Lattice Based Efficient Threshold Public Key Encryption Scheme

Kunwar Singh[1]*, C. Pandu Rangan[2], and A.K.Banerjee[1]
[1]*National Institute of Technology Tiruchirappalli, India*
{kunwar, banerjee}@nitt.edu
[2]*Indian Institute of Technology Madras*
Chennai, Tamil Nadu, India
rangan@cse.iitm.ac.in

Abstract

A *(t,u)*- threshold public key encryption (TPKE) is a public key encryption where a private key is splitted among u decryption servers (insiders) and at least t decryption servers (insiders) are required to decrypt the ciphertext but no group of $t-1$ or less malicious insiders can decrypt the message. Bendlin and Damgard [1] presented first lattice based threshold public key encryption scheme based on Regev's LWE based encryption system [2]. We propose efficient lattice based threshold public key encryption scheme based on [3]. We have reduced size of the public key from $(n^2+1)log\ n \times \|Z_q\|$ to $(n^2+1) \times \|Z_q\|$ with the same ciphertext size where $\|Z_q\|$ is the number of bits required to represent an element of Z_q.
Resplittable threshold public key encryption (RTPKE) was introduced by Hanaoka et al [4] in a generic construction of CCA secure uni-directional proxy re-encryption scheme. RTPKE is a threshold public key encryption with an additional randomized algorithm *Tsplit*. Based on our efficient threshold scheme, we have constructed efficient resplittable threshold public key encryption scheme.

Keywords: Lattice, Resplittable Threshold Public Key Encryption, Learning With Error (LWE)

1 Introduction

A *(t,u)*- threshold public key encryption (TPKE) is a public key encryption where a private key is splitted among u decryption servers (insiders) and at least t decryption servers (insiders) are required to decrypt the ciphertext but no group of $t-1$ or less malicious insiders can decrypt the message. In this *(t,u)*-TPKE scheme an entity called the dealer wishes to decrypt a ciphertext C. The dealer sends C to u-decryption servers. Decryption servers carry out partial decryption on C and send partial decryption shares to dealer. If dealer receives at least t partial decryption shares then it combines these shares into complete decryption of C. With less than t partial decryption shares dealer can not reconstruct the plaintext. Threshold systems are non-interactive if there is no interaction among the decryption servers during decryption phase. Threshold systems are robust if dealer can verify that whether decryption servers have sent valid or invalid partial decryption share. Non-interactiveness and robustness are desirable properties of threshold systems [5, 6].
Resplittable threshold public key encryption (RTPKE) was introduced by Hanaoka et al [4] in a generic construction of CCA secure proxy re-encryption scheme. RTPKE is a threshold public key encryption with an additional randomized algorithm *Tsplit*. In TPKE key splitting is done in key generation algorithm that is also only once but in RTPKE key splitting can be done in polynomial number of times with the condition that the number of corrupted secret key shares output by the randomized algorithm *Tsplit*

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 4, number: 4, pp. 93-107

*Corresponding author: Computer Science and Engineering Department, National Institute of Technology, Trichy, Tiruchirappalli-15, Tamilnadu, India, Tel: +91-43125013212, Web: http://www.nitt.edu/home/academics/departments/cse/faculty/kunwar/

is less than threshold t.
In 1994, Peter Shor [7] showed that prime factorization and discrete logarithm problem can be solved in polynomial time on a quantum computer. In other words, once quantum computer comes into reality all of the public-key algorithms used to protect the Internet will be broken. The discovery facilitated research on new cryptosystems that are secure from quantum computers, collectively called post-quantum cryptography. Lattice based problems are conjectured to remain secure in the advent of quantum computers. Ajtai's seminal result [8] on the average case / worst case equivalence sparked great interest in lattice based cryptography. Informally, it means breaking the lattice based cryptosystem in the average case is as hard as solving some lattice based hard problems in the worst case. Recently Regev [2] defined the learning with error (LWE) problem and proved that it also enjoys similar average case / worst case equivalence hardness properties under a quantum reduction.

Our Contribution: Bendlin and Damgard [1] presented first lattice based threshold public key encryption scheme based on Regev's LWE based encryption system [2] for single bit. Tore Kasper Frderiksen [9] extended this single bit threshold public key encryption scheme to multi bit threshold scheme. In their scheme public key is $(A,b) \in (Z_q^{n\times m}, Z_q^m)$ where m is generally $nlog\ n$. In CT-RSA 2011 conference Lindner and Peikert [3] proposed efficient lattice based public key encryption scheme. Based on this scheme [3] we propose efficient threshold public key encryption scheme. In our scheme public key is $(A,b) \in (Z_q^{n\times n}, Z_q^n)$. We have reduced size of the public key from $(n^2+1)log\ n \times \|Z_q\|$ to $(n^2+1)\times\|Z_q\|$ with same ciphertext size where $\|Z_q\|$ is the number of bits required to represent an element of Z_q .
Hanaoka et al [4] gave a generic construction of CCA secure uni-directional proxy re-encryption scheme from three primitives: resplittable TPKE, PKE and digital signature. Among these three primitives only resplittable TPKE is a new primitive. To the best of our knowledge, there does not exist any lattice based uni-directional proxy re-encryption scheme. Based on our TPKE scheme, we have constructed efficient resplittable threshold public key encryption scheme which can be used to construct lattice based uni-directional proxy re-encryption scheme.

Paper Outline: Our paper is organized as follows. In section 2, we describe basic definitions, security models, results and hard problems required to understand rest of the paper. In section 3, we briefly describe Shamir's secret sharing [10]. In section 4, we describe our schemes. In section 5 we give conclusion and related open problems.

2 Preliminaries

2.1 Notation

We denote $[j] = \{0,1,...,j\}$, set of real numbers by R and the integers by Z. We assume vectors to be in column form and are written using small letters, e.g. x. Matrices are written as capital letters, e.g. X. $\|S\|$ denotes the Euclidean norm of the longest (maximum euclidean norm) vector in matrix S, i.e. $\|S\| := max_i\|s_i\|$ for $1 \le i \le k$.
We say that *negl(n)* is a negligible function in n if it is smaller than the inverse of any polynomial function in n for sufficiently large n.

