コンピュータセキュリティ　18－34
(2002. 7. 19)

情報量的に安全な地域内公開検証可能秘密分散法

鬼頭 大介† 大塚 玲† 今井 秀樹†

† 東京大学生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1
E-mail: {kitou, otsuka}@imailab.iis.u-tokyo.ac.jp, imai@iis.u-tokyo.ac.jp

あらまし 誰でも秘密情報が正しく分散されたことを確認できる方法として，公開検証可能秘密分散法（PVSS）がある．従来の PVSS は，公開鍵暗号をベースに作られているため，その安全性は計算量的なものとなる．本稿では，PVSS の要件を弱めた地域内 PVSS という概念を導入する．また，初期化時のみ存在する Trusted Initializer（TI）を仮定し，情報量的安全性を満たす地域内 PVSS を提案する．

キーワード 公開検証可能秘密分散法，情報量的安全性

Locally PVSS Based on Unconditional Security

Daisuke KITO† Akira OTSUKA† and Hideki IMAI†

† Institute of Industrial Science, University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan
E-mail: {kitou, otsuka}@imailab.iis.u-tokyo.ac.jp, imai@iis.u-tokyo.ac.jp

Abstract Publicly verifiable secret sharing(PVSS) allows everybody, not only the participants, to verify that the shares are correctly distributed. Former PVSS is based on public-key encryption. So the security of secret is computational. In this paper we introduce new notion "locally PVSS" which meets weak condition of PVSS. We use trusted authority called "trusted initializer" that exists only in the initialization protocol of the scheme and propose locally PVSS based on unconditional security.

Keyword publicly verifiable secret sharing, unconditional security

1. はじめに

秘密情報を保管する方法の一つとして，秘密分散法(SSS)がある．これは，秘密情報を分散して各機関に預け管理する方法である．本稿では，以下のように言葉を定義する．

ディーラ：秘密を分散する人
シェア：分散された情報
メンバ：シェアを受け取る人

SSSでは，ディーラが正しくシェアを作らなければ，閾値個以上のシェアを用いても秘密情報を正しく復元できない．そのため，各メンバは，受け取ったシェアが正しく作られたものであるかを確認できるほうが望ましい．また，秘密情報の復元時に，メンバが，ディーラから配られたシェアと異なるものを提示してしまえば，本来の秘密情報が復元できなくなってしまう．これらのディーラ及びメンバの不正を防いで，正しく秘密を分散，復元する手法として考案されたものが検証可能秘密分散法（VSS）である．VSSの安全性には，秘密の安全性とシェア検証の安全性があり，それぞれ，計算量的安全性に基づくものと情報量的安全性に基づくものがある．計算量的安全性とは，安全性に計算量的な想定を必要とする．これに対し，情報量的安全性とは，如何なる計算量的な想定も必要としない．

VSSに公開検証可能性を付加したものを公開検証可能秘密分散法（PVSS）と言う．従来のPVSSは公開鍵暗号をベースにしているため，秘密の安全性は計算量的なものとなる．本稿では，PVSSの要件を弱めた地域内PVSSという概念を導入する．また，初期化時のみ存在するTrusted Initializer（TI）を仮定し，情報量的安全性を満たす地域内PVSSを提案する．

2. 検証可能秘密分散法

2.1. 関連研究

最初のVSSは，秘密分散とゼロ知識対話証明を用いて構成された[1]．また，[2],[3],[4]では情報量的に安全な方式が提案されている．

しかしこれらの方式はインタラクティブであり，シェアの検証にディーラとシェア管理者間で多くの通信のやりとりが必要となる．これに対し，[5],[6]では，剰余指数関数などの準同型性を持った暗号を用いた，非常に効率的なノンインタラクティブなVSSの構成法が提案されている．これらの方式では，ディーラはシェアとその検証用情報をシェア管理者に送るだけでよく，管理者はシェアの検証にディーラとの対話を必要としない．しかしこれらの方式は完全に情報量的安全性に基づいているわけではなく，一部計算量的安全性に基づいている．これに対して，[7]は二つの安全性ともに情報量的安全性を満たしており，ノンインタラクティブである．ただし初期化時のみ存在するTrusted Initializer(TI)を必要とする．PVSSに関しては，[8]で最初にPVSSが導入される．以降[9]や[10]等で効率改善やassumptionを弱める事が成されている．今回提案する方式はPVSSの要件を弱め，[7]をベースにした情報量的に安全な方式である．

