**Article ID**: 1002—1175(2004)02-0210-08

# Publicly Verifiable Secret Redistribution for Threshold Secret Sharing Scheme*

TAN Zuo-Wen LIU Zhuo-Jun

(*Key Lab. of Mathematics Mechanization, Institute of Systems Science, AMSS, Chinese Academy of Sciences, Beijing* 100080, *China*)

(Received 7 April 2003; Revised 28 June 2003)

**Abstract** This paper presents a publicly verifiable secret redistribution model, and proposes a PVSR scheme. The scheme can redistribute secrets from a $(t_1, n_1)$ to $(t_2, n_2)$ access structure. New shareholders can generate valid new shares if they can verify both the validity of the old share and that of the sub-shares which the shareholders give the new shareholders. In the scheme, other new shareholders can determine the validity of the sub-shares of a new shareholder.

**Key words** VSS, PVSR, threshold, access structure

**CLC** TP309

## 1 Introduction

A secret sharing scheme[1,2] first divides a secret into pieces called shares, which are distributed amongst a group of participants so that shares of specific subsets of participants are pooled and then the shared secret can be reconstructed. Since its invention, many schemes have been proposed. Though the threshold secret sharing schemes provide fundamental building blocks, the general threshold secret sharing schemes can't prevent the dealer from distributing the participants false shares, with the result that any subset of participants can't recover the genuine secret, and can't also detect a false shareholder's cheating, so the cheater can obtain the true secret while all the other honest shareholders can't see the shared secret. Chor, Goldwasser and Micali[3] proposed a verifiable secret sharing scheme (VSS) to achieve security against cheating participants. Several secure and efficient VSS schemes were proposed[4,5]. VSS schemes allow each participant to verify that its share is consistent with the other shares, but the schemes can't ensure that everybody is able to verify that the shares have been correctly distributed. Ref. [3] has the very special property. Markus Stadler[6] proposed a new publicly verifiable secret sharing scheme (PVSS) which can be used with general (monotone) access structures, in addition to the threshold secret sharing. PVSS have wider uses than VSS. They can be applied in software key escrow cryptosystems and the design of electronic cash systems providing revocable anonymity[6,7].

It allows for security and management. A distributed storage system, which stores shares of files on a distributed set of servers, needs to produce new shares and invalidate old shares. The technique of redistributing shares of secret between different sets of shareholders is useful for a wider range of applications. Ref. [8] proposed verifiable secret redistribution (VSR) for threshold sharing schemes based on Shamir's threshold sharing scheme, Desmedt and Jajodra's secret redistribution scheme[10] and Feldman's VSS scheme[5]. It is more flexible than proactive secret shar-

* supported by 973 Project (1998030600)

ing (PSS) schemes[11]. Ref.[8] found VSR's applications in multiparty signature schemes and distributed key servers.

We present a publicly verifiable secret redistribution protocol based on Ref.[6] and Ref.[8]. Our scheme can redistribute secrets from a $(t_1, n_1)$ to $(t_2, n_2)$ access structure. New shareholders can generate valid new shares if they can verify both the validity of the old shares and that of the sub-shares that the shareholders give the new shareholders. The scheme overcomes the deficiency of Ref.[8] that other new shareholders cannot determine the validity of the sub-shares of one new shareholder. An adversary, who obtain less than $t_1$ old shares and less than $t_2$ new share, cannot reconstruct the secret. In our scheme, we needn't build secret communication channels between the dealer and every participant, since the dealer doesn't transfer straightly the secret share to any participant.

## 2 Model of PVSR scheme

System parameters are as follows. By $s$, we denote the shared secret. Let $P$, $P'$ be respectively any authorized subset in the access structures $A^{\{t_1, n_1\}}$ and $A^{\{t_2, n_2\}}$. We denote the key pair and the identification of the participants $P_i$ in $P$ by $\{(pk_i, sk_i), ID_i\}$ and that of the participant $P_j'$ in $P'$ by $\{(pk_j', sk_j'), ID_j'\}$, where $i=1, 2, \cdots, n_1$, $j=1, 2, \cdots, n_2$. Before we go further, we make some conventions. When the following verification algorithm outputs 1, we call the scheme passes a validity test. Our goal is to design a scheme which passes all the validity tests. The participant $i$ in $P$ refers to the participant $P_i$ in $P$.

