# Provably Secure Metering Scheme

Wakaha Ogata and Kaoru Kurosawa

Tokyo Institute of Technology,
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
{wakaha, kurosawa}@ss.titech.ac.jp

**Abstract.** Naor and Pinkas introduced metering schemes at Eurocrypt '98 in order to decide on advertisement fees for web servers. In the schemes, any server should be able to construct a *proof* to be sent to an audit agency if and only if it has been visited by at least a certain number, say $k$, of clients. This paper first shows an attack for their schemes such that only two malicious clients can prevent a server from computing a correct proof. We next present provably secure metering schemes. Finally, an efficient robust secret sharing scheme is derived from our metering scheme.

## 1 Introduction

In the Internet, the amount of money paid to a web server from an advertisement company for hosting an ads should depend on the number of clients which have visited the server. A metering scheme is a protocol which measures this number.

We assume an audit agency as well as servers and clients. Any server should be able to construct a *proof* to be sent to the audit agency if and only if it has been visited by at least a certain number, say $k$, of clients during a certain time frame. It should be secure against fraud attempts by servers which inflate the number of their clients and against clients that attempt to disrupt the metering process.

A naive metering scheme could be implemented by using digital signature schemes; each client gives a digital signature to a server which confirms his visit when the clients visits the server. A server can present a list of the digital signatures as a *proof*. This system is, however, not efficient: both the size of the *proof* and the time to verify it are of the same order as $k$. Naor and Pinkas showed much more efficient metering schemes at Eurocrypt'98 [2].

This paper first shows an attack for their schemes such that only two malicious clients can prevent a server from computing a correct proof. We next present provably secure metering schemes, an unconditionally secure one and a computationally secure variant for multiple use under the computational Diffie-Hellman assumption.

We finally derive an efficient robust secret sharing scheme from our unconditionally secure metering scheme. In our robust secret sharing schemes, the size of shares is much smaller than those of the previous ones while the cheating probability is slightly larger.

T. Okamoto (Ed.): ASIACRYPT 2000, LNCS 1976, pp. 388–398, 2000.

## 2 Model and Goal

In the model of metering schemes, there exist *clients* (denoted by $\mathcal{C}_i$), *servers* (denoted by $\mathcal{S}_j$) and an *audit agency* (denoted by $\mathcal{A}$). Each $\mathcal{S}_j$ should be able to construct a *proof* to be sent to $\mathcal{A}$ if and only if $k$ or more clients visit $\mathcal{S}_j$ during a certain time frame.

Some clients and servers are malicious while the audit agency is honest. Assume that there exists an adversary which corrupts some clients and servers. Then our goal is to design metering schemes which satisfy the following two requirements.

**Security for servers** Suppose that $k$ or more clients visit a server $\mathcal{S}_j$ during a time frame $t$. Then $\mathcal{S}_j$ should be able to compute a *proof* with overwhelming probability even if the adversary corrupts all the clients and all the other servers than $\mathcal{S}_j$.

**Security for the audit agency** Suppose that less than $k$ clients visit a server $\mathcal{S}_j$ during a time frame $t$. Then the adversary should not be able to compute a *proof* with nonnegligible probability even if the adversary corrupts $k-1$ clients and some number of servers.

It will be shown that the metering schemes of Naor and Pinkas do not satisfy the security for servers for an adversary who corrupts only two clients. On the other hand, the proposed metering schemes satisfy both of the above two requirements.

## 3 Metering Schemes of Naor and Pinkas

### 3.1 Basic Idea

The metering schemes of Naor and Pinkas are based on Shamir's secret sharing scheme. First, suppose that there exist a single server and a single time frame, and all the participants are honest. Then their basic scheme is described as follows. The audit agency chooses a random polynomial $f(x)$ of degree $k-1$ over $GF(p)$. He gives each client $\mathcal{C}_i$ a share $f(i)$. When a client visits a server, it gives it its share. When the server receives $k$ or more shares, he can compute $f(0)$ and it is the *proof* that he was visited by $k$ or more clients.

