SCIENCE CHINA
Information Sciences

• LETTER •

May 2018, Vol. 61 059102:1–059102:3
doi: 10.1007/s11432-017-9177-7

New constructions for (multiparty) one-round key exchange with strong security

Zheng YANG[1,2] & Junyu LAI[3*]

[1] *School of Computer Science and Engineering, Chongqing University of Technology, Chongqing* 400054*, China;*
[2] *Department of Computer Science, University of Helsinki, Helsinki* 00014*, Finland;*
[3] *School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu* 610054*, China*

Received 23 March 2017/Revised 16 May 2017/Accepted 21 June 2017/Published online 17 November 2017

Citation Yang Z, Lai J Y. New constructions for (multiparty) one-round key exchange with strong security. Sci China Inf Sci, 2018, 61(5): 059102, doi: 10.1007/s11432-017-9177-7

Dear editor,

One-round key exchange (ORKE) represents a special class of two-message key exchange. It enables two parties to establish a shared session key within only one communication round. The first merit of ORKE is its communication efficiency, particularly because two parties are able to simultaneously run the protocol instance. Therefore, the messages of two session participants (in an ORKE protocol) are completely independent. This property of ORKE could dramatically reduce communication latency. Another advantage of ORKE is its strong security. We stress that the recently proposed ORKE protocols have been shown, without exception, to provide security in very strong security models for authenticated key exchange (AKE). A prominent example of the strong AKE model is the g-eCK-PFS model defined by Yang and Zhang [1], which is extended from the two party eCK-PFS model [2]. It is, surprisingly, the g-eCK-PFS model that can be used to provide security arguments for either two party or multiparty one-round key exchange (MORKE). The g-eCK-PFS model formulates almost all other desirable security properties for AKE, such as perfect forward secrecy, resilience to key compromise impersonation (KCI) attacks, resilience to chosen public key attacks, and resilience to exposure attacks on long-term or ephemeral secret keys.

An important factor that affects ORKE constructions is peer setting. The ORKE protocol is normally constructed and analyzed in the so-called pre- and post-specified peer settings (we refer to them respectively as pre- and post-settings for short). In the pre-setting, it is assumed that a party knows all the cryptographic information of its intended peer for key exchange when it starts a protocol instance. In contrast, a party does not know any cryptographic information of the receiver in the post-setting. Up to now, only few strongly secure two party ORKE protocols and MORKE protocols have been introduced in the post-setting without random oracles. However, these schemes are either based on strong security assumptions or quite computationally inefficient. Moreover, after decades of research, we still do not have a concrete strongly secure ORKE solution in the standard model that does not rely on bilinear groups. The question arises as to whether it is possible to build two party ORKE in a post-setting based on the traditional decisional Diffie-Hellman (DDH) assumption (or without pairing). In addition, the first g-eCK secure MORKE protocol without random oracles was introduced by Li and Yang [3] (which is referred to as the LY-g scheme). The LY-g scheme is based on multilinear groups,

[*] Corresponding author (email: laijy@uestc.edu.cn)
The authors declare that they have no conflict of interest.

and is quite inefficient. For n group members, it requires $4n+1$ multilinear map operations. The question then arises as to whether we can build a more efficient multiparty ORKE protocol whose multilinear map operation is independent of the number n of group members.

Our contributions. In order to answer the above questions, we first propose a new construction for two-party ORKE that is secure in the eCK-PFS model without random oracles in the post-setting. This protocol is mainly motivated by the need to overcome the DDH reduction problem introduced in [4]. The security of our scheme is reduced to the DDH problem, digital signature scheme, double pseudo-random function [5], and collision-resistant hash function. Our scheme is also the first eCK(-PFS)-like secure concrete scheme in the post-setting without both pairing and random oracles. Although our two party ORKE protocol is somewhat inefficient, it could also serve as a simple example to exhibit the construction principle of our subsequent multiparty ORKE protocol.