**The First Distribution of The Secret** consists of the following algorithms:

● *A* General Threshold Secret Sharing Algorithm $S$ that takes as input the secret $s$, the parameters $(t_1, n_1)$ in the access structure $A^{\{t_1, n_1\}}$ and the participant identification $ID_i$, where $i \in \{1, 2, \cdots, n_1\}$ and $t_1$ $(1 \leqslant t_1 \leqslant n_1)$ denotes the threshold value, $n_1$ denotes the number of the participants in this distribution. $S$ outputs

$$(s_1, s_2, \cdots, s_{n_1}; w, w_1, w_2, \cdots, w_{n_1}) = S(s, t_1, n_1, ID_1, ID_2, \cdots, ID_{n_1}),$$

where $s_i$ $(i=1, 2, \cdots, n_1)$ specifies the secret share corresponding to the participant $P_i$, $w$ is the shared secret $s$'s witness and $\{w_1, w_2, \cdots, w_{n_1}\}$ is the list of the witnesses corresponding to the list of the secret shares $\{s_1, s_2, \cdots, s_{n_1}\}$. Those witnesses are generated based on the assumption of the difficulty of some problems. Therefore, it is infeasible to compute $s$ and $s_i$ from them. Then the dealer publishes all the witnesses in the form $(ID_i, w_i)$ and keeps $s$ and $s_i$ secret, where $i=1, 2, \cdots, n_1$.

● *A* Randomized Encryption Algorithm $E$ that takes as input the public key list $\{pk_1, pk_2, \cdots, pk_{n_1}\}$ and the secret share list $\{s_1, s_2, \cdots, s_{n_1}\}$, and outputs ciphertexts $c_i = E(pk_i, s_i)$, $i=1, 2, \cdots, n_1$, then makes them public in the form $(ID_i, c_i)$.

● *A* Deterministic Decryption Algorithm $D$ that takes as input $(ID_i, c_i, sk_i)$ and outputs $s_i$. Obviously, only the participant $P_i$ has its own private key and can obtain $s_i$.

● *A* Share Validity Verification Algorithm $V_1$

Every participant $P_i$ runs the algorithm $V_1$ by taking as input its decryption $s_i$ (maybe it isn't a genuine share, but we still take the notation) and the share's witness, and outputting $V_1(ID_i, s_i, w_i) \in \{0, 1\}$. When the output is 1, the scheme passes the own's secret share validity test. The participants publish the outputs.

● *A* Share Validity Public Verification Algorithm $PV$

The participant $P_i$ runs $PV$ which takes as input $(ID_j, c_j, w_j)$, where $i, j \in \{1, 2, \cdots, t_1\}$, $i \neq j$, and outputs $PV(ID_j, c_j, w_j) \in \{0, 1\}$. When the output about $j$ is 1, the participant $P_i$ knows that $P_j$ has received a valid

share.

● *A* Share Presentation Validity Verification Algorithm $V_2$

After the participant $P_i$ sends its share, other participants can execute the algorithm $V_2$ which takes as input $(ID_i, c_i, w_i)$ and output $V_2(ID_i, c_i, w_i) \in \{0, 1\}$.

● *A* Reconstruction Algorithm $R$ which takes as input the valid and correct secret share and outputs the secret $s = R(s_1, s_2, \cdots, s_{t_1})$.

In a scheme, these algorithms should run in turn. Once a validity test fails, the particpants needn't run the subsequent algorithms. They restart the scheme or abort the scheme. The scheme security in the first distribution fully depends on the assumption and the infeasibility of acquiring the secret share from all the witnesses and the ciphertexts.