To accommodate many servers and many time frames, this scheme is generalized as follows. The audit agency chooses a random polynomial $P(x,y)$ over $GF(p)$ of degree $k-1$ in $x$ and degree $d-1$ in $y$. He gives $P(i,y)$ to each client $\mathcal{C}_i$. When client $\mathcal{C}_i$ visits server $\mathcal{S}_j$ during time frame $t$, it gives $P(i, j \circ t)$ to $\mathcal{S}_j$. When $\mathcal{S}_j$ receives $k$ or more shares during time frame $t$, he can compute $P(0, j \circ t)$ and it is the *proof* that he was visited by $k$ or more clients.

This scheme is one-time use because the size of keys is proportional to the number of time frames for fixed $k$ and fixed number of servers.

### 3.2 Unconditionally Secure Scheme [2, Sec.3.3]

Naor and Pinkas then proposed the following unconditionally secure scheme to make the above scheme secure against malicious clients and servers. This is a one-time use scheme as mentioned above.

**Initialization:** The audit agency $\mathcal{A}$ chooses random polynomials $P(x,y), A(x,y)$ and $B(y)$ over $GF(p)$ such that
- $P(x,y)$ is degree $k-1$ in $x$ and degree $d-1$ in $y$,
- $A(x,y)$ is degree $c_k$ in $x$ and degree $c_d$ in $y$,
- $B(y)$ is degree $c_d$ in $y$.

$\mathcal{A}$ computes

$$V(x,y) \stackrel{\triangle}{=} A(x,y)P(x,y) + B(y), \tag{1}$$

It then sends $(V(i,y), P(i,y))$ to $\mathcal{C}_i$ and $(A(x, j \circ 1), \ldots, A(x, j \circ T), B(j \circ 1), \ldots, B(j \circ T))$ to $\mathcal{S}_j$, where $\circ$ denotes concatenation of two strings.

**Interaction between client $\mathcal{C}_i$ and server $\mathcal{S}_j$:** To get a service from server $\mathcal{S}_j$ at time frame $t$, client $\mathcal{C}_i$ sends

$$(P(i, j \circ t), V(i, j \circ t))$$

to $\mathcal{S}_j$. The $\mathcal{S}_j$ checks if

$$V(i, j \circ t) = A(i, j \circ t)P(i, j \circ t) + B(j \circ t).$$

$\mathcal{S}_j$ offers a service to $\mathcal{C}_i$ if the above equality holds. Otherwise, $\mathcal{S}_j$ rejects.

**End of time frame:** If $\mathcal{S}_j$ has been visited by $k$ or more clients at time frame $t$, it can compute $P(0, j \circ t)$ from the received $P(i, j \circ t)$. The $P(0, j \circ t)$ is the *proof* that $\mathcal{S}_j$ has been visited by $k$ or more clients at time frame $t$. $\mathcal{A}$ who received $P(0, j \circ t)$ checks whether it is indeed $P(0, j \circ t)$.

### 3.3 Computationally Secure Scheme [2, Sec.3.5]

Naor and Pinkas further presented a computationally secure variant for multiple use under the computational Diffie-Hellman assumption.

Let $Z_p^*$ be the cyclic group modulo $p$, and let $g$ be a generator of a subgroup of $Z_p^*$ of order $q$, where $q$ is a prime.

**Initialization:** Similarly to the scheme of Sec.3.2, client $\mathcal{C}_i$ receives $P(i,y)$ and $V(i,y)$ and server $\mathcal{S}_j$ receives $A(x,j)$ and $B(j)$.

**Beginning of a time frame:** Each server receives a challenge $h = g^r$ from the audit agency, where $r$ is a random number.

**Interaction between client $\mathcal{C}_i$ and server $\mathcal{S}_j$:** To get a service from server $\mathcal{S}_j$, client $\mathcal{C}_i$ receives $h$ from $\mathcal{S}_j$ and sends

$$c_{i,j} \stackrel{\triangle}{=} (h^{P(i,j)}, h^{V(i,j)})$$

to server $\mathcal{S}_j$. $\mathcal{S}_j$ accepts $c_{i,j}$ if and only if

$$h^{V(i,j)} = (h^{P(i,j)})^{A(i,j)} h^{B(j)} \bmod p.$$

**End of time frame:** $\mathcal{S}_j$ can compute $h^{P(0,j)}$ if it has been visited by $k$ or more clients. The $h^{P(0,j)} (= g^{rP(0,j)})$ is the proof.