In the second contribution, we specifically show how to apply the construction idea of our DDH-based ORKE scheme to build MORKE under multilinear maps. Our new MORKE protocol is much more efficient than the previous schemes, e.g., the one by Li and Yang [3]. As we know, the cost of a multilinear map operation is very expensive. In particular, we reduce the number of multilinear map operations from $O(n)$ in [3] to $O(1)$ in our scheme, where n is the number of group members in a session. We highlight that the multilinear map operation in our scheme is independent of n. This is the first MORKE scheme that achieves this level of performance in the standard model. This result significantly shows the practicability our MORKE protocol in real-world applications such as secure asynchronous message software.

Preliminaries. We let $\lambda \in \mathbb{N}$ be the security parameter and 1^λ be a string that consists of λ ones. The notation $a \xleftarrow{\$} S$ denotes the operation which samples a uniformly random element from a set S. We denote the binary representation of a value h with size μ as $h = (h(1), h(2), \dots, h(\mu)) = \{0,1\}^\mu$.

In our proposed schemes, we mainly makes use of the building blocks including a digital signature scheme SIG, (double) pseudo-random functions, and a collision resistant hash function CRHF.

We consider a digital signature scheme SIG that consists of tree probabilistic polynomial time algorithms: a key generation algorithm $\mathsf{SIG.Gen}$ (which on input 1^λ outputs a verification key vk and a signing key sk), a signing algorithm $\mathsf{SIG.Sign}$ (which generates a signature σ for message m with signing key sk), and a verification algorithm $\mathsf{SIG.Vfy}$ (which on input vk, m and σ outputs 1 if σ is valid, and 0 otherwise). We need SIG to be secure against strong existential forgeries $\mathcal{F}$ under weak chosen-message attacks (SEUF-WCMA), as defined in [6].

Double pseudo-random functions [5] is a family of deterministic functions, which is defined with two keys from distinct key spaces $\mathcal{K}_{\mathsf{DPRF}_1}$ and $\mathcal{K}_{\mathsf{DPRF}_2}$. We here let $\mathsf{DPRF} : \mathcal{K}_{\mathsf{DPRF}_1} \times \mathcal{K}_{\mathsf{DPRF}_2} \times \mathcal{M}_{\mathsf{DPRF}} \rightarrow \mathcal{R}_{\mathsf{DPRF}}$, where $\mathcal{M}_{\mathsf{DPRF}}$ is the domain and $\mathcal{R}_{\mathsf{DPRF}}$ is the range of DPRF. Let $\mathsf{CRHF} : \mathcal{K}_{\mathsf{CRHF}} \times \mathcal{M}_{\mathsf{CRHF}} \rightarrow \mathcal{Y}_{\mathsf{CRHF}}$ be a family of keyed-hash functions, where $\mathcal{K}_{\mathsf{CRHF}}$ is the key space, $\mathcal{M}_{\mathsf{CRHF}}$ is the message space and $\mathcal{Y}_{\mathsf{CRHF}}$ is the hash value space. We usually write $\mathsf{CRHF}(m)$ for $\mathsf{CRHF}(\mathrm{hk}_{\mathsf{CRHF}}, m)$ when hash key $\mathrm{hk}_{\mathsf{CRHF}}$ is obvious from the context.