**Redistribution of the shared secret from the access structure** $A^{\{t_1, n_1\}}$ to $A^{\{t_2, n_2\}}$ consists of the following algorithms:

● *A* General Threshold Secret Share Algorithm $R'$

Each $i$ in any authorized subset $P$ in $A^{\{t_1, n_1\}}$ acts as the dealer to distribute the secret $s_i$ to an authorized subsets $P'$ in $A^{\{t_2, n_2\}}$. $P_i$ runs the algorithm $R'$ which takes as input $\{s_i, t_2, n_2, ID_1', ID_2', \cdots, ID_{n_1}\}$ and outputs $R'(s_i, t_2, n_2, ID_1', ID_2', \cdots, ID_{n_1}) = \{s_{i1}, s_{i2}, \cdots, s_{in_2}; w_{i1}', w_{i2}', \cdots, w_{in_2}'\}$.

We call outputs $\{s_{i1}, s_{i2}, \cdots, s_{in_2}\}$ the sub-shares.

● *A* Randomized Encryption Algorithm $E$

Each $i$ in $P$ runs $E'$ which is: input the identification, the sub-share and the public key of every $j$ in $P'$ and outputs $E'(ID_j', s_{ij}, pk_j') = c_{ij}$ as the ciphertext of the sub-share $s_{ij}$, then makes them public in the form $(ID_j', c_{ij})$, where $j = 1, 2, \cdots, n_2$.

● *A* Share Validity Verification Algorithm $V_1'$

Each $j$ in $P'$ can run $V_1'$ to check the validity of the share distribution $\{s_1, s_2, \cdots, s_{n_1}\}$. $V_1'$ takes as input the witnesses in the first distribution and outputs $V_1'(w_1, w_2, \cdots, w_{n_1}) \in \{0, 1\}$.

● *A* Deterministic Decryption Algorithm $D'$

Each $j$ in $P'$ can decrypt $\{c_{1j}, c_{2j}, \cdots, c_{n_1 j}\}$ and obtain his own sub-shares $\{s_{1j}, s_{2j}, \cdots, s_{n_1 j}\}$ by running $V_1'$:

$$V_1'\{c_{1j}, c_{2j}, \cdots, c_{n_1 j}; sk_j'\} = \{s_{1j}, s_{2j}, \cdots, s_{n_1 j}\}.$$

● *A* Sub-share Validity Verification Algorithm $V_2'$

Every participant $P_j$ in $P'$ can run the algorithm $V_2'$ for testing his own sub-shares' validity by taking as input its decryption $\{s_{1j}, s_{2j}, \cdots, s_{n_1 j}\}$ (maybe they aren't genuine sub-shares, we still take the notation) and the sub-shares' witnesses, and output

$$V_2'(ID_j', s_{1j}, s_{2j}, \cdots, s_{n_1 j}; w_{1j}', w_{2j}' \cdots, w'_{n_1 j}) \in \{0, 1\}.$$

● *A* Sub-share Validity Public Verification Algorithm $PV_1'$

By running $PV_1'$, each $j$ in $P'$ can test the validity of the sub-shares of any other participants in $P'$. The algorithm outputs $PV_1'(ID_i', c_{li}, w_{li}) \in \{0, 1\}$, where $l, i \in \{1, 2, \cdots, n_1\}$, $i \neq j$.

● *A* New-share Generation Algorithm $G$

For each $j$ in $P'$, the algorithm $G$ takes as input all the $s_{lj}$, $l \in \{1, 2, \cdots, n_1\}$, and outputs a secret share $s_j$ which is called $P_j'$'s new-share, and a witness $w_j'$. Then $P_j'$ publishes the witness $w_j'$.

● *A* New-share Public Validity Verification Algorithm $PV_2'$

Through running $PV_2'$, each $i$ in $P'$ can test the new-share validity of any other participant, say $j$, in $P'$, for $PV_2'(ID_j', w_{1j}', w_{2j}', \cdots, w_{n_1 j}') \in \{0, 1\}$.

● *A* New-share Consistence Verification Algorithm $V_3'$ which takes as input the witness $w, w_1', w_2', \cdots, w_{n_2}'$ and outputs $V_3'(w, w_1', w_2', \cdots, w_{t_2}') \in \{0, 1\}$. It is necessary to run the algorithm, since it can test the consistence between the secret of redistribution and the original secret.

