Information Sciences 210 (2012) 67–80

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Provably secure threshold public-key encryption with adaptive security and short ciphertexts

Bo Qin [a,c], Qianhong Wu [a,b,*], Lei Zhang [d], Oriol Farràs [a], Josep Domingo-Ferrer [a]

[a] Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
[b] Key Lab. of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Computer, Wuhan University, China
[c] Department of Maths, School of Science, Xi'an University of Technology, China
[d] Shanghai Key Laboratory of Trustworthy Computing, Software Engineer Institute, East China Normal University, Shanghai, China

ARTICLE INFO

Article history:
Received 14 February 2011
Received in revised form 19 March 2012
Accepted 25 April 2012
Available online 5 May 2012

Keywords:
Public-key cryptosystem
Threshold public-key encryption
Adaptive security
Access control

ABSTRACT

Threshold public-key encryption is a cryptographic primitive allowing decryption control in group-oriented encryption applications. Existing TPKE schemes suffer from long ciphertexts with size linear in the number of authorized users or can only achieve non-adaptive security, which is too weak to capture the capacity of the attackers in the real world. In this paper, we propose an efficient TPKE scheme with constant-size ciphertexts and adaptive security. Security is proven under the decision Bilinear Diffie–Hellman Exponentiation assumption in the standard model. Then we extend our basic construction with efficient trade-offs between the key size and the ciphertext size. Finally, we illustrate improvements to transmit multiple secret session keys in one session with almost no extra cost.

1. Introduction

In many private data escrow applications, one cannot fully trust a single person to access the escrowed information but possibly one can trust a group of individuals. A typical application is an electronic auction in which a set of bodies are trusted to publish the final outcome, but not to disclose any individual bid. A similar application occurs in electronic voting systems. In this case, the trusted bodies publish the tally, but they do not disclose ballots to any individual. Other scenarios include key-escrow and decryption procedures requiring an agreement of a number of trusted bodies. Further, to provide robustness, the escrowed sensitive information should be accessible in case of emergency even if some individuals in the trusted group are unavailable. *Threshold public-key encryption* (TPKE) is an important cryptographic primitive [10,15,19,28] to provide solutions to such applications. In TPKE, each of the n users holds a decryption key corresponding to a public key; one can encrypt a message for an authorized subset of the users; the ciphertext can be decrypted only if at least t users in the authorized set cooperate. Below this threshold, no information about the message is leaked, even if $t-1$ authorized users and all the users outside the authorized set collude.

Current TPKE schemes either achieve only non-adaptive security or they suffer from long ciphertexts of size linear in the number of authorized users. In the non-adaptive security notion, it is assumed that the attacker decides the set of users whom she will attack before the system is initialized. Clearly, this notion is too weak to capture the capacity of the attacker in the real world. In practice, it is more likely for an attacker to corrupt users after the system is deployed and the corruption may be adaptive in the sense that the attacker may bribe the most valuable users based on the previous corruptions and the

* Corresponding author at: Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain. Tel.: +34 977558270; fax: +34 977559710.

E-mail addresses: bo.qin@urv.cat (B. Qin), qianhong.wu@urv.cat (Q. Wu), lei.zhang@urv.cat (L. Zhang), oriol.farras@urv.cat (O. Farràs), josep.domingo@urv.cat (J. Domingo-Ferrer).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.04.038

observation of the system operation, and then decide to attack the set of target users. As to performance, the linear-size ciphertexts are an obstacle for applications with a potential large number of users, *e.g.*, access control to sensitive databases in a distributed environment. These limitations of existing TPKE schemes motivate the work in this paper.

1.1. Our contributions

In this paper, we investigate TPKE systems with adaptive security and short ciphertexts, which are essential features for TPKE schemes to be securely and efficiently deployable in practice. In particular, our contribution can be summarized as follows.

We first present a modular design/proof paradigm to build TPKE systems with adaptive security. To this end, we introduce a useful security notion referred to as *semi-adaptive* security in TPKE systems and present a generic transformation from a semi-adaptively secure TPKE scheme to an adaptively secure scheme. In the semi-adaptive security notion, the attacker commits to a set of users before the system is set up. The attacker can adaptively query the decryption keys of users outside the committed set and at most $t-1$ queries for the decryption keys of users in the committed set. Then the attacker can choose a target group which is a subset of the committed set for the challenge ciphertext. Clearly, a semi-adaptive attacker is weaker than an adaptive attacker, but it is stronger than a non-adaptive attacker because the subset of users in the committed set to be attacked can be chosen adaptively by the attacker. By using the similar idea in [20], we bridge semi-adaptive security and adaptive security with a generic conversion from any semi-adaptively secure TPKE scheme to an adaptively secure one. The only cost is doubling the ciphertext of the underlying semi-adaptively secure TPKE scheme.

Following the new paradigm, we propose a TPKE scheme with constant-size ciphertext and semi-adaptive security. Compared to the previous version of this work [29], we strictly prove the semi-adaptive security of the proposed TPKE scheme under the decision *Bilinear Diffie–Hellman Exponentiation* (BDHE) assumption in the standard model (*i.e.*, without using random oracles). Then by applying the proposed generic transformation, we obtain an adaptively secure TPKE scheme with short ciphertexts. Our scheme allows users to join the system at any point. The sender can encrypt to a dynamically authorized set and the encryption validity is publicly verifiable. Our scheme also enjoys non-interactive decryption and the reconstruction of the message is very efficient. These features seem desirable for applications of TPKE systems.

Finally, we suggest several variants to achieve better efficiency for implementation in different scenarios. We provide an efficient trade-off between ciphertext and key size, which yields the first TPKE scheme with adaptive security and sublinear-size public/decryption keys and ciphertexts. Compared to the previous version of this work [29], a new extension enables multiple session keys to be transmitted in a single session with almost no extra cost. Note that session key transmission is the costliest operation in the motivated applications. This multiple session key transmission variant is very useful in practice.

1.2. Related work

For the purpose of controllable decryption, a number of notions have been proposed, such as threshold public-key encryption [15], identity-based threshold encryption/decryption [2,26], threshold public-key cryptosystem [10,37], threshold signcryption [39], threshold broadcast encryption [13], dynamic threshold encryption [19,28], and *ad hoc* threshold encryption [14]. Unlike regular public-key cryptosystems, the common spirit of these notions is that decryption should be controllable to some extent. That is, for a user to decrypt a ciphertext, the user must be in the authorized set and must cooperate with a certain number of users in the same authorized set which is determined by the encrypter when encrypting the message. There are also slight differences among these notions such as how to determine the threshold and whether it is changeable for different encryption operations, whether a trusted party is employed to set up and maintain the system, whether each user has an explicit public key and, if the user has any public key, whether it is a randomly generated string (like the public key in a regular public-key cryptosystem) or some recognizable information (like a user's identity in an identity-based cryptosystem) or a public key in certificateless cryptography [38]. Among these notions, the most common one is threshold public-key encryption in which a trusted party sets up the system with a threshold as a system parameter and allows users to join the system by generating a decryption key for each of them; a sender can encrypt to a number (no less than the threshold) of authorized users chosen from the set of registered users, the ciphertext can be decrypted by subsets of users in the authorized set whose size is greater than or equal to the threshold. This notion enables the trusted party and the sender to jointly decide how a message is disclosed.

Although the notion of TPKE is conceptually clear and well-studied, its practical deployment and its security have not yet been well addressed. The scheme due to Daza et al. [13] appears to be the first one that has ciphertext of length less than $\mathcal{O}(|\mathbb{R}|)$ (*i.e.*, $\mathcal{O}(|\mathbb{R}|-t)$), where $|\mathbb{R}|$ is the size of the authorized set. Note that t in practice is usually very small, while $|\mathbb{R}|$ might be very large, up to n as the maximal number of the authorized users. The scheme has indeed linear-size ciphertext regarding the receiver scale n. Recently, a scheme with constant-size ciphertext was presented by Delerablée and Pointcheval in [15]. However, as mentioned by the authors [15], their scheme has several limitations. Their proposal has an $\mathcal{O}(n)$-size public key and only achieves non-adaptive security which, as explained above, is too weak to capture the capacity of attackers in the real world. Also, the security of their scheme relies on a new assumption. Indeed, their focus is to achieve dynamic TPKE allowing short ciphertext and a threshold to be decided by the encrypter at each encryption time; they leave it as an open problem to design a scheme with short ciphertext and adaptive security. Subsequent to but independent of our original work [29], Libert and Yung [27] presented a TPKE scheme with similar properties and slightly shorter private keys.

Several notions close to TPKE have been proposed in the literature. By setting the threshold to 1, a TPKE scheme becomes indeed a broadcast encryption scheme [1,17]. In this scenario, a trusted dealer generates and privately distributes decryption keys to n users; a sender can send a message to a dynamically chosen subset of receivers $\mathbb{R} \subseteq \{1, \ldots, n\}$ such that only the users in $\mathbb{R}$ can decrypt the ciphertext. Fiat and Naor [17] were the first to formally explore broadcast encryption. Further improvements [21,23] reduced the decryption key size. Dodis and Fazio [16] extended the subtree difference method into a public-key broadcast system for a small-size public key. Wallner et al. [33] and Wong [34] independently discovered the logical-tree-hierarchy scheme for group multicast. The parameters of the original schemes were improved in further work [9,11,31]. Boneh et al. [6] proposed two efficient broadcast encryption schemes proven to be secure. Their basic scheme has linear-size public keys but constant-size secret keys and ciphertexts. After a trade-off, they obtained a scheme with $\mathcal{O}(\sqrt{n})$-size public keys, decryption keys, and ciphertexts. However, similarly to [15], they used a non-adaptive model of security. Other contributions [7,8,20] focused on stronger adaptive security in the sense that the attacker can adaptively corrupt users, as considered in this paper. Attribute-based encryption [3] is also related to the threshold decryption capability in TPKE systems, according to the number of common attributes owned by the recipient. Ciphertext-policy based encryption [18] can be viewed as a generalization of all the above notions, since it allows the encrypter to specify a decryption policy and only receivers meeting the policy can decrypt. However, no joint computation is required/possible for decryption. This is different from the usual notion of threshold cryptography, where a pool of players are required to cooperate to accomplish the decryption operation.

1.3. Paper organization

The rest of the paper is organized as follows. In Section 2, we review the definition of TPKE systems and present a generic conversion from a TPKE with semi-adaptive security to one with adaptive security. Section 3 proposes a basic secure TPKE scheme with small ciphertexts and semi-adaptive security. Variants are suggested in Section 4 with fully adaptive security and sublinear-size public/decryption keys and ciphertexts. Section 4.4 presents improvements to transmit multiple session keys in a single session, followed by a conclusion in Section 5.