2.2. VSSの安全性

VSSの安全性は秘密自体の安全性と，シェア検証の安全性との二面から推し量ることができる．

2.2.1. 秘密の安全性

これはディーラが保持する秘密Sの安全性である．つまり，シェア管理者が結託し合わない限り，秘密Sは漏れないものであるかどうかという事である．

VSSでは，ディーラは秘密Sに関してコミットメントを生成する．秘密の安全性は，このコミットメントの仕方により，計算量的安全か，情報量的安全か決まる．

2.2.2. シェア検証の安全性

これは，ディーラから送られてきたシェアの安全性である．つまり，ディーラがメンバに対して本当に正しいシェアを送ってきたかどうかという事である．シェア検証の安全性も，コミットメントの仕方により，計算量的安全か，情報量的安全か決まる．

2.3. PVSS

2.3.1. PVSSの特性

PVSSとは，以下の特性を持つVSSである．

特性：シェアを持たない人でも誰でも，他人のシェアの正当性を検証可能である．

VSSでは，自分のシェアのみ検証可能であったが，上記特性により，他人のシェアについても検証可能となる．

2.3.2. PVSSプロトコル（[8]参照）

各変数：

g：離散対数導出が困難なgenerator
s：secret
s_i：各メンバのシェア
z_i：〃　秘密鍵
y_i：〃　公開鍵　$(y_i = h^z)$

コミットメント（公開値）：$S_i = g^{s_i}$, $S = g^s$

プロトコル：

ディーラ：ランダムに α を選択．cipher-text として以下を公開．

$$(A, B) = (h^{\alpha}, s_i^{-1} y^{\alpha})$$

メンバ：公開された cipher-text から自分の秘密鍵を用いてシェアを復号する．

$$s_i = A^z / B$$

メンバ以外の人：(A, B)は s_i を encrypt したものである事を検証する(Pubverify)．図 1 に示す手順による．

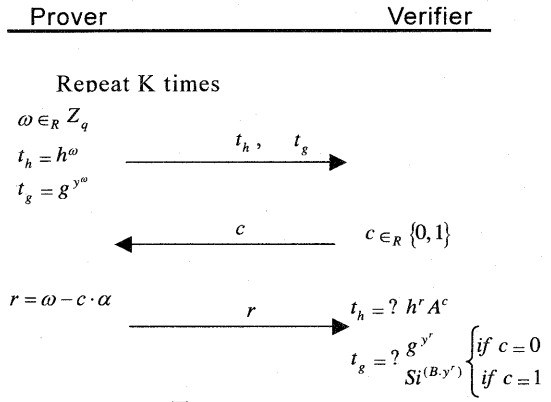

図 1　Pubverfy

上記 Pubverify を non-interactive にする時は，上記の c をハッシュ関数 H を用いて，

$$c = H(S_i \| A \| B \| t_{h_1} \| t_{g_1} \| \cdots \| t_{h_K} \| t_{g_K})$$

とする．

2.3.3. PVSS の安全性

PVSS の安全性に関しては，秘密の安全性は計算量的安全なものとなる．これは，秘密 s に対するコミットメント(Commit(s))として，ディーラが

$$Commit(s) = g^s$$

を公開するためである．また，シェア検証の安全性に関しては情報量的に安全なものとなる．これはシェア s_i に対するコミットメント

$$Commit(s_i) = g^{s_i}$$

に対して零知識対話証明を行っているからである．尚，non-interactive にした場合には計算量的に安全なものとなる．

3. 情報量的に安全な地域内 PVSS

従来の PVSS では公開鍵暗号をベースにしている．そのため，秘密の安全性は計算量的になる．ここでは，[7]をもとに，ある地域内では情報量的に安全な PVSS を提案する．

3.1. 地域内 PVSS

一般の PVSS では，シェアを持たない人でも誰でも，他人のシェアの正当性を検証可能であるという条件を満たす．ここではその条件を弱め，初めに TI に登録をした人（地域内）であれば誰でも他人のシェアの正当性を検証可能であるという条件を導入する．以下ではこの条件を満たす PVSS を地域内 PVSS と呼ぶ．

