Threshold Proxy Signature Schemes

Kan Zhang

Cambridge University Computer Laboratory
Pembroke Street, Cambridge CB2 3QG, UK
Email: kz200@cl.cam.ac.uk

Abstract. Delegation of rights is a common practice in the real world. Proxy signature schemes have been invented to delegate signing capability efficiently and transparently. In this paper, we present a new nonrepudiable proxy signature scheme. Nonrepudiation means the signature signers, both original and proxy signers, cannot falsely deny later that he generated a signature. In practice, it is important and, sometimes, necessary to have the ability to know who is the actual signer of a proxy signature for internal auditing purpose or when there is abusing of signing capability. The new nonrepudiable proxy signature scheme also has other desirable properties, such as proxy signature key generation and updating using insecure channels. We also show how to construct threshold proxy signature schemes with an example. Threshold signatures are motivated both by the need that arises in some organizations to have a group of employees agree on a given message (or a document) before signing it, as well as by the need to protect signature keys from the attack of internal and external adversaries. Our approach can also be applied to other ElGamal-like proxy signature schemes.

1 Introduction

Delegations of various kinds are common practice in society. One of them is delegating signing capabilities within organizations. If a manager of a company goes on holiday, he has to delegate to his deputy the capability to sign on behalf of the company. A solution in the paper based world is to have a corporate seal. A corporate seal represents the organization, not the person who has authority to use the seal. A major difference between signing by hand and signing by seal is that seal is transferable. This transferability allows efficient and flexible delegation of signing capability, and therefore reduces costs.

The requirement here is transparent delegation, which means the customers of the company are not affected. The signatures presented to them are the same as before and they can verify the signatures using the same process. It will be economically infeasible for a big company to notify all its customers each time there is a personnel change in the company. Unfortunately, basic digital signature schemes, like signing by hand, are not transferable. Digital signature schemes rely on a secret signature key which only the certified person knows. If this secret key is delegated to another person, it can no longer be identified with that person and, hence, the underlying assumption of digital signature schemes is broken. What we need is a way to delegate the signing capability without revealing the secret key such that the recipients can verify the proxy signatures from a proxy signer using the original secret signature key with minimum changes to the verification process.

To facilitate delegation of signing capability in the electronic world, proxy signature schemes has been proposed. Mambo, Usuda and Okamoto proposed a proxy signature scheme, hereafter referred to as MUO scheme, based on discrete logarithm for partial delegation of signing capability [13]. In a partial delegation, a proxy signature key s created from the original signature key x is given to a proxy signer. Signatures created using the proxy signature key can be verified using a modified verification equation in such a way that the verifier can be convinced that the signature is from an authorised party of the original signer. Signatures created by a proxy signer are distinguishable from those by the original signer. The verifier needs to verify one signature only and only the public key of the original signer is required. Therefore, compared with delegation by warrant, partial delegation is more efficient and transparent requiring less changes to the original verification process.

However, MUO scheme doesn't provide nonrepudiation of proxy signatures. In MUO scheme, while a valid proxy signature cannot be disavowed by the original or the proxy signer, it is impossible to decide who is the actual signer of a proxy signature since both the original signer and the proxy signer know the proxy signature key. In practice, it is important, and sometimes necessary, to have the capability to know who is the actual signer of a proxy signature for internal auditing purpose or when there is abusing of signing capability. We call this property *Nonrepudiation*. Another drawback of MUO scheme is that it requires a secure channel to pass the proxy signature key from the original signer to the proxy signer. Anyone who can intercept this proxy signature key can impersonate the proxy signer.

In [18], we proposed a nonrepudiable proxy signature scheme which overcomes these shortcomings. The nonrepudiation property is achieved through a partially blind proxy signature key generation protocol adapted from blind Nyberg-Rueppel signature scheme by Camenisch et al. [3]. In this paper, we present another variant of the nonrepudiable proxy signature scheme. It has similar properties to the previous one. We also show how to convert regular proxy signature schemes into threshold proxy signature schemes.

In threshold proxy signature schemes, the secret proxy signature key is shared by a group of n proxy signers. In order to produce a valid proxy signature on a given message m, individual proxy signers produce their *partial signatures* on that message, and then combine them into a full proxy signature on m. Threshold signatures are motivated both by the need that arises in some organizations to have a group of employees agree on a given message (or a document) before signing it, as well as by the need to protect signature keys from the attack of internal and external adversaries.