2. Modeling public-key encryption systems

We review the model of TPKE systems and then formalize the security definitions in TPKE schemes motivated by Delerablée and Pointcheval [15]. We focus on standard TPKE systems where the threshold is determined by a trusted party that we call the dealer in our definition. Compared to the definition in [15], our definition is simplified without requiring public verifiability of the encryption and partial decryption procedures. We argue that, although this public verification might be useful, it can be achieved modularly by employing non-interactive (zero-)knowledge proofs, and for clarity, we do not emphasize this property in the definition of TPKE as an atomic primitive. However, we are interested in providing the stronger adaptive security in TPKE systems, and, to this end, a transitional notion, *i.e.* semi-adaptive security, is defined.

2.1. Definition of TPKE systems

We begin by formally defining a TPKE system. Note that, for content distribution or any encryption of a long message, the current standard technique is the KEM-DEM methodology [32], where a secret session key is generated and distributed with public-key encryption, and then used with an appropriate symmetric cryptosystem to encrypt the content. Hence, for clarity, we define TPKE as a key encapsulation mechanism. A TPKE system consists of the following polynomial-time algorithms:

Setup(1^λ). This algorithm is run by a trusted dealer to set up the system. It takes as input a security parameter λ and it outputs the global system parameters; the latter include n (the maximal size of a TPKE authorized receiver set) and t (the threshold number of cooperating receivers for decryption). We denote the system parameters by π, which is a common input to all the following procedures. However, we explicitly mention π only in the KeyGen procedure; in the other procedures it is omitted for simplicity.

KeyGen(π). This key generation algorithm is run by the dealer to generate the master public/secret key pair for the TPKE system. It takes as input the system parameter π and it outputs $\langle MPK, msk \rangle$ as the master public/secret key pair. *MPK* is published and *msk* is kept secret by the dealer.

Join(msk, ID). This algorithm is run by a dealer to generate a decryption key for a user with identity *ID*. It takes as input the master secret key *msk* and the identity *ID* of a new user who wants to join the system. It outputs the user's keys (UPK, udk), consisting of the user's public key *UPK* for encryption, and the user's decryption key *udk* for decryption. The decryption key *udk* is privately given to the user, whereas *UPK* is widely distributed, with an authentic link to *ID*.

Encrypt($MPK, \mathbb{R}$). This algorithm is run by a sender to distribute a session key to the chosen users so that they can recover the session key only if at least t of them cooperate. It takes as input a recipient set $\mathbb{R} \subseteq \{1, \ldots, n\}$ consisting of the identities (or the public keys) of the chosen users, and the TPKE master public key *MPK*. If $|\mathbb{R}| \leqslant n$, it outputs a pair $\langle Hdr, sk \rangle$, where *Hdr* is called the header of the session key *sk*. Then $\langle Hdr, \mathbb{R} \rangle$ is sent to users in $\mathbb{R}$.

ShareDecrypt$(\mathbb{R}, ID, udk, Hdr, PK)$. This algorithm allows each user in the receiver set to decrypt a share of the secret session key *sk* hidden in the header. It takes as input the receiver set $\mathbb{R}$, an authorized user's identity *ID*, the authorized user's decryption key *udk*, and a header *Hdr*. If the authorized user's identity *ID* lies in the authorized set $\mathbb{R}$ and $|\mathbb{R}| \leqslant n$, then the algorithm outputs a share σ of the secret session key *sk*.

Combine$(MPK, \mathbb{R}, \mathbb{S}, Hdr, \Sigma)$. It takes as input the master public key *MPK*, the authorized receiver set $\mathbb{R}$, a subset $\mathbb{S} \subseteq \mathbb{R}$ of *t* authorized users, and a list $\Sigma = (\sigma_1, \ldots, \sigma_t)$ of *t* decrypted session key shares. It outputs the session key *sk* or $\perp$ representing an error in reconstructing the session key.

2.2. Security definitions

We first define the correctness of a TPKE scheme. It states that any *t* users in the authorized receiver set can decrypt a valid header. Formally, it is defined as follows.

Definition 1 (*Correctness*). A TPKE scheme is said to be correct if for all $\mathbb{R}(t \leqslant |\mathbb{R}| \leqslant n)$, all $\mathbb{A} \subseteq \mathbb{R}(|\mathbb{A}| \geqslant t)$, $\pi \leftarrow \mathsf{Setup}(1^\lambda)$, $(MPK, msk) \leftarrow \mathsf{KeyGen}(\pi)$, $(UPK, udk) \leftarrow \mathsf{Join}(msk, ID)$ for all identities $ID, \langle Hdr, sk\rangle \leftarrow \mathsf{Encrypt}(MPK, \mathbb{R})$, $\Sigma = \{\sigma | \sigma \leftarrow \mathsf{ShareDecrypt}(\mathbb{R}, ID, udk, Hdr, PK), ID \in \mathbb{A}\}$, then $\mathsf{Combine}(MPK, \mathbb{R}, \mathbb{S}, Hdr, \Sigma) = sk$.

We concentrate on adaptive security against corrupted users. For simplicity, we define security against chosen-plaintext attacks (CPA). However, our definition can readily be extended to capture chosen-ciphertext attacks.

As usual in a TPKE scheme, the attacker is allowed to see all the public data including the system parameters, each user's public key and the master public key. To capture *adaptive* security, the attacker is allowed to adaptively ask for the decryption keys of some users before choosing the set of users that it wishes to attack. Formally, adaptive security in a TPKE scheme is defined using the following game between an attacker $\mathcal{A}$ and a challenger $\mathcal{CH}$. Both $\mathcal{CH}$ and $\mathcal{A}$ are given λ as input.

Setup: The challenger runs $\mathsf{Setup}(1^\lambda)$ to obtain the system parameters. The challenger gives the public system parameters to the attacker.

Corruption: The attacker $\mathcal{A}$ can access the public keys of the dealer and the users. $\mathcal{A}$ can adaptively request the decryption keys of some users.

Challenge: At some point, the attacker specifies a challenge set $\mathbb{R}^*$ with a constraint that the number of corrupted users in $\mathbb{R}^*$ be at most $t - 1$. The challenger sets $\langle Hdr^*, sk_0\rangle \leftarrow \mathsf{Encrypt}(MPK, \mathbb{R}^*)$ and $sk_1 \leftarrow \mathbb{K}$, where $\mathbb{K}$ is the session key space. It sets $b \leftarrow \{0,1\}$ and gives $\langle Hdr^*, sk_b\rangle$ to the attacker $\mathcal{A}$.

Guess: The attacker $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ for *b* and wins the game if $b = b'$.

We define $\mathcal{A}$'s advantage in attacking the TPKE system with security parameter λ as

$$Adv^{\mathrm{TPKE}}_{\mathcal{A},n,t}(1^\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right|$$

Definition 2 (*Adaptive security*). We say that a TPKE scheme is adaptively secure if for every polynomial-time algorithm $\mathcal{A}$ we have that $\mathrm{Adv}^{\mathrm{TPKE}}_{\mathcal{A},n,t}(1^\lambda)$ is negligible in λ.

In addition to the adaptive game for TPKE security, we consider two other weaker security notions. The first is non-adaptive security, where the attacker must commit to the set $\mathbb{R}^* \subseteq \mathbb{R}$ of identities that it will attack in an $\mathsf{Initialization}$ phase before the Setup algorithm is run. This is the security definition that is used by recent TPKE systems [15]. Another useful security definition is referred to as *semi-adaptive* security. In this game the attacker must commit to a set $\overline{\mathbb{R}} \subseteq \mathbb{R}$ of indices at the $\mathsf{Initialization}$ phase before the Setup stage. The attacker can query the decryption key for any user outside $\overline{\mathbb{R}}$. The attacker can also query the decryption keys for up to $t - 1$ users in $\overline{\mathbb{R}}$. It has to choose a target group $\mathbb{R}^* \subseteq \overline{\mathbb{R}}$ for the challenge ciphertext, noting that at most $t - 1$ authorized users have been corrupted. A semi-adaptive attacker is weaker than an adaptive attacker, but it is stronger than a non-adaptive attacker since the attacker can adaptively choose the target users to attack.

2.3. From semi-adaptive security to adaptive security

The adaptive security game may appropriately model the attacker against TPKE systems in the real world. However, it seems hard to achieve adaptive security in TPKE systems, since the simulator does not know which users the attacker will corrupt so that it can prepare secret keys for them. A possible way to overcome the problem is to let the simulator guess the target set before initializing the adaptive security game. However, such a reduction suffers from an exponentially small probability of correctly guessing the target set. Hence, this kind of reduction proofs are not meaningful for a realistic number of users in a TPKE system.

In the sequel, we show how to efficiently convert a TPKE system with semi-adaptive security into one with adaptive security. The cost is doubling ciphertexts. Our conversion is motivated by Gentry and Waters's work [20] which transforms a semi-adaptively secure broadcast scheme into one with adaptive security. This technique is derived from the two-key simulation technique introduced by Katz and Wang [25], which was initially used to obtain tightly secure signature and identity-based encryption schemes in the random oracle model. We observe that this idea can also be employed in the TPKE scenario.

Suppose that we are given a semi-adaptively secure TPKE system $\mathtt{TPKE}_{SA}$ with algorithms $Setup_{SA}$, $KeyGen_{SA}$, $Join_{SA}$, $Encrypt_{SA}$, $ShareDecrypt_{SA}$, $Combine_{SA}$. Then we can build an adaptively secure $\mathtt{TPKE}_{A}$ system as follows:

Setup(1^λ). Run $Setup_{SA}(1^\lambda)$ and obtain π' including parameters t and $2n$. Output π which is the same as π' except that the maximal number of authorized users is n rather than $2n$. This implies that if the underlying $\mathtt{TPKE}_{SA}$ allows up to $2n$ users, then the adaptive scheme allows up to n users.

KeyGen(π). Run $(MPK', msk') \leftarrow KeyGen_{SA}(\pi)$. Randomly choose $\theta \leftarrow \{0,1\}^n$. Set $MPK = MPK'$, $msk = (msk', \theta)$. Output (MPK, msk) as the dealer's master public/secret key pair. Denote the ith bit of θ by θ_i.

Join(msk, ID_i). Run $(UPK'_i, udk'_i) \leftarrow Join_{SA}(msk', ID_{2i-\theta_i})$, where $1 \leqslant i \leqslant n$. Set $UPK = UPK'_i$, $udk_i = (udk'_i, \theta_i)$. Output UPK as the public key of the user ID_i, and udk_i as the user's decryption key.