2.2 Threshold Public Key Encryption

Threshold Public Key Encryption (TPKE) [5] consists of five algorithms.

TKeyGen(n,u,t)**:** On input a security parameter n,u and t, this algorithm outputs a threshold secret key $tsk = \{tsk_1,\ldots,tsk_n\}$, a threshold public key tpk and a verification key tvk. Decryption server i is given the private key share tsk_i.

Encrypt(tpk,m)**:** On input a public key tpk, and a message m, this algorithm outputs ciphertext C.

TShareDecryption(tpk,tsk_i,C)**:** It is a deterministic algorithm. On input a threshold public key tpk, a threshold secret key share tsk_i and ciphertext C, this algorithm outputs a decryption share μ_i or a special symbol $\perp$ (invalid share).

TShareVerify(tpk,tvk,C,μ_i)**:** On input a public key tpk, a verification key tvk, a decryption share μ_i and ciphertext C, this algorithm outputs valid if decryption share μ_i is valid or $\perp$ (invalid share).

TCombine$(tpk,tvk,C,\mu_1,\ldots,\mu_t)$**:** On input a public key tpk, a verification key tvk, a t decryption shares $\mu_1,\ldots\mu_t$ and ciphertext C, this algorithm outputs a message m or $\perp$ (invalid ciphertext).

Correctness. Threshold public key encryption is correct if suppose $\mu_i = \text{TShareDecryption}(tpk,tsk_i,C)$ and $C = Encrypt(tpk,m)$, following equation holds.

- TShareVerify$(tpk,tvk,C,\mu_i) = valid$.
- TCombine$(tpk,tvk,C,\mu_1,\ldots,\mu_t) = m$.

2.3 Resplittable Threshold Public Key Encryption (RTPKE)

In traditional TPKE key splitting is done in key generation algorithm that is also only once but in RTPKE key splitting can be done in polynomial number of times (there is separate randomized algorithm *Tsplit*). Resplittable threshold public key encryption scheme [4] consists of six algorithms.

TKeyGen(n,u,t)**:** On input a security parameter n,u and t, this algorithm outputs a threshold secret key tsk and a threshold public key tpk.

Encrypt(tpk,m)**:** On input a threshold public key tpk, and a message m, this algorithm outputs ciphertext C.

Tsplit(tsk)**:** On input a threshold secret key tsk, this algorithm outputs u shares of tsk as $tsk_1,\ldots,tsk_u$ and a verification key tvk.

TShareDecryption(tpk,tsk_i,C)**:** It is a deterministic algorithm. On input a threshold public key tpk, a threshold secret key share tsk_i and ciphertext C, this algorithm outputs a decryption share μ_i or a special symbol $\perp$ (invalid share).

TShareVerify(tpk,tvk,C,μ_i)**:** On input a public key tpk, a verification key tvk, a decryption share μ_i and ciphertext C, this algorithm outputs valid if decryption share μ_i is valid or $\perp$ (invalid share).

TCombine$(tpk,tvk,C,\mu_1,\ldots\mu_t)$**:** On input a public key tpk, a verification key tvk, a t decryption shares $\mu_1,\ldots,\mu_k$ and ciphertext C, this algorithm outputs a message m or $\perp$ (invalid ciphertext).

Correctness. Correctness condition of the RTPKE is same as the correctness condition of TPKE.

2.4 Security Model for Threshold Public Key Encryption

Here security model is adapted from [5]. Security of TPKE is defined using two properties: semantic security (IND-CPA) and consistency of decryptions.

2.4.1 Semantic Security (IND-CPA)

Following security model captures the idea that a polynomially bounded adversary with less than t number of corrupt decryption servers can not have any bit information of the plaintext. We define security model using a game that is played between the challenger and the adversary. The game proceeds as follows.

Setup: The adversary outputs a set $S \subset \{1,\ldots,u\}$ of $t-1$ decryption servers to corrupt. The challenger runs TKeyGen(n,u,t) to obtain a threshold secret key $tsk = tsk_1,\ldots,tsk_u$, a threshold public key tpk and a verification key tvk. Challenger gives a threshold public key tpk, a verification key tvk and all (j,tsk_j) for $j \in S$ to the adversary.

Challenge: The adversary submits target message m. Challenger picks a random bit $r \in \{0,1\}$ and a random ciphertext C. If $r = 0$ it sets the challenge ciphertext to $C^* :=$ Encrypt(tpk,m). If $r = 1$ it sets the challenge ciphertext to $C^* := C$. It sends C^* as challenge to the adversary.

Guess: Finally the adversary outputs a guess $r' \in \{0,1\}$ and wins if $r = r'$.
We refer an adversary $\mathscr{A}$ as an IND-CPA adversary. We define the advantage of the adversary $\mathscr{A}$ in attacking an TPKE scheme ξ as

$$Adv_{A,u,t}(\lambda) = |Pr[r = r'] - 1/2|$$

2.4.2 Decryption Consistency:

Here security model captures the idea that an adversary can not produce two sets of valid decryption shares for any ciphertext such that their corresponding (applying combine algorithm) plaintexts are different. Decryption consistency is defined using following game that is played between the challenger and adversary. In this game *Setup* part is same as above game. The adversary then outputs a ciphertext C and two sets of decryption shares $S = \{\mu_1,\ldots,\mu_t\}$ and $S' = \{\mu'_1,\ldots,\mu'_t\}$ each of size t. The adversary wins the game if:

1. The decryption shares S and S' are valid decryption shares for C under verification key TVK.
2. S and S' each contain decryption shares from t distinct servers; and
3. TCombine$(tpk,TVK,C,S) \neq TCombine(tpk,TVK,C,S')$.

