Research Article

An XTR-Based Constant Round Key Agreement Scheme

Silan Zhang,^{1,2} Jianhua Chen,¹ Jingbo Xia,² and Xiaochuan Ai^{1,3}

¹ School of Math, Wuhan University, Wuhan, Hubei 430072, China

² College of Science, Huazhong Agricultural University, Wuhan, Hubei 430070, China

³ College of Science, Navy Engineering University, Wuhan, Hubei 430032, China

Correspondence should be addressed to Jingbo Xia; xiajingbo.math@gmail.com

Received 15 May 2013; Revised 21 July 2013; Accepted 26 July 2013

Academic Editor: Kwok-Wo Wong

Copyright © 2013 Silan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new XTR-based key agreement scheme with constant rounds is presented. Three theorems are formulated to reveal the logarithmic computational complexity of this scheme. Furthermore, the computation framework of XTR-based key agreement scheme is introduced, and security of the scheme is proven under the formal model.

1. Introduction

Key agreement is to construct secure infrastructures for networks by computing a common session key among a pool of group members. It is a central issue in various multicast applications such as pay preview broadcast of TV program, teleconferencing, military communication, and distributed interactive games.

The pioneer work can be traced to Steiner et al. [1] in 1996, who proposed the first key agreement scheme, GDH. As an extension of Diffie-Hellman key exchange based on the discrete logarithm problem in $\langle g \rangle$, GDH achieves group key exchange, and it is known as the first key agreement scheme. After that, various improved schemes of GDH are presented such as CLIQUES [2] and TGDH [3]. Unfortunately, the improved schemes are still flawed in excessive computation or unreasonable communication.

Nowadays, it has been widely accepted that a reliable key agreement scheme should meet three utmost crucial demands: fast computation, less communication, and provable security. Here the first demand, that is, fast computation, relies heavily on the algorithm one designed. With the outcome of the new ideas in cryptosystems, including elliptic curve cryptography (ECC) and Hash, computation efficiency in key agreement scheme was increased gradually. At first, the computations were based on the discrete logarithm problem, like Steiner's classic work [1, 2], Kim's TGDH [3], and a lot of

forthcoming schemes [4–6]. Afterwards, Hash function was incorporated into some schemes to increase the computation efficiency, referring to Tso et al.'s work [7], Fu et al.'s work [8], and so forth. With the development of elliptic curves cryptography, faster computing module like Weil paring was regarded as a good replacement for common exponentiations or multiplications computations in traditional scheme. By using Weil paring on elliptic curve, the pairwise key agreement protocol [9], the tripartite key agreement protocol [10], and the ID-based authenticated group key agreement scheme [11] were given. A newer tendency was to constructing key agreement without using pairing computation; see He et al.'s work [12].

The second demand is less communicate, which refers to less information exchange between group members. In 2004, the bottleneck of key agreement scheme was illustrated as communication rounds, instead of computation rounds, which reveals the emphasis of design for an efficient key agreement scheme. In the beginning, GDH.3, proposed by Steiner et al. [1], need $n + 1$ rounds of communications. Kim et al. [3] and Dutta and Barua [4] reduced the rounds of communications to $\log_2 n$ and $\log_3 n$, respectively. Afterwards, key agreement with four rounds of communications are proposed by Tso et al. [7], Fu et al. [8], Zheng et al. [13], and so forth, and computation complexity of these schemes is slight. From Tso et al. [7], Fu et al. [8] to Zheng et al. [13], the number of exponentiations is decreased from $2n^2 + 5n, 7n$ to $5n$, and

the numbers of hash are $n^2 + n + 1$, $2n$, and $3n$, respectively. In the meantime, the reduction of communication rounds has been regarded as a central issue. Three rounds scheme is obtained by Nam et al. [5], Augot et al. [6] Yao et al. [14], and so forth. Recently, the rounds of communications have been reduced to two rounds, where Lv and Li [11], He et al. [12], and Feng et al. [15] contribute a lot. In short, key agreement with less communications has drawn many attentions nowadays, and for recent researches and reviews, refer to [16, 23, 24].

Besides the computation cost and communication, another focus on research of the key agreement scheme is the security analysis. The reason lies in two aspects. Firstly, it was widely accepted that a secure key agreement should meet several demands, including key completeness, forward secrecy, and backward secrecy. Secondly, the claim of a secure new scheme should be on the basis of a strict and formal proof, instead of colloquial illustration or informal proof. The strict proof of scheme was commenced by Bresson et al. [17], who modeled the execution of an authenticated group Diffie-Hellman scheme and proved its security by defining a formal model. Raymond Choo et al. [18] gave formal proof of certain known protocol to reveal their weakness in the security and henceforth encourage the future designer to provide proofs of security for new protocols. Actually, the importance of provable security has been widely accepted nowadays, and it has been an indispensable part of almost all key agreement schemes appeared in the newly published literatures.

Unfortunately, few schemes can achieve all the three goals because of the unbalance between security and efficiency. However, a suitable balance between computation, communication, and security is utmost importance in current key agreement research. In terms of this, this study is to propose a reliable scheme which meets the above three demands by taking the advantages of Lenstra's XTR cryptosystem [19, 20]. Here, XTR stands for ECSTR, which is an abbreviation for Efficient and Compact Subgroup Trace Representation. Actually, XTR is an efficient cryptosystem and the mathematics underlying XTR is straightforward while compared to ECC [19]. Moreover, the corresponding XTR public keys are only about twice as large as ECC keys, assuming global system parameters. Unlike RSA and ECC, parameter initiation from scratch for XTR takes a negligible amount of computing time [21]. Furthermore, Verheul [22] showed that XTR is at least as secure as supersingular elliptic curve system. This conclusion relied on a deduction that the elapse of XTR might lead to the elapse of ECC. Henceforth, the security of XTR was ensured.

