SCIS 2003 The 2003 Symposium on Cryptography and Information Security
Hamamatsu,Japan, Jan.26-29,2003
The Institute of Electronics, Information and Communication Engineers

署名鍵漏洩問題における電子証拠生成技術について

Generating Digital Evidence against Exposure of Signing Key

小森 旭* 花岡 悟一郎† 松浦 幹太*† 須藤 修*
Akira Komori Goichiro Hanaoka Kanta Matsuura Osamu Sudo

あらまし PKI が電子社会のインフラになり，電子署名の利用が広まるにつれ，電子署名生成用秘密鍵の漏洩が社会にもたらす影響は大きくなることが予想される．そこで我々は以前，電子署名偽造対策の一つとして”証拠”というものに焦点を当て，MAC 付き電子署名方式を提案した．これに引き続き本稿では，秘密鍵が漏洩する要因と従来の対策技術の関係について検討するとともに，これまでより適用範囲を広げた電子証拠生成方式を提案する．

キーワード PKI，電子署名，秘密鍵漏洩，電子証拠物

1 はじめに

PKI(Public Key Infrastructure)・電子認証は，電子商取引や電子政府などを安全かつ確実なものとするために必要不可欠なセキュリティ基盤である [6]．2002 年 12 月に『電子署名に係る地方公共団体の認証業務に関する法律案』が成立し，2003 年 8 月以降，希望者には全国の市町村が鍵ペアと公開鍵証明書を発行する予定であり，今後 PKI の一層の普及・拡大が期待されている [11]．

PKI を利用するシステムにおいて最大の脅威は，署名生成用秘密鍵 (以下，単に秘密鍵という) の漏洩である．そこで，外部への漏洩を防止するために，秘密鍵は IC カード等の耐タンパーデバイスに格納される．しかし，タイミング解析や電力解析などのさまざまな攻撃法が提案されており，耐タンパー性が破綻して秘密鍵が漏洩する可能性は否定できない．いったん秘密鍵が漏洩すると攻撃者は，証明書の有効期間内に生成されたとされるすべての署名の偽造が可能となる．

さらに，この問題の難しさは，秘密鍵の所有者 (以下，署名生成者という) が秘密鍵の漏洩をすぐに検知できないという点にある．一般に署名生成者が秘密鍵の漏洩に気づくのは，鍵が悪用された後，生成した覚えのない署名が自分の元に持ち込まれてからである [12]．しかも，漏洩発覚後すぐに証明書の廃棄を申請しても，廃棄申請が受理され CRL(Certificate Revocation List) に反映されるまでにはかなりのタイムラグがあり，この間に被害が大きくなる危険性がある．

そこで我々は，IC カードに格納された秘密鍵が漏洩した場合の対策として証拠性の確保が重要であると考え，MAC(Message Authentication Code) 付き電子署名方式 (以下，単に MAC 方式という) を提案した [9]．しかし，MAC 方式は，秘密鍵が IC カード外部で生成され定期的に更新されるという PKI モデルを想定していたため，効果が期待できる範囲が限定されていた．そこで本稿では，秘密鍵が漏洩する要因と従来の対策技術の関係について，時間に焦点を当てて検討するとともに，秘密鍵を IC カード内部で生成するモデルまで適用範囲を広げた電子証拠生成方式を提案する．

2 秘密鍵漏洩の要因と従来の対策技術

2.1 秘密鍵漏洩の要因

秘密鍵が漏洩する要因として，さまざまなケースが想定される．本稿では，それらの要因が鍵のライフサイクル上のいつ起こるかによって 3 つに分類してみた．なお，本論文では，秘密鍵は IC カードなどの耐タンパー性を持ったハードウェアに格納されているものとする．

2.1.1 鍵ペア生成時から公開鍵証明書発行時までの間

- 鍵管理の運用方法の欠陥: 秘密鍵を IC カードの外部で生成し，その後 IC カードに格納する方法が利用される場合，秘密鍵の生成場所に存在する鍵のコピーが完全に消去されずに攻撃者の手に渡る．
- 公開鍵証明書発行時における欠陥: 証明書発行時の本人確認方法に問題があり，他者になりすました攻撃者に，正規ユーザの証明書を発行してしまう．