## 4 Attack for Naor and Pinkas Metering Schemes

In this section, we show an attack for both of the Naor and Pinkas metering schemes. In our attack, two clients, one who has special share and the other is arbitrary, can prevent a server from computing a proof. In other words, the security for servers is not satisfied. We describe our attack for the unconditionally secure scheme. It works for the computationally secure scheme similarly.

For some server $\mathcal{S}_j$ and some time frame $t$, suppose that there exists two clients $\mathcal{C}_{i_0}$ and $\mathcal{C}_{i_1}$ such that

$$P(i_0, j \circ t) = 0$$
$$P(i_1, j \circ t) \neq 0.$$

Then, from Equation (1) they can compute $B(j \circ t)$ and $A(i_1, j \circ t)$ as follows.

$$\begin{aligned} V(i_0, j \circ t) &= A(i_0, j \circ t)P(i_0, j \circ t) + B(j \circ t) \\ &= B(j \circ t) \\ A(i_1, j \circ t) &= (V(i_1, j \circ t) - B(j \circ t))/P(i_1, j \circ t) \\ &= (V(i_1, j \circ t) - V(i_0, j \circ t))/P(i_1, j \circ t). \end{aligned}$$

They next computes a random $(\tilde{P}, \tilde{V})$ such that

$$\tilde{P} \neq P(i_1, j \circ t)$$
$$\tilde{V} = A(i_1, j \circ t)\ \tilde{P} + B(j \circ t).$$

Finally, $\mathcal{C}_{i_1}$ sends $(\tilde{P}, \tilde{V})$ to $\mathcal{S}_j$ at time frame $t$ to get a service. Then $\mathcal{S}_j$ accepts $(\tilde{P}, \tilde{V})$ because eq.(1) is satisfied.

However, at the end of time frame $t$. $\mathcal{S}_j$ cannot compute the correct $P(0, j \circ t)$ even if it has been visited by $k$ clients because $\tilde{P} \neq P(i_1, j \circ t)$

## 5 Proposed Unconditionally Secure Metering Scheme

In this section, we present an unconditionally secure metering scheme which satisfies both the security for servers and the security for the audit agency. We assume that there are $T$ time frames. This scheme is one-time use and the size of keys is essentially proportional to $T$.

### 5.1 Proposed Scheme

**Initialization:** The audit agency $\mathcal{A}$ chooses a key polynomial $F(x, y, z)$ over $GF(p)$ with degree 1 in $x$, degree $d-1$ in $y$ and degree $k-1$ in $z$ randomly. He also chooses a random element $r_j \in Z_p \setminus \{0\}$ for each server $\mathcal{S}_j$. Let

$$P_i(x, y) \stackrel{\triangle}{=} F(x, y, i)$$
$$A_j^t(z) \stackrel{\triangle}{=} F(r_j, j \circ t, z).$$

He then sends $c_i \stackrel{\triangle}{=} P_i(x, y)$ to client $\mathcal{C}_i$ and $s_j \stackrel{\triangle}{=} (A_j^1(z), \ldots, A_j^T(z), r_j)$ to server $\mathcal{S}_j$.

**Interaction between a client $\mathcal{C}_i$ and a server $\mathcal{S}_j$:** To get service from $\mathcal{S}_j$ at time frame $t$, $\mathcal{C}_i$ sends

$$c_{i,j}^t \stackrel{\triangle}{=} P_i(x, j \circ t)$$

to $\mathcal{S}_j$. The $\mathcal{S}_j$ checks if

$$A_j^t(i) = P_i(r_j, j \circ t).$$

$\mathcal{S}_j$ offers a service to $\mathcal{C}_i$ if the above equality holds. Otherwise, $\mathcal{S}_j$ rejects.