The paper is organized as follows. In section 2, we give a short description of previous work on which our work is based. Next, we present a second type of nonrepudiable proxy signature schemes. How to convert regular proxy signature schemes into threshold proxy signature schemes is shown in section 4. We draw our conclusion in section 5.

2 Previous work

2.1 Proxy Signature Scheme by Mambo, Usuda and Okamoto

Recently, Mambo, Usuda and Okamoto proposed a proxy signature scheme for partial delegation of signing capability, which has many desirable properties [13]. In this section, we give a brief description of the MUO scheme. Interested readers are referred to [13] for details.

The system parameters consist of a prime p, a prime factor q of $p-1$, and an element $g \in Z_p^*$ of order q. The original signer's private key is a random element $x \in Z_q$, while the corresponding public key is $y = g^x \pmod{p}$. MUO scheme uses the following protocol.

- Step 1. (Proxy key generation) An original signer selects $k \in Z_q$ at random and computes $r = g^k \bmod p$. After that, he calculates proxy signature key $s = x + kr \bmod q$.
- Step 2. (Proxy key delivery) The original signer gives (s, r) to a proxy signer in a secure way.
- Step 3. (Proxy key verification) The proxy signer checks that

$$g^s = yr^r (\bmod\, p) \tag{1}$$

  If (s, r) passes this congruence, she accepts it as a valid proxy signature key. Otherwise, she rejects it and requests another one, or she can simply stops the protocol.
- Step 4. (Signing by the proxy signer) When the proxy signer signs a message m on behalf of the original signer, she computes a signature S_p using the original signature scheme and s as the secret signature key. The proxy signature is (S_p, r).
- Step 5. (Verification of the proxy signature) The verification of the proxy signature is carried out by the same checking operation as in the original signature scheme except for replacing y with $y' = yr^r (\bmod\, p)$.

The following equation can be used as an alternative to equation (1).

$$g^s = y^r r (\bmod\, p) \tag{2}$$

Equation (1) and (2) correspond to a special case (message m equals 1) of Yen-Laih [17] and Nyberg-Rueppel [14] signature schemes, respectively. The authors of MUO scheme stated the following properties.

- *Unforgeability* Only the original signer and the designated proxy signer can create a valid proxy signature.
- *Proxy signer's deviation* A proxy signer cannot create a valid proxy signature not detected as a proxy signature.
- *Verifiability* From proxy signatures, a verifier can be convinced of the original signer's agreement on the signed message.
- *Distinguishability* Valid proxy signatures are distinguishable from normal self-signing signatures.
- *Identifiability* An original signer can determine the proxy signer's identity from a proxy signature.
- *Undeniability* Once a proxy signer creates a valid proxy signature for an original signer, it is not disavowed even by the proxy signer.

2.2 Nonrepudiable Proxy Signature Scheme (NPS1)

In [18], We have shown how to add nonrepudiation to existing proxy signature schemes. Nonrepudiation means the signature signers, both original and proxy signers, cannot falsely deny later that he generated a signature. This property is achieved through a partially blind signature key generation protocol adapted from fully blind Nyberg-Rueppel signature scheme [3]. The new proxy signature scheme fits in the same general framework as MUO scheme. Therefore, the desirable properties of MUO scheme can also be said about our nonrepudiable scheme. In addition, our scheme has some new properties, such as proxy signature key generation and updating using insecure channels. The protocol goes as follows.

1. Original signer selects $\tilde{k} \in Z_q$, computes $\tilde{r} = g^{\tilde{k}} (\bmod p)$, and sends $\tilde{r}$ to Proxy signer.
2. (a) Proxy signer randomly selects $\alpha \in Z_q$, computes
$$r = g^{\alpha} \tilde{r} (\bmod p) \tag{3}$$
   (b) Proxy signer checks whether $r \in Z_q^*$. If this is not the case, he goes back to step (a). Otherwise, he sends r to Original signer.
3. Original signer computes $\tilde{s} = rx + \tilde{k} (\bmod q)$ and forwards $\tilde{s}$ to Proxy signer.
4. Proxy signer computes $s = \tilde{s} + \alpha (\bmod q)$, and check if the following equality holds:
$$g^s = y^r r (\bmod p) \tag{4}$$
   If it holds, Proxy signer accepts s as a valid proxy signature key from Original signer.
5. Proxy signature generation and verification are the same as in MUO scheme.

