EFFICIENT LARGE-SCALE DISTRIBUTED KEY GENERATION AGAINST BURST INTERRUPTION*

Jheng-Ru Ou, Shi-Chun Tsai and Wen-Guey Tzeng
Dept. Computer Science, National Chiao Tung University, Taiwan

Keywords: Distributed key generation, secret sharing, cryptographic protocol.

Abstract: A distributed key generation scheme allows the key servers to distributively share a secret key and then compute the corresponding public key. Canny and Sorkin (Canny and Sorkin, 2004) proposed a *probabilistic* threshold distributed key generation scheme that is suitable for the case that the number of key servers is large. The communication cost of their scheme is much less than that of previous schemes. Nevertheless, it is possible to improve their scheme in some aspects. In this paper we employ the randomness technique to cope with some problems encountered by their scheme. Our contribution is twofold. Firstly, our scheme is secure against a large cluster of dishonest key servers. Secondly, our scheme has better performance in some aspects. We support this point by a series of simulation experiments. As a result, our scheme and Canny and Sorkin's scheme can be used in different situations.

1 INTRODUCTION

The security of a cryptographic scheme usually relies on protecting a secret key. One way to protect such a key is to distribute it to a set of key servers such that each key server holds a key share. Key sharing not only enhances key protection, but also provides a robustness property for the secret key. For example, in a threshold key sharing scheme, a set of key servers over a threshold number can recover the secret key. Even though some servers do not work, the system works.

A *distributed key generation scheme* allows the key servers to distributively share a secret key and then compute the corresponding public key. In this paper we focus on discrete logarithm-based threshold distributed key generation schemes, in which the secret key is x and the public key is $y = g^x \bmod p$. Almost all threshold distributed key generation schemes use *secret sharing schemes* as building blocks. Each key server runs a secret sharing scheme to share its chosen secret to other key servers. Shamir (Shamir, 1979) proposed the first threshold secret sharing scheme based on polynomial interpolation. Feldman (Feldman, 1987) added verification of secret shares (verifiable secret sharing, VSS) to Shamir's scheme. Pedersen (Pedersen, 1991a) further improved the scheme by making the secret shares unconditionally secure.

Based on his verifiable secret sharing scheme, Pedersen (Pedersen, 1991b) proposed a threshold distributed key generation scheme with some important properties that a threshold distributed key generation scheme should have. Gennaro et al. (Gennaro et al., 1999) found that an adversary can bias the distribution of the generated secret key by a subtle maneuver. They then gave a formal definition and proposed a secure scheme. Chu and Tzeng (Chu and Tzeng, 2002) further pointed out that dishonest key servers should not obtain valid key shares to avoid abuse. Canny and Sorkin (Canny and Sorkin, 2004) proposed a *probabilistic* threshold distributed key generation scheme that is suitable for the case that the number n of involved key servers is large, for example, in the level of hundreds or thousands. The main merit of their scheme is that the total number of communications between key servers is greatly reduced from $O(n^2)$

*Supported in part by NSC projects 94-2213-E-009-110 and 95-2221-E-009-031, and Taiwan Information Security Center at NCTU (TWISC@NCTU).

Ou J., Tsai S. and Tzeng W. (2007).
EFFICIENT LARGE-SCALE DISTRIBUTED KEY GENERATION AGAINST BURST INTERRUPTION.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 197-203
DOI: 10.5220/0002122201970203

to $O(nl/\varepsilon^2)$, where l and ε are security and robustness parameters, respectively. Nevertheless, it is possible to improve their scheme in some aspects. Since the arrangement of key servers is very regular, the scheme is vulnerable to a large cluster of dishonest key servers. If the DoS attack occurs to block a cluster of honest key servers from connecting to Internet, the execution of the scheme would fail. See Section 2.2 for the details.

In this paper we employ the randomness technique to cope with the problems encountered by Canny and Sorkin's scheme. We assign non-zero values to *random entries*, while Canny and Sorkin's scheme assigns non-zero values to fixed entries. Our contribution is twofold. Firstly, our scheme is secure against a large cluster of dishonest key servers. Secondly, its performance is better than Canny and Sorkin's method in some aspects. We support this point by a series of simulation experiments. As a result, our scheme and Canny and Sorkin's scheme can be used in different situations.

