IEICE TRANS. FUNDAMENTALS, VOL.E92–A, NO.7 JULY 2009

LETTER

More Efficient Threshold Signature Scheme in Gap Diffie-Hellman Group

DaeHun NYANG[†a)] *and* Akihiro YAMAMURA[††b)], *Members*

SUMMARY By modifying the private key and the public key setting in Boneh-Lynn-Shacham's short signature shcheme, a variation of BLS' short signature scheme is proposed. Based on this variation, we present a very efficient threshold signature scheme where the number of pairing computation for the signaure share verification reduces to half.

key words: *bilinear pairing, digital signature, threshold signature, discrete logarithm problem*

1. Introduction

Bilinear pairings such as Weil pairing and Tate pairing of algebraic curves were originally used in cryptanalysis for MOV attack [1]. But since A. Joux has proposed the three party Diffie-Hellman key exchange protocol using bilinear pairing in ANTS 2000, bilinear pairings have been extensively used to design various interesting cryptographic protocols [2]. Most significant advances among them are practical identity based cryptography and various digital signatue schemes.

In AsiaCrypt 2001, Boneh, Lynn and Shacham presented a short digital signature scheme using bilinear pairing [4], where the signature length is the shortest among all existing signature schemes. Also, its many variations and extensions have been proposed.

In this paper, we propose a way to reduce the number of bilinear computations in a threshold signature scheme. Computation of bilinear pairing is regarded as the heaviest part of many bilinear pairing based signature schemes. Thus the reduction of the number of bilinear pairing computation improves the overall performance of a system.

More specifically, we will propose a variant of Boneh, Lynn and Shacham's basic digital signature scheme and then based on the variant, we will show a very efficient threshold digital signature scheme which saves lots of bilinear pairing computations.

2. Threshold Digital Signature

Threshold cryptography is a versatile tool for many security applications. The goal of the (t, n) threshold digital signature is to prevent a single point of failure by distributing a private key into many players(n) and thus, even under the existence of t corrupted players by an adversary, the adversary cannot compromise the private key. Or, in the (t, n) threshold signature scheme, at least $t + 1$ number of private key shares are required to restore a private key. And thus it gives more tolerance to the system. The importance of the threshold digital signature scheme was highlited by L. Zhou and Z. J. Hass [5], where they propose to use a threshold digital signature scheme to establish a certificate authority in hostile environment such like ad hoc network. After then, lots of researches have been done to improve the ad hoc network security using threshold signatures [6]. Because of the cost restrictive nature of ad hoc nodes which usually run with battery power, the efficiency of a threshold digital signature scheme is important in critical sense. A threshold digital signature scheme is evaluated by sum of the costs in generating a signature share, verifying signature shares, combining signature shares and finally verifying the combined signature. By the help of pairing-based cryptography, a short threshold signature scheme with verifiable signature shares is proposed by Boldyreva [3], which is the extension of Boneh-Lynn-Scacham's short signature scheme [4].

The Boldyreva's scheme has verifiable signature shares in the sense that every signature share can be verified by its corresponding public key. However, her protocol requires 2 pairing computations per signature share verification and totally, in the best case where every signature shares are correct, $2(t+1)$ number of pairing computations and in the worst case where no signature share is correct, $2n$ pairing computations are required to choose or fail to see $t + 1$ correct signature shares in (t, n) threshold scheme.

In this paper, we propose a threshold signature scheme that requires $t + 2$ pairing computations in the best case and $n+1$ in the worst case. To achieve this, first we design a variant of Boneh-Lynn-Scacham's digital signature scheme and based on that, propose a threshold digital signature scheme whose efficiency in verifying signature shares is almost half of Boldyreva's scheme.

3. Background

Let G be a multiplicative group of the prime order p. G is also a GDH(Gap-Diffie-Hellman) group, where the computational Diffie-Hellman problem is hard but the decisional Diffieh-Hellman problem is easy. H is a random member of a hash function family, of which member maps arbitrary long string s to G. g is a generator of G and the bilinear map

Manuscript received December 1, 2008.

[†]The author is with the Information Security Research Lab., INHA University, Incheon, 402-751, Korea.

