Threshold Public-Key Encryption with Adaptive Security and Short Ciphertexts

Bo Qin[1,3], Qianhong Wu[1,2], Lei Zhang[1], and Josep Domingo-Ferrer[1]

[1] Universitat Rovira i Virgili, Dept. of Comp. Eng. and Maths
UNESCO Chair in Data Privacy, Tarragona, Catalonia
{qianhong.wu,bo.qin,josep.domingo}@urv.cat
[2] Key Lab. of Aerospace Information Security and Trusted Computing
Ministry of Education, School of Computer, Wuhan University, China
[3] Dept. of Maths, School of Science, Xi'an University of Technology, China

Abstract. Threshold public-key encryption (TPKE) allows a set of users to decrypt a ciphertext if a given threshold of authorized users cooperate. Existing TPKE schemes suffer from either long ciphertexts with size linear in the number of authorized users or can only achieve non-adaptive security. A non-adaptive attacker is assumed to disclose her target attacking set of users even before the system parameters are published. The notion of non-adaptive security is too weak to capture the capacity of the attackers in the real world. In this paper, we bridge these gaps by proposing an efficient TPKE scheme with constant-size ciphertexts and adaptive security. Security is proven under the decision Bilinear Diffie-Hellman Exponentiation (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 also propose an efficient tradeoff between the key size and the ciphertext size, which gives the first TPKE scheme with adaptive security and sublinear-size public key, decryption keys and ciphertext.

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

1 Introduction

Threshold public-key encryption (TPKE) is a well-studied cryptographic primitive [9,17,23,13]. In TPKE, each of n users holds a decryption key corresponding to a public key; a sender 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. TPKE systems are applied to access control to sensitive information. In such scenarios, one cannot fully trust a single person but possibly a group of

M. Soriano, S. Qing, and J. López (Eds.): ICICS 2010, LNCS 6476, pp. 62–76, 2010.

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 any individual ballot. Other applications include key-escrow and any decryption procedure requiring an agreement of a number of trusted bodies.

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 decide 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 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 concentrate on TPKE systems with adaptive security and short ciphertexts which are essential features for TPKE schemes to be securely and efficiently deployed in practice. In particular, our contribution includes two folds.

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 setup. The attacker can adaptively query the decryption keys of users outside the committed set of the users 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 since the attacker's choice of which subset of the committed set to attack can be adaptive. By using the similar idea in [19], 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.

By exploiting pairings, we implement a TPKE scheme with constant-size ciphertext and semi-adaptive security. The security is proven under the decision 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 ciphertext. 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 provide an efficient tradeoff between ciphertext and key size, which yields the first TPKE scheme with adaptive security and sublinear-size public/decryption keys and ciphertexts.

1.2 Related Work

To cater for applications of controllable decryption, a number of notions have been proposed such as threshold public key encryption [13], identity-based threshold encryption/decryption [1,22], threshold public key cryptosystem [9], threshold broadcast encryption [11], dynamic threshold encryption [17,23], and *ad hoc* threshold encryption [12]. 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 number of other 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). 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 only if some users in the authorized set would like to cooperate and the number of cooperating authorized users is no less than 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.* 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 is usually very small in practice and $|\mathbb{R}|$ 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 [13]. However, as mentioned by the authors [13], their scheme has several limitations. Their proposal has a $\mathcal{O}(n)$-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 as an open problem to design a scheme with short ciphertext and adaptive security.

Several notions close to TPKE have been proposed in the literature. By setting the threshold to be 1, a TPKE scheme is indeed a broadcast encryption scheme [15]. 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 $\mathbb{R} \subseteq \{1, \cdots, n\}$ of users such that only and any users in $\mathbb{R}$ can decrypt the ciphertext. Fiat and Naor [15] were the first to formally explore broadcast encryption. Further improvements [20,21] reduced the decryption key size. Dodis and Fazio [14] extended the subtree difference method into a public-key broadcast system for a small size public key. Wallner *et al.* [28] and Wong [29] independently discovered the logical-tree-hierarchy scheme for group multicast. The parameters of the original schemes were improved in further work [8,10,26]. Boneh *et al.* [3] 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 tradeoff, they obtained a scheme with $O(\sqrt{n})$-size public keys, decryption keys, and ciphertexts. However, similarly to [13], they used a non-adaptive model of security. Other contributions [19, 5, 7] focused on stronger adaptive security in the sense that the attacker can adaptively corrupt users, as considered in this paper. Attribute-based encryption [6] 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 [16] 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. Section 2 recalls some background materials that will be used for the construction of our schemes. In Section 3, 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 4 proposes a basic secure TPKE scheme with small ciphertexts. Several variants are suggested in Section 5 with fully adaptive security and sublinear-size public/decryption keys and ciphertexts. Section 6 concludes the paper.

