SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.539

SPECIAL ISSUE PAPER

Anonymous communication in overlay networks

Jian Ren[1]*, Yun Li[2], Tingting Jiang[3] and Tongtong Li[1]

[1] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, U.S.A.
[2] Microsoft, Redmond, WA 98052, U.S.A.
[3] Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, U.S.A.

ABSTRACT

Communication anonymity is becoming an increasingly important, or even indispensable, security requirement for many applications. The existing research in anonymous communications can largely be divided into two categories: mix-based systems and secure multiparty computation-based systems, originating from mixnet and DC-net, respectively. However, they either cannot provide provable anonymity or suffer from transmission collision problem. In this paper, we first propose a novel unconditionally secure source anonymous message authentication code that can be applied to any messages without relying on any trusted third parties. While ensuring message sender anonymity, secure source anonymous message authentication code can also provide message content authenticity. We then propose a novel communication protocol that can hide the senders and the recipients from each other and thus can be used for secure file sharing. The security analysis demonstrates that the proposed protocol is secure against various attacks. Our analysis also shows that it is efficient and practical.

KEYWORDS
communication anonymity; source anonymous message authentication code; unconditional security; sender anonymity; recipient anonymity; location privacy; overlay network

***Correspondence**
Jian Ren (GE), Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 95134, U.S.A.
E-mail: renjian@egr.msu.edu

1. INTRODUCTION

The rapid growth of public acceptance of the Internet as a means of communication and information dissemination has made communication privacy an increasingly important requirement for many network applications. Although end-to-end encryption protects the data content of communications from adversarial access, it does not conceal all the relevant information that two users are communicating. Adversaries can still learn not only the network of the sender and receiver but also the network addresses of its end-to-end source and destination.

In many situations, it is highly desirable or indispensable for users to keep their communications anonymous. In other words, anonymity is no longer a feature but a fundamental security requirement for many applications. For example, a customer placing an online order may not want his or her transactions to be traced. For applications such as e-voting, e-cash, and so on, anonymity is a fundamental requirement. As another example, if the item ordered by this person can be delivered electronically, he or she may not want his or her destination address to be identified.

Over the last years, overlay networks have evolved as a natural decentralized way to share data and services among a network of loosely connected components. This proliferation of overlay networks has also been propelled by popular applications, most notably secure file sharing and Internet Protocol telephony (e.g., Gnutella, BitTorrent, and Skype). People seeking for sensitive information have a strong desire to remain anonymous so as to avoid being stigmatized or even to avoid physical or social detriment by suppressors. The freedom of information exchange is another important issue that obtained increasing attention in the last years. Some organizations, such as governments or private companies, may regard a discussion topic or a report as inconvenient or even harmful. They may thus try to censor the exchange of undesired information by either suppressing resource providers or if these are protected by anonymity, taking control of strategic regions of the network, such as gateways and proxies, and filtering the communication.

Without anonymity, there are abundant opportunities for passive eavesdropping on data communications. The exposure of network addresses may result in a number of severe consequences. Adversaries can easily overhear all the messages and perform traffic analysis. In a tactical military communication network, an abrupt change in traffic pattern may indicate some forthcoming activities. This could be extremely dangerous in that adversaries can easily identify critical network nodes and then launch directed attacks on them.

In the past two decades, originated largely from Chaum's mixnet [1] and DC-net [2], a number of anonymous communication protocols (e.g., [1,3–10]) have been proposed. The mixnet family protocols (e.g., [6,9–12]) use a set of "mix" servers that mix the received packets to make the communication path (including the sender and the recipient) ambiguous. They rely on the statistical properties of background traffic that is also referred to as the *cover traffic* to achieve the desired anonymity. The security of mixnet is based on the trust relationship of the mixers and cannot provide provable anonymity. The DC-net family protocols (e.g., [2,3,7,13]) utilize secure multiparty computation techniques. They provide provable anonymity without relying on trusted third parties. However, they suffer from the transmission collision problem that does not have a practical solution [7].

As the computing, communicating, and cryptographic techniques progress rapidly, increasing emphasis has been placed on developing efficient and unconditionally secure anonymous communications schemes for overlay networks without relying on trusted third parties and free of collision.

In this paper, we first propose a novel unconditionally secure and efficient source anonymous message authentication code (SA-MAC) for any messages without relying on any trusted third parties. While ensuring message sender anonymity, it can also provide message content authenticity. We then propose a novel communication protocol that can hide the senders and the receivers from each other and thus can be used for secure file sharing. In the proposed protocol, the participants are referred as the *nodes* and are organized into a set of generalized overlay network rings over the Internet. The nodes are further classified into *normal nodes* and *super nodes*. A normal node is a network node that has no direct connection to the nodes in other network rings. A super node can be a normal node that can also provide message forward services to the other networks. It can also be a special node dedicated to providing message forward services to the other networks. Each network should have many normal nodes and multiple super nodes. We then propose a novel communication protocol that can hide the senders and the recipients and their relationship from traffic analysis [14] and thus can be used for secure file sharing. The security analysis demonstrates that the proposed protocol is secure against various attacks. Our analysis also shows that it is efficient and practical.

The rest of this paper is organized as follows. In Section **2**, we briefly review the terminology, assumptions, and some of the previous works related to this paper. In Section **3**, we describe the proposed unconditionally secure SA-MAC and the security analysis. In Section **4**, we propose an anonymous communication protocol in detail along with security analysis and efficiency evaluation. Finally, in Section **5**, we conclude this paper.

2. TERMINOLOGY AND PRELIMINARY

In this section, we will briefly describe the terminology of anonymity defined in previous research. Then, we will introduce some cryptographic tools that will be used in this paper. Finally, we will present a brief overview of the related works in this area.

2.1. Terminology

The concept of anonymity in information management has been discussed in a number of previous works [1,2,10,15–17]. Anonymity generally refers to the state of being not identifiable within a set of subjects. Primarily, three types of anonymity or anonymous communication properties were defined in [15]: *sender anonymity*, *recipient anonymity*, and *relationship anonymity*. *Sender anonymity* means that a particular message is not linkable to any sender and no message is linkable to a particular sender. *Recipient anonymity* similarly means that a message cannot be linked to any recipient and that no message is linkable to a recipient. *Relationship anonymity* means that the sender and the recipient are unlinkable. In other words, sender and recipient cannot be identified as communicating with each other, though it may be clear that they are participating in some communications. Relationship anonymity is a weaker property than each of the sender anonymity and recipient anonymity. The aforementioned anonymities are also referred to as the *full anonymities* because they guarantee that an adversary cannot infer anything about the sender, the recipient, or the communication relationship from a transmitted message.

