SCIS 2001 The 2001 Symposium on Cryptography and Information Security
Oiso, Japan, January 23-26, 2001
The Institute of Electronics, Information and Communication Engineers

安全性が証明可能な metering scheme
Provably Secure Metering Scheme

尾形わかは* 黒澤 馨 *
Wakaha OGATA Kaoru KUROSAWA

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.

Keywords: metering scheme, web servers, secret sharing scheme

1 はじめに

インターネットにおいて，広告主である企業から広告を掲載しているウェブサーバに支払われる広告料は，サーバを訪れたクライアント数に依存させるべきである．metering scheme は，サーバを訪れたクライアント数を調べるためのプロトコルである．

本研究で扱うモデルにおいては，複数のサーバ，多数のクライアント，audit agency が存在し，ある一定数以上のクライアントがあるサーバを訪れた場合に，そのサーバがそのことを audit agency に示すための *proof* を作成する．*proof* を作成できるクライアント数のしきい値を k とし，クライアント数の計数は，time frame と呼ばれる一定期間ごとに行うものとする．metering scheme には，クライアント数を水増ししようとする不正なサーバや，計数を無効にしようとする不正なクライアントに対する安全性が要求される．

metering scheme を容易に構築するには，以下のようにディジタル署名を用いればよい．クライアントはサーバを訪問する際に自分の署名をサーバに渡す．サーバは署名のリストを *proof* として audit agency に提示する．しかし，このシステムでは *proof* の大きさやこれをチェックする時間が k に比例するため，非効率である．これに対し，Naor と Pinkas は効率的な metering scheme を提案した [2].

本研究では，まず，Naor と Pinkas による metering scheme において，たった 2 人の不正クライアントがサーバをだますことができることを示す．次に，安全性が証明可能な metering scheme を提案する．第 1 の提案方式は無条件に安全であり，第 2 の提案方式は無限の time frame に対して適用でき，Diffie-Hellman 問題の難しさの基づいている．最後に，効率的な robust しきい値法が metering scheme と同様に構成可能であることを示す．提案する robust しきい値法においては，従来提案されている方式に比べ，不正検出能力が若干劣るものの，シェアサイズを小さくすることができる．

* 東京工業大学, 目黒区大岡山 2－12－1 Tokyo Institute of Technology, 2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan

2 モデルと目的

本研究の metering scheme のモデルでは，クライアント $\mathcal{C}_i$, サーバ $\mathcal{S}_j$, audit agency $\mathcal{A}$ の存在を仮定する．それぞれの $\mathcal{S}_j$ は k 以上のクライアントがそのサーバを訪れたことを証明する *proof* を作成する．

Audit agency は常に正しくふるまうとし，何人かのクライアントといくつかのサーバは不正をするかもしれないと仮定する．このとき，安全な metering scheme には，以下の 2 つの条件が要求される．

サーバに関する安全性: ある time frame に k 人以上のクライアントがサーバ $\mathcal{S}_j$ を訪れたとしよう．このとき，全てのクライアントと全てのサーバを買収した敵がいたとしても，$\mathcal{S}_j$ は *proof* を圧倒的確率で求めることができなくてはならない．

audit agency に関する安全性: ある time frame にサーバ $\mathcal{S}_j$ を訪れたクライアントが k 人未満だったとしよう．このとき，$k-1$ 人以上のクライアントといくつかのサーバが結託した敵がいたとしても，敵がこの time frame とサーバ $\mathcal{S}_j$ に対する *proof* を

求めることのできる確率は，無視できるほど小さくなくてはならない．

次の章で，Naor と Pinkas の提案した metering scheme はサーバに関する安全性が満たされていないことを示す．

3 Naor と Pinkas による metering scheme

3.1 基本的な考え方

Naor と Pinkas による metering scheme は，Shamir のしきい値法を基本としている．まず，1 つのサーバと 1 つの time frame のみを考え，すべてのクライアントは不正を行わないとしよう．この場合の metering scheme は，次のようになる．まず，audit agency はランダムに $GF(p)$ 上の $k-1$ 次多項式 $f(x)$ を選ぶ．そしてクライアント $\mathcal{C}_i$ にはシェア $f(i)$ を渡す．クライアントがサーバを訪問するときには，クライアントは自分のシェアをサーバに渡す．サーバが k 人以上のクライアントの訪問を受けたとき，またその時に限って，サーバは $f(0)$ を計算することができるので，これが *proof* となる．