We define the adversary's advantage in winning this game as $AdvCD_{A,u,t}(\lambda)$.

Definition 1. We say that an $TPKE$ scheme is semantic secure if for all probabilistic polynomial time algorithm A and negligible function ε, $Adv_{A,u,t}(\lambda) \leq \varepsilon$ and also $AdvCD_{A,u,t}(\lambda) \leq \varepsilon$.

2.5 Security Model for Resplittable Threshold Public Key Encryption

Here security model is adapted from [5, 4]. Security of RTPKE is defined using two properties: semantic security (IND-CPA) and consistency of decryptions.

2.5.1 Semantic Security (IND-CPA)

Semantic security model is almost same as semantic security model of threshold public key encryption. We define security model using a game that is played between the challenger and the adversary. The game proceeds as follows.

Setup: The challenger runs TKeyGen(n,u,t) and gives the public key tpk to adversary.

Split & Corruption Query: The adversary can issue a query for $t-1$ corrupt keys and challenger has to return $t-1$ corrupt keys without knowing master secret key. Adversary can repeat this query polynomial number of times.

Challenge: The adversary submits target message m. Challenger picks a random bit $r \in \{0,1\}$ and a random ciphertext C. If $r = 0$ it sets the challenge ciphertext to $C^* :=$ Encrypt(tpk, m). If $r = 1$ it sets the challenge ciphertext to $C^* := C$. It sends C^* as challenge to the adversary.

Guess: Finally, the adversary outputs a guess $r' \in \{0,1\}$ and wins if $r = r'$.
We refer an adversary $\mathscr{A}$ as an IND-CPA adversary. We define the advantage of the adversary $\mathscr{A}$ in attacking an RTPKE scheme ξ as

$$AdvR_{A,n,t}(\lambda) = |Pr[r = r'] - 1/2|$$

2.5.2 Decryption Consistency:

Here security model captures the idea that an adversary can not produce two sets of valid decryption shares for any ciphertext such that their corresponding (applying combine algo) plaintexts are different. Decryption consistency is defined using following game that is played between the challenger and the adversary. In this game *Setup* and *Split & Corruption Query* steps are same as above game. The adversary then outputs a ciphertext C and two sets of decryption shares $S = \{\mu_1, \ldots, \mu_t\}$ and $S' = \{\mu'_1, \ldots, \mu'_t\}$ each of the size t. The adversary wins the game if:

1. The decryption shares in S and S' are valid decryption shares for C under verification key TVK.
2. S and S' each contain decryption shares from t distinct servers; and
3. TCombine $(tpk, TVK, C, S) \neq TCombine(tpk, TVK, C, S')$.

We define the adversarys advantage in winning this game as $AdvRCD_{A,n,t}(\lambda)$.

Definition 2. We say that an $RTPKE$ scheme is secure if for all the probabilistic polynomial time algorithm A and negligible function ε, $AdvR_{A,n,t}(\lambda) \leq \varepsilon$ and also $AdvRCD_{A,n,t}(\lambda) \leq \varepsilon$.

2.6 Existentially Unforgeable Digital Signature [11]

A digital signature scheme consists of three algorithms.

KeyGeneration(n)**:** On input of security parameter n, this algorithm outputs the secret key sk and corresponding verifying key vk.

Sign(m, sk)**:** On input of message m and secret key sk, this algorithm outputs a signature σ.

Verify(m, σ, vk)**:** On input of message m, signature σ and verifying key vk, this algorithm accepts (signature σ is a valid signature on m under vk) or rejects.

2.6.1 Existentially Unforgeability

A signature scheme is existentially unforgeable if given a polynomial number of message signature pair

$$(m_1, \sigma_1), (m_2, \sigma_2), \ldots, (m_k, \sigma_k)$$

where σ denotes signature on the message m and k is some polynomial on security parameter n then it is hard to generate a pair (m_{k+1}, σ_{k+1}) of any message $m_{k+1} \notin \{m_1, \ldots, m_k\}$.

2.7 Integer Lattices ([12])

A lattice is defined as the set of all integer combinations

$$L(b_1, ..., b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in Z \text{ for } 1 \leq i \leq n \right\}$$

of n linearly independent vectors $b_1, ..., b_n \in R^n$. The set of vectors $\{b_1, ..., b_n\}$ is called a basis for the lattice. A basis can be represented by the matrix $B = [b_1, ..., b_n] \in R^{n \times n}$ having the basis vectors as columns. Using matrix notation, the lattice generated by a matrix $B \in R^{n \times n}$ can be defined as $L(B) = \{Bx : x \in Z^n\}$, where Bx is the usual matrix-vector multiplication. The determinant of a lattice is the absolute value of the determinant of the basis matrix $det(L(B)) = |det(B)|$.

Definition 3. For q prime, $A \in Z_q^{n \times m}$ and $u \in Z_q^n$, define:

$$\Lambda_q(A) := \{e \in Z^m \ \ s.t. \ \ \exists s \in Z_q^n \ where \ A^T s = e \ (mod \ q)\}$$

$$\Lambda_q^{\perp}(A) := \{e \in Z^m \ \ s.t. \ \ Ae = 0 \ (mod \ q)\}$$

$$\Lambda_q^{u}(A) := \{e \in Z^m \ \ s.t. \ \ Ae = u \ (mod \ q)\}$$

2.8 Gram Schmidt Orthogonalization:

$\widetilde{S} := \{\widetilde{s_1}, ..., \widetilde{s_k}\} \subset R^m$ denotes the Gram-Schmidt orthogonalization of the set of linearly independent vectors $S = \{s_1, ..., s_k\} \subset R^m$. It is defined as follows: $\widetilde{s_1} = s_1$ and $\widetilde{s_i}$ is the component of s_i orthogonal to span$(s_1, ..., s_i)$ where $2 \leq i \leq k$. Since $\widetilde{s_i}$ is the component of s_i so $\|\widetilde{s_i}\| \leq \|s_i\|$ for all i.
We refer to $\|\widetilde{S}\|$ as the Gram-Schmidt norm of S.