Encrypt$(MPK, \mathbb{R}, sk)$. Generate a random set of $|\mathbb{R}|$ bits: $\zeta \leftarrow \{\zeta_i \leftarrow \{0,1\} : i \in \{1,\ldots,|\mathbb{R}|\}\}$. Randomly choose $x \leftarrow \mathbb{K}$. Set

$$\mathbb{R}_0 \leftarrow \{ID_{2i-\zeta_i} : i \in \{1,\ldots,|\mathbb{R}|\}\}, \langle Hdr_0, sk\rangle \leftarrow Encrypt_{SA}(MPK, \mathbb{R}_0)$$
$$\mathbb{R}_1 \leftarrow \{ID_{2i-(1-\zeta_i)} : i \in \{1,\ldots,|\mathbb{R}|\}\}, \langle Hdr_1, sk\rangle \leftarrow Encrypt_{SA}(MPK, \mathbb{R}_1)$$

Set $Hdr = \langle Hdr_0, Hdr_1, \zeta\rangle$. Output $\langle Hdr, sk\rangle$. Send $\langle Hdr, \mathbb{R}\rangle$ to the authorized receivers in $\mathbb{R}$.

ShareDecrypt$(\mathbb{R}, ID_i, udk_i, Hdr, MPK)$. Parse udk_i as (udk'_i, θ_i) and Hdr as $\langle Hdr_0, Hdr_1, \zeta\rangle$. Set $\mathbb{R}_0$ and $\mathbb{R}_1$ as above. Run

$$\sigma_i \leftarrow ShareDecrypt_{SA}(\mathbb{R}_{\theta_i \oplus \zeta_i}, ID_{2i-\theta_i} udk'_i, Hdr_{\theta_i \oplus \zeta_i}, PK).$$

Output σ_i. Let the t authorized users be in $\mathbb{S} \subseteq \mathbb{R}$, and w.l.o.g., the corresponding decryption shares be $\Sigma = (\sigma_1, \ldots, \sigma_t)$.

Combine$(MPK, \mathbb{R}, \mathbb{S}, Hdr, \Sigma)$. Run $sk \leftarrow Combine_{SA}(MPK, \mathbb{R}, \mathbb{S}, Hdr, \Sigma)$. Output sk.

Let us look into the above generic conversion. The spirit is that each user is associated with two potential decryption keys; however, the dealer gives only one of the two to the user. An encrypter (who does not know which decryption key the receiver possesses) encrypts the ciphertext twice, one for each key. The main benefit of this idea is that, in the reduction proof, a simulator will have decryption keys for every user, and then it can always correctly answer the corruption queries from the attacker, hence circumventing the need of guessing the target set in advance. This idea is the same used in [20] to achieve an adaptively secure broadcast from a semi-adaptively secure scheme. The only difference lies in that t authorized users are required to cooperate to recover the session key in our setting. It is easy to see that, for a security proof, the two conversions are identical. This is due to the fact that TPKE and broadcast encryption are the same except for the decryption procedure, but the simulator will provide a decryption service to the attacker in either case. Hence, in the context of a TPKE system, the simulator just needs to do the same job as the simulator in a broadcast scheme. There is no difference for the attacker to communicate with the simulator in a broadcast scheme or a TPKE system. Therefore, the security proof of the Gentry–Waters conversion can be trivially extended for the following theorem regarding the above conversion, noting that we do not need the additional symmetric encryption operations in the Gentry–Waters conversion (which are used to guarantee that the same session key can be decrypted by all the authorized users in their system). Hence, the proof of the following theorem is omitted to avoid repetition.

Theorem 3. *Let $\mathcal{A}$ be an adaptive attacker against $\mathtt{TPKE}_A$. Then, there exist some algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$, each running in about the same time as $\mathcal{A}$, such that*

$$Adv^{\mathtt{TPKE}\ A}_{\mathcal{A},n,t}(\lambda) \leqslant Adv^{\mathtt{TPKE}\ SA}_{\mathcal{B}_1,2n,t}(\lambda) + Adv^{\mathtt{TPKE}\ SA}_{\mathcal{B}_2,2n,t}(\lambda).$$

3. Basic TPKE with short ciphertext and semi-adaptive security

In this section, we propose a basic TPKE construction. The construction is based on Shamir's secret sharing scheme [30]. The basic scheme has constant-size ciphertexts and is proven to be secure without using random oracles.

3.1. A building block: Shamir's secret sharing

Our system exploits the Shamir's (t,n)-threshold secret sharing scheme [30]. The scheme in [30] is defined over finite fields, but we will just present the construction over prime fields we will use in this work. Let $\mathbb{Z}_p$ be a finite field with $p > n$ and $x \in \mathbb{Z}_p$ be the secret to be shared. The dealer picks a polynomial f of degree $t-1$ with coefficients in $\mathbb{Z}_p$ at random, whose free term is the secret x, that is, $f(0) = x$. The polynomial f can be written as

$$f(\alpha) = x + a_1\alpha + \cdots + a_{t-1}\alpha^{t-1} \bmod p,$$

where $a_1, \ldots, a_{t-1} \in \mathbb{Z}_p$. Each shareholder k is assigned a known index $k \in \{1, \ldots, n\}$ and the dealer privately sends to shareholder k a share $x_k = f(k)$. Then any subset of t holders $\mathbb{A} \subset \{1, \ldots, n\}$ can recover the secret $x = f(0)$ by interpolation

$$x = f(0) = \sum_{k\in\mathbb{A}} x_k\lambda_k = \sum_{k\in\mathbb{A}} f(k)\lambda_k,$$

where $\lambda_k = \prod_{\ell\in\mathbb{A}}^{\ell\neq k}\frac{\ell}{\ell-k}$ are the Lagrange coefficients. Actually, shareholders in $\mathbb{A}$ can reconstruct the polynomial because

$$f(\alpha) = \sum_{k\in\mathbb{A}} f(k)\left(\prod_{\ell\in\mathbb{A}}^{\ell\neq k}\frac{\ell-\alpha}{\ell-k}\right).$$

Shamir's secret sharing scheme has several desirable properties:

(1) If an attacker obtains at most $t-1$ shares, the shared secret x stays information-theoretically secure vs the attacker. That is, the attacker cannot get any information about x and, from the attacker's viewpoint, x is uniformly random.

(2) Shamir's secret sharing scheme is linear. So the sum of secrets can be obtained from the sum of their shares. Observe that, if $\phi(\alpha) = f(\alpha) + h(\alpha)$, then we have

$$\phi(\alpha) = \sum_{k\in\mathbb{A}} \phi(k)\left(\prod_{\ell\in\mathbb{A}}^{\ell\neq k}\frac{\ell-\alpha}{\ell-k}\right) = \sum_{k\in\mathbb{A}} f(k)\left(\prod_{\ell\in\mathbb{A}}^{\ell\neq k}\frac{\ell-\alpha}{\ell-k}\right) + \sum_{k\in\mathbb{A}} h(k)\left(\prod_{\ell\in\mathbb{A}}^{\ell\neq k}\frac{\ell-\alpha}{\ell-k}\right).$$

(3) Let $\mathbb{G}$ be a finite cyclic group of order p and let g be the generator of $\mathbb{G}$. A variant of Shamir's secret sharing scheme allows the dealer to distribute shares to users in such a way that t users can only reconstruct g^x, instead of reconstructing x. Furthermore, this variant does not require the dealer to know $x, a_1, \ldots, a_{t-1}$, provided that the dealer knows $g^x, g^{a_1}, \ldots, g^{a_{t-1}}$. Let $F(\alpha) = g^{x+a_1\alpha+\cdots+a_{t-1}\alpha^{t-1}}$; the dealer assigns to the shareholder k

$$F(k) = g^{f(k)} = g^x (g^{a_1})^k \cdots (g^{a_{t-1}})^{k^{t-1}},$$

which can be computed with knowledge of $g^x, g^{a_1}, \ldots, g^{a_{t-1}}$. Then any t holders can recover the secret $g^x = F(0)$ from their shares

$$g^x = F(0) = \prod_{k\in\mathbb{A}} g^{x_k\lambda_k} = \prod_{k\in\mathbb{A}} F(k)^{\prod_{\ell\in\mathbb{A}}^{\ell\neq k}\frac{\ell}{\ell-k}}.$$

3.2. The proposed TPKE scheme

Our schemes are implemented using bilinear pairings on groups which have been widely employed to build cryptographic systems [12,22,24]. Let `PairGen` be an algorithm that, on input a security parameter 1^λ, outputs a tuple $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ have the same prime order p, and $e: \mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ is an efficient non-degenerate bilinear map such that $e(g,g) \neq 1$ for any generator g of $\mathbb{G}$, and for all $x, y \in \mathbb{Z}$, it holds that $e(g^x, g^y) = e(g,g)^{xy}$. By employing the recent Gentry–Waters broadcast scheme [20] and the Shamir secret sharing scheme, our TPKE scheme is realized as follows:

- *Setup.* Let `PairGen` be an algorithm that, on input a security parameter 1^λ, outputs a tuple $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ have the same prime order p, and $e: \mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ is an efficient non-degenerate bilinear map. Let $h_1, \ldots, h_n$ be randomly chosen from $\mathbb{G}$. The system parameters are $\pi = (\Upsilon, g, h_1, \ldots, h_n, t, n)$. In the following, we assume that each user is uniquely identified by an index $i \in \{1, 2, \ldots, n\}$. This can be implemented by ordering the users by the order in which they join the system.
- *KeyGen.* Randomly select $x \in \mathbb{Z}_p$ and a polynomial $f \in \mathbb{Z}_p[\alpha]$ of degree $t-1$ such that $f(0) = x$ and compute

  $$X = e(g,g)^x.$$

  The TPKE master public key is $MPK = X$ and the TPKE master secret key is

  $$msk = \langle x, f\rangle.$$

- *Join.* Assume the polynomial $f(\alpha) = x + a_1\alpha + \cdots + a_{t-1}\alpha^{t-1} \bmod p$. Let the ith user want to join the system. The dealer computes $S_i = g^{f(i)}$. The dealer randomly selects $r_i \in \mathbb{Z}_p$ and computes the secret decryption key of user i as

  $$udk_i = \left(g^{-r_i}, h_1^{r_i}, \ldots, h_{i-1}^{r_i}, g^{f(i)}h_i^{r_i}, h_{i+1}^{r_i}, \ldots, h_n^{r_i}\right).$$

  The dealer privately sends udk_i to user i and sets user i's public key UPK_i to i.
- *Encrypt.* For a receiver set $\mathbb{R}$, randomly pick γ in $\mathbb{Z}_p$ and compute

  $$Hdr = (c_1, c_2) : c_1 = g^\gamma, c_2 = \left(\prod_{j\in\mathbb{R}} h_j\right)^\gamma.$$

  Set $sk = e(g,g)^{x\gamma}$ and output $\langle Hdr, sk\rangle$. Send $\langle\mathbb{R}, Hdr\rangle$ to the authorized receivers. Note that the validity of the encryption can be publicly verified by checking $e(g, c_2) = e(c_1, \prod_{j\in\mathbb{R}} h_j)$.
- *ShareDecrypt.* If $i \in \mathbb{R}$, user i can extract a session key share of sk from Hdr with his decryption key udk_i by computing

$$e\left(g^{f(i)}h_i^{r_i}\prod_{j\in\mathbb{R}\setminus\{i\}}h_j^{r_i},c_1\right)e(g^{-r_i},c_2)=e\left(g^{f(i)}\left(\prod_{j\in\mathbb{R}}h_j\right)^{r_i},g^{\gamma}\right)e\left(g^{-r_i},\left(\prod_{j\in\mathbb{R}}h_j\right)^{\gamma}\right)=e(g,g)^{f(i)\gamma}\overset{\mathtt{Def}}{=}\sigma_i.$$

Note that decryption is non-interactive.