多数のサーバと多くの time frame に適用可能にするためには，audit agency は $GF(p)$ 上のランダムな 2 変数多項式 $P(x,y)$ を選ぶ．ただし，P は x に関しては $k-1$ 次，y に関しては $d-1$ 次とする．クライアント $\mathcal{C}_i$ には $P(i,y)$ を渡す．クライアント $\mathcal{C}_i$ が time frame t にサーバ $\mathcal{S}_j$ を訪れるときには，$P(i,j\circ t)$ を $\mathcal{S}_j$ に渡す．$\mathcal{S}_j$ が time frame t 内に k 以上のシェアを受け取ったならば，$P(0,j\circ t)$ を計算することができるので，これを *proof* として audit agency に渡す．

この方式は，あらかじめ定められた数の time frame のみに適用可能であるため，one-time use であると考えることができる．

3.2 無条件に安全な方式 [2, Sec.3.3]

Naor と Pinkas は，不正なクライアントと不正なサーバに対して安全とするために，前節の基本方式を基にして，以下のような無条件に安全な方式を提案している．この方式も，前節と同様，one-time use である．

初期化: audit agency $\mathcal{A}$ は $GF(p)$ 上のランダムな多項式 $P(x,y)$, $A(x,y)$，$B(y)$ を選ぶ．ただし，P は x に関しては $k-1$ 次，y に関しては $d-1$ 次とし，A は x に関しては c_k 次，y に関しては c_d 次とする．また，B は c_d 次とする．$\mathcal{A}$ は

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

を計算し，$\mathcal{C}_i$ に $(V(i,y),P(i,y))$ を，$\mathcal{S}_j$ に $(A(x,j\circ 1),\ldots,A(x,j\circ T),B(j\circ 1),\ldots,B(j\circ T))$ を送る．ただし，$\circ$ は 2 つのビット列の連結を表すとする．

$\mathcal{C}_i$ と $\mathcal{S}_j$ のやり取り: time frame t 内にサーバ $\mathcal{S}_j$ からサービスを受けたいとき，クライアント $\mathcal{C}_i$ は

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

を $\mathcal{S}_j$ に送る．$\mathcal{S}_j$ は

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

をチェックする．正しければ $\mathcal{C}_i$ にサービスを提供し，そうでなければ受けつけない．

***proof* の作成:** time frame t 内に，$\mathcal{S}_j$ が k 人以上のクライアントの訪問を受けたならば，$P(0,j\circ t)$ を $P(i,j\circ t)$ から計算し，これを *proof* として audit agency に提出する．$\mathcal{A}$ は $P(0,j\circ t)$ が正しいことを確認する．

3.3 計算量理論的に安全な方式 [2, Sec.3.5]

Naor と Pinkas は計算量理論的に安全な方式も提案している．これは，Diffie-Hellman 問題の困難さに基づいている．

Z_p^* は p を法とする巡回群，g は位数 q を持つ部分群 Z_p^* の生成元であるとする．ただし q は素数．

初期化: Sec.3.2 と同様に，クライアント $\mathcal{C}_i$ は $P(i,y)$ と $V(i,y)$ を，サーバ $\mathcal{S}_j$ は $A(x,j)$ と $B(j)$ を $\mathcal{A}$ から受け取る．

各 time frame の始めの処理: サーバは $\mathcal{A}$ から $h=g^r$ を受けとる．ただし r は $\mathcal{A}$ が選んだ乱数である．

$\mathcal{C}_i$ と $\mathcal{S}_j$ のやり取り: $\mathcal{C}_i$ はサーバ $\mathcal{S}_j$ から h を受取り，

$$(h^{P(i,j)}, h^{V(i,j)})$$

を計算して $\mathcal{S}_j$ に送る．$\mathcal{S}_j$ は次式をチェックする．

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

***proof* の作成:** $\mathcal{S}_j$ が k 以上の訪問を受けていれば，$h^{P(0,j)}$ を計算できる．$h^{P(0,j)}(=g^{rP(0,j)})$ が *proof* となる．

4 Naor と Pinkas の metering scheme に対する攻撃

この章では，Naor と Pinkas の 2 つの方式に対する攻撃を示す．我々の攻撃では，特別なシェアを持つ 1 人のクライアントと，もう一人の任意のクライアントが結託することによって，サーバをだますことができる．言い替えれば，彼らの方式は，サーバに関する安全性が確保されていない．ここでは，無条件に安全な方式に対して攻撃方法を示すが，計算量理論的に安全な方式に対しても，全く同様に攻撃することができる．

