One-Round Cross-Domain Group Key Exchange Protocol in the Standard Model

Xiao Lan[1,4], Jing Xu[2(✉)], Hui Guo[3], and Zhenfeng Zhang[2]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
lanxiao@iie.ac.cn
[2] Trusted Computing and Information Assurance Laboratory,
Institute of Software, Chinese Academy of Sciences, Beijing, China
{xujing,zfzhang}@tca.iscas.ac.cn
[3] State Key Laboratory of Cryptology, Beijing, China
sklcguohui@163.com
[4] University of Chinese Academy of Sciences, Beijing, China

Abstract. Cross-domain group key exchange protocols enable participants from different domains, even with various cryptographic settings and system parameters, to establish a common secret session key. In prior cross-domain key exchange works, only the case of two communication parties is considered, and the two parties are required to adopt a common cryptographic setting (e.g., identity-based setting) or shared parameters (e.g., algebraic group), which is not suitable for group data sharing in many cross-domain interoperability scenarios. In this paper, we present the first one-round cross-domain group key exchange protocol, and by using indistinguishability obfuscation as the main tool, we prove our construction can achieve the desired security properties in the standard model. It is especially attractive for our protocol that existing PKIs can be used and all participants do not have to accommodate any other peers (even do not need to know other peers' algebraic settings) to agree on the session key.

Keywords: Group key exchange protocol · Cross-domain · Interoperability · Indistinguishability obfuscation · Standard model

1 Introduction

Secure group communication is an increasingly popular research area and has received much attention in modern collaborative and distributed applications such as distributed social networks, peer-to-peer file sharing, and cloud computing. Group key exchange protocols are fundamental to secure communication among a group of users. In a group key exchange protocol, a group of users are allowed to communicate over an untrusted, open network to agree on a common secret session key and thereafter, they can securely exchange messages using this shared key.

K. Chen et al. (Eds.): Inscrypt 2016, LNCS 10143, pp. 386–400, 2017.
DOI: 10.1007/978-3-319-54705-3_24

With the popularity of group data sharing in distributed networks, cross-domain group key exchange protocols have become the basis of securely connecting distributed multi-domain systems. Each domain environment would have its own users and resources within specific trust domain, however, since diverse type of requirements can be made by the users, which may not be offered by one single domain system, one domain system has to request another domain system or multiple domain systems. Therefore, the demand of cooperative work in multiple domains, i.e., cross-domain interoperability, is rising. Nonetheless, cross-domain group key exchange protocols are hard to design for its complexity in system deployment and user operation, all of which need large amount computation and resource consumption. In particular, there are many differences in the design between cross-domain authenticated group key exchange protocol and two-party key exchange protocol. First, the users' structure is more complex: in two-party case, users are on equal status, while in the group case, users are usually in the ring structure, tree structure, or line structure; second, the parameters setting is more universal: in two-party case, the same cryptographic setting is used, while in the group case, the users may be in various algebraic settings; third, the round of the protocol is more dynamic: the two-party case usually has constant round, while in the group case, the round is closely linked with the group structure and size, usually increasing with the group size.

Over the past several years, many solutions to group key exchange protocols have been proposed [1–13]. However, all of these constructions require all participants to adopt a common cryptographic setting and shared parameters. In practical applications, the common scheme and parameter requirements can be a large barrier when entities coming from different settings wish to communicate with each other. Taking an example of signature, existing users have already established signing keys and algorithms which are entrenched in an existing public key infrastructure. The changing and re-certifying of one's public keys may bring much resource consumption and make the user store many suits of keys, which absolutely results in complexity of operation. Aiming at tackling the challenges above, we propose a one-round cross-domain group key exchange protocol which removes the complex group structure, and most of all, it allows group members to come from different cryptographic settings (e.g., identity-based setting, certificate-based setting) and use different signature schemes (e.g., RSA, ECDSA).

1.1 Related Work

Group Key Exchange Protocol. Burmester et al. [4] proposed an efficient and practical group key exchange protocol, in which the number of the communication rounds is constant when broadcast messages are allowed, however, there is no security proof for it. Later, Bresson et al. [7] introduced a formal security model for group key exchange protocols based on the Bellare and Rogaway model [14] and proposed the first provably secure protocol in this setting. Users in their protocol communicate in a ring structure, and only after receiving

messages from his predecessor, the user can produce his own message. Unfortunately, the essence of their communication structure makes their protocol quite impractical for large groups due to the number of communication rounds linear in the number of group users. In 2003, Katz and Yung [10] analyzed Burmester's protocol [4], who also proposed the first constant round and fully scalable authenticated group key exchange protocol which is provably secure in the standard model. Besides this, there are some identity-based group key exchange protocols [2,5,13], using the identity information in place of public keys to provide authentication. Recently, Boneh and Zhandry [15] constructed the first multi-party non-interactive key exchange protocol requiring trusted setup based on indistinguishability obfuscation, and gave the formal security proof in the static and semi-static models, however, their protocol does not consider entity authentication, and moreover the group session key is generated only by group users' public keys, which makes the session key static and fixed.

