Provable Unlinkability Against Traffic Analysis

Ron Berman, Amos Fiat[⋆], and Amnon Ta-Shma[⋆⋆]

School of Computer Science, Tel-Aviv University.
{bermanro,fiat,amnon}@tau.ac.il

Abstract. Chaum [1, 2] suggested a simple and efficient protocol aimed at providing anonymity in the presence of an adversary watching *all* communication links. Chaum's protocol is known to be insecure. We show that Chaum's protocol becomes secure when the attack model is relaxed and the adversary can control at most 99% of communication links.

Our proof technique is markedly different than previous work. We establish a connection with information theory - a connection we believe is useful also elsewhere, and which we believe supplies the correct language to attack the problem. We introduce "obscurant networks" - networks that can obscure the destination of each particular player, and we show almost all executions of the protocol include such a network.

The security guarantee we supply is very strong. It shows the adversary learns almost no information about any subset of players. Remarkably, we show that this guarantee holds even if the adversary has a-priori information about communication patters (*e.g.*, people tend to speak less with those who do not understand their language). We believe this is an important issue in the real world and is a desirable property any anonymous system should have.

Keywords: Anonymity, Privacy, Traffic Analysis, Unlinkability, Peer to Peer networks.

1 Introduction

Chaum [1, 2] gave a general paradigmatic approach to anonymity. This includes the observation that one can restrict attention to traffic analysis and ignore message content, using encryption as the basic ingredient. These techniques are currently known as onion routing [3, 4]. Chaum also suggested to solve the traffic analysis problem even against an adversary who watches *all* communication links, using a cascade of mixes. Chaum's protocol is flawed and several attacks are known today. In 1993, Rackoff and Simon [5] showed that if all participants play at each time step, then these problems can be solved using secure computation.

The requirement that each participant sends a message every time step, puts a large load burden on the system. Furthermore, if we think of a large peer to

[⋆] This research was supported by a grant from the European Community, Appol II.

[⋆⋆] This research was supported by the Dan David Prize Scholarship.

peer network, say the Internet, then it is inconceivable to require each participant to play each round. Unfortunately, it is not difficult to see that this requirement is necessary if the adversary controls *all* communication links. In this case, if at each time step only a fraction of the participants send a message then the well-known Mix flood attack [12] can isolate messages of any specific player. We therefore set on the task of finding the strongest adversary model, under which we can supply a provably anonymous system, and where the load burden on each player is small.

The model we come up with is one where the adversary can control most, but not all, of the communication links in the system, and the protocol we use is a simplification of Chaum's original protocol. We thus get a simple and efficient protocol (both in terms of delay and load) that is provably anonymous against an all powerful adversary that controls, say, at most 99% of communication links (for formal definitions and statements see Sect. 2). A comparison of our protocol with several other ones can be found in Table 1.

Table 1. Unlinkability protocols for a network of size N. Delay is how long it takes for an anonymous message to arrive after it's been initiated. Load is the number of messages actually sent per anonymous message delivered.

Protocol	Attack Model: Resources under adversary control	Delay	Load	Simple?	Attacks?
Chaum	$O(1)$ fraction of nodes All links	polylog(N)	polylog(N)	Yes	**Yes**
RS93 [5]	$O(1)$ fraction of nodes All links	polylog(N)	$\tilde{O}(N)$	No	—
This paper	$O(1)$ fraction of nodes $O(1)$ fraction of links	polylog(N)	polylog(N)	Yes	—

Our analysis is markedly different than previous work. Relaxing the attack model to one where the adversary does not control a fraction of the communication links makes mixing throughout layers possible. One then has to analyze the information the adversary gets in such a scenario.

Information theory provides a convenient language for expressing and dealing with the question. The notations and definitions used throughout this paper rely heavily on [17]. We show that anonymity can be defined in terms of the mutual information between the actual communication that took place, and the information the adversary knows about it. The mutual information function gives an estimate on how much knowledge can be deduced on one random variable, *e.g.*, the matching of senders and receivers, from another partially correlated variable, *e.g.*, the traffic information gathered by an adversary. We also show that this new definition is equivalent to previous definitions up to small factors.

Using information theory provides us with the language to attack the problem, but not the solution itself. For the proof, we show that with high probability, the information the adversary is missing contains within it communication edges that together form an "obscurant network" - a network that can obscure the destination of each particular player. The exact definition of a protocol execution containing a network is conceptually delicate, and the exact definition, given in Definition 9, is one of the main technical contributions of the paper. We then use information theory to show that this implies that the adversary learns almost no information about any *subset of players*. An alternate formulation of this statement is that the information gleaned by the adversary on the actual communications pattern is close to zero.

An added bonus is the treatment of unlinkability in a scenario where prior information is given to the adversary about the expected communication pattern. We believe this is a rather important issue as in the real world communication patterns are far from being random (*e.g.*, The a-priori probability of a message between two English speaking persons is much larger than that of a message between an English speaking person and a Chinese speaking person). Nevertheless, it seems all previous works avoided the issue. Using our tools, and a nice folding trick (and information theory again, of course), we show that no matter what the prior information is, the adversary learns almost no information from the communication it sees. We believe this result is rather strong and surprising, and is a desirable property any anonymous system should have.

1.1 Related Work

Rackoff and Simon [5] describe a simple protocol secure against passive adversaries (that do not deviate from the given protocol) that is based on sorting networks. Chaum [6] suggested the Dining-Cryptographer networks also secure against such an adversary. Both systems have some extra requirements (*e.g.*, DC require shared secret keys), most notable they both require all players to participate at each stage.

