電子署名アリバイ実現機構——ヒステリシス署名と履歴交差

洲崎誠一[†,††] 松本勉[†]

インターネットのビジネス利用が進むなかで，電子署名技術の重要性が高まっている．電子署名技術は，あるエンティティの電子署名を生成できるのが署名生成鍵を知っている本人だけであるという仮定に基づいている．しかし，署名生成鍵の物理的な盗難や暗号解読技術の進展などといったさまざまな要因によってこのような仮定が成り立たなくなる恐れもある（我々はこれを暗号ブレイクと呼ぶ）．そのような環境下では，不正者は他エンティティの電子署名を容易に偽造することが可能となるため，電子署名技術に期待される相手確認機能や改ざん検知機能が適正に働かなくなってしまう．このような課題に対し，従来よりいくつかの対策技術が提案されているが，それらは我々が想定する暗号ブレイクのすべてをカバーするものではない．そこで，我々は，暗号ブレイクに対処可能な「電子署名アリバイ実現機構」を提案する．本電子署名アリバイ実現機構を利用することにより，各エンティティは，自分が生成した覚えのない電子署名付きメッセージを提示された場合においても，当該電子署名付きメッセージを生成していないこと，すなわち「電子署名アリバイ」を調停者などに証明することができる．本稿では，電子署名アリバイ実現機構の基本的な考え方を示すとともに，その証拠性を高めるために我々が採用した「ヒステリシス署名」と「履歴交差」と呼ぶ2つのコンセプト，ならびにその具体的な実現例について述べる．

Alibi Establishment for Electronic Signatures

Seiichi Susaki[†,††] and Tsutomu Matsumoto[†]

Recently, there is a remarkable tendency oriented toward using electronic signature technology for business. In the electronic signature technology, it is assumed that only the person in question can use a private key of each entity. However, such assumption might not consist of various factors such as the theft of a private key and the progress of various technologies. In such a situation, the attacker can easily forge an electronic signature of another entity. Then, we propose "Alibi establishment mechanism for electronic signatures". We can distinguish valid signature and forged one with the signature log file, which is stored safely in a tamper resistant module. We also propose two technologies, "Hysteresis signature" and "Signature history intercrossing" to be strengthened the evidence of the signature log file.

1. はじめに

1991年のインターネットの商用化以降，キラーアプリケーションであるWWWシステムの登場やコンピュータの低価格化なども手伝って，インターネットの利用者人口は急激に増加している．また，最近では，オンラインショッピングなどインターネットをビジネスに活用する企業も増えており，国内では，2003年度のサービス開始を目指して，電子申請や電子入札などといった各種行政手続きの電子化も進められている[1)~2)]．

このようにインターネットをビジネスに利用する場合，システムの安全性の確保を目的として各種セキュリティ技術が用いられることが多い．電子署名技術は，そのようなセキュリティ技術の1つである．電子署名技術を用いることにより，ネットワークを介して離れたところにいる通信相手の認証や，当該通信相手より受け取った電子データの正当性の確認などといったことが可能となる．電子署名技術を利用するシステムでは，あるエンティティの電子署名を生成できるのは，秘密の署名生成鍵を知っている本人だけであり，それ以外のエンティティには生成できないということを前提としている．そのため，パスワードなどを用いて署名生成鍵の使用を制御したり，ICカードのような耐タンパ性を備えたモジュールに署名生成鍵を格納したりすることで，第三者への署名生成鍵の漏洩を防ぐような方策が採られている．

† 横浜国立大学大学院環境情報学府/研究院
Graduate School of Environment and Information Sciences, Yokohama National University
†† 日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.