Cross-Domain Key Exchange Protocol. Chen et al. [16] introduced the concept of two-party cross-domain communication and proposed an ID-based protocol that allows two parties to communicate through different domains. In 2005, McCullagh et al. [17] proposed a more efficient cross-domain two-party construction. However, both of constructions [16,17] require all parties from different domains adopt the common group parameter. Ustaoğlu [18] also proposed a collection of integrating protocols which support interoperability between two different cryptographic settings, but their protocols still require that the participants use parameters from the same algebraic group. Later, Guo et al. [19] proposed a two-party key exchange protocol where one entity is certificate-based and the other one is identity-based, and the parameters of both entities may come from different groups. Recently, Chen et al. [20] proposed a cross-domain four-party password-based authenticated key exchange protocol in a scenario that two cross-domain clients establish secure communication through their servers, which is a nice work but needs the client share password with its server. In summary, it seems that no existing solutions can perfectly support cross-domain group key exchange while not changing participants' existing cryptographic settings.

Obfuscation and Its Security. Obfuscation was first rigorously defined and studied by Barak et al. [21]. Roughly speaking, obfuscation security requires an obfuscated version $\mathcal{O}(\mathrm{P})$ of a program P to behave like a virtual black box (VBB) in the sense that anything one can compute given $\mathcal{O}(\mathrm{P})$, one could also compute from the input-output behavior of the program P. However, it has been known that it is impossible to realize it in general. This leads to an alternative and weaker notion called indistinguishability obfuscation ($i\mathcal{O}$), which requires that if two programs of the same size compute the same function, then their obfuscations should be indistinguishable. In 2013, Garg et al. [22] (known as GGH13) proposed the first candidate construction of an efficient $i\mathcal{O}$ for all circuits. Since their breakthrough result, an extremely large number of uses for $i\mathcal{O}$ in cryptography have been found, not only in obtaining classical cryptographic primitives, but also in reaching new possibilities. Subsequently, several other candidate $i\mathcal{O}$ schemes have been proposed, and almost all known schemes rely

on multilinear maps. Unfortunately, there have been several attacks [23–25] on multilinear maps that exploit extra information revealed by the zero-test procedure. However, known attacks exploit the correlations among ring elements, and these correlations are much harder to leverage in the case where only "highest-level" zero-encodings can be obtained, which is the case for known obfuscation candidates. Therefore, such attacks are not applicable to candidate $i\mathcal{O}$ schemes. The only known attacks against obfuscation schemes are the recent annihilation attacks of Miles et al. [25]. However, not all the obfuscation candidates are broken by the annihilation attacks. Recently, Garg, Mukherjee et al. [26] gave a beautiful new candidate $i\mathcal{O}$ construction, using a new variant of the GGH13 multilinear map candidate, and proved its security in the weak multilinear map model assuming an explicit PRF in NC^1. Concurrently, Lin [27] also proposed a construction of $i\mathcal{O}$ from a simple assumption (joint-SXDH assumption) on prime-order graded encodings.

1.2 Technical Contributions

Cross-domain group key exchange (CDGKE) protocols are fundamental building blocks for securing communication over public, insecure cross-domain networks. In this paper, we propose the first universal cross-domain group key exchange protocol. In a universal cross-domain group key exchange protocol, users coming from different domains (with various cryptographic settings and system parameters) communicate over an insecure public network and establish a common secret session key.

Our primary challenge is how to create a way to make all the participants have the uniform computation even though they are coming from different settings, and then hide the computation result from the outsiders. Inspired by Boneh and Zhandry's multiparty non-interactive key exchange scheme [15], we use indistinguishability obfuscation as the main tool. The essential idea is the following: the global agreed domain parameter consists of an obfuscated program for a constrained pseudorandom function PRF which requires to operate the verification of signature, and each user P_i generates a signature on the message x_i chosen randomly using its own signature scheme and broadcasts it. By running the global agreed domain parameter program, each user in the group can independently evaluate the obfuscated program to obtain the shared session key, which is the PRF output evaluated at the concatenation of the message x_i. However, such an approach fails because a signature can be replayed by an adversary. To prevent such attacks, we require the random value s_i used for generating the message x_i also as the input of the obfuscated program.

Compared to existing constructions, our protocol has a number of advantages: (i) It is optimal in terms of round complexity, which is a central measure of efficiency for any interactive protocol; (ii) Each participant neither needs to change or re-certify his public keys, nor holds many suites of keys; (iii) Each participant in the group may use different signature scheme (e.g., BLS, RSA, ECDSA, or FS-IBS) even in various algebraic settings (e.g., using RSA in different modulo), which is more suitable for cross-domain setting; (iv) Each participant does not

need to know the exact identity of any other participant, only the identifier in the group; (v) The group session key is different in each protocol execution even though the group users are not changed; (vi) It is provably secure in the standard model. It is also worth noting that since our protocol is built from a generic indistinguishability obfuscation mechanism other than secure multilinear maps, it may eventually depend on a weaker complexity assumption.