2.9 Discrete Gaussians

Let L be a subset of Z^m. For any vector $c \in R^m$ and any positive parameter $\sigma \in R > 0$, define:
$\rho_{\sigma,c}(x) = exp(-\pi \frac{\|x-c\|}{\sigma^2})$: a Gaussian-shaped function on R^m with center c and parameter σ,
$\rho_{\sigma,c}(L) = \sum_{x \in L} \rho_{\sigma,c}(x)$: the (always converging) $\rho_{\sigma,c}$ over L,
$D_{L,\sigma,c}$: the discrete Gaussian distribution over L with parameters σ and c,

$$\forall y \in L \ , \ D_{L,\sigma,c} = \frac{\rho_{\sigma,c}(y)}{\rho_{\sigma,c}(L)}$$

The distribution $D_{L,\sigma,c}$ will most often be defined over the Lattice $L = \Lambda_q^{\perp}$ for a matrix $A \in Z_q^{n \times m}$ or over a coset $L = t + \Lambda_q^{\perp}(A)$ where $t \in Z^m$.

Lemma 1 ([12], Lemma 7.1). Let Λ be an m-dimensional lattice. There is a deterministic polynomial-time algorithm ToBasis(S,B) that, given an arbitrary basis B of Λ and a full-rank set $S = \{s_1, \ldots, s_m\}$ in Λ, returns a basis T of Λ satisfying

$$\|\widetilde{T}\| \leq \|\widetilde{S}\| \ \ and \ \ \|T\| \leq \|S\|\sqrt{m}/2$$

.

2.10 The LWE Hardness Assumption ([2, 13])

The LWE (learning with error) hardness assumption is defined by Regev [2].

Definition 4. LWE: Consider a prime q, a positive integer n, and a Gaussian distribution χ^m over Z_q^m. Given $(A, As + x)$ where matrix $A \in Z_q^{m \times n}$ is uniformly random and $x \in \chi^m$.
LWE hard problem is to find s with non-negligible probability.

Definition 5. Decision LWE: Consider a prime q, a positive integer n, and a Gaussian distribution χ^m over Z_q^m. The input is a pair (A, v) from an unspecified challenge oracle O, where $A \in Z_q^{m \times n}$ is chosen uniformly. An unspecified challenge oracle O is either a noisy pseudo-random sampler O_s or a truly random sampler $O_\$$. It is based on how v is chosen.

1. When v is chosen to be $As + e$ for a uniformly chosen $s \in Z_q^n$ and a vector $e \in \chi^m$, an unspecified challenge oracle O is a noisy pseudo-random sampler O_s.

2. When v is chosen uniformly from Z_q^m, an unspecified challenge oracle O is a truly random sampler $O_\$$.

Goal of the adversary is to distinguish between above two cases with non-negligible probability.
Or we say that an algorithm A decides the (Z_q, n, χ)-LWE problem if $|Pr[A^{O_s} = 1] - Pr[A^{O_\$} = 1]|$ is non-negligible for a random $s \in Z_q^n$.

Above decision LWE is also hard even if s is chosen from the Gaussian distribution rather than the uniform distribution.[14, 3]

3 Shamir's Secret Sharing [10]

Shamir's secret sharing divides the data D into u pieces in such a way that at least t- pieces are required to construct the data D but no information about data D is revealed from $t-1$ pieces or less. Adela Georgescu [15] presented *(u,u)* Shamir secret sharing based on LWE assumption. Shamir's secret sharing is based on the following theorem.

Theorem 1 Given t points in the 2- dimensional plane $(x_1,y_1),\ldots,(x_t,y_t)$ with distinct x's, there is one and only one polynomial of degree $t-1$ such that $q(x_i)=y_i$ for all i.

Shamir's sharing protocol. Our goal is to create n- secret shares of the secret s such that at least t shares are required to compute D.

1. Dealer D pick a random $t-1$ degree polynomial $q(x)=a_0+a_1x+,\ldots,+a_{t-1}x^{t-1}$ in which $a_0=s$. Here all coefficients a_i $(0\leq i\leq t-1)$ are from field $(F_p : prime\ p)$.
2. Dealer D computes $q(1),q(2),\ldots,q(u)$ and secretly distributes each player j the share $q(j)$. Hence the shares are denoted as $q(1),q(2),\ldots,q(u)$.

From these t- points we can construct polynomial $q(x)$ of degree $t-1$ and can find secret $s=q(0)$. One can construct polynomial by using Lagrange interpolation.

Lagrange polynomial. Given t points in the 2- dimensional plane $(x_1,y_1),\ldots,(x_t,y_t)$ with distinct x's, then the unique polynomial passing through these points in the Lagrange form is a linear combination

$$q(x)=L(x)=\sum_{i=0}^{t-1}y_il_i(x)$$

of the Lagrange basis polynomials:

$$l_i(x)=\prod_{0\leq m\leq t-1,m\neq i}\frac{(x-x_m)}{(x_i-x_m)}=\frac{(x-x_0)}{(x_i-x_0)}\cdots\frac{(x-x_{i-1})}{(x_i-x_{i-1})}\frac{(x-x_{i+1})}{(x_i-x_{i+1})}\cdots\frac{(x-x_{k-1})}{(x_i-x_{k-1})}$$

Properties of Shamir's secret sharing:

1. **Perfect Security:** Adversary with knowledge of $t-1$ shares or less can not find any information regarding secret.
2. **Homomorphic Property:**
   - If we add/multiply a constant to all secret shares (y-values) then this constant will be added/-multiplied to secret to get new secret.
   - Suppose we have two secrets s and t. Their corresponding shares are $f(1),\ldots,f(n)$ for polynomial $f(x)$ and $g(1),\ldots,g(n)$ for polynomial $g(x)$. Now we define j^{th} share as $f(j)+g(j)(j\in[1\ldots n])$. New secret will be $s+t$ for new function $h(x)=f(x)+g(x)$ since $h(0)=f(0)+g(0)$.