Implementations of Chaum's ideas appear in [13, 14, 4, 3, 8] and various attacks are described in [12, 15]. Other methods for anonymity appear in [7, 10, 11].

2 What is Anonymity?

2.1 Our Attack Model

We have *nodes* and *communication links* in the system. We assume nodes hold data items which are all of the same length. Some nodes and links are under control of an adversary, others are not and are called *honest*. We distinguish between two types of adversaries. An adversary may instruct the nodes and links under his control to perform some arbitrary behavior based on the information he gathered so far. An *adaptive* adversary may instruct nodes and links under his control to initiate arbitrary new messages even not according to the protocol, but may not instruct to delete them. A malicious adversary may instruct such nodes to perform arbitrary behavior and in particular may delete messages. In this paper we only deal with adaptive adversaries.

We assume that a public key infrastructure (PKI) and a public key directory is widely available. The most significant assumption we make is that at least a constant fraction of the communication links are honest[1].

The delay of a protocol, also known as parallel time, is the number of rounds it takes until a message reaches its destination. The load of a protocol is the total number of messages transmitted throughout the protocol per anonymous message delivered. It is important to realize that a communication network in general, and the Internet in particular, may have a very large number N of potential users, while only very few actual active players at any given time. In particular, for an Internet protocol with only $K << N$ active players, one would hope for load that is $\tilde{O}(K)$ and not $\tilde{O}(N)$.

2.2 Defining unlinkability

Say there are M active players and they wish to communicate with M distinct nodes[2]. Let π be the permutation that describes the communication pattern, *i.e.*, player i communicates with node $\pi(i)$, and let $\mathit{\Pi}$ be the random variable whose value is π. Now, let C be the random variable whose value is all the information available to the adaptive adversary, gathered from adaptive communication links and adaptive nodes. Specifically, C is a 0/1 matrix with rows indexed by time steps an columns indexed by edges and with $C_{t,e}$ being 1 iff there is some communication on edge e in time t. Simon and Rackoff require that $(\mathit{\Pi}, C)$ is α–computationally close to some $(\mathit{\Pi}, C')$ such that for all possible permutations $|\pi_1, \pi_2 \ |(C'|\mathit{\Pi} = \pi_1) - (C'|\mathit{\Pi} = \pi_2)|_1 \leq \alpha$. We now give an equivalent definition using the mutual information function. We define:

Definition 1. *Let $A = \{A_n\}, B = \{B_n\}$ be two families of distributions. We say $d(A, B)_P \leq \delta(n)$, if for every family of polynomial-size Boolean circuits $\{T_n\}$, for every large enough n, $| \ \Pr_{x \in A_n}[T_n(x) = 1] - \Pr_{x' \in B_n}[T_n(x') = 1] \ | \leq \delta(n)$.*

The following definition contains three alternative definitions:

Definition 2. *A family $\{(\mathit{\Pi}, C)\} = \bigcup_n (\mathit{\Pi}_n, C_n)$ is $\alpha(n)$–unlinkable if,*

- *$d(\{(\mathit{\Pi}, C)\}, \{(\mathit{\Pi}, C')\})_P \leq \alpha(n)$ for some $\{(\mathit{\Pi}, C')\} = \cup_n (\mathit{\Pi}_n, C'_n)$, and,*
- *For every n, fix $\mathit{\Pi} = \mathit{\Pi}_n$, $C' = C'_n$ and $\alpha = \alpha(n)$. We require,*
  - · *(Def 1 [5]): $\forall \pi_1, \pi_2 \in \mathit{\Pi}$, $|(C'|\mathit{\Pi} = \pi_1) - (C'|\mathit{\Pi} = \pi_2)|_1 \leq \alpha$.*

[1] Our results remain valid even when the adversary is allowed to eavesdrop *every* honest link 99% of the time, with the caveat that on a random 1% of the time, he fails to do so.

[2] If the M nodes are not distinct, then our protocol w.h.p. makes them distinct by adding a random identifier to each message.

· *(Def 2):* $\Pr_{c \in C'}[\ |(\Pi | C' = c) - \Pi|_1 \geq \alpha\ ] \leq \alpha$.
· *(Def 3):* $I(\Pi : C') \leq \alpha$.

We prove the three definitions are equivalent up to small multiplicative factors:

Lemma 3. *Let $\{(\Pi, C)\} = \bigcup_n (\Pi_n, C_n)$ be a family of arbitrary joint distributions, (Π_n, C_n) is distributed over some domain Λ_n.*

- *If $\{(\Pi, C)\}$ is $\gamma(n)$–unlinkable according to Def 1 (Def 2), then it is $\delta(n) = O(\log(|\Lambda_n|)\sqrt{\gamma(n)})$–unlinkable according to Def 3. Conversely, If $\{(\Pi, C)\}$ is $\delta(n)$–unlinkable according to Def 4, then it is $\gamma(n) = (2 \ln 2 \cdot \delta(n))^{1/3}$–unlinkable according to Def 1 (Def 2).*

The formal proof will appear in the full version of the paper.

We now specialize to our case, and we define when a protocol is unlinkable. The thing to notice is that we allow the adversary a-priori knowledge on the honest player's communication pattern. Specifically this means that we do not require the a-priori distribution $\Pi_N(S_N)$ to be uniform. We say a protocol is $\alpha(N)$–unlinkable according to definition i, $i \in \{1, 2, 3\}$, if, for every N players, every choice of subsets S_N of honest players, and every distribution $\Pi_N(S_N)$ on their actual communication, which is the prior knowledge, if we let $C_N(S_N)$ be the correlated random variable that contains the information known to the adversary, then $\bigcup_N (\Pi_N(S_N), C_N(S_N))$ is $\alpha(N)$– unlinkable according to definition i.