2 Preliminaries

In this section we start by briefly recalling the definitions of different cryptographic primitives essential for our study. Let $x \leftarrow S$ denote a uniformly random element drawn from the set S and λ the security parameter.

2.1 Indistinguishability Obfuscation

Definition 1 (Indistinguishability Obfuscation [22]). An *indistinguishability obfuscator* $i\mathcal{O}$ for a circuit class $\mathcal{C}_\lambda$ is a probabilistic polynomial time (PPT) algorithm satisfying the following conditions:

- $i\mathcal{O}(\lambda, C)$ preserves the functionality of C. That is, for any $C \in \mathcal{C}_\lambda$, if we compute $C' = i\mathcal{O}(\lambda, C)$, then $C'(x) = C(x)$ for all inputs x.
- For any λ and any two circuits $C_0, C_1 \in \mathcal{C}_\lambda$ with the same functionality, the circuits $i\mathcal{O}(\lambda, C_0)$ and $i\mathcal{O}(\lambda, C_1)$ are indistinguishable. More precisely, for all pairs of PPT adversaries $(Samp, D)$ there exists a negligible function α such that, if

$$\Pr[\forall x, C_0(x) = C_1(x) : (C_0, C_1, \tau) \leftarrow Samp(\lambda)] > 1 - \alpha(\lambda),$$

  then

$$|\Pr[D(\tau, i\mathcal{O}(\lambda, C_0)) = 1] - \Pr[D(\tau, i\mathcal{O}(\lambda, C_1)) = 1]| < \alpha(\lambda).$$

In this paper, we will make use of such indistinguishability obfuscators for all polynomial-size circuits.

Definition 2 (Indistinguishability Obfuscation for P/*poly*). A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator for P/*poly* if the following holds: Let $\mathcal{C}_\lambda$ be the class of circuits of size at most λ, Then $i\mathcal{O}$ is an indistinguishability obfuscator for the class $\{\mathcal{C}_\lambda\}$.

2.2 Constrained Pseudorandom Functions

A pseudorandom function (PRF) [28] is a function PRF: $\mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ where $\mathrm{PRF}(k, \cdot)$ is indistinguishable from a random function for a randomly chosen key k. Following Boneh and Waters [29], we recall the definition of constrained pseudorandom function[1].

[1] The Boneh and Waters's construction for the class of circuit-constrained PRFs [29] is based on the multilinear maps, however, to the best of our knowledge, there does

Definition 3 (Constrained Pseudorandom Function [29]). A PRF $F: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is said to be *constrained* with respect to a set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$ if there is an additional key space $\mathcal{K}_{\mathcal{C}}$ and two additional algorithms:

- $F.constrain(k, S)$: On input a PRF key $k \in \mathcal{K}$ and the description of a set $S \in \mathcal{S}$ (so that $S \subseteq \mathcal{X}$), the algorithm outputs a constrained key $k_S \in \mathcal{K}_{\mathcal{C}}$.
- $F.eval(k_S, x)$: On input $k_S \in \mathcal{K}_{\mathcal{C}}$ and $x \in \mathcal{X}$, the algorithm outputs

$$F.eval(k_S, x) = \begin{cases} F(k, x) & \text{if } x \in S \\ \perp & \text{otherwise} \end{cases}$$

For ease of presentation, we use $F(k_S, x)$ to represent $F.eval(k_S, x)$.

Security. Intuitively, we require that even after obtaining several constrained keys, no polynomial time adversary can distinguish a truly random string from the PRF evaluation at a point not queried. This intuition can be formalized by the following security game between a challenger and an adversary $\mathcal{A}$.

Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a constrained PRF with respect to a set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$. The security game consists of three phases:

Setup Phase. The challenger chooses a random key $K \leftarrow \mathcal{K}$ and a random bit $b \leftarrow \{0, 1\}$.

Query Phase. In this phase, $\mathcal{A}$ is allowed to ask for the following queries:

- Evaluation Query: On input $x \in \mathcal{X}$, it returns $F(K, x)$.
- Key Query: On input $S \in \mathcal{S}$, it returns F.constrain(K, S).
- Challenge Query: $\mathcal{A}$ sends $x \in \mathcal{X}$ as a challenge query. If $b = 0$, the challenger outputs $F(K, x)$; else, the challenger outputs a random element $y \leftarrow \mathcal{Y}$.

Guess Phase. $\mathcal{A}$ outputs a guess b' of b.

Let $E \subseteq \mathcal{X}$ be the set of evaluation queries, $C \subseteq \mathcal{S}$ be the set of constrained key queries and $Z \subseteq \mathcal{X}$ the set of challenge queries. $\mathcal{A}$ wins if $b = b'$ and $E \cap Z = \phi$ and $C \cap Z = \phi$. The advantage of $\mathcal{A}$ is defined to be $\mathrm{Adv}^F_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A} \text{ wins}] - 1/2|$.