**End of time frame:** Note that if $\mathcal{C}_i$ visits $\mathcal{S}_j$ during time frame $t$, then $\mathcal{S}_j$ can compute

$$P_i(0, j \circ t) = F(0, j \circ t, i).$$

Therefore, if $\mathcal{S}_j$ has been visited by $k$ or more clients during time frame $t$, then $\mathcal{S}_j$ can compute $F(0, j \circ t, 0)$. The $F(0, j \circ t, 0)$ is the *proof* that $\mathcal{S}_j$ has been visited by $k$ or more clients during time frame $t$.

The audit agency $\mathcal{A}$ who received $F(0, j \circ t, 0)$ checks whether it is indeed $F(0, j \circ t, 0)$.

The scheme is illustrated in Fig. 1.

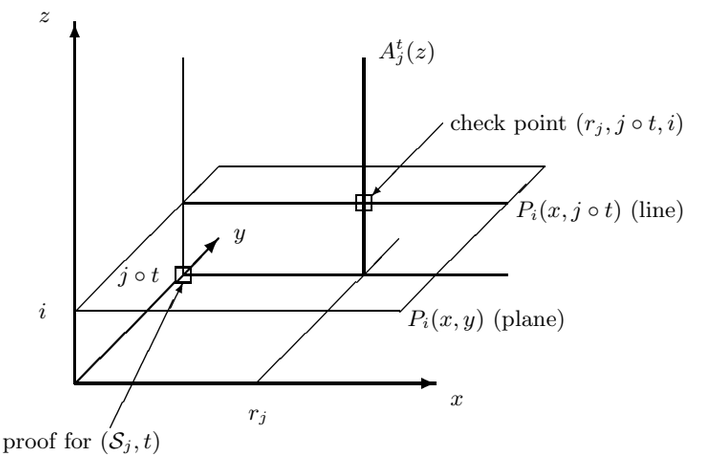


**Fig. 1.** Robust metering scheme

### 5.2 Security for Servers

In this subsection, we prove that the security for servers is satisfied for any infinitely powerful adversaries.

**Theorem 1.** *Suppose that $k$ or more clients visit server $\mathcal{S}_j$ during time frame $t$. Then $\mathcal{S}_j$ can compute the proof $F(0, j \circ t, 0)$ with probability more than $1-1/(p-1)$ for any adversary who corrupts all the clients and all the servers other than $\mathcal{S}_j$.*

*Proof.* Note that $\deg P_j(x, j \circ t) = 1$. Now at least one client $C_i$ must send $\tilde{P}(x)$ of degree 1 to $\mathcal{S}_j$ such that $\tilde{P}(x) \neq P_i(x, j \circ t)$ and $\tilde{P}(r_j) = P_i(r_j, j \circ t)$ to prevent $\mathcal{S}_j$ from computing the proof. For any fixed $\tilde{P}(x)$ such that $\tilde{P}(x) \neq P_i(x, j \circ t)$, we have $\tilde{P}(r_j) = P_i(r_j, j \circ t)$ with probability $1/(p-1)$ because $r_j$ is randomly chosen from $Z_p \setminus \{0\}$ and two lines intersect at one point. This holds for any adversary who corrupts all the clients and all the servers other than $\mathcal{S}_j$ because the adversary does not know $r_j$. □

### 5.3 Security for the Audit Agency

In this subsection, we prove that the security for the audit agency is satisfied for any infinitely powerful adversaries.

**Theorem 2.** *Suppose that less than $k$ clients visit server $\mathcal{S}_j$ during time frame $t$. Then no adversary who corrupts $d/T$ servers and $k-1$ clients can compute the proof $F(0, j \circ t, 0)$ for any $j$ and $t$ with probability more than $1/p$.*

*Proof.* Let $\alpha_s \stackrel{\triangle}{=} d/T$. Without loss of generality, we consider an adversary who corrupts $\alpha_s$ servers $\mathcal{S}_1, \ldots, \mathcal{S}_{\alpha_s}$ and $k-1$ clients $\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$. The adversary tries to forge a false proof $\tilde{F}(0, \alpha_s \circ T, 0)$ for server $\mathcal{S}_{\alpha_s}$ and the last time frame $T$.

