Massive Group Message Authentication with Revocable Anonymity

Boaz Catane and Amir Herzberg
Department of Computer Science, Bar Ilan University, Ramat Gan, 52900, Israel
{*cataneb, herzbea*}*@cs.biu.ac.il*

Keywords: Digital Signatures, Group Signatures, Revocation, Efficient Implementations.

Abstract: We present and implement schemes for authenticating messages from a group of users to a recipient, with revocable anonymity and massive (very high) message rate. Our implementations present a trade-off between the efficiency and the security required: from online group managers that participate in every message sent to offline managers, from assuming a trusted group manager and a trusted recipient to securing against both entities. All implementations have the *traceablity* feature, allowing distributive and efficient tracing of all messages originating from a specific group member without violating anonymity of other members. In addition, our schemes are efficient and practical.

1 INTRODUCTION

Schemes where messages are sent to a central location are widely used for various purposes. In many cases, only a designated group of users may send messages to the central location. In addition, sometimes the designated users' privacy should be kept. In such cases an anonymous messaging or reporting scheme should be used.

Commonly, when questionable or faulty reports are found in a system, the identity of the originator of such reports should be revealed. In such cases schemes allowing anonymity revocation should be used. Moreover, if the originator is found to be malicious there is also a need to efficiently single out all reports generated by this user.

An example where such schemes can be useful is when numerous organizations want to anonymously share sensitive data, such as information regarding cyber security breaches. When an organization (e.g. a bank) identifies its computer systems were breached, it would not want this information to leak out. However, it would like to know if it was a target of a pinpointed attack or if similar computer systems of other organizations were breached as well, as this will affect the organization's reaction. A scheme that allows anonymous reporting of cyber attacks, with the ability to break anonymity in extreme cases (e.g. faulty reports), can be used. Another application of such a scheme is a log of keycard access to restricted areas that permit entry to members of a specific group only, when it is inappropriate to track movements of individual members. Again, in extreme cases such as faulty reports there is a need to retrospectively break anonymity and identify the specific member that entered the restricted zone at a specific time.

To resolve this and other privacy related problems, Chaum and Van Heyst introduced group signatures (Chaum and Van Heyst, 1991). Their scheme enables members of a group to create indistinguishable message signatures, i.e. they provide unlinkability and anonymity within members of the group. A special entity, the group manager, is responsible for identifying the group member that originated the signature. Later works on group signatures provided more efficient constructions based on different assumptions, but usually achieve the same goals. See Related Work (Section 1.4) for further details.

A close observation reveals that some security features of group signatures are not vital in scenarios such as the cyber security database or the keycard access log given above:

- Group signatures are publicly verifiable. However, in the given scenarios only a single entity (the cyber security database or the keycard access system) should be able to verify authentication as no other entity is given a signature.
- Many group signature schemes require that the group manager or the recipient is unable to impersonate members (i.e. create signatures on their behalf). We argue that in the given scenarios as well

as many others this requirement is not needed: Since there is a single group manager as well as a single recipient, trusted parties can be found for both. A national cyber security database for banks managed by the government or a keycard access system run by the army to restrict personnel access to sensitive military zones are such examples. The government or the army are either trusted or have enough power to frame innocent group members without the use of such a system.

1.1 Entities

In our massive group message authentication schemes the following entities are involved:

Recipient The central entity to which messages are sent. When a new message is received, the recipient validates the message was originated by a group member before saving it. In extreme cases (e.g. when a fraudulent message is found) the recipient will send a message it received to the group manager in order to identify the message originator.

Group Members Users that are members of the group. They send messages to the recipient.

Group Manager Manages the group by adding and removing members to or from the group. In addition, the group manager can revoke anonymity and retrospectively reveal the identity of the originator of a given message.

1.2 Contribution

Motivated by the observation that a fully featured group signature scheme is not needed in many applications, we present efficient massive group message authentication schemes that allow anonymity revocation. The schemes vary in the exact trade-offs between efficiency and the trust given in the group manager or the recipient. We also compare the security and efficiency of our schemes with those of related schemes.

1.3 Requirements

The requirements for revocable-anonymity group message authentication schemes are:

Authentication When receiving a report, the recipient should be able to verify that the originator of the report is a group member.

Anonymity Given a report, the recipient should not be able to identify the specific group member that originated the report.