2 Preliminaries

2.1 Bilinear Pairings and Assumptions

Our schemes are implemented in bilinear pairing groups [4,18]. 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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 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 $e(g^x, g^y) = e(g, g)^{xy}$.

The security of the schemes that we propose in this paper relies on the decision n-BDHE problem. The corresponding decision n-BDHE assumption is shown to be sound by Boneh *et al.* [2] in the generic group model. This assumption has been widely followed up for cryptographic constructions (e.g., [3, 5, 19, 30]). We briefly review the decision n-BDHE assumption in $\mathbb{G}$ as follows.

Definition 1 (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 $\mathsf{AdvBDHE}_{\mathcal{A},n}(\lambda) = |\Pr[\mathcal{A} \ \ wins] - \frac{1}{2}|$. The Decision BDHE assumption states that, for any polynomial time probabilistic attacker $\mathcal{A}$, $\mathsf{AdvBDHE}_{\mathcal{A},n}(\lambda)$ is negligible in λ.

2.2 Shamir's Secret Sharing Scheme

Our system exploits the Shamir's (t, T)-threshold secret sharing scheme [25]. Let $\mathbb{Z}_p$ be a finite field with $p > T$ and $x \in \mathbb{Z}_p$ be the secret to be shared. A dealer picks a polynomial $f(\alpha)$ of degree at most $t - 1$ at random, whose free term is the secret x, that is, $f(0) = x$. The polynomial $f(\alpha)$ can be written as

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

where $a_1, \cdots, a_{t-1} \in \mathbb{Z}_p$ are randomly chosen. Each shareholder k is assigned a known index $k \in \{1, \cdots, T\}$ and the dealer privately sends to shareholder k a share $x_k = f(k)$. Then any t holders in $\mathbb{A} \subset \{1, \cdots, T\}$ can recover the secret $x = f(0)$ by interpolating their shares

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

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

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

If an attacker obtains at most $t-1$ shares, the shared secret x stays information-theoretically secure vs the attacker. That is, the attacker can get no information about x and, even if the attacker has unlimited computation power.

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

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

which can be computed with the knowledge of $g^x, g^{a_1}, \cdots, g^{a_{t-1}}$. Then any t holders can recover the secret $g^x = F(0)$ by interpolating 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 Threshold Public-Key Encryption

We review the model of TPKE systems and then formalize the security definitions in TPKE schemes motivated by [13]. We focus on standard TPKE systems where the threshold is determined by a trusted party, *i.e.* the dealer in our definition. Compared to the definition in [13], 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.

3.1 Modeling TPKE Systems

We begin by formally defining a TPKE system. Note that, for content distribution or any encryption of a large ciphertext, the current standard technique is the KEM-DEM methodology [27], 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$(MPK, \mathbb{R}, sk)$**.** This algorithm is run by a sender to distribute a session key to chosen users so that these can recover the session key only if at least t of them cooperate. It takes as input a recipient set $\mathbb{R} \subseteq$ consisting of the identities (or the public keys) of the chosen users, the TPKE master public key MPK, and a secret session key sk. If $|\mathbb{R}| \leq n$, it outputs a pair $\langle Hdr, sk\rangle$, where Hdr is called the header of the session key sk. Send $\langle Hdr, \mathbb{R}\rangle$ 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 $|\mathbb{R}| \leq n$, 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 $\Sigma = (\sigma_1, \cdots, \sigma_t)$ of t decrypted session key shares. It outputs the session key sk or $\perp$ representing an error in reconstruction of the session key.

