A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications

Eiichiro FUJISAKI and Tatsuaki OKAMOTO

NTT Laboratories,
1-1 Hikarinooka, Yokosuka-shi, 239-0847 Japan
Email: {fujisaki, okamoto}@sucaba.isl.ntt.co.jp

Abstract. A publicly verifiable secret sharing (PVSS) scheme, named by Stadler in [Sta96], is a special VSS scheme in which anyone, not only the shareholders, can verify that the secret shares are correctly distributed. The property of public verifiability is what the first proposed VSS scheme [CGMA85] incorporated but later protocols [GMW87, Fel87, Ped91] failed to include. PVSS can provide some interesting properties in the systems using VSS. For instance, it gives a practical solution to (k, l)-threshold VSS assuming no broadcast channel. Stadler proposed two PVSS protocols: one is as secure as the Decision-Diffie-Hellman problem and the other is not formally discussed about security. This paper presents a practical and provably secure PVSS scheme which is $O(|v|)$ times more efficient than Stadler's PVSS schemes where $|v|$ denotes the size of the secret. It can be incorporated into various cryptosystems based on the factoring and the discrete logarithm to transform them into publicly verifiable key escrow (PVKE) systems. In addition, those key escrow cryptosystems can be easily modified into the verifiable partial key escrow (VPKE) ones with the property of delayed recovery [BG97]. To the best of our knowledge, this is the first realization of a VPKE cryptosystem based on the factoring with the delayed recovery.

1 Introduction

1.1 (Verifiable) Secret Sharing

Secret sharing (including verifiable secret sharing) is one of the most important tools in modern cryptography. The concept and first realization of secret sharing were presented independently in [Sha79] and in [Bla79]. In a secret sharing (SS) scheme, there exist a dealer and l shareholders. The dealer splits a secret, s, into l different pieces, called *shares*, and sends each share to each shareholder through the private secure channels. If the dealer is honest, the collaboration of any number of shareholders more than or equal to k can recover s while if the dealer is dishonest, the collaboration of any k shareholders fails to retrieve the unique secret, s. Here k is called a threshold if any $(k-1)$ shareholders cannot recover the secret.

Verifiable secret sharing (VSS) was proposed first in [CGMA85] to overcome the problem of dishonest dealers. In a VSS scheme, the shareholders can verify the validity of their shares, that is, they can be convinced that any group of more than or equal to k shareholders can recover a unique secret. VSS is known to play essential roles in various cryptographic protocols such as the multi-party protocols [BGW88, CCD88], key-escrow cryptosystems [Mic92], and threshold cryptography. A VSS protocol is called *non-interactive* if shareholders can verify their shares without talking to each other or the dealer. Feldman and Pedersen contributed to non-interactiveness and improved efficiency [Fel87, Ped91].

1.2 Publicly Verifiable Secret Sharing and Previous Proposal

The first proposed VSS scheme [CGMA85] has the special property that anyone, not only the shareholders, can verify that the shares were correctly distributed to the shareholders. However, this property was lost in later efficient protocols [Fel87, Ped91]. In these schemes, each shareholder can verify the validity of only his own share.

Recently, Stadler has paid attention to the *lost* property. In [Sta96], the property was called ***public verifiability*** and the VSS schemes with the above property were named ***publicly verifiable secret sharing*** (PVSS) schemes.

Informally speaking, PVSS schemes require public-key encryption functions, $E_1, \ldots, E_l$, assigned to every shareholder P_i. The dealer encrypts the shares, $(E_1(s_1), \ldots, E_l(s_l))$, and sends them to a verifier (not only the shareholders). The dealer demonstrates to the verifier the validity of the encrypted shares without revealing $(s_1, \ldots, s_l)$, and if possible, without revealing any additional information. From this model, clearly, the security for the dealer can not be higher than that of the encryption schemes, E. Therefore security should be treated prudently in PVSS schemes.

In [Sta96], Stadler presented two PVSS protocols: one is based on a discrete logarithm and the other is based on the RSA root problem. The first one is provably secure assuming the intractability of the Decision-Diffie-Hellman (DDH) problem while the other was not formally discussed about security.

1.3 Our Results

Although Stadler's PVSS schemes are certainly more efficient than the first proposed one [CGMA85], they are still inefficient. The efficiency of his schemes is $O((k+l)|v| + l|v|^2)$ where $|v|$ denotes the size of the secret and l denotes the number of shareholders.

We show a more practical PVSS scheme whose complexity is estimated to be $O((k+2l)|v|)$. The validity against a dishonest dealer holds on a modified RSA Assumption and the security against the adversary who can obtain the shares of up to $(k-1)$ shareholders holds on a reasonable assumption (Assumption 8). It can be easily modified to yield publicly verifiable key escrow cryptosystems and publicly verifiable *partial* key escrow ones with the property of delayed recovery [BG97].

2 Notation

Through out this paper, $\mathbb{Z}_N$ denotes the residue class ring modulo N, and $\mathbb{Z}_N^*$ denotes the multiplicative group of invertible elements in $\mathbb{Z}_N$. Let $\phi(N)$ be the number of $\mathbb{Z}_N^*$ (the Eulerian number of N). Let $(\frac{x}{N})$ be the Jacobi symbol of x over N where $x, N \in \mathbb{Z}$ and $QR(N)$ denotes the set of quadratic residue modulo N. "$x \in_R S$" means uniformly choosing a random element, x, from a set, S. $|\cdot|$ denotes binary length. $[\, a,\ b\ )$ denotes $\{x | x \in \mathbb{Z}, a \leq x < b\}$. Other symbols and definitions will be set as needed.

3 Publicly Verifiable Secret Sharing

3.1 The Model of PVSS

In this section, we define the threshold model of PVSS. We don't describe the general access structure model because it is not our aim.

A PVSS scheme is composed of three entities, a dealer, shareholders, and a verifier. Let U be a dealer, $P_1, \ldots, P_l$ be l shareholders, and V be a verifier where V is not required to be one of the shareholders.

Let MES be a space of secret s and SHA be a space of shares. $\mathcal{A}_k$ denotes a family of the shareholders defined by $\mathcal{A}_k := \{A | A \subseteq \{P_1, \ldots, P_l\} \wedge \#A = k\}$. Let $Share : MES \rightarrow SHA^l$ be a probabilistic poly-time algorithm and $Recover : SHA^k \rightarrow MES$ be a deterministic poly-time algorithm. Let $(\mathcal{E}, \mathcal{D})$ be a public-key cryptosystem: $\mathcal{E}$ denotes a set of the encryption functions and $\mathcal{D}$ denotes the set of the corresponding decryption functions. $E \in \mathcal{E}$ and $D \in \mathcal{D}$ are defined by E: $SHA \rightarrow CYPH_E$ and $D \in \mathcal{D}$: $CYPH_E \rightarrow MES$ where $CYPH_E$ is the image of $E(SHA)$. Here assume that $E_1, \ldots, E_l \in \mathcal{E}$ are assigned to the shareholders, $P_1, \ldots, P_l$ respectively. We denote $CYPH_{E_1} \times \cdots \times CYPH_{E_l}$ by $CYPH^l$. $PubVerify : CYPH^l \rightarrow \{0, 1\}$ is defined by a polynomial bounded interactive algorithm between U and V, denoted by (U, V), such that $PubVerify(E_1(s_1), \ldots, E_l(s_l))$ outputs 1 if the interactive algorithm, (U, V), accepts $(E_1(s_1), \ldots, E_l(s_l))$, otherwise outputs 0.