- **Combine.** Assume a set $\mathbb{A}\subseteq\mathbb{R}$ of size at least t. The users in $\mathbb{A}$ decrypt their respective session key shares. Then they can recover the secret session key $sk = (g,g)^{x\gamma} = e(g,g)^{f(0)\gamma}$ from their shares

$$sk=(g,g)^{x\gamma}=e(g,g)^{f(0)\gamma}=\prod_{i\in\mathbb{A}}e(g,g)^{f(i)\lambda_i\gamma}=\prod_{i\in\mathbb{A}}(e(g,g)^{f(i)\gamma})^{\lambda_i}=\prod_{i\in\mathbb{A}}\sigma_i^{\lambda_i},$$

where $\lambda_i=\prod_{j\in\mathbb{A}}^{j\neq i}\frac{j}{j-i}$ are the Lagrange coefficients.

3.3. Security analysis

The security of the schemes that we propose in this paper relies on the decision BDHE problem. The corresponding decision BDHE assumption is shown to be sound by Boneh et al. [5] in the generic group model. This assumption has been widely followed up for cryptographic constructions (e.g., [6,7,20,35,36]). We briefly review the decision BDHE assumption in $\mathbb{G}$ as follows.

Definition 4 (*Decision BDHE Problem*). Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p with bilinear map $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$, and let g be a generator for $\mathbb{G}$. Let $\beta,\gamma\leftarrow\mathbb{Z}_p$ and $b\leftarrow\{0,1\}$. If $b = 0$, set $Z=e(g,g)^{\beta^{n+1}\gamma}$; else, set $Z\leftarrow\mathbb{G}_T$. The problem instance consists of

$$\{g^{\gamma},Z\}\cup\{g^{\beta^i}:i\in[0,n]\cup[n+2,2n]\}.$$

The problem is to guess b. An attacker $\mathcal{A}$ wins if it correctly guesses b and its advantage is defined by $\mathsf{AdvBDHE}_{\mathcal{A},n}(\lambda)=\left|\Pr[\mathcal{A}\ wins]-\frac{1}{2}\right|$. The Decision BDHE assumption states that, for any polynomial-time probabilistic attacker $\mathcal{A}$, $\mathsf{AdvBDHE}_{\mathcal{A},n}(\lambda)$ is negligible in λ.

Based on the decision BDHE assumption, regarding the security of our TPKE scheme we have the following claim.

Theorem 5. *Let $\mathcal{A}$ be a semi-adaptive attacker breaking the above system with advantage ϵ in time τ. Then there is an algorithm $\mathcal{B}$ breaking the Decision BDHE assumption with advantage ϵ' in time τ', where $\epsilon\prime\geqslant 1C_n^{t-1}\varepsilon,\tau'\leqslant\tau+\mathcal{O}(1)\tau_{\mathtt{Pair}}+\mathcal{O}(n^2)\tau_{\mathtt{Exp}}$, where $\tau_{\mathtt{Pair}}$ denotes the overhead to compute a pairing, and $\tau_{\mathtt{Exp}}$ denotes the time complexity to compute one exponentiation without differentiating exponentiations in different groups.*

Proof. The proof outline is as follows. We construct an algorithm $\mathcal{B}$ to break an instance of the Decision *BDHE* assumption by invoking the attacker $\mathcal{A}$ against our scheme as a black box. $\mathcal{B}$ is given an instance of the Decision *BDHE* challenge. With it, $\mathcal{B}$ simulates the system parameters, the public keys of the dealer, the decryption keys of the corrupted users, and the challenging ciphertext which the attacker may query for. The simulated data are indistinguishable from those generated in a real scheme from the viewpoint of the attacker, so that the attacker does not know she is interacting with a simulator. Then $\mathcal{B}$ uses $\mathcal{A}$'s guess to solve the Decision *BDHE* challenge. Since the *BDHE* assumption is assumed to hold, such an algorithm $\mathcal{B}$ does not exist. Therefore, a successful attacker $\mathcal{A}$ against our scheme doest not exist and our scheme is secure.

$\mathcal{B}$ receives the BDHE challenge instance, which includes g^{γ}, Z, and the set $\{g^{\beta^i}:i\in[0,n]\cup[n+2,2n]\}$.

Initialization. $\mathcal{A}$ commits to a set $\overline{\mathbb{R}}\subseteq[1,n]$.$\mathcal{A}$ is allowed to obtain the system parameters, the dealer's public key (*i.e.*, the TPKE master public key), the public keys of all users, the decryption keys of the corrupted users outside $\overline{\mathbb{R}}$, and at most $t-1$ decryption keys of the corrupted users in $\overline{\mathbb{R}}$. The queries can be made at any point, either before or after the attacker is challenged with the challenge header. $\mathcal{B}$ answers the queries in a consistent way.

Setup. $\mathcal{B}$ generates $y_0,\ldots,y_n\leftarrow\mathbb{Z}_p$. It sets $h_i=g^{y_i}$ for $i\in\overline{\mathbb{R}},h_i=g^{y_i+\beta^i}$ for $i\in[1,n]\setminus\overline{\mathbb{R}}$.$\mathcal{B}$ outputs $\pi = (\Upsilon,g,h_1,\ldots,h_n,t,n)$ as the system parameters. Clearly, due to the randomness of y_i, π has the same distribution as in the real scheme and the simulation of the system parameters is perfect.

Public key simulation. Define $x = y_0\beta^{n+1}$ which is unknown to $\mathcal{B}$. However, $\mathcal{B}$ can compute $X=e(g,g)^x=e(g^{\beta},g^{\beta^n})^{y_0}$. The dealer's public key is X. Due to the randomness of y_0, the dealer's public key is well-formed and has the same distribution as in the real scheme. The simulation of the dealer's public key is also perfect. The public key of a user is his corresponding index i which can be trivially simulated by doing as in the real scheme.

Simulation of the decryption keys of the corrupted users. By exploiting Property 3 of Shamir's secret sharing scheme, $\mathcal{B}$ can simulate the decryption keys of users for indices outside $\overline{\mathbb{R}}$ and $t-1$ decryption keys of users for indices in $\overline{\mathbb{R}}$. The simulation is done as follows.

To simulate the decryption keys of the corrupted users, $\mathcal{B}$ first needs to find a polynomial

$$f(\alpha)=x+a_1\alpha+\cdots+a_{t-1}\alpha^{t-1}\ \text{mod}\ p, \tag{1}$$

for some coefficients $x,a_1,\ldots,a_{t-1}\in\mathbb{Z}_p$. To this end, $\mathcal{B}$ randomly chooses a subset $\mathbb{A}^*\subseteq\overline{\mathbb{R}}$ with $t-1$ indices. For $k\in\mathbb{A}^*$, $\mathcal{B}$ randomly selects $S_k\in\mathbb{Z}_p$ and sets $f(k) = S_k$. Note that $f(0) = x$. Then, according to the Lagrange interpolation formula, with these t points $(0,f(0))$ and $(k,f(k))$ for $k\in\mathbb{A}^*$, $f(\alpha)$ can be rewritten in the following form:

$$f(\alpha) = \sum_{k \in \mathbb{A}^* \cup \{0\}} f(k) \left(\prod_{\ell \in \mathbb{A}^* \cup \{0\}}^{\ell \neq k} \frac{\ell - \alpha}{\ell - k} \right) = \prod_{\ell \in \mathbb{A}^*} \frac{\ell - \alpha}{\ell} x + \sum_{k \in \mathbb{A}^*} f(k) \left(\frac{\alpha}{k} \prod_{\ell \in \mathbb{A}^*}^{\ell \neq k} \frac{\ell - \alpha}{\ell - k} \right). \tag{2}$$

Then we can denote $f(\alpha)$ by

$$f(\alpha) = \prod_{\ell \in \mathbb{A}^*} \frac{\ell - \alpha}{\ell} x + f'(\alpha), \tag{3}$$

where $f'(\alpha) = \sum_{k \in \mathbb{A}^*} f(k) \left(\frac{\alpha}{k} \prod_{\ell \in \mathbb{A}^*}^{\ell \neq k} \frac{\ell - \alpha}{\ell - k} \right)$.

To simulate the decryption key of the user $i \in \mathbb{A}^*$, $\mathcal{B}$ randomly selects $r_i \in \mathbb{Z}_p$ and computes

$$udk_i = \left(g^{-r_i}, h_1^{r_i}, \ldots, h_{i-1}^{r_i}, g^{S_i} h_i^{r_i}, h_{i+1}^{r_i}, \ldots, h_n^{r_i} \right). \tag{4}$$

Notice that $g^{S_i} = g^{f(i)}$ and that r_i is random. The decryption key of user $i \in \mathbb{A}^*$ is well-formed and perfectly simulated. However, if the attacker asks for the decryption keys of users in $\overline{\mathbb{R}} \setminus \mathbb{A}^*, \mathcal{B}$ has to declare FAILURE as it does not know. The probability that this bad event does not happen is $\frac{1}{C_{|\overline{\mathbb{R}}|}^{t-1}} \geqslant \frac{1}{C_n^{t-1}}$.

