Bit Security of keys obtained from Tripartite Authenticated Key Agreement Protocol of Type 4

YoungJu Choie*, GeunCheol Lim**

*Department of Mathematics, POSTECH,

**Graduate School for Information Technology, POSTECH.

Abstract

In [5], the bit security of keys obtained from protocols based on pairings has been discussed. However it was not able to give bit security of tripartite authenticated key(TAK) agreement protocol of type 4 . This paper shows the bit security of keys obtained from TAK-4 protocol.

I . Introduction

The Weil and Tate pairings are popular new notions in cryptography and have found many applications. In particular, the pairings have been used for identity based key exchange protocols.

In [5], it was remained an open problem to understand the bit security of keys obtained from the protocol TAK-4 of Al-Riyami and Paterson [1]. In this paper, we show the bit security of keys obtained from the protocol TAK-4.

The remainder of the paper is organized as follows. Section 2 briefly explains the cryptographic bilinear map and mathematical definitions. Section 3 discusses bit security of keys obtained from TAK-4. Finally Section 4 concludes the paper.

II . Previous Works

1. Definitions

We denote by

$$\mathrm{Tr}(z) = z+z^2+\cdots+z^{m-1}, \quad \mathrm{N}(z) = z^{p+p^2+\cdots p^{m-1}}$$

the trace and norm of $z \in F_{p^m}$ to F_p (see Section 2.3 of [8]).

For an integer x, we define

$$\| x \|_p = \min | x - ap | , \ a \in Z$$

and for a given $k > 0$, we define by $MSB_{k,p}(x)$ any integer u, $0 \le u \le p-1$, such that

$$\| x - u \|_p \le p/2^{k+1}$$

Roughly speaking, a value of $MSB_{k,p}(x)$ gives the k most significant bits of the residue of x modulo p. Note that in the above definition k need not be an integer.

Let $w_1, \cdots, w_m$ be a fixed basis of F_{p^m} to F_p and let $\#_1, \cdots, \#_m$ be the dual basis, that is

$$Tr(\#_i w_i) = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i \simeq j \end{cases}$$

see Section 2.3 of [8]. Then any element $\square \in F_{p^m}$ can be represented in the basis $w_1, \cdots, w_m$ as

$$\square = \sum_{i=1}^{m} Tr(\#_i \square) w_i .$$

The "hidden number problem with trace over a subgroup $\subseteq F_{p^m}^*$" can be formulated as follows: Given r elements $t_1, \cdots, t_r \in \subseteq F_{p^m}^*$, chosen independently some $k > 0$, recover the number $\square \in F_{p^m}$. The case of $m = 1$ and $\subseteq F_{p^m}^*$ corresponds to the hidden number problem introduced in [3].

The following statements are partial cases of Theorem 2 of [7]. We denote by $\mathcal{C}$ the set of $z \in F_{p^m}$ with norm equal to q.

Thus $|\mathcal{C}| = (p^m-1)/(p-1)$.

Lemma 2.1 [5]. Let p be a sufficiently large prime number and let $\subseteq$ be a subgroup of $\mathcal{C}$ of order l with $l \geq p^{(m-1)/2+\frac{1}{2}}$ for some fixed $\frac{1}{2}>0$. Then for

$$k = \lceil 2\sqrt{logp} \rceil \text{ and } r = \lceil 4(m+1)\sqrt{logp} \rceil$$

there is a deterministic polynomial time algorithm A as follows. For any $\square \in F_{p^m}$, if $t_1, \cdots, t_r$ are chosen uniformly and independently at random from $\subseteq$ and if $u_i = MSB_{k,p}(Tr(\square t_i))$ for $i = 1, \cdots, r$, the output of A on the 2r values (u_i, t_i) satisfies

$$Pr[A(t_1, \cdots, t_r ; u_1, \cdots, u_r) = \square] \geq 1-p^{-1}$$

Lemma 2.2 [5]. Let p be a sufficiently large prime number and let $\subseteq$ be a subgroup of $F_{p^m}^*$ of order of l with $l \geq p^{\frac{1}{2}}$ for some fixed $\frac{1}{2} > 0$. Then for any $\varepsilon > 0$, let