In a PVSS scheme, the dealer runs algorithm *Share* and divides secret s into l shares:

$$Share(s) = (s_1, \ldots, s_l).$$

He sends $(E_1(s_1), \ldots, E_l(s_l))$ to the verifier. The verifier checks

$$PubVerify(E_1(s_1), \ldots, E_l(s_l)) = 1.$$

A PVSS scheme is called *non-interactive* if V can verify the validity without talking to any shareholder or the dealer. Here we denote $(\overline{s'}_1, \ldots, \overline{s'}_l)$ by $\overline{\mathbf{s}'}$ where $\overline{s'}_i \in CYPH_{E_i}$. If $PubVerify(\overline{\mathbf{s}'}) = 1$, a unique secret, s', should be recovered by the collaboration of any k shareholders with overwhelming probability, that is,

$$\forall A \in \mathcal{A}_k : Recover(\{D_i(\overline{s'}_i) | P_i \in A\}) = s',$$

where $D_i \in \mathcal{D}$ (called *public verifiability*). If $\overline{\mathbf{s}'} = (E_1(s_1), \ldots, E_l(s_l))$, s' is equivalent to the dealer's secret, s (called *correctness*). Here, in the *PubVerify* procedure, the verifier uses public parameters such as the public key of E (say pk) and parameters for secret sharing (say ssp). In the *Recover* procedure, each shareholder uses a secret key (say sk).

Clearly, the security for the dealer depends on the difficulty of inverting of encryption function family $\mathcal{E}$. In PVSS schemes, the most secure case exists when we can prove that the conspiracy of any less than k shareholders can not compute secret s as long as they can not invert any $E_j(s_j)$ where P_j doesn't join the conspiracy. So as to formalize this situation, we define the hard instance generators and the adversary model and then introduce the *secrecy* condition.

Definition 1. Let G_E be a instance generator which on input $1^{|MES|}$ outputs $((pk, sk), ssp, y)$ where $y \in CYPH_{E_{pk}}$. $CYPH_{E_{pk}}$ is defined by encryption function E_{pk} and parameter for secret sharing ssp (ssp defines SHA and $CYPH_{E_{pk}} := E_{pk}(SHA)$). We say G_E is *hard* if every probabilistic polynomial time circuit family $\mathcal{C}$, for given (pk, ssp, y) (taken from the distribution of G_E), computes $D_{sk}(y)$ with negligible probability ϵ in $|MES|$, that is

$$G_E \text{ is } \textit{hard} \text{ if } \forall \mathcal{C} : \Pr[\mathcal{C}(pk, ssp, y) \rightarrow D_{sk}(y)] < \epsilon.$$

The probability is taken over the distributions of G_E and $\mathcal{C}$.

$\mathcal{ADV}$ denotes the set of the adversaries that can obtain up to $(k-1)$ shares from shareholders and output secret s. More formally, it is defined as follows:

Definition 2. $\mathcal{ADV}$ is defined as the set of the adversaries such that $ADV \in \mathcal{ADV}$ is a probabilistic polynomial-time interactive algorithm which on input $\bar{s} := (\bar{s_1}, \ldots, \bar{s_l}) \in CYPH^l$ works as follows:

1. ADV executes, with dealer U, $PubVerify(\bar{s})$ as a verifier.
2. ADV is allowed to ask oracle $\mathcal{O}$ up to $(k-1)$ queries at any time, namely, ADV can ask the inverses of up to $(k-1)$ elements in $\bar{s}$ before, during, and after the execution of $PubVerify(\bar{s})$.
3. ADV goes the following step if oracle $\mathcal{O}$ returns the inverses of the queries otherwise ADV quits.
4. If $PubVerify(\bar{s}) = 1$, ADV outputs $s \in R(\bar{s})$ where

$$R(\bar{s}) := \{s | \exists A \in \mathcal{A}_k : Recover(\{D_i(\bar{s}_i) | P_i \in A\}) = s\}.$$

we denote this adversary model by $ADV^{U,\mathcal{O}}(\bar{s})$.

A PVSS scheme has the *secrecy* condition if the probability that $ADV^{U,\mathcal{O}}(\bar{s})$ outputs $s \in R(\bar{s})$ is negligible in $|MES|$ where the distribution of $\bar{s}$ follows that of a hard instance generator G_E.

We define a secure (k,l)-publicly verifiable secret sharing scheme as follows:

Definition 3. $(Share, PubVerify, Recover)$ is a *secure* (k,l)-publicly verifiable secret sharing scheme if

- **(Correctness)** $\forall s \in MES$:

$$Share(s) = (s_1, \ldots, s_l) \Longrightarrow PubVerify(E_1(s_1), \ldots, E_l(s_l)) = 1,$$

- **(Public Verifiability)** $\forall \bar{s} \in CYPH^l$, $\exists s \in MES$, $\forall A \in \mathcal{A}_k$:

$$\Pr[PubVerify(\bar{s}) = 1] < \Pr[Recover(\{D_i(\bar{s}_i) | P_i \in A\}) = s] + \epsilon,$$

  where ϵ is negligible in $|MES|$, and D_i is the decryption function corresponding to E_i (The probability is taken over the coin tosses of U and V), and
- **(Secrecy)** $\exists$ hard G_E, $\forall \mathcal{C}$, $\forall ADV \in \mathcal{ADV}$:

$$\Pr[ADV^{U,\mathcal{O}}(\bar{s}) \in R(\bar{s})] < \Pr[\mathcal{C}(pk, ssp, \bar{s}) \rightarrow D_{sk}(\bar{s})] + \epsilon,$$

  where ϵ is negligible in $|MES|$ and the distributions of $\bar{s}$, pk, ssp, and $\bar{s}$ are equivalent that of hard instance generator G_E. The probability is taken over the coin tosses of G_E, U, and ADV.

3.2 Applications of PVSS

The VSS schemes proposed by Feldman and Pedersen in [Fel87, Ped91] are extremely efficient. Their schemes are even non-interactive. They are utilized in various cryptographic protocols because of their efficiency and non-interactiveness. However, considering some applications, they also have some demerits.

Assume that U is a dealer and $P_1, \dots, P_l$ are l shareholders in a network. Here suppose that they do not have any broadcast channel nor any bulletin board. In this situation, it seems to be difficult to construct a practical (k,l)-threshold VSS scheme by only using the techniques of [Fel87, Ped91]. Public verifiability can resolve the problem.