● *A* New-share Presentation Validity Verification Algorithm $V_4'$

Since $V_4'(w_j', s_j') \in \{0, 1\}$, it can be detected whether each $j(1 \leqslant j \leqslant t_2)$ in $P'$ sends a false new-share.

● *A* Reconstruction Algorithm $R'$ which takes as input all the new-shares presented by all the members in $P'$ and outputs the shared secret $s$.

$$R'(s_1', s_2', \cdots, s_{t_2}') = s$$

The security in the redistribution stage of the shared secret is based on the following conditions:

Extracting the sub-shares $\{s_{ij} \mid i=1, 2, \cdots, n_1; j=1, 2, \cdots, n_2\}$ from the witnesses $\{w_{ij} \mid i=1, 2, \cdots, n_1; j=1, 2, \cdots, n_2\}$ is infeasible.

Extracting the sub-shares $\{s_{ij} \mid i=1, 2, \cdots, n_1; j=1, 2, \cdots, n_2\}$ from the witnesses $\{w_{ij} \mid i=1, 2, \cdots, n_1; j=1, 2, \cdots, n_2\}$ and the sub-shares' ciphertexts $\{c_{ij} \mid i=1, 2, \cdots, n_1; j=1, 2, \cdots, n_2\}$ is infeasible.

## 3 Double discrete logarithm equality protocol

Say we have a group $G$ of prime order $p$ with generators $g_1$ and $g_2$ so that computing discrete logarithms to the bases $g_1$ and $g_2$ is difficult. Let $h \in Z_p^*$ be an element of prime order $q=(p-1)/2$. By the double discrete logarithm of $y \in G$ to the bases $g$ and $h$, we mean the unique $x \in Z_p$ with $y = g_1^{(h^x)}$.

Now, a prover knows a secret $x \in Z_p$ which $z = g_2^x$ and $y = g_1^{(h^x)}$. The prover proves to the verifier that the double discrete logarithm of $y$ to the bases $g_1$ and $h$ is equal to the discrete logarithm of $z$ to the base $g_2$ on condition that the prover can't disclose the secret number $x$. The construction of the double discrete logarithm equality proof is as follows.

Common input: $G, g_1, g_2, h, y, z$.

Prover's first step: The prover selects randomly an element $w$ in $Z_p$ and computes $y' = g_1^{(h^w)}$ and $z' = g_2^x$, then sends $y'$ and $z'$ to the verifier.

Verifier's first step: The verifier chooses at random a challenge $i \in \{0, 1\}$, and sends the challenge to the prover.

Prover's second step: The prover sends $r$ to the verifier, where $r = w - ix \pmod q$.

Verifier's second step: The verifier checks whether $z' = h^r z^i \pmod p$, and $y' = g_1^{(h^r)} \pmod p$, if $i=0$; or $y' = y \cdot g_1^{(h^r)} \pmod p$, if $i=1$.

The process above repeats $n$ times. When the verifier has found that these equations hold in Verifier's second step, the verifier believes the prover has the secret number $x$. Clearly, the prover's success probability of cheating the verifier is $1/2$ in one round. The probability will be $1/2^n$ after $n$ rounds. If $n$ is a large integer, the probability is negligible.

By the same technique of converting Fieige-Fiat-Shamir identification protocol into Fiat-Shamir signature protocol, the DDLE protocol above can turn into a non-interactive proof. In fact, the non-interactive proof of DDLE is zero-

knowledge[12].

## 4 PVSR scheme

Let $p$ be a large prime such that $q=(p-1)/2$ is also a prime. Let $G$ be a group of order $q$ with a generator $g$ so that the discrete logarithm to the base $g$ and the discrete logarithm to the base $h$ are difficult. Let $h \in Z_p^*$ be an element of order $q$. The secret to be shared is $s \in Z_p$.