[††]The author is with the Department of Computer Science and Engineering, Akita University, Akita-shi, 010-8520 Japan.

a) E-mail: nyang@inha.ac.kr

b) E-mail: yamamura@ie.akita-u.ac.jp

DOI: 10.1587/transfun.E92.A.1720

$e : (G_1, G_2) \rightarrow G_T$ satisfies the followings:

- Bilinearity: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1 \in G_1, g_2 \in G_2$ and $a, b \in Z_p^*$.
- Non-degeneracy: If g_1 is a generator of G_1, then $e(g_1, g_2) \neq 1$.
- Computable: There is an efficient algorithm to compute $e(g_1, g_2)$ for all for all $g_1 \in G_1, g_2 \in G_2$.

Boneh-Lynn-Shacham's short signature scheme is defined as follows:

1. $\mathcal{K}(p, g, H)$: Choose randomly $x \in Z_p^*$ and compute $y = g^x$. Return $(pk = (p, g, H, y), sk = x)$.
2. $\mathcal{S}(p, g, H, sk, m)$: Compute $\sigma = H(m)^x$. Return (m, σ).
3. $\mathcal{V}(pk, m, \sigma)$: If $e(y, H(m)) = e(g, \sigma)$ then return 1, else return 0.

Boldyreva's threshold signatrue scheme can be described as follows:

1. $\mathcal{TK}(p, g, H)$: The system private key $sk = x \in Z_p^*$ of which corresponding public key is $y = g^x$ is shared among users by Shamir's secret sharing scheme such that any subset S of $t+1$ users can reconstruct sk using Lagrange interpolation: $x = \sum_{i \in S} L_i x_i$, where L_i is a Lagrange coefficient, x_i is a private key share and $y_i = g^{x_i}$ is the public key share of a user u_i.
2. $\mathcal{TS}(p, g, H, x_i, y_i, m)$:
   a. Signature share generation: for a message $m \in \{0, 1\}^*$, user u_i outputs $\sigma_i = H(m)^{x_i}$.
   b. Signature share verification: Given m, σ_i, y_i, anyone can check whether the signature share is valid or not by checking $e(g, \sigma_i) = e(y_i, H(m))$.
   c. Signature reconstruction: After collecting $t + 1$ valid signature shares, the signature on m can be reconstructed by computing $\sigma = \prod_{i \in S} \sigma_i^{L_i}$.
3. $\mathcal{V}(p, g, H, y, m, \sigma)$: The resulting signature can be verified by checking $e(g, \sigma) = e(y, H(m))$.

A threshold signature scheme $(\mathcal{TK}, \mathcal{TS}, \mathcal{V})$ is accepted to be secure if *unforgeability* under chosen-message attacks and *robustness* are satisfied in the following sense.

1. Unforgeability: No polynomial time adversary that is allowed to corrupt up to t users and given the view of the protocols $\mathcal{TK}, \mathcal{TS}$ can produce a valid (m, σ) pair such that m has not been shown to the adversary as public input to $\mathcal{TS}$.
2. Robustness: Under the existence of polynomial time adversary that is allowed to corrupt up to t users, the protocols $\mathcal{TK}, \mathcal{TS}$ complete successfully.

4. BLS Variant Digital Signature

Our basic signature is a variant of BLS's short signature scheme as following:

1. $\mathcal{K}(p, g, H)$: Choose randomly $x \in Z_p^*$ and compute $y = g^{1/x}$. Return $(pk = (p, g, H, y), sk = x)$.
2. $\mathcal{S}(p, g, H, sk, m)$: Compute $\sigma = H(m)^x$. Return (m, σ).
3. $\mathcal{V}(p, g, H, y, m, \sigma)$: If $e(g, H(m)) = e(y, \sigma)$ then return 1, else return 0.

Signing a message is the same as that of BLS, but public key setting is little bit different and thus verification procedure is slightly different, also. By this setting, we will have more advantage in the threshold signature scheme.

5. An Efficient Threshold Signature Scheme

Based on the BLS variant in section 4, we show an efficient threshold signature scheme.