In this paper, three algorithms with computation complexity in XTR theory are given. Based on these algorithms, an XTR-based key agreement scheme with constant rounds (XTR-CR) is proposed. The scheme achieves high efficiency and is scalable in computation and security as well. Moreover, the efficiency in computation and communication between XTR-CR and XTR-GDH, which is the natural analogue of GDH in XTR, is compared and the better efficiency of the XTR-CR is shown. Finally, under a decisional Diffie-Hellman (DDH) assumption, the XTR-CR is proved to be secure against active adversary in the formal model. The paper is organized as follows. In Section 2, introductions of XTR cryptosystem with three computation theorems are given.

Section 3 introduces our new scheme. The security proof of the new scheme is presented in Section 4. Conclusions are given in the last section.

2. XTR Cryptosystem

As a reliable key agreement scheme which is aiming to achieve good balance in efficiency and security, this scheme is designed on the basis of XTR. Since the XTR cryptosystem has not ever been incorporated into key agreement scheme, preliminaries of XTR cryptosystem are introduced in Section 2.1. By giving and proving three related computation theorems in Section 2.2, computation complexity of our XTR-based scheme is clarified.

2.1. Preliminaries. For a given $c \in GF(p^2)$, let $F(c, x)$ be the polynomial $X^3 - cX^2 + c^pX - 1 \in GF(p^2)[X]$ with roots h_0, h_1, h_2 in $GF(p^6)$, and denote $c_n = h_0^n + h_1^n + h_2^n$ for all $n \in \mathbb{Z}$. The trace $\text{Tr}(h)$ over $GF(p^2)$ of $h \in GF(p^6)$ is the sum of conjugates over $GF(p^2)$, that is, $\text{Tr}(h) = h + h^{p^2} + h^{p^4}$. Security analysis of the XTR system is based on the difficulties of the following three computational problems:

- (i) XTR-DL problem: discrete logarithm problem in XTR system. Given $a \in \text{Tr}(\langle g \rangle)$, one computes x ($0 \leq x < q$) such that $a = \text{Tr}(g^x)$,
- (ii) XTR-DH problem: Diffie-Hellman problem in XTR system. This problem is the computation of $\text{Tr}(g^{xy})$ with given $\text{Tr}(g^x)$ and $\text{Tr}(g^y)$. The XTR-Diffie-Hellman value $\text{Tr}(g^{xy})$ is denoted by $\text{XDH}(\text{Tr}(g^x), \text{Tr}(g^y))$,
- (iii) XTR-DDH problem: decisional Diffie-Hellman problem of determining whether $\text{XDH}(a, b) = c$ with given $a, b, c \in \text{Tr}(\langle g \rangle)$.

Unlike RSA and Elgamal, the computation in XTR is involved in a subgroup of multiplicative group $GF(p^6)^*$ with order $p^2 - p + 1$. The computation owns polynomial complexity, and as a result, it ensures the high efficiency of the implementation. Furthermore, in order to evaluate the security level of XTR, the following equivalence was proven [19].

- (i) The XTR-DL problem is (1, 1) equivalent to the DL problem in $\langle g \rangle$.
- (ii) The XTR-DH problem is (1, 2) equivalent to the DH problem in $\langle g \rangle$.
- (iii) The XTR-DL problem is (3, 2) equivalent to the DDH problem in $\langle g \rangle$.

where the problem A is (a, b) equivalent to problem B, if any instance of problem A (or B) can be solved by at most a (or b) calls to an algorithm solving problem B (or A).

2.2. Computation Theorem of XTR. For simplicity, denote $c = \text{Tr}(g)$, $c_n = \text{Tr}(g^n) \in GF(p^2)$, and $S_n(c) = (c_{n-1}, c_n, c_{n+1}) \in GF(p^2)^3$, and denote H as the hash function from $GF(p^2)$ to $GF(p)$.

Let

$$A(c) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -c^p \\ 0 & 1 & c \end{pmatrix}, \quad (1)$$

$$M_n(c) = \begin{pmatrix} c_{n-2} & c_{n-1} & c_n \\ c_{n-1} & c_n & c_{n+1} \\ c_n & c_{n+1} & c_{n+2} \end{pmatrix}$$

be 3×3 matrix over $GF(p^2)$ with c and c_n defined above, and let $C(V)$ be the center column of a 3×3 matrix V .

Following six lemmas are necessary for the proof of Theorems 7, 8, and 9.

Lemma 1 (Theorem 2.3.8 in [19]). Given c , computation of $S_n(c)$ takes $8\log_2(n)$ multiplications in $GF(p)$.

Lemma 2 (Lemma 2.3.4 in [19]). $c_{u+v} = c_u c_v - c_v^p c_{u-v} + c_{u-2v}$ for $u, v \in \mathbb{Z}$.

Lemma 3 (Corollary 2.4.3 in [19]). $c_n = S_m(c)C(A(c)^{n-m})$, where $C(V)$ is defined before.

Lemma 4 (Lemma 2.4.4 in [19]). The determinant of $M_0(c)$ equals $D = c^{2p+2} + 18c^{p+1} - 4(c^{3p} + c^3) - 27 \in GF(p)$. If $D \neq 0$, then

$$M_0(c)^{-1} = \frac{1}{D} \begin{pmatrix} 2c^2 - 6c^p & 2c^{2p} + 3c - c^{p+2} & c^{p+1} - 9 \\ 2c^{2p} + 3c - c^{p+2} & (c^2 - 2c^p)^{p+1} - 9 & (2c^{2p} + 3c - c^{p+2})^p \\ c^{p+1} - 9 & (2c^{2p} + 3c - c^{p+2})^p & (2c^2 - 6c^p)^p \end{pmatrix}. \quad (2)$$

Lemma 5 (Corollary 2.4.7 in [19]). $C(A(\text{Tr}(g))^n) = M_0(\text{Tr}(g))^{-1}(S_n(\text{Tr}(g)))^T$.

