IOIs are related or not.

The unlinkability set consists of all possible relations, i.e. for network communication of all combinations of sender and receiver nodes. Just as for anonymity one can define how to measure the *degree of unlinkability*. We introduce a simplified version of the definition of Steinbrecher and Köpsell [13], and Franz et al. [7], which is an information theoretic approach.

In the following we concentrate on the term unlinkability in the context of network communication. First we define the set of possible sender nodes as $S = \{s_1, \cdots, s_n\}$ and the set of possible receiver nodes as $R = \{r_1, \cdots, r_m\}$. With this we can fix the communication links where s_i sent a message to r_j as $l_{i,j}$. The resulting set of all possible communication links is $L = \{l_{1,1}, \cdots, l_{n,m}\}$. The degree of unlinkability is defined as the entropy of all possible communication links:

$$H(L) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(l_{i,j}) log_2 P(l_{i,j}).$$

This definition makes clear that the minimum degree of unlinkability of 0 is reached if the probability of one communication link $l_{i,j}$ is one and the probability of all other communication links is 0. In other words the degree of unlinkability is 0 if the attacker can prove that s_i sent a message to r_j. On the other hand the maximum degree of unlinkability is reached, if all communication links are equiprobable. In this case the degree of unlinkability is $\log_2(n \cdot m)$ because there exist $n \cdot m$ communication links.

2.3 Attacker model

We assume a strong passive attacker who is able to observe every communication in the network. The attacker is not able to break cryptographic functions and has no previous knowledge on the used secret and private keys. Furthermore he cannot actively manipulate the operation of the system, e.g. by inserting, altering or replaying messages at any communication link. Also he is not able to take over any participant of the network. We assume as in [12] that the attacker knows the system being used. Such an attacker can easily break the unlinkability of the before mentioned Tor networks, and also has good chances to break the unlinkability of mix networks.

Later we analyze the impacts of active attackers on the system. In this case the attacker can manipulate messages at every communication link and he can control a fraction of all participants. We will show how many collaborating active attackers the protocol tolerates until it fails.

3 Perfect Unlinkability

We assume that the network topology is known by every user. How this can be guaranteed is not part of this work. For example in ad hoc networks this can be done by using a proactive routing protocol.We assume that every node has a pair of public and private keys and also has access to the public keys of all network members. These asymmetric keys are used only for the establishment of secret keys.

We concentrate on two points to offer unlinkability to the users of our system. In the first step (see Section 3.1) we show a way to communicate so that an attacker cannot receive any information on a communication partner by observing sent messages. In the second step (see Section 3.2) we give rules how communication paths must be chosen, so that an attacker also receives no information on the communication partner while observing the complete path a message travels.

3.1 Routing protocol

Our protocol for hiding the target of a message mainly bases on layered encryption. This technique is also used for instance in Onion Routing [6] and Acimn [8]. If a node N_0 wants to send a message m over a path of nodes $N_1, \cdots, N_{t-1}$ to target node N_t, it first builds a key exchange message. The key exchange is important to reduce the computational overhead while using layered encryption. With only one key exchange message N_0 exchanges secret keys with all nodes on the path. In a simplified version of the protocol N_0 builds the key exchange message as follows: It chooses secret keys s_i for all nodes N_i on the path. Then it builds the key exchange message by successively encrypting the secret key and the rest of the message with the public keys PK_i of the nodes on the path. This results in the message $\left(s_1 \left(s_2, \cdots (s_t)_{PK_t} \cdots\right)_{PK_2}\right)_{PK_1}$.

After having exchanged the secret keys node N_0 can send messages over this path by encrypting them successively with the secret keys of all nodes on the path beginning with the target node. N_0 now sends the layered encrypted message $((m_{s_t}) \cdots)_{s_1}$ to N_1. If a node N_i receives such a message it can decrypt one layer and forwards the message to the next node on the path. We do not go into more detail to the functionality of layered encryption. An optimized version of the protocol and more information on the operation of this technique can be found in [8].

To prevent packet volume attacks we use a constant message size for all messages that are sent using our protocol. Nevertheless with our attacker model as introduced in Section 2.3 the attacker is still able to determine the destination of a message, if the destination node does not resend the message to another node. To prevent this we extend the path $N_0, \cdots, N_t$ by some other nodes $N_{t+1}, \cdots, N_{t+e}$. We use a constant length for all paths of $p = t + e$ so that also a strong passive attacker has no chance to obtain any information on the placement of the target node N_t on the path.

With the protocol as described above it is only possible to send messages from N_0 to N_t. To not break the unlinkability, it is not allowed that N_t sends messages directly back to N_0 because in this case a strong passive attacker can easily match those two nodes. To counter this problem we introduce tickets for sending messages back to the sender node N_0. So after sending a request to N_t, N_0 sends a ticket, which is a random message. Because of the layered encryption the attacker cannot determine that this is a ticket for the backward communication. The target node N_t replaces the random message by its encrypted response and forwards it to the next node on the path. The last node N_{t+e} bounces the message back on the same path to N_0. Because of the layered encryption the attacker cannot detect whether and when a ticket is replaced by the response. If the sender node receives such a response message it can decrypt it one after another with the secret keys of the nodes on the path. In the same way we can produce acknowledgments for the receipt of messages.