3.2. 定義

関数及び記号：以下の四変数関数及び，記号を定義する．

$$F(x,\bar{y},\bar{z},\vec{\omega}) = \sum_{i=0}^{n-1}\sum_{j=0}^{\omega}\sum_{k=0}^{m-1}\sum_{l=0}^{\psi} a_{ijkl} x^i y_j z_k \omega_l$$

$$F'(x,\bar{y},\bar{z},\vec{\omega}) = \sum_{i=0}^{n-1}\sum_{j=0}^{\omega}\sum_{k=0}^{m-1}\sum_{l=0}^{\psi} a'_{ijkl} x^i y_j z_k \omega_l$$

$$i^{(m)} = (1, i, i^2, \ldots, i^{m-1})$$

$$i^{(m)} \setminus t = (1, i, i^2, \ldots, i^{t-1}, 0, \ldots, 0)$$

TI：初期化時のみ存在する信頼機関．シェア生成関数，シェア検証関数，マスク関数のもととなる関数を作成する．

ディーラ：TI が生成したシェア生成関数から，秘密及びシェアを生成する．

ユーザ：シェア受取予定者．ユーザ数は n とする．

登録者：ユーザ，ディーラを含め，最初に TI にディーラの公開情報を検証するために登録した人達．

シェア生成関数：　$F(\delta,\bar{y},\bar{z},\vec{\omega})$

シェア検証関数：　$G_{jk}(x,\bar{z},\vec{\omega})$

$$G_{jk}(x,\bar{z},\vec{\omega}) = F(x,\bar{v}_{pub\_k},\bar{z},\vec{\omega}) + F'(U_j,\bar{v}_{pub\_k},\bar{z},\vec{\omega}),\ (j=1,\ldots,n)$$

シェア検証ベクトル：　$\bar{v}_{pub\_k}$

マスク関数：ユーザ i に送るマスク関数は

$$F'(U_i,\bar{y},\bar{z},\vec{\omega})$$

3.3. 地域内 PVSS 概要

1．初期設定

TI→ディーラ:シェア生成関数,マスク情報を送る.

TI→ユーザ：ディーラが公開するシェアのマスクをとる情報を送る.

TI→登録者：シェア検証用情報を送る.

2．地域内 PVSS 実行

ディーラ：全ての人にマスクをかけたシェアを公開.

登録者：ディーラの公開したシェアの検証.

ユーザ：ディーラの公開情報が正しい時は，マスクをはずして真のシェアを得る.

図 2 に地域内 PVSS の概要を示す.

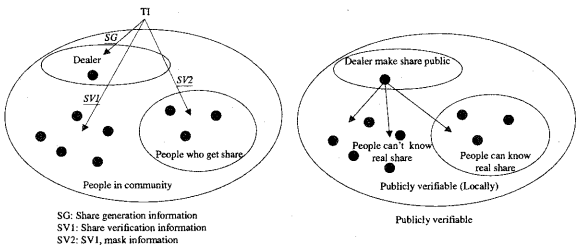

図 2　地域内 PVSS 概要

3.4. プロトコル詳細

プロトコルの詳細を図 3 に示す．具体的には以下の手順による.

1. TI がディーラ，ユーザに漏れない様に，シェア生成関数，シェア検証関数，マスク関数のもととなる以下の関数を生成する.

$$F(x,\bar{y},\bar{z},\vec{\omega})$$
$$F'(x,\bar{y},\bar{z},\vec{\omega})$$

2. TI は，ディーラにシェア生成関数 $F(\delta,\bar{y},\bar{z},\vec{\omega})$,δ を送る．また以下のマスク関数を送る.

$$F'(U_i,\bar{y},\bar{z},\vec{\omega}),(i=1,\cdots,n)$$

3. TI は，ユーザ i には，シェア検証関数 $F(x,\bar{v}_i,\bar{z},\vec{\omega})$ とシェア検証ベクトル $\bar{v}_i$ を送る．また以下のマスク関数を送る.

$$F'(U_i,\bar{y},\bar{z},\vec{\omega})$$

4. TI はユーザを含め，初めに登録した全ての人達 k に以下のシェア検証関数を送る.

$$G_{jk}(x,\bar{z},\vec{\omega}) = F(x,\bar{v}_{pub\_k},\bar{z},\vec{\omega}) + F'(U_j,\bar{v}_{pub\_k},\bar{z},\vec{\omega}),\ (j=1,\cdots,n)$$

5. ディーラは送られた関数 $F(\delta,\bar{y},\bar{z},\vec{\omega})$ をもとに，秘密 s を生成し，s に対してコミットメントを生成する．またシェアにマスクをかけて公開する.

$$s=F(\delta,1,1,\vec{\alpha}),$$
$$s_i(\bar{y})=F(\delta,\bar{y},i^{(m)}\backslash t,\vec{\alpha}) + F'(U_i,\bar{y},i^{(m)}\backslash t,\vec{\alpha}),$$
$$(i=1,\cdots,n),\ (\delta,\vec{\alpha})$$