Moreover, the session key security of our first and second proposals mainly relies on DDH and multilinear decisional Diffie-Hellman (MDDH) assumptions, respectively. Let $\mathbb{G}$ be a group of prime order p. Let g be a random generator of $\mathbb{G}$. DDH problem is stated as follows: given tuple (g, g^a, g^b, g^c) for $a, b, c \xleftarrow{\$} \mathbb{Z}_p^*$ as input, it is hard to distinguish whether $c = ab$. Let MGen be a symmetric multilinear groups generator, which, on input security parameter λ and positive integer $2 \leqslant n \in \mathbb{N}$ outputs two multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of the same prime order p, generator g for $\mathbb{G}$, and multilinear map $\mathrm{me} : \mathbb{G}^n \rightarrow \mathbb{G}_T$. We review a generalization of DDH assumption in multilinear groups $\mathcal{MLG} = (\mathbb{G}, \mathbb{G}_T, g, p, \mathrm{me})$, i.e., MDDH assumption. Roughly speaking, MDDH problem is stated as follows. Given $(g, g^{a_1}, \dots, g^{a_{n+1}}, \mathrm{me}(g, \dots, g)^\gamma)$ for $(a_1, \dots, a_{n+1}, \gamma) \in (\mathbb{Z}_p^*)^{n+1}$ as input, output 1 if $\gamma = \prod_{i=1}^{n+1} a_i$ and 0 otherwise.

DDH-based ORKE protocol. We here introduce a new two party ORKE (2ORKE) protocol as a warm up (for our upcoming MORKE scheme). 2ORKE is proposed to achieve the eCK-PFS security [6] in the standard model from the DDH assumption. Let $\mathcal{K}_{\mathsf{LL}}$ be a long-term key space and $\mathcal{K}_{\mathsf{EPH}}$ be an ephemeral key space for AKE.

Protocol description. The protocol consists of the following parts.

SETUP: Generate a DDH group ($\mathbb{G}$ with a prime order p and a group generator g. Choose a random hash key $\mathrm{hk}_{\mathsf{CRHF}} \xleftarrow{\$} \mathcal{K}_{\mathsf{CRHF}}$.

LONG-TERM KEY GENERATION: The long-term secret/public key pair $(\mathrm{sk}_{\mathsf{id}}, \mathrm{pk}_{\mathsf{id}})$ of a party id is generated as follows: $\mathrm{ss}_{\mathsf{id}} \xleftarrow{\$} \mathcal{K}_{\mathsf{LL}}$, $(\mathrm{sk}^{\mathrm{sig}}_{\mathsf{id}}, \mathrm{vk}^{\mathrm{sig}}_{\mathsf{id}}) \xleftarrow{\$} \mathsf{SIG.Gen}(1^\lambda)$ $\mathrm{sk}_{\mathsf{id}} = (\mathrm{ss}_{\mathsf{id}}, \mathrm{sk}^{\mathrm{sig}}_{\mathsf{id}_1})$, and $\mathrm{pk}_{\mathsf{id}} = \mathrm{vk}^{\mathrm{sig}}_{\mathsf{id}}$.

PROTOCOL EXECUTION: We consider the protocol execution between two parties $(\mathsf{id}_1, \mathsf{id}_2)$, where party id_i $(1 \leqslant i \leqslant 2)$ has long-term key $\mathrm{pk}_{\mathsf{id}_i}$. In the key exchange phase, id_i performs the following steps:

(1) First choose ephemeral secret keys $(\mathrm{es}_{\mathsf{id}_i}, m_{\mathsf{id}_i}) \xleftarrow{\$} \mathcal{K}_{\mathsf{EPH}}$, and then generate the intermediate secrets for ephemeral key generation as: $(x_{\mathsf{id}_i,\eta,\iota})_{(\eta,\iota)\in[\mu]\times\{0,1\}} \leftarrow \mathsf{DPRF}(\mathrm{ss}_{\mathsf{id}_i}, \mathrm{es}_{\mathsf{id}_i}, m_{\mathsf{id}_i})$.