2 PRELIMINARY

Let $p = 2q+1$ be a large prime, where q is also prime. Let G_q be the subgroup of quadratic residues in Z_p^* and g and h be generators of G_q. Hereafter, the operations used in exponents of g and h are over Z_q. Assume that there are n key servers $S_1, S_2, \ldots, S_n$, and the threshold is t, where $t \leq n \ll q$. A bold character is either a matrix, like $\mathbf{E}$, or a vector, like $\mathbf{a_i}$,

A probabilistic threshold distributed key generation (PTDKG) scheme consists of three stages: setup, key share establishment and public key computation. A PTDKG scheme should satisfy the following conditions.

Definition 1 *An (α, β, δ)-PTDKG scheme should satisfy the following conditions:*

C1. *The key shares of any subset of key servers define the same secret key x, or not at all.*

C2. *Any number of βn key servers can recover the secret key x with probability $1-\delta$ at least.*

C3. *The secret key x is uniformly distributed in Z_q.*

S1. *Any adversary who controls probabilistically up to αn key servers cannot get any information about the secret key x except the information computed from the public key y directly.*

In condition $S1$, it is necessary to assume that the adversary *randomly* picks the controlled key servers. Otherwise, if the adversary chooses the controlled key servers, he can choose those that communicate with the key server S_i and gets the secret share of S_i. Thus, α should be less than r_i/n, where r_i is the number of key servers that communicate with S_i, $1 \leq i \leq n$. If we want smaller r_i (communication cost), the security threshold is smaller.

A typical key share establishment stage consists of two sub-stages:

1. Each key server runs a *secret sharing scheme* to share its chosen secret to other key severs.
2. Each key server combines the received secret shares to form its key share.

In the first sub-stage, dishonest key servers are detected and excluded. In the second sub-stage, the remained honest key servers compute their key shares, which define a unique secret key.

In the following two subsections, we introduce conventional and Canny and Sorkin's approaches for the key share establishment stage.

2.1 Conventional Approaches

We first use the matrix representation to explain Shamir's secret sharing scheme. It corresponds to a $t \times n$-dimensional *evaluation matrix*:

$$\mathbf{E} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 2^{t-1} & \cdots & n^{t-1} \end{bmatrix}.$$

For key share establishment, each key server $S_i, 1 \leq i \leq n$, does the following:

1. Choose a random t-dimensional (secret) vector $\mathbf{a_i} = [a_{i,1}\ a_{i,2}\ \cdots\ a_{i,t}]$.
2. Compute $\mathbf{s_i} = \mathbf{a_i}\mathbf{E} = [s_{i,1}\ s_{i,2}\ \cdots s_{i,n}]$. The operations are over Z_q.
3. Send $s_{i,j}$ to the key server S_j, $1 \leq j \neq i \leq n$.
4. Exclude dishonest key servers and compute a key share x_i from the received $s_{i,j}$, $1 \leq j \leq n$.

In the above the verification messages and steps are ignored for simplicity. Let $H \subseteq \{S_1, S_2, \ldots, S_n\}$ be the set of honest key servers established in the key share establishment stage. Each key server S_j in H computes its key share

$$x_j = \sum_{i \in H} s_{i,j}.$$

The secret key defined by the key shares of the key servers in H is

$$x = \sum_{i \in H} a_{i,1}.$$

Since $\mathbf{s_i} = \mathbf{a_i}\mathbf{E}$, we have

$$(\sum_{i \in H} \mathbf{a_i})\mathbf{E} = \sum_{i \in H} \mathbf{s_i} = [x_1\ x_2\ \cdots\ x_n].$$

For $A = \{S_{i_1}, S_{i_2}, \ldots, S_{i_r}\}$, let $\mathbf{E}^A$ be the matrix with the columns $i_1, i_2, \ldots, i_r$ of $\mathbf{E}$. For example, $\mathbf{E}^{\{S_1,S_3,S_4\}}$ is a $t \times 3$-dimensional matrix that has columns 1, 3 and 4 of $\mathbf{E}$. A set T of key servers from H can recover the secret key x if and only if $\mathbf{E}^T$ has the full rank, i.e., $rank(\mathbf{E}^T) = t$. We can solve x by selecting t independent columns $\mathbf{E}^{T'}$ from $\mathbf{E}^T$, $T' \subseteq T$, and compute

$$\sum_{i \in H} \mathbf{a_i} = (\sum_{i \in H} \mathbf{s_i})^{T'} (\mathbf{E}^{T'})^{-1}. \tag{1}$$