We will start with the definition of SA-MAC.

Definition 1 (SA-MAC)
A *SA-MAC* scheme consists of the following two algorithms:

- *Generate* $(m, y_1, y_2, \ldots, y_n)$: Given a message m and the public keys $y_1, y_2, \ldots, y_n$ of the anonymity set $S = \{A_1, A_2, \ldots, A_n\}$, the actual message sender $A_t, 1 \leq t \leq n$ can produce a SA-MAC code $S(m)$ by using her own private key x_t.
- *Verify* $S(m)$: Given a message m and a SA-MAC $S(m)$, which includes the public keys of all members in the AS, a verifier can determine whether $S(m)$ is a valid SA-MAC generated by a member in the AS.

The security requirements for SA-MAC schemes include the following:

- *Sender ambiguity*: The probability that a verifier successfully determines the real sender of a SA-MAC is exactly 1/*n*, where *n* is the total number of AS.
- *Unforgeability*: A SA-MAC is unforgeable if no adversary, given the public keys of all members of the AS and the SA-MACs for messages $m_1, m_2, \ldots, m_l$ adaptively chosen by the adversary, can produce in polynomial time a new valid SA-MAC with nonnegligible probability.

In this paper, the user ID and user public key will be used interchangeably without making any distinction.

2.2. Modified ElGamal signature (MES) scheme

Definition 2 (MES)
The MES scheme [18] consists of the following three algorithms:

- *Key generation algorithm*: The signer chooses a random large prime p and a generator g of $\mathbb{Z}_p^*$. Both p and g are made public. Then, for a random private key $x \in \mathbb{Z}_p$, the public key y is computed from $y = g^x \bmod p$.
- *Signature algorithm*: Similar to the ElGamal signature scheme [19,20], the MES can also have many variants. For the purpose of efficiency, in this paper, we will use a signature variant of ElGamal signature scheme, called *optimal scheme* [19,20]. To sign a message m, one has to choose a random $k \in \mathbb{Z}_{p-1}^*$ and then compute the exponentiation $r = g^k \bmod p$ and solve s from

$$s = rxh(m, r) + k \bmod (p - 1) \qquad (1)$$

where h is a one-way hash function.

The algorithm finally outputs the signature (r, s) of message m.

- *Verification algorithm*: The verifier checks whether the signature equation

$$g^s = ry^{rh(m,r)} \bmod p$$

is true. If the equality holds true, then the verifier “Accepts” the signature and “Rejects” otherwise.

2.3. Previous work

The existing anonymous communication protocols are largely stemmed from either mixnet [1] or DC-net [2]. A mixnet provides anonymity via packet reshuffling through (at least one trusted) “mix.” In a mixnet, a sender encrypts an outgoing message and the ID of recipient by using the public key of the mix. The mix accumulates a batch of encrypted messages, decrypts and reorders these messages, and forwards them to the recipients. An eavesdropper cannot link a decrypted output message with any particular (encrypted) input message. The mixnet thus protects the secrecy of users’ communication relationships. Recently, Möler presented a provably secure public-key encryption algorithm for mixnet [8]. This algorithm has been adopted by Mixminion [6]. However, because mixnet-like protocols rely on the statistical properties of background traffic, they cannot provide provable anonymity. Wright *et al*. [21] showed the degradation of anonymity of some protocols in the face of persistence attackers.

DC-net [2,17] is an anonymous multiparty computation amongst a set of participants; some pairs of which share secret keys. DC-net provides perfect (information theoretic) sender anonymity without requiring trusted servers. In a DC-net, users send encrypted broadcasts to the entire group, thus achieving receiver anonymity. However, all members of the group are made aware of when a message is sent, so DC-net does not have the same level of sender–receiver anonymity. Also, in DC-net, only one user can send at a time, so it takes additional bandwidth to handle collisions and contention. Lastly, a DC-net participant fixes its anonymity versus bandwidth trade-off when joining the system, and there are no provisions to rescale that trade-off when others join the system.

Crowds [10] extends the idea of anonymizer and is designed for anonymous web browsing. However, Crowds only provides sender anonymity. It does not hide the receivers and the packet content from the nodes *en route*. Hordes [22] is built on the Crowds. It uses multicast services and provides only sender anonymity. The *k*-anonymous communication protocol for overlay network introduced in [23] can provide both sender and recipient anonymity; however, the initialization and key chain distribution are quite complex. The communication overhead is also high. Motivated by the buses, a new interesting idea is proposed in [4] to hide the message senders and the message receivers. Unfortunately, the proposed approach has very limited flexibility. In addition, it primarily focuses on the theoretic performance analysis. The protocol was not clearly presented in this paper.

Recently, message sender anonymity based on ring signatures was introduced [24]. This approach can enable message sender to generate source anonymous message signature with contents authenticity assurance, while hiding the real identity of the message sender. The major idea is that the message sender (say Alice) randomly selects n ring members as the AS on her own without awareness of these members. To generate a ring signature, there are n trapdoor one-way functions involved. For each member in the ring other than the actual sender (Alice), Alice randomly selects an input and computes the one-way output by using message signature forgery. For the trapdoor one-way function corresponding to the actual sender Alice, she needs to solve the “message” that can “glue” the ring together and then sign this “message” using her knowledge of the trapdoor information. The original scheme has very limited flexibility, and the complexity of the scheme is quite high. Moreover, the original paper only focuses on the cryptographic algorithm; the relevant network issues were totally left unaddressed.

In this paper, we first propose an unconditionally secure and efficient source anonymous message authentication scheme on the basis of the MES scheme. This is because the original ElGamal signature scheme is existentially forgeable with a generic message attack [25,26], whereas the MES scheme, as proved by Pointcheval and Stern [27], is secure against no-message attack and adaptive chosen message attack in the random oracle model.. In fact, there are two well-known levels of forgeries: one-parameter forgery and two-parameter forgery.