* 東京大学大学院 情報学環・学際情報学府, 〒 113-0033 東京都文京区本郷 7-3-1, Interfaculty Initiative in Information Studies, Graduate School of Interdisciplinary Information Studies, Univ. of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
† 東京大学 生産技術研究所, 〒 153-8505 東京都目黒区駒場 4-6-1, Institute of Industrial Science, Univ. of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo, 153-8505, Japan

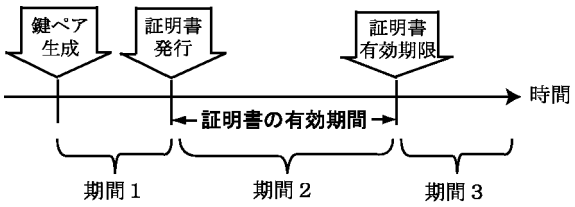

図 1: 鍵のライフサイクルと秘密鍵漏洩時期

2.1.2 公開鍵証明書の有効期間内 (秘密鍵の利用中)

- プローブ解析: ハードウェアのパッケージを除去した上で，内部の回路に直接プローブ (回路測定用電極) をあてて回路の電気信号を探り，秘密鍵等のデータが盗み出される [5].
- 故障利用攻撃: 加熱や放射線照射等によってハードウェア内部の回路に意図的に故障を起こさせ，ハードウェアの正常時の出力データや故障時の出力データ等を利用して回路に格納されているデータが推測される [2].
- サイド・チャネル攻撃: 暗号処理を実行する際にハードウェアから漏洩するデータ (暗号処理時間，消費電力，周波数など) を解析し，ハードウェア内部に格納されているデータが推測される [7][8].
- 電子署名方式の欠陥: 電子署名のアルゴリズム自体に欠陥が存在している場合や，アルゴリズム自体は安全であっても，実装されたプログラムに重大なバグが存在する場合，署名検証用鍵や署名等の公開情報から秘密鍵が効率的に導出される．

2.1.3 公開鍵証明書の有効期限後 (秘密鍵の廃棄後)

- 飛躍的に高い計算能力を持ったコンピュータが実用化されることにより，暗号がブレイクする (署名検証用鍵や署名等の公開情報から秘密鍵が効率的に導出される).

2.2 秘密鍵の漏洩対策技術

本節では，従来の秘密鍵漏洩対策技術を紹介する．

2.2.1 Forward-Secure 署名方式

(概要)

公開鍵は固定したままで，秘密鍵を短い期間 (例えば1日) ごとに一方向性関数を用いて更新していくことにより，ある期間の秘密鍵が盗まれても，その期間以前に生成されたとされる署名の偽造を防止する方式である [1].

(問題点)

- 鍵の更新ルールは公開なので，ある期間における秘密鍵が漏洩した場合，その期間以降のすべての署名の偽造が可能である．
- 偽造された署名を発見した際，署名生成者は漏洩した秘密鍵のうちで最も古いものを検知する必要があるが，どの時点以前の署名が安全かを特定することは困難である [13].
- RSA や DSA 等の署名方式とは異なるタイプの独自のアルゴリズムを採用しており汎用性が低い [15].

2.2.2 Key-Insulated 署名方式

(概要)

基本アイデアは Forward-Secure 署名と同じで，秘密鍵を短い期間ごとに更新していく方式である．ただし，秘密鍵の更新は，その都度マスターキーから出力されるデータを介して行われるので，ある期間の秘密鍵が盗まれても，その期間以前だけではなく以降に生成されたとされる署名の偽造も防止することができる [4].

(問題点)

- Forward-Secure 署名と同様，署名の偽造可能な範囲を制限できるだけであり，鍵が盗まれてしまえばその期間の署名が偽造できることには変わりない．

2.2.3 実行ハードウェア確認タグ付き署名方式

(概要)

検証対象の署名が特定のハードウェアにおいて生成されたか否かを確認することで署名偽造を検知する方式である．署名を生成する際は，物理的に複製困難な耐クローンモジュールの出力等から”タグ”を生成し，署名に添付する [16].

(問題点)

- タグ付き署名を生成するためには，オーソリティとの間の通信が必要である．したがって，署名生成者は自分ひとりだけでは，正当なタグ付き署名を生成することはできない．さらに，オーソリティが DoS 攻撃を受け負荷が集中していると，サービスを受けたいときに受けられない恐れがある．
- 耐クローンモジュールの機能を実現する方法は，今のところ存在しない．
- 署名生成者自身は不正な行為を行わないという前提を設けている．

2.2.4 ヒステリシス署名

(概要)

署名の生成履歴を署名生成者自身でも偽造困難な形態で保管し，秘密鍵漏洩により署名が偽造された場合であっても，署名生成履歴との整合性を確認することによって，署名偽造を検知する方式である [10].