Definition 4. The PRF F is a secure constrained PRF with respect to $\mathcal{S}$ if for all probabilistic polynomial time adversaries $\mathcal{A}$, $\mathrm{Adv}^F_{\mathcal{A}}(\lambda)$ is negligible in λ.

2.3 Signature Scheme

A digital signature scheme is a triple $SIG = (Sig.Gen, Sig.Sign, Sig.Verify)$, consisting of a key generation algorithm $(pk, sk) \leftarrow Sig.Gen(1^{\lambda})$ generating a public verification key pk and a private signing key sk on input of security parameter λ, signing algorithm $\sigma \leftarrow Sig.Sign(sk; m)$ generating a signature for message m, and verification algorithm $Sig.Verify(pk; m, \sigma)$ returning 1 if σ is a valid signature for m under key pk, and 0 otherwise.

not exist any negative result on its security, and the attack [24] on multilinear maps is not applicable to it.

Correctness. For all $\lambda \in \mathbb{N}$, $(pk, sk) \leftarrow Sig.Gen(1^\lambda)$, message $m \in \mathcal{M}(\lambda)$, we require that $Sig.Verify(pk; m, Sig.Sign(sk; m))=1$.

Security. Consider the following security experiment (defined by [30]) played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

1. The challenger generates a public/private key pair $(pk, sk) \leftarrow Sig.Gen(1^\lambda)$, the adversary receives pk as input.
2. The adversary may query arbitrary messages m_i to the challenger. The challenger replies to each query with a signature $\sigma_i = Sig.Sign(sk; m_i)$. Here i is an index, ranging between $1 \leq i \leq q$ for some $q \in \mathbb{N}$. Queries can be made adaptively.
3. Eventually, the adversary outputs a message/signature pair (m^*, σ^*).

Definition 5 (Secure Signatures [30]). We say that SIG is *existentially unforgeable under adaptive chosen-message attacks* (EUF-CMA), if for all adversaries $\mathcal{A}$, there exists a negligible function negl such that

$$\Pr[(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{C}}(1^\lambda, pk) \quad such \quad that \\ Sig.Verify(pk; \sigma^*, m^*) = 1 \wedge m^* \notin \{m_1, \ldots, m_q\}] \leq \mathsf{negl}(\lambda).$$

3 Security Model

In this section, we briefly recall the formal security model for group key exchange protocols as presented in [10] (which is based on the model by Bresson [9]).

Parties and initialization. In a group key exchange protocol, we assume for simplicity a fixed, polynomial-size set $\mathcal{P} = \{P_1, \ldots, P_l\}$ of potential parties. Any subset of $\mathcal{P}$ may decide at any point to establish a session key, and we do not assume that these subsets are always the same size or always include the same participants. There are two different types of party: $\mathcal{CP}$ (certification based party) and $\mathcal{IP}$ (identity based party). Before the protocol is run for the first time, an initialization phase occurs. For each participant $P_i \in \mathcal{CP}$, it runs an algorithm $\mathcal{G}_i(1^\lambda)$ to generate public/private keys (PK_i, SK_i), where each P_i may be from different cryptographic settings (e.g., finite field, elliptic curve, or RSA). For each $P_i \in \mathcal{IP}$, the public key PK_i is its own identity ID_i and the private key SK_i is generated by its private key generator (PKG). Each player P_i stores SK_i, and the public key PK_i is known by all participants (and is also known by the adversary).

Adversary model. We denote instance i of user P as π_P^i. A given instance may be used only once. Each instance π_P^i has associated with it the variables $\mathtt{acc}_P^i$, $\mathtt{sid}_P^i$, $\mathtt{pid}_P^i$, $\mathtt{sk}_P^i$ with the following semantics:

– $\mathtt{acc}_P^i$: 0/1-valued variable which is set to be 1 by π_P^i upon normal termination of the session and 0 otherwise.
– $\mathtt{sid}_P^i$: session identity for instance π_P^i, which is a protocol-specified function of all communication sent and received by π_P^i.

- $\mathtt{pid}_P^i$: partner identity for instance π_P^i, which consists of the identities of the players in the group with whom π_P^i intends to establish a session key (including P itself).
- $\mathtt{sk}_P^i$: session key after the execution of the protocol by π_P^i.

During the execution of the protocol, an adversary $\mathcal{A}$ could interact with protocol participants via several oracle queries, which model adversary's possible attacks in the real execution. All possible oracle queries are listed in the following:

- $\mathsf{Send}(\pi_P^i, m)$: This query is used to simulate active attacks, in which the adversary may tamper with the message being sent over the public channel. It returns the message that the user instance π_P^i would generate upon receipt of message m.
- $\mathsf{Execute}(\pi_{P_1}^{i_1}, \ldots, \pi_{P_n}^{i_n})$: This query models passive attacks in which the attacker eavesdrops on honest executions among the user instances $\pi_{P_1}^{i_1}, \ldots, \pi_{P_n}^{i_n}$. It returns the messages that were exchanged during an honest execution of the protocol.
- $\mathsf{Reveal}(\pi_P^i)$: This query models the possibility that an adversary gets the session key. It returns to the adversary the session key $\mathtt{sk}_P^i$ of the user instance π_P^i.
- $\mathsf{Corrupt}(P)$: This query returns the long-term secret key of player P.
- $\mathsf{Test}(\pi_P^i)$: This query tries to capture the adversary's ability to tell apart a real session key from a random one. It returns the session key for instance π_P^i if $b = 1$ or a random number of the same size if $b = 0$. This query is called only once.