Anonymity Revocation The group manager should be able to retrospectively identify the originator of a given report. This requirement can be defined by any or all the following requirements:

- **Against Peers** Reports sent by group members will not be verified by the recipient unless the group manager is able to identify the true originator of the report.
- **Against the Group Manager** Reports sent by the group manager on behalf of honest (non-colluding) group members will not be verified by the recipient.
- **Against the Recipient** The recipient will not be able to send a report to the group manager such that the manager will identify it as originating from an honest (non-colluding) group member.

Traceability Given the identity of a specific group member, the manager and the recipient can together trace all reports originating from that member, and without revealing the identities of the originators of other reports.

Efficiency Scheme operations should have high efficiency. In particular, they should avoid or require a few group operations (e.g. modular exponentiations) as possible. In addition, when tracing all reports originated from a given group member, the recipient should be able to trace the reports distributively using untrusted agents, i.e. these agents will not be given the recipient's secret keys.

Regarding efficiency, it seems desirable that in addition to low computational costs (as discussed above) the group manager will be able to work offline, i.e. will not have to participate in every report sent, but rather only at offline intervals (e.g. at the initialization phase). See Section 6 for a detailed efficiency analysis and comparison to related work.

1.4 Related Work

Group signature schemes, in which a group member can sign messages on behalf of the group without revealing the signer's identity, were first introduced by Chaum and Van Heyst (Chaum and Van Heyst, 1991). Later, Bellare et al. (Bellare et al., 2003) gave a more provable-security oriented formal definition. They introduced the *full-anonymity* and *full-traceability* security requirements and showed they imply many other security requirements mentioned in the literature. They also provided a construction of a group signature scheme assuming only the existence of trapdoor permutations, and proved the security of the scheme in the standard model. The group is assumed to be static, meaning the number and identi-

ties of group members cannot change after the initial setup. The sizes of all keys depend logarithmically on the number of group members. Bellare et al. (Bellare et al., 2005) continued to formalize a definition and provide a construction for partially dynamic groups in which members join (but not leave) the group over time.

Additional works present efficient schemes, but rely on non-standard pairing-based assumptions (e.g. (Ateniese et al., 2005; Boyen and Waters, 2006)) or are secure in the random oracle model (such as (Camenisch and Groth, 2005)), which is not sound (Canetti et al., 2004).

Kiayias et al. (Kiayias et al., 2004) were the first to introduce the privacy primitive *traceable signatures*, which enables tracing of all signatures of a single group member efficiently and without violating privacy of signatures that do not belong to that member. Based on that work and on bilinear pairing, Choi et al. (Choi et al., 2006) present a more efficient traceable signature scheme using shorter signatures.

In their paper, Przydatek and Wikström (Przydatek and Wikström, 2010) observe that some features of group signatures, such as the public verifiability of signatures, are not necessary in many applications. They present a relaxed notion of group signatures and provide both generic and concrete implementations for it. In our paper we follow the same path: We argue that some observed group signature features are nonessential, and provide implementations for a relaxed notion that renounces the nonessential features. Nevertheless, our work differs in many aspects from Przydatek and Wikström, as shown in Table 1 (regarding security requirements) and in Section 6 (regarding efficiency).

Cheng et al. (Cheng et al., 2011) present an interactive and efficient group signature scheme in which an Opener (i.e. the group manager that breaks anonymity) is actively involved in every signature (i.e. the Opener is online when signing messages). The advantages of their scheme are the efficiency and convenience of using regular signature schemes such as RSA signatures along with the straightforward way for members to join or leave the group. The disadvantages are the need to interact with an online Opener for each signature. We show in Section 2 how to modify their scheme in order to meet the traceability requirement.

2 ONLINE MANAGER SCHEME

In their paper, Cheng et al. (Cheng et al., 2011) present an interactive and efficient signature scheme.

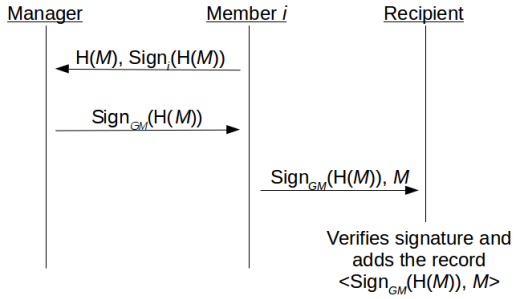

Figure 1: Online manager scheme.