(2) Generate the ephemeral public key as $\mathrm{epk}_{\mathsf{id}_i} = (X_{\mathsf{id}_i,\eta,\iota})_{(\eta,\iota)\in[\mu]\times\{0,1\}} := (g^{x_{\mathsf{id}_i,\eta,\iota}})_{(\eta,\iota)\in[\mu]\times\{0,1\}}$, choose a randomness $\mathrm{rs}_{\mathsf{id}_i} \in \mathcal{RS}_{\mathsf{SIG}}$, and compute a signature $\sigma_{\mathsf{id}_i} \leftarrow \mathsf{SIG.Sign}(\mathrm{sk}^{\mathrm{sig}}_{\mathsf{id}_i}, T_{\mathsf{id}_i}; \mathrm{rs}_{\mathsf{id}_i})$, where $T_{\mathsf{id}_i} = \mathsf{id}_i || \mathrm{epk}_{\mathsf{id}_i}$.

(3) Send $(\sigma_{\mathsf{id}_i}, T_{\mathsf{id}_i})$ to its intended communication partner id_j $(1 \leqslant j \leqslant 2,\ i \neq j)$.

(4) Upon receiving a message tuple $(\sigma_{\mathsf{id}_j}, T_{\mathsf{id}_j})_{1\leqslant j\leqslant 2, j\neq i}$ from its partner, reject the session if the corresponding signatures is invalid, i.e., $\mathsf{SIG.Vfy}(\mathrm{vk}^{\mathrm{sig}}_{\mathsf{id}_j}, \sigma_{\mathsf{id}_j}, T_{\mathsf{id}_j}) \neq 1$.

(5) Set $\mathsf{sid} := T_{\mathsf{id}_1} || \sigma_{\mathsf{id}_1} || T_{\mathsf{id}_2} || \sigma_{\mathsf{id}_2}$ and computes $h = (h(1), h(2), \ldots, h(\mu)) := \mathsf{CRHF}(\mathsf{sid})$.

Finally, the session key is generated as follows:

$$k := \left(\prod_{\eta=1}^{\mu} X_{\mathsf{id}_2,\eta,h(\eta)}\right)^{\sum_{\iota=1}^{\mu} x_{\mathsf{id}_1,\iota,h(\iota)}} = \left(\prod_{\eta=1}^{\mu} X_{\mathsf{id}_1,\eta,h(\eta)}\right)^{\sum_{\iota=1}^{\mu} x_{\mathsf{id}_2,\iota,h(\iota)}}. \qquad (1)$$

Efficient multiparty ORKE protocol. We now extend 2ORKE to the group case that yields an efficient g-eCK-PFS secure [1] MORKE protocol without random oracles. MORKE utilizes similar building blocks to those of 2ORKE, but the multilinear groups are used instead.

Protocol description. The protocol is described below in a general way.

SETUP: Generate multilinear groups $\mathcal{MG} = (\mathbb{G}, g, \mathbb{G}_T, p, e) \xleftarrow{\$} \mathsf{MGen}(1^\lambda, n)$. Choose a random hash key $\mathrm{hk}_{\mathsf{CRHF}} \xleftarrow{\$} \mathcal{K}_{\mathsf{CRHF}}$.

LONG-TERM KEY GENERATION: This part is similar to that of 2ORKE.

PROTOCOL EXECUTION: We consider the protocol execution with $n+1$ group members $(\mathsf{id}_1, \ldots, \mathsf{id}_{n+1})$, where each party id_i $(1 \leqslant i \leqslant n+1)$ has long-term key $\mathrm{pk}_{\mathsf{id}_i}$. In the key exchange phase, each party id_i performs the following steps:

(1) First choose ephemeral secret keys $(\mathrm{es}_{\mathsf{id}_i}, m_{\mathsf{id}_i}) \xleftarrow{\$} \mathcal{K}_{\mathsf{EPH}}$ and then generate the intermediate secrets for ephemeral key generation as $(x_{\mathsf{id}_i,\eta,\iota})_{(\eta,\iota)\in[\mu]\times\{0,1\}} \leftarrow \mathsf{DPRF}(\mathrm{ss}_{\mathsf{id}_i}, \mathrm{es}_{\mathsf{id}_i}, m_{\mathsf{id}_i})$.