We assume that for any $j \leq \alpha_s$ and any $t \leq T$ such that $j \circ t \neq \alpha_s \circ T$, server $\mathcal{S}_j$ has been visited by $k$ or more clients during time frame $t$. Then all the information that the adversary has are (a) the initial secrets of the corrupted clients, (b) the initial secrets of the corrupted servers and (c) the information that the corrupted servers received from honest clients. (a) and (b) are

**(a)** $F(x, y, 1), \ldots, F(x, y, k-1)$,
**(b)** $r_1, \ldots, r_{\alpha_s}$ and

$$\begin{cases} F(r_1, 1 \circ 1, z), \ \ldots, F(r_{\alpha_s - 1}, (\alpha_s - 1) \circ 1, z), \ F(r_{\alpha_s}, \alpha_s \circ 1, z), \\ \qquad \vdots \qquad\qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ F(r_1, 1 \circ T, z), \ldots, F(r_{\alpha_s - 1}, (\alpha_s - 1) \circ T, z), \ F(r_{\alpha_s}, \alpha_s \circ T, z). \end{cases}$$

(c) is at most

$$\{F(x, j \circ t, i) \mid \text{ for all } i \geq k \text{ and for all } j \circ t \text{ such that } j \leq \alpha_s \text{ and } j \circ t \neq \alpha_s \circ T\}.$$

Suppose that the forged proof is $\beta$. For any value of $\beta$, we will show that there exists a key polynomial $\tilde{F}(x, y, z)$ which interpolates all the points of (a), (b), (c) and $\tilde{F}(0, \alpha_s \circ T, 0) = \beta$. This means that the probability that $\beta$ is the correct proof is $1/p$.

Note that $F(x, y, z)$ has degree 1 in $x$. Let $L_{\alpha_s \circ T}(x)$ be a line which interpolates $\tilde{F}(0, \alpha_s \circ T, 0) = \beta$ and $F(r_{\alpha_s}, \alpha_s \circ T, 0)$, where $F(r_{\alpha_s}, \alpha_s \circ T, 0)$ is obtained from (b). Next we can compute $F(0, j \circ t, 0)$ from (c) for all $j \circ t$ such that $j \leq \alpha_s$ and $j \circ t \neq \alpha_s \circ T$. Let $L_{j \circ t}(x)$ be a line which interpolates $F(0, j \circ t, 0)$ and $F(r_j, j \circ t, 0)$, where $F(r_j, j \circ t, 0)$ is obtained from (b).

Then we have $d$ lines

$$\{L_{j \circ t}(x) \mid j \leq \alpha_s \text{ and } t \leq T\}$$

because $\alpha_s T = d$. Next note that $F(x, y, z)$ has degree $d-1$ in $y$. Let $B(x, y)$ be a polynomial of degree 1 in $x$ and degree $d-1$ in $y$ such that $B(x, j \circ t) = L_{j \circ t}(x)$ for all $j \leq \alpha_s$ and $t \leq T$.

Finally, let $\tilde{F}(x, y, z)$ be a polynomial of degree 1 in $x$, degree $d-1$ in $y$ and degree $k-1$ in $z$ such that $\tilde{F}(x, y, 0) = B(x, y)$ and $\tilde{F}(x, y, i) = F(x, y, i)$ for $1 \leq i \leq k-1$, where $F(x, y, i)$ is obtained from (a).

We have to show that $\tilde{F}(x, y, z) = F(x, y, z)$ for all the points of (a), (b) and (c). It is clear that the claim holds for (a). Next we prove the claim for (b). Fix $j \leq \alpha_s$ and $t \leq T$ arbitrarily. Then from our construction, it is easy to see that $\tilde{F}(r_j, j \circ t, i) = F(r_j, j \circ t, i)$ for $0 \leq i \leq k-1$. Therefore,

$$\tilde{F}(r_j, j \circ t, z) = F(r_j, j \circ t, z). \tag{2}$$

Finally, we prove the claim for (c). Similarly to eq.(2), we can show that

$$\tilde{F}(0, j \circ t, z) = F(0, j \circ t, z). \tag{3}$$

From eq.(3), we have $\tilde{F}(0, j \circ t, i) = F(0, j \circ t, i)$. From eq.(2), we have $\tilde{F}(r_j, j \circ t, i) = F(r_j, j \circ t, i)$. Hence, it holds that $\tilde{F}(x, j \circ t, i) = F(x, j \circ t, i)$. □

## 6 Proposed Computationally Secure Scheme

In this section, we present a computationally secure variant for multiple use under the computational Diffie-Hellman assumption.