We say a protocol P is an *efficient* unlinkable protocol according to definition i, if for every possible error function $\alpha(N) \geq N^{-c}$,

- $P_{N,\alpha(N)}$ is $\alpha(N)$–unlinkable according to definition i, and
- $P_{N,\alpha(N)}$ takes $T(N) = O(\text{poly}(\log(\frac{N}{\alpha(N)})))$ rounds, and $O(M \cdot T(N))$ messages, when M is the number of players who wish to send a message at a time.

Because of the equivalence stated before, we have:

Theorem 4. *A protocol P is efficiently unlinkable according to any one definition iff it is efficiently unlinkable according to all definitions.*

Details of the proof will appear in the full version of the paper.

3 The Protocol

Our protocol is a variant of Chaum's protocol. We describe our protocol in a synchronous system. A wants to send a message $a \in \{0, 1\}^S$ to B and get back an answer $b \in \{0, 1\}^S$, where S is the length of data items in the system. A picks $T - 1$ random nodes $v_1, \ldots, v_{T-1}$, and sets $v_0 = A$, $v_T = B$. A also picks T random strings $r_i \in \{0, 1\}^S$, and $z_i \in \{0, 1\}^{\ell_i}$ where ℓ_i is a security parameter for the encryption schemes E_i. We let $E_1, \ldots, E_T$ be the public encryption methods of the T nodes. We denote

$$a_i = E_{i+1}(r_{i+1}, z_{i+1}, v_{i+2}, E_{i+2}(\dots E_{T-1}(r_{T-1}, z_{T-1}, v_T, E_T(r_T, a))) \dots)$$

for $i = 0, \dots, T-1$.

The way from A to B : A sends $(0, v_0, z_0, a_0)$ to v_1. In general, v_i sends (i, v_i, z_i, a_i) to v_{i+1} where $a_i = E_{i+1}(r_{i+1}, z_{i+1}, v_{i+2}, a_{i+1})$. v_{i+1} then decrypts a_i, and sends $(i+1, v_{i+1}, z_{i+1}, a_{i+1})$ to v_{i+2}. It also records v_i,v_{i+2},z_i, z_{i+1} and r_{i+1}. $v_T = B$ recognizes it is the last on the path, and prepares an answer $b \in \{0,1\}^S$ to the message a it receives.

The way back : $B = v_T$ sends (v_T, z_{T-1}, b_T) to v_{T-1} where $b_T = b \oplus r_T$. In general, v_i receives a message (v_{i+1}, z_i, b_{i+1}). v_i recognizes the value z_i, the link (v_{i-1}, v_i) that precedes (v_i, v_{i+1}) and the values r_i, z_{i-1} that are associated with it. It then sends $(v_i, z_{i-1}, b_i = b_{i+1} \oplus r_i)$ to v_{i-1}. Finally, $A = v_0$ receives $(v_0, z_0, b_0 = b_1 \oplus r_1)$ from v_1. The value $b_0 \oplus r_1 \oplus \dots r_T$ is the desired value b.

We prove:

Theorem 5. *Assume the above protocol runs for T steps in a network with N nodes, $\binom{N}{2}$ communication links, some constant fraction of which are honest, and $T \geq \Omega(\log(N) \log^2(N/\alpha(N)))$. Then the protocol is $\alpha(N)$–unlinkable.*

The protocol can be adapted to the asynchronous setting as well, details to appear in the full version of this paper.

4 The Proof

5 Proof sketch

Generally speaking, in order to prove that the above protocol is secure, a process of structuring is needed to be done to the communication patterns, to allow for easy analysis and calculations.

To perform this process a special communication network is constructed, an "Obscurant Network" (See Sect. 5.1). Apart from the data flow properties of this network that allows anonymity, this network has a highly static and structured communication pattern, compared with the patterns created by our protocol.

In order to analyze the amount of data the adversary gathers from the pattern created by our protocol, we show that this pattern has enough honest links within it that together contain an "embedded" obscurant network. After describing what an embedding is and an algorithm to find one in Sect. 5.2, we prove that our protocol's communication pattern contains, w.h.p., such an embedding in Sect. 5.3 for the case of no-prior information.

Our proof makes use of another interesting technique. During most steps of the analysis, information is purposely being revealed to the adversary regarding communication on links that are not under its control. This classifies the links in the network into two, ones where the adversary has full information of data flow, and ones where the adversary has absolutely no information about the flow of data. Showing both the existence of an embedding of an obscurant network as well as telling all other irrelevant data to the adversary allow for a simple proof in the case of no-prior information.

Prior information is dealt with in section 5.3. A folding trick is used to reveal yet some more information to the adversary about the connection between information flowing from the sources of the message and the information arriving at the final destinations of messages. This trick literally folds the communication pattern in half when observed from the adversary's point of view, reducing the analysis to the case of no-prior information, when the interesting layer of the protocol is the middle layer of communications. We then show that the middle layer does not convey enough information to the adversary, resulting in unlinkability. The result requires longer message paths in order to achieve a probable embedding of an obscurant network.

5.1 Obscurant Networks

A network is a layered directed circuit with the same number of vertices on each layer. We say a circuit is a crossover network, if every vertex has in-degree and out-degree one or two. An example is depicted in Fig. 1. We think of the following game: a pebble is put on some input vertex, say on the i'th vertex. If the vertex out-degree is one, we follow that link. Otherwise, we follow each of the crossover links with probability half. By the end of the game we get a distribution O_i over the output elements. We say the network ϵ–obscures the i'th input, if $|O_i - U_M| \leq \epsilon$, when U_M is the uniform distribution. We say a network ϵ-obscures inputs, if it ϵ-obscures every input. We call networks that obscure their inputs obscurant networks.