Since any t rows of $\mathbf{E}$ form a Vandermonde matrix, these rows are independent and any t key servers can recover the secret key x, which is the first entry of $\sum_{i \in H} \mathbf{a_i}$. Any set of less than t key servers cannot compute the secret key x. Thus, the above defines a $((t-1)/n, t/n, 0)$-PTDKG scheme.

One disadvantage of the above method is that each key server S_i has to communicate with each other key server. The total number of communications between the key servers is $O(n^2)$, which shall entail heavy network overhead when n is large.

Distributed key generation schemes based on Feldman's and Pedersen's verifiable secret sharing schemes are similar except that the received shares of each key server are verifiable (Feldman, 1987; Pedersen, 1991b).

2.2 Canny and Sorkin's Approach

The idea of Canny and Sorkin to reduce the communication cost is to make $\mathbf{s_i}$ very sparse by choosing an appropriate $\mathbf{E}$. For a zero entry $s_{i,j}$, the key server S_i need not send $s_{i,j}$ to the key server S_j. By this, the communication cost from S_i to S_j is saved. If $\mathbf{s_i}$ is very sparse, the communication cost from S_i to other key servers S_j is much reduced.

Let $\mathbf{E}$ be a $t \times n$-dimensional evaluation matrix with a band of non-zero entries as follows, where $\star$ means a random number in Z_q, which is non-zero overwhelmingly:

$$\mathbf{E} = \begin{bmatrix} \star & \star & \star & \star & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \star & \star & \star & \star & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \star & \star & \star & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \star & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \star & \star & \star \end{bmatrix}$$

Let l be the width of the band and f be the offset of the band between two consecutive rows. For example, the above band matrix has $l = 4$ and $f = 2$. In the scheme, a dealer chooses $\mathbf{E}$ and publishes it. Each key sever S_i chooses a t-dimensional block vector

$$\mathbf{a_i} = \begin{bmatrix} 0 & \cdots & 0 & a_{i,j} & a_{i,j+1} & \cdots & a_{i,j+k-1} & 0 & \cdots & 0 \end{bmatrix}$$

where j is a pre-determined index and k is the block width. The vector $\mathbf{s_i} = \mathbf{a_i}\mathbf{E}$ has only $(k-1)f + l$ non-zero entries. The key server S_i need send non-zero share $s_{i,j}$ to the key servers S_j. With fixed t and n, we can make $(k-1)f + l$ small by tuning parameters k, l and f.

Canny and Sorkin's PTDKG (called CS-PTDKG hereafter) scheme is $(1/f - \varepsilon, 1/f + \varepsilon, \delta)$, for some small ε and δ, $0 < \varepsilon, \delta < 1$. Overall, their method needs $n((k-1)f + l))$ node-to-node communications, while most previous methods need $n(n-1)$ node-to-node communications. They suggest that $l = O(\log n)$ and $k = l/(2\varepsilon^2)$. This saves quite a lot of communications between key servers overall when n is large.

We note that $\mathbf{E}$ and $\mathbf{a_i}$ is very regular and this regularity makes the system vulnerable to burst interruption. For example, if a burst interruption keeps l consecutive key servers from participating the scheme, the scheme does not work even though the number $n - l$ of alive key servers is much larger than $\beta n = (1/f + \varepsilon)n$.

3 OUR CONSTRUCTION

We employ the randomness technique to cope with the problem of burst interruption. We choose $\mathbf{E}$ and $\mathbf{a_i}$ randomly such that it is more robust against burst interruption. To see this, if l consecutive key servers cannot participate, the rest key servers can compute the secret key with high probability.