2.4. Threat model and assumptions

We assume that the participating network nodes voluntarily cooperate with each other to provide an anonymizing service. All nodes are potential message originators of anonymous communications. The adversaries can collaborate to passively monitor and eavesdrop every network traffic. In addition, they may compromise any node in the target network to become an internal adversary, which could be the internal perpetrators. In this paper, we assume that passive adversaries can only compromise a fraction of nodes. We also assume that the adversaries are computationally bounded so that inverting and reading of encrypted messages are infeasible. Otherwise, it is believed that there is no workable cryptographic solution.

An agent of the adversary at a compromised node observes and collects all the information in the message and thus reports the immediate predecessor and successor node for each message traversing the compromised node. Assume also that the adversary collects this information from all the compromised nodes and uses it to derive the identity of the sender of a message. The sender has no information about the number or identity of nodes being compromised. The adversary collects all the information from the agents on the compromised nodes and attempts to derive the true identity of the sender.

3. SECURE SOURCE ANONYMOUS MESSAGE AUTHENTICATION CODE (SA-MAC)

In this section, we propose an unconditionally secure and efficient scheme to generate SA-MAC for each message. The main idea is that for each message m to be transmitted, the message sender, or the sending node, generates a SA-MAC for the message m. The generation is based on the MES scheme unlike ring signatures, which require to compute a forgery signature for each member in the AS separately. In our scheme, the entire SA-MAC generation requires only three steps, which link all nonsenders and the message sender to the SA-MAC alike. In addition, our design enables the SA-MAC be verified through a single equation without individually verifying the signatures.

3.1. The proposed SA-MAC scheme

Suppose that the message sender (say Alice) wishes to transmit a message m anonymously from her network node to any other node. The AS includes n members, $A_1, A_2, \ldots, A_n$, for example, $S = \{A_1, A_2, \ldots, A_n\}$, where the actual message sender Alice is A_t, for some value t, $1 \le t \le n$.

Let p be a large prime number and g be a primitive element of $\mathbb{Z}_p^*$. Then, g is also a generator of $\mathbb{Z}_p^*$. That is, $\mathbb{Z}_p^* =< g >$. Both p and g are made public and shared by all members in S. Each $A_i \in S$ has a public key $y_i = g^{x_i} \bmod p$, where x_i is the randomly selected private key from $\mathbb{Z}_{p-1}^*$. In this paper, we will not distinguish between the network node A_i and its public key y_i. Therefore, we also have $S = \{y_1, y_2, \ldots, y_n\}$.

Suppose m is a message to be transmitted. The private key of the message sender Alice is x_t, $1 \le t \le n$. To generate an efficient SA-MAC for message m, Alice performs the following three steps:

(1) Select a random and pairwise different k_i for each $1 \le i \le n$, $i \ne t$ and compute $r_i = g^{k_i} \bmod p$.
(2) Choose a random $k \in \mathbb{Z}_p$ and compute $r_t = g^k \prod_{i \ne t} y_i^{-r_i h_i} \bmod p$ such that $r_t \ne 1$ and $r_t \ne r_i$ for any $i \ne t$, where $h_i = h(m, r_i)$.
(3) Compute $s = k + \sum_{i \ne t} k_i + x_t r_t h_t \bmod (p-1)$.

The SA-MAC of the message m is defined as

$$S(m) = (m, S, r_1, \ldots, r_n, s) \tag{2}$$

where

$$g^s = r_1, \ldots, r_n y_1^{r_1 h_1}, \ldots, y_n^{r_n h_n} \bmod p$$

and $h_i = h(m, r_i)$.

3.2. Verification of SA-MAC

A verifier can verify an alleged SA-MAC

$$(m, S, r_1, \ldots, r_n, s)$$

for message m by verifying whether the following equation holds:

$$g^s = r_1, \ldots, r_n y_1^{r_1 h_1}, \ldots, y_n^{r_n h_n} \bmod p \tag{3}$$

If Equation (3) holds true, the verifier "Accepts" the SA-MAC as a valid SA-MAC for message m. Otherwise, the verifier "Rejects" the SA-MAC.

In fact, if the SA-MAC has been correctly generated, then we have the following:

$$\begin{aligned}
& r_1,\ldots,r_n y_1^{r_1h_1},\ldots,y_n^{r_nh_n} \bmod p \\
&= g^{k_1},\ldots,g^{k_n} y_1^{r_1h_1},\ldots,y_n^{r_nh_n} \bmod p \\
&= g^{\sum_{i\neq t} k_i}\left(g^k \prod_{i\neq t} y_i^{-r_ih_i}\right)\left(\prod_{i\neq t} y_i^{r_ih_i}\right) y_t^{r_th_t} \bmod p \\
&= g^{k+\sum_{i\neq t} k_i + x_t r_t h_t} \bmod p \\
&= g^s \bmod p
\end{aligned}$$

Therefore, the verifier should always "Accept" the SA-MAC if it is correctly generated and without being modified.

Remark 1. As a trade-off between computation and transmission, the SA-MAC can also be defined as $S(m) = (m, S, r_1, \ldots, r_n, h_1, \ldots, h_n, s)$. In case S is also clear, it can be eliminated from the SA-MAC.

3.3. Security analysis

In this section, we will prove that the proposed SA-MAC scheme is unconditionally anonymous and provably unforgeable against adaptive chosen message attack.

3.3.1. Anonymity

To prove that the proposed SA-MAC is unconditionally anonymous, we have to prove that (i) for anybody other than the members of S, the probability to successfully identify the real sender is $1/n$, and (ii) anybody from S can generate SA-MACs.

Theorem 1. The proposed SA-MAC can provide unconditional message sender anonymity.

Proof. The identity of the message sender is unconditionally protected with the proposed SA-MAC scheme. This is because that regardless of the sender's identity, there are exactly $(p-1)$,$(p-2)$, . . ., $(p-n)$ different options to generate the SA-MAC, and all of them can be chosen by the SA-MAC generation procedure and by any of the members in the AS with equal probability without depending on any complexity-theoretic assumptions. The proof for the second part, that is anybody from S can generate the SA-MAC, is straightforward. This finishes the proof of this theorem.

3.3.2. Unforgeability

The design of the proposed SA-MAC relies on ElGamal signature schemes. Signature schemes can achieve different levels of security. Security against existential forgery under adaptive chosen message attack is the maximum level of security.

In this section, we will prove that the proposed SA-MAC scheme is secure against existential forgery under adaptive chosen message attacks in the random oracle model [28]. The security of our result is based on the well-known discrete logarithms problem, which assumes that the computation of discrete logarithm in $\mathbb{Z}_p$ with large p is infeasible. In other words, no efficient algorithms are known for nonquantum computers.