(問題点)

- 署名履歴や取引履歴ファイル [14] を保管するための容量が耐タンパーデバイスに必要である．

- 署名生成者自身が不正を行った場合に対する備えが不十分である．例えば，二重帳簿的な不正を行うことが可能である．

2.2.5 MAC 付き電子署名方式

あるデータに対する署名を生成する際に，署名とは別に新たに電子証拠データとして MAC を付加する方式である．MAC 生成に使用する鍵は，固定な方式と内部で更新可能な方式の 2 つがある．詳細についてはそれぞれ **4** 章・**5** 章で説明する．

2.3 従来の対策技術が想定する秘密鍵の漏洩要因

従来の秘密鍵漏洩対策技術が想定する秘密鍵の漏洩要因をまとめると表 1 のようになる．

表 1: 従来の対策技術と秘密鍵漏洩時期の関係

	FS	KI	タグ	履歴	MAC	提案
期間 1	×	×	○	○	◎	○
期間 2	◎	◎	○	○	—	◎
期間 3	×	×	—	◎	—	—

◎: 特に想定, ○: 想定, ×: 使えない, —: どちらともいえない

FS･･･ Forward-Secure 署名　履歴･･･ ヒステリシス署名
KI･･･ Key-Insulated 署名　MAC･･･ MAC 付き電子署名
タグ･･･ タグ付き電子署名　提案･･･ 提案方式

例えば，FS 署名方式は図 1 における期間 2 の漏洩のみを想定している．ヒステリシス署名は，暗号ブレイク時における過去の署名の偽造を特に想定しているので，期間 3 における漏洩対策が中心であると考えられる．また，MAC 方式は「秘密鍵が耐タンパー領域の外部で生成される」という前提を設けているので，期間 1 における対策が中心である．これに対し提案方式 (**5** 章) は，入出力から耐タンパー内のデータを推測する攻撃を想定しているので，期間 2 における対策が中心である．ただし，秘密鍵漏洩対策技術のほとんどが，秘密鍵の漏洩する要因を特定しておらず，方式によって前提条件が異なるので，**必ずしも”○”が多い方が優れているとは限らない**．

3 証明書廃棄の限界

秘密鍵漏洩をすぐに検知することができ，かつ，すぐにすべての人にその事実を知らせることができれば，**2.1** 節で述べたいくつかの漏洩要因は対策を考える必要が省ける．そこで本章では，そのようなことが既存技術で可能かどうか考察を行う．

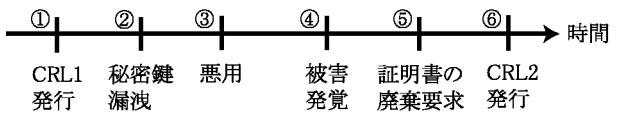

図 2: 公開鍵証明書廃棄の時間経緯

秘密鍵の漏洩した期間が証明書の有効期限内である場合，RFC2527[3] の規定により，ユーザはすぐに証明書の廃棄申請をする義務がある．図 2 に証明書廃棄の時間経緯を示す．

一般に秘密鍵は紙の世界における印鑑とは異なりディジタルデータなので，鍵がコピーされ盗まれても (図 2-②) すぐにそれを検出することは容易ではない．つまり，実際にユーザが秘密鍵の漏洩に気づくのは，鍵を盗んだ第三者により何からの悪用をされ (図 2-③)，被害に遭ってからであり (図 2-④)，その間に被害が大きくなる可能性がある．しかも，漏洩発覚後すぐに証明書の廃棄を申請しても (図 2-⑤)，CRL は一定周期で発行されているので (図 2-①⑥)，申請がすぐに CRL に反映されない．この問題の解決策として，基準となる CRL からの差分の失効情報のみをより短い間隔で発行するデルタ CRL や，オンラインで証明書が有効かどうかを問い合わせる OCSP(Online Certificate Status Protocol) といった方法が提案されているが，タイムラグが発生することに変わりはなく，問題の本質的な解決策にはなっていない [6].