あるサーバ $\mathcal{S}_j$ とある time frame t に対して，次の式を満たす 2 人のクライアント $\mathcal{C}_{i_0}$,$\mathcal{C}_{i_1}$ が存在したとする．

$$\begin{aligned} P(i_0, j\circ t) &= 0 \\ P(i_1, j\circ t) &\neq 0. \end{aligned}$$

すると，式 (1) より，2 人は $B(j\circ t)$ と $A(i_1, j\circ t)$ を次のように求めることができる．

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

次に，2 人は $(\tilde{P}, \tilde{V})$ を，

$$\begin{aligned} \tilde{P} &\neq P(i_1, j\circ t) \\ \tilde{V} &= A(i_1, j\circ t)\ \tilde{P} + B(j\circ t) \end{aligned}$$

を満たすようにランダムに作る．$\mathcal{C}_{i_1}$ が time frame t 内にサーバ $\mathcal{S}_j$ を訪れ $(\tilde{P}, \tilde{V})$ を渡せば，この偽のシェアはチェック式 (1) を必ず満たすのでサービスを受けることができる．

しかし，この time frame t 中に $\mathcal{S}_j$ が k 以上の訪問を受けたとしても，正しい $P(0, j\circ t)$ を得ることができない．

5 無条件に安全な提案方式

この章では，サーバに関する安全性と audit agency に関する安全性を兼ね備えた無条件に安全な metering scheme を提案する．ここでは，T 個の time frame があるとする．提案方式は one-time use であり，情報量は T に比例する．

5.1 プロトコル

初期化: audit agency $\mathcal{A}$ は $GF(p)$ 上の多項式 $F(x, y, z)$ を選ぶ．ただし，次数は x, y, z に対してそれぞれ $1, d-1, k-1$ とする．F を鍵多項式と呼ぶ．$\mathcal{A}$ は各サーバ $\mathcal{S}_j$ に対して乱数 $r_j \in Z_p \setminus \{0\}$ を選ぶ．

$$\begin{aligned} P_i(x, y) &\triangleq F(x, y, i) \\ A_j^t(z) &\triangleq F(r_j, j\circ t, z). \end{aligned}$$

とし，$\mathcal{C}_i$ に $P_i(x, y)$ を，$\mathcal{S}_j$ に $(A_j^1(z), \ldots, A_j^T(z), r_j)$ を渡す．

$\mathcal{C}_i$ と $\mathcal{S}_j$ のやり取り: time frame t 内に $\mathcal{S}_j$ を訪れる $\mathcal{C}_i$ は，

$$P_i(x, j\circ t)$$

を $\mathcal{S}_j$ に送る．$\mathcal{S}_j$ は次式をチェックし，$\mathcal{C}_i$ にサービスを提供するか否かを決定する．

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

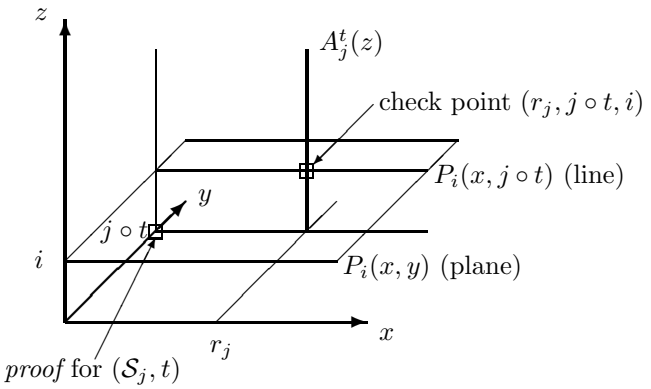

図 1: 提案方式における各情報の関係

***proof* の作成:** もし $\mathcal{C}_i$ が time frame t 内に $\mathcal{S}_j$ を訪れたとすると，$\mathcal{S}_j$ は

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

を知っている．したがって，k 人以上の訪問を受けたサーバ $\mathcal{S}_j$ は $F(0, j\circ t, 0)$ を計算できる. これが *proof* となる．

$\mathcal{A}$ は F を知っているので $F(0, j\circ t, 0)$ が正しいことチェックできる．

図 1 に各情報の関係を示す.