3.2 Path selection

Up to now we have not yet discussed how the nodes of the path are selected. As we mentioned in Section 2.3 we assume that the attacker knows the algorithm for the path selection. On that score the sender node must choose all nodes of the path in a way such that they are equally likely to be the target node.

When applying the protocol to internet communication this can be done by choosing the nodes on the path randomly. As result we have a degree of unlinkability of $\log_2 n$ if the communication path consists of n nodes.

In ad hoc networks it is also possible to choose random waypoints and route all messages over these nodes. But if there are no direct links between these waypoints the messages must be routed over several other nodes. So the path over n random nodes consists of totally $n + x$ nodes, where x are the nodes to connect the random waypoints. These x nodes do not belong to the unlinkability set, because a strong attacker might detect that they are not randomly located. Even if the attacker can detect this, the protocol guarantees a degree of unlinkability of $\log_2 n$. But the more nodes are involved in the routing process the more increases the cost and the chance for loosing the connection. So by minimizing the involved nodes we can minimize the costs and maximize the robustness.

It is important for the route selection, that all nodes that belong to the unlinkability set have the same likelihood for being the target node. By choosing the shortest path from the sender node N_0 to the target node N_t we can reach this condition. By this also the distances from N_0 to all other nodes N_i, for $1 \leq i < t$, are optimal. To reach the required size for the unlinkability set and also to hide the target node in the path we extend the path over node N_t so that for all following nodes N_i, for $t+1 \leq i \leq t+e$, the resulting path N_0 to N_i is also optimal.

If we assume that the network topology is known to the nodes, the sender node N_0 can easily build the shortest path to the target node N_t by using the Dijkstra algorithm [5].

The Dijkstra algorithm initializes the distance of all nodes to ∞ and the distance for N_0 to 0. All nodes are declared as active. In every step it takes the active node N_i with the shortest distance d_i and path P_i from N_0 to N_i, and checks for all outgoing edges $e(N_i, N_j)$ if $d_i + dist(e(N_i, N_j)) < d_j$. If this is the case it updates the distance d_j to $d_i + dist(e(N_i, N_j))$ and P_j to $P_i \cup N_i$. After having checked all outgoing edges of N_i we declare N_i as passive. The algorithm is repeated until the target node N_t is the active node with the shortest distance which means that P_t is the shortest path from N_0 to N_t.

To extend the path only by nodes N_i, $t < i \leq t + e$, so that the resulting path from N_0 to N_i is optimal we continue the Dijkstra algorithm in such a way, that we mark all active nodes, whose shortest path goes over N_t. The algorithm ends if a marked active node N_i is reached, whose path length l_i is equal to the desired size of the unlinkability set or if there is no more marked active node. In this case we choose the marked passive node N_k, who has the maximum path length l_k. As a result we only have an unlinkability set of l_k nodes. Extending the path by nodes N_i so that the path from N_0 over N_t to N_i is not optimal would not increase the unlinkability set because we assume a strong passive attacker who knows how the path is selected.

3.3 Proof of Unlinkability

To prove the unlinkability of the protocol we have to show that for all possible observations an attacker can make, he gets no information on the communication partner N_t of a sender node N_0. This implies that the attacker is not able to decrease the degree of unlinkability.

If we assume that the nodes on the path are selected such that they are equally likely we can calculate the information that the attacker receives by observation. We define n as the number of nodes in the network and p as the length of the path, over which the sender sends messages to the target node. With these parameters we can calculate how much information a strong passive attacker maximally receives if he knows the complete path P beginning at the sender node N_0 and ending at the last node N_p:

$$I(P; N_t) = \log_2 \frac{P(N_t|P)}{P(N_t)} = \log_2 \frac{\frac{1}{n}}{\frac{1}{p}} = \log_2 \frac{p}{n}.$$

If all nodes of the network belong to the communication path then the attacker receives no information on the target node.

If we use the definition of unlinkability (see Section 2.2) and restrict the unlinkability set to the nodes in P then the probability of every node N_i being the target node is $P(N_i) = \frac{1}{p}$. As result also the degree of unlinkability is $\log_2 p$.

This shows that only knowing the nodes on the path offers no information to the attacker if all nodes are chosen in the same way. This is an important requirement, because if the attacker knows that the nodes are chosen in a different way he can reproduce a path for every possible target node. By this he can limit the unlinkability set. In the worst case the attacker might compromise the target node N_t.

The first thing an attacker can see is the packet size and the content of the packets. In Section 3.1 we assumed a constant packet size s, so the mutual information of s and N_t is $I(s; N_t) = 0$. The same holds for the content of the packets: Because they are layered encrypted the attacker observes packets with different content on every hop. The critical point is when N_t replaces a dummy message in the ticket by its own encrypted response. But because the messages change at every hop the attacker cannot distinguish, whether the message content changed because of the layered encryption or because it was exchanged by an encrypted response. It follows for every given content c_m of a message m holds $I(c_m; N_t) = 0$.