Lemma 6 (Theorem 2.4.9 in [19]). Given $M_0(\text{Tr}(g))^{-1}$, $\text{Tr}(g)$ and $S_k(\text{Tr}(g))$, the trace $\text{Tr}(g^a g^{bk})$ of $g^a g^{bk}$ is computed at a cost of $8\log_2(a/b \bmod p + 8\log_2 b + 34)$ multiplications in $GF(p)$, for $a, b \in \mathbb{Z}_q$ and unknown k .

Based on the above lemmas, three theorems crucial to the implementation of XTR-CR scheme can be obtained.

Theorem 7. It takes 12 multiplications in $GF(p)$ to compute $\text{Tr}(g^{a+b})$ with $S_a(\text{Tr}(g))$ and $S_b(\text{Tr}(g))$.

Proof. From Lemmas 3, 4, and 5, it is straight that

$$\text{Tr}(g^{a+b}) = S_a(\text{Tr}(g)) M_0(\text{Tr}(g))^{-1} S_b(\text{Tr}(g))^T. \quad (3)$$

Note that the matrix $M_0(\text{Tr}(g))^{-1}$ is computed in advance, by analyzing the computation cost of $S_a(\text{Tr}(g))$, $S_b(\text{Tr}(g))^T$; the conclusion is direct. $\square$

Theorem 8. For unknown $a, b \in \mathbb{Z}_q$, given $S_a(c)$ and $S_b(c)$, the multiplication $S_{a+b}(c)$ is computed at a cost of 40 multiplications in $GF(p)$.

Proof. First c_{a-2} and c_{a+2} can be computed from Lemma 2, that is,

$$c_{a-2} = c_{a-1+(-1)} = c_{a-1}c_{-1} - c_{-1}^p c_a + c_{a+1}, \quad (4)$$

$$c_{a+2} = c_{(a+1)+1} = c_{a+1}c_1 - c_1^p c_a + c_{a-1}.$$

As a result, $S_{a-1}(c)$, $S_a(c)$ and $S_{a+1}(c)$ can be obtained from the computed vector $(c_{a-2}(c), c_{a-1}(c), c_a(c), c_{a+1}(c), c_{a+2}(c))$. From Theorem 7, we have

$$c_{a+b-1} = \text{Tr}(g^{a+b-1}) = S_{a-1}(c) M_0(c)^{-1} S_b(c)^T,$$

$$c_{a+b} = \text{Tr}(g^{a+b}) = S_a(c) M_0(c)^{-1} S_b(c)^T, \quad (5)$$

$$c_{a+b+1} = \text{Tr}(g^{a+b+1}) = S_{a+1}(c) M_0(c)^{-1} S_b(c)^T.$$

Thus $S_{a+b}(c)$ is computed.

Note that c_{-1}^p and c_1^p are computed in advance, so it takes 4 multiplications to compute c_{a-2} and c_{a+2} . From Theorem 7, it costs 12 rounds multiplications in each turn to compute c_{a+b-1} , c_{a+b} and c_{a+b+1} , summing up to 36. Thus the total demanded multiplication is 40, and the proof of Theorem 8 is now complete. $\square$

Theorem 9. For given b , $S_a(c)$, and unknown a , the computation of $S_{ab}(c)$ takes $24\log_2 b + 8\log_2(-1/b \bmod p) + 8\log_2(1/b \bmod p) + 24$ multiplications in $GF(p)$.

Proof. From given b , $S_a(c)$, and unknown a , one can compute $c_{ab+(-1)}$ via Lemma 6 by $8\log_2(b) + 8\log_2(-1/b \bmod p) + 34$ multiplications. Since the matrix $M_0(\text{Tr}(g))^{-1}$ is pre-computed, 22 multiplications can be reduced. Similarly, the computation of c_{ab+1} takes $8\log_2 b + 8\log_2(-1/b \bmod p) + 12$ multiplications. Thus $S_{ab}(c) = (c_{ab-1}, c_{ab}, c_{ab+1})$ is computed. In conjunction with Lemma 1, the claim is direct. $\square$

3. The XTR-Based Key Agreement Scheme

In this section, the new scheme XTR-CR is presented. For this, the fundamental application of Lenstra and Verheul's work [19] is mentioned first in Section 3.1. After that, two group extension protocols, XTR-GDH and XTR-CR, are listed in Section 3.2. Among them, XTR-GDH is the natural

extension of GDH by combining XTR, while XTR-CR is our proposed new scheme with low computation complexity and two rounds of communications. Finally, explicit comparisons of XTR-CR, XTR-GDH, and other competitive schemes are performed in Section 3.3, so as to reveal the advantage of XTR-based schemes, especially XTR-CR.

3.1. Key Exchange between Alice and Bob. XTR-Diffie-Hellman key exchange protocol between two members is a routine idea in Lenstra and Verheul's work [20], and it is remarkable enough to illustrate here in detail.

Suppose that Alice and Bob, who both have access to the XTR public key data $p, q, \text{Tr}(g)$, want to agree on a shared secret key K . This is done by using the following XTR version of Diffie-Hellman protocol.

Step 1. Alice selects a random $a \in F_p$, uses Algorithm 2.37 in [19], computes $S_a(\text{Tr}(g)) = (\text{Tr}(g^{a-1}), \text{Tr}(g^a), \text{Tr}(g^{a+1})) \in GF(p^2)^3$, and sends $\text{Tr}(g^a)$ to Bob.

Step 2. Bob receives $\text{Tr}(g^a)$ from Alice, selects $b \in F_p$ randomly, similarly computes $S_b(\text{Tr}(g)) = (\text{Tr}(g^{b-1}), \text{Tr}(g^b), \text{Tr}(g^{b+1})) \in GF(p^2)^3$, and sends $\text{Tr}(g^b)$ to Alice.

