Fair threshold decryption with semi-trusted third parties

Jeongdae Hong and Jinil Kim

School of Computer Science and Engineering,
Seoul National University,
Gwanak-gu, Seoul 151-742, South Korea
E-mail: jdhong@theory.snu.ac.kr
E-mail: jikim@theory.snu.ac.kr

Jihye Kim

ISaC and Department of Mathematical Sciences,
Seoul National University,
Gwanak-gu, Seoul 151-742, South Korea
E-mail: jihyek@snu.ac.kr

Matthew K. Franklin

Department of Computer Science,
University of California Davis,
One Shields Avenue, Davis, CA 95616, USA
E-mail: franklin@cs.ucdavis.edu

Kunsoo Park*

School of Computer Science and Engineering,
Seoul National University,
Gwanak-gu, Seoul 151-742, South Korea
E-mail: kpark@theory.snu.ac.kr
*Corresponding author

Abstract: A threshold decryption scheme is a multi-party public key cryptosystem that allows any sufficiently large subset of participants to decrypt a ciphertext, but disallows the decryption otherwise. Many threshold cryptographic schemes have been proposed so far, but fairness is not generally considered in this earlier work. In this paper, we present fair threshold decryption schemes, where either all of the participants can decrypt or none of them can. Our solutions employ semi-trusted third parties (STTP) and offline semi-trusted third parties (OTTP) previously used for fair exchange. We consider a number of variants of our schemes to address realistic alternative trust scenarios. Although we describe our schemes using a simple hashed version of ElGamal encryption, our methods generalise to other threshold decryption schemes and threshold signature schemes as well.

Keywords: threshold decryption; fairness; semi-trusted third parties; STTP; optimistic protocol; ElGamal encryption.

Reference to this paper should be made as follows: Hong, J., Kim, J., Kim, J., Franklin, M.K. and Park, K. (2010) 'Fair threshold decryption with semi-trusted third parties', *Int. J. Applied Cryptography*, Vol. 2, No. 2, pp.139–153.

Biographical notes: Jeongdae Hong is currently a Military Officer in the Republic of Korea Army. He received his BS in Weapons and Mechanical Engineering from Korea Military Academy and at the same time was commissioned as a Military Officer in 1993. He also received his MS and PhD in Computer Science and Engineering from Seoul National University in 2005 and 2010, respectively. His research interests are in all areas of cryptography with a specific interest in military security.

Jinil Kim is presently pursuing his PhD at the Department of Computer Science and Engineering in Seoul National University. He received his BS in Electrical Engineering and MS in Computer Science and Engineering from Seoul National University in 2005 and 2007, respectively. His primary interests are secure multi-party computations, threshold cryptography and efficient implementations of them.

Jihye Kim is currently a Postdoctoral Researcher of Information Security and Cryptology Research Center at Seoul National University. She received her BS and MS in School of Computer Science and Engineering at Seoul National University in 1999 and 2003, respectively, and PhD in Computer Sciences from University of California, Irvine in 2008. Her research interests are network security, applied cryptography and fault-tolerant and distributed computing.

Matt Franklin is currently a Professor in the Computer Science Department at UC Davis. Before this, he was a Research Scientist at AT&T (Bell) Labs and at Xerox PARC. He received his PhD in Computer Science from Columbia University in 1994. His research interests are in cryptography and distributed computing.

Kunsoo Park received his BS and MS in Computer Engineering from Seoul National University in 1983 and 1985, respectively, and PhD in Computer Science from Columbia University in 1991. From 1991 to 1993, he was a Lecturer at King's College, University of London. He is currently a Professor in the Department of Computer Science and Engineering at Seoul National University. His research interests include design and analysis of algorithms, cryptography, and bioinformatics.

1 Introduction

A threshold decryption scheme is a multi-party public key cryptosystem that allows any sufficiently large subset of participants to decrypt a ciphertext, but disallows the decryption otherwise. In a threshold decryption scheme, a secret key is typically split into secret key shares for the participants using a threshold secret sharing scheme. When a sufficiently large subset of participants wants to decrypt a ciphertext, each party computes a partially decrypted value using its secret key share. Any party who collects sufficiently many partially decrypted values can decrypt.

In this paper, we focus on *fairness* of threshold decryption, which is not easy to achieve without a fully trusted third party (TTP). In many scenarios, it is very desirable that all participants in a threshold decryption procedure should receive the correct decrypted plaintext simultaneously, even when those participants are mutually mistrustful. In a stand-alone setting, where valuable data is encrypted, it is a security problem if some cheating participants to the threshold decryption procedure can recover the plaintext while the other participants cannot. In a more complex setting, where the threshold decryption is part of a larger secure multi-party protocol (e.g., Kissner and Song, 2005), the security of the overall protocol may be compromised unless ciphertexts can be decrypted with fairness. Although we focus on threshold decryption in this work, essentially, all of our methods hold for the case of threshold signatures as well (this is also an important primitive in many scenarios).

The previous threshold decryption schemes sometimes include a party called *combiner*, who collects the decryption shares and computes the plaintext. With respect to fairness, *combiner* would need to be a TTP since it obtains the plaintext earlier than others. However, TTP is typically undesirable because all of the parties should totally trust it. Alternatively, each party in the threshold decryption can become a *combiner* by himself if he knows the combining algorithm and all the decryption shares are exchanged among the parties. In that case, however, unfairness occurs when the first party who obtains all of the decryption shares quits the protocol without sending his decryption share. Even worse, a malicious party may obtain all decryption shares exclusively by repeating decryptions without sending his decryption share. Robust threshold decryption can be helpful here, but it does not by itself yield fairness.

We employ semi-trusted third parties (STTP) and offline STTP (OTTP).[1] The STTP is an additional participant that follows the prescribed protocol correctly, while recording all communication in an attempt to learn something more about plaintexts ('semi-honest' or 'honest-but-curious' adversary). The STTP is an online participant, since it connects to all the parties before the protocol starts, and communicates with them during the protocol even when all the parties are honest.

Our first solution uses a single STTP to achieve fair threshold decryption. However, in practice, the reconstructing parties may fail to agree on a mutually satisfactory STTP. For example, imagine a situation where the parties of organisation **A** trust only STTP α, while the parties of organisation **B** trust only STTP β. To cover this kind of situation, we give a second solution that uses multiple 'weak' STTPs to achieve fair threshold decryption. A 'weak' STTP works semi-honestly for all the parties that trust it, but may work maliciously for the other parties. We present fair threshold decryption schemes for two natural variants of the multiple weak STTPs setting.

Our third solution uses a single OTTP to achieve fair threshold decryption. The OTTP is a semi-honest additional participant that is not involved in the protocol unless one or more of the reconstructing parties crashes or attempts to cheat. The OTTP is an offline participant, since it does not connect to any of the parties during normal protocol execution. This kind of protocol is often called 'optimistic', since troubles are resolved afterwards rather than prevented beforehand.

For all of our solutions, no information leaks to the sender about the decryption policy (i.e., encryption looks like ordinary public key encryption). This has always been an essential design principle for threshold cryptography, and

it rules out simple approaches where the sender splits his message and encrypts using multiple keys.

1.1 Related work

Many threshold cryptographic schemes have been proposed so far (Desmedt and Frankel, 1989; Santis et al., 1994; Gennaro et al., 1996, 2000, 2001, 2008; Shoup, 2000; Fouque et al., 2000), but fairness is not generally considered in this earlier work. Cleve (1986) showed the impossibility of completely fair protocols without an honest majority for arbitrary functions, but Gordon et al. (2008) reopened the question for specific functions of interest. Particularly, if $t < n/3$ where n is the total number of parties and t is the number of corrupted parties, fairness for secure multi-party computation (MPC) protocols can be achieved without information-theoretic or computational assumptions (Ben-Or et al., 1988; Chaum et al., 1988; Goldreich et al., 1987; Goldreich, 2000). If $n/3 \le t < n/2$, a broadcast channel is necessary to achieve fairness (Rabin and Ben-Or, 1989; Goldreich et al., 1987; Goldreich, 2000). Our consideration is on the general case where t can be any value between 1 and n. The 'gradual release' paradigm can give a relaxation of complete fairness that is useful in many contexts (see e.g., Blum, 1983; Luby et al., 1983; Ben-Or et al., 1990; Boneh and Naor, 2000; Boudot et al., 2001; Pinkas, 2003; Garay et al., 2004, 2006). Online semi-trusted parties have been used for fair exchange and related functions (e.g., Franklin and Reiter, 1997; Zhou and Gollmann, 1997; Cox et al., 1995). Offline semi-trusted parties have been used for 'optimistic' fair exchange (e.g., Asokan et al., 1997, 2000; Bao et al., 1998; Zhou et al., 2000; Boneh et al., 2003; Dodis and Reyzin , 2003; Dodis et al., 2007) and two-party optimistic fair secure computation (Cachin and Camenisch, 2000). Lindell (2008) considers a related setting in which the offline semi-trusted party is replaced with a legal infrastructure that respects digital signatures. Lepinski et al. (2004) used physical assumptions to realise fair secure function evaluation (SFE), which includes fair MPC as a special case.