To simulate the decryption key of the user i not in $\overline{\mathbb{R}}, \mathcal{B}$ randomly chooses $z_i \leftarrow \mathbb{Z}_p$ and formally sets

$$r_i = \prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (z_i - y_0 \beta^{n+1-i}).$$

It outputs $udk_i = (d_{i,0}, \ldots, d_{i,n})$:

$$d_{i,0} = g^{-r_i}, \quad d_{i,i} = g^{f(i)} h_i^{r_i}, \quad d_{i,j} = h_j^{r_i} (\forall j \neq i).$$

Notice that $\mathcal{B}$ can compute all these terms from the BDHE challenge instance, and from Eq. (3); in particular

$$d_{i,i} = g^{f(i)} h_i^{r_i} = g^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (y_0 \beta^{n+1}) + f'(i)} h_i^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (z_i - y_0 \beta^{n+1-i})} = g^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (y_0 \beta^{n+1}) + f'(i)} (g^{y_i + \beta^i})^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (z_i - y_0 \beta^{n+1-i})}$$
$$= g^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (y_0 \beta^{n+1}) + f'(i) + (y_i + \beta^i) \left(\prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (z_i - y_0 \beta^{n+1-i}) \right)} = g^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell} (y_i (z_i - y_0 \beta^{n+1-i}) + \beta^i z_i) + f'(i)}$$

which can be computed since the term β^{n+1} in the exponent cancels out. Clearly, udk_i is well-formed, and due to the randomness of z_i and y_0, udk_i has the same distribution as in the real world. Hence, for all users outside $\overline{\mathbb{R}}$, the simulation of their decryption keys is perfect. $\mathcal{B}$ can correctly answer all the decryption keys queries for users outside $\overline{\mathbb{R}}$.

Challenge. $\mathcal{A}$ chooses a subset $\mathbb{R}^* \subset \overline{\mathbb{R}}$. $\mathcal{B}$ sets $Hdr = (c_1, c_2)$ with

$$c_1 = g^{\gamma}, c_2 = \left(\prod_{j \in \mathbb{R}^*} h_j \right)^{\gamma}.$$

$\mathcal{B}$ sets $sk \leftarrow Z^{y_0}$. $\mathcal{B}$ sends $\langle Hdr, sk \rangle$ to $\mathcal{A}$.

Notice that $\mathcal{B}$ can compute these terms from the BDHE instance. The values of c_1 and sk come directly from the instance. And $\mathcal{B}$ can compute c_2 since it knows $\log_g(h_i)$ for all $i \in \mathbb{R}^*$; in particular,

$$c_2 = \left(\prod_{j \in \mathbb{R}^*} h_j \right)^{\gamma} = \left(\prod_{j \in \mathbb{R}^*} g^{y_j} \right)^{\gamma} = (g^{\gamma})^{\sum_{j \in \mathbb{R}^*} y_j}$$

Guess. Eventually, $\mathcal{A}$ outputs a bit b'. Upon receiving $\mathcal{A}$'s guess bit $b', \mathcal{B}$ sends b' to the BDHE challenger.

Success probability. We compute the advantage of $\mathcal{B}$ to break the BDHE assumption. Assume that $\mathcal{B}$ does not declare FAILURE during the semi-adaptive game. When $b = 0$ in the semi-adaptive game, $\langle Hdr, sk \rangle$ is generated according to the same distribution as in the real world. This is also true in $\mathcal{B}$'s simulation: when $b = 0$, then we have that $Z = e(g, g)^{\beta^{n+1}\gamma}$, $sk = Z^{y_0} = e(g, g)^{x\gamma}$, where $x = y_0 \beta^{n+1}$, and so the challenge is a valid ciphertext under random γ. When $b = 1$ in the semi-adaptive game, $\langle Hdr, sk' \rangle$ is generated as in the real world, but sk' is replaced by $sk \leftarrow \mathbb{K}$, and $\langle Hdr, sk \rangle$ is sent to the attacker. This distribution is identical to that of $\mathcal{B}$'s simulation, where Hdr is valid for random γ, but $sk = Z^{y_0}$ is a uniformly random element of $\mathbb{G}_T$ because Z is randomly chosen from $\mathbb{G}_T$. From this, we see that $\mathcal{B}$'s advantage in deciding the BDHE instance is precisely $\mathcal{A}$'s advantage against TPKE, if $\mathcal{B}$ has not declared FAILURE during the game (this event occurs with probability at least $\frac{1}{C_n^{t-1}}$). Considering this factor, we have that the advantage of $\mathcal{B}$ is $\epsilon' \geqslant \frac{1}{C_n^{t-1}} \varepsilon$.

Time complexity. The additional overhead for $\mathcal{B}$ is to simulate the answers that $\mathcal{A}$ may query. In the Setup stage, $\mathcal{B}$ needs $n + 1$ exponentiations in $\mathbb{G}$. In the Public key simulation, $\mathcal{B}$ needs one exponentiation and one pairing operation to compute $e(g^{\beta}, g^{\beta^n})$. In the Simulation of the decryption keys of corrupted users, the main computation cost of $\mathcal{B}$ is at most $(t - 1)(n + 2) + (n - t)(n + t + 1)$ exponentiations. In the Challenge stage, $\mathcal{B}$ needs one exponentiation in $\mathbb{G}$ to compute the challenge ciphertext. Let $\tau_{\texttt{Pair}}$ denote the overhead to compute a pairing, and $\tau_{\texttt{Exp}}$ denote the time complexity to compute one exponentiation without differentiation of exponentiations in different groups. Note that $1 \leqslant t \leqslant n$. By summing up, the time complexity of $\mathcal{B}$ is $\tau' \leqslant \tau + \mathcal{O}(1)\tau_{\mathtt{Pair}} + \mathcal{O}(n^2)\tau_{\mathtt{Exp}}$. □

One may note that we have a reduction loss by a factor $\frac{1}{C_n^t}$. However, since t is usually very small and can be viewed as a constant in practice, the reduction loss is $\frac{1}{poly(\lambda)}$ even if n is a polynomial in λ.

4. Variants

4.1. Shortening system parameters

In the basic construction, we need $h_1,\ldots,h_n$ as system parameters. One may observe that $h_1,\ldots,h_n$ can be generated with a hash function $H:\{0,1\}^* \to \mathbb{G}$, *e.g.*, $h_i = H(i)$. After applying this modification, one can remove $h_1,\ldots,h_n$ from the system parameter list to shorten the system parameters. The cost is that the proof needs a random oracle to model the hash function.

4.2. TPKE with adaptive security

The above constructions only achieve semi-adaptive security. However, by applying the generic transformation from semi-adaptive security to fully adaptive security in Section 2.3, the basic scheme and its above variant can be readily improved to meet fully adaptive security, at a cost of doubling ciphertexts.

4.3. Trade-off between ciphertext size and decryption key size

In the above short-parameter variants (with semi-adaptive or adaptive security), the public key requires $\mathcal{O}(1)$ elements and the ciphertext is also of $\mathcal{O}(1)$ size. However, the decryption key of each user consists of $\mathcal{O}(n)$ elements. In the following, we illustrate an efficient trade-off between the size of the decryption keys and the ciphertexts.

Let $n = n_1^2$. Divide the maximal receiver group $\{1,\ldots,n\}$ into n_1 subgroups each of which hosts at most n_1 receivers. Then one can concurrently apply our basic TPKE scheme to each subgroup when a sender wants to broadcast to a set of users $\mathbb{R} \subseteq \{1,\ldots,n\}$. After employing this approach, the public broadcast key, the decryption key of each user, and the ciphertext all consist of $\mathcal{O}(n_1)$ elements. The detailed variant is given as follows.

- *Setup.* Let $\mathtt{PairGen}$ be an algorithm that, on input a security parameter 1^λ, outputs a tuple $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ have the same prime order p, and $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficient non-degenerate bilinear map. Let $H:\{0,1\}^* \to \mathbb{G}$ be a cryptographic hash function. The system parameters are $\pi = (\Upsilon, g, H, t, n)$.
- *KeyGen.* Randomly select $x_1,\ldots,x_{n_1},\ a_1,\ldots,a_{t-1} \in \mathbb{Z}_p$ and compute

  $$X_1 = e(g,g)^{x_1},\ldots,X_{n_1} = e(g,g)^{x_{n_1}}.$$

  The TPKE master public key is $MPK = \{X_1,\ldots,X_{n_1}\}$ and the TPKE master secret key is

  $$msk = \langle x_1,\ldots,x_{n_1},a_1,\ldots,a_{t-1}\rangle.$$

- *Join.* Let the ith user want to join the system. Assume that $i = un_1 + v$ where $1 \leqslant v \leqslant n_1$. The dealer generates a secret polynomial

  $$f(\alpha) = x + a_1\alpha + \cdots + a_{t-1}\alpha^{t-1} \bmod p$$

  and computes

  $$S_i = g^{f(i)}.$$

  The dealer randomly selects $r_i \in \mathbb{Z}_p$ and computes the secret decryption key of user i by computing udk_i which is

  $$\left(g^{-r_i}, H(u,1)^{r_i},\ldots,H(u,v-1)^{r_i}, g^{f(i)}H(u,v)^{r_i}, H(u,v+1)^{r_i},\ldots,H(u,n_1)^{r_i}\right)$$

  The dealer privately sends udk_i to user i and sets user i's public key UPK_i to i.
- *Encrypt.* For a receiver set $\mathbb{R}$, randomly pick γ in $\mathbb{Z}_p$ and compute

  $$Hdr = (c_0, c_1,\ldots,c_{n_1}) : c_0 = g^\gamma, c_{u+1} = \left(\prod_{k\in\mathbb{R}_u} H(u,k)\right)^\gamma,$$

  where $u = 0,\ldots,n_1-1$; $\mathbb{R}_u = \mathbb{R} \cap \{un_1+1,\ldots,un_1+n_1\}$. Set $sk = e(g,g)^{x\gamma}$ and output $\langle Hdr, sk\rangle$. Send $\langle\mathbb{R}, Hdr\rangle$ to the authorized receivers.
- *ShareDecrypt.* If $i = un_1 + v \in \mathbb{R}$, user i can extract a session key share of sk from Hdr with his decryption key udk_i by computing

$$e\left(g^{f(i)}H(u,v)^{r_i}\prod_{k\in\mathbb{R}_u\setminus\{v\}}H(u,k)^{r_i},c_0\right)e(g^{-r_i},c_{u+1}) = e\left(g^{f(i)}\left(\prod_{k\in\mathbb{R}_u}H(u,k)\right)^{r_i},g^{\gamma}\right)e\left(g^{-r_i},\left(\prod_{k\in\mathbb{R}_u}H(u,k)\right)^{\gamma}\right) = e(g,g)^{f(i)\gamma}\overset{\mathrm{Def}}{=}\sigma_i.$$

- *Combine.* Assume that t users in $\mathbb{A}\subseteq\mathbb{R}$ decrypt their respective session key shares. Then they can recover the secret session key $sk = e(g,g)^{x\gamma} = e(g,g)^{f(0)\gamma}$ by interpolating

  $$sk = e(g,g)^{x\gamma} = e(g,g)^{f(0)\gamma} = \prod_{i\in\mathbb{A}}e(g,g)^{f(i)\lambda_i\gamma} = \prod_{i\in\mathbb{A}}(e(g,g)^{f(i)\gamma})^{\lambda_i} = \prod_{i\in\mathbb{A}}\sigma_i^{\lambda_i},$$

  where $\lambda_i = \prod_{j\in\mathbb{A}}^{j\neq i}\frac{j}{j-i}$ are the Lagrange coefficients.