### 6.1 Proposed Scheme

Let $Z_p^*$ be the cyclic group modulo $p$, and let $g$ be a generator of a subgroup of $Z_p^*$ of order $q$, where $q$ is a prime.

**Initialization:** Similarly to the scheme of Sec.5.1, the audit agency $\mathcal{A}$ chooses a key polynomial $F(x, y, z)$ over $GF(q)$ with degree 1 in $x$, degree $d-1$ in $y$ and degree $k-1$ in $z$ randomly. He also chooses a random element $r_j \in Z_q \setminus \{0\}$ for each server $\mathcal{S}_j$. Let

$$P_i(x, y) \stackrel{\triangle}{=} F(x, y, i), \quad A_j(z) \stackrel{\triangle}{=} F(r_j, j, z).$$

He then sends $P_i(x, y)$ to client $\mathcal{C}_i$ and $(A_j(z), r_j)$ to server $\mathcal{S}_j$.

**Beginning of a time frame :** At the beginning of time frame $t$, the audit agency $\mathcal{A}$ publishes a challenge $h_t = g^{u_t} \bmod p$, where $u_t$ is a random number.

**Interaction between a client and a server:** When client $\mathcal{C}_i$ gets a service from server $\mathcal{S}_j$ at time frame $t$, he computes

$$\begin{aligned} P_i(x, j) &= a_{i,j} + b_{i,j}x, \\ d_{i,j}^t &= h_t^{a_{i,j}} \bmod p \\ e_{i,j}^t &= h_t^{b_{i,j}} \bmod p \end{aligned}$$

He then sends $(d_{i,j}^t, e_{i,j}^t)$ to $\mathcal{S}_j$. The $\mathcal{S}_j$ accepts $(d_{i,j}^t, e_{i,j}^t)$ if and only if

$$h_t^{A_j(i)} = d_{i,j}(e_{i,j})^{r_j} \bmod p \ (= h_t^{P_i(r_j,j)} \bmod p).$$

**End of time frame:** Note that if $\mathcal{C}_i$ visits $\mathcal{S}_j$ during time frame $t$, then $\mathcal{S}_j$ obtains

$$d_{i,j}^t = h_t^{a_{i,j}} = h_t^{P_i(0,j)} = h_t^{F(0,j,i)}.$$

Therefore, if $\mathcal{S}_j$ has been visited by $k$ or more clients during time frame $t$, then $\mathcal{S}_j$ can compute $h_t^{F(0,j,0)}$ by using Lagrange formula. The $h_t^{F(0,j,0)}$ is the *proof* that $\mathcal{S}_j$ has been visited by $k$ or more clients during time frame $t$. The audit agency $\mathcal{A}$ who received $h_t^{F(0,j,0)}$ checks whether it is indeed $h_t^{F(0,j,0)}$.

### 6.2 Security for Servers

In our scheme, the security for servers is satisfied for any infinitely powerful adversaries.

**Theorem 3.** *Suppose that $k$ or more clients visit server $\mathcal{S}_j$ during time frame $t$. Then $\mathcal{S}_j$ can compute the proof $g^{u_t F(0,j,0)}$ with probability more than $1-1/(q-1)$ for any adversary who corrupts all the clients and all the servers other than $\mathcal{S}_j$.*

*Proof.* Similar to the proof of Theorem 1. □

### 6.3 Security for the Audit Agency

In this subsection, we consider probabilistic polynomial time adversaries who can corrupt $d$ servers and $k-1$ clients.