3 A New Nonrepudiable Proxy Signature Scheme (NPS2)

It is obvious that the above proxy signature key generation process is an adapted version of the blind Nyberg-Rueppel scheme proposed by Camenisch et al. [3] (see Appendix). In [3], two blinding factors, $\alpha \in Z_q$ and $\beta \in Z_q^*$, are used. It is proved that given a valid signature (r, s) and any view V representing the original signer's complete view of an execution of the protocol, i.e. his random coin tosses and all exchanged values, there exists a unique pair of blinding factors $\alpha \in Z_q$ and $\beta \in Z_q^*$. Since the original signer chooses the blinding factors α and β randomly, the full blindness of the signature scheme follows.

To convert the fully blind signature scheme into a partial one, we can use a fixed public known value for β. For simplicity, we choose $\beta = 1 (\bmod q)$. In this case, r, and, therefore mg^{α} are known to the original signer. To further uniquely decide a α and, therefore, s, we need to let message m publicly known. For simplicity, we choose $m = 1$. This reduces the underlying Nyberg-Rueppel scheme (see Appendix) to the one used in MUO scheme (Eq 2). And the original blind Nyberg-Rueppel scheme becomes the one we use in [18].

Another way to generate a partially blind proxy signature key as in [18] is to let $\alpha = 0$ and use β as the only blinding factor. The protocol goes as follows.

1. Original signer selects $\tilde{k} \in Z_q$, computes $\tilde{r} = g^{\tilde{k}} (\bmod p)$, and sends $\tilde{r}$ to Proxy signer.
2. (a) Proxy signer randomly selects $\beta \in Z_q^*$, computes $r = \tilde{r}^{\beta} (\bmod p)$ and $\tilde{m} = r\beta^{-1} (\bmod q)$.
   (b) Proxy signer checks whether $\tilde{m} \in Z_q^*$. If this is not the case, he goes back to step (a). Otherwise, he sends $\tilde{m}$ to Original signer.
3. Original signer computes $\tilde{s} = \tilde{m}x + \tilde{k} (\bmod q)$ and forwards $\tilde{s}$ to Proxy signer.
4. Proxy signer computes $s = \tilde{s}\beta (\bmod q)$, and check if the following equality holds:
$$g^s = y^r r (\bmod p) \tag{5}$$
   If it holds, Proxy signer accepts s as a valid proxy signature key from Original signer.
5. Proxy signature generation and verification are the same as in MUO scheme.

This new nonrepudiable proxy signature scheme has the same properties as the one given in [18], e.g. proxy key generation and updating using insecure channels. The difficulty of finding the corresponding proxy signature key s by the original signer knowing r is equivalent to solving a discrete logarithm problem. On the other hand, the difficulty of computing x, $\tilde{k}$ or forging another valid signature that satisfies equation (5) by the proxy signer is guaranteed by the underlying signature scheme [14]. Note that message substitution attack is not feasible in our scheme since we require a constant message $m = 1$. For more detailed discussions, interested readers are referred to [18].

We notice that in [9], Horster et al. classified a large number of hidden and weak ElGamal-like signature schemes under a single framework. Our partially blind signature protocols can also be seen as a special form (when $m = 1$) of some of the variants in [9].

4 Threshold Proxy Signature Schemes

Threshold signatures are part of a general approach known as *threshold cryptography* which was introduced by the works of Boyd [2], Desmedt [4], and Desmedt and Frankel [6]. For an overview of the field, the reader is referred to [5]. Particular examples of solutions to threshold ElGamal-like signatures can be found in [8, 12, 10, 7].

Generally speaking, threshold signature schemes can be classified into the following two categories: (1) schemes with the assistance of a mutually trusted party to decide the group secret signature key s and generate individual shares for all group members [16, 1, 8]; and (2) schemes without the assistance of a mutually trusted party. In schemes without a mutually trusted party, all group members *collectively* choose and distribute the group secret signature key s [8, 12, 10, 7].