For each row of $\mathbf{E}$, we randomly choose l entries and assign random values in Z_q to them. For example, the following $\mathbf{E}$ has $t = 3, n = 5$, and $l = 2$:

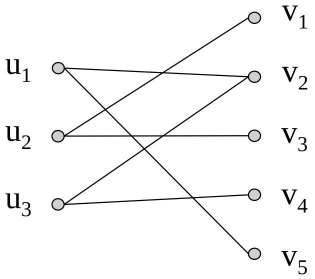
$$\mathbf{E} = \begin{bmatrix} 0 & 1 & 0 & 0 & 5 \\ 3 & 0 & 2 & 0 & 0 \\ 0 & 4 & 0 & 3 & 0 \end{bmatrix}. \tag{2}$$

Each key server $S_i, 1 \le i \le n$, randomly chooses k entries of $\mathbf{a_i}$ and assigns random values in Z_q to them. We see that $\mathbf{s_i} = \mathbf{a_i}\mathbf{E}$ has kl non-zero entries at most. Although the number of non-zero entries is more than $(k-1)f + l$ in the CS-PTDKG scheme if k and l are the same. We shall show that our system needs smaller k and l to achieve the same level of robustness in simulation.

Before presenting our scheme, we need to discuss some theoretical problems concerning the feasibility of our construction. The framework is to consider the probability that $\mathbf{E}'$, which is obtained from $\mathbf{E}$ by deleting some columns randomly, has the full rank. If $rank(\mathbf{E}') = t$, the key shares of honest key servers define the secret key uniquely.

First, the following are some terminologies about graphs. Let U and V be two sets of vertices. A graph $G=(U,V,E)$ is *bipartite* if the edge set $E\subseteq U\times V$, that is, the vertices in U (and V) are not connected. A bipartite graph $G=(U,V,E)$ is *left l-regular* if all vertices in U have degree l. A *perfect matching* for a bipartite graph $G=(U,V,E)$ with $|U|\leq|V|$ is a set of edges $M\subseteq E$ with $|M|=|U|$ such that every vertex $x\in U$ is incident to one edge in M and every vertex $y\in V$ is incident to at most edge in M.

We consider $\mathbf{E}$ as the matrix representation of a bipartite graph $G=(U,V,E)$, where each row is a vertex in U, each column is a vertex in V and $(u,v)\in E$ if the (u,v)-entry of $\mathbf{E}$ is non-zero. Thus, $|U|=t$ and $|V|=n$. For example, the bipartite graph corresponding to the matrix in Equation (2) is:



It is left l-regular since every vertex $u\in U$ has degree l. We see that $M=\{(u_1,v_5),\ (u_2,v_1),\ (u_3,v_2)\}$ is a perfect matching for the graph.

The property of the full rank of $\mathbf{E}$ is related to *perfect matching* of $G=(U,V,E)$, $|U|\leq|V|$. Assume that $M\subseteq E$ is a perfect matching of G. We can use the matching edge $(u,v)\in M$ as the pivot entry (u,v) of $\mathbf{E}$ to eliminate non-zero entries in column v. Furthermore, since the values in non-zero entries are randomly selected from a very large set Z_q, it is very unlikely that the elimination process by a pivot would cause another pivot to be zero. Therefore, the t columns associated with the perfect matching M are independent. We would say that $\mathbf{E}$ has the full rank t if and only if G has a perfect matching. The criteria for a bipartite graph to have a perfect matching is known as Hall's lemma.

Lemma 1 (Hall) *A bipartite graph $G=(U,V,E)$ has a perfect matching from U to V if and only if for every subset $S\subseteq U$, $|\Gamma(S)|\geq|S|$, where $\Gamma(S)$ is the set of S's neighbor vertices in V.*

We show that the probability that a random left l-regular bipartite graph has a perfect matching is close to 1. In the following two theorems, we allow multiple edges in bipartite graphs for a simpler analysis. If no multiple edges are allowed, which is like our construction, the probability of forming a perfect matching is higher. This means that our construction is better than the analyzed one.

Theorem 1 *For appropriate positive integers t,l and n such that, for $3\leq j\leq t$,*

$$\frac{j(j-1)}{(t-j+1)(n-j+2)}(\frac{j-2}{j-1})^{(j-1)l}(\frac{n}{j-1})^{l}\geq 1.$$

The probability that a random left l-regular bipartite graph $G=(U,V,E)$ has a perfect matching is $1-\frac{t^3}{2}(\frac{1}{n})^{2l-1}$ at least , where $|U|=t$, $|V|=n$ and $t\leq n$.

