Efficient delegation of pairing computation

Bo Gyeong Kang[1], Moon Sung Lee[1], and Je Hong Park[2]

[1] Department of Mathematics, Korea Advanced Institute of Science and Technology,
373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, Korea
{snubogus,ms.lee}@kaist.ac.kr
[2] National Security Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
jhpark@etri.re.kr

Abstract. Pairing computation requires a lot of efforts for portable small devices such as smart cards. It was first considered concretely by Chevallier-Mames *et al.* that the cards delegate computation of pairings to a powerful device. In this paper, we propose more efficient protocols than those of Chevallier-Mames *et al.* in two cases, and provide two new variants that would be useful in real applications.

1 Introduction

Pairing based cryptosystem has become one of the most popular areas in modern cryptography since Boneh and Franklin had solved the open problem by constructing identity based encryption based on pairings [5]. Unfortunately, although many efforts have been put into improving the computation of pairings [9, 2, 3, 1], it has been considered as a burden to implement cryptographic protocols [8], specially in a small device that has limited computational power.

As a solution to this problem, Chevallier-Mames *et al.* recently proposed simple protocols, so called, secure delegation of pairing computation that enables a computationally limited device to delegate the computation of pairings to a powerful device [7]. They focused on the privacy of the limited device in such a way that

1. the powerful device learns nothing about the points A and B when the computation of pairing $e(A, B)$ is delegated,
2. the limited device is able to detect when the powerful device is cheating.

Their protocols are divided according to the condition for A and B. They first presented a general description concerning private A and B, and then extended it to several other cases, where one of the points of A and B, or both are already publicly known or constant.

In this paper, we propose an improved delegation protocol for private A and B, and further derive several efficient variants according to the condition for A and B. Our protocols satisfy security requirements and offer efficiency improvements over the previous ones.

The rest of this paper is organized as follows. We briefly review several definitions in Section 2. In Section 3, we describe a basic delegation protocol and prove its security. Then several variants are derived in Section 4. Finally, we draw our conclusion in Section 5.

2 Preliminaries

Let $\mathbb{G}_1$, $\mathbb{G}_2$ denote prime order subgroups of an elliptic curve E over the field $\mathbb{F}_q$. Let the order of $\mathbb{G}_1$ and $\mathbb{G}_2$ be denoted by ℓ and define k to be the smallest integer such that $\ell | q^k - 1$. And let G_1 and G_2 be the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. By a pairing we shall mean a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $\mathbb{G}_T$ be a multiplicative subgroup of $\mathbb{F}_{q^k}^*$ of order ℓ if

1. For all a, $b \in \mathbb{Z}_\ell$, $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, $e(aP, bQ) = e(P, Q)^{ab}$ is efficiently computable;
2. The map is non-degenerate, namely, $e(G_1, G_2) \neq 1$.

Now, we provide the security notions for secure pairing delegation. The formal security notions derived from the general framework of secure multiparty computation was first considered in [7].

Definition 1. *A protocol for pairing delegation is secure if it satisfies the three following requirements:*

Completeness *The limited device obtains $e(A, B)$ when the protocol has processed with honest powerful device.*

Secrecy *Points A and B should be kept secret. Formally speaking, there is a simulator S so that for any A, B, the output of S is computationally indistinguishable from the powerful device's view.*

Correctness *When a powerful device cheats, the limited device should be capable of detecting except with negligible probability.*

3 Efficient delegation of elliptic curves

We describe our efficient secure pairing delegation protocol. It is assumed that the card knows the pairing value $e(G_1, G_2)$.

Step-1: The card generates a random $g_1, g_2 \in \mathbb{Z}_\ell$ and requests three following pairings to the terminal:

$$\alpha_1 = e(g_1 A, G_2), \quad \alpha_2 = e(G_1, g_2 B), \quad \alpha_3 = e(g_1 A, g_2 B).$$

Step-2: The card checks that $\alpha_1, \alpha_2, \alpha_3 \in G_T$ by checking $\alpha_i^\ell = 1$. Otherwise, the card outputs $\perp$ and halts.