Step 3. Alice receives $\text{Tr}(g^b)$, computes $S_a(\text{Tr}(g^b)) = (\text{Tr}(g^{(a-1)b}), \text{Tr}(g^{ab}), \text{Tr}(g^{(a+1)b}))$, and determines K based on $\text{Tr}(g^{ab})$.

Step 4. Bob receives $\text{Tr}(g^a)$, computes $S_b(\text{Tr}(g^a)) = (\text{Tr}(g^{(b-1)a}), \text{Tr}(g^{ba}), \text{Tr}(g^{(b+1)a}))$, and also determines K based on $\text{Tr}(g^{ab})$.

3.2. Key Agreement of Group. Here, we present two constant-round communication protocols for the group key agreement. One is an analogue of GDH in XTR, denoted as XTR-GDH, another is the proposed scheme, XTR-CR.

Assume that there are n members in group, that is, $M_1, M_2, \dots, M_n$, and an identity code ID_i has been assigned to each member M_i ($i = 1, 2, \dots, n$) in advance.

3.2.1. Analogue of GDH in XTR: XTR-GDH. The first protocol XTR-GDH is a natural extension of R. Steiner's Group Diffie-Hellman protocol [19].

Algorithm XTR-GDH

Step 1. M_1 selects $r, r_1 \in [1, p^2 - p]$ randomly, computes $S_{r_1}(\text{Tr}(g)) = (\text{Tr}(g^{r_1-1}), \text{Tr}(g^{r_1}), \text{Tr}(g^{r_1+1}))$, and broadcast message $m_1 = (\text{ID}_1, \text{Tr}(g^{r_1}))$ to M_i ($i = 2, 3, \dots, n$).

Step 2. M_i ($i = 2, 3, \dots, n$) selects a random $r_i \in [1, p^2 - p]$, computes $S_{r_i}(\text{Tr}(g)) = (\text{Tr}(g^{r_i-1}), \text{Tr}(g^{r_i}), \text{Tr}(g^{r_i+1}))$, and sends message $m_i = (\text{ID}_i, \text{Tr}(g^{r_i}))$ back to M_1 .

Step 3. M_1 computes $\text{Tr}(g^{r_1 r_i})$ via XTR-DH method, hashes it to b_i ($i = 2, 3, \dots, n$). Let $c_i = \prod_{1 \leq j \leq n-1, j \neq i} b_j$, and M_1

broadcasts messages $m_{n-1+i} = (\text{ID}_1, \text{ID}_i, \text{Tr}(g^{r_1 c_i}))$ to M_i ($i = 2, 3, \dots, n$).

Step 4. The session key is $K = \text{Tr}(g^{b_1 b_2 \dots b_{n-1} r})$.

3.2.2. Proposed Scheme: XTR-CR. Below is the scheme we proposed, Algorithm XTR-CR.

Step 1. M_i ($i = 1, 2, \dots, n-1$) selects a random $r_i \in [1, p^2 - p]$, computes $S_{r_i}(\text{Tr}(g)) = (\text{Tr}(g^{r_i-1}), \text{Tr}(g^{r_i}), \text{Tr}(g^{r_i+1}))$, and broadcasts message $m_i = (\text{ID}_i, \text{Tr}(g^{r_i}))$ to M_n .

Step 2. M_n selects $r, r_n \in [1, p^2 - p]$ at random. With $S_{r_i}(\text{Tr}(g))$ and r , M_n computes $S_{r r_i}(\text{Tr}(g)) = (\text{Tr}(g^{r r_i-1}), \text{Tr}(g^{r r_i}), \text{Tr}(g^{r r_i+1}))$ by the algorithm in Theorem 9 of last section, where $i = 1, 2, \dots, n-1$.

Let $b_i = \sum_{1 \leq j \leq n, j \neq i} r_j$, M_n computes $S_{b_i r}(\text{Tr}(g))$ by Theorem 8, $i = 1, 2, \dots, n-1$.

Then, M_n broadcasts message $m_n = (\text{ID}_n, S_{r r_n}(\text{Tr}(g)), S_{b_1 r}(\text{Tr}(g)), S_{b_2 r}(\text{Tr}(g)), \dots, S_{b_{n-1} r}(\text{Tr}(g)))$ to M_i ($i = 1, 2, \dots, n-1$).

Step 3. Each M_i ($i = 1, 2, \dots, n-1$) computes $S_{r r_i}(\text{Tr}(g)) = (\text{Tr}(g^{r r_i-1}), \text{Tr}(g^{r r_i}), \text{Tr}(g^{r r_i+1}))$ by Theorem 9. By doing this, M_i reveal the session key K by computing $S_{r r_i + r b_i}(\text{Tr}(g)) = S_{r(r_1 + r_2 + \dots + r_n)}(\text{Tr}(g))$. Thus $K = \text{Tr}(g^{r(r_1 + r_2 + \dots + r_n)})$ is the session key. The key agreement procedure is finished.

The flow charts of two schemes are depicted in Figure 1.

During the whole process, the latter member M_n acts as a sponsor which carries heavier computation burden than other members. The obligation of sponsor is reasonable and necessary, because the presence of sponsor not only provides high efficiency for the scheme but also keeps the member equality in the group. This property is similar to that in the scheme of GDH [1] and TGDH [3].

3.3. Comparison of XTR-GDH, XTR-CR, and Other Competitive Key Agreement Schemes in Communication and Computation. The performances of XTR-based scheme are compared with several competitive key agreement schemes by considering the computations, message amount, and communications. Twelve typical key agreement schemes are listed in Table 1 for comparison with XTR-based schemes in terms of efficiency. All of the chosen schemes are listed with the descending order according to the number of communication rounds. Among them, GDH.3 [1] and TGDH [3] are classic and traditional protocols, while Dutta95 [4] show better performance in the rounds of communication. Other schemes are typical and competitive key agreement schemes in the literature, as introduced in the first section, and the number of communication rounds is sorted from four, three to two. As a typical one round protocol, Shim's work [23] is designed for three-party key agreement instead of arbitrary n entities, and signature is demanded; therefore, it is not equal to give a computation comparison. The explicit information of these schemes could be found in Table 1.