4 Threshold Public Key Encryption Scheme

Our scheme is based on schemes [1, 3]. Now we describe our lattice based efficient threshold public key encryption scheme.

TKeyGen(n,u,t)**:** On input a security parameter n, we set the parameter modulus $q = poly(n)$, a public prime number p such that discrete logarthimic problem is intractable in cyclic group Z_p with generator g.
We choose random vector $(tsk) = (s_1,\ldots,s_n) \in Z_q^n$ from Gaussian distribution χ_{sk}^n with Gaussian parameter sk. We randomly generate n- polynomials $f_i \leftarrow Z_q[x]$ such that $deg(f_i) = t-1$ and $f_i(0) = s_i$ for $1 \le i \le n$. We set threshold secret key $tsk = (s_1,\ldots,s_n)$.
For $i = 1,\ldots,u$ threshold share key for server i is defined as $tsk_i = (s_{i,1},\ldots,s_{i,n}) = (f_1(i),\ldots,f_n(i))$. Corresponding threshold verifying key for server i is $tvk_i = (g^{s_{i,1}},\ldots,g^{s_{i,n}})$. So we define threshold verifying key $TVK = (tvk_1,\ldots,tvk_u)$.
Out of the u players, we make $\binom{u}{t}$ groups of size t. We choose a random integer value K_A where A is group number. This K_A is given to players *not* in given group A. This means every player will receive $\binom{u-1}{t}$ random values. Player i is given tsk_i and $\binom{u-1}{t}$ values. We define threshold secret key for the i^{th} player $= TSK_i = (tsk_i, \binom{u-1}{t}$ values).
We choose a uniformly random matrix $A \in Z_q^{n\times n}$. We compute $b = e - A(tsk)\ mod\ q$, where $e \in Z_q^n$ is an error vector chosen from Gaussian distribution χ_{sk}^n.
So output is share secret keys $TSK_1,\ldots,TSK_n$, threshold public key $tpk = (A,b) \in (Z_q^{n\times n} \times Z_q^n)$ and threshold verifying key $TVK = (tvk_1,\ldots,tvk_n)$.

Encrypt(tpk,γ)**:** To encrypt a bit $\gamma \in \{0,1\}$, we do the following.

- We choose $(sk,vk) \leftarrow\ KeyGeneration(n)$.
- We choose $e = (e_1,e_2,e_3) \in Z^n \times Z^n \times Z$ from Gaussian distribution $\chi_{se}^n \times \chi_{se}^n \times \chi_{se}$ with Gaussian parameter se.
- Compute $c_1 = e_1^T A + e_2^T \in Z_q^n$ and $c_2 = e_1^T b + e_3 + \gamma\lfloor\frac{q}{2}\rfloor \in Z_q$.
- We assign $\sigma \leftarrow\ Sign(sk,c_1,c_2)$.
- Output the ciphertext $C = (c_1,c_2,vk,\sigma)$.

TShareDecryption$(tpk,TSK_i,C = (c_1,c_2,vk,\sigma))$**:** To obtain a share decryption μ_i of a ciphertext C under secret share key TSK_i, we do the following.

- If Verify$(vk,c_1,c_2,\sigma) = \perp$ then output $\mu_i = (i,\perp)$ and exit.
- We parse TSK_i as $TSK_i = (tsk_i, \binom{u-1}{k}$ values) then compute $sd_i = c_2 + c_1 tsk_i$.
- There is a pseudorandom function $\phi_{K_A}(c_1,c_2)$ which takes input as ciphertext (which is common to all players) and key K_A assigned to player and returns a number in the range $[-\sqrt{q},\sqrt{q}]$. We compute a unique $x_i = \sum \phi_{K_A}(c_1,c_2)$: summation over all key K_A assigned to player i.
- We define share decryption of i^{th} player $e_i = (sd_i + x_i)$.
- We output $\mu_i = (e_i, g^{x_i})$ as share decryption part for the i^{th} player.

TShareVerify$(tpk,TVK,\mu_i,C = (c_1,c_2,vk,\sigma))$**:** To verify a share decryption μ_i of a ciphertext C under verification key TVK, we do the following.

- If Verify$(c_1,c_2,\sigma,vk) = \perp$ then output $\mu_i = (i,\perp)$ and exit.

- We parse μ_i and TVK as $\mu_i = (e_i, g^{x_i})$ and $TVK = (tvk_1, \ldots, tvk_n)$ where $tvk_i = (g^{s_{i,1}}, \ldots, g^{s_{i,n}})$. Again we parse $c_1 = (r_1, \ldots, r_n)$.
- if $(g^{e_i} \neq g^{x_i} g^{c_2} (g^{s_{i,1}})^{r_1} \ldots (g^{s_{i,n}})^{r_n})$ then return invalid share decryption otherwise valid share decryption.

TCombine$(tpk, TVK, C = (c_1, c_2, vk, \sigma), \{\mu_1, \ldots \mu_k\})$**:** To obtain a decryption of a ciphertext C given k share decryptions $\mu_1, \ldots \mu_k$, we do the following.

- If there exist i such that $\mu_i = (i, \perp)$ or TShareVerify$(tpk, TVK, \mu_i, C = (c_1, c_2, vk, \sigma)) =$ invalid then output $\perp$ and exit.
- After receiving t decryption parts from the t players we apply Lagrange interpolation using player number as $x-$coordinate and decryption share μ_i as $y-$coordinate. This way we obtain resultant Lagrange polynomial $f(x)$ and then $f(0)$.
- If $f(0)$ is closer to 0 than $\lfloor \frac{q}{2} \rfloor$ *mod* q output 0 otherwise output 1.

Extra term small x_i is added so that adversary can not get any extra information about the share secret key given e_i and ciphertext C.