Proof 1 *We compute the probability that the condition in Hall's lemma is not satisfied. For a subset $S\subseteq U$ of j vertices and a subset $T\subseteq V$ of $j-1$ vertices, the probability that all edges from S hit into the set T is*

$$(\frac{j-1}{n})^{jl}.$$

The probability that there is a subset $S\subseteq U$ of j vertices whose edges hit within a subset of fewer than j vertices of V is at most

$$p_j=\binom{t}{j}\binom{n}{j-1}(\frac{j-1}{n})^{jl}.$$

Since, for $3\leq j\leq t$, $p_{j-1}/p_j=$

$$\frac{j(j-1)}{(t-j+1)(n-j+2)}(\frac{j-2}{j-1})^{(j-1)l}(\frac{n}{j-1})^{l}\geq 1,$$

the probability that a left l-regular random bipartite graph does not satisfy Hall's lemma is at most

$$\sum_{j=2}^{t}p_j\leq(t-1)p_2=\frac{t(t-1)^2}{2}(\frac{1}{n})^{2l-1}<\frac{t^3}{2}(\frac{1}{n})^{2l-1}.$$

Thus, the theorem holds.

We notice that the probability can be made arbitrarily small even with rather small l since l is in the exponent of $1/n$ and n is large.

Now, we consider the recoverability of the secret key after the key share establishment stage. After dishonest and unavailable key servers are discarded, a set H of honest key servers is formed. The secret key is computed from the key shares of the key servers in H. The key servers in H can recover the secret key if and only if $\mathbf{E}^H$ has the full rank t, as explained in Equation (1). Assume that H is randomly selected from $\{S_1,S_2,\ldots,S_n\}$. The probability that $\mathbf{E}^H$ has the full rank depends on the size of H. We show that as long as H is not too small, the probability is close to 1.

Let V' (that is, the set H of honest key servers) be a subset of V by randomly deleting m vertices from V. Then, the bipartite graphy $G'=(U,V',E|_{U\cup V'})$ has a perfect matching with an overwhelming probability with proper parameters, where $E|_{U\cup V'}$ is the set of edges incident to vertices in $U\cup V'$.

Theorem 2 *For appropriate positive intgers t,l,m and n such that, for $3 \leq j \leq t$,*

$$\frac{j(j-1)}{(t-j+1)(n-m-j+2)}\cdot (\frac{j-2+m}{j-1+m})^{(j-1)l}(\frac{n}{j-1+m})^{l} \geq 1.$$

Let $G=(U,V,E)$ be a left l-regular random bipartitate graph. After deteting random m vertices from V, the probability that the remainded bipartite graph has a perfect matching is $1-(n-m)\frac{t(t-1)^2}{2}(\frac{m+1}{n})^{2l}$ at least , where $|U|=t$, $|V|=n$ and $t \leq n$.

Proof 2 *Let V' be the subset of V after deleting m vertices, where $|V'|=n'=n-m$. An edge from a vertex in U that hits a vertex in $V-V'$ makes no contribution to Hall's lemma. For a subset $S \subseteq U$ of j vertices and $T \subseteq V'$ of $j-1$ vertices, the probability that Hall's lemma does not hold on S to T is*

$$(\frac{j-1+m}{n})^{jl}.$$

Thus, the probability that there is a subset of $S \subseteq U$ of j vertices whose edges hit a subset of fewer than $j-1$ vertices in V' or $V-V'$ is at most

$$p_j = \binom{t}{j}\binom{n-m}{j-1}(\frac{j-1+m}{n})^{jl}$$

Since

$$\frac{p_{j-1}}{p_j} = \frac{j(j-1)}{(t-j+1)(n-m-j+2)}\cdot (\frac{j-2+m}{j-1+m})^{(j-1)l}(\frac{n}{j-1+m})^{l} \geq 1,$$

we have

$$\sum_{j=2}^{t} p_j \leq (t-1)p_2 = \frac{t(t-1)^2}{2}(n-m)(\frac{m+1}{n})^{2l},$$

which is an upper bound for the proability that Hall's lemma fails.