しかし，さまざまな要因により，署名生成鍵を知っているのは本人だけであるという仮定が崩れてしまうことも考えられる．

その 1 つの要因は，署名生成鍵，あるいはそれにアクセスするために必要なパスワードなどの盗難である．たとえば，署名生成鍵をパソコンなどに格納し，使用時にパスワードを入力させるようなシステムでは，いわゆる，“*Shoulder Surfing*” によってパスワードを盗み見られた場合などに署名生成鍵を第三者に不正使用される恐れがある．

もう 1 つの要因は，技術の進展である．たとえば，現在利用されている多くの電子署名方式の安全性は，署名検査鍵から署名生成鍵を算出するのに莫大な計算量を要するという仮定のもとに成り立っている．そのため，時間経過にともなうコンピュータの処理能力の向上や解読手法の進展などにより署名検査鍵から署名生成鍵が算出可能となり，その結果として，不正者に電子署名が偽造されてしまうという恐れがある．現在最も広く利用されている電子署名方式である RSA 署名方式では，提案された 1978 年時点では 10 進 80 桁（約 263 ビット）の長さの鍵を利用すれば当面の安全性は保たれるとされ，また，将来に備えては 10 進 200 桁（約 662 ビット）の長さの鍵の利用が推奨されていた[3]．しかし，20 年後の現在では，すでに 10 進 155 桁（512 ビット）の数が素因数分解されるに至っており，10 進 200 桁の数が安全とはいいきれなくなってきている[4]．

このような技術進展にともなう証拠性の喪失は，時間経過と密接な関係にあることは自明である．そのため，電子署名付きメッセージが何らかの効力を有する期間が比較的長期間に及ぶような場合に特に問題となる．たとえば，ある一定期間が経過した後に換金される債券や手形を電子化したシステムなどがその例である．

図 1 は，技術進展にともなって *Alice* の署名生成鍵が *Oscar* に知られてしまった場合に生じる脅威を時間経過に従って表した図である．図 1 において，日時 $T1$ に *Alice* がその時点で安全とされる電子署名方式 G，および署名生成鍵 K を用いて電子署名 S 付き電子債券 M を生成したとする．この時点においては，電子署名 S を生成できるのは署名生成鍵 K の正しい値を知っている *Alice* のみであり，したがって，電子署名 S 付き電子債券 M を生成したのは *Alice* であると考えられる．しかし，その後の技術の進展により，15 年後の日時 $T2$ において，*Alice* の署名生成鍵 K を *Oscar* が手に入れたとする．それにより，*Oscar* は，日時 $T3$ に電子署名方式 G，および署名生成鍵 K を用いて *Alice* の電子署名 S' を偽造して，電子署名 S' 付き電子債券 M' を生成し，それをもって日時 $T4$ に *Alice* に換金を要求することが可能となる．*Alice* としては，生成した覚えのない電子債券を持ち込まれたのだから，即座に換金に応じるとは考えにくい．したがって，日時 $T5$ に何らかの調停作業が発生するものと想定されるが，*Alice* にとって，*Oscar* が *Alice* の署名生成鍵 K を手に入れていること，すなわち持ち込まれた電子債券 M' が *Oscar* によって偽造されたものであることを証明することは現状では困難である．

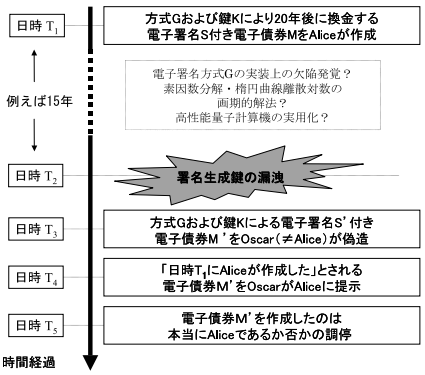

図 1 技術進展にともなう証拠性の喪失
Fig. 1 Loss of evidence by the technological innovation.