We will introduce two lemmas first. Lemma 1, or Splitting lemma, is a well-known probabilistic lemma from reference [27]. The basic idea of the Splitting lemma is that when a subset Z is "large" in a product space $X \times Y$, it has many "large" sections. Lemma 2 is a slight modification of the Forking lemma presented in [27]. The proof of this theorem is mainly probability theory related. We will skip the proof of these two lemmas here.

Lemma 1. (The Splitting lemma)

Let $A \subset X \times Y$ such that $Pr[(x, y) \in A] \geq \varepsilon$. For any $\alpha < \varepsilon$, define $B = (\mathrm{x}, \mathrm{y}) \in \mathrm{X} \times \mathrm{Y} | \, Pr_{y' \in Y}\left[\left(x, y'\right) \in A\right] \geq \varepsilon - \alpha$ and $\bar{B} = (X \times Y)/B$, and then the following statements hold:

(1) $Pr[B] \geq \alpha$

(2) $\forall (x, y) \in B, Pr_{y' \in Y}\left[\left(x, y'\right) \in A\right] \geq \varepsilon - \alpha$

(3) $Pr[B|A] \geq \alpha/\varepsilon$

Lemma 2. The Forking lemma

Let A be a Probabilistic Polynomial Time (PPT) Turing machine given only the public data as input. If A can find, with nonnegligible probability, a valid SA-MAC $(m, S, r_1,$ r_n, $h_1, \ldots, h_n, s)$ within a bounded polynomial time T, then with nonnegligible probability, a replay of this machine that has control over A and a different oracle outputs another valid SA-MAC $(m, S, r_1, \ldots, r_n, h'_1, \ldots, h'_n, s)$, such that $h_i = h'_i$, for all $1 \leq i \leq n$, $i \neq j$ for some fixed j.

Theorem 2. The proposed source anonymous communication message authentication code is secure against adaptive chosen message attack in the random oracle model.

Proof. (Sketch) If an adversary can forge a valid SA-MAC with nonnegligible probability, then according to the Forking lemma, the adversary can obtain two valid SA-MACs

$$(m, S, r_1, \ldots, r_n, h_1, \ldots, h_n, s)$$

and

$$\left(m, S, r_1, \ldots, r_n, h'_1, \ldots, h'_n, s'\right)$$

such that for $1 \leq i \leq n$, $i \neq j$, $h_i = h_i$, and $h_j \neq h'_j$. That is

$$g^s = r_1, \ldots, r_n y_1^{r_1h_1}, \ldots, y_n^{r_nh_n} \bmod p \tag{4}$$

and

$$g^{s'} = r_1, \ldots, r_n y_1^{r_1h'_1}, \ldots, y_n^{r_nh'_n} \bmod p \tag{5}$$

Dividing Equations (4) and (5), we obtain

$$g^{s-s'} = y_t^{r_t\left(h_t - h'_t\right)} \mod p \tag{6}$$

Equivalently, we have

$$y_t = g^{\frac{s-s'}{r_t\left(h_t - h'_t\right)}} \mod p \tag{7}$$

Therefore, we can compute the discrete logarithm of y_t in base g with nonnegligible probability, which contradicts to the assumption that it is computationally infeasible to compute the discrete of y_j in base g.

4. THE PROPOSED ANONYMOUS COMMUNICATION PROTOCOL

4.1. Network model

In this paper, we adopt a structured overlay network topology used in many peer-to-peer systems such as KaZaa [29], Gnutella v0.6 [30], Herbivore [13], and Chord [10] to organize the network. That is, the participating nodes are divided into a set of small subgroups. From the topology of the overlay network, we expect that the adversary should not be able to distinguish the initiator traffic from the indirection traffic on an observable and open network. On the other hand, we know that no scheme can hide the fact that a node is participating. The best a scheme can do is to guarantee that no adversary can distinguish actively that a node initiates from mere participation in the protocol. In other words, a node can hide its own activities by handling traffic for other nodes.

Considering the overlay topology given in Figure 1, for node *A*, because it is the top of the branch, it can only be the initiator node. Whereas for node *B*, it has one predecessor and two successors; it could be hard to distinguish between initiator traffic and indirection from node *B*. However, the traffic volume in branch *BC* and *BD* will generally be different from branch *AB*. In fact, the difference is the volume of traffic that can be used to measure the traffic that initiates from node *B*. Finally, for node *C* and node *D* in Figure 1, because they do not have successor, all traffic that go to them are final; therefore, they can only be the recipients.

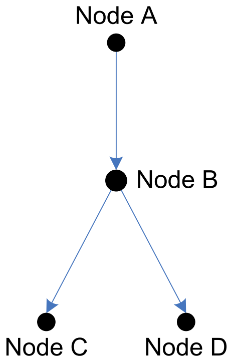

Figure 1. The tree topology illustration.

To prevent the node from being either the beginning or end of the branch and avoid possible traffic balance analysis, in this paper, the nodes in each subgroup are logically organized into an overlay shaped as a ring shown in Figure 2. In each ring, there are n nodes, where n is a predefined security parameter. Each node/link can route message towards the *successor*, that is, the next hop in the clockwise direction of the ring. We refer this direction as the *ring direction*.

Besides network topology, the more traffic a node creates, the more foreign traffic it must forward. Moreover, the messages that each node receive and send should be balanced to obscure its own actions and prevent timing analysis.

We classify the network nodes into two categories, *normal nodes* and *super nodes*. A normal node is a network node that has no direct connection to the nodes in other networks. A super node can be a normal node that can also provide message forward services to other network nodes. It can also be a special node dedicated to providing message forward services to the other network nodes. Each network may have multiple super nodes as highlighted in Figure 2.

Prior to network deployment, there should be an administrator. The administrator is responsible for selection of security parameters and a group-wise master key $s_G \in \mathbb{Z}_p^*$. The group master key should be well safeguarded from unauthorized access and never be disclosed to the ordinary group members. The administrator then chooses a collision-resistant cryptographic hash function h, mapping arbitrary inputs to fixed-length outputs on $\mathbb{Z}_p$, for example, SHA-1 [31].