The rest of the paper is organised as follows. Section 2 gives some preliminary cryptographic background. We present security models and definitions in Section 3. We present fair threshold decryption with (online) STTP (both strong and multiple weak) in Section 4 and optimistic fair threshold decryption with OTTP in Section 5.

2 Cryptographic background

In this section, we briefly review the scenario of threshold decryption and a threshold version of Hash-ElGamal cryptosystem.

2.1 Threshold decryption

The scenario of $(t+1,\ell)$ – threshold decryption is as follows. There are a dealer, ℓ shareholders who can participate in the decryption, and a combiner who actually decrypts the ciphertext. The dealer initialises a public key/secret key pair of the underlying non-threshold cryptosystem and splits the secret key into ℓ *secret key shares*. Each shareholder P_i keeps the corresponding secret key share s_i privately. Anyone can encrypt a message by the public key. When a group of $t+1$ shareholders wants to decrypt a ciphertext, each shareholder computes a partially decrypted value called *decryption share* by its secret key share. Then, the combiner collects $t+1$ decryption shares and obtains the plaintext by the combining algorithm. Throughout this paper, we use the following notations:

- *secret key share* – denotes a piece of the secret key shared by the secret sharing scheme
- *decryption share* – denotes a partially decrypted value of a ciphertext by a secret key share
- *reconstruction group* – denotes a group of $t+1$ parties participating in the decryption process.

2.2 Threshold version of Hash-ElGamal cryptosystem

Throughout the paper, we use the threshold version of Hash-ElGamal cryptosystem (ElGamal, 1985), which allows us to focus on the main ideas of our schemes and the security arguments introduced by the fairness.

Setup

Let G be a multiplicative group of large prime order q. Let H be a Hash function from G to plaintext-length bitstrings. Let g be a generator of G.

Key generation

A dealer does the following:

1 chooses a secret key $SK = x \in_R [1..q-1]$ and computes public key $PK = g^x$

2 picks a random polynomial $f(\cdot)$ with degree t for Shamir's (1979) secret sharing scheme whose coefficients are picked in $[0..q-1]$ and $f(0) = x$

3 for all $1 \le i \le \ell$, computes *secret key share* $s_i = f(i) \bmod q$, verification key $VK_i = g^{s_i}$, sends s_i to party P_i, and publishes $g, PK, \{(i, VK_i)\}_{1 \le i \le \ell}$.

Encryption

To encrypt a message m, one randomly chooses $r \in_R [1..q-1]$ and computes $E(m) = \left(g^r, m \oplus H\left(g^{rx}\right)\right)$.

Threshold decryption

Let $E(m)=(u,v)$, and let $S\subseteq[1..\ell]$ be a reconstruction group with $|S|=t+1$. For each $i\in S, P_i$ sends its *decryption share* $w_i=g^{rs_i}$ to the combiner. The combiner computes $g^{rx}=\prod_{i\in S}(w_i)^{\lambda_i}$, where $\{\lambda_i\}_{i\in S}$ are the appropriate Lagrange coefficients:

$$\left(\text{i.e., } \lambda_i=\prod_{b\in S\setminus\{i\}}\frac{i}{b-i}\right).$$

Then the combiner computes $v\oplus H\left(g^{rx}\right)=m$.

Robust threshold decryption

P_i also sends to the combiner a zk-proof of equality of discrete logarithms that $disc\log_g\left(VK_i\right)=disc\log_{g^r}\left(w_i\right)$. The combiner rejects the decryption share from P_i unless this zk-proof is good.

This threshold decryption scheme is semantically secure against a chosen plaintext attack if H is modelled as a random oracle, and if the computational Diffie-Hellman problem (CDH) is hard in G (where the chosen plaintext attack is performed by an adversary that can see up to t shares of the decryption key). The robust threshold decryption scheme has similar security if (in addition) the proof of equality of discrete logarithms is sound, complete and zero knowledge.

3 Security models and definitions

In this section, we introduce members of our scenarios and define *fairness* of threshold decryption.

3.1 Members and security models

The members of our fair threshold decryption scheme consist of a dealer D, ℓ shareholders and additional STTP.

- *Dealer*. A dealer D initialises the scheme as in the usual threshold decryption. If desired, a distributed key generation protocol can replace this trusted dealer using standard methods (e.g., Gennaro et al., 2007).
- *Shareholder*. A shareholder is a legal member with a secret key share who can participate in the decryption. We assume that each shareholder works in a malicious model. That is, it may arbitrarily deviate from a specified protocol. It may refuse to participate in the protocol or abort the protocol prematurely.
- *Strong STTP*. A strong STTP is an STTP trusted to work semi-honestly by all the shareholders. Only one strong STTP suffices in our fair threshold decryption scheme.
- *Weak STTP*. A weak STTP is an extended notion of STTP which is trusted to work semi-honestly by some of the shareholders but not trusted by other shareholders. To all the shareholders who trust it, it faithfully works in semi-honest model, while it can work maliciously to other shareholders. We assume that every weak STTP works semi-honestly to other weak STTPs regardless of trust relationship between shareholders and weak STTPs. In our schemes with multiple weak STTPs, several weak STTPs work together like one strong STTP.
- *Offline STTP (OTTP)*. An offline STTP is an STTP trusted by all the shareholders but it does not attend the protocol if all the shareholders behave honestly.

Every online (strong or weak) STTP has its secret key share distributed privately during the key generation and can compute its decryption share like a shareholder.

3.1.1 Communication model

We assume that every pair of participants has a private and reliable communication channel connecting them. This includes all combinations of shareholder-to-shareholder, shareholder-to-STTP, and STTP-to-STTP communication.

3.2 Formal definition of fairness

We define fairness of $(t+1,\ell)$-threshold decryption as follows.

Definition 1 (Fairness of threshold decryption): If any shareholder P_i decrypts a ciphertext, then there exists at least one reconstruction group S with $P_i\in S$ and $|S|=t+1$ such that all the shareholders of S can get the plaintext.

In the above definition, fairness means that all the shareholders of S can obtain the plaintext or no shareholder can. Any shareholder out of S should not be able to decrypt the ciphertext unless some shareholders of S send the plaintext to it. This requirement is important for applications in which the plaintext should be kept secret among the participants (e.g., Kissner and Song, 2005).

Our definition of fairness implies that any fair threshold decryption scheme should be robust against such an attack whereby malicious shareholders initiate two or more reconstruction protocols for the same ciphertext (e.g., with disjoint subsets of honest shareholders), aborting so that no honest shareholder succeeds in any single reconstruction effort, while the malicious shareholders can decrypt by combining honest decryption shares from all of the reconstruction efforts.

When the number of corrupted shareholders is less than $(t+1)/2$ (i.e., majority of any reconstruction group is honest), fairness can be achieved by general results of fair MPC (Ben-Or et al., 1988; Chaum et al., 1988; Goldreich et al., 1987; Goldreich, 2000). We employ STTPs to cover the

general case where up to t shareholders are corrupted. The following definition states fairness of threshold decryption when STTPs are involved in.

Definition 2 (Fairness of threshold decryption with STTPs): A threshold decryption with STTPs is fair if all the shareholders achieve fairness in Definition 1 with the help of STTPs, while no STTP can learn anything about the decrypted message in polynomial time.

4 Fair threshold decryption with (online) STTP

4.1 Security notions

We formally define two security notions, *STTP-assistance* and *STTP-obliviousness* by challenge-adversary games. Informally, *STTP-assistance* means that if any one of STTPs does not contribute its decryption share, no coalition of shareholders can decrypt any ciphertext even with the secret key shares of the other STTPs. And *STTP-obliviousness* means that no STTP, even with t secret key shares of shareholders, can learn anything about the decrypted message during polynomial number of decryptions on any ciphertexts.

4.1.1 STTP-assistance

We say that a threshold scheme satisfies STTP-assistance if any polynomially bounded adversary $\mathcal{A}$ cannot win the following game with non-negligible probability. The game proceeds between $\mathcal{A}$ and a challenger $\mathcal{CH}$ where there are k STTPs:

1 $\mathcal{CH}$ runs setup and key generation algorithms taking a security parameter. $\mathcal{CH}$ gives $\mathcal{A}$ the resulting common parameters.

2 $\mathcal{A}$ receives all the secret key shares of ℓ shareholders and $k-1$ secret key shares of the STTPs from $\mathcal{CH}$.

3 A adaptively makes a polynomial number of queries to $\mathcal{CH}$ on any messages. For each message M,$\mathcal{CH}$ generates encryption C of M and responds with C and the corresponding decryption share of the remaining STTP with zk-proof.

4 A selects two target messages (M_0, M_1). $\mathcal{CH}$ picks one message M_b by selecting a random bit $b \leftarrow \{0,1\}$ and sends a ciphertext C_b of M_b to $\mathcal{A}$.

5 Repeat Step 3.

6 $\mathcal{A}$ outputs b' (and wins if $b' = b$).

