社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＣＳＥＣ－38 (50)
2007／7／20

双方向放送サービスのための効率的なStrong Key-Insulated署名

大竹 剛† 花岡悟一郎†† 小川 一人†

† 日本放送協会 〒157-8510 東京都世田谷区砧 1-10-11
†† 独立行政法人産業技術総合研究所 〒101-0021 東京都千代田区外神田 1-18-13
E-mail: †{ohtake.g-fw,ogawa.k-cm}@nhk.or.jp, ††hanaoka-goichiro@aist.go.jp

あらまし 双方向放送サービスにおいて，視聴者の個人情報を安全に送受信するため，放送局へのなりすましを防止するプロバイダ認証が必要である．しかし，放送局の署名鍵が漏洩した場合，第三者による放送局へのなりすましが可能となる．一方，プロバイダ認証として Key-Insulated 署名を双方向放送サービスに適用する場合，署名長や鍵長に関する効率性が強く要求される．本稿では，Abe, Okamoto によって提案された署名方式を応用し，署名鍵およびマスター鍵の漏洩に耐性を有する，効率的な Strong Key-Insulated 署名の構成法を提案する．また，提案方式は，離散対数問題 (DLP) が困難であるという仮定の下で，安全であることを示す．

キーワード 双方向放送サービス，プロバイダ認証，個人情報保護，鍵漏洩，Strong Key-Insulated 署名，Forking Lemma

Efficient Strong Key-Insulated Signature Scheme for Bidirectional Broadcasting Service

Go OHTAKE†, Goichiro HANAOKA††, and Kazuto OGAWA†

† Japan Broadcasting Corporation 1-10-11 Kinuta, Segataya-ku, Tokyo, 157-8510 Japan
†† National Institute of Advanced Industrial Science and Technology 1-18-13 Sotokanda, Chiyoda-ku, Tokyo, 101-0021 Japan
E-mail: †{ohtake.g-fw,ogawa.k-cm}@nhk.or.jp, ††hanaoka-goichiro@aist.go.jp

Abstract In the bidirectional broadcasting service via networks, provider authentication is necessary for secure transmission of user's personal information. If the broadcaster's signing key is leaked, the third party that gets the signing key can impersonate the broadcaster. We propose a strong key-insulated signature scheme which is based on the signature scheme proposed by Abe and Okamoto. Our scheme is more efficient strong key-insulated signature scheme than ever before, on signature size, key size, and computation cost. We show that our scheme is provably secure under the assumption of the hardness of discrete logarithm problem.

Key words bidirectional broadcasting service, provider authentication, personal information protection, key leakage, strong key-insulated signature, forking lemma

1. はじめに

デジタル放送の普及やネットワークの高速・大容量化に伴い，放送コンテンツをブロードバンドネットワークを用いて配信するサービスが実現しつつある．このような状況で，視聴者から放送局への上り回線を用いた双方向放送サービスの実現が期待されている．例えば，TV ショッピングサービスは，視聴者の氏名・住所・電話番号等の個人情報を放送局に送信することにより実現できる．しかし，ネットワーク経由で送受信する場合，放送局（プロバイダ）になりすました第三者に視聴者の個人情報が漏洩してしまう危険があり，電子署名を用いたプロバイダ認証が必要となる．

放送局が保有する署名鍵が漏洩した場合，漏洩鍵を取得した第三者が放送局になりすまして署名生成を行うことが可能となるため，鍵更新を行う必要がある．従来，通信ネットワークにおける認証方式として一般に PKI (Public Key Infrastracture) が用いられている．PKI における鍵更新では，検証鍵や証明書失効リスト (Certificate Revocation List: CRL) の伝送，証明書の正当性チェックが必要である．これらの処理は，通信ネットワーク上では問題なく動作するが，リアルタイム性を有す

る放送では負荷が大きい．また，放送局が検証鍵の入ったセキュリティカード（CAS カード）を視聴者に事前配布する方法も考えられる．この場合の鍵更新は，新しい検証鍵の入った CAS カードを全視聴者に再配布する必要があるため，利便性が悪い．そこで，我々は上記の問題点を解消し，署名鍵漏洩に耐性を有する安全なプロバイダ認証を実現するためのコンテンツ配信システムを提案した [1]．プロバイダ認証に適した署名方式として，放送局が検証鍵を変更せずに署名鍵の更新が可能な Key-Insulated 署名 [2] が有効であることを示した．また，Key-Insulated 署名を適用することにより生じる問題点を，運用により解決する方法を示した．しかし，Key-Insulated 署名を双方向放送サービスに適用する場合，署名長や鍵長に関する効率性が強く要求される．そこで，従来よりも署名長や鍵長が短い方式の実現が重要となる．

我々は，Abe, Okamoto によって提案された署名方式 [3] を応用し，署名鍵およびマスター鍵の漏洩に耐性を有する，効率的な Strong Key-Insulated 署名 [2] の構成法を提案した [4]．本稿では，我々が提案する Strong Key-Insulated 署名について述べるとともに，提案方式の安全性証明を行ったので報告する．

2. 双方向放送サービスモデル

想定するモデルを図 1 に示す．放送局側には，コンテンツを配信するコンテンツサーバ，および，視聴者の個人情報を管理する個人情報管理サーバがある．一方，視聴者側には受信端末 (デジタル放送受信機) があり，CAS カードが挿入されている．

まず，放送局のコンテンツサーバは，下り回線を通してコンテンツを視聴者に配信する．コンテンツは不正コピー防止のため暗号化される．視聴者は，CAS カードを用いて受信機内でコンテンツを復号する．次に，放送局の個人情報管理サーバは，視聴者に対し個人情報リクエストを送信する．視聴者の受信機はこのリクエストに従い，受信機内に蓄積されている個人情報を放送局に送信する．