1. $\mathcal{TK}(p, g, H)$: The system private key $sk = x \in Z_p^*$ of which corresponding public key is $g^{1/x}$ is shared among users by Shamir's secret sharing scheme such that any subset S of $t+1$ users can reconstruct sk using Lagrange interpolation: $sk = \sum_{i \in S} L_i x_i$, where L_i is a Lagrange coefficient, x_i is a private key share and $y_i = g^{1/x_i}$ is the public key of a user u_i.
2. $\mathcal{TS}(p, g, H, x_i, y_i, m)$:
   a. Signature share generation: for a message $m \in \{0, 1\}^*$, user u_i outputs $\sigma_i = H(m)^{x_i}$.
   b. Signature share verification: Given m, σ_i, y_i, anyone can check whether the signature share is valid or not by checking $e(g, H(m)) = e(y_i, \sigma_i)$.
   c. Signature reconstruction: After collecting $t + 1$ valid signature shares, the signature on m can be reconstructed by computing $\sigma = \prod_{i \in S} \sigma_i^{L_i}$.
3. $\mathcal{V}(p, g, H, y, m, \sigma)$: The resulting signature can be verified by checking $e(g, H(m)) = e(y, \sigma)$.

When a signature is generated, the most time consuming part in $\mathcal{TS}$ is the signature share verification step, because under the existence of adversary, one must collect $t+1$ valid signature shares, which requires quite many share verification steps. Fortunately, in our scheme, however, half of the pairing computations are not necessary. Comparing our verification step with Boldyreva's, the difference is how to validate the shares, that is, $e(g, H(m)) = e(y_i, \sigma_i)$ in our scheme and $e(g, \sigma_i) = e(y_i, H(m))$ in Boldyreva's. The left hand of the expression in our scheme(or $e(g, H(m))$) always remains the same for every signature share verification, whereas in Boldyreava's, both hands are dependent upon individual signature shares. Consequently, our scheme requires $1 + t + 1$ pairing computations in the best case and $1 + n$ in the worst case, whereas Boldyreva's scheme, $2(t+1)$ and $2n$, respectively.

6. Security

The security of our BLS variant signature scheme is little bit different from BLS's scheme, because public key is $g^{1/x}$. Thus, we first show the security of the BLS variant signature scheme and then that of its threshold version.

Theorem 6.1: The probablistic polynomial time adversary that is required to find a valid $(m, \sigma = H(m)^x)$ pair such that $m, H, g, p, y = g^{1/x}$ are given and m has not been shown to the adversary as public input to $\mathcal{S}$ does not exist, if $G_1 = G_2$.

Proof Compared with BLS signature scheme, the difference is the public key setting. In BLS scheme, an adversary must extract $(m, \sigma = H(m)^x)$ given $(m, H, g, p, y = g^x)$ tuple, wheares the adversary attacking our BLS variant must find out the same information from $(m, H, g, p, y = g^{1/x})$ tuple. Thus, if we are able to prove that these two problems are the same, then the security of BLS variant is equavalent to that of BLS.

Now, let's see some hard problems:

1. Computational co-Diffie-Hellman on (G_1, G_2). Given $g_2, g_2^a \in G_2$ and $h \in G_1$ as input, compute $h^a \in G_1$.
2. Decision co-Diffie-Hellman on (G_1, G_2). Given $g_2, g_2^a \in G_2$ and $h, h^b \in G_1$ as input, output "yes" if $a = b$ and "no", otherwise. When the answer is "yes", (g_2, g_2^a, h, h^a) is called a co-Diffie-Hellman tuple.

Informally, the security of BLS scheme is dependent upon the co-GDH where the computational co-DH is intractable, but the decision co-DH is efficient. For the security of our scheme, see the following problems:

1. Reversion of Computational Diffie-Hellman on G_1. Given $g^a, g^b \in G_1$ as input, compute $g^{a/b} \in G_1$.
2. Reversion of Computational co-Diffie-Hellman on (G_1, G_2). Given $g_2, g_2^{1/a} \in G_2$ and $h \in G_1$ as input, compute $h^a \in G_1$.
3. Reversion of Decision co-Diffie-Hellman on (G_1, G_2). Given $g_2, g_2^{1/a} \in G_2$ and $h, h^b \in G_1$ as input, output "yes" if $a = b$ and "no", otherwise. When the answer is "yes", $(g_2, g_2^{1/a}, h, h^a)$ is called a co-Diffie-Hellman tuple.