3.2 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 2. (Correctness.) *A TPKE scheme is correct if the following holds: for all* $\mathbb{R}(t \leq |\mathbb{R}| \leq n)$*, all* $\mathbb{A} \subseteq \mathbb{R}(|\mathbb{A}| \geq t)$*,* $\pi \leftarrow \mathsf{Setup}(1^\lambda)$*,* $(MPK, msk) \leftarrow \mathsf{KeyGen}(\pi)$*,* $(UPK, udk) \leftarrow \mathsf{Join}(msk, ID)$ *for all identities* ID*,* $\langle Hdr, sk\rangle \leftarrow \mathsf{Encryption}(MPK, \mathbb{R}, sk)$*,* $\Sigma = \{\sigma | \sigma \leftarrow \mathsf{ShareDecrypt}(\mathbb{R}, ID, udk, Hdr, PK), ID \in \mathbb{A}\}$*, 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 dealer'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. 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 $\mathbb{R}^*$ is at most $t-1$. The challenger sets $\langle Hdr^*, sk_0\rangle \leftarrow \mathsf{Encryption}(MPK, \mathbb{R}^*, sk_0)$ 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 attacker $\mathcal{A}$.

Guess. 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

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

Definition 3. (Adaptive security.) *We say that a TPKE scheme is adaptively secure if for all polynomial time algorithms $\mathcal{A}$ we have that $Adv^{TPKE}_{\mathcal{A},n,t}(1^\lambda)$ 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 $\mathbb{R}^*$ of identities that it will attack in an Initialization phase before the Setup algorithm is run. This is the security definition that is used by recent TPKE systems [13]. Another useful security definition is referred to as *semi-adaptive* security. In this game the attacker must commit to a set $\bar{\mathbb{R}}$ of indices at the Initialization phase before the Setup stage. The attacker can query the decryption key for any user outside $\bar{\mathbb{R}}$. The attacker can also query for the decryption keys for some users in $\bar{\mathbb{R}}$ up to $t-1$ users. It has to choose a target group $\mathbb{R}^* \subseteq \bar{\mathbb{R}}$ for the challenge ciphertext, noting that at most $t-1$ authorized users haven been corrupted. A semi-adaptive attacker is weaker than an adaptive attacker, but it is stronger than a non-adaptive attacker since the attacker's choice of which users to attack can be adaptive.

3.3 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 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 [19] 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 [24], 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 $t, 2n$. 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 i-th bit of θ by θ_i.

Join(msk, ID_i)**.** Run $UPK'_i, udk'_i \leftarrow UKeyGen_{SA}(msk', ID_{2i-\theta_i})$, where $1 \leq i \leq n$. 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.

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

$$\mathbb{R}_0 \leftarrow \{ID_{2i-\zeta_i} : i \in \{1, \cdots, |\mathbb{R}|\}\},$$

$$\langle Hdr_0, sk\rangle \leftarrow Encryption_{SA}(MPK, \mathbb{R}_0, sk);$$

$$\mathbb{R}_1 \leftarrow \{ID_{2i-(1-\zeta_i)} : i \in \{1, \cdots, |\mathbb{R}|\}\},$$

$$\langle Hdr_1, sk\rangle \leftarrow Encryption_{SA}(MPK, \mathbb{R}_1, sk).$$

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 $\Sigma = (\sigma_1, \cdots, \sigma_t)$.

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 the user only one of the two. 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 [19] 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 any decryption service to the attacker in either case. Hence, in the context of TPKE systems, 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. Hence, 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).

Theorem 1. *Let $\mathcal{A}$ be an adaptive attacker against $\mathtt{TPKE}_A$. Then, there exist algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$, each running in about the same time as $\mathcal{A}$, such that*

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

Proof. It is omitted to avoid repetition. □

4 Basic TPKE with Short Ciphertext

In this section, we propose a basic TPKE construction. The construction is based on Shamir's secret sharing scheme [25] and the recent Gentry-Waters broadcast scheme [19]. The basic scheme has constant-size ciphertexts and is proven to be secure without using random oracles.