リクエストに何の処理も施さず，誰でも自由にリクエストを送信できる場合，視聴者の個人情報が，放送局になりすました第三者に誤って送信され，個人情報が漏洩する危険性がある．このため，放送局へのなりすましを防止するプロバイダ認証機能が必要である．プロバイダ認証は電子署名の技術を用いることにより実現できる．すなわち，放送局が個人情報リクエストを送信する際に，電子署名を付加して送信する．視聴者の受信

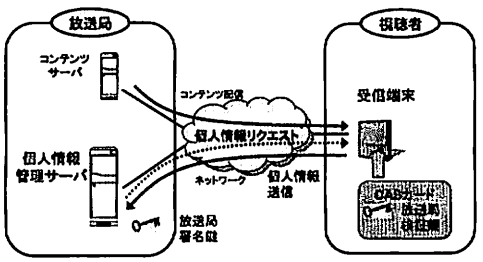

図 1　想定するコンテンツ配信モデル

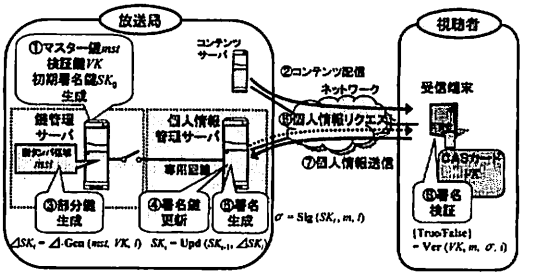

図 2　提案するコンテンツ配信システム

機は，リクエストに付加されている電子署名を，CAS カードに予め格納されている検証鍵を用いて検証し，正規の放送局からのリクエストであることを確認する．この手順により，視聴者が安心して個人情報を放送局に提供することが可能となる．

これに対し，我々は，放送局が署名鍵を定期的に更新することで，署名鍵漏洩の被害を最小化する方式を提案した [1]．図 2 にその概念を示す．ここでは，署名鍵の効率的な運用のために，Key-Insulated 署名 [2] を用いた．Key-Insulated 署名は，署名鍵更新の際に検証鍵の更新が不要であるため，CAS カードの再配布が不要となり，放送サービスのリアルタイム性を損なうこともない．さらに，マスター鍵を管理する「鍵管理サーバ」と，署名鍵を更新する「個人情報管理サーバ」の 2 つのサーバを用いて鍵更新処理を行う．各サーバは，異なる部屋で，個別の管理者により管理される．サーバ間の接続を最小限とし，鍵管理サーバを外部ネットワークに直接接続しないことで，サーバアタックによるマスター鍵の漏洩を防止する．

図 2 のシステムを用いて双方向放送サービスを実現する場合，多数の署名付きデータを送受信することになる．また，検証鍵を格納する CAS カードのメモリ領域は限られている．従って，署名長，鍵長，計算量において効率的な Key-Insulated 署名を構成する必要がある．

3. Key-Insulated 署名

本節では，Key-Insulated 署名のモデルと安全性の定義について述べる．

3.1 Key-Insulated 署名のモデル

Key-Insulated 署名のモデルを以下に示す．(t, N)-key-insulated とは，N 回の鍵更新があった場合に，攻撃者が最大 t 回分の鍵を取得してもその他の鍵に影響を与えず，署名を偽造することができないことを意味する．すなわち，最大 t 回の鍵漏洩を許容しても安全な署名方式である．$(N-1, N)$-key-insulated の場合，無条件の鍵漏洩を許容しても安全な署名方式となる．

定義 3.1. *Key-Insulated* 署名 Π は，下記に示す 5 つの多項式時間アルゴリズム (**Gen, Δ-Gen, Upd, Sig, Ver**) で構成される．

- **Gen: 鍵生成アルゴリズム**

セキュリティパラメータ 1^k, (最大) 鍵更新回数 N を入力とし，

マスター鍵 mst, 検証鍵 VK, 初期署名鍵 SK_0 を出力とする確率的アルゴリズム.

- **Δ-Gen: 部分鍵生成アルゴリズム**

マスター鍵 mst, 検証鍵 VK, 時刻 i を入力とし, 部分鍵 Δ-SK_i を出力とする確率的アルゴリズム.

- **Upd: 鍵更新アルゴリズム**

時刻 $i-1$ における署名鍵 SK_{i-1}, 時刻 i における部分鍵 Δ-SK_i を入力とし, 時刻 i における署名鍵 SK_i を出力とする確定的アルゴリズム.

- **Sig: 署名生成アルゴリズム**

メッセージ m, 時刻 i における署名鍵 SK_i を入力とし, 時刻 i におけるメッセージ m の署名 $\sigma_{i,m}$ を出力とする確率的アルゴリズム.

- **Ver: 署名検証アルゴリズム**

メッセージ m, 時刻 i におけるメッセージ m の署名 $\sigma_{i,m}$, 時刻 i, 検証鍵 VK を入力とし, *1 ("True")* または *0 ("False")* を出力とする確定的アルゴリズム.

Key-Insulated 署名では, 署名の検証において時刻情報を用いているため, 署名鍵を更新しても検証鍵は更新する必要がない.

3.2 安全性の定義

時刻を入力し対応する署名鍵を出力する鍵漏洩オラクルと, 時刻とメッセージを入力し対応する署名を出力する署名オラクルが攻撃者に与えられていると仮定する. このとき, 攻撃者が任意に選んだメッセージに対する署名を偽造し, この署名の検証に成功する確率が無視できるならば, 安全であると定義する. 具体的には, 以下のように定式化される.