We now show an explicit construction of a simple shallow obscurant network that has depth $O((\log(M) + \log \epsilon^{-1}) \log(M))$ for M inputs.

Let Z be the largest power of two not larger than M. We use two components: a butterfly network B_Z, with comparators replaced with crossovers, and a network over $2k$ elements and two layers with k crossovers connecting vertex i in the first layer with both vertex i and vertex $k + i$ $(mod\ 2k)$ in the second layer, for $i = 1, \ldots, 2k$. We call this later network P_{2k}. We distinguish between two cases. If $Z = M$ we put B_Z on the Z inputs. Otherwise, $\frac{M}{2} < Z < M$. For the first level, we put B_Z on the Z rightmost elements. For the second level, We put B_Z on the Z leftmost elements. For the third level, We put $P_{2(M-Z)}$ on the $2(M - Z)$ rightmost elements. For the fourth level, we put B_Z on the Z leftmost elements. We then iterate the third and fourth levels $\log(M) + \log \epsilon^{-1}$ times (see Fig. 1).

We claim:

Lemma 6. *When using a depth of* $O((\log(M) + \log \epsilon^{-1}) \log(M))$*, the network is* ϵ*–obscurant.*

Proof. If $M = Z$, then for every input vertex i spans a tree. It follows that $O_i = U_M$ and the network is 0–obscurant.

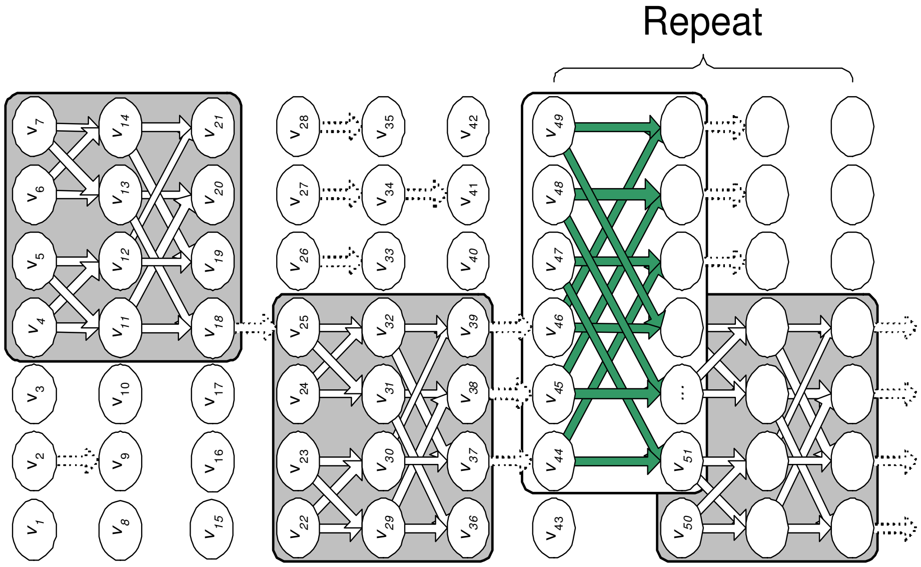

Fig. 1. *An obscurant network for $M = 7$, $Z = 4$. Simple connections are often omitted. The boxes with the grey background are B_4 butterflies. The other boxed sub-circuit is P_6.*

Suppose $\frac{M}{2} < Z < M$. Let i be a starting vertex. Notice that B_Z gives equal weight to each of its Z outputs. When applying B_Z on the right followed by B_Z on the left, all the Z leftmost elements have one weight, ℓ_0, while the rest $M - Z$ rightmost elements have the same (possibly different) weight, r_0. One can observe that this property is an invariant that remains valid throughout the protocol, *i.e.*,

- The invariant: After applying the pair $P_{2(M-Z)}$ and B_Z i times, $i \geq 0$, the Z leftmost elements all have weight l_i and the $M - Z$ rightmost elements all have weight r_i.
- After applying $P_{2(M-Z)}$ of the $i + 1$'st pair, all the $2(M - Z)$ rightmost elements have one weight $r_{i+1} = (r_i + l_i)/2$ and all the remaining $M - 2(M - Z) = 2Z - M = K$, $1 \leq K < Z$, leftmost elements remain at weight l_i.
- After applying B_Z of the $i + 1$'st pair, all the Z leftmost elements have one weight $\ell_{i+1} = \frac{K \cdot l_i + (Z-K) \cdot r_{i+1}}{Z}$ and all the $M - Z$ rightmost elements remain at weight r_{i+1}.

Calculating, we see that $|\ell_{i+1} - r_{i+1}| < \frac{1}{2}|\ell_i - r_i|$, which leads to the fact that $|\ell_t - \frac{1}{M}| \leq \frac{(M-Z)}{M}2^{-t}$. To conclude the proof we note that $|O_i^{(t)} - U_M|_1 \leq 2M \cdot |\ell_t - \frac{1}{M}| \leq 2(M-Z)2^{-t}$. As $M - Z < M/2 < M$, we get that $|O_i^{(t)} - U_M|_1 \leq M2^{-t} \leq \epsilon$, for $t = \log(M) + \log \epsilon^{-1}$.

□

5.2 Finding Obscurant Networks in Protocol Executions

Say that M honest players start sending messages and that we have in mind an obscurant, crossover network G over M inputs and of depth D. Our goal is to show that if the M players run the protocol T steps, for some T large enough, then the network G, in a sense, appears as a subgraph of the protocol execution graph, which we call P. The precise notion of G appearing in P is somewhat delicate and we explain it in detail soon.