Our scheme first distributes the secret $s$ to the access structure $A^{\{t_1, n_1\}}$, then redistributes $s$ from $A^{\{t_1, n_1\}}$ to $A^{\{t_2, n_2\}}$ structure. $H: \{0,1\}^* \rightarrow \{0,1\}^m$ is a public cryptographically strong Hash function. Note that $n_1 < p$, $n_2 < p$. Each $i$ in any authorized subset $P=\{P_1, P_2, \cdots, P_{n_1}\}$ of $A^{\{t_1, n_1\}}$. The participant $P_i$ in $P$ at random chooses an integer $z_i$ in $Z_p^*$ as its secret key $sk_i$, and computes $y_i = g^{z_i} \pmod p$ as its public key.

**The first distribution:**

Step 1 The dealer selects at random a polynomial,

$$f(x) = s + a_1 x + a_2 x^2 + \cdots + a_{t_1-1} x^{(t_1-1)} \in_R Z_p[x], \text{ where } \deg(f(x)) = t_1 - 1,$$

and computes

$$s_i = f(ID_i) \pmod p, w = g^s \pmod p, F_k = g^a k \pmod p, \tag{1}$$

where $i=1, 2, \cdots, n_1$, $k=1, 2, \cdots, t_1-1$, then publishes $(w, F_1, F_2, \cdots, F_{t_1-1})$.

Step 2 The dealer encrypts the share $s_i$:

$$\alpha_i \in_R Z_q, \ \nu_i = g^{\alpha_i} \pmod p, b_i = s_i^{-1} y_i^{\alpha} \pmod p, \tag{2}$$

where $(\nu_i, b_i)$ is the ciphertext of the secret share $s_i$. Then the dealer chooses randomly $u_j, w_j \in_R Z_q$, $j=1, 2, \cdots, m$, and computes

$$h_{ij} = h^{u_j} \pmod p, \quad h_{ij}' = g^{y_i^{w_i}} \pmod p,$$
$$R_i = (r_{i1}, r_{i2}, \cdots, r_{im})$$
$$= (w_1 - c_{i1}\alpha_i, w_2 - c_{i2}\alpha_i, \cdots, w_m - c_{im}\alpha_i) \pmod p,$$

where $c_i = H(ID_i \| w_i \| \nu_i \| b_i \| h_{i1} \| h_{i1}' \| \cdots \| h_{im} \| h_{im}')$, $c_{ij}$ denotes the $j$-th bit of $c_i$.

Finally, the dealer publishes the tuple $(w, F_1, F_2, \cdots, F_{t_1-1})$ and $(ID_i, \nu_i, b_i, R_i, c_i)$, where $i=1, 2, \cdots, t_1$.

Step 3 Every $i$ in $P$ obtains the ciphertext $c_i$ from the bulletin board and decrypts $s_i = D(c_i, sk_i) = \nu_i^{z_i}/b_i \pmod p$, and computes $g^{s_i} \pmod p$, $w_i = w \circ \prod_{j=1}^{t_1-1} F_j^{(ID_i)^j} \pmod p$. Then $P_j$ checks if $w_i = g^{s_i} \pmod p$, and publishes the result of validity test.

Step 4 Every $j$ checks the validity of the share $s_i$, where $i, j \in \{1, 2, \cdots, t_1-1\}$ and $j \neq i$. First, $P_j$ computes

$$w_i = w \circ \prod_{j=1}^{t_1-1} F_j^{(ID_i)^j} \pmod p, \quad h_{ik} = h^{r_k} \pmod p,$$

$$h_{ik}' = (g^{1-c_{ik}} w_{ik}^{c_{ik}})^{(y_i)^{r_k}} \pmod p, \quad k = 1, 2, \cdots, m,$$
$$c_i' = H(ID_i \| w_i \| \nu_i \| b_i \| h_{i1} \| h_{i1}' \| \cdots \| h_{im} \| h_{im}').$$

Then $P_j$ checks whether $c_i = c_i'$. Finally, $P_j$ publishes the result of the check about $s_i$.

Setp 5 After each $i$ sends its share $s_i$, other participants can decide whether $P_i$'s behavior is honest. If $i$ sends its correct secret share, the equation $w_i = g^{s_i} \pmod p$ holds.