3.1 Our Distributed Key Generation Scheme

The structure of our scheme is based on Gennaro et al.'s study on secure distributed key generation (Gennaro et al., 1999). Their scheme is secure against the attack of skewing the secret key distribution by dishonest key servers. Note that the key shares of their scheme are unconditionally secure.

At beginning, a dealer chooses a $t \times n$-dimensional evaluation matrix $\mathbf{E}$ and publishes it in a public bulletin board. Our distributed key generation scheme is as follows:

Setup:

1. A dealer does:
   (a) Select a large prime $p = 2q+1$, where q is also prime.
   (b) Compute generators g and h of G_q, where $G_q = \{a^2 \mid a \in Z_p^*\}$ is the subgroup of quadratic residues of Z_p^*.
   (c) Choose a $t \times n$-dimensional evaluation matrix $\mathbf{E}$ such that each row has l non-zero entries.

Key share establishment:

1. Each key server S_i does the following:
   (a) Select two t-dimensional vectors $\mathbf{a_i}$ and $\mathbf{a'_i}$ which each consists of k non-zero random entries. The non-zero entries are in the same indexes of $\mathbf{a_i}$ and $\mathbf{a'_i}$.
   (b) Compute $\mathbf{s_i} = \mathbf{a_i}\mathbf{E}$, $\mathbf{s'_i} = \mathbf{a'_i}\mathbf{E}$ and the set of his communication key servers $Q_i = \{j \mid s_{i,j} \neq 0 \vee s'_{i,j} \neq 0\}$.
   (c) Send $s_{i,j}$ and $s'_{i,j}$ to key server S_j via a secure channel, $j \in Q_i$.
   (d) Broadcast $C_{i,j} = g^{a_{i,j}}h^{a'_{i,j}} \bmod p$, $1 \leq j \leq t$, to all the key servers in Q_i.
2. Each key server S_j does the following:
   (a) Check validity of the received shares, for each i, $j \in Q_i$,

   $$g^{s_{i,j}}h^{s'_{i,j}} \equiv \prod_{k=1}^{t} C_{i,k}^{\mathbf{E}_{k,j}} \pmod{p}. \quad (3)$$

   If the check fails for i, S_j broadcasts a complaint against S_i to the key servers in Q_j.
   (b) If S_j is complained by S_i, it sends $s_{j,i}$ and $s'_{j,i}$ to the key servers in Q_j.
   The other key servers in Q_j check validity of $s_{j,i}$ and $s'_{j,i}$ by Equation (3).
   If S_j fails the test, it is marked as "dishonest" by the key servers in Q_j.
3. Each key server S_j builds a set H of honest key servers and sets his key share as

   $$x_j = \sum_{i \in H, j \in Q_i} s_{i,j} \bmod q,$$

   which is the jth entry of $(\sum_{i \in H}\mathbf{a_i})\mathbf{E}$. Note that the secret key is

   $$x = (\sum_{i \in H}\mathbf{a_i}) \cdot \vec{\mathbf{1}},$$

   where $\vec{\mathbf{1}} = [1\ 1\ \cdots\ 1]$.

Public-key computation:

1. Each key server $S_i \in H$ broadcasts $A_{i,k} = g^{a_{i,k}} \bmod p$, $1 \leq k \leq t$, to the key servers in H.

2. Each key server S_j in Q_i checks validity of $A_{i,k}$ by verifying whether
$$g^{s_{i,j}} \equiv \prod_{k=1}^{t} A_{i,k}^{\mathbf{E}_{k,j}} \pmod{p}. \tag{4}$$
If the check fails, S_j broadcasts a compliant against S_i and sends $s_{i,j}$ and $s'_{i,j}$ to the key servers in Q_i.
3. If S_i is ever complained, all the key servers in Q_i reconstruct $\mathbf{a_i}$ by solving $\mathbf{s_i} = \mathbf{a_i}\mathbf{E}$ and compute correct $A_{i,k}$, $1 \le k \le t$.
4. Then, each key server in H computes the public key as
$$y = \prod_{i\in H}\prod_{j=1}^{t} A_{i,j} \bmod p = g^{(\sum_{i\in G}\mathbf{a_i})\cdot\vec{\mathbf{1}}} \bmod p.$$

Secret key recovery: Note that in some situations, we don't need to recover the secret key x to finish a task. Only each S_i computes a partial result from its key share x_i.