我々は，署名生成鍵にアクセスするためのパスワードの漏洩や，電子署名の新たな偽造方法の発見，署名検査鍵からの署名生成鍵の算出などといった，第三者による電子署名の偽造を可能とする事象を「暗号ブレイク」と呼んでいる．

インターネットのビジネス利用の拡大によって電子署名技術が担っている役割の重要性が増している今日において，電子署名の偽造がもたらす影響は計り知れないものがある．これに対し，電子署名の正当性を確認可能とするために，タイムスタンプ[5)~7)] や電子公証[8)~9)]，*Fail-stop Signature*[10)]，*Forward Secure Digital Signature*[11),12)] などといった技術が提案されている．しかしながら，これら従来技術は，我々が想定する「暗号ブレイク」のすべてをカバーするものではない☆．

そこで，我々は，電子社会における証拠性基盤の確立という，より大きな目的を実現するための要素技術の 1 つとして，暗号ブレイクに対処可能な「電子署名

☆ タイムスタンプは，本来，ある日時に当該電子データが存在していたことを証明するための技術であるが，後述のように電子署名偽造対策としても利用可能である．

アリバイ実現機構」を提案する．電子署名アリバイとは，過去における電子署名の非生成を証明することである．アリバイとは，元来，犯罪などの事件が発生したときに被疑者がその場にいなかったという「現場不在証明」のことであるが，我々は,「電子署名非生成証明」の意味も担わせる．

本稿では，2 章において，電子署名を偽造される要因や従来の電子署名偽造対策などについて整理する．3 章では，我々が提案する電子署名アリバイ実現機構の基本アイデアを説明するとともに，ヒステリシス署名と履歴交差，と呼ぶ電子署名アリバイ実現機構の証拠性をより確かなものにするための 2 つのコンセプトについて述べる．さらに，4 章において，電子署名アリバイ実現機構の具体的な実装方式例を示し，5 章で提案方式の評価を行う☆．

2. 電子署名の偽造

2.1 定　　義

通常，署名生成者 *Alice* が生成した（と思われる）電子署名付きメッセージを受け取った場合，署名検証者 *Bob* は，当該電子署名付きメッセージと，信頼できるオーソリティが発行した *Alice* の公開鍵証明書（署名検査鍵）とを用い，自己の署名検証プログラムを使って署名検証処理を行い，受け取った電子署名付きメッセージの正当性を確認する．その際，*Alice* の公開鍵証明書が失効されていないかどうかの確認も併せて行う．

これに対し，電子署名付きメッセージが何らかの効果を有する期間が，当該電子署名付きメッセージの生成時に使用した公開鍵証明書の有効期間より長い場合も考えられる．前述の電子債券はその一例である．

以上のようなことをふまえ，本稿では,「電子署名の偽造」を以下のように定義する．

Alice とは異なる署名偽造者 _Oscar_ が生成した電子署名付きメッセージに対し，_Bob_ が，当該電子署名付きメッセージ生成時点で有効であった _Alice_ の公開鍵証明書を用いて署名検証を行った結果，正当なものであると判定された場合，_Oscar_ は _Alice_ の電子署名の偽造に成功したという☆☆．

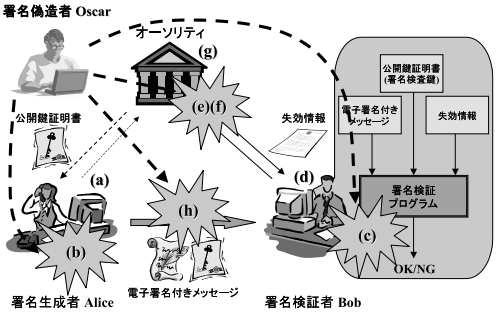

図 2 電子署名を偽造される要因
Fig. 2 Factors which electronic signatures are forged.

