Remarks on Saeednia's Identity-based Society Oriented Signature Scheme with Anonymous Signers

Guilin Wang, and Bo Zhu

Infocomm Security Department,
Institute for Infocomm Research.
21 Heng Mui Keng Terrace, Singapore 119613.
http://i2r.a-star.edu.sg/icsd.
{glwang, bozhu}@i2r.a-star.edu.sg

Abstract. Recently, based on Guillou-Quisquater signature scheme, Saeednia proposed an identity-based society oriented signature scheme. However, in this note, we point out that Saeednia's scheme does not satisfy the claimed properties.

Keywords: cryptography, digital signatures, identity-based, attacks.

1 Introduction

Society oriented cryptography was introduced by Desmedt [1]. A society oriented signature is essentially like a single signature except that it is produced by several signers simultaneously. There are two kinds of these schemes: with known signers and with anonymous signers. In this note, we are only interested in the second type schemes, in which a verifier can check the validity of a signature by using a single group public key of an organization but does not know the identities of the co-signers who generated the signature.

As pointed out in [6], there are two difficulties in the design of a society oriented signature scheme with anonymous signers. The first one is that when members leave or join the organization, how to keep the public verification key and the signature verification procedure unchanged. In other words, how can we change or eliminate some of those secret shares or add some new ones, while the organization public key should remain the same? The other one is that when a user participates in several organizations at the same time, he wishes to have the same secret key. This problem is more significant when we design an identity-based society oriented signature scheme, where the public key is simply the real identity of the corresponding entity. Therefore, the question is how a number of users can sign messages together with their own individual secret keys in relation with a predetermined public key (the organization identity)?

Based on the Guillou-Quisquater signature [5] and the notion of identity-based crtyptosystems [7], Saeednia constructed an identity-based society oriented signature scheme to answer these questions [6].

Undoubtedly, it is highly desirable to find out solutions to the above questions. However, in this note, we point out that Saeednia does not solve these two questions in the essence. More specifically, we show that some public information in his scheme can be exploited by malicious members, organizations or the clerk such that in several scenarios, they can get the individual secret keys of other users or organizations. At the same time, we note that there is a simple improvement of his scheme, i.e., requiring that the clerk is a *trusted* and *on-line* party (On-line means that the clerk is needed for generating any signature on behalf of the organization). However, our later discussion shows that this improvement does not have much sense.

The rest of this note is organized as follows. We first review the Guillou-Quisquater signature [5] and the Saeednia's identity-based society oriented signature scheme in section 2 and section 3, respectively. Then, our remarks are presented in section 4. Finally, the conclusion is given.

2 Review of Guillou-Quisquater Signature Scheme

Like all identity-based cryptosystems [7], Guillou-Quisquater signature scheme [5] assumes the existence of a trusted authority that is responsible of generating the secret keys for all users, linked to their identities. Before generating the keys, the authority chooses the following system parameters:

1. An integer $n = pq$ such that $p = 2p' + 1$, $q = 2q' + 1$, and p, q, p', q' all are large primes;
2. A prime $v < \min\{p, q\}$, with some prescribed size (about 72 bits); and
3. An one-way hash function $h(\cdot)$ such that for any input m, $|h(m)| < v$.

Now, the trusted authority publishes (n, v) as his public key, but keep (p, q) as his private key. When a user U, with an identity string J that is half shorter than n, wants to join the system, the trusted authority computes and sends the following x to U securely

$$x = I^{-v^{-1}} \bmod n. \tag{1}$$

Where $I = \mathrm{Red}(J)$ is a number as large as n, $\mathrm{Red}(J)$ is the concatenation of J and a redundancy depending on J [5], and v^{-1} is computed modulo the trap-door information $\lambda(n) = 2p'q'$.

To generate a signature (d, z) for a message m, the user U first chooses a random integer $r \in_R Z_n$ and computes

$$t = r^v \bmod n, \quad d = h(t, m), \quad \text{and } z = r \cdot x^d \bmod n.$$

To verify the validity of a signature (d, z) for a message m, a verifier check whether

$$d \equiv h(z^v \cdot I^d \bmod n, m).$$

3 Reivew of Saeednia's Scheme

Suppose that an organization G with an identity J_G wants to enable k members U_i's, with identities J_i's, $i = 1, 2, \cdots, k$, to sign messages together on behalf of the organization. Suppose also that these k members already have their individual secret keys x_i's computed from equation (1). The organization sends the identities of these members to the trusted authority, which computes

$$I_c = I_G \cdot (I_1 \cdot I_2 \cdots I_k)^{-1} \bmod n, \quad \text{and } x_c = I_c^{-v^{-1}} \bmod n, \tag{2}$$

and gives back x_c to the organization.