Step-3: The card generates two random values $r_1, r_2 \in \mathbb{Z}_\ell$ and requests the pairing:

$$\alpha_4 = e(A + r_1 G_1, B + r_2 G_2).$$

Step-4: The card finally computes:

$$\alpha_4' = \alpha_3^{(g_1 g_2)^{-1}} \cdot \alpha_1^{g_1^{-1} r_2} \cdot \alpha_2^{g_2^{-1} r_1} \cdot e(G_1, G_2)^{r_1 r_2},$$

and checks that $\alpha_4' = \alpha_4$. In this case, the card outputs $\alpha_3^{(g_1 g_2)^{-1}} = e(A, B)$; otherwise it outputs $\perp$.

Our protocol might be seen as being quite similar with that by Chevallier-Mames *et al.* in [7, Section 4.1]. The difference is that we use $g_1 A, g_2 B, A + r_1 G_1$ and $B + r_2 G_2$ in place of $A + g_1 G_1, B + g_2 G_2, a_1 A + r_1 G_1$ and $a_2 B + r_2 G_2$, respectively. As a result, our protocol requires 4 scalar multiplications in $\mathbb{G}_1, \mathbb{G}_2$ and 7 exponentiations in $\mathbb{G}_T$ whereas the protocol by Chevallier-Mames *et al.* requires 2 scalar multiplications, 2 simultaneous scalar multiplications in $\mathbb{G}_1, \mathbb{G}_2$ and 10 exponentiations in $\mathbb{G}_T$. Now we show that this protocol satisfies security notions in Definition 1.

Theorem 1. *The above protocol is a secure pairing delegation protocol.*

Proof. The *completeness* property is easily checked as below. Recall

$$\begin{aligned}
\alpha_1 &= e(g_1 A, G_2) = e(A, G_2)^{g_1}, \\
\alpha_2 &= e(G_1, g_2 B) = e(G_1, B)^{g_2}, \\
\alpha_3 &= e(g_1 A, g_2 B) = e(A, B)^{g_1 g_2}, \\
\alpha_4 &= e(A + r_1 G_1, B + r_2 G_2) = e(A, B) \cdot e(A, r_2 G_2) \cdot e(r_1 G_1, B) \cdot e(r_1 G_1, r_2 G_2) \\
&= e(A, B) \cdot e(A, G_2)^{r_2} \cdot e(G_1, B)^{r_1} \cdot e(G_1, G_2)^{r_1 r_2}.
\end{aligned}$$

This gives $\alpha_4 = \alpha_3^{(g_1 g_2)^{-1}} \cdot \alpha_1^{g_1^{-1} r_2} \cdot \alpha_2^{g_2^{-1} r_1} \cdot e(G_1, G_2)^{r_1 r_2}$.

The *secrecy* property is guaranteed from the fact that delivered points such as $g_1 A, g_2 B, A + r_1 G_1, B + r_2 G_2$ are random points in $\mathbb{G}_1, \mathbb{G}_2$. This implies that A, B are kept secret.

The *correctness* property is guaranteed as follows: If α_3 is not equal to $e(A, B)^{g_1 g_2}$, then α_4' is almost uniformly distributed among $\mathbb{G}_T$. So the probability of $\alpha_4 = \alpha_4'$ is roughly $1/\ell$. We compute it more concretely as below.