$$k = \lceil (1-\tfrac{1}{2}/m+\varepsilon)logp \rceil \text{ and } r = \lceil 4m/\varepsilon^2 \rceil$$

there is a deterministic polynomial time algorithm A as follows. For any $\square \in F_{p^m}$, if $t_1, \cdots, t_r$ are chosen uniformly and independently at random from $\subseteq$ and if $u_i = MSB_{k,p}(Tr(\square t_i))$ for $i = 1, \cdots, r$, the output of A on the 2r values (t_i, u_i) satisfies

$$Pr[A(t_1, \cdots, t_r ; u_1, \cdots, u_r) = \square] \geq 1-p^{-1}$$

2. Pairings

We use the same notation as in [4]. We let G_1 be an additive group of prime order q and G_2 be a multiplicative group of the same order q. We assume the existence of an efficiently computable bilinear map $\hat{e}$ from $G_1 \times G_1$ to G_2. Typically, G_1 will be a subgroup of the multiplicative group of a related finite field and the map $\hat{e}$ will be derived from either the Weil or Tate pairing on the elliptic curve.

We also assume that an element $P \in G_1$ satisfying $\hat{e}(P, P) \neq 1_{G_2}$ is known. When $a \in Z_q$, we write aP for P added to itself "a" times, also called scalar multiplication of P by "a". As a consequence of bilinearity, we have that, for $a, b \in Z_q$;

$$\hat{e}(aP, bP) = \hat{e}(P, P)^{ab} = \hat{e}(abP, P)$$

a fact that will be used repeatedly in the sequel without comment. We simply assume in what follows that suitable group G_1, a map $\hat{e}$ and an element $P \in G_1$ have been chosen, and that elements of G_1 can be represented by bit strings of the appropriate lengths. We note that the computations that need to be out by entities in our protocols will always involve pairing computations, and that the complexity of these will generally dominate any other calculations. However, with recent advances in efficient implementation of pairings [2], the complexity of a pairing computation is now of a similar order to that of elliptic curve point multiplication.

We now assume that there is an algorithm which can provide some information about one of the components $Tr(\#_i \hat{e}(P, P)^{abc})$ of above representation and show that it leads to an efficient algorithm to compute the whole value $\hat{e}(P, P)^{abc}$ hence the key $Tr(\hat{e}(P, P)^{abc})$. It follows that the partial information about one of

the components is as hard as the whole key.

To make this precise, for every $k > 0$, we denote by O_k the oracle which, for some fixed $\#\in F_{p^m}^*$ and any a, b, c $\in\{0, \cdots, l-1\}$, takes as input the pairs

(P, P), (aP, aP), (bP, bP), (cP, cP),

and outputs $MSB_{k,p}(\mathrm{Tr}(\#_i \hat{e}(P, P)^{abc}))$.

3. Tripartite Authenticated Key(TAK) Agreement Protocol

In [6], the advantage of Joux's tripartite protocol over any previous tripartite key agreement protocol is that a session key can be established in just one round. But this key is not authenticated and this allows a man-in-the-middle attack. In [1], in order to provide session key authentication, some form of authenticated long term private/public key pair are needed. As with the other protocols, a certification authority(CA) is used in the initial set-up stage to provide certificates which bind user's identities to long term keys.

The following statements are TAK key generation. As usual, in the protocol below, short-term keys a, b, $c\in F_q^*$ are selected uniformly at random by A,B and C respectively. An entity A broadcasting to B and C, sends his fresh short-term public value aP along with a certificate Cert_A containing his long-term public key. Corresponding values and certificates are broadcast by B and C to A, C and A, B respectively. Notice that the protocol messages authenticity of the two certificates he receives. If any check fails, the protocol should be aborted. When no check fails, one of four possible session keys described below should be computed. Below, H denotes a suitable hash function.

1) Type 1 (TAK-1)

The keys computed by the entities are:

$K_A = H(\hat{e}(bP, cP)^a \| \hat{e}(yP, zP)^x)$,

$K_B = H(\hat{e}(aP, cP)^b \| \hat{e}(xP, zP)^y)$,

$K_C = H(\hat{e}(aP, bP)^c \| \hat{e}(xP, yP)^z)$.