We consider other applications. Assume a public-key cryptosystem with the key escrow property. Let U be a user of the network, V be the network manager or the Government and the other shareholders, $P_1, \dots, P_l$, be escrow agencies. In the key escrow cryptosystems using the ordinary VSS schemes, V has to interact with escrow agencies every time each user U first joins in the network. Publicly verifiable secret sharing (PVSS) schemes can resolve this problem. V never have to wait for any acknowledgments from escrow agencies, while V itself can verify the validity of the shares.

Moreover, the PVSS schemes can construct key escrow cryptosystems with hierarchy. This means that there exist classes of escrow agents and the secret key of a lower escrow agent is escrowed by upper class escrow agents.

4 Building Blocks

In this section, we prepare some building blocks for our protocol. By the space limitation, some protocols are described in Appendix. First of all, we set up two commitment schemes.

Set-up: A verifier (or a trusted third party) generates (N, b, g_1, g_2) and a security parameter $m(= O(|N|))$, where $N = PQ$ where $p = (P-1)/2$ and $q = (Q-1)/2$ $(p \neq q)$ are primes and $< b > = < g_1 > = < g_2 > = G_{pq}$. The prover or the verifier decides v where v is co-prime to $\phi(N)$. The verifier proves that b, g_1, g_2 have the same order[FO97] and he knows $b^{1/v}$ by using zero-knowledge protocols [BCDG87, FFS88].

Commitment: Three different forms of the commitments for secret s are utilized: $BC_{(b,v)}(s,r) := b^s r^v \bmod N$, $\overline{BC}_{(b,g_1)}(s,r_1) := b^s g_1^{r_1} \bmod N$, and $\overline{BC}_{(b,g_1,g_2)}(s, r_1, r_2) := b^s g_1^{r_1} g_2^{r_2} \bmod N$, where $r \in \mathbb{Z}_N^*$, $r_1, r_2 \in \mathbb{Z}$.

The first commitment scheme, BC, has appeared in some previous papers and it is well known that opening the commitment with different representations is equivalent to breaking RSA. The latter commitment scheme, $\overline{BC}$, was presented in [FO97]. In [FO97], it was proved that opening the commitment with different representations is as hard as breaking the factoring.

In Crypto'97, [FO97] presented practical statistical zero-knowledge protocols to prove modular polynomial relations in the secrets by using commitment $\overline{BC}$. The following protocols are an improvement. By using commitment BC, the protocols can demonstrate the statement above more efficiently and in a perfect zero-knowledge manner (if including the set-up protocol) [Fuj98]. Those protocols are as secure against cheating provers as the RSA assumption. See Appendix for the actual protocols.

Checking Protocol: a (perfect) witness hiding protocol between a prover and a verifier in which the prover, on input $((N,b,v),c)$, convinces the verifier that he knows (x,r) such that $c = BC_{(b,v)}(x,r)$. We denote the checking protocol by $CHCK_{(b,v)}(c)$.

Mod-Multi Protocol: a (perfect) witness hiding protocol between a prover and a verifier in which the prover, on input $((N,b,v)c_1,c_2,c_3)$, convinces the verifier that he knows $(x_1,x_2,x_3,r_1,r_2,r_3)$ such that $x_3 \equiv x_1x_2 \pmod v$, $c_1 = BC_{(b,v)}(x_1,r_1)$, $c_2 = BC_{(b,v)}(x_2,r_2)$ and $c_3 = BC_{(b,v)}(x_3,r_3)$. We denote the mod-multi protocol by $MUL_{(b,v)}$
$(c_1,c_2;c_3)$.

Squaring Protocol: a (perfect) witness hiding protocol between a prover and a verifier in which the prover, on input $((N,b,v)c_1,c_2)$, convinces the verifier that he knows (x_1,x_2,r_1,r_2) such that $x_2 \equiv x_1^2 \pmod v$, $c_1 = BC_{(b,v)}(x_1,r_1)$, and $c_2 = BC_{(b,v)}(x_2,r_2)$. We denotes the squaring protocol by $SQU_{(b,v)}(c_1;c_2)$.

PROOF$[c = BC(s) \wedge f(s) = 0\ ]$: a (perfect) witness hiding protocol between a prover and a verifier in which the prover, on input $((N,b,v),c)$, convinces the verifier that he knows (s,r) such that $f(s) = 0 \bmod v$ and $c = BC_{(b,v)}(s,r)$.

Example 1. Suppose that $f(X) = X^3 - a \pmod v$. **PROOF**$[c = BC(s) \wedge f(s) = 0]$ is executed as follows: The prover sets $(c',c'') = (BC(s^2), BC(s^3))$ and executes with a verifier $SQR_{(b,v)}(c : c')$ and $MUL_{(b,v)}(c,c' : c'')$. The prover then opens $(c''b^{-a})^{1/v}$. In general, provided $f(X) = X^e - a \pmod v$, **PROOF**$[c = BC(s) \wedge f(s) = 0]$ is executed by $(\log_2 e)SQR$ and $d_H(\log_2 e)MUL$, where $d_H(\log_2 e)$ is the Hamming distance of e.

4.1 Checking Protocol II

The following protocol is a modification of "Basic Protocol" in [FO97]. It is a statistical witness hiding protocol in which the prover, on input $((N,b,g_1,g_2),c,m)$, convinces the verifier that he knows (x,r) such that $c = \overline{BC}_{(b,g_1)}(x,r)$ and $x \in (a-2^mv, a+2^mv)$, $r \in (-2^{2m}v,\ 2^{2m}v)$. We denotes that by $CHCK2_{(b,g_1,(a,v,m))}(c)$.

Protocol: $CHCK2_{(b,g_1,(a,v,m))}(c)$
Common Input: $((N,b,g_1,g_2),(a,v,m),c)$ where $m = O(|N|)$ and $1/4N < v$.
Prover's Input: (x,r) such that $c = \overline{BC}_{(b,g_1)}(x,r)$ where $x \in (a-v, a+v)$ and $r \in (-2^mv,\ 2^mv)$.
Prover's Claim: (x,r) such that $c = \overline{BC}_{(b,g_1)}(x,r)$ where $x \in (\ -2^mv + a,\ 2^mv + a\ )$ and $r \in (\ -2^{2m}v,\ 2^{2m}v\ )$.

1. P sets (w_1^0, w_0^1) such that
   - $w_1^0 \in_R [\,0,\ 2^mv)$ if $x \in [\,a,\ a+v\ )$, otherwise $w_1^0 \in_R [\,2^mv,\ 2^{m+1}v\ )$, and
   - $w_1^1 \in_R [\,0,\ 2^{2m}v\ )$ if $r \in [\,0,\ 2^mv\ )$, otherwise $w_1^1 \in_R [\,2^{2m}v,\ 2^{2m+1}v\ )$.

   P sets w_2^0, w_2^1 by $w_2^0 = w_1^0 - 2^mv$ and $w_2^1 = w_1^1 - 2^{2m}v$. P picks four elements, $w_{i,j}^2$'s $\in_R [\,0,\ 2^mv\ )$, then computes $t_{i,j} = \overline{BC}_{(b,g_1,g_2)}(w_i^0, w_j^1, w_{ij}^2)$, where $1 \le i,j \le 2$.
2. P sends to V, four unordered commitments, $t_{i,j}$'s.
3. V picks a challenge $e \in_R [\,0, 2^m)$ and sends it to P.