5.2 サーバに関する安全性

ここでは，無限大の計算能力を持つ敵に対してもサーバに関する安全性が守られることを示す．

Theorem 1 *time frame t* 内にサーバ $\mathcal{S}_j$ を k 人以上のクライアントが訪れたとする．このとき，全てのクライアントと $\mathcal{S}_j$ 以外の全てのサーバを買収する敵がいたとしても，$\mathcal{S}_j$ は *proof:*$F(0, j\circ t, 0)$ を $1 - 1/(p-1)$ 以上の確率で正しく計算することができる．

(Proof) $\deg P_j(x, j\circ t) = 1$ であることに注意．敵が $\mathcal{S}_j$ を妨害するには，少なくとも 1 人のクライアント C_i が $\tilde{P}(x) \neq P_i(x, j\circ t)$ かつ $\tilde{P}(r_j) = P_i(r_j, j\circ t)$ である 1 次の（偽の）多項式 $\tilde{P}(x)$ を $\mathcal{S}_j$ に送らなければならない．r_j は $Z_p \setminus \{0\}$ から選ばれた乱数であり，2 つの 1 次関数:$\tilde{P}(x)$ と $P_i(x, j\circ t)$ は高々 1 つの共通点しか持たない．よって，$\tilde{P}(r_j) = P_i(r_j, j\circ t)$ が成り立つ確率は $1/(p-1)$ である．これは，敵が全てのクライアントと $\mathcal{S}_j$ 以外の全てのサーバの情報を知り得たとしても，変わらない．
Q.E.D.

5.3 audit agency に関する安全性

次に，無限の能力を持つ敵に対して，audit agency に関する安全性が保たれていることを示す．

Theorem 2 ある *time frame* t に $\mathcal{S}_j$ を訪れたクライアントが k 未満であったとする．このとき，高々 d/T のサーバと $k-1$ 人のクライアントを買収した敵は，$F(0, j\circ t, 0)$ を $1/p$ より大きい確率で計算することはできない．

(Proof) $\alpha_s \stackrel{\triangle}{=} d/T$ とする．一般性を失うことなく，敵は α_s 個のサーバ：$\mathcal{S}_1, \ldots, \mathcal{S}_{\alpha_s}$ と，$k-1$ 人のクライアント：$\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$ を買収しているとする．敵は最後の time frame T におけるサーバ $\mathcal{S}_{\alpha_s}$ のための *proof*:$\tilde{F}(0, \alpha_s \circ T, 0)$ を偽造しようとする．

任意の $j \le \alpha_s$ と任意の $t \le T$ (ただし，$j\circ t \neq \alpha_s \circ T$) に対して，$\mathcal{S}_j$ は k 人以上の訪問を受けているとしよう．すると，敵の知ることのできる情報は，(a) 買収したクライアントが初期化時にもらった情報，(b) 買収したサーバが初期化時にもらった情報，(c) 買収したサーバが，クライアントからもらった情報である．(a) と (b) は，

(a) $F(x, y, 1), \ldots, F(x, y, k-1)$,

(b) $r_1, \ldots, r_{\alpha_s}$

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

であり，(c) は最大で，全ての $i \ge k$ と，$j \circ t = \alpha_s \circ T$ を除く全ての $j \circ t$ に対する

$$F(x, j \circ t, i)$$

である．

proof の推測値を β とする．任意の β に対して，(a), (b), (c) の情報を満足し，$\tilde{F}(0, \alpha_s \circ T, 0) = \beta$ を満たす唯一の鍵多項式 $\tilde{F}(x, y, z)$ が存在することを以下のように示すことができる．これは，β が正しい *proof* である確率が $1/p$ であることを表している．

さて，$F(x, y, z)$ は x に関しては 1 次関数である．まず，(b) から得られる情報 $F(r_{\alpha_s}, \alpha_s \circ T, 0)$ を用いて，$\tilde{F}(0, \alpha_s \circ T, 0) = \beta$ と $F(r_{\alpha_s}, \alpha_s \circ T, 0)$ を内挿する 1 次式 $L_{\alpha_s \circ T}(x)$ を 1 つ決めることができる．