さて，前述のような電子署名の利用環境においては，「電子署名の偽造」以外にも，

- *Alice* が自己の署名生成鍵を故意に暴露し，自己が生成した電子署名付きメッセージであるにもかかわらず，その事実を否認する，
- 過去のある日時 T において存在しなかった電子署名付きメッセージを，*Alice* が事後に生成し，日時 T に生成した電子署名付きメッセージであると主張する，

などといった不正行為を署名生成者 *Alice* 自身が行うことも考えられる．これらについては，5 章で考察する．

2.2 電子署名を偽造される要因

Oscar によって *Alice* の電子署名が偽造される主たる要因を以下に示す（**図 2** 参照）．

(1) 署名生成者 *Alice* に問題がある場合
  (a) すでに安全性が確保できないことが分かっている電子署名方式を使用したり，短い鍵長の署名生成鍵を使用したりする．
  (b) 運用も含めた鍵の管理方法に問題があり，物理的手段によって *Oscar* に署名生成鍵を盗まれてしまう．

(2) 署名検証者 *Bob* に問題がある場合
  (c) *Bob* の使用している署名検証プログラムにバグやデバッグ用のバイパスルートが残っていたり，*Oscar* によって不正なものに置き換えられていたりするなど，署名検証プログラムが *Bob* の意図したとおりに正しく機能していない．
  (d) 失効情報を確認しないなど，*Bob* が署名検証で手抜きをする．

(3) オーソリティに問題がある場合
  (e) 運用も含めた鍵の管理方法に問題があり，物理的手段によって *Oscar* に署名生成鍵を盗まれてしまう（*Oscar* が公開鍵証明

☆ 我々は，文献 13)〜15) において，電子署名アリバイ実現機構の概略を紹介しているが，本稿は，それらをベースとしてさらなる検討を加えるとともに，定性的な評価を行ったものである．

☆☆ 公開鍵証明書の有効期限後に電子署名付きメッセージの確認を行うときにすでに失効情報が失われており，当該公開鍵証明書が電子署名生成時に有効であったか否か確認できない場合も想定され，これに対する対策も検討されているが[16),17)]，本稿ではそのような状態にはない場合のみを考える．

書を偽造可能となる）．

(f) *Alice* の公開鍵証明書発行時の本人確認方法に問題があり，*Oscar* に対して *Alice* の公開鍵証明書（対応する署名生成鍵は *Oscar* が持っている）を発行してしまう．

(g) 各エンティティに対して失効情報をきちんと公開しない．

(4) 電子署名方式に問題がある場合

(h) コンピュータの処理能力の向上や新しい解読方法の考案などにより，*Oscar* が *Alice* の電子署名を容易に偽造可能となる（それまで安全だとされていた電子署名方式が安全ではなくなる）．

なお，*Oscar* がランダムに生成した電子署名付きメッセージが，偶々署名検証処理を合格する場合なども考えられるが，この問題については本稿では扱わないものとする．

上記要因の中で，要因 (a)，(d)，(g) については，*Oscar* による意図的な攻撃ではないため，各エンティティがそのようなことをしないように注意すれば十分防ぐことができる．その他の要因は，*Oscar* の意図的な攻撃によるものである．そこで，そのような攻撃が行われる可能性について考える．個人とオーソリティとを比較すると，一般にオーソリティ（特に多くのエンティティから信頼を受けているオーソリティ）のほうが安全性や信頼性に対する感度は高い．したがって，*Oscar* が攻撃を行う場合，オーソリティに対する攻撃（要因 (e)，(f)）よりも個人に対する攻撃（要因 (b)，(c)）のほうが成功する確率が高いものと想定される．一方，攻撃が成功したときの影響度を考えてみると，要因 (b)，(f) は，基本的には 1 人のエンティティだけが被害を受けるだけである．それに対し，要因 (c) は複数のエンティティが被害を受ける恐れがある．また，要因 (e)，(h) に至ってはすべてのエンティティが被害を受けることになる．通常，*Oscar* は，電子署名の偽造が容易である，あるいは，電子署名の偽造によって大きな影響がでるような攻撃を行うものと考えられることから，要因 (b)，(c)，(h) に起因する電子署名の偽造に対する対策がより重要と考える．ただし，要因 (c) に関しては，他の正しい署名検証プログラムを利用すれば回避できる問題であり，以降では要因 (b)，(h) を中心に考察を加える．