The administrator assigns each super node a sufficiently large set of collusion-free pseudonyms that can be used to substitute the real IDs in communications to defend against passive attacks. If a super node uses one pseudonym continuously for some time, then it will not help to defend against possible attacks because the pseudonym can be

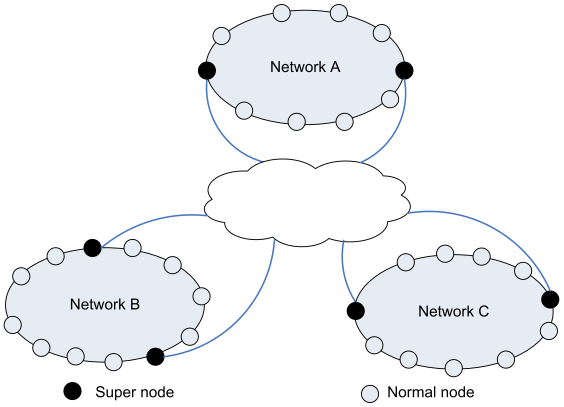

Figure 2. An illustration of network topology of the proposed scheme.

analyzed the same way as its real ID. To solve this problem, each node should use dynamic pseudonyms instead. This requires each super node to sign up with the administrator, who will assign each super node a list of random and collision-resistant pseudonyms:

$$N_A = \left\{ id_1^A, \cdots, id_\tau^A \right\}$$

In addition, each super node will also be assigned a corresponding *secret set*:

$$S_s = \left\{ g^{s_G h\left(id_1^A\right)}, \ldots, g^{s_G h\left(id_\tau^A\right)} \right\}$$

4.2. Anonymous local ring communication

To realize anonymous network-layer communications, obviously, there should be no explicit information (such as the message sender and recipient Internet addresses) in the message in the overlay network. All of the information related to overlay addresses, including the destination ring where the recipient resides, should be embedded into the anonymizing message payload.

Prior to network deployment, the administrator needs to select a set of security parameters for the entire system, including a large prime p and a generator g of $\mathbb{Z}_p^*$. The network nodes $A_1, A_2, \ldots, A_n$ and the corresponding public keys $y_1, y_2, \ldots, y_n$ of the n participating network nodes, where $x_i \in \mathbb{Z}_p$, are a randomly selected private key of node A_i, and y_i is computed from $y_i = g^{x_i} \bmod p$.

In each local ring, a normal node only has connection to other nodes in the same ring. The communication between two normal nodes in different rings has to be forwarded through the supper nodes in the respected local rings.

Each message contains a nonce (N), a message flag (mF), a recipient flag (rF), and a secret key. The nonce is a random number that is used only once to prevent message replay attack. The message flag carries the priority of the message. The message flag value 0 means that the transmitted message is a dummy message or the cell is empty. The dummy message can be replaced if the current node has a message to transmit. The message flag 1 means the message is meaningful and has highest priority to be transmitted. Optionally, we can define message flag 2 to represent that the message is meaningful and should be delivered if possible. However, it has a lower priority. If there is a confliction with message priority 1, then it can be delayed. Certainly, more priority can be defined and enforced. The recipient flag enables the recipient to know whether he is the targeted receiver. The secret key is used to encrypt the subsequent block(s) by using symmetric encryption algorithm.

Prior to data transmission, a super node needs to first initiate the data transmission in the local ring, which is a dummy message transmitted to the next super node following the ring direction in the local ring (Figure 3). However, the message flag is set to be 0. When the packet reaches a normal node, if that node has data to transmit, it can replace the dummy data with its own message. However, to continue the message transmission and prevent the node from being identified as the receiver, the recipient node creates a new dummy message and to be delivered to the closest super node. Similarly, any node can substitute the dummy message with its own message. However, when a dummy message arrives at a super node, it needs to regenerate a dummy message and sends to the next super node so that this process can be continued.

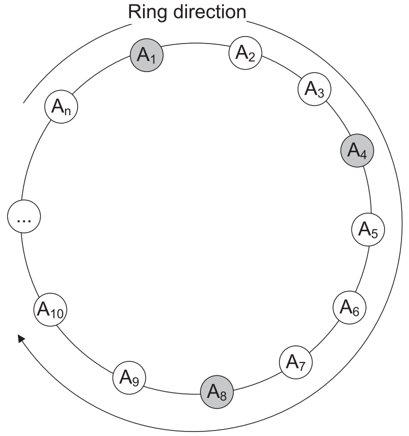

Figure 3. Local ring.

$$\begin{aligned} M(i,j) &= pk_{i+1}(N_{i+1}, mF_{i+1}, rF_{i+1}, sk_{i+1}) sk_{i+1}(M(i+1,j)) \\ M(i+1,j) &= pk_{i+2}(N_{i+2}, mF_{i+2}, rF_{i+2}, sk_{i+2}) sk_{i+2}(M(i+2,j)) \\ &\vdots \\ M(j-1,j) &= pk_j\left(N_j, mF_j, rF_j, sk_j\right) sk_j(S(m)) \end{aligned} \tag{8}$$

At any time, multiple concurrent messages may be transmitted in the local ring. The number of such messages can be determined by the data transmission requirement as well as the node transmission capacity. The mix-up of dummy information with the real messages makes the adversaries unable to detect the real message senders and the receivers. Because no single node will receive the same packet more than once, therefore, no single node is able to identify the real receiver of each message packet.

More specifically, for a node A_i to transmit a message m anonymously to a node A_j in the local ring, where $j > i$, node A_i generates a new message $M(i,j)$ defined in Equation (8), where for $l = i+1, \ldots, j$, N_l is a nonce, mF_l is a message flag, rF_l is a recipient flag, sk_l is the secret key used for one-time message encryption, and $\|$ stands for message concatenation.

The message $M(i,j)$ can be transmitted when a dummy message is received. The node substitutes the dummy message with $M(i,j)$. The message will then be forwarded node to node to the successor nodes $A_{i+1}, A_{i+2}, \ldots, A_j$ until it reaches the message recipient in the ring direction, which is the clockwise direction.

When the node A_{i+1} receives the message packet, the node decrypts the first block of the received message by using its private key corresponding to pk_{i+1}. After that, the node will take the recipient flag and message flag with the instruction for the following actions.