4.1.2 STTP-obliviousness

We say that a threshold scheme with STTP satisfies STTP-obliviousness if it satisfies any polynomially bounded adversary $\mathcal{A}$ cannot win the following game with non-negligible probability. The game proceeds between $\mathcal{A}$ and a challenger $\mathcal{CH}$ where there are k STTPs:

1 $\mathcal{CH}$ runs setup and key generation algorithms taking a security parameter. $\mathcal{CH}$ gives $\mathcal{A}$ the resulting common parameters.

2 $\mathcal{A}$ is given all the secret key shares of k STTPs and t secret key shares of shareholders.[2]

3 $\mathcal{A}$ adaptively makes a polynomial number of queries to $\mathcal{CH}$ on (M,S) where $s \subseteq [1..\ell], |S| = t+1$. For each (M,S), $\mathcal{CH}$ generates encryption C of M, and responds with C and the corresponding decryption shares with zk-proofs following the protocol on behalf of the shareholders of S.

4 A selects two target messages (M_0, M_1). $\mathcal{CH}$ picks one message M_b by selecting a random bit $b \leftarrow \{0,1\}$ and sends a ciphertext C_b of M_b to $\mathcal{A}$. $\mathcal{CH}$ simulates executions of all the shareholders so that $\mathcal{A}$ can participate in the decryption of C_b on behalf of the STTPs.

5 Repeat Step 3.

6 A outputs b' (and wins if $b' = b$).

4.2 Strong STTP fair threshold Hash-ElGamal

4.2.1 Description

The main idea of this scheme is that the secret key is split into $\ell+1$ secret key shares instead of ℓ secret key shares so that a secret key share is assigned for the strong STTP. However, we define the secret key share of the strong STTP as a special one, thus even more than $t+1$ shareholders cannot decrypt a ciphertext without the decryption share of the strong STTP. The strong STTP can *trigger* the decryption by sending its decryption share after the shareholders of S exchange their decryption shares with one another. This scenario is applicable to all schemes which follow the general threshold decryption scenario. Figure 1 shows this scenario graphically. The details of the protocol are as follows.

Key generation

D chooses $x, R \in_R [1..q-1]$, and computes $PK = g^x$, $VK_{STTP} = g^R$. D picks an otherwise random degree t polynomial $f(\cdot)$ with coefficients in $[0..q-1]$ such that $f(0) = (x - R) \bmod q$. Then for all $i (1 \le i \le \ell)$, D computes $s_i = f_i \bmod q$, $VK_i = g^{s_i}$, and sends s_i to shareholder P_i. D sends $s_{STTP} = R$ to the STTP. Lastly, D publishes $g, PK, VK_{STTP}, \{(i, VK_i)\}_{1 \le i \le \ell}$.

Strong STTP fair threshold decryption

Let $E(m)=(u,v)$, and let $S\subseteq[1..\ell],|S|=t-1$. The STTP do not have to know (u,v) or S.

For each $i,j\in S$, P_i sends $w_i=u^{s_i}$ and a zk-proof that $disc\log_g(VK_i)=disc\log_u(w_i)$ to P_j. If P_i succeeds in verifying $t+1$ decryption shares (including her own), then P_i sends a $(READY,S,(u,v))$ signal to the STTP. When the STTP receives consistent and well-formed $READY$ signals from at least $t+1$ shareholders, the STTP sends to each $READY$ signaller $w_{STTP}=u^{s_{STTP}}$ and a zk-proof that $disc\log_g(VK_{STTP})=disc\log_u(w_{STTP})$. If P_i was a $READY$ signaller, and if the STTP sent a valid decryption share, then P_i can now decrypt, since $g^{rx}=w_{STTP}\cdot\prod_{i\in S}w_i^{\lambda_i}$ where $\lambda_i=\prod_{b\in S\setminus\{i\}}\frac{i}{b-i}$.

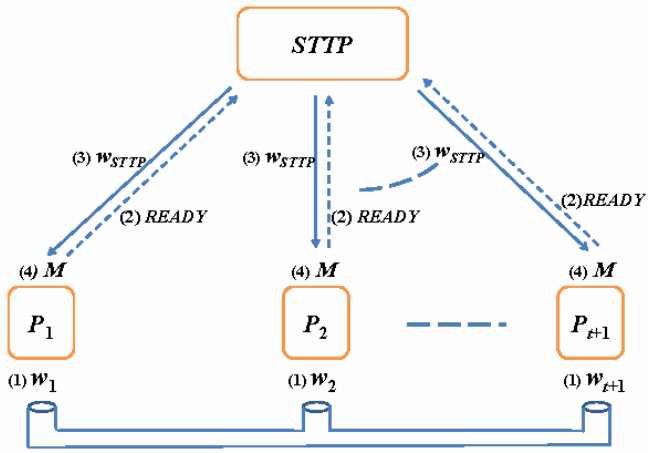
Non-threshold access structure

Instead of viewing this as a $(t+1)$-out-of-ℓ threshold decryption scheme with ℓ shareholders $(P_1,...,P_\ell)$ and a separate STTP, we can view it as a 'non-threshold' access structure with $\ell+1$ shareholders $(STTP,P_1,...,P_\ell)$. Specifically, the access structure Γ' for successful decryption is $STTP\wedge\Gamma_{t+1,\ell}$, where $\Gamma_{t+1,\ell}$ is the $(t+1)$-out-of-ℓ threshold access structure on $(P_1,...,P_\ell)$. Strong STTP fair threshold decryption is possible whenever this non-threshold access structure Γ' can be realised.

Communication cost

Each shareholder's communication cost is $O(t)$ and the total communication cost is $O(t^2)$ except the key generation step. These costs are necessary for each shareholder to collect decryption shares from $t+1$ shareholders.

Figure 1 Fair threshold decryption with one strong STTP (see online version for colours)



4.2.2 Security proof of strong-STTP version

Now we prove that the strong-STTP protocol satisfies *STTP-assistance*, *STTP-obliviousness*, and *fairness* from the CDH assumption.

Theorem 1 (STTP-assistance): Under the CDH assumption, the strong-STTP protocol satisfies STTP-assistance in the random oracle model.

Proof: Let us assume the existence of an adversary $\mathcal{A}$ able to break STTP-assistance. We now describe that a challenger $\mathcal{CH}$ can solve the CDH problem using the adversary $\mathcal{A}$ with non-negligible probability. When starting the STTP-assistance game, $\mathcal{CH}$ gets an instance of CDH problem (g,g^x,g^y) whose goal is computing g^{xy}.

In Step 1, $\mathcal{CH}$ chooses a random polynomial $f(\cdot)$ with degree t. He simulates the strong-STTP protocol with initialising $PK=g^x,s_1=f(1),...,s_\ell=f(\ell)$. Then, s_{STTP} is automatically chosen as $s_{STTP}=x-f(0)$ but $\mathcal{CH}$ knows neither x nor s_{STTP}. The verification keys are easily computed by $VK_1=g^{s_1},...,VK_\ell=g^{s_\ell}$, $VK_{STTP}=g^x/g^{f(0)}$. $\mathcal{CH}$ sends common parameters $(PK,VK_1,...,VK_\ell,VK_{STTP})$ to $\mathcal{A}$.

In Step 2, $\mathcal{CH}$ sends $(s_1,...,s_\ell)$ to $\mathcal{A}$. $\mathcal{A}$ cannot distinguish these from a set of normal parameters because all of them are valid.

In Step 3, for each query M of $\mathcal{A},\mathcal{CH}$ generates a ciphertext:

$$Enc(M)=\left(g^r,M\oplus H\left(g^{rx}\right)\right)$$

where

$$r\in_R[0..q-1]$$

and responds with $Enc(M),w_{STTP}=(VK_{STTP})^r$ with the corresponding simulated zk-proof.

In Step 4, for given two messages M_0,M_1 from $\mathcal{A},\mathcal{CH}$ picks one message M_b by selecting a random bit $b\leftarrow\{0,1\}$ and generates a dummy ciphertext:

$$\hat{E}nc(M_b)=\left(g^y,M^b\oplus h\right)$$

where $h\in_R[0..q-1]$. $\mathcal{CH}$ sends $\hat{E}nc(M_b)$ to $\mathcal{A}$.

In Step 5, for every query $M,\mathcal{CH}$ works the same way as in Step 3. Finally, $\mathcal{A}$ outputs b' such that $\Pr[b'=b]$ is non-negligibly higher than 1/2.

The success probability of $\mathcal{CH}$ using $\mathcal{A}$ is analysed as follows. Let Q be the set of queries which $\mathcal{A}$ has asked to the random oracle during the execution, and the success probability of $\mathcal{A}$ be $\Pr[\mathcal{A}_{success}]=1/2+Adv$, where Adv is the non-negligible advantage of $\mathcal{A}$.

If $\mathcal{A}$ does not query g^{xy} to the random oracle, $M_b \oplus h$ in $\hat{E}nc(M_b)$ is the same as the one-time pad whose key is h. Since the one-time pad has perfect secrecy, no adversary can guess b with probability higher than 1/2 $\left(\text{i.e., } \Pr\left[\mathcal{A}_{success} \middle| g^{xy} \in Q\right] = 1/2\right)$.