By hi-linearity, all parties now share the session key $K_{ABC}=H(\hat{e}(P, P)^{abc} \| \hat{e}(P, P)^{xyz})$.

2) Type 2 (TAK-2)

The keys computed by the entities are:

$K_A = \hat{e}(bP, zP)^a \cdot \hat{e}(yP, cP)^a \cdot \hat{e}(bP, cP)^x$,

$K_B = \hat{e}(aP, zP)^b \cdot \hat{e}(xP, cP)^b \cdot \hat{e}(aP, cP)^y$,

$K_C = \hat{e}(aP, yP)^c \cdot \hat{e}(xP, bP)^c \cdot \hat{e}(aP, bP)^z$.

$K_{ABC} = \hat{e}(P, P)^{(ab)z+(ac)y+(bc)x}$

3) Type 3 (TAK-3)

The keys computed by the entities are:

$K_A = \hat{e}(yP, cP)^x \cdot \hat{e}(bP, zP)^x \cdot \hat{e}(yP, zP)^a$,

$K_B = \hat{e}(aP, zP)^y \cdot \hat{e}(xP, cP)^y \cdot \hat{e}(xP, zP)^b$,

$K_C = \hat{e}(aP, yP)^z \cdot \hat{e}(xP, bP)^z \cdot \hat{e}(xP, yP)^c$.

$K_{ABC} = \hat{e}(P, P)^{(xy)c+(xz)b+(yz)a}$

4) Type 4 (TAK-4)

The keys computed by the entities are:

$K_A = \hat{e}(bP + H(bP \| yP)yP,$

$cP + H(cP \| zP)zP)^{a+H(aP \| xP)x}$,

$K_B = \hat{e}(aP + H(aP \| xP)xP,$

$cP + H(cP \| zP)zP)^{b+H(bP \| yP)y}$,

$K_C = \hat{e}(aP + H(aP \| xP)xP,$

$bP + H(bP \| yP)yP)^{c+H(cP \| zP)z}$.

The session key is K_{ABC} =

$$\hat{e}(P, P)^{(a+H(aP \| xP)x)(b+H(bP \| yP)y)(c+H(cP \| zP)z)}$$

For simplicity, we set $H_A := H(aP \| xP)$, $H_B := H(bP \| yP)$, $H_C := H(cP \| zP)$. Then the session key is

$$K_{ABC} = \hat{e}(P, P)^{(a+H_Ax)(b+H_By)(c+H_Cz)}.$$

Ⅲ. Bit Security of keys otained from the protocol TAK-4

In [5], it was remained as an open problem to understand the bit security of keys obtained from the protocol TAK-4 of Al-Riyami and Paterson [1].

We have already described the tripartite Diffie-Hellman system of Joux. In that case, an adversary sees (P, P), (aP, aP), (bP, bP) and (cP, cP) and the key is derived from $Tr(\hat{e}(P, P)^{abc}) \in \{0, 1, \cdots, p-1\}$. In this section, we discuss the bit security of keys obtained from the protocol version 4 of Al-Riyami and Paterson [1].

Theorem 3.1 Assume that p is an n-bit prime (for sufficiently large n) and l is the order of group G_1 such that $\gcd(l, p(p-1)) = 1$ and $l \geq p^{1/2+\frac{1}{2}}$ for some fixed $\frac{1}{2} > 0$. Then, there exists a polynomial time algorithm which, given the pairs

$$(P, P), (aP, aP), (bP, bP), (cP, cP)$$

for some a, b, c $\in \{0, \cdots, l-1\}$, makes $O(n^{1/2})$ calls of the oracle O_k with $k = \lceil 2n^{1/2} \rceil$ and computes $\hat{e}(P, P)^{abc}$ correctly with probability at least $1-p^{-1}$.

Proof. In general case, choose a random $r \in \{0, \cdots, l-1\}$ and call the oracle O_k on the pairs

$$(P, P), ((a+H_A x)P, (a+H_A x)P), ((b+H_B y)P, (b+H_B y)P), (((c+H_C z) +r)P, ((c+H_C z) +r)P)$$

(the point $((c+H_C z) +r)P$ can be computed from the values of $(c+H_C z)P$ and r.)