In [6], it is pointed that x_c can only be calculated by the authority, but once computed may be made public. Note that from equation (2), we have

$$x_c \cdot x_1 \cdots x_k = I_G^{-v^{-1}} = x_G \bmod n. \tag{3}$$

To sign a message m, each member U_i, $i = 1, 2, \cdots, k$, first chooses a random number $r_i \in_R Z_n$ and computes $t_i = r_i^v \bmod n$ and sends it to all other members. When all t_j's are available, each U_i computes the following values T, d and z_i:

$$T = t_1 \cdot t_2 \cdots t_k \bmod n, \quad d = h(T, m), \quad \text{and} \quad z_i = r_i \cdot x_i^d \bmod n. \tag{4}$$

Then, each U_i sends (t_i, z_i) to a clerk (or called a designated combiner). The clerk first computes T and d in the same way as the k members did, and verifies each individual signature (t_i, z_i) by checking whether $t_i \equiv z_i^v \cdot I_i^d \bmod n$. If all (t_i, z_i)'s are valid, the clerk outputs a pair (Z, d) as the organization's signature for message m, where Z is calculated as follows:

$$Z = x_c^d \cdot z_1 \cdot z_2 \cdots z_k \bmod n. \tag{5}$$

To verify a signature pair (Z, d) for message m, a verifier checks whether

$$d \equiv h(Z^v \cdot I_G^d \bmod n, m).$$

It is easy to see that if all individual signatures are correct and Z is computed following the protocol, the verifier will accept the multisignature (Z, d) as valid.

4 Our Remarks

In [6], Saeednia claimed that his scheme has the following properties:

- Each user may participate in different groups (within the same organization or not) with his individual secret key (that is linked to his identity and is also used for own signatures).
- Furthermore, if some members of a given group leave that group or if some new members join the group, the remaining members can still sign messages with their unique secret keys, without even being aware of any change in the structure of the group. The only value modified is x_c, which is public and is only used by the clerk.

However, we will show that these claims are not true because the public value x_c in equation (3) can be exploited by malicious entities (members, organizations or the clerk). The result is that, in several scenarios, they can get the individual keys (or related sensitive information) of other users or organizations, and then forge regular signatures on behalf of the victims.

First of all, we note that the value of $x_j/x_i \bmod n$ should not be revealed to anybody, where x_i and x_j are the individual secret keys of two users U_i and U_j. Otherwise, U_i and/or U_j will become victims. To see the problem in details, let $l = x_j/x_i \bmod n$. Then, two attacks can be mounted as follows:

Attack 1. If user U_i knows the value of $l = x_j/x_i \bmod n$, he can derive the individual secret key x_j of user U_j from $x_j = lx_i \bmod n$, and then forges signatures for arbitrary messages on behalf of user U_j.

Attack 2 If it is known that user U_i has signed a valid signature (z, d) for a message m, anybody (not necessarily a member in the system) can forge a signature $(\bar{z}, d)$ for the same message but on behalf of U_j by computing $\bar{z} = z \cdot l^d \bmod n$. Since $\bar{z}^v \cdot I_j^d = z^v \cdot l^{dv} \cdot x_j^{-dv} = z^v \cdot (l^{-1}x_j)^{-dv} = z^v \cdot x_i^{-dv} = z^v \cdot I_i^d \bmod n$, we have $d = h(z^v \cdot I_i^d \bmod n, m) = h(\bar{z}^v \cdot I_j^d \bmod n, m)$. Therefore, $(\bar{z}, d)$ is a valid signature for the same message m on behalf of user U_j.

Now, we point out the following scenarios in which malicious entities can get the individual secret keys of other entities or the value of the ratio of two entities' individual secret keys:

1) *A member leaves a group.* If $\{U_1, \cdots, U_k\}$ is a representative group of an organization G, and x_c is the public value such that $x_c \cdot x_1 \cdots x_k = x_G$. Later, member U_1 leaves the group, and a new $\bar{x}_c$ will be published such that $\bar{x}_c \cdot x_2 \cdots x_k = x_G$. Anyone hence knows that $x_1 = \bar{x}_c/x_c \bmod n$, i.e., user U_1's individual secret key x_1 is revealed. However, in Saeednia's scheme, a user leaves the group does not means that he quits the system, because he still uses his individual secret key to generate his individual signatures (see Saeednia's first claim listed in the beginning of this section). Therefore, by using x_1, anybody can forge individual signatures on behalf of user U_1.
2) *A member joins a group.* Similarly, when a new member joins a representative group of an organization G, his individual secret key will be revealed.
3) *Different groups represent the same organization.* If $\{U_1, \cdots, U_{k-1}, U_k\}$ and $\{U_1, U_2, \cdots, U_{k-1}, U_{k+1}\}$ are two representative groups of an organization G, then from $x_c \cdot x_1 \cdots x_{k-1} \cdot x_k = x_G$ and $\bar{x}_c \cdot x_1 \cdots x_{k-1} \cdot x_{k+1} = x_G$, anybody knows that $x_c/\bar{x}_c = x_{k+1}/x_k \bmod n$, since x_c and $\bar{x}_c$ are public information. Therefore, user U_k (and user U_{k+1}) can mount the above attack 1, and anybody can mount the above attack 2.
4) *The same group represents different organizations.* If $\{U_1, \cdots, U_k\}$ is a representative group of both organization G and organization $\bar{G}$, then from $x_c \cdot x_1 \cdots \cdot x_k = x_G$ and $\bar{x}_c \cdot x_1 \cdots \cdot x_k = x_{\bar{G}}$, anybody knows that $x_c/\bar{x}_c = x_G/x_{\bar{G}} \bmod n$. Therefore, similar to the above case, organization G (and organization $\bar{G}$) can mount the above attack 1, and anybody can mount the above attack 2.

5) *Representative groups are renewed.* If at first, $\{U_1, \cdots, U_k\}$ is a representative group of an organization G, but later all of them leave (may be deleted by G), and G selects his new representative groups. In this setting, $\{U_1, \cdots, U_k\}$ should be *unable* to generate signature on behalf of the organization G anymore. However, from the above description of Saeednia's scheme, these k members can also sign on the behalf of G if they store the value x_c. The situation is much worse: these malicious entities can not be identified even if they generated a signature which results a bad effect on the organization, because Saeednia's scheme is an anonymous society oriented signature.

To overcome the above attacks, one natural improvement is to require that only the clerk knows the value of each x_c. Furthermore, we have to assume that the clerk is a trusted party, otherwise he can also mount above attacks and reveal other entities's individual secret keys or other sensitive information to some third party who has interests in them. In addition, note that x_c^d is always needed to generate any signature on behalf of the organization. So, we have to assume that the clerk is on-line whenever a representative group wants to sign a message on behalf of the organization. However, this improvement does not have much sense, because the existence of a trusted party is one of the most strongest assumptions in information security field, and one motivation of society oriented cryptosystems is to eliminate trusted parties or at least reduce our trust on them [1]. Furthermore, if there is a trusted and on-line clerk, the organization G can simply enable the clerk to sign messages on behalf of himself by sending his secret key x_G to the clerk, instead of selecting a group of members to represent himself.

At the same time, we note that He [4] also pointed out several attacks against two multisignature schemes [3, 2], which are based on the fact that the untrusted clerk in these two schemes has much more ability than usual group members.

5 Conclusion

In this note, by demonstrating several attacks, we showed that Saeednia's identity-based society oriented signature scheme [6] does not satisfy the claimed properties.

References

1. Y. Desmedt. Society and group oriented cryptography: A new concept. In: *Advances in Cryptology, Proceedings of Crypto'87, Lecture Notes in Computer Science, Vol. 293.* Springer-Verlag, 1988, pp. 120-127.
2. S.J. Hwang, C.Y. Chen, C.C. Chang. An encryption/multisignature scheme with specified receiving groups. *Comput. Systems Sci. Engrg.*, 1998, 13(2): 109-112.
3. C.S. Laih, and S.M. Yen. Multisignature for specified group of verifiers. *J. Inform. Sci. Engrg.*, 1996, 12(1): 143-152.
4. W.-H. He. Weaknesses in some multisignature schemes for specified group of verifiers. *Information Processing Letters*, 2002, 83: 95-99.

5. L. Guillou, and J.J. Quisquater. A paradoxical identity-based signature scheme resulting from zero-knowledge. In: *Advances in Cryptology, Proceedings of Crypto'88, Lecture Notes in Computer Science, Vol. 403*. Springer, Berlin, 1989, pp. 216-231.
6. S. Saeednia. An identity-based society oriented signature scheme with anonymous signers. *Information Processing Letters*, 2002, 83: 295-299.
7. A. Shamir. Identity-based cryptosystem based on the discrete logarithm problem. In: *Advances in Cryptology, Proceedings of Crypto'84, Lecture Notes in Computer Science, Vol. 196*. Springer-Verlag, 1985, pp. 47-53.