We show below that although their scheme does not satisfy the traceability requirement, it can be easily modified in order to satisfy it with minor performance overhead. When storing $(i, H(M), t_i)$ in the signature table (as stated in their *GSig* algorithm), the Opener can sort the data by the values of $H(M)$ and also store an index of the data according to the group member identifier i. This will allow the Opener, when receiving a message by the Database (i.e. the recipient), to not only identify the originator of the report (by finding the record containing the value $H(M)$), but also to single out all hash values of reports that originated from that same member. These values can be given to the Database. The Database, which is also modified to store each report with its hash value and sort the records according to these values, can efficiently find all relevant reports, effectively identifying all reports originating from the aforementioned member.

Based on this scheme, we present in Figure 1 a new scheme that fits the needs of a single recipient serving multiple members of a specific group. In this scenario group members send reports anonymously to the recipient while the recipient can verify only that a group member sent a report, but cannot identify the member. The scheme is as follows:

1. A group member sends a hash value of the report with her[1] signature on that value to the manager.
2. The manager signs the hash value and returns it to the member.
3. The member sends the report along with its manager-signed hash value to the recipient.
4. After verification of the manager's signature, the report and the signature are added as a record to the recipient's database.

If anonymity revocation is needed, the recipient can send the hash value of a report to the manager. The manager will consult his table, in which he stored

[1]For clarity of reference, we use she, her, etc. to refer to the group members, and he, his, etc. to refer to other entities (the group manager and the recipient).

Table 1: Security requirements comparison.

Security Requirements	**Przydatek and Wikström, 2010**	**Cheng et al., 2011**	**Online Manager Scheme (Section 2)**	**Offline Manager Schemes (Sections 3 and 4)**	**Enhanced Security Scheme (Section 5)**
Anonymity	Yes				
Authentication	Yes[1]	Yes	Yes[1]	Yes[1]	Yes
Traceability	No	No	Yes	Yes	Yes
Anonymity revocation					
- Against peers	Yes				
- Against the group manager	Yes	Yes	No	No	Yes
- Against the recipient	No	Yes	Yes	No	Yes

$(H(M), i)$ pairs, and reveal the identity of the report's originator. He will then single out all hash values of reports originating from that member and forward them to the recipient, who will single out all reports received from that member.

2.1 Evaluation

As with the scheme presented in (Cheng et al., 2011), the advantages of this scheme is the efficiency obtained by using regular digital signatures and ease of adding or removing group members. Furthermore, we do not rely on bilinear pairings, but at the cost of security. Specifically, this scheme assumes a trusted group manager. As mentioned, our scheme also has the traceability requirement. The disadvantage is the need for an online manager to participate in every report.

3 OFFLINE MANAGER SCHEME

Figure 2 presents an offline manager scheme in which group members send reports to the recipient along with one-time tokens. The tokens are created and handed out to the members by the manager at the offline phase, and each token consists of a random string of bits and its corresponding MAC. The MAC is computed using a secret key known only to the manager and the recipient. When receiving a report and its attached token the recipient acknowledges the report if and only if he can authenticate the MAC. By authenticating it the recipient is convinced that the manager issued this token and will be able to identify the member that received it and sent it with her report. The scheme is as follows:

[1]Assuming a trusted group manager.

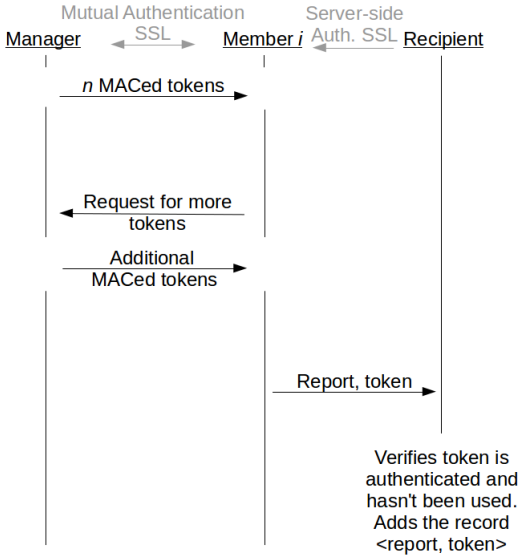

Figure 2: Offline manager scheme.

At the beginning of each time period t the following steps are taken:

1. the manager and the recipient agree on a new secret MAC key k_t.
2. The manager sends each member n tokens, each consists of a random string and its MAC, generated using k_t.

During time period t group members may do any of the following steps:

1. Request more tokens from the manager (who will respond offline, i.e. at some point in the future, but during the current time period t).
2. Send a report:
   (a) Send a (report, token) pair to the recipient.