Step 6　The participants pool their shares to recover the shared secret by the Lagrange interpolation formula:

$$s=\sum_{i=1}^{t_1-1} s_i \cdot d_{P_i}, \quad \text{where } d_{P_i}=\prod_{j\in P/\{i\}}\frac{ID_j}{ID_j-ID_i}. \tag{3}$$

**The Redistribution of secret from $A^{\{t_1,n_1\}}$ to $A^{\{t_2,n_2\}}$:**

Suppose $P'=\{P_1', P_2', \cdots, P'_{n_2}\}$ is an authorized subset of $A^{\{t_2,n_2\}}$. Each $i$ in $P'$ has an identification $ID_i'$ and a key pair $(pk_i', sk_i')=(y_i', z_i')$, where $y_i'=g^{z_i'} \pmod p$.

Step 1　Each $i$ in $P$ acts as a dealer. $P_i$ selects at random $f_i(x)=s_i+a_{i1}x+a_{i2}x^2+\cdots+a_{i,t_2-1}x^{(t_2-1)}\in_R Z_p[x]$ and computes the sub-share $s_{ij}=f_i(ID_j') \pmod p$, the witness $w_{ij}=g^{s_{ij}} \pmod p$ and $F_{ik}=g^{a_{ik}} \pmod p$, where $j=1, 2, \cdots, n_2$ and $k=1, 2, \cdots, t_2-1$. Then $P_i$ encrypts the sub-shares as follows:

$$\alpha_{ij}\in_R Z_q, \quad a_{ij}=g^{\alpha_{ij}} \pmod p,\ b_{ij}=s_{ij}^{-1}(y_j')^{\alpha_{ij}} \pmod p. \tag{4}$$

The ciphertext of $s_{ij}$ is $(a_{ij}, b_{ij})$. $P_i$ chooses randomly $\mu_{ijl}\in_R Z_q$, where $l=1, 2, \cdots, m$ and computes

$$\begin{aligned}
&h_{ijl}=h^{\mu_{ijl}} \pmod p,\ h'_{ijl}=g^{(y_j^{\mu_{ijl}})} \pmod p,\\
&R_{ij}=(r_{ij1}, r_{ij2}, \cdots, r_{ijm})\\
&\quad=(\mu_{ij1}-c_{ij1}\alpha_{ij}, \mu_{ij2}-c_{ij2}\alpha_{ij}, \cdots, \mu_{ijm}-c_{ijm}\alpha_{ij}) \pmod p,
\end{aligned}$$

where $c_{ij}=H(ID_j' \| w_{ij} \| a_{ij} \| b_{ij} \| h_{ij1} \| h'_{ij1} \| \cdots \| h_{ijm} \| h'_{ijm})$, $c_{ijk}$ specifies the $k$-th bit of $c_{ij}$. Finally, $P_i$ publishes $(F_{i1}, F_{i2}, \cdots, F_{i,t_2-1})$ and $(ID_j', R_{ij}, c_{ij})$, where $j=1, 2, \cdots, t_2-1$.

Step 2　Each $i$ in $P'$ can check the validity of shares. Through the dealer's bulletin board, $P_i'$ can obtain $F_j$ $(j=1, 2, \cdots, t_1-1)$ and compute $w_j$, then checks whether $w_i=\prod_{i=1}^{t_1-1}F_k^{(ID_i)^k} \pmod p$, and

$$w=\prod_{i=1}^{t_1-1}(w_i)^{d_{P_i}}, \quad \text{where } d_{P_i}=\prod_{j\in P/\{i\}}\frac{ID_j}{ID_j-ID_i}. \tag{5}$$

Step 3　Each $i$ in $P'$ obtains the sub-share $s_{ij}$ from $j$ in $P$ by decrying $(a_{ij}, b_{ij})$, $s_{ij}=a_{ij}^{z_i}/b_{ij} \pmod p$. Then $P_i'$ computes the witness of $s_{ij}$,

$$w_{ij}=w_i\prod_{k=1}^{t_2-1}F_{ik}^{(ID_j')^k} \pmod p, \tag{6}$$

and checks if $P'$ computes $w_{ij}=g^{s_{ij}} \pmod p$.