Correctness: Here correctness is similar to [1, 9]. We know that decryption part from the player i $e_i = sd_i + x_i = c_2 + c_1 tsk_i^T + x_i$. After receiving t decryption parts from the t players we apply Lagrange interpolation using player number as $x-$coordinate and decryption share y_i as $y-$coordinate. Along this we use homomorphic properties of Lagrange interpolation mentioned in section 3. This way we obtain resultant Lagrange polynomial $f(x)$ and $f(0) = c_2 + c_1(tsk) + h(0) = \gamma \lfloor \frac{q}{2} \rfloor + \text{error} + h(0)$, where $h(0)$ is the result of another Lagrange interpolation function $h(x)$ using i as $x-$coordinate and x_i as $y-$coordinate and $h(0)$ will be less than $x_1 + \ldots + x_t$. The *error* term consists of e, e_1, e_2, e_3 and $tsk_1, \ldots, tsk_n$ which are small (drawn from Gaussian distribution).
Our scheme will be correct if $(\text{error} + h(0)) \leq \text{error} + x_1 + \ldots + x_t \leq \frac{q}{4}$. From lemma 3.1 of [3] it is clear that we can choose Gaussian parameters s_e and s_k such that error tolerance of message encoding is greater than $\sqrt{q}$ with probability less than $2^{-O(n)}$. We choose t such that $\binom{u}{t} < \frac{1}{4t}\sqrt{q} - 1$. Since each $\phi_{k_A}(a,b) \in [-\sqrt{q}, \sqrt{q}]$ and we take sum of $\binom{u}{t}$ ϕ_{k_A}'s so $\|x_i\| < \frac{q}{4t} - \sqrt{q}$. We get that $\|e_i + x_i\| < \frac{q}{4t}$ with probability at least $1 - 2^{-O(n)}$. So probability of decryption error is negligible.

Table 1. We compare our scheme with Bendlin and Damgard's scheme [1] for single bit encryption.

	Ciphertext size	Public key size
Bendlin and Damgard [1]	$(n+1)\|Z_q\|$	$(n^2+1)log\ n\|Z_q\|$
Our scheme	$(n+1)\|Z_q\|$	$(n^2+1)\|Z_q\|$

Multi-bit encryption Above scheme can be extended to encrypt $l = poly(n)$ bits by following two ways.

1. **Scheme 1**: We can repeat the encryption for l bits. In this case for l different bits, we will have l different s values but with same public key b. Size of ciphertext $= O(l(nlog\ n + log\ n)) = \tilde{O}(ln)$[1] and size of public key $= O(nlog\ n) = \tilde{O}(n)$.

[1] A function $g(n)$ is in $\tilde{O}(f(n))$ if there exist constants $a, c \geq 0$ such that $g(n) \leq af(n)log^c f(n)$ for all sufficiently large n

2. **Scheme 2**: We can include l independent syndroms $u_1,\ldots,u_l$ in the public key. Now we can use same s for encryption of all l- bits. In this case size of the ciphertext $= O(nlog\ n + llog\ n) = \tilde{O}(n+l)$ and size of public key $= O(lnlog\ n) = \tilde{O}(ln)$[16].

So for multi bit encryption there is a trade-off between size of the ciphertext and the size of the public key.

Table 2. We compare our schemes with Tore Kasper Frderiksen's scheme [9] for l bit encryption.

	Ciphertext size	Public key size
Tore Kasper Frderiksen [9]	$(n+l)\|\|Z_q\|\|$	$l(n^2+1)log\ n\|\|Z_q\|\|$
Our scheme 1	$(n+1)l\|\|Z_q\|\|$	$(n^2+1)\|\|Z_q\|\|$
Our scheme 2	$(n+l)\|\|Z_q\|\|$	$l(n^2+1)\|\|Z_q\|\|$

Theorem 2. Lattice based threshold public key encryption scheme is secure assuming the $LWE_{q,\chi}$ is hard or $Adv_{B,LWE_{q,\chi}}(n) = Adv_{\chi,A}(n)$.

Proof for Semantic Security(IND-CPA): Here proof is similar to proof of theorem 7.2 of [16].
We first show semantic security of TPKE scheme. We will show that if there exist a PPT adversary $\mathscr{A}$ that breaks TPKE scheme with non-negligible probability then there must exist a PPT challenger $\mathscr{B}$ that solves LWE hard problem by simulating views of A.
Initialization. Challenger $\mathscr{B}$ obtains set $S \subset \{1,\ldots,u\}$ of $t-1$ decryption servers from adversary $\mathscr{A}$. Challenger $\mathscr{B}$ choose $t-1$ random vectors $s_1,\ldots,s_{t-1} \in Z_q^n$ from Gaussian distribution χ_{sk}^n and gives these random vectors to adversary $\mathscr{A}$ for secret share of corrupt servers. Challenger obtains $(n+1)$ LWE samples i.e. $(u_i, v_i) \in Z_q^n \times Z_q (0 \le i \le n-1)$ from oracle, the first n samples get parsed as $A = (u_0 \ldots u_{n-1})$ and $(n+1)^{th}$ sample as $b = u_n$. So public key $= (A, b)$.
Challenge: The adversary submits target message bit γ. Now challenger $\mathscr{B}$ proceeds as follows:

1. Set
$$c_1^* = \begin{bmatrix} v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} \in Z_q^n$$

2. Blind the message bit by $c_2^* = v_n + \gamma\lfloor\frac{q}{2}\rceil$.

3. Set $CT^* = (c_1^*, c_2^*)$ and send it to adversary $\mathscr{A}$.

When Oracle O is a pseudo-random LWE oracle then $C_2^* = v_n + \gamma\lfloor\frac{q}{2}\rceil = e_1^T b + e_3 + \gamma\lfloor\frac{q}{2}\rceil$ for some e_1 and e_3. Similarly
$$c_1^* = \begin{bmatrix} v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} = e_1^T A + e_2$$
for some e_1 and e_2. So $CT^* = (c_1^*, c_2^*)$ is a valid encryption of γ for public key b. When Oracle O is a random oracle then $v_0,\ldots,v_{n-1}$ are uniform and therefore $CT^* = (c_1^*, c_2^*)$ is a uniform in $(Z_q^n \times Z_q)$.
Finally adversary $\mathscr{A}$ terminates with some output, challenger $\mathscr{B}$ terminates with same output and ends the simulation. So if adversary $\mathscr{A}$ breaks the scheme then there exists challenger $\mathscr{B}$ which solves LWE hard problem. Hence our scheme is semantic secure.