4. P sets $X := e(x-a) + w_i^0$ and $R := er + w_j^1$ such that $X \in [\,0,\ 2^m v\,)$ and $R \in [\,0,\ 2^{2m} v\,)$, and he sends to V, the pair, $(\ X,\ R,\ w_{i,j}^2\ )$.
5. V checks $X \in [\,0, 2^m v\,)$ and $R \in [\,0, 2^{2m} v\,)$, and there exists a $t_{i,j}$ such that $\overline{BC}_{(b,g_1,g_2)}(X, R, w_{ij}^2) \equiv t_{i,j}(c \cdot b_0^{-a})^e \pmod N$.

Here so as to prove soundness we state the modified RSA assumption [FO97].

Assumption 4. (Modified RSA Assumption) There exists a probabilistic polynomial-time generator Δ which on input $1^{|N|}$ outputs (N, Y) such that for any probabilistic polynomial-size circuit family $\mathcal{C}$, the probability that $\mathcal{C}$ can on input $(N, Y) \in \mathbb{Z} \times \mathbb{Z}_N^*$ outputs $(e, Y^{1/e} \bmod N)$ where $e > 1$ is negligible in $|N|$. The probability is taken over the random choices of Δ and $\mathcal{C}$.

Lemma 5. *Under Assumption 4, there exists a probabilistic poly-time knowledge extractor M such that for any probabilistic poly-time algorithm P^* if probabilistic interactive algorithm (P^*, V) accepts with non-negligible probability in $|N|$, then M with P^* as an oracle can extract (x, r) with non-negligible probability in $|N|$ such that $c = BC(x, r)$ where $x \in (\,-2^m v + a, 2^m v + a\,)$ and $r \in (\,-2^{2m} v, 2^{2m} v\,)$.*

Sketch of Proof:
M can extract the pairs, $(t_{i,j}, e, X, R, w_{i,j}^2)$ and $(t_{i,j}, e', X', R', w_{i,j}^2)$ where $e, e' \in [\,0,\ 2^m\,)$ $(e \neq e')$, $X, X' \in [\,0,\ 2^m v\,)$, and $R, R' \in [\,0,\ 2^{2m} v\,)$. Then, under the modified RSA assumption, we can prove that $c \equiv b^{\Delta x} g_1^{\Delta r} \pmod N$ where $\Delta x - a = \frac{X - X'}{e - e'}$ and $\Delta r = \frac{R - R'}{e - e'}$ (See [FO97]). Clearly $\Delta x \in (\,-2^m v + a, 2^m v + a\,)$, $\Delta r \in (\,-2^{2m} v, 2^{2m} v\,)$. □

Lemma 6. *$CHCK2_{(b,g_1,(a,v,m))}(c)$ is statistical witness indistinguishable if $x \in (\,-v + a,\ v + a\,)$, and $r \in (\,-2^m v, 2^m v\,)$.*

4.2 Transit Protocol

The transit protocol is a statistical witness hiding protocol in which the prover, on input $((N, b, g_1, g_2, v_1, v_2) c_1, c_2)$, convinces the verifier that he knows (x_1, x_2, r_1, r_2) such that $x_1 \bmod v_1 = x_2 \bmod v_2$, $c_1 = BC_{(b,v_1)}(x_1, r_1)$, and $c_2 = BC_{(b,v_2)}(x_2, r_2)$. We denotes the transit protocol by $TRAN_{(b,v_1,v_2)}(c_1, c_2)$. The complexity of $TRAN_{(b,v_1,v_2)}(c_1, c_2)$ is that of $4 * CHCK + CHCK2$.

Protocol: $TRAN_{(b,v_1,v_2)}(c_1, c_2)$
Common Input: $((N, b, v_1, v_2, g_1, g_2), m, (c_1, c_2))$ where $m = O(|N|)$ and $0 < 2^{m+1} v_1 < v_2$.
Prover's Input: (x_1, x_2, r_1, r_2) such that $c_1 = BC_{(b,v_1)}(x_1, r_1)$ and $c_2 = BC_{(b,v_2)}(x_2, r_2)$ where $x_1 \in [\,0, v_1\,)$, $x_2 \in [\,0, v_2\,)$, $x_2 = x_1 + (2^m - \delta) v_1$, and $\delta \in \{0, 1\}$.
Prover's Claim: (x_1, x_2, r_1, r_2) such that $c_1 = BC_{(b,v_1)}(x_1, r_1)$ and $c_2 = BC_{(b,v_2)}(x_2, r_2)$ where $x_1 \equiv x_2 \pmod{v_1}$.

1. P sets $\bar{c} = \overline{BC}_{(b,g_1)}(x_2, \alpha)$ and sends it to V.
2. P executes with V,

$$CHCK_{(b,v_1)}(c_1), CHCK_{(b,v_2)}(c_2), CHCK_{(g_1,v_1)}(\frac{\bar{c}}{c_1}), CHCK_{(g_1,v_2)}(\frac{\bar{c}}{c_2}),$$

and $CHCK2_{(b,g_1,(2^m v_1, v_1, m))}(\bar{c})$.

Lemma 7. *Under Assumption 4, there exists a probabilistic poly-time knowledge extractor M such that for any probabilistic poly-time algorithm P^* if probabilistic interactive algorithm (P^*, V) accepts with non-negligible probability in $|N|$, then M with P^* as an oracle can extract (x_1, x_2, r_1, r_2) with non-negligible probability in $|N|$ such that $c_1 = BC_{(b,v_1)}(x_1, r_1)$, $c_2 = BC_{(b,v_2)}(x_2, r_2)$, and $x_1 \equiv x_2 \pmod{v_1}$.*

Sketch of Proof:

Assume that $\bar{c} = \overline{BC}_{((b,g_1)}(x, \alpha)$. By $CHCK_{(g_1,v_1)}(\frac{\bar{c}}{c_1})$ and $CHCK_{(g_1,v_2)}(\frac{\bar{c}}{c_2})$, we can show that $x = x_1 + k_1 v_1 = x_2 + k_2 v_2$. As $CHCK2_{(b,g_1,(2^m v_1, v_1, m))}(\bar{c})$ gives that $0 < x < 2^{m+1} v_1$ $(< v_2)$, we can show $x = x_2$ and $x_1 = x_2 \bmod v_1$. □

5 Publicly Verifiable Secret Sharing Scheme

Hereafter, $(\mathcal{E}, \mathcal{D})$ denotes the RSA cryptosystem. $pk := (n, e)$, $sk := d$, $ssp := (v, m)$, $MES := \mathbb{Z}_v$, $SHA := (\ (2^m - 1)v, (2^m + 1)v\ )$, $E := RSA_{n,e}$, where $RSA_{n,e}(x) := x^e \bmod n$, and $CYPH_E := \{y | x \in SHA \wedge y := x^e \bmod n\}$, where $\frac{1}{2}|n| < |v| < |n| - (m+1)$, and $m = O(|v|)$.