Let $U = A + r_1 G_1,\ V = B + r_2 G_2,\ W = g_1 A,\ Z = g_2 B$ and let $A = aG_1, B = bG_2, U = uG_1, V = vG_2, W = wG_1, Z = zG_2$ where $a, b, u, v, w, z \in \mathbb{Z}_\ell$. This gives

$$u = a + r_1, v = b + r_2, w = g_1 a, \text{ and } z = g_2 b. \tag{1}$$

Suppose $\beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{Z}_\ell$ such that

$$\begin{aligned}
\alpha_1 &= e(g_1 A, G_2) \cdot e(G_1, G_2)^{\beta_1} = e(A, G_2)^{g_1} \cdot e(G_1, G_2)^{\beta_1}, \\
\alpha_2 &= e(G_1, g_2 B) \cdot e(G_1, G_2)^{\beta_2} = e(G_1, B)^{g_2} \cdot e(G_1, G_2)^{\beta_2}, \\
\alpha_3 &= e(g_1 A, g_2 B) \cdot e(G_1, G_2)^{\beta_3} = e(A, B)^{g_1 g_2} \cdot e(G_1, G_2)^{\beta_3}, \\
\alpha_4 &= e(A + r_1 G_1, B + r_2 G_2) \cdot e(G_1, G_2)^{\beta_4}.
\end{aligned}$$

The terminal's view includes the points U, V, W and Z, and its view is entirely determined by $(\beta_1, \beta_2, \beta_3, \beta_4, u, v, w, z, r)$ where r is terminal's randomness. It is obvious that α_3 is $e(A, B)^{g_1 g_2}$ iff $\beta_3 = 0$. Next, we show that if $\beta_3 \neq 0$ then the card outputs $\perp$ except with negligible probability. From the equation (1), we have

$$\begin{aligned}\alpha_4' &= \alpha_3^{(g_1 g_2)^{-1}} \cdot \alpha_1{}^{g_1^{-1} r_2} \cdot \alpha_2{}^{g_2^{-1} r_1} \cdot e(G_1, G_2)^{r_1 r_2} \\ &= e(A + r_1 G_1, B + r_2 G_2) \cdot e(G_1, G_2)^{\beta_3 (g_1 g_2)^{-1} + g_1^{-1} r_2 \beta_1 + g_2^{-1} r_1 \beta_2}.\end{aligned}$$

In order to derive the probability of $\alpha_4' = \alpha_4$, we should know the probability of holding the equation

$$\beta_4 = \beta_3 (g_1 g_2)^{-1} + g_1^{-1} r_2 \beta_1 + g_2^{-1} r_1 \beta_2 \bmod \ell.$$

This can be rewritten as

$$(g_1 g_2)^{-1} (\beta_3 - z\beta_1 - w\beta_2) + g_1^{-1} (v\beta_1) + g_2^{-1} (u\beta_2) = \beta_4 \bmod \ell$$

by using the Eq. (1). If $u, v \neq 0$, then $\beta_3 \neq 0$ implies $(\beta_3 - z\beta_1 - w\beta_2, v\beta_1, u\beta_2) \neq (0, 0, 0)$. By [7, Lemma 1], the number of solutions under this condition is at most $2\ell - 1$. Since g_1 and g_2 are uniformly distributed independent from the terminal's view, the probability of $\beta_4' = \beta_4$ is at most $\frac{2\ell - 1}{\ell^2} \leq \frac{2}{\ell}$. Moreover, $u = 0$ or $v = 0$ with probability at most $2/\ell$, we conclude that the card can detect if $\alpha_3 \neq e(A, B)^{g_1 g_2}$ except with probability at most $4/\ell$ that is negligible.

4 Several Variants

Here, we present variant protocols according to the condition for the input values of the pairing. Security proof of each protocol is easily followed along with small modifications of proof of Theorem 1, so we only describe protocols and their efficiency.

4.1 Private *A*, Public *B*

In this subsection, we consider the case when the point B is already publicly known. As commented in [7, Section 5], when decrypting with Boneh and Franklin's identity-based encryption scheme, the point A is the user's private key, and the point B is a part of given ciphertext. In this case, the point B does not need to be protected. Thus, we set $g_2 = 1$ and apply several appropriate changes to the previous protocol. Assume that $e(G_1, G_2)$ has been known to the card in advance.