さらに，(c) から，全ての $j \circ t (\neq \alpha_s \circ T)$ $(j \le \alpha_s)$ に対する $F(0, j \circ t, 0)$ を得ることができる．そこで，(b) から得られる $F(r_j, j \circ t, 0)$ を用いて，$F(0, j \circ t, 0)$ と $F(r_j, j \circ t, 0)$ を内挿する 1 次関数 $L_{j\circ t}(x)$ を 1 つ決めることができる．

ここで，$d = \alpha_s T$ 個の 1 次関数：

$$\{L_{j\circ t}(x) \mid j \le \alpha_s, t \le T\}$$

を得た．

次に $F(x, y, z)$ は y に関しては $d-1$ 次であることに注意．全ての $j \le \alpha_s$ と $t \le T$ に対して $B(x, j\circ t) = L_{j\circ t}(x)$ が成り立ち，x に関しては 1 次，y に関しては $d-1$ であるような $B(x, y)$ を 1 つ決めることができる．

最後に，x に関しては 1 次，y に関しては $d-1$，z に関しては $k-1$ 次であるような $\tilde{F}(x, y, z)$ を，(a) から得られる $F(x, y, i)$ を用いて，$\tilde{F}(x, y, 0) = B(x, y)$ と $\tilde{F}(x, y, i) = F(x, y, i)$ $(1 \le i \le k-1)$ が成り立つように 1 つ決めることができる．

ここで，$\tilde{F}(x, y, z) = F(x, y, z)$ が，(a), (b), (c) の全ての点を通ることを示さなくてはならない．(a) に関しては自明である．(b) について示そう．任意に $j \le \alpha_s$ と $t \le T$ を固定する．すると，上記の $\tilde{F}$ の構成方法より，$0 \le i \le k-1$ に対して $\tilde{F}(r_j, j \circ t, i) = F(r_j, j \circ t, i)$ である．従って，

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

最後に，(c) について示す．式 (2) と同様に，

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

が得られる．式 (3) から $\tilde{F}(0, j \circ t, i) = F(0, j \circ t, i)$ であり，式 (2) から $\tilde{F}(r_j, j \circ t, i) = F(r_j, j \circ t, i)$ である，したがって，$\tilde{F}(x, j \circ t, i) = F(x, j \circ t, i)$ が成り立つ．
Q.E.D.

6 計算量理論的に安全な提案方式

この章では，計算量理論的に安全な方式を提案する．この方式は，Diffie-Hellman 問題の難しさに基づいている．

6.1 プロトコル

Z_p^* は p を法とする巡回群，g は位数 q を持つ部分群 Z_p^* の生成元であるとする．ただし q は素数．

初期化: Sec.5.1 と同様に，$\mathcal{A}$ は $GF(q)$ 上の鍵多項式 $F(x, y, z)$ を選ぶ．ただし，F は x, y, z に関してそれぞれ次数 $1, d-1, k-1$ を持つ．また，サーバ $\mathcal{S}_j$ のための乱数 $r_j \in Z_q \setminus \{0\}$ を選ぶ，

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

を計算し，$\mathcal{C}_i$ には $P_i(x, y)$ を，$\mathcal{S}_j$ には $(A_j(z), r_j)$ を送る．

各 time frame の始めの処理: time frame t の始めに，$\mathcal{A}$ は u_t をランダムに選び，チャレンジ $h_t = g^{u_t} \bmod p$ をサーバに送る．

$\mathcal{S}_j$ と $\mathcal{C}_i$ のやり取り: $\mathcal{C}_i$ が time frame t 内に $\mathcal{S}_j$ からサービスを受ける場合，

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

を計算し，$(d_{i,j}^t, e_{i,j}^t)$ を $\mathcal{S}_j$ に送る．$\mathcal{S}_j$ は

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

が成り立っていれば $(d_{i,j}^t, e_{i,j}^t)$ を受理する．

proof の作成: $\mathcal{C}_i$ が $\mathcal{S}_j$ を time frame t 内に訪れたとき，$\mathcal{S}_j$ は