**Theorem 4.** *Suppose that there exists an adversary $M_0$ who can compute the proof $h_t^{F(0,j,0)}$ for some $j$ and $t$ with nonnegligible probability. Then there exists a probabilistic polynomial time Turing machine $M_1$ which can solve the computational Diffie-Hellman problem.*

*Proof.* Without loss of generality, we assume that $M_0$ corrupts $d$ servers $\mathcal{S}_1,\ldots,\mathcal{S}_d$ and $k-1$ clients $\mathcal{C}_1,\ldots,\mathcal{C}_{k-1}$, and then compute the proof $h_T^{F(0,d,0)}$ for server $\mathcal{S}_d$ and time frame $T$ with nonnegligible probability.

Let the input to $M_1$ be $p, g, X(= g^{\alpha} \bmod p)$ and $Y(= g^{\beta} \bmod p)$. We will show that $M_1$ can generate a view of the adversary $M_0$ such that $\alpha = u_T$ and $\beta = F(0,d,0)$. Then by using $M_0$ as a subroutine, $M_1$ can obtain

$$h_T^{F(0,d,0)} = g^{u_T F(0,d,0)} = g^{\alpha\beta}$$

with nonnegligible probability. This means that $M_1$ can solve the computational Diffie-Hellman problem.

The view of the adversary $M_0$ consists of

**(a)** the initial secrets of the corrupted clients : $P_1(x,y),\ldots,P_{k-1}(x,y)$,
**(b)** the initial secrets of the corrupted servers : $r_1,\ldots,r_d$ and $A_1(z),\ldots,A_d(z)$,
**(c)** the challenges : $h_1,\ldots,h_T$,
**(d)** the information that the corrupted servers received from honest clients. This is at most

$$\{(d_{i,j}^t, e_{i,j}^t) \mid \text{for all } i \geq k \text{ and for all } (j,t) \text{ such that } j \leq d \text{ and } (j,t) \neq (d,T)\}.$$

Now $M_1$ generates (a),(b),(c) and (d) as follows.

**(a)** $M_1$ randomly chooses $P_1(x,y),\ldots,P_{k-1}(x,y)$.
**(b)** $M_1$ randomly chooses $r_1,\ldots,r_d$ and $A_1(0),\ldots,A_d(0)$. These determine $A_1(z),\ldots,A_d(z)$ because $\deg A_j(z) = k-1$ and

$$A_j(i) = P_i(r_j, j) \text{ for } 1 \leq i \leq k-1.$$

**(c)** $M_1$ randomly chooses $u_1,\ldots,u_{T-1}$ and let $h_i = g^{u_i}$ for $1 \leq i \leq T-1$. Let $h_T = X$.

Finally, $M_1$ generates the elements of (d) as follows. $M_1$ has

$$Y = g^{\beta} = g^{F(0,d,0)}.$$

By slightly modifying the proof of Theorem 2, $M_1$ can compute $(a_{i,j}, b_{i,j})$ for $j < d$ and $(g^{a_{i,d}}, g^{b_{i,d}})$, where

$$a_{i,j} + b_{i,j}x = P_i(x,j) = F(x,j,i).$$

Therefore,

1. For $j = d$ and $t \leq T-1$, $M_1$ can compute $(d_{i,d}^t, e_{i,d}^t)$ from $(g^{a_{i,d}}, g^{b_{i,d}})$ and $u_t$.
2. For $j < d$ and $t = T$, $M_1$ can compute $(d_{i,j}^T, e_{i,j}^T)$ from $(a_{i,j}, b_{i,j})$ and $X = g^{\alpha} = g^{u_T}$.
3. For $j < d$ and $t \leq T-1$, $M_1$ can compute $(d_{i,j}^t, e_{i,j}^t)$ from $(a_{i,j}, b_{i,j})$ and $u_t$.

□

# 7 New Robust Secret Sharing Scheme

## 7.1 Previous Schemes

In a $(k, n)$ threshold secret sharing scheme, a dealer distributes a secret $s$ to $n$ participants, $\mathcal{P}_1, \ldots, \mathcal{P}_n$ in such a way that $k$ or more participants can recover $s$ and $k-1$ or less participants have no information on $s$. A piece of information held by $\mathcal{P}_i$ is called a share and it is denoted by $v_i$.

In the reconstruction phase, $\mathcal{P}_i$ opens $v_i$ if he is honest. However, he may lie about $v_i$ if he is a cheater. A secret sharing scheme is called robust if a cheater can be identified with overwhelming probability.