Decryption Consistency: We prove decryption consistency by contradiction. Suppose adversary can find two sets of valid shares such that decryption leads to different messages. It means two different set of t- points give two different polynomials (or $s = q(0)$) of degree $t-1$, which contradicts theorem 1. Hence $AdvCD_{A,u,k}(\lambda) = 0$. Hence our scheme is secure.

5 Resplittable Threshold Public Key Encryption Scheme

We can construct resplittable TPKE from our efficient TPKE scheme.

TKeyGen(n,u,k)**:** On input a security parameter n, we set the parameter modulus $q = poly(n)$, a public prime number p such that discrete logarthimic problem is intractable in cyclic group Z_p with generator g.
We choose random vector $(tsk) = (s_1,\ldots,s_n) \in Z_q^n$ from Gaussian distribution χ_{sk}^n. We choose a uniformly random matrix $A \in Z_q^{n\times n}$. We compute $b = e - A(tsk)\ mod\ q$, where $e \in Z_q^n$ is an error vector chosen from Gaussian distribution χ_{sk}^n.
So output is threshold public key (tpk)= (A,b) and the threshold secret key $(tsk) = (s_1,\ldots,s_n) \in Z_q^n$.

Encrypt(tpk,γ)**:** To encrypt a bit $\gamma \in \{0,1\}$, we do the following.

- We choose $(sk,vk) \leftarrow KeyGeneration(n)$.
- We choose $e = (e_1,e_2,e_3) \in Z^n \times Z^n \times Z$ from Gaussian distribution $\chi_{se}^n \times \chi_{se}^n \times \chi_{se}$.
- Compute $c_1 = e_1^T A + e_2^T \in Z_q^n$ and $c_2 = e_1^T b + e_3 + \gamma\lfloor\frac{q}{2}\rfloor \in Z_q$.
- We assign $\sigma \leftarrow Sign(sk,c_1,c_2)$.
- Output the ciphertext $C = (c_1,c_2,vk,\sigma)$.

Tsplit$(tsk = s_1,\ldots,s_n)$ We randomly generate n- polynomials $f_i \leftarrow Z_q[x]$ such that $deg(f_i) = k-1$ and $f_i(0) = s_i$ for $1 \le i \le n$. We set threshold secret key $tsk = (s_1,\ldots,s_n)$.
For $i = 1,\ldots,u$ threshold share key for server i is defined as $tsk_i = (s_{i,1},\ldots,s_{i,n}) = (f_1(i),\ldots,f_n(i))$. Corresponding threshold verifying key for server i is $tvk_i = (g^{s_{i,1}},\ldots,g^{s_{i,n}})$. So we define threshold verifying key $TVK = (tvk_1,\ldots,tvk_u)$.
Out of the u players, we make $\binom{u}{t}$ groups of size t. We choose a random integer value K_A where A is group number. This K_A is given to players *not* in given group A. This means every player will receive $\binom{u-1}{t}$ random values. Player i is given tsk_i and $\binom{u-1}{t}$ values.
We define threshold secret key for the i^{th} player $= TSK_i = (tsk_i, \binom{u-1}{t}$ values).
We choose a uniformly random matrix $A \in Z_q^{n\times n}$. We compute $b = e - A(tsk)\ mod\ q$, where $e \in Z_q^n$ is an error vector chosen from Gaussian distribution χ_{sk}^n.
So output is share secret keys $TSK_1,\ldots,TSK_n$.

TShareDecryption$(tpk,TSK_i,C = (c_1,c_2,vk,\sigma))$**:** To obtain a share decryption μ_i of a ciphertext C under secret share key TSK_i, we do the following.

- If Verify$(vk,c_1,c_2,\sigma) = \perp$ then output $\mu_i = (i,\perp)$ and exit.
- We parse TSK_i as $TSK_i = (tsk_i, \binom{u-1}{k}$ values) then compute $sd_i = c_2 + c_1 tsk_i$.

- There is a pseudorandom function $\phi_{K_A}(c_1,c_2)$ which takes input as ciphertext (which is common to all players) and key K_A assigned to player and returns a number in the range $[-\sqrt{q},\sqrt{q}]$. We compute a unique $x_i = \sum \phi_{K_A}(c_1,c_2)$: summation over all key K_A assigned to player i.

- We define share decryption of i^{th} player $e_i = (sd_i + x_i)$.

- We output $\mu_i = (e_i, g^{x_i})$ as share decryption part for the i^{th} player.

TShareVerify$(tpk, TVK, \mu_i, C = (c_1,c_2,vk,\sigma))$**:** To verify a share decryption μ_i of a ciphertext C under verification key TVK, we do the following.

- If Verify$(vk,c_1,c_2,\sigma) = \perp$ then output $\mu_i = (i,\perp)$ and exit.

- We parse μ_i and TVK as $\mu_i = (e_i, g^{x_i})$ and $TVK = (tvk_1,\ldots,tvk_n)$ where $tvk_i = (g^{s_{i,1}},\ldots,g^{s_{i,n}})$. Again we parse $c_1 = (r_1,\ldots,r_n)$.

- if $(g^{e_i} \neq g^{x_i} g^{c_2} (g^{s_{i,1}})^{r_1} \ldots (g^{s_{i,n}})^{r_n})$ then return invalid share decryption otherwise valid share decryption.