攻撃者に対し, 時刻 i を入力し対応する署名鍵 SK_i を出力する鍵漏洩オラクル $\mathbf{Exp}_{mst,SK_0}(\cdot)$ へのアクセスを許可する. また, 攻撃者に対し, 時刻 i, メッセージ m を入力し対応する署名 $\mathbf{Sig}_{SK_i}(i,m)$ を出力する署名オラクル $\mathbf{Sig}_{mst,SK_0}(\cdot,\cdot)$ へのアクセスを許可する. このとき, Key-Insulated 署名の安全性を以下のように定義する.

定義 3.2. Π = **(Gen, Δ-Gen, Upd, Sig, Ver)** を *Key-Insulated* 署名とする. 任意の攻撃者 $\mathcal{A}$ に対し, 次式を定義する.

$$\mathbf{Succ}_{\mathcal{A},\Pi}(k) := Pr\Big[\ \mathbf{Ver}_{VK}(m^*,i^*,\sigma_{i^*,m^*})=1 \ \Big|\ (mst,VK,SK_0)\leftarrow \mathbf{Gen}(1^k,N),\ (m^*,i^*,\sigma_{i^*,m^*})\leftarrow \mathcal{A}^{\mathbf{Sig}_{mst,SK_0}(\cdot,\cdot),\mathbf{Exp}_{mst,SK_0}(\cdot)}(VK)\ \Big]$$

ここで, (i^*,m^*) は署名オラクルに与えず, i^* は鍵漏洩オラクルに与えないものとする. もし, 任意の $\mathcal{A}$ が鍵漏洩オラクルに最大 t 回のリクエストを行ったとき, $\mathbf{Succ}_{\mathcal{A},\Pi}(k)$ の値が無視できるならば, Π は (t,N)-*key-insulated* であるという. また, Π が $(N-1,N)$-*key-insulated* ならば, Π は *perfectly key-insulated* であるという.

さらに, Strong Key-Insulated 署名の安全性を以下のように定義する.

攻撃者に対し, 鍵漏洩オラクル $\mathbf{Exp1}(\cdot), \mathbf{Exp2}_{mst,SK_0}(\cdot)$ へのアクセスを許可する. ただし, $\mathbf{Exp1}(\cdot)$ は, マスター鍵要求メッセージを入力しマスター鍵を出力する鍵漏洩オラクル, $\mathbf{Exp2}_{mst,SK_0}(\cdot)$ は, 時刻 i を入力し対応する署名鍵 SK_i を出力する鍵漏洩オラクルであり, 攻撃者は適応的にどちらにアクセスするかを選択できる. また, 攻撃者に対し, 時刻 i, メッセージ m を入力し対応する署名 $\mathbf{Sig}_{SK_i}(i,m)$ を出力する署名オラクル $\mathbf{Sig}_{mst,SK_0}(\cdot,\cdot)$ へのアクセスを許可する.

定義 3.3. Π = **(Gen, Δ-Gen, Upd, Sig, Ver)** を *Key-Insulated* 署名とする. 任意の攻撃者 $\mathcal{B}$ に対し, 次式を定義する.

$$\mathbf{Succ}_{\mathcal{B},\Pi}(k) := Pr\Big[\ \mathbf{Ver}_{VK}(m^*,i^*,\sigma_{i^*,m^*})=1 \ \Big|\ (mst,VK,SK_0)\leftarrow \mathbf{Gen}(1^k,N),\ (m^*,i^*,\sigma_{i^*,m^*})\leftarrow \mathcal{B}^{\mathbf{Sig}_{mst,SK_0}(\cdot,\cdot),\mathbf{Exp1}(\cdot),\mathbf{Exp2}_{mst,SK_0}(\cdot)}(VK)\ \Big]$$

ここで, (i^*,m^*) は署名オラクルに与えず, i^* は鍵漏洩オラクルに与えないものとする. もし, 任意の $\mathcal{B}$ が鍵漏洩オラクルに最大 t 回のリクエストを行ったとき, $\mathbf{Succ}_{\mathcal{B},\Pi}(k)$ の値が無視できるならば, Π は *strong* (t,N)-*key-insulated* であるという.

4. 提案方式

本章では, 署名鍵およびマスター鍵の漏洩に耐性を有する, 効率的な Strong Key-Insulated 署名の構成法について提案する.

4.1 提案方式の設計方針

提案する Strong Key-Insulated 署名は, Abe-Okamoto 方式[3] を応用することで構成される. Abe-Okamoto 方式は, Schnorr 署名を改良し, 署名の権限を第三者に委譲するための Delegation Chain を構成する手法である. 従来の方式に比べて, 検証速度が速く, 通信量も少ない効率的な署名方式である. なお, Delegation Chain を構成する署名方式を応用することにより, Key-Insulated 署名を構成できることが知られている[5]. そこで, 本稿では, Abe-Okamoto 方式を改良することにより, 効率的な Strong Key-Insulated 署名を構成した. 提案する署名方式は, 署名鍵の漏洩回数の上限がなく, 鍵長・署名長が短く, 計算量が少ない方式であり, 2. 章のコンテンツ配信モデルに適した Strong Key-Insulated 署名方式である.

4.2 提案する Strong Key-insulated 署名

本稿で提案する, Strong Key-insulated 署名の構成法を示す.

- 鍵生成

素数 p,q $(q|p-1)$ と, 乗法群 Z_p^* の元 g を選ぶ. ここで, g は Z_p^* 上に構成される位数 q の部分群の生成元となるものとする. 次に, x,x' を Z_q からランダムに選び, マスター鍵 $x_0=x-x'$ をセキュアデバイス内に管理し, x' を署名者が管理する. そして, $y_0=g^{x_0} \bmod p$, $y'=g^{x'} \bmod p$ を求め, 検証鍵 $VK=\langle p,q,g,y_0,y',G(\cdot,\cdot),H(\cdot,\cdot,\cdot,\cdot)\rangle$ を公開する. ここで, G,H はハッシュ関数 $G: Z_p\times\{0,1\}^*\rightarrow Z_q$, $H: Z_p\times Z_p\times\{0,1\}^*\times\{0,1\}^*\rightarrow Z_q$ である.