Here we give an assumption. If the modified RSA assumption and the following assumption hold true, our proposed PVSS scheme can be proved to be secure (k, l)-publicly verifiable secret sharing schemes (Our proposed PVKE and PVPKE systems based on the factoring can also be proved to be secure under these assumptions while those based on the discrete logarithm are not).

Assumption 8. Let $\mathbf{E} := (E_1, \ldots, E_l)$ where $E_i \in \mathcal{E}$ and

$$\mathcal{S} := \{\mathbf{C} \in CYPH^l | \exists s \in \mathbb{Z}_v, \forall A \in \mathcal{A}_k : Recover(\{D_i(C_i) | P_i \in A\}) = s\}.$$

There exist two probabilistic polynomial-time l-tuple instance generators, $G_{\mathbf{E}}^{(1)}$ and $G_{\mathbf{E}}^{(2)}$, having the following properties:

- $G_{\mathbf{E}}^{(1)}$ and $G_{\mathbf{E}}^{(2)}$ are l-tuple *hard* instance generators (*hard instance generator assumption* (see Def. 1)),
- $\mathbf{C_1} \leftarrow G_{\mathbf{E}}^{(1)}$, $\mathbf{C_2} \leftarrow G_{\mathbf{E}}^{(2)}$: $\mathbf{C_1} \in \mathcal{S}$ and $\mathbf{C_2} \notin \mathcal{S}$, and
- there exists no probabilistic polynomial-time circuit family A, which on input $(\mathbf{C_1}, \mathbf{C_2})$, such that $\exists poly(\cdot)$, $n \to \infty$: A guesses correctly with the probability greater than $\frac{1}{2} + \frac{1}{poly(n)}$ (*computationally indistinguishable assumption*).

Here we present a secure (k, l)-threshold publicly verifiable secret sharing scheme (PVSS scheme). Let D be a dealer, $P_1, \ldots, P_l$ be participants (or shareholders), and V be a verifier. (N, b, g_1, g_2, v, m) denotes a set of the public parameters and (n_i, e_i) denotes an RSA public key of P_i, where v is a prime of $1/2|n_i| < |v| < |n_i| - (m+1)$, and $m = O(|v|) = O(|N|)$. D has secret $s \in \mathbb{Z}_v$.

The dealer D chooses a polynomial

$$f(X) := s + \sum_{j=1}^{k-1} a_j X^j \pmod{v},$$

where s is D's secret, each $a_j \in_R \mathbb{Z}_v$, and $a_{k-1} \neq 0$. Then D sets, for $i = 1, \ldots, l$, $s_i := (f(i) \bmod v) + (2^m - \delta_i)v$ where $\delta_i \in \{0, 1\}$. Here note that $(2^m - 1)v < s_i < (2^m + 1)v$. D sets and broadcasts $C_i := s_i^{e_i} \bmod n_i$ for $i = 1, \ldots, l$.

The protocol is as follows:

[Common Input] (N, b, g_1, g_2, v, m), $\{(n_i, e_i)\}$, $(C_1, \ldots, C_l)$.

[Dealer's Claim] he knows $(C_1^{1/e_1} \bmod n_1, \ldots, C_l^{1/e_l} \bmod n_l)$ such that

$$(C_i^{1/e_i} \bmod n_i) \equiv s + \sum_{j=1}^{k-1} a_j i^j \pmod{v}.$$

[Protocol]

1. D sets and sends to V $c_0 = BC_{(b,v)}(s, r_0)$ where $r_0 \in_R Z_N^*$.
2. D sets, for $j = 1, \ldots, k-1$, $c_j = BC_{(b,v)}(a_j, r_j)$, where $r_1, \ldots, r_{k-1} \in_R Z_N^*$. Then, for $i = 1, \ldots, l$, D sets $A_i = BC_{(b,v)}(s_i, t_i)$ and $B_i = BC_{(b,n_i)}(s_i, r_i')$ where $t_i := \prod_{j=0}^{k-1} r_j^{i^j} \bmod N$ and $r_i' \in_R Z_N^*$. D broadcasts $(c_1, \ldots, c_{k-1})$, $(A_1, \ldots, A_l)$, and $(B_1, \ldots, B_l)$.
3. V checks
$$A_i \equiv \prod_{j=0}^{k-1} c_j^{i^j} \pmod{N}.$$
4. For $i = 1, \ldots, l$, D executes with V $TRAN_{(b,v,n_i)}(A_i, B_i)$ and PROOF$[B_i = BC_{(b,n_i)}(s_i) \wedge D_i(s_i) = 0 \bmod n_i]$ where $D_i(X) := X^{e_i} - C_i$.

Lemma 9. *Under Assumption 4,*

$$f(i) \equiv (C_i^{1/e_i} \bmod n_i) \pmod{v}.$$

From Lemma 7, the proof is obvious.

Theorem 10. *Under Assumptions, 4 and 8, the proposed protocol is secure (k, l)-publicly verifiable secret sharing.*

Sketch of Proof:

From Lemma 9 and analogies of the traditional VSS schemes, the *correctness* and *public verifiability* are obvious, because any k shareholders, $P_{j_1}, \ldots, P_{j_k}$, can recover a unique and valid secret s by

$$s = \sum_{i=1}^{k} \left(\prod_{i' \neq i} \frac{j_i}{j_i - j_{i'}}\right)(C_{j_i}^{d_{j_i}} \bmod n_{j_i}) \bmod v. \tag{1}$$

We consider the *secrecy* condition. Our protocol is statistical zero-knowledge if the set-up protocol is executed. Therefore, without loss of generality, we can consider $ADV^{U,\mathcal{O}}$ as $ADV^{\mathcal{O}}$ that works as follows:

1. Input, to ADV, $\mathbf{C} := (C_1, \ldots, C_l) \in CYPH^l$ generated by $G_{\mathbf{E}}$.
2. ADV sends $(C_{j_1}, \ldots, C_{j_{k-1}})$ to oracle $\mathcal{O}$.

3. If oracle $\mathcal{O}$ returns $(D_{j_1}(C_{j_1}), \dots, D_{j_{k-1}}(C_{j_{k-1}}))$, ADV outputs $s \in R(\overline{s})$ with non-negligible probability where
$$R(\overline{s}) := \{s | \exists A \in \mathcal{A}_k : Recover(\{D_i(\overline{s}_i) | P_i \in A\}) = s\},$$
otherwise ADV quits.