2.3 従来の電子署名偽造対策

電子署名偽造対策として，電子署名の偽造が起こらないようにするというアプローチが考えられる．たとえば，電子署名付きメッセージや署名検査鍵などから署名生成鍵を算出することが情報量的に困難であるようにすることで，コンピュータの処理能力が向上したとしても安全性が損なわれないような署名方式も提案されている[18]．しかし，そのような署名方式を利用した場合においても，*Oscar* が *Alice* の署名生成鍵を盗みだし，それを用いて電子署名を偽造するという不正は十分起こりうることである．

一方，電子署名が偽造された場合にその被害を最小限に食い止めるという考え方に基づく対策も提案されている．以降では，それら従来方式の概要ならびに問題点について述べる．

2.3.1 タイムスタンプ

(1) 概要

Alice が通常の電子署名方式で生成した電子署名付きメッセージ，またはそのハッシュ値をオーソリティに送付すると，オーソリティはタイムスタンプを付加して返送するとともに，その履歴を安全に保管しておく．これにより，タイムスタンプが正しく付加されており，当該タイムスタンプに関する情報がオーソリティの保管する履歴の中に含まれているか否かで，正当な電子署名付きメッセージと偽造されたものとを事後になっても区別できる（要因 (b)，(h) の対策）．

(2) 問題点

- 正当な電子署名付きメッセージと偽造されたものとを区別できるようにするためには，電子署名付きメッセージを生成したら必ずオーソリティにタイムスタンプを押してもらわなければならない．
- タイムスタンプを押す過程において不正が行われる恐れがある．すなわち，オーソリティに送られてきた電子署名付きメッセージ自体がすでに偽造されたものである場合，偽造されたものにタイムスタンプを押してしまう恐れがある．
- オーソリティへの負荷集中などによってリアルタイムに処理されない（サービスを受けたいときに受けられない）恐れがある．

2.3.2 電子公証☆

(1) 概要

Alice が通常の電子署名方式で生成した電子署名付きメッセージをオーソリティに送付すると，オーソリティは内容を確認した後，当該電子署名付きメッセージに副署して返送するとともに，その履歴を安全に保管しておく．これにより，電子署名付きメッセージに正しく副署されており，当該電子署名付きメッセージに関する情報がオーソリティの保管する履歴の中に含

☆ 現行の公証業務の「私署証書の認証」に相当する．

まれているか否かで，正当な電子署名付きメッセージと偽造されたものとを事後になっても区別できる（要因 (b)，(h) の対策）．

(2) 問題点

- 正当な電子署名付きメッセージと偽造されたものとを区別できるようにするためには，電子署名付きメッセージを生成したら必ずオーソリティに公証してもらわなければならない．
- 公証を依頼する過程において不正が行われる恐れがある．すなわち，オーソリティに送られてきた電子署名付きメッセージ自体がすでに偽造されたものである場合，偽造されたものを公証してしまう恐れがある．
- オーソリティへの負荷集中などによってリアルタイムに処理されない（サービスを受けたいときに受けられない）恐れがある．

2.3.3 Fail-stop Signature

(1) 概要

Oscar が，*Alice* の電子署名を偽造するに足る膨大なコンピュータパワーを利用可能となった場合（要因 (h)）に対する対策であり，メッセージとそれに対応する電子署名付きメッセージが与えられた場合に，*Alice* の署名生成鍵と同値となるものが複数個存在するようにする．これにより，*Oscar* によって電子署名が偽造された場合に，それが偽造されたものであることを *Alice* が高い確率で証明する（*Alice* の正しい署名生成鍵と *Oscar* によって算出されたものとが不一致であることを示す）ことができるようにする．