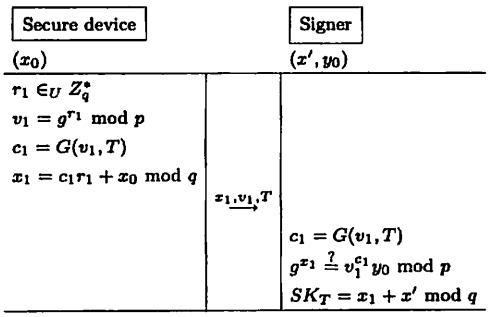

図 3 提案する Strong Key-Insulated 署名 (1) 部分鍵生成・鍵更新

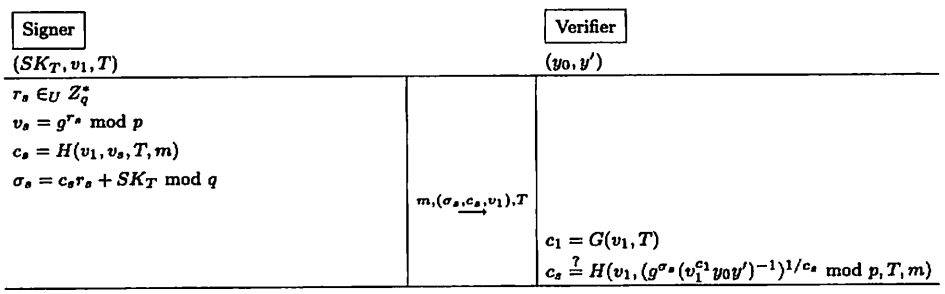

図 4 提案する Strong Key-Insulated 署名 (2) 署名生成・検証

- 部分鍵生成（図 3）

セキュアデバイス内において，乱数 r_1 を Z_q^* からランダムに選び，$v_1 = g^{r_1} \bmod p$ を求める．次に，時刻情報 T を用いて $c_1 = G(v_1, T)$ を求め，さらにマスター鍵 x_0 を用いて部分鍵 $x_1 = c_1 r_1 + x_0 \bmod q$ を求め，x_1, v_1, T を署名者に送る．署名者は $c_1 = G(v_1, T)$ を求め，$g^{x_1} = v_1^{c_1} y_0 \bmod p$ が成り立つかどうか検証を行う．

- 鍵更新（図 3）

部分鍵の検証に成功した場合のみ，x' を用いて時刻 T の署名鍵 $SK_T = x_1 + x' \bmod q$ を求め，署名鍵の更新を行う．

- 署名生成（図 4）

署名者は，乱数 r_s を Z_q^* からランダムに選び，$v_s = g^{r_s} \bmod p$ を求める．次に，メッセージ m と署名生成時刻 T を用いて $c_s = H(v_1, v_s, T, m)$，$\sigma_s = c_s r_s + SK_T \bmod q$ を求め，$m, (\sigma_s, c_s, v_1), T$ を検証者に送る．

- 署名検証（図 4）

検証者は $c_1 = G(v_1, T)$ を求め

$$c_s = H(v_1, (g^{\sigma_s}(v_1^{c_1} y_0 y')^{-1})^{1/c_s} \bmod p, T, m)$$

が成り立つかどうか検証を行う．

4.3 提案方式の拡張

提案方式の部分鍵生成アルゴリズムでは，セキュアデバイスから送信された部分鍵が正当なものであるかどうかを，署名者側で検証することが可能である．これは，従来の Strong Key-Insulated 署名にはない特長である．

マスター鍵 x_0 を管理するセキュアデバイスが十分信頼できると仮定すれば，署名鍵更新時の部分鍵検証は不要となる．すなわち，検証鍵として y_0, y' の代わりに $y := y_0 y'$ のみを配布する．これにより，検証鍵のサイズを半分にすることが可能となる．

4.4 安 全 性

4.2 節で提案した Strong Key-insulated 署名の安全性について述べる．

本稿では，以下のような方針で提案方式の安全性証明を行う．まず，離散対数問題 (DLP) が困難であるという仮定の下で，提案方式のベースとなる変形 Schnorr 署名が安全であることを示す．次に，変形 Schnorr 署名が安全であるという仮定の下で，提案方式が安全であることを示す．以上の結果より，提案方式の安全性が離散対数問題の困難性に帰着されることを証明する．

定義 4.1. *Schnorr* 署名を変形した以下の署名方式を，変形 *Schnorr* 署名と定義する．

- 鍵生成

素数 p, q ($q | p-1$) と，乗法群 Z_p^* の元 g によって生成される部分群の位数が q となるような生成元 g を定める．次に，署名鍵 x を Z_q からランダムに選び，$y = g^x \bmod p$ を求め，検証鍵 $VK = \langle p, q, g, y, G(\cdot,\cdot)\rangle$ を公開する．ここで，G はハッシュ関数 $G : Z_p \times \{0,1\}^* \to Z_q$ である．

- 署名生成

署名者は，乱数 r を Z_q^* からランダムに選び，$v = g^r \bmod p$ を求める．次に，メッセージ m を用いて $c = G(v, m)$，$\sigma = cr + x \bmod q$ を求め，$m, (v, \sigma)$ を検証者に送る．