Step 4　Without knowing $s_{ij}$, each $l(\neq j)$ in $P'$ can check $P_j'$'s sub-share $s_{ij}$ from $P_i$. $P_l'$ first computes $w_{ij}$, then computes

$$\begin{aligned}
&h_{ijv}=h^{r_{ijv}}a_{ij}^{c_{ijv}} \pmod p,\\
&h'_{ijv}=(g^{1-c_{ijv}}w_{ij}^{b_{ij}c_{ijv}})^{(y_j')r_{ijv}} \pmod p,\\
&c'_{ij}=H(ID_j' \| w_{ij} \| a_{ij} \| b_{ij} \| h_{ij1} \| h'_{ij1} \| \cdots \| h_{ijm} \| h'_{ijm}),
\end{aligned}$$

finally checks if $c_{ij}'=c_{ij}$.

Step 5　Each $j$ in $P'$ generates its new-share $s_j'$:

$$s_j'=\sum_{i\in P}s_{ij}d_{P_i} \pmod p, \quad \text{where } d_{P_i}=\prod_{k\in P/\{i\}}\frac{ID_k}{ID_k-ID_i}, \tag{7}$$

then computes $w_j'=\sum_{i\in P}w_{ij}^{d_{P_i}} \pmod p$, and decides the validity of its new-share $s_j'$ by whether the equation $w_j'=g^{s_j'} \pmod p$ holds.

Step 6　Participants checks the validity of the new-shares:

$$w=\prod_{j\in P'}(w_j')^{d_{P_j'}} \pmod p, \quad \text{where } d_{P_j'}=\prod_{k\in P'/\{j\}}ID_k'^{ID_k'}-ID_j'. \tag{8}$$

Step 7 Every participant in $P'$ sends its new-share and other participants can test its validity by computing the equation in Step 5.

Step 8 If all the validity algorithms output 1, the participants in $P'$ pool to recover the shared secret $s$.

## 5 Analysis of the PVSR scheme

The following three theorems show the correctness of the validity tests.

**Theorem 1** In the first distribution of the secret, if $w_i = g^{s_i} \pmod p$ holds, $s_i$ is valid, where $i = 1, 2, \cdots, n_1$.

**Proof** By Equation 1, we have

$$s_i = f(ID_i) = s + a_1 ID_i + a_2 ID_i^2 + \cdots + a_{t_1-1} ID_i^{t_1-1} \pmod p,$$

thus we can obtain

$$g^{s_i} = g^s \prod_{k \in P} g^{a_k ID_i^k} = g^s \prod_{k \in P} F_k^{(ID_i)^k} = w \prod_{k \in P} F_k^{(ID_i)^k} \pmod p.$$

**Theorem 2** In the redistribution of the secret, if $w_j' = g^{s_j'} \pmod p$ holds, $s_j'$ is valid.

**Proof** By Equation 7, we have

$$s_j' = \sum_{i \in P} s_{ij} d_{P_i}, \text{ where } d_{P_i} = \prod_{k \in P \setminus \{i\}} \frac{ID_k}{ID_k - ID_i}.$$

We can obtain

$$g^{s_j'} = g^{\sum_{i \in P} s_{ij} d_{P_i}} = \prod_{i \in P} w_{ij}^{d_{P_i}} \pmod p = w_j'.$$

**Theorem 3** If $g^s = \prod_{j \in P'} g^{s_j' d_{P_j'}} \pmod p$, the new-shares are valid.

**Proof** Since Equation 3 and Equation 7, we have

$$s = \sum_{i \in P} s_i d_{P_i}, \quad s_i = \sum_{j \in P'} s_{ij} d_{P_j'},$$

hence

$$\begin{aligned} g^s &\equiv \prod_{i \in P} \left( \prod_{j \in P'} g^{s_{ij} d_{P_j'}} \right)^{d_{P_i}} \\ &\equiv \prod_{j \in P'} \prod_{i \in P} \left( \prod_{j \in P'} g^{s_{ij} d_{P_j'}} \right)^{d_{P_i}} \\ &\equiv \prod_{j \in P'} g^{s_j' d_{P_j'}} \pmod p. \end{aligned}$$

Since the non-interactive proof in the scheme is zero knowledge[12], so it won't leak the secret $\alpha_i$ or $\alpha_{ij}$. According to the discussion about the security of the scheme in Ref. [6] and Ref. [8], we have the security theorem.