Partnering. Two instances π_P^i and $\pi_{P'}^j$ are said to be partnered if and only if (1) $\mathtt{pid}_P^i$=$\mathtt{pid}_{P'}^j$, (2) $\mathtt{sid}_P^i$=$\mathtt{sid}_{P'}^j$ and (3) $\mathtt{acc}_P^i = \mathtt{acc}_{P'}^j = 1$.

Freshness. We say an instance π_P^i is *fresh* if none of the following conditions hold:

(1) the adversary queries $\mathsf{Reveal}(\pi_P^i)$ or $\mathsf{Reveal}(\pi_{P'}^j)$, where $\pi_{P'}^j$ is partnered with π_P^i;
(2) the adversary queries $\mathsf{Corrupt}(V)$ (with $V \in \mathtt{pid}_P^i$) before a query of the form $\mathsf{Send}(\pi_{P'}^j, *)$, where $P' \in \mathtt{pid}_P^i$.

Correctness. The correctness of group key exchange protocol requires that, whenever two instances π_P^i and $\pi_{P'}^j$ are partnered, both instances should hold the same non-null session key.

Security. For any adversary $\mathcal{A}$, let $Succ(\mathcal{A})$ be the event that $\mathcal{A}$ makes a single Test query directed to some fresh instance π_P^i at the end of a protocol Π and correctly guesses the bit b used in the Test query. The advantage of $\mathcal{A}$ in violating the semantic security of the protocol Π is defined as:

$$\mathsf{Adv}_{\Pi}(\mathcal{A}) = |2\Pr[Succ(\mathcal{A})] - 1|.$$

Definition 6. We say a group key exchange protocol Π is selectively secure if, for any PPT adversary $\mathcal{A}$ satisfying the following properties, $\mathsf{Adv}_{\Pi}(\mathcal{A})$ is negligible:

– $\mathcal{A}$ commits to a set $\hat{S}$ of users at the beginning of the security game.
– Test query must be on a subset S of $\hat{S}$.

4 One-Round Cross-Domain Group Key Exchange Protocol

In this section we present our construction of a one-round cross-domain group key exchange protocol.

4.1 Protocol Description

The idea of our cross-domain group key exchange (CDGKE) protocol is the following: In the setup phase, a trusted third party chooses a key K for a constrained pseudorandom function PRF and publishes an obfuscated program for the PRF as the global agreed domain parameter. In the group key exchange phase, each participant P_i broadcasts a signature σ_i of the random x_i generated by P_i using his own signature scheme. The shared session key will be the function PRF evaluated at the concatenation of the identity P_i and x_i. However, to make the session key shared only among legal participants, the knowledge of a seed s will be required to operate an obfuscated program for PRF. More precisely, each participant generates a seed s_i and computes $x_i = PRG(s_i)$, where PRG is a pseudorandom generator. In this way, all users can compute the session key, but anyone else without the corresponding private key or seed, will therefore be unable to compute the session key.

A formal description of our protocol appears in Fig. 1.

4.2 Correctness and Security

The correctness is obvious by inspection. For security, we have the following theorem.

Theorem 1. Let $PRG : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda}$ be a secure pseudorandom generator, let F be a secure constrained PRF, let SIG_i ($i \in \{1, 2, \cdots, n\}$) be a signature scheme that is existentially unforgeable under adaptive chosen-message attacks, and let $i\mathcal{O}$ be a secure indistinguishability obfuscator. Then, the protocol in Fig. 1 is a secure group key exchange protocol.

Proof. Fix a PPT adversary $\mathcal{A}$ attacking the cross-domain group key exchange protocol. We use a hybrid argument to bound the advantage of $\mathcal{A}$. We define a sequence of experiments $\mathbf{Hyb}_0, \cdots, \mathbf{Hyb}_3$, and denote the advantage of adversary $\mathcal{A}$ in experiment $\mathbf{Hyb}_i$ as:

$$\mathrm{Adv}_i(\mathcal{A}) \stackrel{\mathrm{def}}{=} |2 \cdot \Pr[\mathcal{A} \text{ succeeds in } \mathbf{Hyb}_i] - 1|.$$

Protocol

Consider an execution of the protocol among participants $P_1, \cdots, P_n$ belonging to different security domains and wishing to establish a common session key. Let F be a pseudorandom function, PRG be a pseudorandom generator, and $i\mathcal{O}$ be a program indistinguishability obfuscator.