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

を知る．したがって，$\mathcal{S}_j$ が k 以上の訪問を受けたときには Lagrange の公式を用いて $h_t^{F(0,j,0)}$ を計算できる．$h_t^{F(0,j,0)}$ が *proof* となる．

$\mathcal{A}$ は $h_t^{F(0,j,0)}$ が正しいかチェックする．

6.2 サーバに関する安全性

提案方式では，サーバに関する安全性は，無限大の能力を持つ敵に対しても成り立つ．

Theorem 3 *time frame* t 内にサーバ $\mathcal{S}_j$ を k 人以上のクライアントが訪れたとする．このとき，全てのクライアントと $\mathcal{S}_j$ 以外の全てのサーバからなる敵がいたとしても，$\mathcal{S}_j$ は *proof*:$g^{u_t F(0,j,0)}$ を $1-1/(q-1)$ 以上の確率で正しく計算することができる．

証明は Theorem 1 の証明と同様なので省略する．

6.3 audit agency に関する安全性

この節では，d 個のサーバと $k-1$ 人のクライアントを買収する確率的で多項式時間の能力を持つ敵を考える．

Theorem 4 $\mathcal{S}_j$ がある *time frame* に k 人未満の訪問しか受けていないにもかかわらず，*proof*:$h_t^{F(0,j,0)}$ を無視できない確率で計算することのできる敵 M_0 があったとする．すると，*Diffie-Hellman* 問題を無視できない確率で解くことのできる確率的多項式時間チューリング機械 M_1 が存在する．

(Proof) 一般性を失うことなく，M_0 は d 個のサーバ：$\mathcal{S}_1, \ldots, \mathcal{S}_d$ と $k-1$ 人のクライアント：$\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$ を買収し，サーバ $\mathcal{S}_d$ の time frame T における *proof*:$h_T^{F(0,d,0)}$ を無視できない確率で計算できる．

M_1 の入力を $p, g, X(= g^\alpha \bmod p), Y(= g^\beta \bmod p)$ とする．以下で，M_1 は敵 M_0 の見ることのできる全ての情報を，$\alpha = u_T$ かつ $\beta = F(0,d,0)$ であるように作ることができることを示す．したがって，M_0 をサブルーチンとして用いることにより，M_1 は

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

を無視できない確率で得ることができる．これは，M_1 が Diffie-Hellman 問題を解けることを意味する．

さて，敵 M_0 の見ることのできる全ての情報は

(a) 買収したクライアントが初期化時に受けとる情報: $P_1(x,y), \ldots, P_{k-1}(x,y)$,

(b) 買収したサーバが初期化時に受けとる情報: $r_1, \ldots, r_d$ and $A_1(z), \ldots, A_d(z)$,

(c) チャレンジ：$h_1, \ldots, h_T$,

(d) 買収したサーバがクライアントから受けとる情報．これは最大で，全ての $i \geq k$ と $(j,t) = (d,T)$ を除く全ての (j,t) に対する

$$(d_{i,j}^t, e_{i,j}^t)$$

である．

ここで，M_1 は (a),(b),(c),(d) を次のように作る．

(a) $P_1(x,y), \ldots, P_{k-1}(x,y)$ をランダムに選ぶ．

(b) $r_1, \ldots, r_d$ と $A_1(0), \ldots, A_d(0)$ をランダムに選ぶ．

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

であるから，これらから $A_1(z), \ldots, A_d(z)$ を 1 つ決定できる．($\deg A_j(z) = k-1$ に注意．)

(c) $u_1, \ldots, u_{T-1}$ をランダムに選び，$h_i = g^{u_i}$ $(1 \leq i \leq T-1)$ とする．$h_T = X$ とする．

最後に，M_1 に (d) の値を以下のように決める．M_1 は

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

を持っている．Theorem 2 の証明とほぼ同様にして，M_1 は $(a_{i,j}, b_{i,j})$ $(j < d)$ と $(g^{a_{i,d}}, g^{b_{i,d}})$ を計算できる．ただし，

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

したがって，

1. $j = d$ と $t \leq T-1$ に対して，M_1 は $(g^{a_{i,d}}, g^{b_{i,d}})$ と u_t から $(d_{i,d}^t, e_{i,d}^t)$ を計算できる．
2. $j < d$ と $t = T$ に対して，M_1 は $(a_{i,j}, b_{i,j})$ は $X = g^\alpha = g^{u_T}$ から $(d_{i,j}^T, e_{i,j}^T)$ を計算できる．
3. $j < d$ と $t \leq T-1$ に対して，M_1 は $(a_{i,j}, b_{i,j})$ と u_t から $(d_{i,j}^t, e_{i,j}^t)$ を計算できる．

Q.E.D.