This trade-off approach is also applicable to the above adaptively secure variant with short parameters. Hence, the resulting adaptively secure TPKE scheme has sublinear complexity, *i.e.*, $\mathcal{O}(\sqrt{n})$ size public keys, decryption keys and ciphertexts.

4.4. Improvements for transmission of multiple session keys

The core of TPKE is to transmit the session key that is used in symmetric encryption for the purpose of access control of sensitive information. In principle, if multiple session keys are required, they can be transmitted by running the underlying TPKE protocol for multiple runs. In the following, based on ramp secret sharing schemes [4], we present a more efficient way to transmit multiple session keys by slightly modifying the above TPKE instantiations.

A $(\triangle,t,n)-$ *ramp secret sharing scheme* is a scheme in which the secret is shared among a set of n participants in such a way that for every subset of shareholders $\mathbb{A}$, if $|\mathbb{A}|\leqslant t-1$ then $\mathbb{A}$ does not obtain any information about the secret, and if $|\mathbb{A}|\geqslant t+\triangle-1$ then $\mathbb{A}$ can obtain the secret. Clearly, threshold secret sharing schemes are ramp schemes with $\triangle = 1$. In a threshold secret sharing scheme the size of each share must be greater than or equal to the size of the secret. However, this drawback can be overcome by using ramp schemes. In a $(\triangle,t,n)-$ ramp scheme, the size of each share must be bigger or equal than $1/\triangle$ the size of the secret. The ramp secret sharing schemes we describe below, which are based on the Shamir construction and were presented in [4], reach this bound. As in Section 3, our construction is defined over prime fields. We use these ramp secret sharing schemes to build TPKE schemes allowing efficient transmission of multiple session keys.

Let $\{1,\ldots,n\}$ be the set of participants of the scheme and $\mathbb{Z}_p$ be a finite field with $p > n + \triangle$and $n\geqslant t+\triangle-1$. Assume that $\mathrm{x} = (x_0,\ldots,x_{\triangle-1})\in\mathbb{Z}_p^{\triangle}$ is the secret to be shared. The dealer chooses $\xi_0,\ldots,\xi_{\triangle-1}\in\mathbb{Z}_p\setminus\{1,\ldots,n\}$ and a secret polynomial $F\in\mathbb{Z}_p[X]$ of degree $t + \triangle - 2$ such that

$$F(\xi_i) = x_i$$

for every $0\leqslant i\leqslant\triangle-1$. Then the dealer privately sends $F(j)$ to the jth participant for $j\in\{1,\ldots,n\}$. Observe that any set $\mathbb{A}\subseteq\{1,\ldots,n\}$ of $t + \triangle - 1$ participants can compute $\mathbf{x}$, since for every $0\leqslant i\leqslant\triangle-1$, it holds that

$$x_i = \sum_{j\in\mathbb{A}}\lambda_{j,i}F(j),\quad \text{where } \lambda_{j,i} = \prod_{k\in\mathbb{A}}^{k\neq j}\frac{k-\xi_i}{k-j}.$$

Moreover, any set $\mathbb{A}\subseteq\{1,\ldots,n\}$ with $|\mathbb{A}|\leqslant t-1$ will not obtain any information about $\mathbf{x}$.

In the following, we only show the improvements on the basic TPKE. The modifications on the extended TPKE variants are similar and omitted to avoid repetition. The modifications on the basic TPKE are as follows:

- *KeyGen.* Randomly select $x_0,\ldots,x_{\triangle-1}$ and compute

  $$X_0 = e(g,g)^{x_0},\ldots,X_{\triangle-1} = e(g,g)^{x_{\triangle-1}}.$$

  The dealer chooses $\xi_0,\ldots,\xi_{\triangle-1}\in\mathbb{Z}_p\setminus\{1,\ldots,n\}$ and a secret polynomial $F\in\mathbb{Z}_p[X]$ of degree $t + \triangle - 2$ such that

  $$F(\xi_i) = x_i$$

  for every $0\leqslant i\leqslant\triangle-1$. Without loss of generality, we can set $\xi_0 = 0, \xi_1 = n + 1,\ldots,\xi_{\triangle-1} = n + \triangle - 1$. The TPKE master public key is $MPK = \langle X_0,\ldots,X_{\triangle-1}\rangle$ and the TPKE master secret key is

  $$msk = \langle x_0,\ldots,x_{\triangle-1},F\rangle.$$

- *Join.* Let the ith user want to join the system. The dealer randomly selects $r_i\in\mathbb{Z}_p$ and computes the secret decryption key of user i as

  $$udk_i = (g^{-r_i},h_1^{r_i},\ldots,h_{i-1}^{r_i},g^{F(i)}h_i^{r_i},h_{i+1}^{r_i},\ldots,h_n^{r_i}).$$

  Privately send udk_i to user i and set user i's public key UPK_i to i.
- *Encrypt.* For a receiver set $\mathbb{R}$, randomly pick γ in $\mathbb{Z}_p$ and compute

  $$Hdr = (c_1,c_2) : c_1 = g^{\gamma}, c_2 = (\prod_{j\in\mathbb{R}}h_j)^{\gamma}.$$

Set the session key vector

$$\mathtt{sk} = (sk_0, \ldots, sk_{\triangle-1}) = (e(g,g)^{x_0\gamma}, \ldots, e(g,g)^{x_{\triangle-1}\gamma})$$

and output $\langle Hdr, \mathbf{sk}\rangle$. Send $\langle \mathbb{R}, Hdr\rangle$ to the authorized receivers.

- *ShareDecrypt.* If $i \in \mathbb{R}$, the user i can extract a session key share of **sk** from *Hdr* with his decryption key udk_i by computing

$$e\left(g^{f(i)}H(u,v)^{r_i}\prod_{k\in\mathbb{R}_u\setminus\{v\}}H(u,k)^{r_i}, c_0\right)e(g^{-r_i}, c_{u+1}) = e\left(g^{f(i)}\left(\prod_{k\in\mathbb{R}_u}H(u,k)\right)^{r_i}, g^{\gamma}\right)e\left(g^{-r_i}, \left(\prod_{k\in\mathbb{R}_u}H(u,k)\right)^{\gamma}\right) = e(g,g)^{f(i)\gamma}\overset{\mathtt{Def}}{=}\sigma_i.$$

Note that decryption is non-interactive.

- *Combine.* Assume that $t + \triangle - 1$ users in $\mathbb{A} \subseteq \mathbb{R}$ decrypt their respective session key shares. Then they can recover the secret session key vector

$$\mathtt{sk} = \left(e(g,g)^{x_0\gamma}, \ldots, e(g,g)^{x_{\triangle-1}\gamma}\right) = \left(e(g,g)^{F(\xi_1)\gamma}, \ldots, e(g,g)^{F(\xi_{\triangle-1})\gamma}\right)$$

because for $i = 0, \ldots, \triangle - 1$,

$$sk_i = e(g,g)^{x_i\gamma} = e(g,g)^{F(\xi_i)\gamma} = \prod_{j\in\mathbb{A}}e(g,g)^{\lambda_{j,i}F(j)\gamma} = \prod_{j\in\mathbb{A}}\sigma_j^{\lambda_{j,i}}.$$

By applying the above improvements, the same header allows transmitting a secret session key vector rather than a single session key. Due to the independence of $x_0, \ldots, x_{\triangle-1}$, it can be seen that $sk_i = e(g,g)^{x_i\gamma}$ for $i = 0, \ldots, \triangle - 1$ are also independent. Hence, the $\triangle$ session keys can be used for accessing sensitive information more efficiently than a single session key.

Regarding the security of the improved protocol above, we have the following claim.

Theorem 6. *Let $\mathcal{A}$ be a semi-adaptive attacker who can distinguish any session key, i.e., sk_k for $k \in \{0, \ldots, \triangle - 1\}$, from a random element in the session key space with advantage ϵ in time τ. Then, there is an algorithm $\mathcal{B}$ breaking the Decision BDHE assumption with advantage ϵ' in time τ', where $\epsilon' \geqslant \frac{1}{C_n^{t-1}}\varepsilon, \tau' \leqslant \tau + \mathcal{O}(1)\tau_{\mathtt{Pair}} + \mathcal{O}(n^2)\tau_{\mathtt{Exp}}$.*

Proof. The proof is similar to that of Theorem 5. The main difference is that $\mathcal{B}$ in this case has to simulate more public key components using the same BDHE challenge and the decryption keys of the corrupted users correspond to a polynomial F of higher order.

Let $\mathcal{B}$ receive the BDHE challenge instance, which includes g^{γ}, Z and the set $\{g^{\beta^i} : i \in [0,n] \cup [n+2, 2n]\}$. The Initialization procedure is the same as the one in the proof of Theorem 5.

Setup. $\mathcal{B}$ generates $y_0, \ldots, y_n \leftarrow \mathbb{Z}_p$. It sets $h_i = g^{y_i}$ for $i \in \overline{\mathbb{R}}, h_i = g^{y_i+\beta^i}$ for $i \in [1,n] \setminus \overline{\mathbb{R}}. \mathcal{B}$ outputs $\pi = (\Upsilon, g, h_1, \ldots, h_n, t, \triangle, n)$ as the system parameters. Clearly, due to the randomness of y_i, π has the same distribution as in the real scheme and the simulation of the system parameters is perfect.

Public key simulation. $\mathcal{B}$ first does some preparations to simulate the public key of the dealer.

In this preparation stage, $\mathcal{B}$ computes a polynomial f of degree $t - 1$ as that in the proof of Theorem 5. To this end, $\mathcal{B}$ randomly chooses a subset $\mathbb{A}^* \subseteq \overline{\mathbb{R}}$ with $t - 1$ indices. For $k \in \mathbb{A}^*$, $\mathcal{B}$ randomly selects $S_k \in \mathbb{Z}_p$ and sets $f(k) = S_k$. Set $f(0) = x_0 = y_0\beta^{n+1}$ for a randomly chosen value $y_0 \in \mathbb{Z}_p$. Then, according to the Lagrange interpolation formula, with these t points $(0, f(0))$ and $(k, f(k))$ for $k \in \mathbb{A}^*$, $f(\alpha) = x_0 + a_1\alpha + \cdots + a_{t-1}\alpha^{t-1} \bmod p$ can be rewritten in the following form

$$f(\alpha) = \prod_{\ell\in\mathbb{A}^*}\frac{\ell-\alpha}{\ell}x_0 + f'(\alpha) \tag{5}$$

where $f'(\alpha) = \sum_{k\in\mathbb{A}^*}f(k)\left(\frac{\alpha}{k}\prod_{\ell\in\mathbb{A}^*}^{\ell\neq k}\frac{\ell-\alpha}{\ell-k}\right)$.