Global Agreed Domain Parameter: A trusted third party chooses a random key K to obtain an instance of a pseudorandom function F, builds the program P_{CDGKE} in Fig. 2, and then outputs $P_{i\mathcal{O}} = i\mathcal{O}(P_{\text{CDGKE}})$ as the global agreed domain parameter.

Setup: Build the global agreed domain parameter and publish it. Each participant P_i chooses his own signature scheme SIG_i= ($SIG_i.Gen$, $SIG_i.Sign$, $SIG_i.Verify$), runs the key generation algorithm $SIG_i.Gen$ on input $1^\lambda$to obtain a public/private key pair (pk_i, sk_i) (i.e., $(pk_i, sk_i) \leftarrow SIG_i.Gen(1^\lambda)$), where $\lambda \in \mathbb{N}$ is a security parameter.

Round 1: Each participant P_i proceeds as:

1. Choose s_i randomly, compute $x_i = PRG(s_i)$, and generate the signature $\sigma_i = SIG_i.Sign(sk_i; x_i||P_1|| \cdots ||P_n)$.
2. Broadcast $m_i = (P_i, S_i, pk_i, x_i, \sigma_i)$.

Key Generation: Each participant P_i runs $P_{i\mathcal{O}}$ on $(m_1, m_2, \cdots, m_n, i, s_i)$ to obtain the session key SK or $\perp$.

Fig. 1. An honest execution of the cross-domain group key exchange protocol

Inputs: $m_1, m_2, \cdots, m_n$, i, s_i
Constants: F key K
if $x_i \neq PRG(s_i)$ **then**
  Output $\perp$
else if there exists $j \leq n$ such that $SIG_j.Verify(pk_j; x_j||P_1|| \cdots ||P_n, \sigma_j) = 0$ **then**
  Output $\perp$
else Output $F(K, x_1, x_2, \ldots, x_n, P_1, P_2, \ldots, P_n)$

Fig. 2. The program P_{CDGKE}

We bound the difference between the adversary's advantage in successive experiments, and then bound the adversary's advantage in the final experiment. Finally, combining all the above results, we get the desired bound on $\mathrm{Adv}_0(\mathcal{A})$, the adversary's advantage when attacking the real protocol.

Experiment $\mathbf{Hyb}_0$. This is the original experiment with respect to a given polynomial-time adversary $\mathcal{A}$, in which $\mathcal{A}$ commits to a set $\hat{S} = \{\hat{P}_1, \hat{P}_2, \cdots, \hat{P}_n\}$ and interacts with the real protocol as defined in Sect. 3.

Experiment $\mathbf{Hyb}_1$. This experiment is different from $\mathbf{Hyb}_0$ only in that it is aborted and the adversary does not succeed if the following event **Forge** occurs.

Forge: Let **Forge** be the event that, the adversary makes send query of the form $\mathsf{Send}(\pi_P^i, m)$ such that the message m contains a new, valid message/signature pair with respect to the public key pk_U of some user U before querying Corrupt(U).

Lemma 1. $|Adv_1(\mathcal{A}) - Adv_0(\mathcal{A})| < negl(\lambda)$.

Proof. Assuming that the event **Forge** occurs, we can construct an algorithm $\mathcal{F}$ which outputs, with a non-negligible probability, a forgery against a signature scheme SIG_i for some $i \in \{1, 2, \cdots, n\}$ as follows.

The given public key PK is assigned to one of the n participants. All other parties are initialized as normal according to the protocol. All queries to the parties can be easily answered by following the protocol specification since all secret keys are known, except for the private key corresponding to the public key of the forgery attack game. In the latter case the signing oracle that is available as part of the chosen message attack can be used to simulate the answers. If **Forge** occurs against an instance who holds PK, $\mathcal{F}$ halts and outputs the message/signature pair generated by $\mathcal{A}$ as its forgery. Otherwise, $\mathcal{F}$ halts and outputs a failure indication.

The success probability of $\mathcal{F}$ is exactly $\Pr[\mathbf{Forge}]/n$. Then, the lemma follows by noticing that the signature scheme SIG_i ($i \in \{1, 2, \cdots, n\}$) is existentially unforgeable under adaptive chosen-message attacks.

Experiment Hyb$_2$. In this experiment, for $P_i \in \hat{S}$, we will choose random $x_i \in \{0,1\}^{2\lambda}$ instead of generating them from PRG. The security of PRG yields the lemma 2.

Lemma 2. $|Adv_2(\mathcal{A}) - Adv_1(\mathcal{A})| < negl(\lambda)$.

Experiment Hyb$_3$. Replace the $F(\cdot)$ in P_{CDGKE} by a constrained pseudorandom function $F^C(\cdot)$, arriving at the program P'_{CDGKE} given in Fig. 3. The constrained set C is defined as $C = \{(x_1, x_2, \ldots, x_n, P_1, P_2, \ldots, P_n) :$ there exists some P_j (and respective x_j) that is not contained in the set $\hat{S}\}$.