Let $G_{\mathbf{E}}^{(1)}$ and $G_{\mathbf{E}}^{(2)}$ be the same defined in Assumption 8. We have $G_{\mathbf{E}}^{(2)}$ generate $\mathbf{C} := (C_1, \dots, C_l)$. We randomly choose $(j_1, \dots, j_{l-1}) \in_R (1, \dots, l)$ and then choose $s'_{j_1}, \dots, s'_{j_{l-1}} \in ((2^m - 1)v, (2^m + 1)v)$ such that $E_{j_1}(s'_{j_1}), \dots, E_{j_{l-1}}(s'_{j_{l-1}})$ are taken from the distribution of $G_{\mathbf{E}}^{(2)}$. Let $C'_{j_t} := E_{j_t}(s'_{j_t})$. We input $(C'_{j_1}, \dots, C'_{j_{l-1}}, C_{j_l})$ into ADV. If C_{j_l} is not included in $(k-1)$ queries of ADV, we answer their corresponding RSA roots to ADV, otherwise quit. If ADV outputs $s \in R(\overline{s}) = \{s | \exists A \in \mathcal{A}_k : Recover(\{D_i(\overline{s}_i) | P_i \in A\}) = s\}$, and $P_{j_l} \in A$, from Lemma 9 and the property of $G_{\mathbf{E}}^{(2)}$, we can compute $C_{j_l}^{d_{j_l}} \bmod n_{j_l}$ by
$$C_{j_l}^{d_{j_l}} \bmod n_{j_l} = (s + \sum_{t=1}^{k-1} a_t {j_l}^t \bmod v) + (2^m - \delta)v$$
where $\delta \in \{0, 1\}$ and $a_t = \sum_{i=1}^{k} (\prod_{i' \neq i} \frac{j_i^t}{j_i - j_{i'}}) s'_{j_i} \bmod v$. This strategy succeeds with non-negligible probability in $|v|$. □

6 Publicly Verifiable Key Escrow (PVKE) Cryptosystems

6.1 PVKE cryptosystem based on the factoring

In this section, we describe a (k, l)-threshold publicly verifiable key escrow (PVKE) scheme based on the factoring.

Let U be a user, $P_1, \dots, P_l$ be trustees and V be a verifier. Let (N, b, g_1, g_2, m) be the public parameters, (n_i, e_i) denotes a public key of P_i and (n, e) denotes a public key of U, where $(\frac{-1}{n}) = 1$, $1/2|n_i| < |n| < |n_i| - (m+1)$, and $m = O(|n_i|) = O(|N|)$.

The user U generates (u, s) such that $u^2 = s^4 \bmod n$ and $(\frac{u}{n}) = -1$. U choose a polynomial
$$f(X) := s + \sum_{j=1}^{k-1} a_j X^j \pmod{n},$$
where $a_j \in_R \mathbb{Z}_n$, and $a_{k-1} \neq 0$. Then U sets, for $i = 1, \dots, l$, $s_i := (f(i) \bmod n) + (2^m - \delta_i)n$ where $\delta_i \in \{0, 1\}$. Here note that $(2^m - 1)n < s_i < (2^m + 1)n$. U broadcasts $(u, C_1, \dots, C_l)$ where $C_i := s_i^{e_i} \bmod n_i$.

U executes with V the following protocol:

[Common Input] (N, b, g_1, g_2, m), $\{(n_i, e_i)\}$, (n, e, u), $(C_1, \dots, C_l)$.
[User's Claim] he knows $(C_1^{1/e_1}, \dots, C_l^{1/e_l})$ such that
$$(C_i^{1/e_i} \bmod n_i) \equiv s + \sum_{j=1}^{k-1} a_j i^j \pmod{n} \text{ and } u^2 \equiv s^4 \pmod{n}.$$

[Protocol]

1. U sets and sends to V $c_0 = BC_{(b,n)}(s, r_0)$ where $r_0 \in_R Z_N^*$.
2. U executes with V PROOF$[c_0 = BC_{(b,n)}(s) \wedge h(s) = 0 \bmod n]$ where $h(X) = X^4 - u^2$.
3. U executes with V the above-mentioned PVSS scheme from Step 2, replacing v with n.

Note: U has to give V the evidence that n consists of two primes before the protocol begins. A non-interactive proof for this purpose is described in [Mic92].

Lemma 11. *Suppose that* $(\frac{u}{n}) = -1$, $(\frac{-1}{n}) = 1$ *and* $u^2 \equiv s^4 \pmod{n}$. $(u \pm s^2)$ *are non-trivial factors of* n.

Theorem 12. *Under Assumptions, 4 and 8, the proposed protocol is secure* (k,l)*-publicly verifiable secret sharing.*

6.2 PVKE cryptosystems based on the discrete logarithm

Due to the space limitation, we describe it in Appendix.

7 Publicly Verifiable Partial Key Escrow Cryptosystems

This section presents (publicly) verifiable partial key escrow cryptosystems. Verifiable partial key escrow (VPKE) is well described in [BG97]. In VPKE cryptosystems, the escrow agencies (shareholders) are only allowed to share *partial information* of users' private keys, so as not to recover a large number of users' private keys at the same time by malicious authorities. On the key recovery phase, the escrow agencies require a non-trivial amount of work to find private keys from the corresponding partial informations. In the VPKE cryptosystems, the most serious problem is the *early recovery* attack, pointed out in [BG97]. The early recovery attack is that the escrow agencies compute the unescrowed information of the private key before the collaboration and recover the whole private key quickly after the collaboration. To overcome this attack, [BG97] presented a VPKE cryptosystem with the *delayed recovery* property. The delayed recovery is opposite to the early recovery: In VPKE systems with the delayed recovery property, the agencies can not compute unescrowed informations before the collaboration and require non-trivial amount to recover the private keys after the collaboration (See [BG97] for details). Their scheme was based on the discrete logarithm and, to the best of our knowledge, no one seems to have presented any VPKE cryptosystem with the delayed recovery based on the factoring after their work [Mao97]. In the following, we describe (Publicly) VPKE cryptosystems, one based on the factoring and the other based on the discrete logarithm. As mentioned above, the factoring-based one seems to be the first VPKE cryptosystem based on the factoring with the delayed recovery.

Let U, $P_1, \ldots, P_l$, and V be the same as described above. Let $(N, b, g_1, g_2, m, \{(n_i, e_i)\})$ be the same as above, too. $MES := \mathbb{Z}_v$. However, SHA and $CYPH_E$ are little changed as $SHA := (\ -2^m v, 2^m v\ )$ and $CYPH_E := \{y | x \in SHA \wedge y := x^e \bmod n\}$.

$s \in \mathbb{Z}_v$ denotes the secret of user U, $f(X)$ is a random polynomial, and $(s_1, \ldots, s_l)$ denote its shares. In stead of setting $s_i := (f(i) \bmod n) + (2^m - \delta_i)n$ as in the prior PVKE schemes, U sets $s_i := (f(i) \bmod n) - \delta_i(n_i - v)$, where $\delta \in \{0,1\}$ and v is n in the factoring systems and q in the discrete logarithm systems. U sends $(C_1, \dots, C_l)$ to verifier V where $C_i = E_i(s_i)$. U executes with V, on input $(C_1, \dots, C_l)$, the mentioned-above PVKE cryptosystems replacing protocol $TRAN$ with $TRAN2$ (See Appendix). Then the following lemma can be proved.