- 署名検証

検証者は $c = G(v, m)$ を求め，$g^\sigma = v^c y \bmod p$ が成り立つかどうか検証を行う．

定義 4.2. 署名方式 Π = **(Gen, Sig, Ver)** の安全性を以下の

ように定義する．

攻撃者に対し，メッセージ m を入力し対応する署名 σ を出力する署名オラクル $\mathbf{Sig}(\cdot)$ へのアクセスを許可する．任意の攻撃者 $\mathcal{F}$ に対し，次式を定義する．

$$\mathbf{Succ}_{\mathcal{F},\Pi}(k) := Pr\Big[\ \mathbf{Ver}_{VK}(m^*,\sigma^*)=1\ \Big|\ VK \leftarrow \mathbf{Gen}(1^k),\ (m^*,\sigma^*) \leftarrow \mathcal{F}^{\mathbf{Sig}(\cdot)}(VK)\ \Big]$$

ここで，m^* は署名オラクルに与えないものとする．このとき，$\mathbf{Succ}_{\mathcal{F},\Pi}(k)$ の値が無視できるならば，Π は *EUF-ACMA* 安全であるという．

補題 4.3. 離散対数問題 *(DLP)* を解くことが困難であると仮定する．このとき，変形 *Schnorr* 署名は *EUF-ACMA* 安全である．

Proof. 変形 Schnorr 署名を多項式時間で偽造可能な確率的アルゴリズム $\mathcal{A}$ が存在すると仮定する．このとき，Forking Lemma [6] より，$\mathcal{A}$ は無視できない確率で，異なる 2 つの有効な署名 (m,v,c,σ), (m,v,c',σ') を多項式時間で出力する．よって，2 つの連立方程式 $\sigma = cr+x, \sigma' = c'r+x$ が得られ，これを r,x について解くと，

$$x = \frac{c'\sigma - c\sigma'}{c'-c} \bmod q$$

が求まる．これは，$y=g^x \bmod p$ の離散対数 x が求められることを示しており，離散対数問題 (DLP) を解くことが困難であるという仮定に反する．ゆえに，変形 Schnorr 署名を多項式時間で偽造可能な確率的アルゴリズム $\mathcal{A}$ は存在しない． □

補題 4.4. 変形 *Schnorr* 署名が *EUF-ACMA* 安全であると仮定する．このとき，4.2 節で提案した *Strong Key-insulated* 署名（以下，提案方式とする）は *strong key-insulated* である．

Proof. 提案方式の署名を多項式時間で偽造可能な確率的アルゴリズム $\mathcal{A}_p$ が存在すると仮定する．このとき，$\mathcal{A}_p$ を用いて変形 Schnorr 署名を多項式時間で偽造することのできるシミュレータ $\mathcal{B}_{ms}$ が存在することを示す．

素数 p,q ($q|p-1$) と，乗法群 Z_q^* の元 g によって生成される部分群の位数が q となるような生成元 g を定める．次に，マスター鍵 x_0 を Z_q からランダムに選び，$y_0 = g^{x_0} \bmod p$ を求める．いま，p,q,g,y_0 がシミュレータ $\mathcal{B}_{ms}$ に与えられているものとする．$\mathcal{B}_{ms}$ は x' を Z_q^* からランダムに選び，$y' = y_0 g^{x'}$ を求める．そして，p,q,g,y' を $\mathcal{A}_p$ に送る．

$\mathcal{A}_p$ に対し，鍵漏洩オラクル $\mathbf{Exp1}(\cdot)$, $\mathbf{Exp2}(\cdot)$ へのアクセスを許可する．$\mathbf{Exp1}(\cdot)$ は，マスター鍵要求メッセージを入力しマスター鍵を出力する鍵漏洩オラクル，$\mathbf{Exp2}(\cdot)$ は，時刻を入力し対応する署名鍵を出力する鍵漏洩オラクルである．攻撃者は $\mathbf{Exp1}(\cdot)$, $\mathbf{Exp2}(\cdot)$ のどちらにアクセスするかを適応的に選択できるが，一方の鍵漏洩オラクルにアクセスした後は，他方の鍵漏洩オラクルにアクセスすることはできない．また，$\mathcal{A}_p$ の署名偽造対象である時刻を $\mathbf{Exp2}(\cdot)$ へ入力することはできないものとする．さらに，$\mathcal{A}_p$ に対し，メッセージ m，時刻 T を入力し対応する署名 (σ_s,v_s,v_1) を出力する署名オラクル $\mathbf{Sig}(\cdot,\cdot)$，および，ランダムオラクル $\mathbf{G}(\cdot,\cdot)$, $\mathbf{H}(\cdot,\cdot,\cdot,\cdot)$ へのアクセスを許可する．ただし，$\mathcal{A}_p$ の署名偽造対象であるメッセージと時刻のペアを $\mathbf{Sig}(\cdot,\cdot)$ へ入力することはできないものとする．一方，変形 Schnorr 署名を偽造する $\mathcal{B}_{ms}$ に対し，メッセージを入力し対応する署名を出力する署名オラクル $\mathbf{Sig}'(\cdot)$，および，ランダムオラクル $\mathbf{G}'(\cdot,\cdot)$ へのアクセスを許可する．ただし，$\mathcal{B}_{ms}$ の署名偽造対象であるメッセージを $\mathbf{Sig}'(\cdot)$ へ入力することはできないものとする．$\mathcal{B}_{ms}$ が鍵漏洩オラクル $\mathbf{Exp1}(\cdot)$, $\mathbf{Exp2}(\cdot)$, および署名オラクル $\mathbf{Sig}(\cdot,\cdot)$ の挙動をシミュレーション可能であることを，以下に示す．

$\mathcal{A}_p$ が $\mathbf{Exp1}(\cdot)$ に対しマスター鍵要求メッセージ req を送った場合，$\mathbf{Exp1}(\cdot)$ は $\mathcal{B}_{ms}$ が予めランダムに生成した x' をマスター鍵として $\mathcal{A}_p$ に送る．