FIGURE 1: Flow charts of XTR-GDH and XTR-CR.

TABLE 1: Comparison between XTR-GDH, XTR-CR, and other competitive key agreements schemes.

Protocol	Number of exponentiations	Number of pairings	Number of scalar multiplications	Number of hash	Message amount	Round of communications
GDH.3 [1]	$2n - 3$	—	—	—	$2n - 1$	$n + 1$
TGDH [3]	$n(\log_2 n + 1)$	—	—	—	n	$\log_2 n$
Dutta05 [4]	$n \log_3 n$	—	—	—	n	$\log_3 n$
TYO07 [7]	$2n^2 + 5n$	—	—	$n^2 + n + 1$	$5n^2 + 6n + 1$	4
FXW09 [8]	$7n$	—	—	$2n$	$2n^2 + 6n$	4
ZZLC09 [13]	$5n$	—	—	$3n$	$4n^2$	4
YWJ08 [14]	—	$2n^2 + 4n$	$n^2 + 5n$	—	$5n^2$	3
NPKW07 [5]	$5.5n + 5n \log_2 n - 1$	—	—	—	$7n + 10n \log_2 n - 4$	3
Daniel07 [6]	$n - 2$	—	—	—	$3n$	3
XTR-GDH	—	—	$16n \log_2 p$	1	$2n - 1$	3
FWM08 [15]	—	—	$2n^3 - 2n^2 + n$	$2n$	$2n^3 - n^2$	2
LL10 [11]	—	$5n$	$10n$	—	$3n^2$	2
Xiong13 [24]	n	n	$5n$	—	$< 2n$	2
XTR-CR	—	—	$(n - 1)(88 \log_2 p + 128)$	—	n	2

As shown in Table 1, both XTR-GDH and XTR-CR perform good in the rounds of communications. In the following, performances in XTR-based schemes will be compared with other two or three round communication schemes.

Among these schemes with three rounds communications, XTR-GDH only demands $16n \log_2 p$ scalar multiplications, which seems better than YWJ08 [9] and slightly weaker than NPKW07 [5], and the latter need $5.5n + 5n \log_2 n - 1$ exponentiations. Though Daniel07 [6] need much less computations than XTR-GDH, it could be easily found that

XTR-GDH owns the least message amount. These results show that XTR-GDH is also a competitive scheme.

While observing the performance of XTR-CR among two rounds communication schemes, $(n - 1)(88 \log_2 p + 128)$ scalar multiplications are counted for XTR-CR. The computation load in XTR-CR is less than that in FWM08 [15] but a little heavier than that in LL10 [11] and Xiong13 [24]. Taking into account that $5n$ or n pairing computation is time-consuming for LL10 [11] and Xiong13 [24], XTR-CR needs comparatively less computations than the above schemes.

Besides, message amount of XTR-CR is the least among all of the schemes.

If compared XTR-GDH with XTR-CR, the latter performs better in communication and message amount. Meanwhile, XTR-CR shows slight weakness in computation complexity. For the sake of rapid development of XTR cryptosystem, the weakness in XTR-CR is subtle. Moreover, as mentioned in Section 1, key agreement with less communication is critical in the implementation. Since XTR-GDH needs one more communication than XTR-CR, the proposed scheme is a better and more efficient scheme than XTR-GDH.

In short, result from full comparisons of fourteen competitive key agreement schemes shows that XTR-GDH and XTR-CR achieve good balance in the performance of computation and communication. Moreover, XTR-CR performs the best both in computation and communication.

4. Security Analysis of XTR-CR

In this section, a formal model is utilized to prove that XTR-CR scheme is secure against adversary under XTR-DDH assumption. The explicit security proof of the proposed scheme is given in Section 4.2.

4.1. Security Basis of Formal Model. Let $S = \{M_1, M_2, \dots, M_n\}$ be a set of n users who wish to participate in a group key agreement. One assumes that n is polynomially bounded in the security parameter k . A player M_i has many instances called oracles, involved in distinct concurrent executions of protocols. The instances s of player M_i are denoted as $\prod_i^s$.

The adversary A has an endless supply of oracles $\prod_i^s$ and makes various queries to them. Each query models a capability of adversary.

- (i) Execute (S). This query returns a transcript of an honest protocol execution among instances of users in S .
- (ii) Send ($\prod_i^s, m$). This query sends message m to oracle $\prod_i^s$. When oracle $\prod_i^s$ receives the message m , it proceeds as specified in the protocol; the oracle updates its state and then generates and sends out a response message as needed. The response message is returned to the adversary A . A query of form Send ($\text{prod}_i^s, \text{Start}'$) allows adversary A to initiate an execution of the protocol.
- (iii) Reveal ($\prod_i^s$). This query returns the session key K if oracle $\prod_i^s$ has computed a session key. This query models the capability of adversary to obtain some session keys.
- (iv) Corrupt (M_i). The long-term private key of user M_i is returned in response to the query which is considered to deal with forward secrecy.
- (v) Dump ($\prod_i^s$). This query returns all short-term secret values used by oracle $\prod_i^s$, modeling the adversary's capability to embed a Trojan horse or other forms of malicious code into a system and then log all the session-specific information of victim. But neither the

session key K computed by $\prod_i^s$ nor the long-term private key SK_i is returned.

- (vi) Test ($\prod_i^s$). This query is asked only once when the adversary A wants to attempt to distinguish the real session key K from a random fake key, modeling the semantic security of session key K . To answer the query, one flips a secret coin b and returns the real session key K if $b = 1$, or else a random string chosen from $\{0, 1\}^l$, if $b = 0$, where l is the length of session key to be distributed in the protocol. This query is made only if oracle $\prod_i^s$ is fresh, and the definition of which will be given below.