The reversion of CDH was first shown in [7]. Also, note that if $G_1 = G_2$, all the above co-Diffie-Hellman problems reduce to standard CDH/DDH and Reversion of CDH/DDH, respectively. The reversion of CDH/DDH is proved to be equivalent to CDH/DDH in [7]. Thus, our BLS variant signature scheme is as secure as CDH is if $G_1 = G_2$. ∎

Now, we must prove security of the threshold version of our BLS variant signature scheme, which requires proofs of both unforgeability and robustness, where the unforgeability is the same as that of Boldyreva's because the BLS variant scheme is secure and the $\mathcal{TK}, \mathcal{TS}$ is the same as that of Boldyreva's. More specifically, if the underlying signature scheme is secure and $(\mathcal{TK}, \mathcal{TS}, \mathcal{V})$ is simulatable, then we call the threshold scheme "unforgeable" [8]. "Simulatable" means that for every probablistic polynomial time adversary $\mathcal{A}$ that is allowed to corrupt up to t users, there exists a probablistic polynomial time simulator SIM that can output the indistinguishable view from $\mathcal{A}$'s view of $\mathcal{TK}, \mathcal{TS}$ on input y, m, σ. Our simulator for $\mathcal{A}$'s view of $\mathcal{TK}$ is the same as that shown in [9], and the simulator for $\mathcal{A}$'s view of $\mathcal{TS}$ is the same as that presented in [3]. Thus, in the following theorem, we will prove only the "robustness" of our protocol.

Theorem 6.2: Let $G = G_1 = G_2$ be a GDH group. Then our threshold signature scheme is a robust threshold signature scheme in the random oracle model against an adversary which is allowed up to $t < n/2$ users.

Proof Under the existence of an adversary that corrupts $t < n/2$ users, all subsets of $t + 1$ signature shares define the same unique signature σ. Note that owing to the signature verification step where a signature share σ_i is a valid BLS variant signature corresponding to the public key $y_i = g^{1/x_i}$, only valid signature shares can pass the verification step of $\mathcal{TS}$. The verification step is the same as the signature verification of the BLS variant signature. The signature σ is computed as multiplication of $t + 1$ valid signature shares due to Lagrange interpolation, and thus $\mathcal{TS}$ completes successfully, which means "robustness" of our protocol. ∎

7. Conclusion

A threshold signature scheme with efficient signature share verification was proposed. The number of bilinear pairing computation for signature share verification is half compared with Boldyreva's scheme. Considering that the speed of pairing computation is still 3-4 times slower than that of the scalar multiplication and the number of pairing computations is proportional to the threshold value, our scheme greatly improves the overall system performance. Also, we proved that our threshold signature scheme is secure under the random oracle model if GDH group exists.

Acknowledgement

This work was partly supported by the IT R&D program of MKE/IITA [2008-F-036-02, Development of Anonymity-based u-Knowledge Security Technology] and by Invitation Fellowship Programs by Japan Society for the Promotion of Science.

References

[1] A. Menezes, T. Okamoto, and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," IEEE Trans. Inf. Theory, vol.39, no.5, pp.1639–1646, 1993.
[2] A. Joux, "A one round protocol for tripartite Diffie-Hellman," Proc. ANTS 4, LNCS 1838, pp.385–394, 2000.
[3] A. Boldyreva, "Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman-Group signature scheme," Proc. PKC 2003, LNCS 2139, pp.31–46, Springer-Verlag, 2003.
[4] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," Advances in Cryptology — Asiacrypt'01, LNCS 2248, Springer-Verlag, 2001.
[5] L. Zhou and Z.J. Hass, "Securing ad hoc networks," IEEE Netw., vol.13, no.6, pp.24–30, Nov./Dec. 1999.
[6] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad hoc networks," Proc. 9th International Conference on Networks Protocols (ICNP), pp.251–260, Riverside, CA, Nov. 2001.

[7] X. Chen, F. Zhang, and K. Kim, "A new ID-based group signature scheme from bilinear pairings," Cryptology ePrint Archive, Report 2003/116, available at http://eprint.iarc.org/2003/116

[8] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold DSS signatures," Advances in Cryptology — Eurocrypt'96, LNCS 1070, Springer-Verlag, 1996.

[9] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," Advances in Cryptology — Eurocrypt'99, LNCS 1592, Springer-Verlag, 1999.