$\mathcal{A}_p$ が $\mathbf{Sig}(\cdot,\cdot)$ に対しメッセージ m および時刻 T を送る場合を考える．このとき，$\mathcal{A}_p$ が時刻 T に対応する署名鍵の問い合わせを事前に $\mathbf{Exp2}(\cdot)$ へ行うか否かで，場合分けを行う．

Case 1: 署名鍵の問い合わせを事前に行う場合

まず，署名鍵の問い合わせを行う．$\mathcal{A}_p$ は $\mathbf{Exp2}(\cdot)$ に対し時刻 T を送る．次に，$\mathbf{Exp2}(\cdot)$ は $\mathbf{Sig}'(\cdot)$ に対し時刻 T をそのまま送る．$\mathbf{Sig}'(\cdot)$ は r_1 を Z_q^* からランダムに選び，$v_1 = g^{r_1}$ を求める．そして，v_1 および時刻 T をランダムオラクル $\mathbf{G}'(\cdot,\cdot)$ に送る．$\mathbf{G}'(\cdot,\cdot)$ は c_1 を $\mathbf{Sig}'(\cdot)$ に返す．$\mathbf{Sig}'(\cdot)$ はマスター鍵 x_0 を用いて $x_1 = c_1 r_1 + x_0$ を求め，(x_1,v_1) を $\mathbf{Exp2}(\cdot)$ に送る．$\mathbf{Exp2}(\cdot)$ は受信履歴として (T,x_1,v_1) をリストに追加する．そして，マスター鍵 x' を用いて署名鍵 $SK_T = x_1 + x'$ を求め，(SK_T,v_1) を $\mathcal{A}_p$ に送る．

次に，署名の問い合わせを行う．$\mathcal{A}_p$ は $\mathbf{Sig}(\cdot,\cdot)$ に対し，メッセージ m および時刻 T を送る．$\mathbf{Sig}(\cdot,\cdot)$ は，署名鍵の問い合わせ時に求めた署名鍵 SK_T および v_1 を用いて署名 (σ_s,v_s,v_1) を生成し，$\mathcal{A}_p$ に送る．

Case 2: 署名鍵の問い合わせを事前に行わない場合

通常，$\mathcal{A}_p$ から署名の問い合わせがあった場合，$\mathcal{B}_{ms}$ は以下のように対応する．

$\mathcal{A}_p$ は $\mathbf{Sig}(\cdot,\cdot)$ に対し，メッセージ m および時刻 T を送る．$\mathbf{Sig}(\cdot,\cdot)$ は $\mathbf{Exp2}(\cdot)$ にアクセスし，時刻 T に対応する署名鍵 SK_T および v_1 を得る．$\mathbf{Sig}(\cdot,\cdot)$ は (SK_T,v_1) を用いて，(m,T) に対応する署名 (σ_s,v_s,v_1) を生成し，$\mathcal{A}_p$ に送る．

しかし，$\mathcal{A}_p$ は自分が署名偽造したい対象のメッセージおよび時刻である (m^*,T^*) を $\mathbf{Sig}(\cdot,\cdot)$ に送信することはできず，また，$\mathcal{B}_{ms}$ は自分が署名偽造したい対象のメッセージである T^* を $\mathbf{Sig}'(\cdot)$ に送信することはできない．従って，上記のような対応だけでは，$\mathcal{A}_p$ は (m^*,T^*) に関する情報を何も得られないことになる．

そこで，$\mathcal{B}_{ms}$ は以下のような戦略をとる．$\mathcal{A}_p$ は署名の偽造を行う前に，自分が署名偽造したい対象の時刻 T^* に対応するハッシュ値をランダムオラクル $\mathbf{H}(\cdot,\cdot,\cdot,\cdot)$ に問い合わせるものと仮定する．$\mathbf{Sig}(\cdot,\cdot)$ は署名鍵を得ることなく，シミュレーションにより (v_1,v_s,T^*,m), c_s のペアを求め，ランダムオラ

クル $\mathbf{H}(\cdot,\cdot,\cdot,\cdot)$ の入出力関係を示すリストに追加する．ランダムオラクル $\mathbf{H}(\cdot,\cdot,\cdot,\cdot)$ への問い合わせ回数を q_H 回としたとき，$\mathcal{B}_{ms}$ は j 番目の問い合わせが (m^*, T^*) に関する問い合わせであると予想する．もし，この予想が当たれば，$\mathcal{B}_{ms}$ のシミュレーションは成功となる．具体的には，$\mathcal{B}_{ms}$ は $\mathcal{A}_p$ による署名の問い合わせに対し，以下のような対応を行う．

$\mathcal{A}_p$ は $\mathbf{Sig}(\cdot,\cdot)$ に対し，メッセージ m および時刻 T^* を送る．$\mathbf{Sig}(\cdot,\cdot)$ は r_1 を Z_q^* からランダムに選び，$v_1 = g^{r_1}$ を求める．そして，(v_1, T^*) をランダムオラクル $\mathbf{G}'(\cdot,\cdot)$ に送る．$\mathbf{G}'(\cdot,\cdot)$ は c_1 を $\mathbf{Sig}(\cdot,\cdot)$ に返す．$\mathbf{Sig}(\cdot,\cdot)$ は c_s, σ_s を Z_q^* からランダムに選び，

$$v_s = \left(\frac{g^{\sigma_s}}{v_1^{c_1} y^{r}}\right)^{1/c_s}$$

を求める．そして，ランダムオラクル $\mathbf{H}(\cdot,\cdot,\cdot,\cdot)$ の入出力関係を示すリストに，(v_1, v_s, T^*, m), c_s のペアを追加する．最後に，$\mathbf{Sig}(\cdot,\cdot)$ は $\mathcal{A}_p$ に署名 (σ_s, v_s, v_1) を送る．