1. Let T be the set of shown-up key servers in H. If E^T is full-ranked, solve $\sum_{i\in H}\mathbf{a_i}$ by the system of equations
$$(\sum_{i\in H}\mathbf{a_i})\mathbf{E}^T = \sum_{i\in H}\mathbf{s_i}.$$
2. The secret key is $x = \sum_{i\in H}\mathbf{a_i}\cdot\vec{\mathbf{1}}$.

3.2 Analysis

The correctness and security of our scheme is shown in the following theorem.

Theorem 3 *Assume that n, t, l, and m satisfy the condition in Theorem 2. The scheme in Section 3.1 is a secure $(\frac{1}{l} - \varepsilon, 1 - \frac{m}{n}, (n-m)\frac{t(t-1)^2}{2}(\frac{m+1}{n})^{2l})$-PTDKG scheme for some small ε, $0 < \varepsilon < 1$.*

Proof 3 *(Sketch) Correctness follows from the results of Gennaro et al. (Gennaro et al., 1999) almost in the same way.*

The bounds $\beta = 1 - m/n$ and $\delta = (n-m)(t(t-1)^2/2)((m+1)/n)^{2l}$ are from Theorem 2 directly. For $\alpha = 1/l - \varepsilon$, each Q_i contains kl key servers at most. Any adversary who controls up to a random fraction α of them contains less than k dishonest key servers in Q_i in average. Since there are k unknown entries in each $\mathbf{a_i}$, the adversary who controls less than k key servers in Q_i cannot know the information about $\mathbf{a_i}$.

For the uniform distribution of x over Z_q, we construct a simulator for the scheme. The details are deferred to the full paper.

4 EXPERIMENTS AND COMPARISON

We first analyze the probability that the full rank is achieved after deleting about a half of key servers. Recall that l is the band of $\mathbf{E}$, f is the offset, and t is the number of rows.

The choice of parameters affects the communication cost of the scheme. We discuss the parameters first. For the CS-PTDKG scheme, due to the arrangement of $\mathbf{E}$, the number of rows is fixed to $t = (n-l)/f$. On allowing $n(1/2-\varepsilon)$ dishonest key servers (eg., $\varepsilon = 1/10$), Canny and Sorkin suggests $f = 2$, $l = 17\log n$ and $t = (n - 17\log n)/2$. Theoretically, the probability of achieving the full rank is $O(n^{-2})$.

For our PTDKG scheme, we shall do some simulation experiments to obtain appropriate l' on the condition that the probability of achieving the full rank is the same as that of the CS-PTDKG scheme.

We take $n = 1000$ and delete about $m = 500$ dishonest key servers randomly. We consider different offsets ($f = 2$, $f = 3$, and $f = 4$) for the CS-PTDKG scheme. The results are shown in Figures 1-3. In each figure, the y-axis indicates the probability of achieving the full rank and the x-axis indicates the number l of non-zero entries in each row of $\mathbf{E}$. The probability is computed by randomly sampling 500 key servers as ”dishonest” many times. We summarize the comparison results in Table 1 on 90% of achieving the full rank. From the table, we can see that the number l' of non-zero entries in each row of our $\mathbf{E}$ is much smaller than that (l) of the CS-PTDKG scheme.

Table 1: Comparison of l with 90% of achieving the full rank. There are $n = 1000$ key servers and m=500 of them are dishonest.

	$t = 408$	$t = 318$	$t = 242$
	$(f = 2)$	$(f = 3)$	$(f = 4)$
CS-PTDKG	$l = 185$	$l = 45$	$l = 33$
Ours	$l' = 14$	$l' = 11$	$l' = 8$

Communication cost. The total communication cost of our scheme is $k'l'n$ and that of the CS-PTDKG scheme is $((k-1)f + l)n$. If we want our scheme to have the same communication cost as that of the CS-PTDKG scheme, we set $k' = ((k-1)f + l)/l'$, the number of non-zero entries in each $\mathbf{a}_i$ of our scheme.