We can get $\Pr\left[g^{xy} \in Q\right] \geq 2 \cdot Adv$ as follows:

$$\begin{aligned} Adv &= \Pr[\mathcal{A}_{success}] - \frac{1}{2} \\ &= \Pr\left[\mathcal{A}_{success} \middle| g^{xy} \in Q\right] \cdot \Pr\left[g^{xy} \notin Q\right] \\ &\quad + \Pr\left[\mathcal{A}_{success} \middle| g^{xy} \in Q\right] \cdot \Pr\left[g^{xy} \in Q\right] - \frac{1}{2} \\ &\leq \frac{1}{2} \cdot \Pr\left[g^{xy} \notin Q\right] + 1 \cdot \Pr\left[g^{xy} \in Q\right] - \frac{1}{2} \\ &\leq \frac{1}{2} \cdot \left(1 - \Pr\left[g^{xy} \in Q\right]\right) + 1 \cdot \Pr\left[g^{xy} \in Q\right] - \frac{1}{2} \\ &= \frac{1}{2} \cdot \Pr\left[g^{xy} \in Q\right]. \end{aligned}$$

On the other hand, if $\mathcal{A}$ queries g^{xy} to the random oracle, $\mathcal{CH}$ can solve the CDH problem with probability $1/|Q|$ simply by returning a random element of Q $\left(\text{i.e., } \Pr\left[\mathcal{A}_{success} \middle| g^{xy} \in Q\right] \geq 1/|Q|\right)$.

Now, we can compute a lower bound of $\mathcal{CH}$'s success probability:

$$\begin{aligned} \Pr(\mathcal{CH}_{success}) &\geq \Pr\left[g^{xy} \in Q\right] \cdot \Pr\left[\mathcal{A}_{success} \middle| g^{xy} \in Q\right] \\ &\geq 2 \cdot Adv/|Q|. \end{aligned}$$

$|Q|$ is polynomially bounded in the security parameter because the running time of $\mathcal{A}$ is polynomially bounded. Therefore, the success probability of $\mathcal{CH}$ is non-negligible, which contradicts CDH assumption. □

Theorem 2 (STTP-obliviousness): Under the CDH assumption, the strong-STTP protocol satisfies STTP-obliviousness in the random oracle model.

Proof: The proof is similar to that of Theorem 1 except simulation setup. In Step 1, $\mathcal{CH}$ chooses random $s_1,...,s_t,\ s_{STTP} \in [1..q-1]$. $\mathcal{CH}$ simulates verification keys $VK_1 = g^{s_1},...,VK_t = g^{s_t}, VK_{STTP} = g^{s_{STTP}}$. For each $i \in [t+1..\ell]$, $\mathcal{CH}$ simulates:

$$VK_i = \left(\frac{PK}{VK_1^{\lambda_1} ... VK_t^{\lambda_t} \cdot VK_{STTP}}\right)^{1/\lambda_i}.$$

$\mathcal{CH}$ sends common parameters:

$$(PK, VK_1, ..., VK_\ell, VK_{STTP})$$

to $\mathcal{A}$. In Step 2, $\mathcal{CH}$ sends t secret key shares $(s_1,...,s_t)$ and s_{STTP} to $\mathcal{A}$.

Then, since the arguments of Theorem 1 hold in the same way, no adversary can obtain non-negligible advantage from the above game if the CDH problem is hard. That leads to STTP obliviousness under the CDH assumption in the random oracle model.

Theorem 3 (Fairness): The strong-STTP protocol satisfies fairness of threshold decryption with STTP in Definition 2.

Proof: By Theorem 1, no t shareholders can succeed in decryption without following the protocol until the STTPs sending its decryption share. The STTP contributes its decryption share only after all the shareholders of S exchange their decryption shares with one another. Since the STTP sends its share to the shareholders at the same time, it guarantees that they can get the decrypted message simultaneously. From this and Theorem 2, the strong-STTP protocol satisfies the fairness with STTP in Definition 2. □

4.3 Multiple-weak STTP fair threshold Hash-ElGamal

4.3.1 Description

When all the shareholders cannot agree on a mutually satisfactory STTP, they can make use of several weak STTPs instead of one strong STTP. The members of this scenario are a dealer D, ℓ shareholders and k weak STTPs. We assume that each shareholder trusts at least one weak STTP. Moreover, we assume that any subset of less than $t+1$ shareholders has at least one common trustworthy weak STTP, which we call the *technical covering condition*.

The reliability between weak STTPs and all the shareholders is public. Let S be a reconstruction group of $t+1$ shareholders that collaboratively want to decrypt the ciphertext. Before introducing our scheme, we formally define the technical covering condition as follows.

Definition 3: (Technical covering condition) Given a reconstruction group S with $|S| = t+1$, and k weak STTPs, let $T_i \subseteq [1..k]$ be the subset of indices of the weak STTPs that P_i trusts (i.e., P_i's trustworthy weak STTPs), and let $U_i = [1..k] - T_i$ be that of the weak STTPs that P_i do not trust (i.e., P_i's untrustworthy weak STTPs). For any subset $F \subset [1..\ell]$ with $|F| \leq t$, if $\bigcap_{i \in F} T_i$ is non-empty, then we say the technical covering condition is satisfied.

Table 1 and Figure 2 show an example relationship between weak STTPs and shareholders with $t = 2$ satisfying the technical covering condition. All the shareholders cannot agree on a common STTP but every combination of two shareholders trusts at least one of the weak STTPs together.

Once we assume that the technical covering condition is satisfied, we can use the following scheme for fair threshold decryption. In this scenario, each shareholder P_i of S collects the decryption shares of U_i (his untrustworthy weak STTPs) so that it can proceed to the threshold decryption without doubting them. Then, all the shareholders of S exchange their decryption shares with

one another and send $READY$ signals to all the weak STTPs. When each weak STTP receives $READY$ signals from all the shareholders of S, it sends its own $READY$ signals to all other weak STTPs. When each weak STTP receives $READY$ signals from all the remaining weak STTPs, it triggers the decryption by sending its decryption share to the shareholders who trust it. Figure 3 shows this scenario graphically. The details of the protocol are as follows.

Table 1 Trust table

Shareholder	*Trustworthy STTP*	*Untrustworthy STTP*
P_1	$STTP_1, STTP_2$	$STTP_3$
P_2	$STTP_1, STTP_3$	$STTP_2$
P_3	$STTP_1, STTP_3$	$STTP_2$
P_4	$STTP_2, STTP_3$	$STTP_1$

Figure 2 Relation graph (see online version for colours)

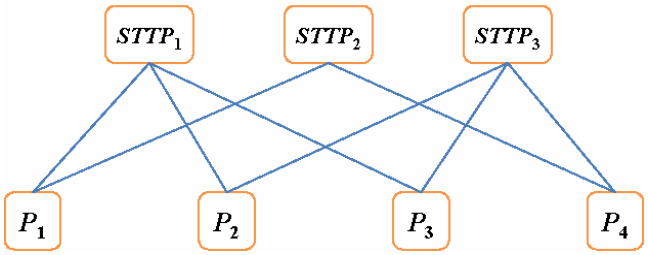
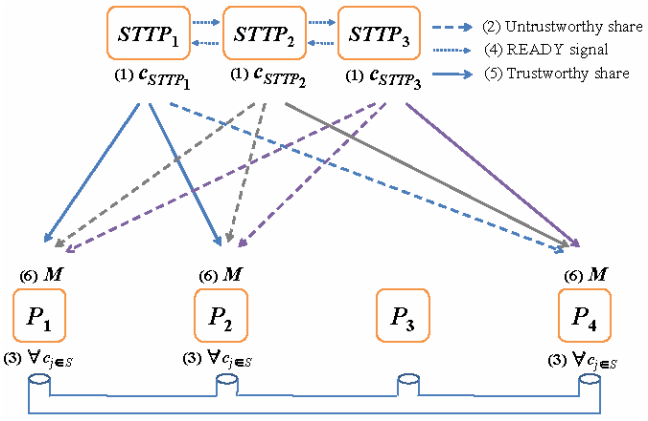


Figure 3 An example of fair threshold decryption with weak STTPs under the trust table in Figure 2, where a reconstruction group is P_1, P_2, P_4 and $k=3$ (see online version for colours)



Note: Dotted line indicates that each STTP sends its decryption share to the shareholders who do not trust it in Step 1, and solid line indicates that each STTP sends it to the shareholders who trust it in Step 4.