そこで我々は，正当な電子署名付きメッセージとは別により強い証拠を付加することで，秘密鍵が盗まれ電子署名が偽造された場合でも，それによって生じる署名生成者の損失をできる限り保護する方式を考えた．

4 MAC 付き電子署名方式

MAC 付き電子署名方式の特徴は，署名生成者と署名検証者の間でやり取りされるデータの証拠性を確保するため，証拠機能を電子署名ではなく MAC に持たせている点である．ただし，論文 [9] の方式では，**署名生成者と署名検証者が結託することによる MAC の偽造がコストなしで可能であった**．(ここでいう”偽造”とは，ある検証をパスするようにデータを偽造するという意味ではなく，元々は検証をパスするはずだったデータをでたらめな値にして検証をパスしないものを作ることを意味する)．そこで本稿では，この問題を解決策をするため，MAC を署名に含めた改良方式を示す．

4.1 エンティティ

我々の提案方式は，CA，IC カード発行者，署名生成者，署名検証者，紛争解決時の調停者 (TTP) などのエンティティから構成される．調停者を除くエンティティは，不正な行為を行う可能性がある．

4.2 記号の定義

本論文で使用する記号の意味を以下に記す．

- SK_A：エンティティ A の署名生成用鍵 (秘密鍵)
- K：IC カード固有の秘密データ (鍵)
- M：契約書などの署名対象データ
- $SIG_{SK}(M)$：秘密鍵 SK を使ってデータ M に電子署名を施したもの＋平文のデータ M
- H(M)：データ M のハッシュ値

- $MAC_K(M)$：鍵 K と署名対象データ M の 2 つから，一方向的に計算したもの．

4.3 想定環境・脅威

MAC 方式では，次のような利用環境と脅威を想定している．

- **想定 1(IC カードの製造)**：耐タンパー領域を，「**レベル 1**：更新する必要のないデータを格納する領域」と「**レベル 2**：更新する可能性のあるデータを格納する領域」の 2 つの領域に分類し，カード製造時にデバイス固有の秘密鍵 K をレベル 1 の領域に格納しておく．また，ホログラムを用いることにより，カード自体の偽造は困難である．
- **想定 2(IC カードの発行)**：IC カード発行時の本人確認は，オンラインや郵送などの簡易なものではなく，対面で複数の書類を提示させ厳密に行う．また，カードの再発行時を除き，一人のユーザには一枚の IC カードしか発行しない．
- **想定 3(鍵ペアの生成・格納)**：公開鍵・秘密鍵の鍵ペアは CA(Certification Authority) が一括して生成し，公開鍵証明書とセットでユーザに配布する．これらのデータは定期的に更新する必要があるため，耐タンパーレベル 2 の領域に格納する．
- **想定 4(秘密データの漏洩)**：**2.1.1** 節で述べた要因により秘密鍵が漏洩する．あるいは，秘密鍵を CA からユーザへ配布するまでの経路上で漏洩する．また，耐タンパーレベル 2 の方が，セキュリティホールが発見される可能性が高いものと予想される．
- **想定 5(IC カードの利用)**：電子署名を施す作業をするときは，署名検証者側に設置された R/W を使って署名データを送信する．つまり，対面でのやり取りに限定する．
- **想定 6(IC カードの紛失)**：IC カードごとなくせば，秘密鍵の漏洩とは異なりすぐ紛失に気付くと考えられるので，カード自体を紛失することは想定しない．もし仮に IC カードが盗まれたとしても，IC カードには使用者認証機能 (PIN) が組み込まれているので，容易に盗用することはできない．
- **想定 7(IC カード内部の通信)**：IC カード内部におけるレベル 1 とレベル 2 の間でやりとりされるデータは，不正に改ざんすることはできない．

4.4 署名生成・紛争解決

署名生成及び検証の手順は以下の通りである．

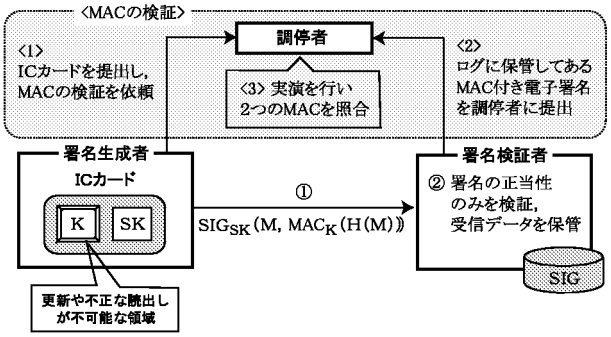

図 3: MAC 付き署名の生成と検証 (文献 [15] の図 12 を修正)