In the following, we illustrate how to prepare a secret polynomial $F \in \mathbb{Z}_p[X]$ of degree $t + \triangle - 2$ such that $F(\xi_k) = x_k$ for $k = 0, \ldots, \triangle - 1$, where x_k is the kth entry of the session key vector (*i.e.*, the secret session key vector is $x = (x_0, \ldots, x_{\triangle-1}) \in \mathbb{Z}_p^{\triangle}), \xi_0 = 0, \xi_1 = n + 1, \ldots, \xi_{\triangle-1} = n + \triangle - 1$.

For $\triangle = 1$, we set $F(\alpha) = f(\alpha)$. Note that in this case $\triangle = 1$, the security claim has been addressed in Theorem 5. In the following, we assume that $\triangle \geqslant 2$. Set

$$F(\alpha) = f(\alpha) + a_t\alpha^t + \cdots + a_{t+\triangle-2}\alpha^{t+\triangle-2}\overset{\mathtt{Def}}{=}\prod_{\ell\in\mathbb{A}^*}\frac{\ell-\alpha}{\ell}x_0 + F'(\alpha) \tag{6}$$

where $F'(\alpha) = f'(\alpha) + a_t\alpha^t + \cdots + a_{t+\triangle-2}\alpha^{t+\triangle-2} = \sum_{k\in\mathbb{A}^*}f(k)\left(\frac{\alpha}{k}\prod_{\ell\in\mathbb{A}^*}^{\ell\neq k}\frac{\ell-\alpha}{\ell-k}\right) + a_t\alpha^t + \cdots + a_{t+\triangle-2}\alpha^{t+\triangle-2}$, and $(a_t, \ldots, a_{t+\triangle-2})$ is randomly chosen from $\mathbb{Z}_p^{\triangle-1}$.

After the above preparation, $\mathcal{B}$ can simulate the public key of the dealer. For $k \in \{0, \ldots, \triangle - 1\}$, compute

$$X_k = e(g,g)^{F(\xi_k)} = e(g,g)^{\prod_{\ell\in\mathbb{A}^*}\frac{\ell-\alpha}{\ell}x_0 + F'(\xi_k)} = e(g,g)^{\prod_{\ell\in\mathbb{A}^*}\frac{\ell-\alpha}{\ell}y_0\beta^{n+1} + F'(\xi_k)} = e(g^{\beta}, g^{\beta^n})^{y_0\prod_{\ell\in\mathbb{A}^*}\frac{\ell-\alpha}{\ell}}e(g,g)^{F'(\xi_k)},$$

where we define $x_0 = F(\xi_0),\ldots,x_{\triangle-1} = F(\xi_{\triangle-1})$ as the secret keys which correspond to $X_0,\ldots,X_{\triangle-1}$ but are unknown to $\mathcal{B}$. However, from the above equations, the computation of $X_0,\ldots,X_{\triangle-1}$ does not require knowledge of β^{n+1} or $x_0,\ldots,x_{\triangle-1}$ which are unknown to $\mathcal{B}$. Clearly, the dealer's public key is $(X_0,\ldots,X_{\triangle-1})$ which is well-formed, and due to the randomness of x_k (whose randomness comes from the randomness of the polynomial F) for $k = 0,\ldots,\triangle-1$, it can be seen that $(X_0,\ldots,X_{\triangle-1})$ has the same distribution as that in the real world. The public key i of a user i can be trivially simulated.

Simulation of the decryption keys of the corrupted users. $\mathcal{A}$ is allowed to query the decryption keys of at most $t-1$ users in $\mathcal{B}$. If $\mathcal{A}$ queries the decryption key of user $i \in \mathbb{A}^*, \mathcal{B}$ randomly selects $r_i \in \mathbb{Z}_p$ and computes

$$udk_i = \left(g^{-r_i}, h_1^{r_i}, \ldots, h_{i-1}^{r_i}, g^{S_i+a_t i^t+\cdots+a_{t+\triangle-2} i^{t+\triangle-2}} h_i^{r_i}, h_{i+1}^{r_i}, \ldots, h_n^{r_i}\right). \tag{7}$$

Notice that $g^{S_i+a_t i^t+\cdots+a_{t+\triangle-2} i^{t+\triangle-2}} = g^{F(i)}$. The decryption key of user $i \in \mathbb{A}^*$ is well-formed and perfectly simulated. However, if the attacker asks for the decryption keys of users in $\overline{\mathbb{R}} \setminus \mathbb{A}^*, \mathcal{B}$ has to declare FAILURE as it does not know. The probability that this bad event does not happen is $\frac{1}{C_{|\overline{\mathbb{R}}|}^{t-1}} \geqslant \frac{1}{C_n^{t-1}}$.

To simulate the decryption key of user i outside $\overline{\mathbb{R}}, \mathcal{B}$ randomly chooses $z_i \leftarrow \mathbb{Z}_p$ and formally sets

$$r_i = \prod_{\ell \in \mathbb{A}^*} \frac{\ell - i}{\ell}(z_i - y_0\beta^{n+1-i}).$$

It outputs $udk_i = (d_{i,0},\ldots,d_{i,n})$:

$$d_{i,0} = g^{-r_i}, \quad d_{i,i} = g^{F(i)} h_i^{r_i}, \quad d_{i,j} = h_j^{r_i} (\forall j \neq i).$$

Similarly to the proof of Theorem 5, $\mathcal{B}$ can compute all these terms from the BDHE challenge instance, and from Eq. (6), in particular

$$d_{i,i} = g^{F(i)} h_i^{r_i} = g^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell-i}{\ell}(y_i(z_i - y_0\beta^{n+1-i}) + \beta^i z_i) + F'(i)}$$

which can be computed since the term β^{n+1} in the exponent cancels out. Clearly, udk_i is well-formed, and due to the randomness of z_i, y_0, udk_i has the same distribution as in the real world. Hence, for all users outside $\overline{\mathbb{R}}$, the simulation of their decryption keys is perfect. $\mathcal{B}$ can correctly answer all the decryption keys queries for users outside $\overline{\mathbb{R}}$.

Challenge. $\mathcal{A}$ chooses a subset $\mathbb{R}^* \subset \overline{\mathbb{R}}$. $\mathcal{B}$ sets $Hdr = (c_1,c_2)$:

$$c_1 = g^\gamma, c_2 = \left(\prod_{j \in \mathbb{R}^*} h_j\right)^\gamma.$$

It sets $\mathbf{sk} = (sk_0,sk_1,\ldots,sk_{\triangle-1})$, where $sk_0 = Z^{y_0} = X_0^\gamma$ and $sk_i = X_i^\gamma$ for $i = 1,\ldots,\triangle-1$. It sends $\langle Hdr,\mathbf{sk}\rangle$ to $\mathcal{A}$.

Notice that $\mathcal{B}$ can compute these terms from the BDHE instance. The values of c_1 and $\mathbf{sk}$ come directly from the instance and the knowledge of γ and X_i for $i = 0,\ldots,\triangle-1$. $\mathcal{B}$ can compute c_2 since it knows $\log_g(h_i)$ for all $i \in \mathbb{R}^*$; in particular,

$$c_2 = \left(\prod_{j \in \mathbb{R}^*} h_j\right)^\gamma = \left(\prod_{j \in \mathbb{R}^*} g^{y_j}\right)^\gamma = (g^\gamma)^{\sum_{j \in \mathbb{R}^*} y_j}$$

Guess. Eventually, $\mathcal{A}$ outputs a bit b'.$\mathcal{B}$ sends b' to the BDHE challenger.

Success probability. We compute the advantage of $\mathcal{B}$ to break the BDHE assumption. Assume that $\mathcal{B}$ does not declare FAILURE during the semi-adaptive game. When $b = 0$ in the semi-adaptive game, $\langle Hdr,sk\rangle$ is generated according to the same distribution as in the real world. This is also true in $\mathcal{B}$'s simulation: when $b = 0$, $\mathtt{sk} = \langle sk_0,\ldots,sk_{\triangle-1}\rangle = \langle e(g,g)^{x_0\gamma},\ldots, e(g,g)^{x_{\triangle-1}\gamma}\rangle$. We explain this fact with more details. Since $X_k = e(g,g)^{F(\xi_k)} = e(g,g)^{\prod_{\ell \in \mathbb{A}^*} \frac{\ell-\alpha}{\ell} x_0 + F'(\xi_k)}$, we have that $X_k = X_0 e(g,g)^{F'(\xi_k)}$, where $F'(\xi_k)$ is computable by the simulator $\mathcal{B}$. Since $\mathcal{B}$ knows $Z, y_0, F'(\xi_k), g^\gamma, \mathcal{B}$ can perfectly simulate sk_i by computing $sk_i = Z^{y_0}(e(g^\gamma,g)^{F'(\xi_k)}) = X_0^\gamma(e(g,g)^{F'(\xi_k)})^\gamma = (X_0 e(g,g)^{F'(\xi_k)})^\gamma = X_i^\gamma$. So in the case $b = 0$, sk_i for $i = 0,\ldots,\triangle-1$ are all perfectly simulated and the challenge is a valid ciphertext under random integer γ.

We next consider the case $b = 1$ in the semi-adaptive game. In this case, $\langle Hdr,\mathbf{sk}'\rangle$ is generated as in the real world, but $\mathbf{sk}'$ is replaced by $\mathtt{sk} \leftarrow \mathbb{K}^\triangle$, and $\langle Hdr,\mathbf{sk}\rangle$ is sent to the attacker. This distribution is identical to that of $\mathcal{B}$'s simulation, where Hdr is valid for random γ, but sk_k is a uniformly random element of $\mathbb{G}_T$ for every $0 \leqslant k \leqslant \triangle-1$.

From the above, we see that $\mathcal{B}$'s advantage in deciding the BDHE instance is precisely $\mathcal{A}$'s advantage to distinguish a session key vector $\mathbf{sk}$ from a random element in $\mathbb{G}_T^\triangle$, if $\mathcal{B}$ has not declared FAILURE during the game (this event occurs with probability at least $\frac{1}{C_n^{t-1}}$). Considering this factor, we have that the advantage of $\mathcal{B}$ is $\epsilon' \geqslant \frac{1}{C_n^{t-1}}\varepsilon$.