Key generation

D chooses random $x, R_1, ..., R_k \in [1..q-1]$, and computes $PK = g^x$, $VK_{STTP_1} = g^{R_1}, ..., VK_{STTP_k} = g^{R_k}$. D picks an otherwise random degree t polynomial $f(\cdot)$ with coefficients in $[0..q-1]$ such that $f(0) = \left(x - \sum_{i=1}^{k} R_i\right) \bmod q$. Then for all $i, 1 \le i \le \ell, D$ computes $s_i = f(i) \bmod q$, $VK_i = g^{s_i}$ and sends s_i to P_i. Then for all $j, 1 \le j \le k, D$ sends $s_{STTP_j} = R_j$ to $STTP_j$. Lastly, D publishes:

$$g, PK, \left\{\left(j, VK_{STTP_j}\right)\right\}_{1 \le j \le k}, \left\{\left(i, VK_i\right)\right\}_{1 \le i \le \ell}.$$

Multiple-weak STTP fair threshold decryption

Let $E(m) = (u, v)$, and let $S \subseteq [1..\ell], |S| = t+1$. Let T_i and U_i be the same as those of Definition 3.1. We assume that the technical covering condition is satisfied. We assume that the STTPs know $(u, v), S$ and $\{T_i\}_{i \in S}$.

1 For every $j \in [1..k]$, and for every $i \in [1..\ell]$ such that $j \in U_i, STTP_j$ sends $w_{STTP_j} = u^{R_j}$ and a zk-proof that $disc \log_g \left(VK_{STTP_j}\right) = disc \log_u \left(w_{STTP_j}\right)$ to P_i.

2 Each P_i checks the decryption shares it received from U_i (its untrustworthy weak STTPs) and it halts if any is bad. Otherwise, P_i sends $w_i = u^{s_i}$ and a zk-proof that $disc \log_g \left(VK_i\right) = disc \log_u \left(w_i\right)$ to every shareholder in S. When P_i receives good decryption shares from all the other shareholders in S, then P_i sends a $READY$ signal to all the STTPs.

3 Each STTP waits for $READY$ signals from all the shareholders in S, and then sends its own $READY$ signal to all the other STTPs. Each STTP goes on to the next step after receiving $READY$ signals from all the other STTPs.

4 For every $j \in [1..k]$, and for every $i \in [1..\ell]$ such that $j \in T_i, STTP_j$ sends w_{STTP_j} and a zk-proof that $disc \log_g \left(VK_{STTP_j}\right) = disc \log_u \left(w_{STTP_j}\right)$ to P_i.

5 Each P_i can now decrypt, since $g^{rx} = \prod_{1 \le j \le k} w_{STTP_j} \cdot \prod_{i \in S} w_i^{\lambda_i}$ where $\lambda_i = \prod_{b \in S \setminus \{i\}} \frac{i}{b-i}$.

Non-threshold access structure

Instead of viewing this as a $(t+1)$-out-of-ℓ threshold decryption scheme with ℓ shareholders $(P_1, ..., P_\ell)$ and k separate STTPs $(STTP_1, ..., STTP_k)$, we can view it as a non-threshold access structure with $\ell + k$ shareholders

$(STTP_1,...,STTP_k,P_1,...,P_\ell)$. Specifically, the access structure Γ' for successful decryption is $(STTP_1 \wedge ... \wedge STTP_k) \wedge \Gamma_{t+1,\ell}$, where $\Gamma_{t+1,\ell}$ is the $(t+1)$-out-of-ℓ threshold access structure on $\{P_1,...,P_\ell\}$. Multiple-weak STTP fair threshold decryption is possible whenever this non-threshold access structure Γ' can be realised.

Communication cost

Each shareholder's communication cost is $O(t+k)$ and the total communication cost is $O((t+k)^2)$.

4.3.2 Security proof of multiple weak-STTP versions

Now we prove that the multiple weak-STTP protocol satisfies *STTP-assistance*, *STTP-obliviousness*, and *fairness* from the CDH assumption.

Theorem 4 (STTP-assistance): Under the CDH assumption, the multiple weak-STTP protocol satisfies STTP-assistance in the random oracle model.

Proof: This proof is a simple extension of Theorem 1. $\mathcal{CH}$ gets an instance of CDH problem (g,g^x,g^y) whose goal is computing g^{xy}. Without loss of generality, we assume that the secret share of kth weak STTP is not given to $\mathcal{CH}$.

In Step 1, $\mathcal{CH}$ chooses random $R_1,...,R_{k-1} \in [1..q-1]$, and a random polynomial $f(\cdot)$ with degree t. He simulates the multiple weak-STTP protocol with initialising $PK = g^x$, $s_1 = f(1),...,s_\ell = f(\ell), s_{STTP_j} = R_j$ for $1 \le j \le k-1$. Then, s_{STTP_k} is automatically chosen as $s_{STTP_k} = x - f(0) - \sum_{i=1}^{k-1} R_i$ but $\mathcal{CH}$ knows neither x nor s_{STTP_k}. The verification keys are easily computed by $VK_1 = g^{s_1},...,VK_\ell = g^{s_\ell}, VK_{STTP_1} = g^{R_1},...,VK_{STTP_{k-1}} = g^{R_{k-1}}$, $VK_{STTP_k} = \frac{g^x}{g^{f(0)} \cdot \prod_{1 \le i \le k-1} g^{R_i}}$. $\mathcal{CH}$ sends common parameters $(PK, VK_1,...,VK_\ell, VK_{STTP})$ to $\mathcal{A}$.

In Step 2, $\mathcal{CH}$ sends $(s_1,...,s_\ell, s_{STTP_1},...,s_{STTP_{k-1}})$ to $\mathcal{A}$. $\mathcal{A}$ cannot distinguish these from a set of normal parameters because all of them are valid.

Then, the remaining steps are simulated as in the proof of Theorem 1 and the STTP-assistance is derived easily. □

Theorem 5 (STTP-obliviousness): Under the CDH assumption, the multiple weak-STTP protocol satisfies STTP-obliviousness in the random oracle model.

Proof: This proof is also a simple extension of Theorem 2. In Step 1, $\mathcal{CH}$ chooses random $s_1,...,s_t$, $s_{STTP_1},...,s_{STTP_k} \in [1..q-1]$. The verification keys of these secret key shares are computed by $VK_i = g^{s_i}$ and $VK_{STTP_j} = g^{s_{STTP_j}}$ respectively. $\mathcal{CH}$ does not know $s_{t+1},...,s_\ell$ but he can simulate their verification keys by $VK_i = \left(\frac{PK}{VK_1^{\lambda_1} ... VK_t^{\lambda_t} \cdot VK_{STTP}} \right)^{1/\lambda_i}$ for $t+1 \le i \le \ell$. $\mathcal{CH}$ sends common parameters $(PK, VK_1,...,VK_\ell, VK_{STTP_1},...,VK_{STTP_k})$ to $\mathcal{A}$. In Step 2, $\mathcal{CH}$ sends $(s_1,...,s_t, s_{STTP_1},...,s_{STTP_k})$ to $\mathcal{A}$. Then, the STTP-oblivious is derived as in Theorem 2. □

Theorem 6 (Fairness): The weak-STTP protocol satisfies fairness of threshold decryption with STTP in Definition 2.

Proof: From the technical covering condition, there exists a weak STTP, $STTP_j$, which t shareholders trust together. Then, all the arguments in Theorem 3 hold similarly and the proof of the fairness in Definition 2 is straightforward. □

4.3.3 A Variant avoiding the technical covering condition

In the previous multiple-weak STTP fair threshold decryption, the reliability graph between shareholders and weak STTPs should satisfy the technical covering condition to prevent malicious shareholders from receiving all the decryption shares of the group in Step 1, where the shareholders get decryption shares of his untrustworthy STTPs. However, as the threshold t, the number of shareholders ℓ, and the number of weak STTPs k are increasing, the reliability graph is more and more difficult to satisfy the technical covering condition. For example, for any two shareholders P_i and P_j, if $T_i = U_j$ and $T_j = U_i$, then $T_i \bigcap T_j = \phi$ and the technical covering condition cannot be satisfied regardless of the reliability between other shareholders and weak STTPs.

We present a multiple-weak STTP fair threshold Hash-ElGamal which works correctly even if the reliability graph does not satisfy the technical covering condition. In this protocol, each weak STTP sends different decryption shares to the shareholders so that any coalition of t shareholders cannot collect all the decryption shares of weak STTPs until they send $READY$ signals. If the reliability graph satisfies the technical covering condition, we do not have to use this protocol since it requires $O(\ell)$ storage for each weak STTP while the previous one requires $O(1)$. We assume that every shareholder trusts at least one STTP and the reliability graph does not have to satisfy the technical covering condition.