【署名生成手続き (図 3 参照)】

(1) 署名生成者は，$SIG_{SK}(M, MAC_K(H(M)))$ を生成して署名検証者に送る．

(2) 署名検証者は，$SIG_{SK}(M, MAC_K(H(M)))$ を検証し，紛争発生時に備えて安全なログに保管する．

【署名偽造の疑いが生じた際の MAC 検証手続き (図 3)】

(1) 署名生成者は IC カードを調停者に提出し，MAC の検証を依頼する．(署名生成者自身が調停の場に出頭する)．

(2) 署名検証者は，署名が偽造された疑いのある署名データ $SIG_{SK}(M, MAC_K(H(M)))$ を調停者に提出する．

(3) 調停者は実際にデータ M と IC カードを使って実演してみて，IC カードから出力される MAC と署名検証者から提出された MAC を照合する．一致する場合には M に対する署名が正当なものであると判定し，一致しない場合には，署名が偽造されたものであると判定する．

4.5 考察

(1) MAC 検証時の出頭

紛争解決時に署名生成者が調停の場に出頭することは，一見すると利便性が欠けるように思えるが，署名生成者自身が不正を行う場合について考えてみると，不正行為を成功させるためには自ら出頭する必要があるので，出頭という行為によって犯罪に対する心理的な抑止力が働くものと期待される．

(2) IC カードの利用場所

もし仮に何からの方法で署名生成者が自分の秘密鍵を知り得ると，IC カードから出力される MAC 付き署名データが，署名検証者の端末に届くまでの間の通信路上にコンバータなどの装置を設置し，MAC 部分のデータをすり替えることが可能である．それゆえ，本方式では対面での使用に限定している (想定 5)．この”対面”という前提を取り外すには，デバイス間に強力なセキュアチャネルを確保するなど，通信路上でのデータの改ざんを不可能にするための対策が必要であるといえる．

5　提案方式

MAC 付き電子署名方式は，主に PKI モデルの欠陥による秘密鍵漏洩を想定しており，対応できる脅威が限定されていた．そこで本章では，MAC 方式における仮定をゆるめ，必要に応じて新しい鍵ペアを生成できる機能を持った IC カードを想定に入れ，さらには対応できる脅威も増やすことによって，MAC 方式より適用範囲を広げた証拠生成方式を提案する (表 1 参照).

5.1　追加的な記号の定義

4.2 節に追加して以下の記号を定義する．

- PK_A：エンティティ A の暗号化用の鍵 (公開鍵)
- ID：IC カードの識別子．カード表面にも記載.
- CNT：ハッシュを行った回数を記憶するカウンター
- $K_i (i = 1,2,\cdots)$：MAC 生成用ワンタイムキー
- $E_k(X)$：鍵 k を使ってデータ X を暗号化したもの

5.2　想定環境・脅威

今回新たに提案する方式において想定する利用環境と脅威を，MAC 方式とは異なる部分のみ以下に示す.

- **想定 1(IC カードの製造)**：耐タンパー領域の分類は行わない．カード製造時に K と CNT(=0) を耐タンパー領域に，ID と PK_T を読み書き可能な領域にそれぞれ格納しておく．K は探索キーとして用いる ID と一緒に，TTP へ秘密裏に預ける．図 4 にその一例を示す.
- **想定 3(鍵ペアの生成・格納)**：公開鍵・秘密鍵の鍵ペアは，IC カードの鍵ペア生成機能を使って IC カード内部でユーザが生成する．その後，秘密鍵は耐タンパー領域に格納し，公開鍵は読み書き可能な領域に格納する.
- **想定 4(秘密データの漏洩)**：**2.1.2** 節で述べた要因により秘密データが漏洩する．例えば，署名検証者が不正者だと考えた場合，署名生成者が IC カードを検証者側の R/W に挿入している際に，署名生成プロトコルとは別の余分な実行を複数回走らせ，IC カードへの入力と出力を観察することにより，カード内部の秘密データを推測される危険性がある.