The amount of traffic flow that a node creates as the initiator is concealed in the traffic that it forwards because the overall traffic that it receives is the same as the traffic that it forwards. In addition to the balanced traffic, the message is encrypted with the private key that only the recipient can recover, while the intermediate nodes can only view the instruction of the message allowed. As the sender's message is indistinguishable by other nodes, the sender and the recipient are thus hidden amongst the other nodes. *It is infeasible for the adversary to correlate messages by using traffic analysis and timing analysis because of message encryption*. Therefore, perfect obscure of its own messages can be assured. Detailed security analysis will be presented later on.

With the measurement of dummy messages that it receives, the super nodes can determinate whether the volume of messages in concurrent transmission in the local ring should be increased or decreased to optimize the overall system performance.

In the proposed protocol, a node's joining and leaving in the overlay ring are straightforward. When a node wishes to join a ring, it only needs to find two adjacent nodes where it would like to join the ring. For a node to leave the ring, the predecessor of the node should simply skip the current node and communicate directly to its successor as long as they each has the other node's necessary communication information.

4.3. Anonymous communications between two arbitrary super nodes

In the previous section, we present the mechanism that allows two arbitrary nodes to communicate anonymously in the same local ring. This includes communications between two super nodes in the same local ring. For two arbitrary super nodes in different rings to communicate anonymously, we will first introduce the concept of anonymous authentication or secret handshake by Balfanz *et al*. [32]. Anonymous authentication allows two nodes in the same group to authenticate each other *secretly* in the sense that each party reveals its group membership to the other party only if the other party is also a group member. Nonmembers are not able to recognize group members. Secret handshake has been applied in anonymous routing in mobile ad hoc networks [33].

The scheme consists of a set of super nodes, an administrator who creates groups and enrolls super nodes in groups. For this purpose, the administrator will assign each super node A a set of pseudonyms $id_1^A, \ldots, id_\tau^A$, where τ is a large security parameter. In addition, the administrator also calculates a corresponding *secret set* $\left\{g^{s_G h\left(id_1^A\right)} \bmod p, \ldots, g^{s_G h\left(id_\tau^A\right)} \bmod p\right\}$ for super node A, where s_G is the group secret and h is a hash function. The pseudonyms will be dynamically selected and used to substitute the real IDs in each communications. This means that two super nodes A and B can know each other's group membership only if they belong to the same group.

When the super node A wants to authenticate to the super node B, the following secret handshakes can be conducted:

(1) $A \rightarrow B$: Super node A randomly selects an unused pseudonym id_i^A and a random nonce N_1 and then sends id_i^A, N_1 to super node B.
(2) $B \rightarrow A$: Super node B randomly selects an unused pseudonym id_i^B and a random nonce N_2 and then sends $id_j^B, N_2, V_0 = h(K_{BA}||id_i^A||id_j^B||N_1||N_2||0)$ to super node A, where $K_{BA} = g^{s_G h\left(id_i^A\right) \cdot h\left(id_j^B\right)} \bmod p$.
(3) $A \rightarrow B$: Super node A sends $V_1 = h(K_{AB}||id_i^A||id_j^B||N_1||N_2||1)$ to super node B, where $K_{AB} = g^{s_G h\left(id_j^B\right) \cdot h\left(id_i^A\right)} \bmod p$.

Because

$$\begin{aligned} K_{BA} &= g^{s_G h\left(id_i^A\right) \cdot h\left(id_j^B\right)} \bmod p \\ &= g^{s_G h\left(id_j^B\right) \cdot h\left(id_i^A\right)} \bmod p = K_{AB} \end{aligned}$$

A can verify V_0 by checking whether

$$V_0 \overset{?}{=} h\left(K_{AB}||id_i^A||id_j^B||N_1||N_2||0\right)$$

If the verification succeeds, then A knows that B is an authentic group peer. Similarly, B can verify A by checking whether

$$V_1 \overset{?}{=} h\left(K_{BA}||id_i^A||id_j^B||N_1||N_2||1\right)$$

If the verification succeeds, then B knows that A is also an authentic group peer. However, in this authentication process, neither super node A nor super node B can obtain the real identity of the other node. In other words, the real identities of super node A and super node B remain anonymous after the authentication process.

4.4. Anonymous communication between two arbitrary normal nodes

We already described the mechanisms for two nodes in the same ring to communicate anonymously. In this section, we will introduce the anonymous communication mechanism for two arbitrary nodes in different overlay network rings.

We already mentioned in the previous section that there should be no explicit exposure about the Internet addresses of the message sender and recipient. To transmit a message, the sender first randomly selects a local super node and transmits the message to the super node according to the mechanism described before. On receiving the message, the local super node first determines the destination ring ID by checking the message recipient flag *rF*, either

0 or 1. If it is 0, then the recipient and the super node are in the same local ring. The message can be forwarded in the recipient node by using the previously described mechanism. If *rF* is 1, then the recipient is in different ring, The super node forwards the message to a super node in the destination ring as described in the previous section. Finally, when the super node in the recipient's local ring receives the message, the communications again become local ring communications. The message can now be transmitted in the same way that the sender and the recipient are in the same local ring.

Alternatively, the super node can broadcast the received message in the recipient's ring. While providing message recipient anonymity, the message can also be encrypted so that only the message recipient can decrypt the message.

The proposed anonymous communication is quite general and can be used in a variety of situations for communication anonymity, including anonymous file sharing. For anonymous file sharing, the file requester needs to send a query to a local super node about the desired file, the requester's ring ID, and optionally the requester's public key in the anonymous message payload so that no broadcasting is needed in the recipient's ring.

4.5. Security analysis

In this section, we will analyze anonymity, impersonation attack, and replay attack of the proposed anonymous communication protocol.

4.5.1. Anonymity

We will first prove that the proposed communication protocol can provide both message sender and the recipient anonymity in the local ring communications.

Theorem 3. It is computationally infeasible for an adversary to identify the message sender and recipient in the local ring. Therefore, the proposed anonymous communication protocol provides both sender and recipient anonymity in the local ring.

Proof. (Sketch) The number of message packages that each node receives from its immediate predecessor is the same as the number of packets that it forwards to its immediate successor. Moreover, each message package is encrypted using either the public keys or the shared secret keys of the intermediate nodes. No adversary can distinguish the real meaningful message from the dummy message in the transmission because of the traffic balance property and message content encryption. Therefore, the adversary cannot distinguish the initiator traffic from the indirection traffic and learn whether the node is a recipient, a receiver, or simply a node that provides message forward service. Consequently, both the message sender and recipient information are anonymous from the adversary attack.