A strong passive attacker can also observe all sending events S_i and receiving events R_i at a node N_i. Because the messages are sent over the complete path, all nodes $N_1, \cdots, N_{t+e}$ send and receive a message. It follows that $P(N_t|S_i) = P(N_t)$ and therefore $I(N_t; S_i) = 0$, respectively $I(N_t; R_i) = 0$ for all $0 \leq i \leq t + e$.

Our protocol is also robust to packet counting attacks. In those attacks the attacker counts the number of incoming and outgoing packets and tries to correlate the sender node N_0 to the receiver node N_t. Because all nodes on the path receive a packet, the attacker is neither able to reduce the size of the unlinkability set nor is he able to adjust the probability for a node N_i for being the target node N_t.

Because during the communication from N_0 to N_t all nodes have to perform the same calculations the computational overhead is the same at every hop. From this it follows that the time between the receiving of a packet and forwarding it to the next node only depends on the computational power of the node and not whether it is the target node or not. For building the response it is different: All nodes except N_t perform a decryption before they forward the message. N_t can prepare the encrypted response message m_r before it receives the ticket because N_0 first sends a request message to N_t. In this case N_t decrypts the ticket message and replaces it with m_r. Because the computations are the same, also the latency between receiving the ticket and forwarding the response is the same compared to forwarding a message by any other node $N_i, i \neq t$. As a result the attacker also gains no information by measuring the latencies at nodes forwarding packets.

If a sender node N_0 communicates frequently with the same target node N_t, N_0 must choose the same path for communication in order to prevent intersection attacks. In

this kind of attacks the attacker intersects the different unlinkability sets and if a node N_i is member of several unlinkability sets he is more likely to be the communication partner of N_0.

3.4 Discussion

As seen in the previous section our protocol offers perfect unlinkability even in the presence of strong passive attackers. The established mix and Tor networks do not fulfill this requirement, because in these networks the attacker has the chance of performing traffic analysis attacks. Approaches which base on the DC net protocol can offer perfect unlinkability but with noticeably higher cost. While our protocol only needs $\Theta(n)$ messages to build an absolute degree of unlinkability of $\log_2 n$, the DC net protocol needs $\Theta(n^2)$ messages. The number of messages only increases linear to the size of the unlinkability set because it only depends on the length of the communication path.

Both, for our protocol and for DC net, it is possible to adjust the size of the unlinkability set. They can increase the unlinkability set maximally to the number of all network participants. Adjusting the unlinkability set is useful to find a tradeoff between performance and degree of unlinkability, and also to adapt the protocols to different scenarios.

In contrast to mix and Tor networks our protocol offers besides unlinkability also perfect receiver anonymity. The DC net protocol also offers receiver anonymity but as seen before at higher costs.

The protocol still offers unlinkability in the presence of active attackers. Active attackers are able to disrupt the execution of the protocol but they are not able to break the unlinkability. If m of the n nodes of the unlinkability set are malicious then the degree of unlinkability decreases to $\log_2(n-m)$. To break the unlinkability the attacker has to control $n-1$ nodes of the unlinkability set. In the same way it is only possible to break the unlinkability in DC nets if the attacker controls $n-1$ nodes of the unlinkability set.

4 Conclusion

We introduced a definition for the degree of unlinkability in order to measure and evaluate the unlinkability a protocol offers. It was shown that the protocol we presented offers perfect and unbreakable unlinkability. Any observation a strong passive attacker can make does not decrease the degree of unlinkability. The overhead generated by the protocol is linear to the size of the unlinkability set in contrast to the well known DC net, which also offers unbreakable unlinkability but has a quadratic message overhead. We have shown how to use the protocol in practice, both for the scenario of internet communication and for communication in ad hoc networks. We plan a prototype implementation in order to analyze the performance during the communication.

References

[1] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against tor. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, 2007.

[2] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[3] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[4] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.

[5] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 21–21, August 2004.

[7] M. Franz, B. Meyer, and A. Pashalidis. Attacking unlinkability: The importance of context. In N. Borisov and P. Golle, editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2007.

[8] V. Fusenig, D. Spiewak, and T. Engel. Acimn: A protocol for anonymous communication in multi hop wireless networks. In L. Brankovic and M. Miller, editors, *Sixth Australasian Information Security Conference (AISC 2008)*, CRPIT.

[9] S. Köpsell, R. Wendolsky, and H. Federrath. Revocable anonymity. In G. Müller, editor, *ETRICS*, volume 3995 of *Lecture Notes in Computer Science*, pages 206–220. Springer, 2006.

[10] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In H. Federrath, editor, *Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 2000.

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

[12] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1948.

[13] S. Steinbrecher and S. Köpsell. Modelling unlinkability. In R. Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, pages 32–47. Springer-Verlag, LNCS 2760, March 2003.