Lemma 13. *Under Assumption 4,*

$$f(i) \equiv ((C_{j_i}^{1/e_i} - \delta_i v) \bmod n_i) \pmod{n}, \quad \textit{where } \delta_i \in \{0,1\}.$$

Theorem 14. *Under Assumptions, 4 and 8, the PVPKE system based on the factoring is secure (k,l)-publicly verifiable secret sharing.*

Note that Assumption 8 in this section is a little different from this assumption in the previous sections because SHA and $CYPH_E$ have been a little changed in this section.

If k is enough large (e.g. $k = 48$), these protocols are partial key escrow cryptosystems with the delayed recovery property. Before the collaboration, it is impossible to determine δ_i, and after the collaboration, trustees require almost 2^k trials to find s such that $s^4 - u^2 \equiv 0 \pmod{v}$ (or $y = g^s \bmod p$) from

$$s := \sum_{i=1}^{k} \left(\prod_{i' \neq i} \frac{j_i}{j_i - j_{i'}}\right)((C_{j_i}^{d_{j_i}} - \delta_{j_i} v) \bmod n_{j_i}) \bmod v, \text{ where } \delta_{j_i} \in \{0,1\}.$$

8 Conclusions

We have presented a provably secure PVSS scheme which is $O(|v|)$ times more efficient than Stadler's PVSS schemes, where $|v|$ denotes the size of the secret. It can be incorporated into key escrow cryptosystems based on the factoring and the discrete logarithm, and can transform them to publicly verifiable key escrow (PVKE) ones or publicly verifiable *partial* key escrow ones. The PVKE and PVPKE systems based on the factoring can also be proved to be secure while those based on the discrete logarithm were not.

References

[BCDG87] Brickell, E. F., Chaum, D., Damgård, I., and Gråf, J. van de, "Gradual and verifiable release of a secret", Proceedings of CRYPTO'87, pp.156–166. (1988).

[BG97] Bellare, M., and Goldwasser, S., "Verifiable Partial Key Escrow", To appear in Proceedings of the Fourth Annual Conference and Communications Security, ACM, 1997.

[Bla79] Blakley, G., "Safeguarding cryptographic keys", AFIPS Conference Proceeding, June 1979.

[BGW88] Ben-Or, M., Goldwasser, S., and Wigderson, A., "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation", Proceeding of STOC88, pp 11–17.

[CCD88] Chaum, D., Crepeau, C., and Damgård, I., "Multiparty Unconditionally Secure Protocol", Proceeding of STOC88, pp 1–10.

[CGMA85] Chor, B., Goldwasser, S., Micali, S. and Awerbuch, B.: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults, Proc. of FOCS, pp.383-395 (1985).

[Fel87] Feldman, P., "A Practical Scheme for Non-interactive Verifiable Secret Sharing," In Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science, pp.427–437 (1987).

[Fuj98] Fujisaki, E., "Efficient PZK Proofs for Boolean Formulae whose atoms are Polynomials", Manuscript.

[FO97] Fujisaki, E., and Okamoto, T., "Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations," Proceedings of Crypto'97, LNCS 1294, Springer, pp.16–30 (1997).

[FFS88] U.Feige, A.Fiat and A.Shamir, "Zero Knowledge Proofs of Identity," Journal of Cryptology, Vol. 1, pp.77-94 (1988).

[FS90] U.Feige, and A.Shamir, "Witness Indistinguishable and Witness Hiding Protocols," Proc. of STOC90.

[GMW87] Goldreich, O., Micali, S. and Wiederson, A., "How to play any mental game", Proceedings of STOC87, pp. 218–229.

[Mao97] Mao, W., "Publicly Verifiable Partial Key Escrow", Proceedings of ICICS'97, Beijing, pp.409-413 (1997).

[Mic92] Micali, S., "Fair Public-Key Cryptosystems", Proceedings of CRYPTO'92, pp.113-138 (1993).

[Ped91] Pedersen, T. P., "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing", Proceedings of Crypto 91, pp. 129–140 (1992).

[Sha79] Shamir, A., "How to share a secret", CACM, Vol 22, No.11, pp.612–613 (1979).

[Sta96] Stadler, M. : Publicly Verifiable Secret Sharing, Proc. of Eurocrypt'96, LNCS 1070, Springer, pp.190–199 (1996)

A Building Blocks

A.1 Building Blocks for PROOF[$c = BC(s) \wedge f(s) = 0$]

[Checking Protocol]: $CHCK_{(b,v)}(c)$

1. P computes and sends to verifier $t := BC_{(b,v)}(w, \eta)$ where $w \in_R \mathbb{Z}_v$, and $\eta \in_R \mathbb{Z}_N^*$.
2. V chooses $e \in_R \mathbb{Z}_v$ and sends it to the prover.
3. P sends (X, R) to the verifier such that $X := es + w \bmod v$, and $R := r^e \eta b^k \bmod N$ where $k = \lfloor \frac{es+w}{v} \rfloor$.
4. V checks that $BC_{(b,v)}(X, R) \equiv t \cdot c^e \pmod{N}$.

[Comparing Protocol]: $COM_{(b_1,b_2,v)}(c_1, c_2)$
A perfect witness hiding protocol between a prover and a verifier in which the prover, on input $((N, b_1, b_2, v), c_1, c_2)$, convinces the verifier that he knows (x, r_1, r_2) such that $c_1 = BC_{(b_1,v)}(x, r_1)$ and $c_2 = BC_{(b_2,v)}(x, r_2)$.

1. P computes and sends to verifier, $t_1 := BC_{(b_1,v)}(w, \eta_1)$ and $t_2 := BC_{(b_2,v)}(w, \eta_2)$ where $w \in_R \mathbb{Z}_v$, and $\eta_1, \eta_2 \in_R \mathbb{Z}_N^*$.
2. V chooses $e \in_R \mathbb{Z}_v$ and sends it to the prover.
3. P sends (X, R_1, R_2) to the verifier such that $X := ex + w \bmod v$, $R_1 := r_1^e \eta_1 b_1^k \bmod N$, and $R_2 := r_2^e \eta_2 b_2^k \bmod N$, where $k = \lfloor \frac{ex+w}{v} \rfloor$.
4. V checks that $BC_{(b_1,v)}(X, R_1) \equiv t_1 \cdot c_1^e \pmod{N}$ and $BC_{(b_2,v)}(X, R_2) \equiv t_2 \cdot c_2^e \pmod{N}$.

[Adding Protocol]: $ADD_{(b,v)}(c_1, c_2 : c_3)$
A perfect witness hiding protocol between a prover and a verifier in which the prover, on input $((N, b, v), c_1, c_2, c_3)$, convinces the verifier that he knows $(x_1, x_2, x_3, r_1, r_2, r_3)$ such that $x_3 \equiv x_1 + x_2 \pmod{v}$, $c_1 = BC_{(b,v)}(x_1, r_1)$, $c_2 = BC_{(b,v)}(x_2, r_2)$ and $c_3 = BC_{(b,v)}(x_3, r_3)$. The protocol is executed as follows: P executes with V $CHCK_{(b,v)}(c_1)$, $CHCK_{(b,v)}(c_2)$, and $CHCK_{(b,v)}(c_3)$. P then reveals $(c_1 c_2 c_3^{-1})^{1/v}$ and V checks it.