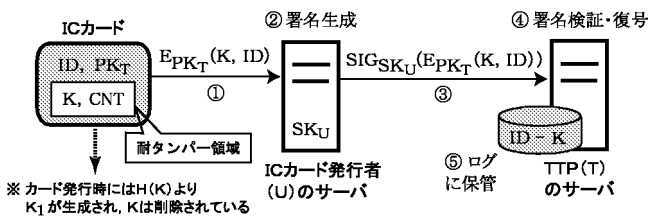

図 4: IC カードの製造過程

- **想定 8(証拠生成鍵の更新)**：証拠生成に使用する鍵は，署名を生成するたびに一方向性関数を用いて新しい鍵に更新される．すると，カウンター CNT の値は 1 増え，古い鍵は削除される．また，ユーザに IC カードを配布する際，カード内にはシード K にハッシュをかけた値 $K_1(= H(K))$ が格納され，K はカード内から削除されている (このとき CNT の値は 1 となっている)．証拠生成鍵 K_i をワンタイムキーにすることにより，K_i と秘密鍵 SK のアクセス回数に差を設けることができる.

5.3　署名生成・紛争解決

CNT の値が i の時の，署名生成及び検証の手順は以下の通りである.

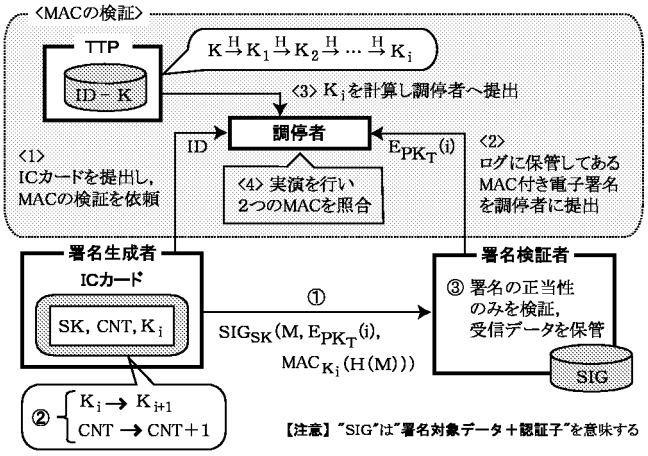

図 5: 提案方式における署名&証拠の生成と検証

【署名生成手続き (図 5 参照)】

(1) 署名生成者は, $SIG_{SK}(M, E_{PK_T}(i), MAC_{K_i}(H(M)))$ を生成し，署名検証者に送る.

(2) 証拠生成に使用する鍵を更新する $(K_{i+1} = H(K_i))$ し，カウンタ CNT の値を 1 増やす.

(3) 署名検証者は $SIG_{SK}(M, E_{PK_T}(i), MAC_{K_i}(H(M)))$ を検証し，紛争発生時に備えて安全なログに保管する.

【署名偽造の疑いが生じた際の MAC 検証手続き (図 5)】

(1) 署名生成者は IC カードを調停者に提出し，MAC の検証を依頼する．(IC カードの ID を調停者に伝える).

(2) 署名検証者は，署名が偽造された疑いのある署名データ $SIG_{SK}(M, E_{PK_T}(i), MAC_{K_i}(H(M)))$ を調停者に提出する.

(3) 調停者は,「ID に対応した鍵 K に対して，$E_{PK_T}(i)$ を復号して得られる値 (i) の回数だけハッシュをかけた値 (K_i)」の提出を TTP へ依頼する.

(4) 調停者は実際にデータ M と鍵 K_i を使って $MAC_{K_i}(H(M))$ を生成し，生成した MAC と署名検証者から提出された MAC を照合する．一致する場合には M に対する署名が正当なものであると判定し，一致しない場合には，署名が偽造されたものであると判定する．

5.4 考察

(1) 証拠生成鍵更新の利点

提案方式では，署名を生成するたびに証拠生成に利用する鍵が進化していくので，想定 4 のような攻撃により秘密鍵を類推する能力を持った攻撃者でも，1 回の IC カードの入出力情報から証拠生成鍵 K_i を推測することは極めて困難である．よって，仮に秘密鍵が盗まれ，署名が偽造された場合であっても，MAC の整合性を検証することにより，署名の偽造を検知することができる．また，鍵が使い捨てになっているおかげで，仮にサイドチャネル攻撃によってその時点で使用された鍵 K_i に関する情報が漏洩したとしても，そこから K_i 以外の情報を得ることは難しい．さらに，シード K は IC カードから削除されているので，ハッシュ関数の一方向性を破らない限り K に関する情報を得ることはできない．