The basic fact that we know about our system is that at least an f fraction of the links are honest. The following combinatorial lemma asserts that no matter which edges are honest, if we choose four vertices a, b, c, d at random from V, then there is a crossover structure on the four vertices with probability at least f^4.

Fact 7 *([16], Corollary 2.1) Let $G = (V, E)$ be a graph and assume $|E| \geq f \cdot \binom{|V|}{2}$. Then $\Pr_{a,b,c,d \in V}\ [\ \{(a, c), (a, d), (b, c), (b, d)\} \subseteq E\ ] \geq f^4$.*

Good Embeddings We represent a crossover network G as $G = (V_G, \boxtimes_G, I_G)$ where V_G is the set of DM vertices of G, $\boxtimes_G$ is the set of all crossovers $(a, b; c, d)$ in G, and I_G is the set of all simple links in G (*i.e.*, network edges of G not participating in any crossover). We represent a protocol P as $P = (V_P, T_P, C_P)$ where V_P is the set of TM vertices participating in the protocol, T_P is the set of all links carrying traffic in the execution of the protocol, and C_P is the set of all links that are under adversary control (whether used to carry traffic or not).

Definition 8. *A function $\phi : V_G \times \{0, 1\} \to V_P$ is an embedding if:*

- *The mapping ϕ respects T_P. I.e.,*
  - $\forall_{e=(v,w)\in I_G}\quad (\phi(v, 1), \phi(w, 0)) \in T_P$.
  - $\forall_{(a,b;c,d)\in\boxtimes_G} \left| \left\{ \begin{array}{l} (\phi(a,1), \phi(c,0)),\ (\phi(a,1), \phi(d,0)), \\ (\phi(b,1), \phi(c,0)),\ (\phi(b,1), \phi(d,0)) \end{array} \right\} \cap T_P \right| = 2.$
- *For every $v \in V_G$, $\phi(v, 0)$ and $\phi(v, 1)$ are connected in T_P.*
- *The adversary does not know any link in any crossover. I.e., for every $(v_1, v_2; w_1, w_2) \in \boxtimes_G$ and every $i, j \in \{1, 2\}$, $(\phi(v_i, 1), \phi(w_j, 0)) \notin C_P$.*

We define $\phi_P(\boxtimes_G)$ to be the image of $\boxtimes_G$ under the embedding ϕ. *I.e.*, the set of all $(u_1, u_2; u_3, u_4) \in V_P^4$ for which there exist $(v_1, v_2; v_3, v_4) \in \boxtimes_G$ s.t. $\phi(v_1, 1) = u_1$, $\phi(v_2, 1) = u_2$, $\phi(v_3, 0) = u_3$ and $\phi(v_4, 0) = u_4$.

A delicate point is that right now the embedding ϕ may depend on the actual communication P that took place. We therefore add the requirement that ϕ is independent of the communication that took place on the embedded copy of G. Formally this takes the following form:

Definition 9. *Let G be defined as before. Let $\mathcal{P}$ be a protocol (e.g., the protocol of Sect. 3). An embedding strategy for the protocol, with ϵ error, is an algorithm that given an execution $P = (V_P, T_P, C_P)$ of the protocol, outputs a function $\phi_P : V_G \times \{0, 1\} \to V_P$ such that:*

- $\Pr_{coins\ of\ P}[\phi_P \textit{ is an embedding}] \geq 1 - \epsilon$, *and,*
- *For every two protocol executions P and P' that use the same sets of vertices, if $T_{P'}$ agrees with T_P on all edges not participating in $\phi_P(\boxtimes_G)$ then $\phi_P = \phi_{P'}$.*

A Good Embedding Exists We prove:

Lemma 10. *Let G be any network over M inputs and of depth D. Let us run the protocol $\mathcal{P}$ of Sect. 3 for $T = 2Dk$ steps. Then there is an embedding strategy for $\mathcal{P}$ with $\epsilon = DM(1-f^4)^k$ error.*

Proof. We first label each vertex v of G by $v_i^{(d)}$, where $0 \leq d \leq D$ is the depth of v, and i comes from an arbitrary labelling of the d'th layer with labels $\{1, \ldots, M\}$, such that all edges in G are either of the simple form $(v_i^{(d)}, v_i^{(d+1)})$ or of the crossover form $(v_i^{(d)}, v_j^{(d+1)})$ where $(v_i^{(d)}, v_j^{(d)}; v_i^{(d+1)}, v_j^{(d+1)}) \in \boxtimes_G$.

Algorithm 11 *(An algorithm for labelling V_P and constructing $\phi : V_G \to V_P$)*

Bottom layer *: The algorithm labels all the vertices in the bottom layer of P with $u_i^{(0)}$, where $i \in \{1, \ldots, M\}$ and the labels inside the layer are chosen arbitrarily (say, by lexicographic order on the identity of the vertex). We define $\phi(v_i^{(0)}, 0) = u_i^{(0)}$.*

Odd layer *: We reveal all communication on links going from a vertex in layer $2t$ to a vertex in layer $2t+1$, for every $0 \leq t \leq \frac{T}{2}$. For every revealed edge $(u_i^{(2t)}, w) \in T_P$ we label w with $u_i^{(2t+1)}$.*

Even layer $t = d \cdot 2k + \ell$**,** $0 \leq \ell < 2k$ *:*

First, if $\ell = 0$ we set $ok(i) = false$ for every $i \in \{1, \ldots, M\}$. This tells us that we still have to take care of all vertices in the d'th layer of G. Otherwise, if $\ell \geq 2$ then for every $i \in \{1, \ldots, M\}$ we do the following:

- *If $v_i^{(d)}$ belongs to a simple edge $(v_i^{(d)}, v_i^{(d+1)})$, we reveal the edge $(u_i^{(t-1)}, w)$ of T_P and we label w with $u_i^{(t)}$. Also, if $ok(i) = false$ then set $\phi(v_i^{(d)}, 1) = u_i^{(t-1)}$, $\phi(v_i^{(d+1)}, 0) = u_i^{(t)}$ and $ok(i) = true$.*
- *Otherwise, $v_i^{(d)}$ belongs to a crossover form $(v_i^{(d)}, v_j^{(d)}; v_i^{(d+1)}, v_j^{(d+1)}) \in \boxtimes_G$. Let w and z be the vertices such that $(u_i^{(t-1)}, w), (u_j^{(t-1)}, z) \in T_P$.*

  If $ok(i) = true$ or if one of the edges $(u_i^{(t-1)}, w)$, $(u_i^{(t-1)}, z)$, $(u_j^{(t-1)}, w)$ or $(u_j^{(t-1)}, z)$ is in C_P we reveal all the above four edges. We also label w with $u_i^{(t)}$.

  If, however, $ok(i) = false$ and all these four edges are honest, we label $\{w, z\}$ with the labels $\left\{u_i^{(t)}, u_j^{(t)}\right\}$ in an arbitrary order (say, by the natural order on i and j as numbers) and we set $\phi(v_i^{(d)}, 1) = u_i^{(t-1)}$, $\phi(v_i^{(d+1)}, 0) = u_i^{(t)}$ and $ok(i) = true$. We also say, then, that we have found the crossover $(v_i^{(d)}, v_j^{(d)}; v_i^{(d+1)}, v_j^{(d+1)}) \in \boxtimes_G$ in P.

The first two conditions of Definition 8 hold directly from the way we choose the embedding ϕ. Also, let us say that we find G in P if we find every crossover of $\boxtimes_G$ in P. Whenever this happens the third condition also holds, because we then embed every crossover of G in V_P in a clean way.

To see that Algorithm 11 is an embedding strategy, fix two executions P and P' of the protocol that use the same sets of vertices, and that agree on all communication over links not in $\phi_P(\boxtimes_G)$. As P and P' differ only on crossovers, and the labelling of the vertices at the last layer of the crossover depends only on a pre-determined order, the labelling in P is the same as in P'. This means that $\phi_P = \phi_{P'}$.

To complete the argument we show that with high probability (over the random coins of the protocol $\mathcal{P}$ from Sect. 3) we find all crossovers of $\boxtimes_G$ in P.

Claim. For every crossover $(v_i^{(d)}, v_j^{(d)}; v_i^{(d+1)}, v_j^{(d+1)}) \in \boxtimes_G$, the probability we do not find it in an execution P of the protocol from Sect. 3 is at most $(1-f^4)^k$.

Proof. Fix $(v_i^{(d)}, v_j^{(d)}; v_i^{(d+1)}, v_j^{(d+1)}) \in \boxtimes_G$. For every time step $t = 2kd + \ell$, $2 \leq \ell < 2k$, look at the vertices $u_i^{(t-1)}, u_j^{(t-1)}, u_i^{(t)}, u_j^{(t)}$. The vertices in each path are chosen at random, and we reveal all edges going from even layers to odd layers. Thus, the vertices in the $t-1$ and t'th layers are chosen at random and independent of history. Specifically, the above four vertices are chosen at random, and independent of history. By Fact 7 we find a crossover with probability at least f^4. As different steps are independent, the probability we do not find a crossover in any of the k attempts is at most $(1-f^4)^k$. □

Using the union bound we see that:

Claim. Let G be any crossover network with M inputs and depth D. Let us run the protocol of Sect. 3 with M honest nodes and for $T = 2Dk$ steps. Let P be the resulting network. Then $\Pr\ [G \text{ does not appear in } P] \leq DM(1-f^4)^k$.

□

5.3 The Unlinkability Proof

Our goal now is to prove that our protocol is unlinkable. We first deal with the no prior knowledge case, *i.e.*, when the a-priori distribution is uniform. We then show in section how the no prior knowledge case implies the general case.

We show that given knowledge of how players $1, \ldots, j$ behave, the adversary does not know how player $j+1$ behaves. For every $j = 1, \ldots, M$, we display a *different* obscurant network G_j, over $M - j$ players, in the actual execution of the protocol.

Suppose there are M honest players sending messages in a network with N players, and let $\alpha(N) > N^{-c}$. Let $G = G_M$ be an ϵ-obscurant network over M inputs and of depth $D = O(\log(\frac{M}{\epsilon})\log(M))$. Suppose we run the protocol for $T = 2Dk$ steps. We would like to set values for ϵ and k such that we receive $\alpha(N) - \mathit{unlinkability}$ with our protocol.

We define the following random variables:

X : X contains all the actual information generated throughout the protocol. *I.e.*, for every link $(v_i^{(t)}, v_j^{(t+1)})$ it contains the information whether there was traffic on that link or not.

Π : $\Pi(i)$ contains the actual destination of the i'th honest player. The random variable $\Pi = \Pi(1) \dots \Pi(M)$ contains the actual communication pattern between the M honest players and the M destinations.

C' : C' contains all the traffic information the adversary knows. *I.e.*, for every dishonest link $(v_i^{(t)}, v_j^{(t+1)})$ it contains the information whether there was traffic on that link during the t'th step or not.