(b) The recipient will authenticate the MAC in the token and validate it was not used before. Then it will add the (report, token) pair as a record.

Notes about token reuse: Firstly, schemes in which a single token can be used m times with $m > 1$ could be constructed, but for simplicity we consider only one-time token schemes. Secondly, before acknowledging a report the recipient checks whether the token was used before. Actually, a more optimistic approach can be used: When a (report, token) pair is received the recipient adds it as a record without further inspection. Additionally, the recipient performs occasional searches for tokens that are used multiple times. If such a token is found the recipient can contact the manager in order to reveal the identify of the misbehaving member.

Adding or removing members from the group is done as follows: To join the group a user needs only to contact the manager, which will give her tokens. To remove a member from the group the manager needs only to send the recipient all tokens (actually, all random string portions of the tokens) sent to that member in the current time period t. The recipient will not accept reports with the corresponding tokens until the end of t. Note that at the beginning of time period $t+1$ the manager and the recipient will exchange a new key k_{t+1}, rendering tokens from previous time periods obsolete.

In order to prevent eavesdropping or modification of tokens confidentiality and integrity of messages should be kept, and group members and the manager should be able to authenticate each other when communicating. Additionally, when group members communicate with the recipient they should be able to authenticate the recipient (but not vice versa, else group members' privacy is lost). These requirements can be achieved by using the SSL protocol throughout all communications, with certificates for all parties: in communications between the manager and the group members both sides should present certificates, while in communications between a group member and the recipient only the recipient should present a certificate. Note that anonymous communication channels are assumed (e.g. the recipient cannot identify a group member by her IP address).

3.1 Evaluation

This offline manager scheme is efficient, does not require the manager to participate in every signature, and requires less communication than the online manager scheme. However, in this scheme the manager and the recipient should be trusted since each of them can frame members. The manager can lie when queried about a report's originator. The recipient can frame group members by modifying reports (e.g. inserting inappropriate content) but without modifying the related tokens. These modified reports would look to the manager as if originated by an honest group member.

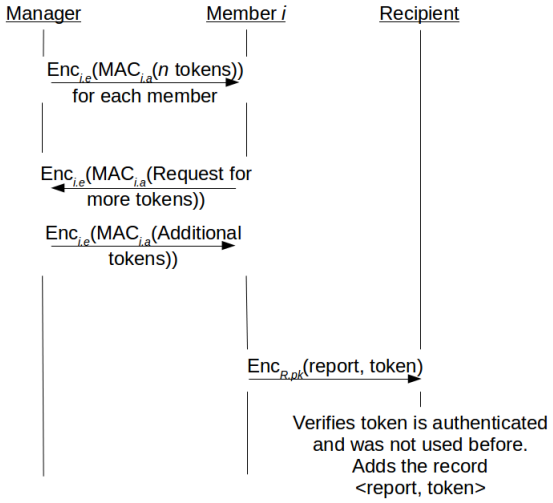

Figure 3: Offline manager scheme without SSL.

4 OFFLINE MANAGER SCHEME WITHOUT SSL

The creation of SSL connections for every report (or keeping alive previously used SSL connections for long durations) may introduce a substantial overhead. Thus, we present an improved version of the aforementioned offline manager scheme (Section 3) which does not use SSL. This scheme is shown in Figure 3.

The scheme is as follows:

1. Initialization: Each member i shares a symmetric encryption key $i.e$ and a MAC key $i.a$ with the manager. All communication between group members and the manager are encrypted and authenticated using these keys.

   When users join the group they negotiate keys with the manager. This can be done using common methods such as the Diffie-Hellman key exchange (Diffie and Hellman, 1976).

2. Setup: At the beginning of each time period t the following steps are taken:

   (a) the manager and the recipient agree on a new secret MAC key k_t.

   (b) The manager sends n tokens for each member i. Each token consist of a random string and

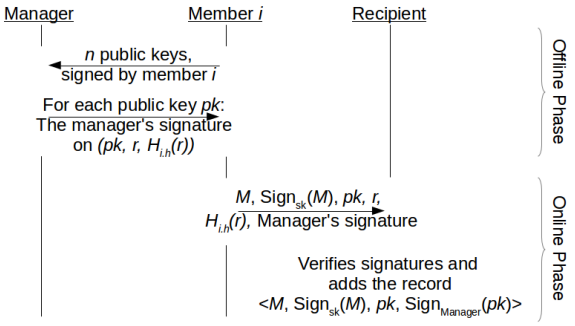

Figure 4: Enhanced security scheme.

its MAC, generated using k_t. The tokens are encrypted and MACed using keys shared with member i.