Inputs: $m_1, m_2, \cdots, m_n$, i, s_i
Constants: Constrained F key K_C
if $x_i \neq PRG(s_i)$ **then**
  Output $\perp$
else if there exists $j \leq n$ such that $SIG_j.Verify(pk_j; x_j||P_1|| \cdots ||P_n, \sigma_j) = 0$ **then**
  Output $\perp$
else Output $F^C(K_C, x_1, x_2, \ldots, x_n, P_1, P_2, \ldots, P_n)$

Fig. 3. The program P'_{CDGKE}

Lemma 3. $|Adv_3(\mathcal{A}) - Adv_2(\mathcal{A})| < negl(\lambda)$.

Proof. Note that with overwhelming probability, none of x_i (the corresponding $P_i \in \hat{S}$) in Experiment Hyb_2 has a pre-image under PRG. Therefore, with overwhelming probability, there is no input to P_{CDGKE} that will cause F to be evaluated on points of the form $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n, \hat{P}_1, \hat{P}_2, \ldots, \hat{P}_n)$, where $\hat{P}_i \in \hat{S}$. We can conclude that the modified program P'_{CDGKE} has the same functionality with the original program P_{CDGKE}. Then based on the property of indistinguishability obfuscation, it is easy to see that the experiments Hyb_2 and Hyb_3 are computationally indistinguishable. Thus, security of $i\mathcal{O}$ yields the lemma.

Bounding the advantage in Hyb_3. We reduce the non-negligible advantage of the adversary $\mathcal{A}$ in the experiment $\mathbf{Hyb}_3$ to the security of the constrained PRF presented above. We construct a PRF adversary $\mathcal{B}$ that breaks the security of F as a constrained PRF as follows: adversary $\mathcal{B}$ simulates the entire experiment for $\mathcal{A}$. In response to Execute query, $\mathcal{B}$ computes the signature of m_i with correct private key sk_i exactly as in experiment Hyb_3. In response to Reveal query, $\mathcal{B}$ also queries its PRF oracle and thus always reveals the correct session key. At the end of the experiment, for a test query, $\mathcal{B}$ makes a real-or-random challenge query for the constrained function F^C as defined above. One can easily see that, $\mathcal{B}$ is given a real PRF or a random value, then its simulation is performed exactly as in experiment Hyb_3. Thus, the advantage of $\mathcal{B}$ is exactly $\text{Adv}_3(\mathcal{A})$. It conflicts with the security of the constrained PRF. Thus the advantage of the adversary $\mathcal{A}$ in this experiment is negligible.

4.3 Comparison with Related Protocols

The core of our protocol is an obfuscation program, therefore, any polynomial-time bounded indistinguishability obfuscation candidates (e.g., [26,27]) can be adopted to instantiate our scheme. In this subsection, we compare our protocol with Katz *et al.*'s protocol [10], Neupane *et al.*'s protcol [12], Ustaoğlu's protocol [18], and Guo *et al.*'s protocol [19] from many respects. Table 1 summarizes the comparison results[2].

In Table 1, both Katz *et al.*'s protocol [10] and Neupane *et al.*'s protocol [12] are group key exchange protocols proven to be secure in the standard model. However, their constructions require all participants to adopt a common cryptographic setting and shared parameters, which means that cross-domain interaction is not supported. Both Ustaoğlu's protocol [18] and Guo *et al.*'s protocol [19] are two-party key exchange protocols supporting cross-domain interaction. However, as the authors commented, the protocol in [18] requires the participants to use parameters from the same algebraic group and the protocol in [19] requires one party being identity-based and the other one being certificate-based, which means that the involved cryptographic setting is not universal. Meanwhile,

[2] Since our protocol is universal, the concrete computation & communication complexity relies on the instantiated schemes, and we omit it in the comparison.

Table 1. Comparison of related protocols

Protocol	Type	Communication rounds	Cross-domain support?	Universal?	Standard model?
Protocol in [10]	Group	3	✗	✗	✓
Protocol in [12]	Group	2	✗	✗	✓
Protocol in [18]	2-Party	2	✓	✗	✗
Protocol in [19]	2-Party	3	✓	✗	✗
Our protocol	Group	1	✓	✓	✓

our protocol is a group key exchange protocol supporting cross-domain interaction. Moreover, the participants may come from various cryptographic settings (universal) and do not need anything special to generate the shared session key.

In summary, our protocol only has one round, and supports cross-domain interaction from different cryptographic settings, and it is proven secure in the standard model. To the best of our knowledge, there is no cross-domain group key exchange protocol (until this work) whose security directly relies on standard model and does not need to use the same algebraic setting and shared parameters.

5 Conclusion

In this paper, we investigate cross-domain group key exchange protocol for interoperability scenarios. Our main contribution is to propose the first one-round group key exchange protocol which supports participants coming from different domains. Besides, different signature schemes and different system parameters can be used, which is more flexible and more suitable for interoperability scenarios. We also prove that our protocol can achieve the desired security goals in the standard model. It remains an open problem to further reduce the computational costs of group participants, whilst maintaining its optimal communication round.