(2) 問題点

- *Oscar* が物理的手段によって署名生成鍵を盗み出した場合（要因 (b)），*Alice* の電子署名を偽造することができてしまう．さらに，*Alice* はそれが偽造されたものであることを証明できない．
- 基本的に *One-Time Signature* であり，毎回署名生成鍵を生成しなければならず．通常の電子署名方式と比較して効率が良くない．

2.3.4 Forward-Secure Digital Signature

(1) 概要

Oscar により，*Alice* の署名生成鍵を物理的手段で盗まれた場合（要因 (b)）に対する対策であり，署名生成鍵を短い期間（たとえば 1 日ごと）で更新する（n 番目の鍵から $n+1$ 番目の鍵を生成する）ことで，ある日時 T における *Alice* の署名生成鍵を *Oscar* によって盗まれたとしても，それを用いて日時 T 以前の *Alice* の電子署名付きメッセージを生成することができないようにする．

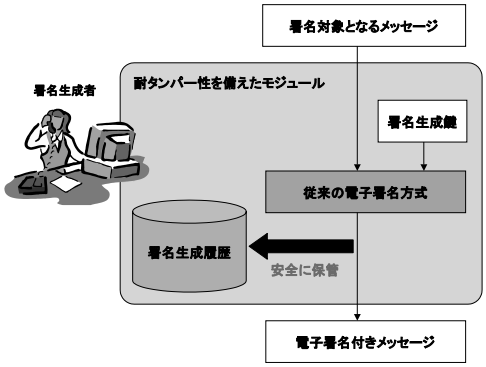

図 3 基本アイデア
Fig. 3 Basic idea of alibi establishment for electronic signatures.

(2) 問題点

- *Oscar* が鍵更新ルールを知っている場合，日時 T 以降（*Alice* が当該署名生成鍵の使用を止めるまでの間）の電子署名付きメッセージならば容易に偽造できてしまう．
- コンピュータの処理能力の向上や新しい解読方法の考案などにより，*Oscar* が *Alice* の電子署名を容易に偽造可能となった場合（要因 (h)），日時 T 以前の *Alice* の電子署名を偽造することができてしまう．
- 署名生成鍵や署名検査鍵，あるいは電子署名付きメッセージなどのビット長が，通常の電子署名方式と比較して長くなってしまい効率が良くない．

3. 電子署名アリバイ実現機構

2.3 節で述べたように，従来の電子署名偽造対策は，我々が想定している暗号ブレイクに対する対策として必ずしも十分ではない．本章では我々が提案する電子署名アリバイ実現機構について説明する．

3.1 基本アイデア

電子署名アリバイ実現機構は，各エンティティが自分の署名生成履歴を本人自身にも偽造困難な形で安全に保管することで，当該エンティティが生成した電子署名は署名生成履歴に含まれているものだけであり，それ以外は不正者によって偽造された電子署名である，ということを調停者に証明可能とする仕組みである（**図 3** 参照）．

本方式では，当該電子署名付きメッセージに関するデータが署名生成履歴に含まれているか否かで，正当な電子署名付きメッセージと偽造されたものとを事後になっても区別できるようにしている．すなわち，たとえ同じ署名生成鍵を使って生成された電子署名で

あっても，当該エンティティの署名生成履歴に含まれていないものは偽造されたものだと判別できるようになっている．したがって，本方式は，2.2 節で示した要因 (b)，(h) の両方に対処することができる．

前記基本アイデアに基づくシステムを構築するうえで最も重要なことは，各エンティティの署名生成履歴の完全性（署名生成履歴に含まれる個々の署名生成記録の追加や削除などが事後になって行われていないこと）をいかに保証するかということである．このような課題に対し，電子署名アリバイ実現機構では，署名生成履歴を耐タンパ性を備えたモジュール（IC カードなど）に格納するようにする．