**Theorem 4** (PVSR security) Under the assumptions that computing the discrete logarithm is infeasible and ElGamal-like encryption systems is unbroken, the scheme won't disclose the shares and new-shares and any unauthorized subset in $A^{\langle t_1, n_1 \rangle}$ and $A^{\langle t_2, n_2 \rangle}$ can obtain the shared secret.

## 6 Conclusion

This paper proposes a PVSR model and presents a PSVR scheme based on Ref. [6] and Ref. [8]. In the scheme, we needn't build any secret channel between the dealer and every participant, between the participants at the different distribution levels. In the public system, the scheme is convenient since the participant in the scheme won't need to have one more key pair. The validity of a participant's share (new-share) can be checked not only by

itself but also by all other participants on the condition that they don't know the share (new-share). Thus, the scheme can prevent the cheating behavior.

## References


[1] B Blakley. Safeguading cryptographic keys. In: Proceedings of the National Computer Conference. 1979. 313—317

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

[3] B Chor, S Goldwasser, S Milica, B Awerbuch. Verifiable secret sharing and acheiving simultaneity in the presence of faults. In: Proceedings of the 26th IEEE Syposium on Foundations of Computer Science. Washington: IEEE Computer Society Press, 1985. 383—395

[4] T Pedersen. Non-Interactive and information-theoretic secure verifiable secret sharing. In: Advances in Cryptology Crypt'91. Berlin: Springer-Verlag, 1991. 129—140

[5] P Feldman. A practical scheme for non-interactive verifiable secret sharing. In: Processing of the 28th IEEE Syposium on Foundatons of Computer Science. Washington: IEEE Computer Society Press, 1987. 427—437

[6] Markus Stadler. Publicly verifiable secret sharing. In: EUROCRYPT'96 Proceeding: LNCS 1070. 1996. 190—199

[7] J Camenishch, J M Piveteau, M Stadler. An efficient fair payment system. In: Proc 3rd ACM Conference on Computer and Communications Security. 1996

[8] Theodore M Wong, Chjen-Xi Wang, Jeannette M Wing. Verifiable secret redistribution for threshold sharing schemes. In: ASIAOCRYPT'02 Proceeding: LNCS. 2002

[9] R Gennaro. Theory and practice of veriable secret sharing: [Ph. D Thesis]. Massarchusetts Institute of Technology (MIT), 1996

[10] Y Desmedt, S Jajodia. Redistributing secret shares to new access structures and its applications. Technical Report ISSE TR-97-01, Fairfax: George Mason University, 1997

[11] Y Frankel, P Gemmell, P D MacKenzie, M Yung. Proactive RSA. In: Processing of CRYPTO'97: LNCS 1294. 1997. 440—454

[12] J Camenish. Group signature schemes and payment systems based on discrete logarithm problem: [Ph. D. Thesis]. Konstanz: Hartung-Gorre Verlag, 1998


# 公开可验证的秘密重新分配门限方案


谭作文　刘卓军

(中国科学院数学与系统科学研究院数学机械化重点实验室, 北京 100080)



**摘　要**　建立了一个公开可验证秘密重新分配模型, 并提出了一个公开可验证秘密重新分配方案. 这个方案将秘密从$(t_1, n_1)$接入结构分配到$(t_2, n_2)$接入结构. 在对旧的份额和子份额的有效性进行检验后, 新的分享者能产生有效的新份额. 在此方案中, 新的分享者能验证其他分享者的子份额的有效性.



**关键词**　可验证秘密共享, 秘密重新分配, 门限方案, 接入结构