Z : X and C' together determine whether the process described in section 5.2 finds the crossover network G in the protocol or not. If we do, we let Z contain all the information available on links that do not belong to $\phi_{X,C'}(\boxtimes_G)$. *I.e.*, for every link $(v_i^{(t)}, v_j^{(t+1)})$ that does not belong to $\phi_{X,C'}(\boxtimes_G)$, it contains the information as to whether there was traffic on that link during the t'th step or not.

Notice that Z is correlated with X, Π and C'. Nevertheless, the chain rule for information ([17], Theorem 2.5.2, page 22) tells us that $I(\Pi : C') \leq I(\Pi : C', Z)$. It would therefore suffice to show that $I(\Pi : C', Z) \leq \alpha(N)$. Now comes the crux of the argument, and we do it in detail.

Suppose the embedding strategy finds G in an execution P of the protocol. By Definition 9, all executions P' of the protocol that use the same set of vertices and agree with P outside $\phi_P(\boxtimes_G)$ result in the same embedding. As all edges revealed are outside $\phi_P(\boxtimes_G)$, the random variable Z has the same value in both cases. Also, C' has the same value in both cases as $\phi_P(\boxtimes_G)$ contains only honest edges. Thus, the adversary can not distinguish P from P'. As the a-priori probabilities of the executions P and P' are the same, both are equally likely from the adversary point of view. *I.e.*, any possible communication pattern on $\phi_P(\boxtimes_G)$ is equally likely.

Now, G is an ϵ-obscurant network. From the adversary point of view, any crossover is resolved to be identity with probability half, and a switch with probability half (because all possible communication patterns are equally likely), and so by the obscurant network properties $|(\Pi(1) \mid C', Z) - U_M|_1 \leq \epsilon$.

Using lemma 10 it follows that $\Pr_{c',z}[|(\Pi(1)|C' = c', Z = z) - U_M|_1 \geq \epsilon] \leq DM(1 - f^4)^k = \epsilon$, when k is set to $log_{\frac{1}{1-f^4}}(\frac{DM}{\epsilon})$.

We now continue with standard manipulations. From Lemma 3 we see that $I(\Pi(1) : C', Z) \leq log(|\Lambda_N|) \cdot \sqrt{\epsilon} = O(TM^2\sqrt{\epsilon})$. Taking $\epsilon = \frac{\alpha^6(N)}{M^{12}}$, we receive $I(\Pi(1) : C', Z) \leq O(\frac{\alpha(N)}{M})$.

Using the chain rule for information, $I(C' : \Pi) =$
$I(C' : \Pi(1)) + I(C' : \Pi(2)|\Pi(1)) + \ldots + I(C' : \Pi(M)|\Pi(1), \ldots, \Pi(M-1))$.

We can bound the j'th term $I(C' : \Pi(j) \mid \Pi(j-1), \ldots, \Pi(1))$ in this equation, by adding to the adversary the knowledge of the communication paths of the first $j-1$ players. We then see that we get a new game with only $M-j+1$ players. Our analysis from before shows that $I(C', Z : \Pi(j) \mid \Pi(j-1), \ldots, \Pi(1)) \leq O(\frac{\alpha(N)}{M})$. We therefore conclude that $I(C' : \Pi) \leq M \cdot O(\frac{\alpha(N)}{M}) \leq \alpha(N)$ as desired.

The Prior Information Case In the general case the adversary knows that the actual communication that took place has a-priori distribution Π. The adversary may use this knowledge to deduce things about the next to last layer, the one preceding it and so forth. Thus, the information the adversary sees flows both from bottom up (because the adversary knows who initiates messages, and follows whatever links he can), and from top down (because the adversary has some partial information about who sent who a message, and he follows links from top down). We note that we would like to deal with priors that have extremely low probability in a uniform world. *E.g.*, the adversary might know that residents of Kandahar tend to communicate with residents of Karachi.

The way we show our protocol works is by concentrating on the *middle* layer. This is intuitively natural because the adversary knows the permutation at the beginning, and has partial information about the final permutation (given by the prior), but the middle layer seems to be masked by the random choices made throughout the protocol. We let $\Pi^{(T/2)}$ be the random variable whose value is the actual permutation that took place between the first and middle layer. To show that even in the prior knowledge scenario the adversary does not learn much about the middle layer we give the adversary additional information so as to make the information flow only in one direction. Details follow.

Lemma 12. *Let Π be an arbitrary distribution. Suppose we run the protocol for $T = \Omega(\log(M) \log^2(\frac{M}{\alpha}))$ steps. Then $I(C' : \Pi^{(T/2)}) \leq \alpha$.*

Proof. We say a vertex $v^{(t)}$ from the t'th layer is associated with a vertex $w^{(T-t)}$ from the $T-t$'th layer, if the message that $v^{(t)}$ forwards eventually arrives at $w^{(T-t)}$. We also say the link (w, v) is associated with the link (v', w') if w is associated with w', and v is associated with v'.

We give the adversary the extra knowledge about which vertex at level t is associated with which vertex at level $T-t$, for every $0 \leq t \leq \frac{T}{2}$. We see that under this additional information the adversary gets to see M players playing our protocol for $T/2$ steps, and where a link $(v^{(t)}, v^{(t+1)})$ is honest iff both the link $(v^{(t)}, v^{(t+1)})$ and its associated link are honest.

Thus, the only difference from the case of no prior knowledge is that now the probability each link is honest is f^2 rather than f. We therefore can use the theorem for no prior-knowledge and conclude that $I(C' : \Pi^{(T/2)}) \leq \alpha$ as desired. □

We now show that it must be the case that the adversary did not gain much information about the last layer. *I.e.*,

Lemma 13. $I(C' : \Pi^{(T)}) \leq I(C' : \Pi^{(T/2)})$.