TCombine$(tpk, TVK, C = (c_1,c_2,vk,\sigma), \{\mu_1,\ldots\mu_k\})$**:** To obtain a decryption of a ciphertext C given k share decryptions $\mu_1,\ldots\mu_k$, we do the following.

- If there exists i such that $\mu_i = (i,\perp)$ or TShareVerify$(tpk, TVK, \mu_i, C = (c_1,c_2,vk,\sigma)) =$ invalid then output $\perp$ and exit.

- After receiving t decryption parts from the t players we apply Lagrange interpolation using player number as $x-$coordinate and decryption share μ_i as $y-$coordinate. This way we obtain resultant Lagrange polynomial $f(x)$ and then $f(0)$.

- If $f(0)$ is closer to 0 than $\lfloor \frac{q}{2} \rfloor \ mod\ q$ output 0 otherwise output 1.

Theorem 3. Lattice based resplittable threshold public key encryption scheme is secure assuming the $LWE_{q,\chi}$ is hard.

Proof: Here proof is similar to the proof of Theorem 2.

6 Conclusion

We have proved our lattice based threshold public key encryption scheme to be semantically secure (IND-CPA). Finding the application of resplittable threshold public key encryption scheme other than generic construction of proxy re-encryption is an open problem.

Acknowledgments.

We would like to thank anonymous reviewers for their useful comments.

References

[1] R. Bendlin and I. Damgard, "Threshold Decryption and Zero-Knowledge Proofs for Lattice-Based Cryptosystems," in *Proc. of the 7th Theory of Cryptography Conference on Theory of Cryptography (TCC'10), Zurich, Switzerland, LNCS*, vol. 5978. Springer-Verlag, February 2010, pp. 201–218.

[2] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proc. of the 37th Annual ACM Symposium on Theory of Computing (STOC'05), Baltimore, Maryland, USA*. ACM, May 2005, pp. 84–93.

[3] R. Lindner and C. Peikert., "Better key sizes (and attacks) for LWE-based encryption," in *Proc. of the Cryptographers' Track at the RSA Conference 2011 (CT-RSA'11), San Francisco, California, USA, LNCS*, vol. 6558. Springer-Verlag, February 2011, pp. 319–339.

[4] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang, and Y. Zhao, "Generic construction of chosen ciphertext secure proxy re-encryption," in *Proc. of the Cryptographers' Track at the RSA Conference 2012 (CT-RSA'12), San Francisco, California, USA, LNCS*, vol. 7178. Springer-Verlag, February–March 2012, pp. 349–364.

[5] D. Boneh, X. Boyen, and S. Halevi, "Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles," in *Proc. of the Cryptographers' Track at the RSA Conference 2006 (CT-RSA'06), San Jose, California, USA, LNCS*, vol. 3860. Springer-Verlag, February 2006, pp. 226–243.

[6] S. Arita and K. Tsurudome, "Construction of Threshold Public-Key Encryptions through Tag-Based Encryptionsn," in *Proc. of the 7th International Conference on Applied Cryptography and Network Security (ACNS'09), Paris-Rocquencourt, France, LNCS*, vol. 5536. Springer-Verlag, June 2009, pp. 186–200.

[7] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, October 1997.

[8] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *Proc. of the 28th Annual ACM symposium on Theory of computing (STOC'96), Philadelphia, Pennsylvania, USA*. ACM, May 1996, pp. 99–108.

[9] T. K. Frederiksen, "A Multi-bit Threshold Variant of Regev's LWE-based Cryptosystem," January 2011. [Online]. Available: http://daimi.au.dk/~jot2re/lwe/resources/report2.pdf

[10] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[11] C. Dwork and M. Naor, "An Efficient Existentially Unforgeable Signature Scheme and its Applications," *Journal of Cryptology*, vol. 11, no. 3, pp. 187–208, 1998.

[12] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective (The Springer International Series in Engineering and Computer Science)*. Springer, 2002, vol. 671.

[13] S. Agrawal, D. Boneh, and X. Boyen, "Efficient Lattice (H)IBE in the Standard Model," in *Proc. of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10), French Riviera, LNCS*, vol. 6110. Springer-Verlag, May-June 2010, pp. 553–572.

[14] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems," in *Proc. of the 29th Annual International Cryptology Conference (Crypto'09), Santa Barbara, California, USA, LNCS*, vol. 5677. Springer-Verlag, August 2009, pp. 595–618.

[15] A. Georgescu, "A LWE-based Secret Sharing Scheme," in *IJCA Special issue on Network Security and Cryptography*, vol. NSC, no. 3. Foundation of Computer Science, December 2011, pp. 27–29.

[16] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. of the the 40th annual ACM Symposium on Theory of Computing (STOC'08), Victoria, British Columbia, Canada*. ACM, May 2008, pp. 197–206.

Author Biography

Kunwar Singh received the M.Tech degree in Computer Science and Engineering from Jawaharlal University, New Delhi, India in 2003. Currently he is pursuing PhD degree in computer science and engineering from IIT Madras. He is Assistant Professor in Computer Science and Engineering Department at NIT Trichy, India since 2006. Before that he worked in AEC Agra, Uttar Pradesh from 2004 to 2006. His research interest includes Public Key Cryptography, Identity-Based Encryption and Lattice Based Cryptography.

C.Pandu Rangan is a Professor in the department of computer science and engineering of Indian Institute of Technology - Madras, Chennai, India. He heads the Theoretical Computer Science Lab in IIT Madras. His areas of interest are in theoretical computer science mainly focusing on Cryptography, Algorithms and Data Structures, Game Theory, Graph Theory and Distributed Computing.

A.K.Banerjee is a Professor in the Mathematics Department at NIT Trichy, India. His research interest includes Fluid Mechanics and Cryptology. He is the member of Advisory Editorial Board of 'SCIENTIA IRANICA' an International Journal of Science and Technology and the International Journal of Computer Science and Engineering.