- **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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficient non-degenerate bilinear map. Let $h_1, \cdots, h_n$ be randomly chosen from $\mathbb{G}$. The system parameters are $\pi = (\Upsilon, g, h_1, \cdots, h_n, t, n)$. In the following, we assume that each user is uniquely identified by an index $i \in \{1, 2, \cdots, n\}$. This can be implemented by ordering the users by the order in which they join the system.
- **KeyGen.** Randomly select $x, a_1, \cdots, a_{t-1} \in \mathbb{Z}_p$ and compute

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

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

$$msk = \langle x, a_1, \cdots, a_{t-1} \rangle.$$

- **Join.** Let the i-th user want to join the system. The dealer generates a secret polynomial

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

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 as

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

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 $sk = e(g, g)^{x\gamma}$ 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

$$e(g^{f(i)}h_i^{r_i} \prod_{j \in \mathbb{R} \setminus \{i\}} h_j^r, c_1)e(g^{-r}, c_2) = e(g^{f(i)}(\prod_{j \in \mathbb{R}} h_j)^r, g^{\gamma})e(g^{-r}, (\prod_{j \in \mathbb{R}} h_j)^{\gamma})$$

$$= e(g, g)^{f(i)\gamma} \stackrel{\mathtt{Def}}{=} \sigma_i.$$

Note that decryption is non-interactive.

- **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 $sk = (g, g)^{x\gamma} = e(g, g)^{f(0)\gamma}$ by interpolating 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.

As to security, we have the following theorem. The proof is provided in the full version of the paper.

Theorem 2. *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 n-BDHE assumption with advantage ϵ' in time τ', where $\epsilon' \geq \frac{1}{C_n^t}\varepsilon$, $\tau' \leq \tau + \mathcal{O}(1)\tau_{Pair} + \mathcal{O}(n^2)\tau_{Exp}$, where τ_{Exp} denotes the overhead to compute a pairing, and τ_{Exp} denotes the time complexity to compute one exponentiation without differentiating exponentiations in different groups.*

One may note that we have a reduction loss by a factor $\frac{1}{C_n^t}$. However, since t is usually very small, it is reasonable to assume that $t \leq \mathcal{O}(poly(\log \lambda))$. The reduction loss is $\frac{1}{poly(\lambda)}$ even if n is a polynomial in λ.

5 Extensions

5.1 Shortening System Parameters

In the basic construction, we need $h_1, \cdots, h_n$ as system parameters. One may observe that $h_1, \cdots, h_n$ 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 $h_1, \cdots, h_n$ 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.

5.2 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 , the basic scheme and its above variant can be readily improved to meet fully-adaptive security, at a cost of doubling ciphertexts.

5.3 Tradeoff 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 $\mathcal{O}(1)$ size. However, the decryption key of each user consists of $\mathcal{O}(n)$ elements. In the following, we illustrate an efficient tradeoff between the decryption keys and ciphertexts.

Let $n = n_1^2$ and divide the maximal receiver group $\{1, \cdots, n\}$ 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 $\mathbb{R} \subseteq \{1, \cdots, n\}$. 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 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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 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 $x_1, \cdots, x_{n_1}, a_1, \cdots, a_{t-1} \in \mathbb{Z}_p$ and compute

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

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

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

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

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

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:

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

Privately send udk_i to user i and set user i's public key UPK_i as i.

– **Encrypt.** For a receiver set $\mathbb{R}$, randomly pick γ in $\mathbb{Z}_p$ and compute

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

where $u = 0, \cdots, n_1 - 1; \mathbb{R}_u = \mathbb{R} \cap \{un_1 + 1, \cdots, un_1 + n_1\}$. 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

$$e(g^{f(i)}H(u,v)^{r_i} \prod_{k \in \mathbb{R}_u \setminus \{v\}} H(u,k)^{r_i}, c_0) e(g^{-r_i}, c_{u+1})$$

$$= e(g^{f(i)}(\prod_{k \in \mathbb{R}_u} H(u,k))^{r}, g^{\gamma}) e(g^{-r}, (\prod_{k \in \mathbb{R}_u} H(u,k))^{\gamma}) = e(g,g)^{f(i)\gamma} \stackrel{\mathtt{Def}}{=} \sigma_i.$$

– **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 $sk = (g,g)^{x\gamma} = e(g,g)^{f(0)\gamma}$ by interpolating 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.

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

6 Conclusion

In this paper, we 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. The security is 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 also proposed an efficient tradeoff between the key size and the ciphertext size. The size of the public key, the decryption keys and the ciphertexts in the scheme resulting from the tradeoff is sublinear in the number of authorized users.