A threshold proxy signature scheme is a threshold signature scheme with the additional requirement that the group secret s, i.e. proxy signature key, should have a special form satisfying Eq (4). In this paper, we will focus on how to generate the individual shares for every group member such that Eq (4) is satisfied. Once individual shares are in place, the signing and verification of threshold proxy signatures are the same as existing threshold schemes [8, 7].

In MUO scheme, since the original signer is trusted to choose the secret proxy signature key, naturally he can serve as a trusted party to generate and deliver the shares to a group of proxy signers. It turns out to be direct applications of some (verifiable) secret sharing schemes [16, 1, 15], which we will not discuss any more. In the following, we will present a threshold proxy share generation scheme based on nonrepudiable proxy signature schemes. The shares have to be generated in such a way that nonrepudiation between the original signer and the proxy group is preserved, i.e. without a trusted party.

We assume there is a group of n proxy signers p_i, $i = 1, ..., n$, where $n \in Z_q^*$. We further assume that all the participants have access to a dedicated broadcast channel; by dedicated we mean that if a participant broadcasts a message, it will be recognized by the other participants as coming from him.

4.1 (n, n) proxy share generation based on NPS1

1. The original signer selects $\tilde{k} \in Z_q$, computes $\tilde{r} = g^{\tilde{k}} (\bmod p)$, and broadcasts $\tilde{r}$.

2. (a) Each proxy signer p_i randomly selects $\alpha_i \in Z_q$, computes

$$r_i = g^{\alpha_i} \tilde{r} (\bmod p) \tag{6}$$

   (b) Each proxy signer p_i checks whether $r_i \in Z_q^*$. If this is not the case, he goes back to step (a). Otherwise, he broadcasts r_i.
3. The original signer computes $r = \prod_{i=1}^{n} r_i$ and $\tilde{s} = n^{-1} r x + \tilde{k} (\bmod q)$ and broadcasts $\tilde{s}$.
4. Each proxy signer computes $r = \prod_{i=1}^{n} r_i$, $s_i = \tilde{s} + \alpha_i (\bmod q)$, and check if the following equality holds:

$$g^{\tilde{s}} = y^{n^{-1} r} \tilde{r} (\bmod p) \tag{7}$$

   If it holds, proxy signer p_i accepts s_i as a valid proxy share from the original signer. Otherwise, p_i broadcasts an error and stops.

Theorem If the above protocol completes successfully, the n proxy signers share a valid proxy signature key, i.e. the following equality holds:

$$g^{\Sigma s_i} = y^r r (\bmod p) \tag{8}$$

(Proof will be given in the final paper.)

4.2 (t, n) threshold proxy share generation

A (t, n) threshold secret sharing scheme is a protocol in which partial information, i.e. *shares*, about a *secret* are distributed to n participants such that

- Any group of fewer than t participants cannot obtain any information about the secret.
- Any group of at least t participants can compute the secret in polynomial time.

Having a (n, n) proxy signature key sharing from the above protocol, a (t, n) threshold proxy signature key sharing can be obtained by directly applying the verifiable secret sharing protocol by Pedersen [15], since the public information about each secret share, i.e. g^{s_i}, are publicly known. In doing so, we avoid Langford's attack [11].

4.3 Discussions

The security analysis and properties (apart from the threshold property) of the threshold NPS1 scheme presented above are very similar to those of NPS1 itself. Interested readers are referred to [18] for detailed discussions.

5 Conclusions

In this paper, we present another nonrepudiable proxy signature scheme which has similar properties as a previous one. The general approach to construct threshold proxy signature schemes is shown with an example. Generalizations to other ElGamal-like proxy signature schemes will be given in the full paper.

Acknowledgements I am grateful to Roger Needham, Ross Anderson and one of the anonymous referees for helpful comments.