For two normal nodes in different rings to communicate anonymously, the communication can be broken into three segments: the communication between the sender and a local super node, the communication between two super nodes in the corresponding rings, and the communication between the recipient super node and the recipient. Theorem 3 has assured the communication anonymity between a super node and a normal node in the local rings. Therefore, we only need to ensure the anonymity between two super nodes in different rings to achieve full anonymity between the sender and recipient.

We already described before that each super node has a large set of pseudonyms. A dynamically selected pseudonym will be used for each communications. The pseudonyms do not carry the user information. Therefore, the adversary cannot obtain any information of the super nodes from the network. This result can be summarized into the following theorem.

Theorem 4. The proposed communication protocol between two super nodes can provide both message sender and recipient anonymity.

Corollary 1. The proposed anonymous communication protocol can provide full anonymity for any sender and recipient in the overlay network ring(s).

4.5.2. Impersonation attacks

For an adversary that elects to perform impersonation attack, if his target is the normal node, then he needs to conduct forgery attack. We already proved in Theorem 2 that this is infeasible. Therefore, we only need to consider whether it is feasible for an adversary to forge a super node.

For an adversary to impersonate as a super node, he needs to perform secret authentication with a super node *A*. This requires the adversary *A* to compute $g^{s_G id^A \cdot id_i^A} \bmod p$, where id^A is the identity of the adversary and id_i^A is the *i*th pseudonym of the super node *A*. However, because the adversary does not know the master secret s_G, he or she is unable to compute $g^{s_G h\left(id^A\right) \cdot h\left(id_i^A\right)} \bmod p$ and impersonate as a super node. Therefore, we have the following theorem.

Theorem 5. It is computationally infeasible for a PPT adversary *A* to impersonate as a super node.

Like all other network communication protocols, in our proposed protocol, an adversary may choose to drop some of the messages. However, if the immediate predecessor and the successor nodes are honest and willing to cooperate, then the messages being dropped and the substitution of the valid messages with the dummy messages can be effectively tracked using the provided message flags.

An adversary that is elected as a super node may refuse to forward messages across the rings and thus block the anonymous communications between the sender and the receiver. This attack can be hard to detect if the sender does not have the capability to monitor all network traffic. However, the sender can randomly select the super nodes for each data transmission. If the nonce is properly generated, when a packet is lost, the recipient should be able to know.

4.5.3. Message replay attacks

The message replay attack occurs when an adversary can intercept the communication packet, correlate the message to the corresponding sender and recipient, and retransmit it.

Theorem 6. It is computationally infeasible for an adversary to successfully modify/reply an (honest) node's message.

Proof. (Sketch) According to Equation (8), each message package in communication has a unique one-time session identifier (nonce) to protect the message package from modification and replay attacks. In addition, these fields are encrypted using the intermediate receiver nodes' public keys so that only the designated receiver nodes can decrypt the message. Each packet bears different and uncorrelated IDs when transmitted across different rings. Therefore, it is computationally infeasible for the adversary to modify or replay any messages in the overlay network ring. In fact, even if the same message is transmitted multiple times, the adversary cannot link them together without knowing all the private keys of the intermediate nodes.

4.6. Efficiency and performance evaluation

Anonymity is achieved as a result of trade-off with efficiency and computational complexity. In our case, the transmission of dummy messages is required as a message carrier in the local ring. It thus increases the communication overhead and the average data latency. In terms of *communication complexity* (the messages transmitted in the network for every anonymous message), *time complexity* (time required to transmit a message) and *buffer complexity* (the buffer size required for each processor to the messages) [4], we have the following theorem.

Theorem 7. In the proposed protocol, the communication complexity of the proposed protocol is $O(n)$, time complexity is $O(n)$, and buffer complexity is $O(n)$.

Proof. (Sketch) For our proposed protocol, the communication complexity and time complexity is the same, which is $O(n)$ because we need to transmit $O(n)$ message for each anonymous message sent. It is clear that the buffer complexity of the proposed protocol is at most $O(n)$.

In addition, the proposed protocol also increases extra computational complexity of each node because it has to decrypt every received message and verify the message authentication code.

There is always a trade-off between time complexity and communication complexity. For example, to reduce the transmission latency, multiple messages can be transmitted in a ring concurrently. However, this will increase the computational complexity.

4.7. Simulations

Simulation results are provided in Figure 4 to demonstrate the communication delay and delivery ratio of the proposed scheme. Our simulation was performed using ns-2 on Linux system. Our simulation results are based on the average of randomly selected packets from each node. In the simulation, the target area is a square field of size 8000 × 8000 m. We partition this field into 2500 normal grids/nodes.

The mixing ring is composed of 80 grids, that is, $r = 80$. There are four relay ring nodes in the mixing ring, that is, $n = 4$. We assume that the randomly selected intermediate node is at least 600 m away from the real message source. The data messages are 8-bit long, that is, $l = 8$. The vehicle messages are 16-bit long, that is, $L = 16$.

Our simulation results demonstrate while enabling 100% package delivery ratio that the proposed scheme is also very efficient and can be used for practical applications.

5. CONCLUSION

In this paper, we first propose a novel and efficient SA-MAC that can be applied to any messages. While ensuring unconditional message sender anonymity, SA-MAC can also provide message content authenticity. To provide provable anonymity without suffering from transmission collusion problem, we then propose a novel anonymous communication protocol for both message sender and recipient. Security analysis shows that the proposed protocol is secure against various attacks. Our analysis also shows that it is efficient and practical. The proposed protocol can be applied for secure file sharing.

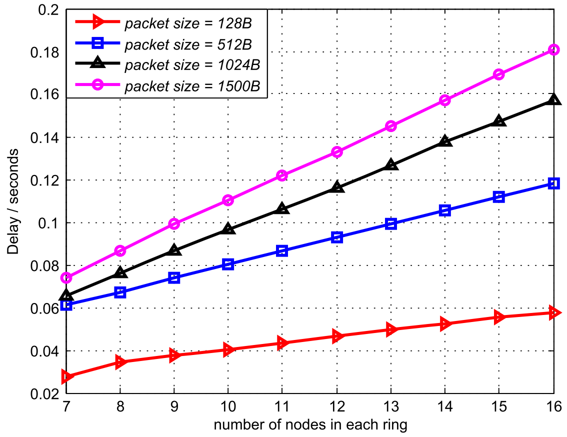

Figure 4. Message transmission delay.