6. 各ユーザおよび，ユーザ以外の登録者は，そのシェアが正しいかどうか，最初に TI から受け取った検証情報から検証する.

$$ver_k(s_i,\vec{\alpha}) = \begin{cases} 1 & if\ s_i(\bar{y})=G_{ik}(x,\bar{z},\vec{\omega})\Big|_{x=\delta,\bar{y}=\bar{v}_{pub\_k},\bar{z}=i^{(m)}\backslash t,\vec{\omega}=\vec{\alpha}} \\ 0 & otherwise \end{cases}$$

7. 公開されたシェアが正しい場合は，マスク関数を知るユーザだけが本当のシェアを知ることができる.

$$S_i(\bar{y})=s_i(\bar{y})-F'(U_i,\bar{y},i^{(m)}\backslash t,\vec{\alpha})$$

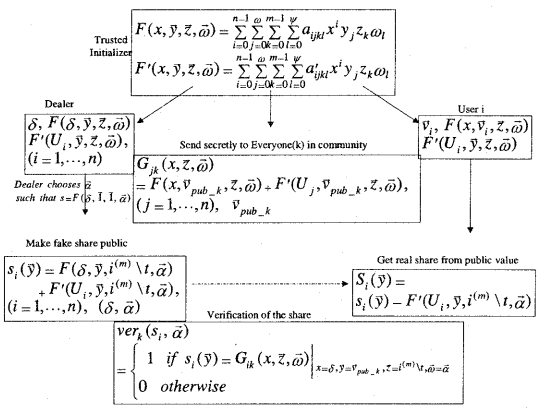

図 3　地域内 PVSS プロトコル

3.5. プロトコル安全性

3.5.1. 秘密の安全性

この方式では，マスク情報を知らない人は，ディーラの公開情報からシェアの検証はできるが，本当のシェアを知ることはできない．そのため，ディーラの公開情報に関して秘密の安全性は情報量的である．次に，シェア検証関数からの秘密の漏洩について述べる．シェア検証関数は，n+1 個の未知関数を含んでいる．しかし等式は n 個しか存在しないため，シェア検証関数を二つの関数に分離することはできない．そのため，シェア検証関数に関しても秘密の安全性は情報量的である．但し，ユーザに関しては，シェア検証関数を二つの関数に分離可能であるが，結局閾値 t 以上の正しいユーザが集まらなければ，秘密を復元できない．以上より，秘密の安全性は情報量的である．

3.5.2. シェア検証の安全性

ディーラは，コミュニティ内の人達がどの様なシェア検証関数およびシェア検証ベクトルをもっているのか知らない．そのためディーラが，ユーザに偽のシェアを公開して成功する確率は，ある一定値以下である．これより，シェア検証の安全性は情報量的になる．

4. まとめ

一般的な PVSS の要件を減らした地域内 PVSS という概念を導入し，情報量的安全性を満たす方式を提案した．

文　献

[1] B. Chor, "Verifiable secret sharing and achieving simultaneity in the presence of faults", FOCS, p383-395, 1985.

[2] M.Ben-Or, S.Goldwasser and A.Widgerson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation", Proc. of 20th ACM Symp. on Theory of Computing, p1-10, 1988.

[3] D. Chaum, C. Crepeau and I. Damgard, " Multiparty unconditionally secure protocols", Proc. of 20th Annual ACM Symp. on Theory of Computing, p11-19, 1988.

[4] Douglas R. Stinson and R. Wei, "Unconditionally Secure Proactive Secret Sharing Scheme with Combinatorial Structures", SAC '99 p200-214 2000.

[5] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing", Proc. of the 28th IEEE Symp. on the Foundations of Computer Science, p427-437, 1987.

[6] T. P. Pedersen, "Non Interactive and Information Theoretic Secure Verifiable Secret Sharing", Proc. of CRYPTO '91, p129-140, 1991.

[7] Akira Ohtsuka, Junji Shikata, H. Imai, "Discussions on Unconditionally Secure VSS", SCIS 2002.

[8] M. Stadler, "Publicly Verifiable Secret Sharing", EUROCRYPT '96, p190-199, 1996.

[9] Berry Schoenmakers, "A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting", CRYPTO '99, p148-164, 1999.

[10] A. Young, M. Yung, "A PVSS as Hard as Discrete Log and Shareholder Separability", PKC 2001, p287-299, 2001.