Key generation

D chooses random $x, R, R_{(1,1)}, R_{(1,2)},..., R_{(\ell,k-1)} \in [1..q-2]$, and computes:

$$R_{(i,k)} = R - \sum_{j=1}^{k-1} R_{(i,j)}$$

so that $R = \sum_{j=1}^{k} R_{(i,j)}$ for every $i \in [1..\ell]$, and computes $PK = g^x$, $\left\{VK_{STTP_{(i,j)}} = g^{R_{(i,j)}}\right\}_{1\le i\le \ell, 1\le j\le k}$. D picks an otherwise random degree t polynomial $f(\cdot)$ with coefficients in $[0..q-1]$ such that $f(0) = (x - R) \bmod q$. Then for all $i, 1 \le i \le \ell, D$ computes $s_i = f(i) \bmod q$, $VK_i = g^{s_i}$, and sends s_i to shareholder P_i. Then for all $(i,j) \in [1..\ell]\times[1..k], D$ sends $s_{STTP_{(i,j)}} = R_{(i,j)}$ to $STTP_j$. Lastly, D publishes g, PK,

$$\left\{VK_{STTP_{(i,j)}}\right\}_{1\le i\le \ell, 1\le j\le k}, \left\{(i, VK_i)\right\}_{1\le i\le \ell}.$$

Multiple-weak STTP fair threshold decryption (variant)

Let $E(m) = (u, v)$, and let $S \subseteq [1..\ell]$, $|S| = t+1$. For each $i \in S$, let $T_i \subseteq [1..k]$ be the subset of indices of the weak STTPs that P_i trusts, and let $U_i = [1..k] - T_i$ be that of the weak STTPs that P_i does not trust. We assume that the STTPs know $(u, v), S$ and $\{T_i\}_{i\in S}$.

1 For every $j \in [1..k]$, and for every $P_i \in S$ such that $j \in U_i, STTP_j$ sends $w_{STTP_{(i,j)}=u^{R_{(i,j)}}}$ and a zk-proof that $disc\log_g\left(VK_{STTP_{(i,j)}}\right) = disc\log_u\left(w_{STTP_{(i,j)}}\right)$ to P_i.

2 Each P_i checks the decryption shares it received from U_i (the STTPs it does not trust), and halts if any is bad. Otherwise, P_i sends $w_i = u^{s_i}$ and a zk-proof that $disc\log_g\left(VK_i\right) = disc\log_u\left(w_i\right)$ to every shareholder in S. When P_i receives good decryption shares from all the other shareholders in S, then P_i sends a $READY$ signal to all the STTPs.

3 Each STTP waits for $READY$ signals from all the shareholders of S, and then sends its own $READY$ signal to all the other STTPs. Each STTP goes on to the next step after receiving $READY$ signals from all the other STTPs.

4 For every $j \in [1..k]$, and for every $P_i \in S$ such that $j \in T_i, STTP_j$ sends $w_{STTP_{(i,j)}}$ and a zk-proof that $disc\log_g\left(VK_{STTP_j}\right) = disc\log_u\left(w_{STTP_{(i,j)}}\right)$ to P_i.

5 Each P_i can now decrypt the ciphertext by combining the decryption shares:

$$g^{rx} = \prod_{1\le j\le k} w_{STTP_{(i,j)}} \cdot \prod_{i\in S} w_i^{\lambda_i}$$

where $\lambda_i = \prod_{b\in S\setminus\{i\}} \frac{i}{b-i}$.

Multiple-weak STTP fair threshold security (variant)

For brevity, we give a proof sketch of security and fairness, which helps to convey some intuition behind our protocol.

(Sketch) Let F be a coalition of at most t cheating shareholders. This coalition of dishonest shareholders cannot decrypt any ciphertext unless they collect $\left\{w_{STTP_{(i,j)}}\right\}_{1\le j\le k}$ for some i from all the weak STTPs. Since each of cheating shareholder P_i receives $w_{STTP_{(i,j)}}$ from each $STTP_j$ he does not trust (i.e., $j \in U_i$) in Step 1 and there is at least one weak STTP which P_i trusts, he cannot collect $\left\{w_{STTP_{(i,j)}}\right\}_{1\le j\le k}$ by himself until Step 3 even when he repetitively initialises and aborts the protocol. Since the shareholders of F have distinct i values, they cannot collect $\left\{w_{STTP_{(i,j)}}\right\}_{1\le j\le k}$ for any $1 \le i \le \ell$ even together. Thus, if any shareholder of F decrypts, that implies the protocol proceeds to Step 4 in which the shareholders of S receive the decryption shares of weak STTPs they trust. This implies each shareholder of some reconstruction group S has ℓ decryption shares of the shareholders and decryption shares of weak STTPs he does not trust. Since each weak STTP sends his decryption share to the shareholders who trust him, every shareholder can collect enough decryption shares to decrypt and the fairness holds.

5 Optimistic fair threshold decryption

5.1 Security notions

Our notion of optimistic fair threshold decryption uses an OTTP which is semi-honest but not attending the protocol if all the shareholders behave honestly. We do not require optimistic protocols to satisfy *STTP-assistance* because any $t+1$ honest shareholders can decrypt the ciphertext without OTTP. We use the following security notions for optimistic fair threshold decryption.

5.1.1 Security for threshold decryption

Informally, a $(t+1)$-threshold decryption scheme is secure if t number of shareholders cannot decrypt any ciphertext. Formally, we say that a threshold scheme is semantically secure if any polynomially bounded adversary $\mathcal{A}$ cannot win the following game with non-negligible probability. The game proceeds between $\mathcal{A}$ and $\mathcal{CH}$:

1 $\mathcal{CH}$ runs setup and key generation algorithms taking a security parameter. $\mathcal{CH}$ gives $\mathcal{A}$ the resulting common parameters and t secret key shares.

2 $\mathcal{A}$ adaptively makes a polynomial number of queries to $\mathcal{CH}$ on (M,S) where $S \subseteq [1..\ell], |S| = t+1$. For each (M,S), $\mathcal{CH}$ generates encryption C of M and responds with C and the messages each shareholder sends in the execution of optimistic decryption on (C,S).

3 $\mathcal{A}$ adaptively queries for OTTPs responses to $\mathcal{CH}$.

4 A selects two target messages (M_0, M_1). $\mathcal{CH}$ picks one message M_b by selecting a random bit $b \leftarrow \{0,1\}$ and sends a ciphertext C_b of M_b to $\mathcal{A}$. $\mathcal{CH}$ also sends to $\mathcal{A}$ all the intermediate information except decryption shares.

5 Repeat Step 2.

6 A outputs b' (and wins if $b' = b$).

5.1.2 OTTP obliviousness

We say that a threshold scheme is OTTP-oblivious if any polynomially bounded adversary $\mathcal{A}$ cannot win the following game with non-negligible probability. The game proceeds between $\mathcal{A}$ and $\mathcal{CH}$:

1 $\mathcal{CH}$ runs setup and key generation algorithms taking a security parameter. $\mathcal{CH}$ gives $\mathcal{A}$ the resulting common parameters and OTTPs secret key.

2 A adaptively makes a polynomial number of queries to $\mathcal{CH}$ on (M,S) where $S \subseteq [1..\ell], |S| = t+1$. For each (M,S), $\mathcal{CH}$ generates encryption C of M and responds with C and the messages each shareholder sends in the execution of optimistic decryption on (C,S).

3 A selects two target messages (M_0, M_1). $\mathcal{CH}$ picks one message M_b by selecting a random bit $b \leftarrow \{0,1\}$ and sends a ciphertext C_b of M_b to $\mathcal{A}$. $\mathcal{A}$ participates in a decryption protocol on (C_b, S) on behalf of the OTTP.

4 Repeat Step 2.

5 A outputs b' (and wins if $b' = b$).

5.2 Optimistic fair threshold Hash-ElGamal

5.2.1 Description

The main idea of this protocol is that all the shareholders of S exchange the promises of decryption shares before they exchange the decryption shares. A promise of decryption share is an encrypted value containing partial information of the decryption share using OTTPs public key. It assures the receiver that he can obtain the decryption shares with the help of OTTP. Thus, once each shareholder receives all the promises, it can obtain all the decryption shares even when some shareholders behave maliciously.

In this protocol, the OTTP does not respond to any query before all the promises are exchanged. To guarantee this, each shareholder sends its signed $READY$ signal to all the other shareholders of S, so that only shareholders who receive all the signed $READY$ signals can query to the OTTP in order to get the decryption shares that they have not received. The OTTP accepts only queries enclosed with all the signed $READY$ signals of S. (We can regard it as OTTPs decryption policy.) Once the OTTP receives a query with the signed $READY$ signals, it can be assured that the shareholders of S already received all the promises of S. At this moment, the OTTP sends all the signed $READY$ signals to the other shareholders of S. It guarantees that the shareholders have the right to query to the OTTP.

One main tool is *verifiable encryption*, which allows someone to prove that an encrypted value is the discrete logarithm of an unencrypted value (with respect to an unencrypted base). Let ENC be a public key encryption scheme that supports verifiable encryption as in Camenisch-Shoup (Camenisch and Shoup, 2003). The details of the protocol are as follows.

Key generation

A dealer initialises common parameters of ordinary threshold Hash-ElGamal as in Section 2.2. OTTP creates a key pair (SK_{OTTP}, PK_{OTTP}) for $ENC(\cdot)$.

Optimistic fair threshold decryption

Let $E(m) = (u,v)$, $S \subseteq [1..\ell], |S| = t+1$. We assume that all of the shareholders agree on the session information inf which includes $S, E(m)$.