Time complexity. The additional overhead for $\mathcal{B}$ is to simulate the answers that $\mathcal{A}$ may query. In the Setup stage, $\mathcal{B}$ needs $n+1$ exponentiations in $\mathbb{G}$. In the Public key simulation, $\mathcal{B}$ needs $2\triangle$ exponentiations and $\triangle$ pairing operations to compute $e(g^\beta, g^{\beta^n})$. In the Simulation of the decryption keys of corrupted users, $\mathcal{B}$ needs at most $(t-1)(n+2)+(n-t)(n+t+1)$ exponentiations. In the Challenge stage, $\mathcal{B}$ needs one exponentiation in $\mathbb{G}$ to compute the challenge ciphertext. Let $\tau_{\mathtt{Pair}}$ denote the overhead to compute a pairing, and $\tau_{\mathtt{Exp}}$ denote the time complexity to compute one exponentiation without differentiation of exponentiations in different groups. Note that $1 \leqslant t + \triangle \leqslant n$. By summing up, the time complexity of $\mathcal{B}$ is $\tau' \leqslant \tau + \mathcal{O}(1)\tau_{\texttt{Pair}} + \mathcal{O}(n^2)\tau_{\texttt{Exp}}$. □

5. Conclusion

In this paper, we have proposed an efficient TPKE scheme with constant-size ciphertexts and adaptive security, by observing that existing TPKE schemes suffer from either long ciphertexts or can only achieve non-adaptive security. Security has been proven under the decision BDHE assumption in the standard model. This implies that our proposal preserves security even if the attacker adaptively corrupts all the users outside the authorized set and some users in the authorized set, provided that the number of corrupted users in the authorized set is less than a threshold. We have also proposed a number of extensions which allow: (i) an efficient trade-off between the key size and the ciphertext size; (ii) transmission of multiple session keys in a single session with little extra cost.

Acknowledgments

This work is partly supported by the Spanish Government through Projects TSI2007-65406-C03-01 “E-AEGIS”, TIN2009-11689 “RIPUP”, “eVerification” TSI-020100-2009-720, TIN2011–27076-C03-01 “CO-PRIVACY” and CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES”, and by the Government of Catalonia under Grant 2009 SGR 1135, by the national Nature Science Foundation of China through Projects 60970114, 60970115, 60970116, 61173154, 61021004, 11061130539 and 61003214, and by the Fundamental Research Funds for the Central Universities of China through Project 3103004. The last author is partially supported as an ICREA-Acadèmia researcher by the Catalan Government. The authors are with the UNESCO Chair in Data Privacy, but this paper does not necessarily reflect the position of UNESCO nor does it commit that organization.

References

[1] M. Ak, K. Kaya, K. Onarlioglu, A.-A. Selçuk, Efficient broadcast encryption with user profiles, Information Sciences 180 (6) (2010) 1060–1072.

[2] J. Baek, Y. Zheng, Identity-based threshold decryption, in: Proceedings of PKC 2004, LNCS, vol. 2947, Springer-Verlag, 2004, pp. 262–276.

[3] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, Proceedings of the IEEE Symposium on Security and Privacy (S&P) 2007, IEEE Press, 2007, pp. 321–334.

[4] G.R. Blakley, C. Meadows, Security of ramp schemes, in: Proceedings of CRYPTO 1985, LNCS, vol. 196, Springer-Verlag, 1985, pp. 242–268.

[5] D. Boneh, X. Boyen, E.J. Goh, Hierarchical identity based encryption with constant size ciphertext, in: Proceedings of EUROCRYPT 2005, LNCS, vol. 3494, Springer-Verlag, 2005, pp. 440–456.

[6] D. Boneh, C. Gentry, B. Waters, Collusion resistant broadcast encryption with short ciphertexts and private keys, in: Proceedings of CRYPTO 2005, LNCS, vol. 3621, Springer-Verlag, 2005, pp. 258–275.

[7] D. Boneh, A. Sahai, B. Waters, Fully collusion resistant traitor tracing with short ciphertexts and private keys, in: Proceedings of EUROCRYPT'06, LNCS, vol. 4004, Springer-Verlag, 2006, pp. 573–592.

[8] D. Boneh, B. Waters, A fully collusion resistant broadcast, trace, and revoke system, Proceedings of ACM CCS 2006, ACM Press, 2006, pp. 211–220.

[9] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas, Multicast security: a taxonomy and some efficient constructions, Proceedings of IEEE INFOCOM 1999, vol. 2, IEEE Press, 1999, pp. 708–716.

[10] R. Canetti, S. Goldwasser, An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack, in: Proceedings of EUROCRYPT 1999, LNCS, vol. 1592, Springer-Verlag, 1999, pp. 90–106.

[11] R. Canetti, T. Malkin, K. Nissim, Efficient communication-storage tradeoffs for multicast encryption, in: Proceedings of EUROCRYPT 1999, LNCS, vol. 1592, Springer-Verlag, 1999, pp. 459–474.

[12] X. Chen, F. Zhang, H. Tian, B. Wei, W. Susilo, Y. Mu, H. Lee, K. Kim, Efficient generic on-line/off-line (threshold) signatures without key exposure, Information Sciences 178 (21) (2008) 4192–4203.

[13] V. Daza, J. Herranz, P. Morillo, C. Ràfols, CCA2-secure threshold broadcast encryption with shorter ciphertexts, in: Proceedings of ProvSection 2007, LNCS, vol. 4784, Springer-Verlag, 2007, pp. 35–50.

[14] V. Daza, J. Herranz, P. Morillo, C. Ràfols, Ad-hoc threshold broadcast encryption with shorter ciphertexts, Electronic Notes in Theoretical Computer Science 192 (22) (2008) 3–5.

[15] C. Delerablée, D. Pointcheval, Dynamic threshold public-key encryption, in: Proceedings of CRYPTO 2008, LNCS, vol. 5157, Springer-Verlag, 2008, pp. 317–334.

[16] Y. Dodis, N. Fazio, Public key broadcast encryption for stateless receivers, in: Proceedings of DRM 2002, LNCS, vol. 2696, Springer-Verlag, 2003, pp. 61–80.

[17] A. Fiat, M. Naor, Broadcast encryption, in: Proceedings of CRYPTO'93, LNCS, vol. 773, Springer-Verlag, 1994, pp. 480–491.

[18] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for finegrained access control of encrypted data, Proceedings of ACM CCS 2006, ACM Press, 2006, pp. 89–98.

[19] H. Ghodosi, J. Pieprzyk, R. Safavi-Naini, Dynamic threshold cryptosystems: a new scheme in group oriented cryptography, in: Proceedings of Pragocrypt 1996, CTU Publishing house, 1996, pp. 370–379.

[20] C. Gentry, B. Waters, Adaptive security in broadcast encryption systems (with short ciphertexts), in: Proceedings of EUROCRYPT 2009, LNCS, vol. 5479, Springer-Verlag, 2009, pp. 171–188.

[21] M.T. Goodrich, J.Z. Sun, R. Tamassia, Efficient tree-based revocation in groups of low-state devices, in: Proceedings of CRYPTO 2004, LNCS, vol. 3152, Springer-Verlag, 2004, pp. 511–527.

[22] H. Guo, Z. Li, Y. Mu, X. Zhang, Provably secure identity-based authenticated key agreement protocols with malicious private key generators, Information Sciences 181 (3) (2011) 628–647.

[23] D. Halevy, A. Shamir, The LSD broadcast encryption scheme, in: Proceedings of CRYPTO 2002, LNCS, vol. 2442, Springer-Verlag, 2002, pp. 47–60.

[24] X. Huang, W. Susilo, Y. Mu, W. Wu, Secure universal designated verifier signature without random oracles, International Journal of Information Security 7 (3) (2008) 171–183.

[25] J. Katz, N. Wang, Efficiency improvements for signature schemes with tight security reductions, Proceedings of ACM CCS'03, ACM Press, 2003, pp. 155–164.

[26] B. Libert, J.-J. Quisquater, Efficient revocation and threshold pairing based cryptosystems, Proceedings of the 22nd ACM PODC, ACM Press, 2003, pp. 163–171.

[27] B. Libert, M. Yung, Adaptively secure non-interactive threshold cryptosystems, in: Proceedings of ICALP 2011, LNCS, vol. 6756, Springer-Verlag, 2011, pp. 588–600.

[28] C.H. Lim, P.J. Lee, Directed signatures and application to threshold cryptosystems, in: Proceedings of Security Protocols 1997, LNCS, vol. 1189, Springer-Verlag, 1997, pp. 131–138.

[29] B. Qin, Q. Wu, L. Zhang, J. Domingo-Ferrer, Threshold public-key encryption with adaptive security and short ciphertexts, in: Proceedings of ICICS 2010, LNCS, vol. 6476, Springer-Verlag, 2010, pp. 62–76.

[30] A. Shamir, How to share a secret, Communications of the ACM 22 (1979) 612–613.

[31] A.T. Sherman, D.A. McGrew, Key establishment in large dynamic groups using one-way function trees, IEEE Transactions on Software Engineering 29 (5) (2003) 444–458.

[32] V. Shoup, ISO 18033-2: An Emerging Standard for Public-Key Encryption, Final Committee Draft (December 2004).

[33] D.M. Wallner, E.J. Harder, R.C. Agee, Key Management for Multicast: Issues and Architectures, IETF draft wallner-key (1997).

[34] C.K. Wong, M. Gouda, S. Lam, Secure group communications using key graphs, IEEE/ACM Transactions on Networking 8 (1) (2000) 16–30.

[35] Q. Wu, Y. Mu, W. Susilo, B. Qin, J. Domingo-Ferrer, Asymmetric group key agreement, in: Proceedings of EUROCRYPT 2009, LNCS, vol. 5479, Springer-Verlag, 2009, pp. 153–170.

[36] Q. Wu, B. Qin, L. Zhang, J. Domingo-Ferrer, O. Farràs, Bridging broadcast encryption and group key agreement, in: Proceedings of ASIACRYPT 2011, LNCS, vol. 7073, Springer-Verlag, 2011, pp. 143–160.

[37] H. Yuan, F. Zhang, X. Huang, Y. Mu, W. Susilo, L. Zhang, Certificateless threshold signature scheme from bilinear maps, Information Sciences 180 (23) (2010) 4714–4728.

[38] L. Zhang, F. Zhang, Q. Wu, J. Domingo-Ferrer, Simulatable certificateless two-party authenticated key agreement protocol, Information Sciences 180 (6) (2010) 1020–1030.

[39] M. Zhang, B. Yang, T. Takagi, Group-oriented setting's multisigncryption scheme with threshold designcryption, Information Sciences 181 (18) (2011) 4041–4050.