Step-1: The card generates a random number $g_1 \in \mathbb{Z}_\ell$ and requests the three following pairings to the terminal:

$$\alpha_1 = e(g_1 A, G_2), \quad \alpha_2 = e(G_1, B), \quad \alpha_3 = e(g_1 A, B).$$

Step-2: The card checks that $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{G}_T$ by checking $\alpha_i^\ell = 1$. Otherwise, the card outputs $\perp$ and halts.

Step-3: The card generates random values $a_2, r_1, r_2 \in \mathbb{Z}_\ell$ and requests the pairing

$$\alpha_4 = e(A + r_1 G_1, a_2 B + r_2 G_2).$$

Step-4: The card finally computes:

$$\alpha_4' = \alpha_3^{g_1^{-1} a_2} \cdot \alpha_1^{g_1^{-1} r_2} \cdot \alpha_2^{a_2 r_1} \cdot e(G_1, G_2)^{r_1 r_2}$$

and checks that $\alpha_4' = \alpha_4$. In this case, the card outputs $\alpha_3^{g_2^{-1}} = e(A, B)$; otherwise it outputs $\perp$.

This protocol requires 2 scalar multiplications and 1 simultaneous scalar multiplication in $\mathbb{G}_1, \mathbb{G}_2$ and 8 exponentiations in $\mathbb{G}_T$. The protocol in [7, Section 5.1] requires 1 scalar multiplication and 2 simultaneous scalar multiplications in $\mathbb{G}_1, \mathbb{G}_2$ and same number of exponentiations in $\mathbb{G}_T$. Thus, our protocol is a bit more efficient. It's security can be proved in the same way with Theorem 1.

4.2 Private *A*, constant private *B*

To decrypt a ciphertext $C = (X, Y, Z)$ generated by the ID-based encryption scheme of Boneh and Boyen [4], the card needs to compute $e(X + rY, K)$ for a given private key $S_{\mathrm{ID}} = (r, K)$. One can regard $X + rY$ and K as private and constant private points, respectively. For this case, we propose a new variant. It is assumed that the card already knows $e(Q, B)$ for a random point Q in $\mathbb{G}_1$. The points Q, B and the value $e(Q, B)$ are kept secret by the card.

Step-1: The card generates random numbers $r_1, r_2, g_1, g_2 \in \mathbb{Z}_\ell$ and requests the following pairings to the terminal

$$\alpha_1 = e(A + r_1 Q, r_2 B), \quad \alpha_2 = e(g_1 A, g_2 B).$$

Step-2: The card checks $\alpha_1 = \alpha_2^{(g_1 g_2)^{-1} r_2} \cdot e(Q, B)^{r_1 r_2}$ and $\alpha_1^\ell = 1$. If it is satisfied, it outputs $\alpha_2^{(g_1 g_2)^{-1}} = e(A, B)$, otherwise it outputs $\perp$.

This protocol requires 4 scalar multiplications in $\mathbb{G}_1, \mathbb{G}_2$ and 4 exponentiations in $\mathbb{G}_T$. This protocol reduces the exponentiation in $\mathbb{G}_T$ from 7 to 4 compared to the protocol in Section 3.

4.3 Private *A* and constant public *B*

In the ID-based signature scheme of Hess [10], the signing stage requires one pairing computation $e(P_1, P)$ for a randomly chosen $P_1 \in \mathbb{G}_1$ and a system parameter $P \in \mathbb{G}_2$. Since P_1 should not be exposed to anyone else but signer, one can regard it as a private point. For the delegation protocol in this case, we modify our basic delegation protocol. It is assumed that the card knows $e(Q, B)$ for a random point Q in $\mathbb{G}_1$. The point Q and the value $e(Q, B)$ are kept secret by the card.

Step-1: The card generates a random $r_1, g_1 \in \mathbb{Z}_\ell$ and requests the following pairings to the terminal, $\alpha_1 = e(A + r_1 Q, B)$ and $\alpha_2 = e(g_1 A, B)$.

Step-2: The card computes $\alpha_2^{g_1^{-1}}$ that is supposed to be $e(A, B)$.