[Mod-Multi Protocol]: $MUL_{(b,v)}(c_1, c_2 : c_3)$
P executes with V $CHCK_{(b,v)}(c_1)$. P then executes with V $COM_{(b,c_1,v)}(c_2, c_3)$.

Note: As $c_3 = b^{x_3} r_3{}^v = c_1^{x_2}(b^k r_1^{-x_2} r_3)^v$ where $x_3 = x_1 \cdot x_2 + kv$, P can execute $COM_{(b,c_1,v)}(c_2, c_3)$.

[Squaring Protocol]: $SQR_{(b,v)}(c_1 : c_2)$
The prover executes with the verifier $COM_{(b,c_1,v)}(c_1, c_2)$.

Lemma 15. *Under the RSA assumption, the protocols mentioned above are perfect witness hiding.*

Details and some security remarks are described in [Fuj98].

A.2 Building Blocks for the PVPKE cryptosystems

Protocol: $TRAN2_{(b,v_1,v_2)}(c_1, c_2)$
Common Input: $((N, b, v_1, v_2, g_1, g_2), m, (c_1, c_2))$ where $m = O(|N|)$ and $0 < 2^m v_1 < v_2$.
Prover's Input: (x_1, x_2, r_1, r_2) such that $c_1 = BC_{(b,v_1)}(x_1, r_1)$ and $c_2 = BC_{(b,v_2)}(x_2, r_2)$ where $x_1 \in [\,0, v_1\,)$, $x_2 \in [\,0, v_2\,)$ and $x_2 = x_1$ or $x_1 - v_1 + v_2$.
Prover's Claim: (x_1, r_1, x_2, r_2) such that $c_1 = BC_{(b,v_1)}(x_1, r_1)$ and $c_2 = BC_{(b,v_2)}(x_2, r_2)$ where $x_1 \equiv (x_2 - \delta v_2) \pmod{v_1}$ where $\delta \in \{0, 1\}$.

1. P sets $\bar{c} = \overline{BC}_{(b,g_1)}(x, \alpha)$ and sends it to V where $x := x_1$ or $x_1 - v_1$.
2. P executes with V,
$$CHCK_{(b,v_1)}(c_1), CHCK_{(b,v_2)}(c_2), CHCK_{(g_1,v_1)}(\frac{\bar{c}}{c_1}), CHCK_{(g_1,v_2)}(\frac{\bar{c}}{c_2}),$$
and $CHCK2_{(b,g_1,(0,v_1,m))}(\bar{c})$.

Lemma 16. *Under Assumption 4, there exists a probabilistic poly-time knowledge extractor M such that for any probabilistic poly-time algorithm P^* if probabilistic interactive algorithm (P^*, V) accepts with non-negligible probability in $|N|$, then M with P^* as an oracle can extract (x_1, x_2, r_1, r_2) with non-negligible probability in $|N|$ such that $c_1 = BC_{(b,v_1)}(x_1, r_1)$, $c_2 = BC_{(b,v_2)}(x_2, r_2)$, and $x_1 \equiv (x_2 - \delta v_2) \pmod{v_1}$ where $\delta \in \{0, 1\}$.*

Sketch of Proof:
Assume that $\bar{c} = \overline{BC}_{(b,g_1)}(x, \alpha)$. By $CHCK_{(g_1,v_1)}(\frac{\bar{c}}{c_1})$ and $CHCK_{(g_1,v_2)}(\frac{\bar{c}}{c_2})$, $x = x_1 + k_1 v_1 = x_2 + k_2 v_2$. As $CHCK2_{(b,g_1,(0,v_1,m))}(\bar{c})$ gives that $(-v_2 <)$ $-2^m v_1 < x < 2^m v_1$ $(< v_2)$, we can show that $x = x_2$ or $x_2 - v_2$. Therefore, $x_1 \equiv (x_2 - \delta v_2) \pmod{v_1}$ where $\delta \in \{0, 1\}$. □

B PVKE cryptosystem based on the discrete logarithm

[Comparing-with-DL Protocol] $COMDL_{(b,g,q)}(c,y)$
Common Input: (N,b,g,p,q) and (c,y) where $q|p-1$ and $<g>=G_q \subset \mathbb{Z}_p^*$.
Knowledge of Prover: $(x,r) \in \mathbb{Z}_q \times \mathbb{Z}_N^*$ such that $y = g^x \bmod p$, and $c = BC_{(b,q)}(x,r)$.

1. P chooses $(w,\eta) \in_R \mathbb{Z}_q \times \mathbb{Z}_N^*$ and computes $t = BC_{(b,q)}(w,\eta)$ and $u = g^w \bmod p$.
2. P sends to V, (t,u).
3. V picks $e \in_R \mathbb{Z}_q$ and sends it to P.
4. P sets $X := ex + w \bmod q$ and $R := r^e \eta b^k \bmod N$ where $k = \lfloor \frac{ex+w}{q} \rfloor$. P then sends to V, (X,R).
5. V checks $BC_{(b,q)}(X,R) \equiv tc^e \pmod N$ and $g^X \equiv uy^e \pmod p$.

Lemma 17. *Under the discrete logarithm assumption and Assumption 4, the protocol above is perfect witness hiding.*

Let U, $P_1,\ldots,P_l$, V be the same as described above. Let $(N,b,g_1,g_2,m,\{(n_i,e_i)\})$ be the same as above, too. (p,q,g,y) denote public parameters of U for the discrete logarithm cryptosystems where $ord(g)=q$, $1/2|n_i| < |q| < |n_i| - (m+1)$, and $y \in G_q$. $s \in \mathbb{Z}_q$ denotes the secret key of U where $y = g^s \bmod p$.

U chooses a polynomial $f(X) := s + \sum_{j=1}^{k-1} a_j X^j \pmod n$, where $a_j \in_R \mathbb{Z}_n$, and $a_{k-1} \neq 0$. Then U sets, for $i = 1,\ldots,l$, $s_i := (f(i) \bmod q) + (2^m - \delta_i)q$ where $\delta_i \in \{0,1\}$. Here note that $(2^m-1)q < s_i < (2^m+1)q$. U broadcasts $(C_1,\ldots,C_l)$ where $C_i := s_i^{e_i} \bmod n_i$.

U executes with V the following protocol:

[Common Input] (N,b,g_1,g_2,m), $\{(n_i,e_i)\}$, (g,p,q,y), $(C_1,\ldots,C_l)$.
[User's Claim] he knows $(C_1^{1/e_1},\ldots,C_l^{1/e_l})$ such that

$$(C_i^{1/e_i} \bmod n_i) \equiv s + \sum_{j=1}^{k-1} a_j i^j \pmod q, \text{ and } y \equiv g^s \pmod p.$$

[Protocol]

1. User U sets and sends to V $c_0 = BC(s,r_0)$ where $r_0 \in_R \mathbb{Z}_N^*$.
2. U executes with V $COMDL_{(b,g,q)}(c_0,y)$ (See Appendix).
3. U executes with V the PVSS scheme in Sec.5 from Step 2, replacing v with q.