To quantify the ability of an adversary A , one consider the query action of A here. During the execution of protocol, the adversary A , at any time, asks a Test query to a fresh oracle, then gets back an l -bit string as the response to this query, and at some later point, outputs a bit b' as a guess for the hidden bit b . Let Correct Guess (CG) be the event that $b' = b$. Then we define the advantage of A in attacking protocol P to be

$$\text{Adv}_{A,P}(k) = 2\Pr[\text{CG}] - 1. \quad (6)$$

We define protocol P as secure against an adversary A if $\text{Adv}_{A,P}(k)$ is negligible.

According to the illustration of XTR-DDH problem in Section 2, a formal description of XTR-DDH assumption is displayed below.

Let $G = \text{Tr}(\langle g \rangle)$ be XTR group, and let x, y, z be randomly chosen elements in Z_p . Informally, the XTR-DDH assumption is that it is difficult to distinguish between the distributions of $(\text{Tr}(g^x), \text{Tr}(g^y), \text{Tr}(g^{xy}))$ and $(\text{Tr}(g^x), \text{Tr}(g^y), \text{Tr}(g^z))$.

More formally, if we define $\text{Adv}_G^{\text{XTR-DDH}}(A)$ as

$$\begin{aligned} \text{Adv}_G^{\text{XTR-DDH}}(A) \\ = |\Pr[A(\text{Tr}(g), \text{Tr}(g^x), \text{Tr}(g^y), \text{Tr}(g^{xy})) = 1] \\ - \Pr[A(\text{Tr}(g), \text{Tr}(g^x), \text{Tr}(g^y), \text{Tr}(g^z)) = 1]|, \end{aligned} \quad (7)$$

the XTR-DDH assumption holds in $G = \text{Tr}(\langle g \rangle)$ if $\text{Adv}_G^{\text{XTR-DDH}}(A)$ is negligible for any probabilistic polynomial time adversary A . We denote by $\text{Adv}_G^{\text{XTR-DDH}}(t)$ the maximum value of $\text{Adv}_G^{\text{XTR-DDH}}(A)$ over all adversaries A running in time at most t .

4.2. Security Proof of the Key Agreement Protocol

Theorem 10. Let A be a passive adversary attacking protocol XTR-CR, running in time t and making q_{ex} Execute queries. Then one has

$$\text{Adv}_{A,P}(k) \leq 2q_{\text{ex}}\text{Adv}_G^{\text{XTR-DDH}}(t'), \quad (8)$$

where $t' = t + O(nq_{\text{ex}}t_{\text{exp}})$, with t_{exp} being the time required to compute an exponentiation in $G = \text{Tr}(\langle g \rangle)$.

Proof. Assume that A can guess the hidden bit b correctly with probability $1/2 + \varepsilon$. Then we construct from a distinguisher D that solves the XTR-DDH problem in G with probability $\varepsilon/q_{\text{ex}}$.

Before describing the construction of D , we define the following two distributions.

Consider the following

$$\begin{aligned} \text{Real} &= \left\{ (T, K) \left| \begin{array}{l} r_1, r_2, \dots, r_n, r \in_R Z_p, \\ z_1 = \text{Tr}(g^{r_1}), \dots, z_n = \text{Tr}(g^{r_n}), \quad z = \text{Tr}(g^r), \\ x_1 = \text{Tr}(g^{rr_1}), \dots, x_n = \text{Tr}(g^{rr_n}), \\ y_1 = x_2 x_3 \cdots x_n = S_{b_1 r}(\text{Tr}(g)), \dots, \\ y_{n-1} = x_1 x_2 \cdots x_{n-2} x_n = S_{b_{n-1} r}(\text{Tr}(g)). \end{array} \right. \right\} \\ \text{Rand} &= \left\{ (T, K) \left| \begin{array}{l} r_1, r_2, \dots, r_n, r, s_1, s_2, \dots, s_n \in_R Z_p, \\ z_1 = \text{Tr}(g^{r_1}), \dots, z_n = \text{Tr}(g^{r_n}), \quad z = \text{Tr}(g^r), \\ x_1 = \text{Tr}(g^{s_1}), \dots, x_n = \text{Tr}(g^{s_n}), \\ y_1 = x_2 x_3 \cdots x_n = S_{s_2 + s_3 + \dots + s_n}(\text{Tr}(g)), \dots, \\ y_{n-1} = x_1 x_2 \cdots x_{n-2} x_n = S_{s_1 + \dots + s_{n-2} + s_n}(\text{Tr}(g)). \end{array} \right. \right\}, \end{aligned} \quad (9)$$

where $b_i = \sum_{1 \leq j \leq n, j \neq i} r_j$ as in protocol XTR-CR, $T = (z, z_1, \dots, z_{n-1}, y_1, \dots, y_{n-1})$ is the transcript of message and $K = H(y_1, \dots, y_{n-1}, X)$ is the hash value. We now construct

a distribution Dist by the triple $(\text{Tr}(g^r), \text{Tr}(g^{r_2}), \text{Tr}(g^{r'r_2}))$, as below:

$$\text{Dist} = \left\{ (T, K) \left| \begin{array}{l} r_1, r_2, \alpha_3, \beta_3, \dots, \alpha_n, \beta_n \in_R Z_p, \\ z_1 = \text{Tr}(g^{r_1}), \quad z_2 = \text{Tr}(g^{r_2}), \quad z_3 = \text{Tr}(g^{r_1 \alpha_3 + r_2 \beta_3}), \dots, \\ z_n = \text{Tr}(g^{r_1 \alpha_n + r_2 \beta_n}), \quad z = \text{Tr}(g^r), \\ x_1 = \text{Tr}(g^{rr_1}), \quad x_2 = \text{Tr}(g^{rr_2}), \quad x_3 = \text{Tr}(g^{rr_1 \alpha_3 + r'r_2 \beta_3}), \dots, \\ x_n = \text{Tr}(g^{rr_1 \alpha_n + r'r_2 \beta_n}), \\ y_1 = x_2 x_3 \cdots x_n = S_{c_2 + c_3 + \dots + c_n}(\text{Tr}(g)), \dots, \\ y_{n-1} = x_1 x_2 \cdots x_{n-2} x_n = S_{c_1 + \dots + c_{n-2} + c_n}(\text{Tr}(g)). \end{array} \right. \right\}, \quad (10)$$

where T, K are defined above and $c_1 = rr_1, c_2 = rr_2, c_3 = rr_1 \alpha_3 + r'r_2 \beta_3, \dots, c_n = rr_1 \alpha_n + r'r_2 \beta_n$.

If $(\text{Tr}(g^r), \text{Tr}(g^{r_2}), \text{Tr}(g^{r'r_2}))$ is a XTR-DH triple (i.e., $r = r'$), then $\text{Dist} = \text{Real}$, since triple $(\text{Tr}(g), z_i, x_i) = (\text{Tr}(g), \text{Tr}(g^a), \text{Tr}(g^{ar}))$ for all $i \in [1, n]$. If not, $(\text{Tr}(g^r), \text{Tr}(g^{r_2}), \text{Tr}(g^{r'r_2}))$ is a random triple, so $\text{Dist} = \text{Rand}$. Thus a distribution is constructed to display the advantage of XTR-DDH.

Hence, let A' be an algorithm that, given (T, K) computing from Real and Rand distributions, runs in time t and outputs 0 or 1. Then, we have

$$\begin{aligned} & \left| \Pr[A'(T, K) = 1 \mid (T, K) \leftarrow \text{Real}] \right. \\ & \quad \left. - \Pr[A'(T, K) = 1 \mid (T, K) \leftarrow \text{Rand}] \right| \quad (11) \\ & \leq \text{Adv}_G^{\text{XTR-DDH}}(t + O(n \cdot t_{\text{exp}})). \end{aligned}$$

On the other hand, it is clear to find that the transcript T in experiment Rand is independent of s_i , which implies that for any computationally unbounded adversary A , we have

$$\begin{aligned} & \Pr[A(T, K_b) \mid (T, K_1) \leftarrow \text{Rand}; k_0 \leftarrow \{0, 1\}^l; b \leftarrow \{0, 1\}] \\ & \quad = \frac{1}{2}. \end{aligned} \quad (12)$$

The detail of construction of distinguisher D is given now. Assume without loss of generality that A makes its Test query to an oracle activated by the δ th Execute query. The distinguisher D begins by choosing a random bit as a guess for the value of δ . Then D invokes A and simulates the queries of A . D answers all the queries from A in a specified way, following the protocol, except in the d th query. In this latter case, D embed the DDH problem instance into the transcript as follows.

Given a triple $(\text{Tr}(g^r), \text{Tr}(g^{r_2}), \text{Tr}(g^{r'r_2})) \in \text{GF}(p^2)^3$, D generates (T, K) according to the distribution Dist and answers the d th Execute query of A with T . The distinguisher D aborts and outputs a random bit of $d \neq \delta$. Otherwise, D answers the Test query of A with K .

When A terminates and outputs its guess b' , D outputs 1 if $b = b'$, and 0 otherwise. If $(\text{Tr}(g^r), \text{Tr}(g^{r_2}), \text{Tr}(g^{r'r_2}))$ is a

XTR-DDH triple, from the deduction mentioned before, it is deduced that $\text{Dist} = \text{Real}$, and thus $\Pr[b = b'] = 1/2 + \varepsilon$. If not, $(\text{Tr}(g^r), \text{Tr}(g^{r_2}), \text{Tr}(g^{r_2}))$ is a random triple, and then $\text{Dist} = \text{Rand}$ and $\Pr[b = b'] = 1/2$. Finally, since $\Pr[d = \delta] = 1/q_{\text{ex}}$, we obtain

$$\text{Adv}_g^{\text{XTR-DDH}}(D) = \frac{\varepsilon}{q_{\text{ex}}}. \quad (13)$$

From $\text{Adv}_{A,P}(k) = 2\Pr[\text{CG}] - 1$, we have that

$$\frac{1}{2} + \varepsilon = \Pr[\text{CG}] = \Pr[b = b'] = \frac{(\text{Adv}_{A,P}(k) + 1)}{2}. \quad (14)$$

Therefore,

$$\begin{aligned} \text{Adv}_{A,P}(k) &= 2\varepsilon = 2q_{\text{ex}}\text{Adv}_G^{\text{XTR-DDH}}(D) \\ &\leq \text{Adv}_G^{\text{XTR-DDH}}(t + O(n \cdot t_{\text{exp}})). \end{aligned} \quad (15)$$

Thus the theorem follows. $\square$

The result of the above theorem shows that the security of proposed key agreement scheme is based on the computational difficulty of XTR-DDH problem. Also, it is shown that the proposed scheme is secure against adversary's attacking under XTR-DDH assumption.

Moreover, the result of this security proof also supports the idea of Steiner et al. [1], who proved that DDH assumption implied the G-DDH assumption. Here, G-DDH is short for group decisional Diffie-Hellman assumption, which refers to the difficulty of distinguishing $g^{\prod x_i}$ from a random value by knowing elements $g^{\prod x_i}$ for some subsets of indices i . Actually, Theorem 10 could be regarded as an evidence for this result.