(2) 従来の対策技術との比較

MAC 方式は証拠生成に直接 K を利用するのに対し，提案方式で利用するのは K_i でありシード K は利用されないので，その分 K に対する安全が高まっているといえる．

Forward-Secure 署名や Key-Insulated 署名のような，秘密鍵を定期的に更新する従来の方式の場合，鍵ペアを取り替える必要が生じると，過去に生成した署名の証拠性が失われてしまう．これに対し，我々の方式では，署名と証拠が切り離されているため，鍵ペアを更新しても署名の証拠性は維持される．また，従来の方式は，署名方式に独自のアルゴリズムを用いるしかなく汎用性が低いのに対し，MAC を用いた電子証拠生成技術は，PKI を補完する技術であり実用性が高いといえる．

(3) 今後の拡張

署名を生成するたびに CNT の値を必ず 1 増やす (1 回だけハッシュをかける) のではなく，ハッシュをかける回数に自由度を持たせランダムな値を設定できるようにすれば，より安全性が高まるものと予想される．

6 まとめ

署名生成用秘密鍵の漏洩問題を解決するためには，証拠性の確保が重要である．そこで本稿では，電子証拠物として用いる MAC 生成方法の工夫について提案し考察を行った．今後は，量子コンピュータの脅威に対応した方式へのさらなる拡張や，TTP をまったく利用しない証拠基盤構築方法などについて検討する予定である．

参考文献

[1] M. Bellare, S. Miner,: "A Forward-Secure Digital Signature Scheme", CRYPTO'99, LNCS 1666, pp.431-448, 1999.

[2] D. Boneh, et.al.: "On the Importance of Checking Cryptographic Protocols for Faults", EUROCRYPT'97, LNCS 1233, pp.37-51, 1997.

[3] S. Chokhani, et al.: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, RFC 2527, Mar. 1999.

[4] Y. Dodis, et al.: "Strong Key-Insulated Signature Schemes", PKC2003, to appear.

[5] H. Handschuh, et.al.: "Probing Attacks on Tamper-Resistant Devices", CHES'99, LNCS 1717, pp.303-315, 1999.

[6] R. Housley, et al.: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, RFC 3280, Apr. 2002.

[7] P. Kocher: "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", Advances in Cryptology: CRYPTO'96, LNCS 1109, pp.104-113, 1996.

[8] P. Kocher, et al.: "Differential Power Analysis", CRYPTO'99, LNCS 1666, pp.388-397, 1999.

[9] 小森 旭, 松浦幹太, 須藤 修: 電子商取引における紛争解決のための電子証拠物に関する分析, 2002 年暗号と情報セキュリティシンポジウム予稿集, pp.627-632, 電子情報通信学会, Jan. 2002.

[10] 松本 勉, 岩村 充, 佐々木良一, 松木 武: 暗号ブレイク対応電子署名アリバイ実現機構 (その 1) -コンセプトと概要-, 情報処理学会研究報告, 2000-CSEC-8, pp.13-17, Mar. 2000.

[11] 総務省: 地方公共団体による公的個人認証サービスのあり方検討委員会報告書, Feb. 2002.

[12] 洲崎誠一, 松本 勉: 署名生成機能の危殆化に関する一考察, コンピュータセキュリティシンポジウム 2002 論文集, 情報処理学会, pp.279-284, Oct. 2002.

[13] 高橋知史, 洲崎誠一, 松本 勉: 「Forward-Secure Digital Signature」は役に立つか, 2002 年暗号と情報セキュリティシンポジウム予稿集, pp.837-842, 電子情報通信学会, Jan. 2002.

[14] 谷本, 宮崎, 伊藤, 吉浦: 署名履歴交差を利用したヒステリシス署名の実現方法, コンピュータセキュリティシンポジウム 2002 論文集, 情報処理学会, pp.285-290, Oct. 2002.

[15] 宇根正志: デジタル署名生成用秘密鍵の漏洩を巡る問題とその対策, IMES Discussion Paper Series No.2002-J-32, 日本銀行金融研究所, Oct. 2002.

[16] 宇根正志, 松本 勉: 実行ハードウェア確認タグ付きディジタル署名方式, 情報処理学会研究報告, 2001-CSEC-18, pp.245-252, Jul. 2002.