3. During time period t group members may do any of the following steps:
   (a) Request more tokens from the manager (who will respond offline, i.e. at some point in the future, but during the current time period t).
   (b) Post a report:
      i. Send a (report, token) pair to the recipient.
      ii. The recipient will verify the MAC is correct and the token has not been used before. Then it will add the (report, token) pair as a record.

   All reports sent from group members to the recipient are encrypted using the recipient's public encryption key $R.pk$. In order to achieve integrity, provably secure non-malleable encryption schemes should be used (such as the Cramer-Shoup scheme (Cramer and Shoup, 1998)). If the recipient replies to members (e.g. to state that a report was added) integrity of the reply can be achieved by the recipient MACing the reply with a one-time MAC key that was previously created by the member and sent to the recipient alongside the report.

4.1 Evaluation

As can be seen, this scheme achieves confidentiality, integrity and authenticity as the Offline Manager Scheme (Section 3) without the use of the SSL protocol.

5 ENHANCED SECURITY SCHEME

When the manager or the recipient might frame group members a more secure scheme should be used, such as the scheme shown in Figure 4. It ensures a higher degree of security but is less efficient. The scheme is as follows:

1. Offline phase:
   (a) Each member creates n (public key, secret key) one-time signature pairs to be used later. These pairs can be created efficiently using schemes such as presented by Lamport (Lamport, 1979). The member then sends all public keys from these pairs to the manager, signed with her private signing key $i.sk$.
   (b) If the member is a new member the manager creates a key $i.h$ and stores it along with the member's identity i.
   (c) The manager sends back a list containing, for each received public key pk, the values $(pk, r, H_{i.h}(r))$ signed by his private signature key; r is a random number and $H_{i.h}(r)$ is the value of a cryptographic hash function computed on r using the key $i.h$.
2. Online phase:
   (a) To send a report, a member sends the following to the recipient: M and $\text{Sign}_{\text{sk}}(M)$) - the report and its signature created using a previously unused secret key from a one time signature pair; pk - the public key corresponding to the used secret key; the values r and $H_{i.h}(r)$; and $\text{Sign}_{\text{Manager}}(pk, r, H_{i.h}(r))$ - the manager's signature on the public key, the random number and the hash value.
   (b) After validating both the manager's signature on pk (using the manager's public key) and the member's signature on the report (using the given key pk) the recipient adds the record

$$\langle M, \text{Sign}_{\text{sk}}(M), pk, r, H_{i.h}(r), \text{Sign}_{\text{Manager}}(pk, r, H_{i.h}(r)) \rangle \quad (1)$$

When there is a need to identify the originator of a report M the following steps are taken:

1. The recipient sends the manager the values $\langle M, \text{Sign}_{\text{sk}}(M), pk, r, H_{i.h}(r) \rangle$ that correspond to the report M.
2. After validating the signature (using the given pk), the manager tries all member keys $i.h$ until the one corresponding to the received $r, H_{i.h}(r)$ is found. He then outputs i as the identity of the report's originator and the value $\text{Sign}_{i.sk}(pk)$, i.e. i's signature on pk, as a proof of the originator's identity.

   If there is a need to trace all reports generated by this member the manager sends the recipient $\langle i, \text{Sign}_{i.sk}(pk), i.h \rangle$.

3. If all i's reports are to be traced the recipient validates $\text{Sign}_{i.sk}(pk)$ (using i's publicly known signature verification key). Then, for each record the recipient computes the value $H_{i.h}(r)$ using the received $i.h$ and the value r in the report[2]. Reports whose stored $H_{i.h}(r)$ value can be recomputed using H, r, and $i.h$ are identified as belonging to user i.

5.1 Security

Lemma 1. *The given scheme is secure against framing of honest users.*

Proof. Proof sketch (A complete proof would appear in this paper's full version): When a report is received by the recipient it is validated both by the member's one time signature on the report and by the manager's signature on the public key pk of the one time signature. So in order to forge a report that would be validated by the recipient an attacker (who is not the manager) needs to forge the manager's signature. Such an attacker can be reduced to an attacker that breaks the security of digital signatures.