1 In Round 1, for every $i, j \in S$:

a P_i sends the following promise to P_j (unfair, blinded, partially signed): $\left(\alpha_i, \beta_i, \text{inf}, g^{r/\alpha_i}, ENC_{OTTP}(\beta_i s_i), \sigma_i, proof\left(disc\log_g\left(VK_i^{\beta_i}\right) = \beta_i s_i\right)\right)$, where $\alpha_i \beta_i \in_R [1..q-1]$ chosen by P_i and σ_i is the signature by P_i of $\left(\text{inf}, g^{r/\alpha_i}, ENC_{OTTP}(\beta_i s_i)\right)$. The disclog proof is Camenisch-Shoup style on the verifiably encrypted value $\beta_i s_i$.

b P_j checks if the promise from P_i is valid: well-formed, well-signed, well-blinded, disclog proof.

c If P_i receives $t+1$ valid promises of S (including its own), P_i proceeds to Round 2.

2 In Round 2, for every $i \in S$:
   a P_i sends $(READY, \text{inf})$ and its signature to the other shareholders in S.
   b If P_i receives $t+1$ signed $(READY, \text{inf})$ signals of S (including its own) from S or (possibly) OTTP, P_i proceeds to Round 3.

3 In Round 3, for every $i \in S$:
   a P_i sends to the other shareholders in S: $\left(g^{rs_i}, proof\left(disc\log_{g^r}\left(g^{rs_i}\right) = disc\log_g\left(VK_i\right)\right)\right)$. Here, the proof is an ordinary zk-proof for equality of discrete logs.
   b If P_i receives $t+1$ good decryption shares (including its own), then P_i decrypts as in ordinary (robust) threshold decryption, and halts. Otherwise, P_i proceeds to Round 4.

4 In Round 4, for every $i \in S$:
   a Let $\hat{S}$ be the subset of S which did not send P_i a good decryption share in Round 3.
   b P_i sends OTTP all the signed $(READY, \text{inf})$ signals of S, and the partially signed promise $\left(\text{inf}, g^{r/\alpha_i}, ENC_{OTTP}\left(\beta_j s_j\right), \sigma_j\right)$ from every shareholder j in $\hat{S}$.
   c OTTP checks all the signed $(READY, \text{inf})$ signals of S and all the signed part of promises of $\hat{S}$. If any of them is invalid, OTTP rejects the query.
   d If it is the first query of the session inf, OTTP sends all the signed $(READY, \text{inf})$ signals of S to all the shareholders of S to finish their waiting in Round 2.
   e OTTP notifies each P_j in $\hat{S}$ that P_i asks OTTP so that P_j does not wait for P_i's decryption share in Round 3.
   f OTTP decrypts all the $\left(ENC_{OTTP}\left(\beta_j s_j\right)\right)_{j\in\hat{S}}$.
   g OTTP computes and sends to P_i all the $\left(\left(g^{r/\alpha_j}\right)^{\beta_j s_j}\right)_{j\in\hat{S}}$.
   h P_i now unbinds all the $\left(\left(g^{r/\alpha_j}\right)^{\beta_j s_j}\right)_{j\in\hat{S}}$. using all the $\left(\alpha_j, \beta_j\right)_{j\in\hat{S}}$. P_i can obtain the plaintext as in ordinary threshold decryption, and halts.

Remark 1: The promises of Step 1 are 'blinded' so that the OTTP cannot compute any decryption share even if it responds to a number of queries. In the blinded promises, we use two random numbers α_i, β_i to hide the secret key shares and the decryption shares from the OTTP. Any shareholder, who receives a blinded promise, can verify its validity by checking $\left(g^{r/\alpha_i}\right)^{\alpha_i} = g^r$ and $proof\left(disc\log_g\left(VK_i^{\beta_i}\right) = \beta_i s_i\right)$.

Remark 2: This protocol does not satisfy a useful property *timely termination* (Asokan et al., 2000) with which a shareholder can leave the protocol immediately in a fair manner without waiting for the responses of other shareholders. For instance, let's consider the following case in which a shareholder P_i is in Round 2, and the other shareholders in Round 3 do not send their decryption shares for a long-time. In this case, P_i can neither go to Round 3 since he has not collected all the signed $READY$ signals, nor leave the protocol since the other shareholders will succeed in decryption by querying the OTTP. Nevertheless, this protocol is *fair* since P_i can succeed in decryption whenever another shareholder succeeds in decryption if he has not leaved the protocol. We can modify this protocol so that timely termination is satisfied. For space constraints, this variant will be presented in the full version.

Communication cost

Each shareholder's communication cost is $O(t)$ and the total communication cost is $O\left(t^2\right)$.

5.2.2 Security proof of OTTP version

Theorem 7: (Security for threshold decryption) Under the CDH assumption, unforgeability of the signature scheme and the security of verifiable encryption, the above optimistic protocol is a semantically secure threshold decryption protocol in the random oracle model.

Proof: The proof is very similar with the proof for Theorem 2. Assume that $\mathcal{A}$ able to break security for threshold decryption. Using this adversary $\mathcal{A}$, we can build an algorithm to solve the CDH problem with non-negligible probability: Given an instance of CDH problem $\left(g, g^x, g^y\right)$, the algorithm computes g^{xy}. $\mathcal{CH}$ initialises $PK = g^x$, $s_1, ..., s_t \in_R [0..q-1]$, and $VK_1 = g^{s_1}, ..., VK_t = g^{s_t}$. For each $i \in [t+1..\ell]$, $\mathcal{CH}$ sets $VK_i = \left(\frac{PK}{VK_1^{\lambda_1} ... VK_t^{\lambda_t}}\right)^{1/\lambda_i}$. $\mathcal{CH}$ also generates $\left(SK_{OTTP}, PK_{OTTP}\right)$. $\mathcal{A}$ is given public parameters $\left(PK, VK_1, ..., VK_\ell, PK_{OTTP}\right)$ and t secret key shares $\left(s_1, ..., s_t\right)$. In Step 2, for each query (M, S) of $\mathcal{A}, \mathcal{CH}$ responds with $t+1$ decryption shares $\left\{w_j = \left(VK_j\right)^r\right\}_{j\in S}$. He also computes $t+1$ promises with the simulated signatures σ_i and zk-proofs

$proof\left(disc\log_g\left(VK_i^{\beta_i}=\beta_i s_i\right)\right)$. In Step 4, for given two messages M_0, M_1 from $\mathcal{A}$,$\mathcal{CH}$ picks one message M_b and sends a dummy ciphertext $\hat{Enc}(M_b)$. $\mathcal{CH}$ also simulates promises, signatures and zk-proofs. Under the unforgeability of the signature scheme, $\mathcal{A}$'s query to the OTTP on input created by $\mathcal{A}$ cannot pass the validity test of the input. Moreover, under the security of verifiable encryption, no information is revealed from the promises in Round 1. $READY$ signals in Round 2 do not contain any information about the plaintext. Thus, similar to Theorem 2, $\mathcal{A}$ can win the game in polynomial time with non-negligible probability only by querying g^{xy} to the random oracle, which contradicts to the CDH assumption. □

Theorem 8 (OTTP-obliviousness): Under the CDH assumption, the optimistic fair threshold satisfies *OTTP-obliviousness*.

Proof: The proof is the same as the proof for Theorem 7 except: $\mathcal{A}$ is given SKOTTP instead of t secret shares. Still, it is clear that unless α_i, β_i are given, $\mathcal{A}$ cannot learn anything about s_i from the partially signed promises. Thus, the only way to win the game is to make a query on g^{xy}. Again, it contradicts to the CDH assumption. □

Theorem 9 (Fairness): Under the CDH assumption, unforgeability of the signature scheme, security of the verifiable encryption, the optimistic protocol satisfies fairness of threshold decryption with OTTP in Definition 2.

Proof: Let F be a coalition of at most t cheating shareholders. All the information which F can receive comes from the promises in Round 1, the $READY$ signals in Round 2, and the decryption shares in Round 3. Since $READY$ signals have nothing to do with any secret information, we only need to consider promises and decryption shares.

Claim 1: If F decrypts the ciphertext without querying to the OTTP, all the honest shareholders of S can decrypt it too.

If F does not receive decryption shares from the shareholders in Round 3, F fails to decrypt by Theorem 7. Otherwise, there exists at least one honest shareholder P_i in Round 3 who sends its decryption share to F. In Rounds 1 and 2, P_i received all the promises and the signed $READY$ signals of S. It implies that P_i can obtain all the decryption shares with the help of the OTTP whenever it wants. Furthermore, from the signed $READY$ signals which P_i received, it is obvious that all the honest shareholders had received all the promises in Round 1 and proceeded to (at least) Round 2. When P_i queries to the OTTP, all the honest shareholders in Round 2 can proceed to Round 3 due to the signed $READY$ signals from the OTTP. Then, all the honest shareholders of S can decrypt the ciphertext with the help of the OTTP and Claim 1 holds.

Claim 2: If F decrypts the ciphertext by querying to the OTTP, all the honest shareholders of S can decrypt it too.