Acknowledgments

This work is partly supported by the Spanish Government under projects TSI2007-65406-C03-01 "E-AEGIS", TIN2009-11689 "RIPUP", "eVerification" TSI-020100-2009-720, SeCloud TSI-020302-2010-153 and CO-NSOLIDER 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 and 61003214, and by the Fundamental Research Funds for the Central Universities of China through project 3103004. The fourth 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. Baek, J., Zheng, Y.: Identity-Based Threshold Decryption. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 262–276. Springer, Heidelberg (2004)
2. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
3. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
4. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
5. Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
6. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Computer Society, Los Alamitos (2007)
7. Boneh, D., Waters, B.: A Fully Collusion Resistant Broadcast, Trace, and Revoke System. In: Juels, A., Wright, R.-N., De Capitani di, V.S. (eds.) ACM CCS 2006, pp. 211–220. ACM Press, New York (2006)
8. Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., Pinkas, B.: Multicast Security: A Taxonomy and some Efficient Constructions. In: IEEE INFOCOM 1999, New York, NY, vol. 2, pp. 708–716 (1999)
9. Canetti, R., Goldwasser, S.: An Efficient Threshold Public Key Cryptosystem Secure Against Adaptive Chosen Ciphertext Attack. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 90–106. Springer, Heidelberg (1999)
10. Canetti, R., Malkin, T., Nissim, K.: Efficient Communication-storage Tradeoffs for Multicast Encryption. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 459–474. Springer, Heidelberg (1999)
11. Daza, V., Herranz, J., Morillo, P., Ráfols, C.: CCA2-Secure Threshold Broadcast Encryption with Shorter Ciphertexts. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 35–50. Springer, Heidelberg (2007)

12. Daza, V., Herranz, J., Morillo, P., Ràfols, C.: Ad-hoc Threshold Broadcast Encryption with Shorter Ciphertexts. Electronic Notes in Theoretical Computer Science 192(22), 3–5 (2008)
13. Delerablée, C., Pointcheval, D.: Dynamic Threshold Public-Key Encryption. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 317–334. Springer, Heidelberg (2008)
14. Dodis, Y., Fazio, N.: Public Key Broadcast Encryption for Stateless Receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)
15. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
16. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data. In: ACM CCS 2006, pp. 89–98. ACM Press, New York (2006)
17. Ghodosi, H., Pieprzyk, J., Safavi-Naini, R.: Dynamic Threshold Cryptosystems: A New Scheme in Group Oriented Cryptography. In: Proceedings of Pragocrypt 1996, pp. 370–379. CTU Publishing house (1996)
18. Galbraith, S.D., Rotger, V.: Easy Decision Diffie-Hellman Groups. Journal of Computation and Mathematics 7, 201–218 (2004)
19. Gentry, C., Waters, B.: Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2010)
20. Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient Tree-Based Revocation in Groups of Low-State Devices. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004)
21. Halevy, D., Shamir, A.: The LSD Broadcast Encryption Scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
22. Libert, B., Quisquater, J.-J.: Efficient Revocation and Threshold Pairing Based Cryptosystems. In: 22nd ACM PODC, pp. 163–171. ACM Press, New York (2003)
23. Lim, C.H., Lee, P.J.: Directed Signatures and Application to Threshold Cryptosystems. In: Lomas, M. (ed.) Security Protocols 1996. LNCS, vol. 1189, pp. 131–138. Springer, Heidelberg (1997)
24. Katz, J., Wang, N.: Efficiency Improvements for Signature Schemes with Tight Security Reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003, pp. 155–164. ACM Press, New York (2003)
25. Shamir, A.: How to Share a Secret. Communications of the ACM 22, 612–613 (1979)
26. Sherman, A.T., McGrew, D.A.: Key Establishment in Large Dynamic Groups using One-way Function Trees. IEEE Transactions on Software Engineering 29(5), 444–458 (2003)
27. Shoup, V.: ISO 18033-2: An Emerging Standard for Public-Key Encryption, Final Committee Draft (December 2004)
28. Wallner, D.M., Harder, E.J., Agee, R.C.: Key Management for Multicast: Issues and Architectures. IETF draft wallner-key (1997)
29. Wong, C.K., Gouda, M., Lam, S.: Secure Group Communications using Key Graphs. IEEE/ACM Transactions on Networking 8(1), 16–30 (2000)
30. Wu, Q., Mu, Y., Susilo, W., Qin, B., Domingo-Ferrer, J.: Asymmetric Group Key Agreement. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 153–170. Springer, Heidelberg (2010)