Notation

AS or S	anonymity set
MAC	message authentication code
MAC_i	the ith message authentication code
SA-MAC	source anonymous MAC
MES	modified ElGamal signature
PPT	probabilistic polynomial time
p	a large prime number
$\mathbb{Z}_p$	the integer field module p
g	a primitive element in $\mathbb{Z}_p$
x_i	the private key of the ith user
y_i	the public key of the ith user
r_i	the ith signature component
s	the ElGamal signature component
m	the message
n	the number of users in the AS
A_i	the ith member in the AS
h	hash function
h_i	hash value $h(m, r_i)$
pk_i	A_i's public key for message header encryption
$pk_i(m)$	encrypt m using public key pk_i
sk_i	A_i's secret key for symmetric encryption
$sk_i(m)$	encrypt m using secret key sk_i
N_i	the ith nonce
mF_i	the ith message flag
rF_i	the ith recipient flag
$\|$	message concatenation
$M(i,j)$	message to send from node i to node j
$S(m)$	the SA-MAC of message m

ACKNOWLEDGEMENTS

This research was supported in part by the NSF under grants CNS-0845812 and CNS-1050326, and CNS-1117831.

REFERENCES

1. Chaum D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM* February 1981; **24**: 84–88.
2. Chaum D. The dinning cryptographer problem: Unconditional sender and recipient untraceability. *Journal of Cryptology* 1988; **1**(1):65–75.
3. von Ahn L, Bortz A, Hopper N. *k*-Anonymous message transmission. In Proceedings of CCS, Washington D.C., USA, 2003; 122–130.
4. Beimel A, Dolev S. Buses for anonymous message delivery. *Journal of Cryptology* 2003; **16**:25–39.
5. Berthold O, Federrath H, Köpsell S. Web MIXes: a system for anonymous and unobservable Internet access. Lecture Notes in Computer Science, 2001; 115–129.
6. Danezis G, Dingledine R, Mathewson N. Mixminion: design of a type III anonymous remailer protocol. IEEE Symposium on Security and Privacy, 2003; 2–15.
7. Golle P, Juels A. Dining cryptographers revisited. In Advances in Cryptology - Eurocrypt 2004. LNCS 3027. 2004; 456–473.
8. Möller B. Provably secure public-key encryption for length-preserving Chaumian mixes. In Proceedings of CT-RSA 2003, LNCS 2612. April 2003; 244–262.
9. Reed M, Syverson P, Goldschlag D. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* 1998; **16**(4): 482–494.
10. Reiter M, Rubin A. Crowds: anonymity for web transaction. *ACM Transactions on Information and System Security* 1998; **1**(1):66–92.
11. Gülcü C, Tsudik G. Mixing email with Babel. In Proceedings of the Symposium on Network and Distributed System Security, San Diego, CA, 1996.
12. Möller U, Cottrell L, Palfrader P, Sassaman L. Mixmaster protocol. Version 2. July 2003.
13. Goel S, Robson M, Polte M, Sirer E. Herbivore: a scalable and efficient protocol for anonymous communication. Tech. Rep. 2003–1890, Cornell University, Ithaca, NY, February 2003.
14. Fu X, Zhu Y, Graham B, Bettati R, Zhao W. On flow marking attacks in wireless anonymous communication networks. In Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), April 2005.
15. Pfitzmann A, Hansen M. Anonymity, unlinkability, unobservability, pseudonymity, and identity management a proposal for terminology. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf, Feb. 15 2008.
16. Pfitzmann A, Waidner M. Networks without user observability--design options. In Advances in Cryptology - EUROCRYPT 1985, Lecture Notes in Computer Science. 1985; **219**: 245–253.
17. Waidner M. Unconditional sender and recipient untraceability in spite of active attacks. In Advances in Cryptology - EUROCRYPT 1989. Lecture Notes in Computer Science, Vol. **434**. 1989; 302–319.
18. Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures. *Journal of Cryptology* 2000; **13**(3):361–396.
19. Harn L, Xu Y. Design of generalized ElGamal type digital signature schemes based on discret logarithm. *Electronics Letters* 1994; **30**(24):2025–2026.
20. Nyberg K, Rueppel RA. Message recovery for signature schemes based on the discrete logarithm problem. In Advances in Cryptology - EUROCRYPT 1995. Lecture Notes in Computer Science, Vol. **950**. 1995; 182–193.

21. Wright M, Adler M, Levine B, Shields C. An analysis of the degraduation of anonymous protocols. In Proceedings of the Network and Distributed Security Symposium, San Diego, CA, 2002.
22. Shields C, Levine BN. A protocol for anonymous communication over the Internet. In Proceedings of the 7th ACM Conference on Computer and Communication Security, Gritzalis D (ed.). ACM Press: Athens, Greece, 2000.
23. Wang P, Ning P, Reeves DS. A *k*-anonymous communication protocol for overlay networks. In ASIACCS--07, Singapore, March 20–22, 2007.
24. Rivest R, Shamir A, Tauman Y. How to leak a secret. In Advances in Cryptology–ASIACRYPT. Lecture Notes in Computer Science, Vol. **2248/2001**. Springer: Berlin/ Heidelberg, 2001.
25. ElGamal TA. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory* 1985; **31**(4):469–472.
26. Goldwasser S, Micali S, Rivest R. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing* April 1988; **17**: 281–308.
27. Pointcheval D, Stern J. Security proofs for signature schemes. In Advances in Cryptology - EUROCRYPT 1996. Lecture Notes in Computer Science, Vol. **1070**. 1996; 387–398.
28. Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. In CCS'93, 1993; 62–73.
29. T. homepage of KaZaa, http://www.kazaa.com.
30. T. homepage of Gnutella, http://gnutella.wego.com.
31. F. P. 180–1. Secure hash standard. http://itl.nist.gov/fipspubs/fips180-1.htm, Apr. 1995.
32. Balfanz D, Durfee G, Shankar N, Smetters D, Staddon J, Wong HC. Secret handshakes from pairing-based key agreements. In IEEE Symposium on Security & Privacy, Oakland, CA, May 2003.
33. Zhang Y, Liu W, Lou W, Fang Y. MASK: anonymous on-demand routing in mobile ad hoc networks. *IEEE Transactions on Wireless Communications* September 2006; **5**: 2376–2385.