References

1. G.R. Blakley, Safeguarding Cryptographic Keys, *Proc. AFIPS 1979 Nat. Computer Conf.*, 48 (1979), pp 313-317.
2. C. Boyd, Digital Multisignatures, In H. Baker and F. Piper, editors, *Cryptography and Coding*, pages 241-246, Claredon Press, 1986.
3. J.L. Camenisch, J-M. Piveteau, and M.A. Stadler, Blind Signatures Based on the Discrete Logarithm Problem, *Proc. EUROCRYPT'94*, Springer Verlag, 1994, pp. 428-432.
4. Y. Desmedt, Society and group oriented cryptography: A new concept. In Carl Pomerance, editor, *Proc. CRYPTO'87*, LNCS 293, Springer-Verlag, 1988.
5. Y. Desmedt, Threshold Cryptography, *European Transaction on Telecommunications and Related Technologies*, Vol. 5, No. 4, 1994, pp 35-43.
6. Y. Desmedt and Y. Frankel, Threshold cryptosystems. In G. Brassard, editor, *Proc. CRYPTO'89*, LNCS 435, Springer-Verlag, 1990.
7. R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, Robust Threshold DSS Signatures, *Proc. EUROCRYPT'96*, LNCS 1070, Springer-Verlag, 1996.
8. L. Harn, Group oriented (t, n) digital signature scheme and digital multisignature, *IEE Proc.-Comp. Digit. Tech.*, 141(5), Sept 1994.
9. P. Horster, M. Michels, H. Petersen, Hidden signature schemes based on the discrete logarithm problem and related concepts, *Proc. Communications and Multimedia Security*, Chapman & Hall, 1995, pp 162-177.
10. S. Langford, Threshold DSS Signatures without a Trusted Party, *Proc. CRYPTO'95*, LNCS 963, Springer-Verlag, 1995.
11. S. Langford, Weaknesses in Some Threshold Cryptosystems, *Proc. CRYPTO'96*, Springer-Verlag, 1996.
12. C. Li, T. Hwang and N. Lee, *(t, n)*-threshold signature scheme based on discrete logarithm, *Proc. EUROCRYPT'94*, Springer-Verlag, 1995.
13. M. Mambo, K. Usuda and E. Okamoto, Proxy Signatures for Delegating Signing Operation, *Proc. 3rd ACM Conference on Computer and Communications Security*, 1996.
14. K. Nyberg and R.A. Rueppel, A New Signature Scheme Based on the DSA Giving Message Recovery, *Proc. 1st ACM Conference on Computer and Communications Security*, 1993.
15. T. Pedersen, Distributed provers with applications to undeniable signatures, *Proc. EUROCRYPT'91*, Springer-Verlag, 1991.
16. A. Shamir, How to Share a Secret, *Communications of the ACM*, 22:612-613, 1979.

17. S.M. Yen and C.S. Laih, New Digital Signature Scheme Based on Discrete Logarithm, *Electronics Letters*, Vol. 29, No. 12, pp. 1120-1121, 1993.
18. K. Zhang, Nonrepudiable Proxy Signature Schemes Based on Discrete Logarithm Problem, manuscript, 1997.

A Nyberg-Rueppel Scheme

Nyberg-Rueppel scheme [14] is briefly described here. The system parameters consist of a prime p, a prime factor q of $p-1$, and an element $g \in Z_p^*$ of order q. The signer's private key is a random element $x \in Z_q$, while the corresponding public key is $y = g^x \pmod{p}$. To sign a message m, which is an integer relatively prime to q, the signer selects $k \in Z_q$ at random and computes r and s as follows:

$$r = mg^k (\bmod\, p)$$

$$s = xr + k(\bmod\, q)$$

The pair (r, s) is the signature of the message m. To verify the validity of a signature, one checks that the following equality holds:

$$m = g^{-s} y^r r(\bmod\, p).$$

B Blind Nyberg-Rueppel Scheme by Camenisch, Piveteau and Stadler

1. Alice selects $\tilde{k} \in Z_q$, computes $\tilde{r} = g^{\tilde{k}} (\bmod\, p)$, and sends $\tilde{r}$ to Bob.
2. (a) Bob randomly selects $\alpha \in Z_q$ and $\beta \in Z_q^*$, computes $r = mg^{\alpha} \tilde{r}^{\beta} (\bmod\, p)$ and $\tilde{m} = r\beta^{-1} (\bmod\, q)$.
   (b) Bob checks whether $\tilde{m} \in Z_q^*$. If this is not the case, he goes back to step a). Otherwise, he sends $\tilde{m}$ to Alice.
3. Alice computes $\tilde{s} = \tilde{m}x + \tilde{k}(\bmod\, q)$ and forwards $\tilde{s}$ to Bob.
4. Bob computes $s = \tilde{s}\beta + \alpha(\bmod\, q)$.

The pair (r, s) is a Nyberg-Rueppel signature of the message m and the above protocol is a blind signature scheme.