ところで，Forking Lemma [6] より，$\mathcal{A}_p$ は無視できない確率で，時刻 T^* およびメッセージ m^* に対する，異なる 2 つの有効な署名 $(m^*, (\sigma_s, c_s, v_1), T^*)$, $(m^*, (\sigma'_s, c'_s, v_1), T^*)$ を多項式時間で出力する．よって，2 つの連立方程式 $\sigma_s = c_s r_s + SK_{T^*}, \sigma'_s = c'_s r_s + SK_{T^*}$ が得られ，これを r_s, SK_{T^*} について解くと，

$$SK_{T^*} = \frac{c'_s\sigma_s - c_s\sigma'_s}{c'_s - c_s} \bmod q$$

が求まる．よって，$\mathcal{B}_{ms}$ は $x_1^* = SK_{T^*} - x'$ を求め，(x_1^*, v_1, T^*) をメッセージ T^* に対する変形 Schnorr 署名として出力する．

ここで，$\mathcal{B}_{ms}$ が (x_1^*, v_1, T^*) の偽造に成功する確率を $\epsilon_{\mathcal{B}_{ms}}$，$\mathcal{A}_p$ が $(m^*, (\sigma_s, c_s, v_1), T^*)$, $(m^*, (\sigma'_s, c'_s, v_1), T^*)$ の偽造に成功する確率を $\epsilon_{\mathcal{A}_p}$ とすると，$\epsilon_{\mathcal{B}_{ms}}$ は $\epsilon_{\mathcal{A}_p}$ に（ランダムオラクル $\mathbf{H}(\cdot,\cdot,\cdot,\cdot)$ への問い合わせ予想が当たる確率）を掛け合わせたものであるから，

$$\epsilon_{\mathcal{B}_{ms}} = \epsilon_{\mathcal{A}_p} \cdot \frac{1}{q_H}$$

となる．

また，ランダムオラクル $\mathbf{H}(\cdot,\cdot,\cdot,\cdot)$ への入力を $Q_1, Q_2, \ldots, Q_{q_H}$，出力を $\rho_1, \rho_2, \ldots, \rho_{q_H}$ とし，$\mathcal{A}_p$ が $Q_j = (v_1, v_s, T^*, m^*)$ に対応する署名 $(m^*, (\sigma_s, c_s, v_1), T^*)$ を確率 $1/P(n)$ 以上で出力するものとする（$P(n)$: 検証鍵長 n の多項式）．このとき，Forking Lemma [6] より，$\mathcal{A}_p$ が $Q_j = (v_1, v_s, T^*, m^*)$ に対応する，異なる 2 つの有効な署名 $(m^*, (\sigma_s, c_s, v_1), T^*)$, $(m^*, (\sigma'_s, c'_s, v_1), T^*)$ を確率 $1/4P(n)$ 以上で出力するような，ランダムテープ ω の集合 Ω_j およびランダムオラクルの出力 $(\rho_1, \rho_2, \ldots, \rho_{j-1})$ の集合 $R_{j,\omega}$ が存在する．従って，

$$\epsilon_{\mathcal{A}_p} \geq \frac{1}{4P(n)}$$

となる．

ゆえに，

$$\epsilon_{\mathcal{B}_{ms}} \geq \frac{1}{4q_H P(n)}$$

となり，$\mathcal{B}_{ms}$ は無視できない確率で，$\mathcal{A}_p$ を用いて変形 Schnorr 署名を多項式時間で偽造できる．□

補題 4.3，4.4 より，以下の定理が得られる．

定理 4.5. 離散対数問題 *(DLP)* を解くことが困難であると仮定する．このとき，提案方式は ***strong key-insulated*** である．

4.5 性能比較

4.2 節で提案した Strong Key-Insulated 署名と，従来の Strong Key-Insulated 署名の比較結果を表 1 に示す．比較対象は，Schnorr 署名と証明書を用いた方式（付録 A 参照）と，[2] で述べられている Guillou-Quisquater 署名に基づく方式（付録 B 参照）とし，検証鍵のビット長 $(p, q, g, G(\cdot,\cdot), H(\cdot,\cdot,\cdot,\cdot)$ はシステム共通のパラメータのため除く），署名のビット長，計算量（べき乗剰余演算の回数）の各項目を比較した．なお，表中のビット長は，有限体上の変数を 1024bit，部分群上の変数を 160bit として計算した．表 1 に示す通り，提案方式は，証明書に基づく方式よりも，検証鍵長，署名長，計算量の全ての点において効率的である．また，Guillou-Quisquater 署名に基づく方式よりも，署名長，署名生成計算量の点において効率的である．従って，提案方式は従来よりも効率的な Strong Key-Insulated 署名である．

また，Guillou-Quisquater 署名に基づく方式は，RSA 問題の困難性 (RSA 仮定) を安全性の根拠としているのに対し，提案方式は，離散対数問題の困難性 (DL 仮定) を安全性の根拠としている．従って，提案方式は安全性の面でも優位である．

4.6 コンテンツ配信システムにおける効果

4.2 節で提案した Strong Key-Insulated 署名を 2. 章で述べたコンテンツ配信システムに適用した場合の効果を示す．

- マスター鍵の漏洩被害の軽減

マスター鍵 x_0 を鍵管理サーバ，x' を個人情報管理サーバが管理することで，マスター鍵 x_0 が外部に漏洩しても，x' が管理されているサーバがなければ署名鍵の更新はできず，マスター鍵の漏洩被害を軽減できる．