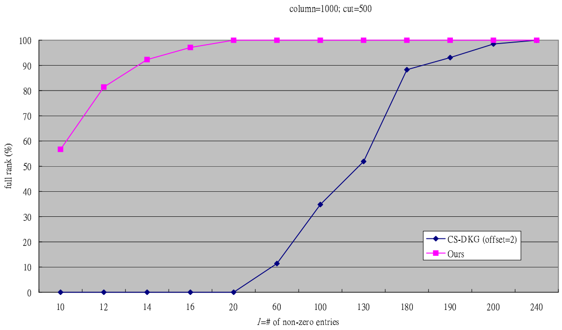


Figure 1: Probability of achieving the full rank for different l, when $f = 2$.

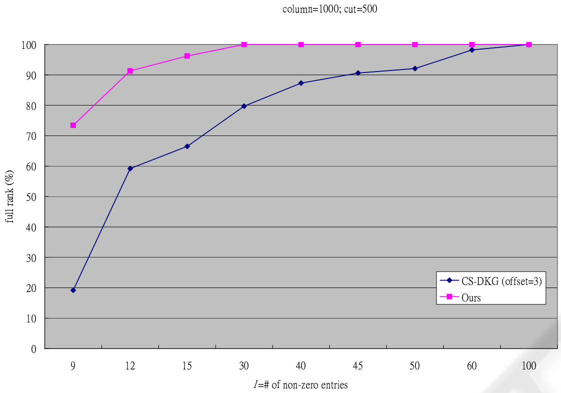


Figure 2: Probability of achieving the full rank for different l, when $f = 3$.

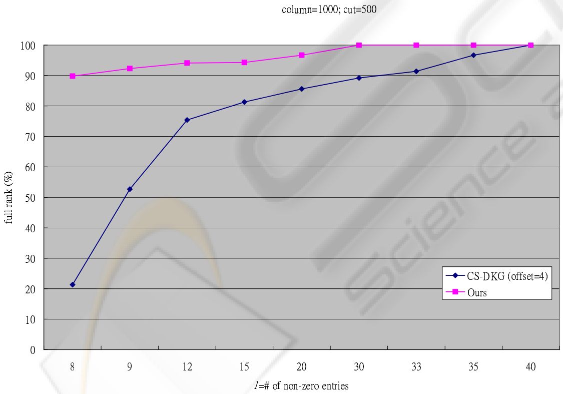


Figure 3: Probability of achieving the full rank for different l, when $f = 4$.

5 DISCUSSION

Our scheme and the CS-PTDKG scheme have different security parameters. For ours, $\alpha = 1/l' - \varepsilon$ and $\beta = 1 - m/n$. For the CS-PTDKG scheme, $\alpha = 1/f - \varepsilon$ and $\beta = 1/f + \varepsilon$. These two set of parameters can be used for different situations. For example, if the number of dishonest key server is relatively small (about one in l' key servers), our scheme is suitable. Since we are dealing with a large number of key servers, a small percent of dishonest key servers is very likely. Our β is adjustable under some constraints. If larger β is desirable, our scheme provides such choice.

REFERENCES

Canny, J. and Sorkin, S. (2004). Practical large-scale distributed key generation. In *Proceedings of Advances in Cryptology - EUROCRYPT '04*, volume 3027 of *LNCS*, pages 138–152. Springer-Verlag.

Chu, C.-K. and Tzeng, W.-G. (2002). Distributed key generation as a component of an integrated protocol. In *Proceedings of the 4th Information and Communications Security - ICICS '02*, volume 2513 of *LNCS*, pages 411–421. Springer-Verlag.

Feldman, P. (1987). A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 427–437. IEEE.

Gennaro, R., Jarecki, S., Krawczyk, H., and Rabin, T. (1999). Secure distributed key generation for discrete-log based cryptosystems. In *Proceedings of Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 295–310. Springer-Verlag.

Pedersen, T. P. (1991a). Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of Advances in Cryptology - CRYPTO '91*, volume 576 of *LNCS*, pages 129–140. Springer-Verlag.

Pedersen, T. P. (1991b). A threshold cryptosystem without a trusted party. In *Proceedings of Advances in Cryptology - EUROCRYPT '91*, volume 547 of *LNCS*, pages 522–526. Springer-Verlag.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.