Proof. We represent the random variable C' that contains the communication the adversary sees as $C' = (C_1, C_2)$ where C_1 is the communication seen throughout the first $T/2$ steps, and C_2 is the communication seen throughout the last $T/2$ steps.

$$I(\Pi^{(T)} \ : \ C_1, C_2) = I(\Pi^{(T)} \ : \ C_2) \ + \ I(\Pi^{(T)} \ : \ C_1 \mid C_2) \ =$$
$$I(\Pi^{(T)} \ : \ C_1 \mid C_2) \leq I(\Pi^{(T/2)} \ : \ C_1 \mid C_2) \ \leq \ I(\Pi^{(T/2)} \ : \ C_1, C_2)$$

The first equality and the last inequality are applications of the chain rule for information.

To see the second equality, notice that $(C_2 | \Pi^{(T)} = \pi)$ is the same distribution for all permutations π that are valid values of $\Pi^{(T)}$. This is because we can think of the protocol as if the players first pick $\pi \in \Pi^{(T)}$, then pick the top $T-1$ levels at random, and then complete the first layer to implement π. Thus, $I(\Pi^{(T)} : C_2) = 0$.

The crux of the argument is the first inequality. For it, we use the data-processing inequality ([17], Theorem 2.8.1, page 32) and the probabilistic function $f(\sigma, c_2)$ that given $\sigma \in \Pi^{(T/2)}$ and $c_2 \in C_2$ chooses the permutation π with probability $\Pr(\Pi^{(T)} = \pi \mid \Pi^{(T/2)} = \sigma \wedge C_2 = c_2)$. The important thing to notice is that it suffices to know σ and c_2 alone to know the value of $f(\sigma, c_2)$. □

6 Open Problems

We show an efficient protocol (both in terms of delay and load) secure against adaptive adversaries. However, in our opinion, this is only the beginning of a systematic study of unlinkability in anonymous networks. We mention a few interesting open problems:

- Our work (and most previous work) assume a complete communication network. In reality, the network is a low-degree graph. Simple calculations show that an adaptive adversary can easily isolate all messages that come from any specific user. Is there a reasonable relaxed attack model, that allows anonymous communication?
- Our work (and most previous work) assume the communication network (*i.e.*, the vertices in the network, and which vertices and edges are honest) is fixed in advance. Can one design a protocol that handles dynamic changes in the topology (users joining and leaving) of the system?
- Our work (and most previous work) assumes each participant has full knowledge of the network topology, users' keys, etc. This does not conform, for example, with the fully distributed nature of peer to peer systems. Can we do better in this respect, and still retain efficiency and provable security?
- Extending the protocol to malicious adversaries.

7 Acknowledgements

We thank Benny Chor for enlightening discussions, and for insisting on prior knowledge. We thank the vibrant Hebrew University theory seminar for many important comments. We are especially indebted to Yonatan Bilu for pointing out, during the seminar, a fundamental mistake in an earlier version of the paper.

References

1. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Thesis (M.S. in Computer Science), University of California, Berkeley, Berkeley, CA, USA (1979)
2. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the Association for Computing Machinery **24** (1981) 84–88
3. Reed, M.G., Syverson, P.F., Goldschlag, D.M.: Anonymous connections and onion routing. IEEE Journal on Selected Areas in Communications **16** (1998) 482–494
4. Syverson, P.F., Goldschlag, D.M., Reed, M.G.: Anonymous connections and onion routing. In: 1997 IEEE Symposium on Security and Privacy. (1997) 44–54
5. Rackoff, C., Simon, D.R.: Cryptographic defense against traffic analysis. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing, San Diego, California (1993) 672–681
6. Chaum, D.: The Dining Cryptographers Problem: Unconditional sender and recipient untraceability. Journal of Cryptology **1** (1988) 65–75
7. Reiter, M.K., Rubin, A.D.: Crowds: anonymity for Web transactions. ACM Transactions on Information and System Security **1** (1998) 66–92
8. Abe, M.: Mix-networks on permutation networks. In: Advances in Cryptology - ASIACRYPT '99, International Conference on the Theory and Applications of Cryptology and Information Security, Singapore, November 14-18, 1999, Proceedings. Volume 1716 of Lecture Notes in Computer Science. (1999) 258–273
9. Abe, M., Hoshino, F.: Remarks on mix-network based on permutation networks. Lecture Notes in Computer Science **1992** (2001) 317–324
10. Malkhi, D., Pavlov, E.: Anonymity without 'cryptography' (extended abstract). In: FC: International Conference on Financial Cryptography, LNCS, Springer-Verlag (2001)
11. Beimel, Dolev: Buses for anonymous message delivery. JCRYPTOL: Journal of Cryptology **16** (2003)
12. Raymond, J.F.: Traffic analysis: Protocols, attacks, design issues, and open problems. Lecture Notes in Computer Science **2009** (2001) 10–29
13. : The anonymizer. (http://anonymizer.com)
14. : Anonymous remailer information. (http://anon.efga.org/Remailers.)
15. Federrath, H., ed.: Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA, July 25-26, 2000, Proceedings. In Federrath, H., ed.: International Workshop on Design Issues in Anonymity and Unobservability. Volume 2009 of Lecture Notes in Computer Science., Springer (2001)
16. Alon, N.: Testing subgraphs in large graphs. In: 42nd IEEE Symposium on Foundations of Computer Science. (2001) 434–439
17. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA (1991)
18. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge (2000)