- 効率的な署名生成・検証

検証鍵は CAS カードに格納されるが，提案方式における検証鍵の長さは従来方式と比較して短いため，限られた CAS カードのメモリ領域でも十分に格納できる．また，署名長も従来方式と比較して短く，計算量も少ないため，署名付き個人情報リクエストを全視聴者に配信しても，システムに与える影響は少ない．従って，効率的な署名生成・検証を行うことができる．

5. まとめ

本稿では，署名鍵およびマスター鍵の漏洩に耐性を有する Strong Key-Insulated 署名の構成法を提案し，安全性の証明を行った．本方式は従来よりも効率的な方式である．また，我々が提案するコンテンツ配信システムに適用することにより，双方向放送サービスにおいて，視聴者の個人情報を安全に送受信可能であることを示した．

文献

[1] 大竹剛，花岡悟一郎，小川一人，"双方向放送サービスにおける検証鍵不変なプロバイダ認証システム，" 2006 年 暗号と情報セキュリティシンポジウム (SCIS'06) 予稿集，2F4-3, 2006.

表 1 Strong Key-Insulated 署名の性能比較

	検証鍵長 (bit)	署名長 (bit)	計算量 (べき乗剰余演算回数)	安全性の根拠
証明書に基づく方式 (Schnorr)	2048	1984	署名生成: 3, 署名検証: 6	DL 仮定
GQ 署名に基づく方式 [2]	1024	2048	署名生成: 2, 署名検証: 2	RSA 仮定
提案方式	1024	1344	署名生成: 1, 署名検証: 3	DL 仮定

[2] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Strong Key-Insulated Signature Schemes," Proc. of PKC'03, pp. 130–144, 2003.

[3] M. Abe and T. Okamoto, "Delegation Chains Secure up to Constant Length," IEICE Trans. Fundamentals, vol. E85-A, no. 1, pp. 110–116, 2002.

[4] 大竹剛, 花岡悟一郎, 小川一人, "双方向放送サービスのための効率的な Strong Key-Insulated 署名," 第 29 回情報理論とその応用シンポジウム (SITA'06) 予稿集, pp. 311–314, 2006.

[5] T. Malkin, S. Obana, and M. Yung, "The Hierarchy of Key Evolving Signatures and a Characterization of Proxy Signatures," Proc. of Eurocrypt'04, pp. 306–322, 2004.

[6] D. Pointcheval and J. Stern, "Security Proofs for Signature Schemes," Proc. of Eurocrypt'96, pp. 387–398, 1996.

付　　録

付録 A　証明書に基づく方式

Schnorr 署名と証明書を用いた Strong Key-Insulated 署名の構成法を以下に示す.

まず, 署名鍵を発行するセキュアデバイス内にマスター鍵 x_0 を, 署名者のデバイス内に x' をそれぞれ格納しておく. また, 対応する公開鍵 $y_0 = g^{x_0} \bmod p$, $y' = g^{x'} \bmod p$ を公開しておく. 次に, セキュアデバイス内において, 時刻 T に対応する署名鍵 SK_T と検証鍵 VK_T を生成し, 証明書 $\mathrm{Sig}_{x_0}(VK_T)$ を添えて署名者に送る. ここで, $\mathrm{Sig}_x(y)$ は鍵 x を用いたメッセージ y に対する Schnorr 署名を表す. 署名者は, メッセージ m に対し, 署名鍵 SK_T を用いた署名 $\mathrm{Sig}_{SK_T}(m)$, および, x' を用いた証明書 $\mathrm{Sig}_{x'}(m)$ を付加し, 検証鍵 VK_T と証明書 $\mathrm{Sig}_{x_0}(VK_T)$ を添えて, 検証者に送る. 検証者は, 公開鍵 y_0, y' を用いて証明書 $\mathrm{Sig}_{x_0}(VK_T), \mathrm{Sig}_{x'}(m)$ の検証を行い, さらに, 検証鍵 VK_T を用いて署名 $\mathrm{Sig}_{SK_T}(m)$ の検証を行う.

付録 B　Guillou-Quisquater 署名に基づく方式

Guillou-Quisquater 署名に基づく Strong Key-Insulated 署名の構成法を以下に示す.

まず, 素数 p, q を Z_n からランダムに選び, $n = pq$ を求め, 公開する. 次に, $(p-1)(q-1)$ と互いに素となるような e を Z_n から選び, $d = 1/e \bmod (p-1)(q-1)$ を求める. e は検証鍵として公開する. そして, $d_1 + d_2 = d \bmod (p-1)(q-1)$ となるように d_1, d_2 を定め, マスター鍵 d_1 をセキュアデバイス内に, d_2 を署名者のデバイス内にそれぞれ格納しておく. セキュアデバイスはマスター鍵 d_1 と時刻情報 T を用いて部分鍵 $x_1 = (1/f(T))^{d_1}$ を生成する (f は関数 $f : \{0,1\}^* \to Z_n^*$). 部分鍵 x_1 と時刻情報 T を署名者に送る. 署名者は, d_2 を用いて, 時刻 T における署名鍵 $SK_T = x_1 \cdot (1/f(T))^{d_2}$ を生成する. 次に, 乱数 $k \in_U Z_n$ を生成し, $r = k^e \bmod n$ を求める. そして, メッセージ m に対し, 署名鍵 SK_T を用いて $l = h(m||r)$, $s = k \cdot SK_T^l \bmod n$ を求め, 署名 (s, l) を生成する (h はハッシュ関数 $h : \{0,1\}^* \to \{0,1\}^c$, c は定数). メッセージ m, 署名 (s, l), 署名生成時刻 T を検証者に送る. 検証者は, 検証鍵 e を用いて $u = s^e \cdot f(T)^l \bmod n$, および $l' = h(m||u)$ を求め, $l = l'$ が成り立つかどうか検証を行う.