(2) Generate the ephemeral public key as $\mathrm{epk}_{\mathsf{id}_i} = (X_{\mathsf{id}_i,\eta,\iota})_{(\eta,\iota)\in[\mu]\times\{0,1\}} := (g^{x_{\mathsf{id}_i,\eta,\iota}})_{(\eta,\iota)\in[\mu]\times\{0,1\}}$, choose a randomness $\mathrm{rs}_{\mathsf{id}_i} \in \mathcal{RS}_{\mathsf{SIG}}$, and compute a signature $\sigma_{\mathsf{id}_i} \leftarrow \mathsf{SIG.Sign}(\mathrm{sk}^{\mathrm{sig}}_{\mathsf{id}_i}, T_{\mathsf{id}_i}; \mathrm{rs}_{\mathsf{id}_i})$, where $T_{\mathsf{id}_i} = \mathsf{id}_i || \mathrm{epk}_{\mathsf{id}_i}$.

(3) Broadcast $(\sigma_{\mathsf{id}_i}, T_{\mathsf{id}_i})$ to its intended communication partners.

(4) Upon receiving $\{(\sigma_{\mathsf{id}_j}, T_{\mathsf{id}_j})\}_{1\leqslant j\leqslant n+1, j\neq i}$ from each session participant, reject the session if one of the signatures is invalid, i.e., $\mathsf{SIG.Vfy}(\mathrm{vk}^{\mathrm{sig}}_{\mathsf{id}_j}, \sigma_{\mathsf{id}_j}, T_{\mathsf{id}_j}) \neq 1$.

(5) Set $\mathsf{sid} := T_{\mathsf{id}_1} || \sigma_{\mathsf{id}_1} || \ldots || T_{\mathsf{id}_n} || \sigma_{\mathsf{id}_{n+1}}$ and compute $h = (h(1), h(2), \ldots, h(\mu)) := \mathsf{CRHF}(\mathsf{sid})$.

Finally, the session key is generated as follows:

$$k := \mathrm{me}\left(\prod_{\iota=1}^{\mu} X_{\mathsf{id}_1,\iota,h(\iota)}, \ldots, \prod_{\iota=1}^{\mu} X_{\mathsf{id}_{i-1},\iota,h(\iota)}, \prod_{\iota=1}^{\mu} X_{\mathsf{id}_{i+1},\iota,h(\iota)}, \ldots, \prod_{\iota=1}^{\mu} X_{\mathsf{id}_{n+1},\iota,h(\iota)}\right)^{\sum_{\eta=1}^{\mu} x_{\mathsf{id}_i,\eta,h(\eta)}}. \qquad (2)$$

Acknowledgements This work was supported by Research Project of National Natural Science Foundation of China (Grant Nos. 11647097, 61402085) and Humanities and Social Sciences of the Ministry of Education of China (Grant No. 16YJC870018).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Yang Z, Zhang D G. Towards modelling perfect forward secrecy for one-round group key exchange. Int J Netw Secur, 2016, 18: 304–315

2 Cremers C, Feltz M. Beyond eCK: perfect forward secrecy under actor compromise and ephemeral-key reveal. Des Code Crypt, 2015, 74: 183–218

3 Li Y, Yang Z. Strongly secure one-round group authenticated key exchange in the standard model. In: Proceedings of International Conference on Cryptology and Network Security. Berlin: Springer, 2013. 122–138

4 Yang Z, Li S. On security analysis of an after-the-fact leakage resilient key exchange protocol. Inf Process Lett, 2016, 116: 33–40

5 Li Y, Schäge S, Yang Z, et al. On the security of the pre-shared key ciphersuites of TLS. In: Proceedings of International Workshop on Public Key Cryptography. Berlin: Springer, 2014. 669–684

6 Yang Z, Lai J Y, Liu C, et al. Simpler generic constructions for strongly secure one-round key exchange from weaker assumptions. Comput J, 2017, 60: 1145–1160