Let $S$ denote the set of secrets and $V_i$ denote the set of possible shares of $\mathcal{P}_i$. T.Rabin and Ben-Or [3] showed a robust $(k, n)$ threshold secret sharing scheme (RB scheme) such that

$$\log_2 |V_i| = (3n-2) \log_2 |S|.$$

Carpentieri [1] showed a robust $(k, n)$ threshold secret sharing scheme such that

$$\log_2 |V_i| = (2n+k-1) \log_2 |S|.$$

In these schemes, $n-1$ cheaters cannot cheat an honest participant with probability more than $1/|S|$.

## 7.2 Proposed Scheme

In this section, we derive a new robust $(k, n)$ threshold secret sharing scheme from our metering scheme such that $|V_i|$ is much smaller than those of the previous schemes with slightly less cheater detection capability.

In our scheme,

$$\log_2 |V_i| = (2k+1) \log_2 |S|.$$

and $k-1$ cheaters cannot cheat an honest participant with probability more than $(k-1)/(|S|-1)$. Note that it is assumed that there are at most $k-1$ cheaters instead of $n-1$ cheaters. This is, however, not a problem because any $k$ participants (cheaters) can recover the secret.

Let $p$ be a prime and let $S \triangleq GF(p)$.

**Distribution phase:** For a secret $s \in GF(p)$, the dealer chooses a bivariate random polynomial over $GF(p)$ such that

$$F(x, y) = \sum_{l=0}^{k-1} \sum_{m=0}^{k-1} a_{lm} x^l y^m$$

with $F(0, 0) = a_{00} = s$. He also chooses $n$ random elements $r_i \in Z_p \setminus \{0\}$. Let

$$B_i(x) \triangleq F(x, i)$$
$$A_i(y) \triangleq F(r_i, y),$$

The dealer then gives $v_i = (B_i(x), r_i, A_i(y))$ to the participant $\mathcal{P}_i$.

**Reconstruction phase:** Each $\mathcal{P}_i$ opens $B_i(x)$. Each other $\mathcal{P}_j$ accepts $B_i(x)$ if and only if

$$B_i(r_j) = A_j(i).$$

Note that $k$ or more correct $B_i(x)(= F(x, i))$ uniquely determine $s = F(0, 0)$.

**Theorem 5.** *$k-1$ cheaters cannot cheat an honest participant with probability more than $(k-1)/(p-1)$.*

*Proof.* Suppose $k-1$ participants $\mathcal{P}_1, \ldots, \mathcal{P}_{k-1}$ conspire and try to cheat $\mathcal{P}_k$. Suppose $\mathcal{P}_1$ opens $\tilde{B}(x)$ such that $\tilde{B}(x) \neq B_1(x)$. $\mathcal{P}_1$ succeeds if $\tilde{B}(r_j) = A_j(1)$. Since $\tilde{B}(x)$ can be written as $\tilde{B}(x) = B_1(x) + \Delta B(x)$ for some polynomial $\Delta B(x) \neq 0$ with degree $k-1$ or less, this cheating probability is computed as follows.

$$\begin{aligned}
\Pr[\tilde{B}(r_j) = A_j(1)] &= \Pr[B_1(r_j) + \Delta B(r_j) = A_j(1)] \\
&= \Pr[\Delta B(r_j) = 0] \\
&= |\{r \in Z_p \setminus \{0\} \mid \Delta B(r) = 0\}| \;/\; |\{r \in Z_p \setminus \{0\}\}| \\
&\leq (k-1)/(p-1)
\end{aligned}$$

□

## References

1. Carpentieri, M.: A perfect threshold secret sharing scheme to identify cheaters. Designs, Codes and Cryptography, **5** (1995) 183–187
2. Naor, M., Pinkas, B.: Secure and Efficient Metering. Proc. of Eurocrypt'98, Lecture Notes in Computer Science, **1403**, Springer Verlag (1998) 576–589
3. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority. Proc. 21st ACM Symposium on Theory of Computing (1989) 73–85
4. Shamir, A.: How to share a secret. Comm. ACM, **22** (1979) 612–613