

# A New $k$ -Anonymous Message Transmission Protocol

Gang Yao, Dengguo Feng

State Key Laboratory of Information Security  
Institute of Software, Chinese Academy of Sciences

# Introduction

These protocols address the problem of concealing who communicates with whom, as in the case of letters from a secret admirer.

The goal is usually to guarantee full anonymity: an adversary looking at the communication patterns should not learn anything about the origin or destination of a particular message.

# **$k$ -anonymity**

$k$ -anonymity guarantees that in a network with  $n$  honest participants, the adversary is not able to guess the sender or recipient of a particular message with probability non-negligibly greater than  $1/k$ .

$k$ -anonymity is a weaker guarantee, it is still sufficient for a variety of applications.

# Reasons for anonymity

- Want to participate in a Usenet discussion without their identity being revealed to the rest of the newsgroup.
- Want to search for sensitive information, without any eavesdropper knowing about it.
- Take part in a discussion list among patients with sensitive diseases like AIDS.
- Express opinion, such as a comment about one's boss, which may have repercussions.

# Related Work

- Proxies and Proxy Chaining.
- DC-Nets.
- Mix-Nets.
- Crowds.
- CliqueNet.
- $k$ -Anonymous Message Transmission.
- Receiver anonymity via incomparable public keys.

# Notation

- message space:  $\mathcal{M}$
- $n$  parties:  $P_1, \dots, P_n$
- private input:  $(msg_i, p_i) \in (\mathcal{M} \times [n]) \cup \{(nil, nil)\}$
- $H \subset \{P_1, \dots, P_n\}$ : the set of honest parties
- $\mathcal{P}(P_1(msg_1, p_1), \dots, P_n(msg_n, p_n))$ : the adversary's view of the protocol  $\mathcal{P}$  when each  $P_i$  has input  $(msg_i, p_i)$
- $\mathcal{P}(P_i(msg_i, p_i), *)$ : the adversary's view of  $\mathcal{P}$  when  $P_i$  has input  $(msg_i, p_i)$  and the other inputs are set arbitrarily

# sender anonymous

A protocol  $\mathcal{P}$  is *sender anonymous* if for every pair  $P_i, P_j \in H$ , and every pair  $(msg, p) \in (\mathcal{M} \times [n]) \cup \{(\text{nil}, \text{nil})\}$ ,  $\mathcal{P}(P_i(msg, p), *)$  and  $\mathcal{P}(P_j(msg, p), *)$  are computationally indistinguishable.

## receiver anonymous

A protocol  $\mathcal{P}$  is *receiver anonymous* if for every  $P' \in H$ , for every  $msg \in \mathcal{M}$  and every  $P_i, P_j \in H$ ,  $\mathcal{P}(P'(msg, P_i), *)$  and  $\mathcal{P}(P'(msg, P_j), *)$  are computationally indistinguishable.

# **unlinkability of sender and receiver**

Unlinkability of sender and receiver means that though the sender and receiver can each be identified as participating in some communication, they cannot be identified as communicating with each other.

## sender $k$ -anonymous

A protocol  $\mathcal{P}$  is *sender  $k$ -anonymous* if it induces a partition  $\{V_1, \dots, V_l\}$  of  $H$  such that:

1.  $|V_s| \geq k$  for all  $1 \leq s \leq l$ ;
2. For every  $1 \leq s \leq l$ , for all  $P_i, P_j \in V_s$ , for every  $(msg, p) \in (\mathcal{M} \times [n]) \cup \{(\text{nil}, \text{nil})\}$ ,  $\mathcal{P}(P_i(msg, p), *)$  and  $\mathcal{P}(P_j(msg, p), *)$  are computationally indistinguishable.

# receiver $k$ -anonymous

A protocol  $\mathcal{P}$  is *receiver  $k$ -anonymous* if it induces a partition  $\{V_1, \dots, V_l\}$  of  $H$  such that:

1.  $|V_s| \geq k$  for all  $1 \leq s \leq l$ ;
2. For every  $1 \leq s \leq l$ , for all  $P_i, P_j \in V_s$ , for every  $P' \in H$ ,  $msg \in \mathcal{M}$ :  $\mathcal{P}(P'(msg, P_i), *)$  and  $\mathcal{P}(P'(msg, P_j), *)$  are computationally indistinguishable.

# Our Protocol

- Protocol Preparation.
- Collection Phase.
- Transmission Phase.