Even if the manager maliciously sent a report on behalf of a member and the recipient accepted the report the member would not be held responsible. When a group member is declared to be the originator of a specific report the claim is proven by the one time signature on that report and the member's signature (using her private key $i.sk$) on the public key corresponding to that one time signature. In order to convince others that a report originated from an honest member an attacker would have to forge the member's digital signature on the public key. Again, such an attacker can be reduced to an attacker that can brake the security of standard digital signatures. □

6 EFFICIENCY COMPARISON

An efficiency comparison of the presented and related schemes is shown in Table 2. The table compares whether the group manager needs to actively participate in each report sent (e.g. personally sign each report), and the number of group operations (e.g. modular exponentiations or bilinear pairings) needed in basic scheme operations.

[2]This computation can be distributed between many agents, each with access to the reports stored by the recipient and to $i.h$.

7 CONCLUSIONS

In this paper we presented various practical schemes for a relaxed notion of group signatures, allowing a single recipient to validate that received messages were sent by group members while preserving members' anonymity. If needed, a trusted group manager can retrospectively break anonymity and reveal the identity of the originator of a message. In addition, the group manager is able to efficiently trace all messages originated from a given group member without affecting the anonymity of other messages. The security and efficiency of the schemes proposed are compared to state of the art, and are shown to be more efficient.

ACKNOWLEDGEMENTS

We thank the following organizations for financially supporting this research: The Ministry of Science and Technology, Israel, and the RSA division of EMC corporation. In addition, this work is part of the Kabarnit-Cyber Consortium (2012-2014) under Magnet program, funded by the chief scientist of the Israeli Ministry of Industry, Trade and Labor.

REFERENCES

Ateniese, G., Camenisch, J., Hohenberger, S., and De Medeiros, B. (2005). Practical group signatures without random oracles. *EUROCRYPT06*.

Bellare, M., Micciancio, D., and Warinschi, B. (2003). Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, pages 614–629.

Bellare, M., Shi, H., and Zhang, C. (2005). Foundations of group signatures: The case of dynamic groups. *Topics in Cryptology–CT-RSA 2005*, pages 136–153.

Boyen, X. and Waters, B. (2006). Compact group signatures without random oracles. *Advances in Cryptology-EUROCRYPT 2006*, pages 427–444.

Camenisch, J. and Groth, J. (2005). Group signatures: Better efficiency and new theoretical aspects. *Security in Communication Networks*, pages 120–133.

Canetti, R., Goldreich, O., and Halevi, S. (2004). The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594.

Chaum, D. and Van Heyst, E. (1991). Group signatures. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, pages 257–265. Springer-Verlag.

Table 2: Efficiency comparison.

	Przydatek and Wikström, 2010	**Cheng et al., 2011**	**Online Manager Scheme (Section 2)**	**Offline Manager Scheme (Section 3)**	**Offline Manager Scheme without SSL (Section 4)**	**Enhanced Security Scheme (Section 5)**
Manager participates in every report	No	Yes	Yes	No	No	No
Number of Group Operations (e.g. modular exponentiations)						
Creation of group sign. or authentication token	5	4 (1 by member, 3 by manager + 1 bilinear pairing)	2	0	0	2 (1 by member and 1 by manager)
Verification of group sign. or authentication token	6	0 (1 bilinear pairing)	1	0	0	1
Commun. between member and recipient	Member - 19, recipient - 17, (full authentication protocol, including Zero Knowledge Proofs)	0	0	0	Member - 5, recipient - 6, (encryption and decryption by the Cramer-Shoup scheme (Cramer and Shoup, 1998))	Member - 0, recipient - 1

Cheng, X., Yang, C., and Yu, J. (2011). A new approach to group signature schemes. *Journal of Computers*, 6(4):812–817.

Choi, S. G., Park, K., and Yung, M. (2006). Short traceable signatures based on bilinear pairings. In *IWSEC*, pages 88–103.

Cramer, R. and Shoup, V. (1998). A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology-CRYPTO'98*, pages 13–25. Springer.

Diffie, W. and Hellman, M. E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654.

Kiayias, A., Tsiounis, Y., and Yung, M. (2004). Traceable signatures. In *Advances in Cryptology-EUROCRYPT 2004*, pages 571–589. Springer.

Lamport, L. (1979). Constructing digital signatures from a one-way function. Technical report, Technical Report CSL-98, SRI International.

Przydatek, B. and Wikström, D. (2010). Group message authentication. *Security and Cryptography for Networks*, pages 399–417.