Let $\square = \#\hat{e}(P, P)^{(a+H_A x)(b+H_B y)(c+H_C z)}$ be hidden number and let $t = \hat{e}(P, P)^{(a+H_A x)(b+H_B y)r}$ which can be computed as $t = \hat{e}((a+H_A x)P, (b+H_B y)P)^r$. The oracle returns

$$MSB_{k,p}(Tr(\#\hat{e}(P, P)^{(a+H_A x)(b+H_B y)((c+H_C z)+r)})) = MSB_{k,p}(Tr(\#t)).$$

Since l is prime and $ab \equiv 0 \pmod{l}$ it follows that the "multipliers" t are uniformly and independently distributed in $G_2 \subseteq F_{p^m}$, when the shifts k are chosen uniformly and independently at random from 0, ⋯, l-1. Now from Lemma 2.1 we derive the result.

Similarly, from Lemma 2.2 we derive:

Theorem 3.2 Assume that p is an n-bit prime (for sufficiently large n) and l is the order of group G_1 such that $\gcd(l, p(p-1)) = 1$ and $l \geq p^{\frac{1}{2}}$ for some fixed $\frac{1}{2} > 0$. Then, for any $^2 > 0$, there exsits a polynomial time algorithm which, given the pairs

$$(P, P), (aP, aP), (bP, bP), (cP, cP)$$

for some a, b, c $\in \{0, \cdots, l-1\}$, makes $O(^{2-1})$ calls of the oracle O_k with $k = \lceil (1-\frac{1}{2}/m+^2)n \rceil$ and computes $\hat{e}(P, P)^{abc}$ correctly with probability at least $1-p^{-1}$.

Acknowledgement : This work was partially supported by university IT Research Center Project.

Ⅳ. Conclusion

[5] shows that obtaining certain bits of the common keys is as hard as computing the entire key. Our result gives bit security obtained from TAK-4 protocol. That is, we show that obtaining certain bits of the TAK-4 common keys is as hard as computing the entire key.

References

[1] S. Al-Riyami and K. G. Paterson, "Authenticated three party key agreement protocols from pairings", Cryptology ePrint Archive: Report 2002/35.

[2] P. S. L. M. Barreto, H. Y. Kim, B. Lynn and M. Scott, "Efficient algorithms for pairing-based cryptosystems", In Advances in Cryptology - CRYPTO 2002, LNCS. Springer-Verlag, 2002.

[3] D. Boneh and R. Venkatesan, "Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related

schemes", Proc. Crypto'1996. Lect. Notes in Comp. Sci, Springer-Verlag, 1109(1996), 129-142.

[4] D. Boneh and M.Franklin, "Identity-based encryption from the Weil pairing", SIAM J. Computing, to appear, full version of [11].

[5] S. D. Galbraith, Herbie J. Hopkins, and Igor E. Shparlinski, "Secure Bilinear Diffie Hellman Bits", Cryptology, ePrint Archive: Report 2002/155.

[6] A. Joux, "A one round protocol for tripartite Diffie-Hellman", Proc. ANTS-4, Lect. Notes in Comp. Sci., Springer-Verlag, Berlin, 1838(2000), 385-393

[7] W. -C. W. Li, M. Naslund and I. E. Shparlinski, "The hidden number problem with the trace and bit security of XTR and LUC", Proc. Crypto'2002, Lect. Notes in Comp. Sci., Springer-Verlag, Berlin, 2442(2002), 433-448

[8] R. Lidl and H. Niederreiter, "Finite fields", Cambridge University Press, Cambridge, 1997

[9] I. E. Shparlinski , "On the generalized hidden number problem and bit security of XTR", Proc. AAECC-14, Lect. Notes in Comp. Sci., Springer-Verlag, Berlin, 2227(2001), 268-277

[10] M. I. Gonzalez Vasco and I. E. Shparlinski, "On the security of Diffie-Hellman Bits", Proc. Workshop on Cryptography Number Theory, Singapore 1999, Birkhauser, 2001, 257-268