# Protocol Preparation

- All the members need to get a long term public/private key pair.
- In order to band the public/private key pair to a legal member, may use certificate system.
- Partition the  $n$  members in the protocol into smaller groups of size  $O(k)$ .

# Collection Phase (I)

Each member  $P_i$  in the group chooses  $g_i$ , index of the group that the receiver belongs to, and  $l_i$ , the length of the message he wants to transfer. If  $P_i$  do not want to send message, he may choose  $g_i$  greater than  $g$ .

# Collection Phase (II)

- Each  $P_i$  chooses  $g_i$  and  $l_i$ , then computes  $x_i = (g_i || l_i)^{e_0}$ .
- $P_i$  randomly chooses  $j \in [1, k]$  and sends  $x_i$  to  $P_j$ .
- Received  $x_i$ ,  $P_j$  sends it to  $GL$ , the group leader.
- Received  $x_i$ ,  $GL$  computes  $y_i = x_i^{d_0}$  to get  $g_i$  and  $l_i$ .
- If  $n/k \leq g_i$ ,  $GL$  discards this message; if  $1 \leq g_i \leq n/k$ ,  $GL$  publishes  $g_i$ . Let  $l$  denote the biggest  $l_i$  among all the  $l_i$  satisfies  $1 \leq g_i \leq n/k$ .  $GL$  also publishes  $l$ .

## Collection Phase (III)

After  $GL$  publish  $l$  and all the indexes, each  $P_i$  checks that whether the group index  $g_i$  is in the publish index set and whether the length  $l_i$  is smaller than  $l$ . If both conditions are satisfied,  $P_i$  broadcasts “yes” message, otherwise, he broadcasts “no” message.

If all the members in the group broadcast “yes” message, the protocol goes to the next step; otherwise, stops.

# Transmission Phase (I)

In the last step, each member in the group, say group  $A$ , knows  $g_i$ , the index of the group which a message will be transferred to, and  $l$ , the length of the message that can be transferred. Now, suppose that a message will be transferred to a member in the group  $g_i$ , say group  $B$ , then all the members in the group may perform the following protocol.

## Transmission Phase (II)

- $P_i$  (group  $A$ ) gets the public keys of all the members in group  $B$ . The  $P_i$  randomly chooses  $k_{i,j}$ .
- If  $P_i$  wants to transfer message  $msg_{i,j}$  to  $Q_j$ , computes  $k'_{i,j} = (k_{i,j})^{e_{B,j}}$  and  $msg'_{i,j} = E_{k_{i,j}}(msg_{i,j})$ . If  $P_i$  does not want to transfer message to  $Q_j$ , he chooses a random string as the message  $msg_{i,j}$ . If the length of the message  $msg_{i,j}$  is smaller than  $l$ ,  $P_i$  may pad 0 at the end of the message.

## Transmission Phase (III)

■  $P_i$  sends the message  $k'_{i,1} || msg'_{i,1} || \dots || k'_{i,n_B} || msg'_{i,n_B}$  to  $GL$ .

■ Receiving all the messages, the group leader construct a message  $M_j$  and send it to the  $j$ -th member of group  $B$ , where  $1 \leq j \leq n_B$ . Here,

$$M_j = k'_{i_1,j} || msg'_{i_1,j} || \dots || k'_{i_{n_A},j} || msg'_{i_{n_A},j},$$

where  $\{i_1, \dots, i_{n_A}\}$  is a permutation of  $\{1, \dots, n_A\}$ .

## Transmission Phase (IV)

Receiving the message  $M_j$ ,  $Q_j$  can obtain the  $k_{i,j}$  by computing  $(k'_{i,j})^{d_{B,j}}$ , and  $msg_{i,j}$  by  $D_{k_{i,j}}(msg'_{i,j})$ .

# Analysis of Our Protocol

- Performance.
- Robustness.
- Anonymity.
- Privacy.
- More discussion on receiver.
- Efficiency.

# Conclusion

- Our protocol provides the privacy for the message.
- A member in a group may send different messages to different members in another group at one time.
- A member in a group may receive several messages from the different members in another group at one time.
- The total message transferred in one round is  $O(k)$  and the total bits transferred in one round is  $O(k^2(l + L))$ .

# Future Works

- We will focus on how to detect the cheater in the group.
- We are looking into techniques for distributing the data and signatories in a decentralized way.

*THANKS*

Any Question?