7 Robust な秘密分散共有法

7.1 従来の方式

(k,n) しきい値法では，ディーラは秘密 s を持ち，k 人以上の参加者は s を復元でき，k 人未満の参加者には s に関する情報を漏らさないように n 人の参加者 $(\mathcal{P}_1, \ldots, \mathcal{P}_n)$

に s を分散させる．ここでは，$\mathcal{P}_i$ によって保持されるシェアを v_i で表す．

秘密再構築時に，honest な $\mathcal{P}_i$ は v_i を公開する．しかし，$\mathcal{P}_i$ が cheater である場合は v_i を偽造するかもしれない．cheater を圧倒的確率で検出・特定することのできるしきい値法は，robust であると言われる．

S を秘密を表す集合とし，V_i を $\mathcal{P}_i$ のシェアとして取りうるものの集合とする．Rabin と Ben-Or [3] は，次式のシェアサイズを持つ robust (k,n) しきい値法 (RB scheme) を示した．

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

また，Carpentieri [1] は次式のシェアサイズを持つ robust (k,n) しきい値法を示した．

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

これらの方式では，$n-1$ 人の cheater が $1/|S|$ より大きい確率で honest な参加者をだますことができない．.

7.2 提案する robust しきい値法

ここでは，先に提案した metering scheme を応用した新しい robust (k,n) しきい値法を提案する．この方式では，$|V_i|$ は従来方式よりずっと小さく，cheater の検出能力は若干劣っている．

提案方式では

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

であり，$k-1$ 人の cheater は $(k-1)/(|S|-1)$ より大きい確率で honest な参加者をだますことができない．我々は，cheater は $n-1$ 人ではなく，高々$k-1$ 人しかいないと仮定している．しかし，任意の k 人の参加者 (cheater) は秘密を復元することができるため，この仮定は妥当なものである．

以下では，p を素数とし，$S \triangleq Z_p$ とする．

Distribution phase: 秘密 $s \in Z_p$ に対し，ディーラは $GF(p)$ 上の 2 変数多項式：

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

をランダムに選ぶ (ただし $F(0,0) = a_{00} = s$)．また，n 個の乱数：$r_i \in Z_p \setminus \{0\}$ を選ぶ．

$$\begin{aligned} B_i(x) &\triangleq F(x,i) \\ A_i(y) &\triangleq F(r_i,y), \end{aligned}$$

をとし，$v_i = (B_i(x), r_i, A_i(y))$ を各参加者 $\mathcal{P}_i$ に渡す.

Reconstruction phase: それぞれの参加者 $\mathcal{P}_i$ は $B_i(x)$ を公開する．それぞれの参加者 $\mathcal{P}_j$ は，自分以外の参加者 $\mathcal{P}_i$ に対し

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

をチェックし，成り立てば $B_i(x)$ を正しいとする．k 以上の正しいシェア： $B_i(x)(= F(x,i))$ からは，$s = F(0,0)$ が一意に定まる．

Theorem 5 $k-1$ 人の *cheater* は，$(k-1)/(p-1)$ より大きい確率で他の参加者をだますことはできない．

(Proof) $k-1$ 人の参加者：$\mathcal{P}_1, \ldots, \mathcal{P}_{k-1}$ が結託して $\mathcal{P}_k$ をだますことを考える．$\mathcal{P}_1$ は $\tilde{B}(x)(\neq B_1(x))$ を公開するとしよう．$\mathcal{P}_1$ は，$\tilde{B}(r_k) = A_k(1)$ であれば $\mathcal{P}_k$ をだますことに成功する．$\tilde{B}(x)$ は，$k-1$ 次以下の多項式 $\Delta B(x)(\neq 0)$ によって $\tilde{B}(x) = B_1(x) + \Delta B(x)$ と書くことができるので，この不正の成功確率は

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

となる． Q.E.D.

参考文献

[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