Acknowledgments. We want to thank the anonymous reviewers for their comments which helped to improve the paper. This work was supported by the National Grand Fundamental Research (973) Program of China under Grant 2013CB338003, and the National Natural Science Foundation of China (NSFC) under Grants U1536205 and 61572485.

References

1. Ingemarsson, I., Tang, D.T., Wong, C.K.: A conference key distribution system. IEEE Trans. Inf. Theory **28**(5), 714–720 (1982)
2. Koyama, K., Ohta, K.: Identity-based conference key distribution systems. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 175–184. Springer, Heidelberg (1988). doi:10.1007/3-540-48184-2_13
3. Steer, D.G., Strawczynski, L., Diffie, W., Wiener, M.: A secure audio teleconference system. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 520–528. Springer, New York (1990). doi:10.1007/0-387-34799-2_37
4. Burmester, M., Desmedt, Y.: A secure and efficient conference key distribution system. In: Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 275–286. Springer, Heidelberg (1995). doi:10.1007/BFb0053443
5. Saeednia, S., Safavi-Naini, R.: Efficient identity-based conference key distribution protocols. In: Boyd, C., Dawson, E. (eds.) ACISP 1998. LNCS, vol. 1438, pp. 320–331. Springer, Heidelberg (1998). doi:10.1007/BFb0053744
6. Tzeng, W.-G., Tzeng, Z.-J.: Round-efficient conference key agreement protocols with provable security. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 614–627. Springer, Heidelberg (2000). doi:10.1007/3-540-44448-3_47
7. Bresson, E., Chevassut, O., Pointcheval, D., Quisquater, J.J.: Provably authenticated group diffie-hellman key exchange. In: CCS 2001, pp. 255–264. ACM (2001)
8. Bresson, E., Chevassut, O., Pointcheval, D.: Provably authenticated group diffie-hellman key exchange — the dynamic case. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 290–309. Springer, Heidelberg (2001). doi:10.1007/3-540-45682-1_18
9. Bresson, E., Chevassut, O., Pointcheval, D.: Dynamic group diffie-hellman key exchange under standard assumptions. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 321–336. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_21
10. Katz, J., Yung, M.: Scalable protocols for authenticated group key exchange. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 110–125. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45146-4_7
11. Burmester, M., Desmedt, Y.: A secure and scalable group key exchange system. Inf. Process Lett. (IPL) **94**(3), 137–143 (2005)
12. Neupane, K., Steinwandt, R.: Communication-efficient 2-round group key establishment from pairings. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 65–76. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19074-2_5
13. Arifi, M., Gardeshi, M., Farash, M.S.: A new efficient authenticated id-based group key agreement protocol. Cryptology ePrint Archive: Report 2012/395 (2012)
14. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994). doi:10.1007/3-540-48329-2_21
15. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 480–499. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44371-2_27
16. Chen, L., Kudla, C.: Identity based authenticated key agreement protocols from pairings. In: CSFW 2003, pp. 219–233. IEEE Computer Society (2003)
17. McCullagh, N., Barreto, P.S.L.M.: A new two-party identity-based authenticated key agreement. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 262–274. Springer, Heidelberg (2005). doi:10.1007/978-3-540-30574-3_18

18. Ustaoğlu, B.: Integrating identity-based and certificate-based authenticated key exchange protocols. Int. J. Inf. Secur. **10**(4), 201–212 (2011)
19. Guo, Y., Zhang, Z.: Authenticated key exchange with entities from different settings and varied groups. In: Takagi, T., Wang, G., Qin, Z., Jiang, S., Yu, Y. (eds.) ProvSec 2012. LNCS, vol. 7496, pp. 276–287. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33272-2_18
20. Chen, L., Lim, H.W., Yang, G.: Cross-domain password-based authenticated key exchange revisited. ACM Trans. Inf. Syst. Secur. (TISSEC) **15:16**(4), 1–15:32 (2014)
21. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im) possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)
22. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS 2013, pp. 40–49. IEEE (2013)
23. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38348-9_1
24. Hu, Y., Jia, H.: Cryptanalysis of GGH map. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 537–565. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49890-3_21
25. Miles, E., Sahai, A., Zhandry, M.: Annihilation attacks for multilinear maps: cryptanalysis of indistinguishability obfuscation over GGH13. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 629–658. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53008-5_22
26. Garg, S., Mukherjee, P., Srinivasan, A.: Obfuscation without the vulnerabilities of multilinear maps. Cryptology ePrint Archive: Report 2016/390 (2016)
27. Lin, H., Vaikuntanathan, V.: Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In: FOCS 2016, pp. 11–20. IEEE (2016)
28. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM (JACM) **33**(4), 792–807 (1986)
29. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 280–300. Springer, Heidelberg (2013). doi:10.1007/978-3-642-42045-0_15
30. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988)