5. Conclusions

In this paper, a constant round key agreement scheme XTR-CR is proposed on the basis of XTR cryptosystem. Three theorems are given to make a theoretical guarantee of the quick implementation of system. Moreover, under XTR-DDH assumption, the security of scheme is proved in the formal model. It is believed that this scheme is efficient both in communication and computation. Hence, this proposed scheme is reliable in the sense of achieving three efficient and secure demands: less communication, fast computation, and provable security.

Acknowledgments

This work is supported by doctoral fund of Huazhong Agricultural University, 52204-06072, National Natural Science Foundation of China (Grant no. 61202305), and Project 2013PY120 supported by the fundamental research funds for the central universities. Thanks are also given to the anonymous reviewers for their appreciated suggestions.

References

- [1] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in *Proceedings*

of the 3rd ACM Conference on Computer and Communications Security, pp. 31–37, March 1996.

- [2] M. Steiner, *Secure group key agreement [Ph.D. thesis]*, Universitat des Saarlandes, 2002.
- [3] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 60–96, 2004.
- [4] R. Dutta and R. Barua, "Dynamic group key agreement in tree-based setting," in *Proceedings of the 10th Australasian Conference on Information Security and Privacy (ACISP '05)*, pp. 101–112, July 2005.
- [5] J. Nam, J. Paik, U. M. Kim, and D. Won, "Constant-round authenticated group key exchange with logarithmic computation complexity," *Applied Cryptography and Network Security*, Springer, Heidelberg, Germany, vol. 4521, pp. 158–176, 2007.
- [6] D. Augot, R. Bhaskar, V. Issarny, and D. Sacchetti, "A three round authenticated group key agreement protocol for ad hoc networks," *Pervasive and Mobile Computing*, vol. 3, no. 1, pp. 36–52, 2007.
- [7] R. Tso, X. Yi, and E. Okamoto, "ID-based key agreement for dynamic peer groups in mobile computing environments," in *Proceedings of the 2nd IEEE Asia-Pacific Services Computing Conference (APSCC '07)*, pp. 103–108, December 2007.
- [8] X. Fu, Q. Xu, and H. Wang, "A provably-secure password-authenticated group key agreement in the standard model," *Journal of Networks*, vol. 4, no. 8, pp. 763–770, 2009.
- [9] G. Yao and D.-G. Feng, "Pairwise key agreement protocols based on the weil pairing," *Journal of Software*, vol. 17, no. 4, pp. 907–914, 2006.
- [10] A. Joux, "A one round protocol for tripartite Diffie-Hellman," in *Algorithmic Number Theory*, vol. 1838 of *Lecture Notes in Computer Science*, pp. 385–394, 2001.
- [11] X. Lv and H. Li, "ID-based authenticated group key agreement from bilinear maps," *Frontiers of Computer Science in China*, vol. 4, no. 2, pp. 302–307, 2010.
- [12] D. He, Y. Chen, J. Chen, R. Zhang, and W. Han, "A new two-round certificateless authenticated key agreement protocol without bilinear pairings," *Mathematical and Computer Modelling*, vol. 54, no. 11–12, pp. 3143–3152, 2011.
- [13] M.-H. Zheng, H.-H. Zhou, J. Li, and G.-H. Cui, "Efficient and provably secure password-based group key agreement protocol," *Computer Standards and Interfaces*, vol. 31, no. 5, pp. 948–953, 2009.
- [14] G. Yao, H. Wang, and Q. Jiang, "An authenticated 3-round identity-based group key agreement protocol," in *Proceedings of the 3rd International Conference on Availability, Security, and Reliability (ARES '08)*, pp. 538–543, March 2008.
- [15] T. Feng, Y. Wang, and J. Ma, "A secure and efficient group key agreement for ad hoc networks," in *Proceedings of the International Symposium on Computer Science and Computational Technology (ISCST '08)*, pp. 540–543, December 2008.
- [16] E. Makri and E. Konstantinou, "Constant round group key agreement protocols: a comparative study," *Computers and Security*, vol. 30, no. 8, pp. 643–678, 2011.
- [17] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably authenticated group Diffie-Hellman key exchange—the dynamic case," in *Advances in Cryptology—ASIACRYPT 2001*, C. Boyd, Ed., vol. 2248 of *Lecture Notes in Computer Science*, pp. 290–309, Springer, Berlin, Germany, 2001.

- [18] K. K. Raymond Choo, C. Boyd, and Y. Hitchcock, "The importance of proofs of security for key establishment protocols: formal analysis of Jan-Chen, Yang-Shen-Shieh, Kim-Huh-Hwang-Lee, Lin-Sun-Hwang, and Yeh-Sun protocols," *Computer Communications*, vol. 29, no. 15, pp. 2788–2797, 2006.
- [19] A. K. Lenstra and E. R. Verheul, "The XTR public key system," in *Advances in Cryptology*, vol. 1880 of *Lecture Notes in Computer Science*, pp. 1–19, Springer, Berlin, Germany, 2000.
- [20] A. K. Lenstra and E. R. Verheul, "An overview of the XTR public key system," in *Proceedings of the Conference on Public Key Cryptography and Computational Number Theory*, pp. 151–180, Warsaw, Poland, 2000.
- [21] M. Stam and A. K. Lenstra, "Speeding up XTR," in *Advances in Cryptology—ASIACRYPT 2001*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 125–143, Springer, Berlin, Germany, 2001.
- [22] E. Verheul, "Evidence that XTR is more secure than supersingular ellipticcurve cryptosystems," in *Advances in Cryptology—EUROCRYPT 2001*, B. Pfitzmann, Ed., vol. 2045 of *Lecture Notes in Computer Science*, pp. 195–210, Springer, Berlin, Germany, 2001.
- [23] K.-A. Shim, "A round-optimal three-party ID-based authenticated key agreement protocol," *Information Sciences*, vol. 186, no. 1, pp. 239–248, 2012.
- [24] H. Xiong, Z. Chen, and F. Li, "New identity-based three-party authenticated key agreement protocol with provable security," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 927–932, 2013.