F can query to the OTTP only when he has all the signed READY signals. It implies that all the shareholders of S have sent their signed $READY$ signals to F. It also implies that all the honest shareholders had received all the promises in Round 1 and proceeded to Round 2. Thus, whenever F queries to the OTTP, all the honest shareholders can proceed to Round 3 by receiving the signed $READY$ signals which the OTTP sends. Then, all the honest shareholders of S can query to the OTTP and Claim 2 holds.

By both claims, F cannot cause any unfairness whatever they do. Thus, the OTTP protocol satisfies the fairness of Definition 1.

6 Conclusions

In this paper, we formally define the notion of fairness in threshold decryption and present various fair threshold decryption schemes to address realistic trust scenarios. When a strong-STTP is available, one strong-STTP suffices to achieve fairness, while multiple weak-STTPs are required in weaker trustful situations. We also propose *optimistic* fair threshold decryption scheme where no STTP attend the protocol when all the shareholders behave honestly. All our schemes are provably secure under CDH assumption in the random oracle model.

Although we describe our schemes using a simple hashed version of ElGamal encryption, our methods generalise to other threshold decryption schemes and threshold signature schemes as well. We expect that our ideas are applicable to other fair computational problems too.

Acknowledgements

This work was supported by NAP of Korea Research Council of Fundamental Science and Technology. The ICT at Seoul National University provides research facilities for this study. Jihye Kim was supported by 'Developing Future Internet Network Model – Mathematical Approach' of the National Institute for Mathematical Sciences.

References

Asokan, N., Schunter, M. and Waidner, M. (1997) 'Optimistic protocols for fair exchange', in *ACM Conference on Computer and Communications Security*, pp.7–17.

Asokan, N., Shoup, V. and Waidner, M. (2000) 'Optimistic fair exchange of digital signatures', *IEEE J. Selected Areas in Communication*, April, Vol. 18, No. 4, pp.593–610.

Bao, F., Deng, R.H. and Mao, W. (1998) 'Efficient and practical fair exchange protocols with off-line TTP', in *Proceedings of IEEE Symposium on Security and Privacy*, May, pp.77–85.

Ben-Or, M., Goldreich, O., Micali, S. and Rivest, R.L. (1990) 'A fair protocol for signing contracts', *IEEE Transactions on Information Theory*, Vol. 36, No. 1, pp.40–46.

Ben-Or, M., Goldwasser, S. and Wigderson, A. (1988) 'Completeness theorems for non-cryptographic fault-tolerant distributed computation', in *STOC '88: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pp.1–10, ACM, New York, NY, USA.

Blum, M. (1983) 'How to exchange (secret) keys', *ACM Trans. Comput. Syst*., Vol. 1, No. 2, pp.175–193.

Boneh, D. and Naor, M. (2000) 'Timed commitments', in *CRYPTO*, pp.236–254.

Boneh, D., Gentry, C., Lynn, B. and Shacham, H. (2003) 'Aggregate and verifiably encrypted signatures from bilinear maps', in *EUROCRYPT*, pp.416–432.

Boudot, F., Schoenmakers, B. and Traoré, J. (2001) 'A fair and efficient solution to the socialist millionaires' problem', *Discrete Applied Mathematics*, Vol. 111, Nos. 1–2, pp.23–36.

Cachin, C. and Camenisch, J. (2000) 'Optimistic fair secure computation', in *CRYPTO*, pp.93–111.

Camenisch, J. and Shoup, V. (2003) 'Practical verifiable encryption and decryption of discrete logarithms', in *CRYPTO*, pp.126–144.

Chaum, D., Crépeau, C. and Damgard, I. (1988) 'Multiparty unconditionally secure protocols', in *STOC '88: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pp.11–19, ACM, New York, NY, USA.

Cleve, R. (1986) 'Limits on the security of coin flips when half the processors are faulty (extended abstract)', in *STOC*, pp.364–369.

Cox, B., Tygar, J.D. and Sirbu, M. (1995) 'Netbill security and transaction protocol', in *First USENIX Workshop on Electronic Commerce*, pp.77–88.

Desmedt, Y. and Frankel, Y. (1989) 'Threshold cryptosystems', in *CRYPTO*, pp.307–315.

Dodis, Y. and Reyzin, L. (2003) 'Breaking and repairing optimistic fair exchange from PODC 2003', in *Digital Rights Management Workshop*, pp.47–54.

Dodis, Y., Lee, P.J. and Yum, D.H. (2007) 'Optimistic fair exchange in a multi-user setting', in *Public Key Cryptography*, pp.118–133.

ElGamal, T. (1985) 'A public key cryptosystem and signature scheme based on discrete logarithms', *IEEE Transactions on Information Theory*, July, Vol. 31, No. 4, pp.469–472.

Fouque, P., Poupard, G. and Stern, J. (2000) 'Sharing decryption in the context of voting of lotteries', in *Proceedings of the 4th International Conference on Financial Cryptography*, Lecture Notes in Computer Science, Vol. 1962, pp.90–104, Springer-Verlag.

Franklin, M.K. and Reiter, M.K. (1997) 'Fair exchange with a semi-trusted third party', in *Proceedings of the 4th ACM Conference on Computer and Communications Security (CCS)*, April, pp.1–5.

Garay, J.A., MacKenzie, P. and Yang, K. (2004) 'Efficient and secure multi-party computation with faulty majority and complete fairness', Cryptology ePrint Archive, Report 2004/009, available at http://eprint.iacr.org/.

Garay, J.A., MacKenzie, P.D., Prabhakaran, M. and Yang, K. (2006) 'Resource fairness and composability of cryptographic protocols', in *TCC*, pp.404–428.

Gennaro, R., Halevi, S., Krawczyk, H. and Rabin, T. (2008) 'Threshold RSA for dynamic and ad-hoc groups', in *EUROCRYPT*, pp.88–107.

Gennaro, R., Jarecki, S., Krawczyk, H. and Rabin, T. (1996) 'Robust and efficient sharing of RSA functions', in *CRYPTO*, pp.157–172.

Gennaro, R., Jarecki, S., Krawczyk, H. and Rabin, T. (2001) 'Robust threshold DSS signatures', *Inf. Comput.*, Vol. 164, No. 1, pp.54–84.

Gennaro, R., Jarecki, S., Krawczyk, H. and Rabin, T. (2007) 'Secure distributed key generation for discrete log based cryptosystems', *J. Cryptology*, Vol. 20, No. 1, pp.51–83.

Gennaro, R., Rabin, T., Jarecki, S. and Krawczyk, H. (2000) 'Robust and efficient sharing of RSA functions', *J. Cryptology*, Vol. 13, No. 2, pp.273–300.

Goldreich, O. (2000) 'Secure multi-party computation (working draft, version 1.2)', available at http://www.wisdom.weizmann.ac.il/oded/pp.html.

Goldreich, O., Micali, S. and Wigderson, A. (1987) 'How to play any mental game', in *STOC '87: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pp.218–229, ACM, New York, NY, USA.

Gordon, S.D., Hazay, C., Katz, J. and Lindell, Y. (2008) 'Complete fairness in secure two-party computation', in *STOC*, pp.413–422.

Kissner, L. and Song, D. (2005) 'Privacy-preserving set operations', in *Advances in Cryptology – CRYPTO 2005*, Lecture Notes in Computer Science, Vol. 3621, pp.241–257, Springer-Verlag.

Lepinski, M., Micali, S., Peikert, C. and Shelat, A. (2004) 'Completely fair safe and coalition-safe cheap talk', in *PODC '04: Proceedings of the Twenty-third Annual ACM Symposium on Principles of Distributed Computing*, pp.1–10, ACM, New York, NY, USA.

Lindell, A.Y. (2008) 'Legally-enforceable fairness in secure two-party computation', in *CT-RSA*, pp.121–137.

Luby, M., Micali, S. and Rackoff, C. (1983) 'How to simultaneously exchange a secret bit by flipping a symmetrically-biased coin', in *FOCS*, pp.11–21.

Pinkas, B. (2003) 'Fair secure two-party computation', in *EUROCRYPT*, pp.87–105.

Rabin, T. and Ben-Or, M. (1989) 'Verifiable secret sharing and multiparty protocols with honest majority', in *STOC '89: Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, pp.73–85, ACM, New York, NY, USA.

Santis, A.D., Desmedt, Y., Frankel, Y. and Yung, M. (1994) 'How to share a function securely', in *STOC*, pp.522–533.

Shamir, A. (1979) 'How to share a secret', *Commun. ACM*, Vol. 22, No. 11, pp.612–613.

Shoup, V. (2000) 'Practical threshold signatures', in *EUROCRYPT*, pp.207–220.

Zhou, J. and Gollmann, D. (1997) 'An efficient non-repudiation protocol', in *IEEE Computer Society Press*, pp.126–132, Society Press, Los Alamitos.

Zhou, J., Deng, R.H. and Bao, F. (2000) 'Some remarks on a fair exchange protocol', in *Public Key Cryptography*, pp.46–57.

Notes

1 Some previous work in fair exchange used the term ‘OTTP’ to indicate offline third party which is fully-trusted rather than semi-trusted.

2 This can be regarded as a passive collusion among the STTPs and t shareholders where the colluding shareholders provide the STTPs with their secret key shares.