Step-3: The card checks that $\alpha_1 = \alpha_2^{g_1^{-1}} \cdot e(Q, B)^{r_1}$ and that $\alpha_2^r = 1$. In this case, it outputs $\alpha_2^{g_1^{-1}}$; otherwise it outputs $\perp$.

This protocol can also be applied to cryptographic schemes based on the signature scheme of Hess, for example ring signature scheme [11]. Our protocol requires 2 scalar multiplications in $\mathbb{G}_1$ and 3 exponentiations in $\mathbb{G}_T$.

5 Efficiency Comparison and Conclusion

The main purpose of this paper is to improve delegation protocols for pairing computation and to cover some cases that have not been considered in [7].

Here we provide a concrete comparison of efficiency between Chevallier-Mames *et al.*'s and our protocols in terms of scalar multiplications in $\mathbb{G}_1, \mathbb{G}_2$ and exponentiations in $\mathbb{G}_T$. In Table 1, SM, SSM and EXP denote scalar multiplication in $\mathbb{G}_1$, $\mathbb{G}_2$, simultaneous scalar multiplication in $\mathbb{G}_1$, $\mathbb{G}_2$, and exponentiation in $\mathbb{G}_T$, respectively. Since SSM requires more computational efforts than SM in $\mathbb{G}_1$ or $\mathbb{G}_2$, we can easily deduce that our protocol is more efficient than the previous approach.

Table 1. Comparison

Efficiency	[7]			Ours		
	SM	SSM	EXP	SM	SSM	EXP
Private A, Private B	2	2	10	4	0	7
B constant	-			4	0	4
Private A, Public B	1	2	8	2	1	8
B constant	-			2	0	3
A constant	3	0	3	-		

Acknowledgement

The second author was supported by grant No.R01-2002-000-00151-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

References

1. P.S.L.M. Barreto, S. Galbraith, C.O. hEigeartaigh and M. Scott. Efficient pairing computation on supersingular abelain varieties. *Cryptology ePrint Archive*, Report **2004/375**.

2. P.S.L.M. Barreto, H.Y. Kim, B. Lynn and M. Scott. Efficient algorithms for pairing-based cryptosystems. *Advances in Cryptology - CRYPTO 2002*, Lecture Notes in Comput. Sci. **2442**, pp. 354–368, 2002.
3. P.S.L.M. Barreto, B. Lynn and M. Scott. Efficient algorithms for pairing-based cryptosystems. *J. Cryptology*, Vol. **17**(4): 321–334 (2004).
4. D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. *Advances in Cryptology - EUROCRYPT 2004*, Lecture Notes in Comput. Sci. **3027**, Springer-Verlag, pp. 223–238, 2004.
5. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, Vol. **32**(3): 586–615 (2003).
6. D. Boneh, B. Lynn and H. Shacham. Short signatures from the Weil pairing. *J. Cryptology*, Vol. **17**(4): 287–319 (2004).
7. B. Chevallier-Mames, J.-S. Coron, N. McCullagh, D. Naccache and M. Scott. Secure delegation of elliptic-curve pairing. *Cryptology ePrint Archive*, Report **2005/150**.
8. R. Dutta, R. Barua and P. Sarkar. Pairing-based cryptographic protocols: A Survey. *Cryptology ePrint Archive*, Report **2004/064**.
9. S. Galbraith, K. Harrison and D. Soldera. Implementing the Tate pairing. *Algorithmic Number Theory - ANTS V*, Lecture Notes in Comput. Sci. **2369**, pp. 324–337, 2002.
10. F. Hess. Efficient identity based signature schemes based on pairings. *Selected Areas in Cryptography - SAC 2002*, Lecture Notes in Comput. Sci. **2595**, Springer-Verlag, pp. 310–324, 2002.
11. F. Zhang and K. Kim. ID-based blind signature and ring signature from pairings. *Advances in Cryptology - ASIACRYPT 2002*, Lecture Notes in Comput. Sci. **2501**, Springer-Verlag, pp. 533–547, 2002.