ただし，保管されている署名生成履歴を用いて電子署名付きメッセージの正当性を確認する場合や，IC カードなどの耐タンパモジュールの記憶容量に制限があり，内部に保管しきれなくなった場合，単純に署名生成履歴を保管しておくだけでは，外部出力された署名生成履歴に対して何らかの改ざんが加えられた場合にその事実を検出することが困難である．

そこで，我々は，署名生成履歴の改ざんを著しく困難にする手法として，新たに「ヒステリシス署名」と「履歴交差」という 2 つのコンセプトを提案・採用する．

3.2 ヒステリシス署名

3.2.1 コンセプト

我々が提案するヒステリシス署名とは，**図 4** に示すように，RSA 署名や DSA 署名のような従来の電子署名方式を構成要素の一部として利用する電子署名方式の一形態であり☆，あるエンティティが署名対象となるメッセージに電子署名を施す際に，当該エンティティのそれ以前の署名生成履歴などといったヒステリシス情報を電子署名付きメッセージの中に綴り込んでいくものである．

3.2.2 ヒステリシス署名生成/検査

(1) ヒステリシス署名生成処理

ヒステリシス署名生成を行うエンティティは，署名対象となるメッセージ M と，その時点の署名生成履歴 H_{T-1} との組に対して，自己の署名生成鍵 K_s を用いて従来の署名生成処理を行い，ヒステリシス署名付きメッセージ $S = Sign_{K_s}(M \| H_{T-1})$ を生成するとともに，署名生成履歴の更新処理 $H_T = H_{T-1} \| S$ を行う．

☆ 電子署名方式には，メッセージ復元型（署名対象メッセージを署名生成鍵で直接変換する方式）と署名付加型（署名対象メッセージのハッシュ値を署名生成鍵で変換したのち，その結果を元の署名対象メッセージと連結する方式）の 2 種類があるが，ヒステリシス署名は，いずれの電子署名方式を用いて実現可能である．

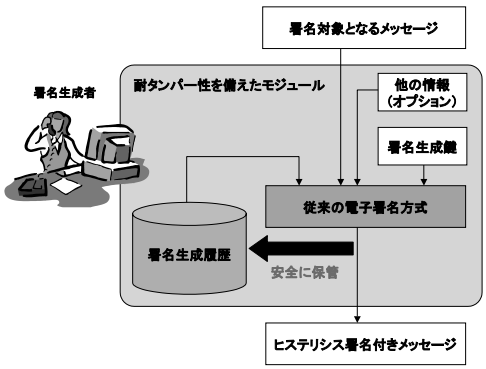

図 4 ヒステリシス署名
Fig. 4 Hysteresis signature scheme.

(2) ヒステリシス署名検査処理

ヒステリシス署名検査を行うエンティティは，ヒステリシス署名付きメッセージ S に対して，署名生成エンティティの公開鍵証明書（署名検査鍵 K_s を含む）を用いて従来の署名検査処理 $True/False = Verify_{K_v}(S)$ を行う．

また，何らかのトラブル（たとえば，生成した覚えのないヒステリシス署名付きメッセージ S' が持ち込まれるなど）が生じた場合には，調停者などが署名生成履歴の整合性照合処理 $True/False = Check(S', H_{T+n})$ $(n \geq 0)$ を行い，ヒステリシス署名付きメッセージ S' の有効性を確認する．ここでの整合性照合処理とは，ヒステリシス署名付きメッセージ S' を生成したとされるエンティティの署名生成履歴 H_{T+n}) の中に，当該ヒステリシス署名付きメッセージに関する情報があるかどうかを確認する処理である．

3.2.3 ヒステリシス署名がもたらす効果

Oscar が「過去のある時刻 T に生成したとされる *Alice* のヒステリシス署名付きメッセージ」を偽造するためには，*Alice* の署名生成鍵を用いて単純に署名を偽造するだけでは十分ではなく，時刻 T 以前の *Alice* の署名生成履歴をきちんと反映したヒステリシス署名付きメッセージを偽造しなければならない．さらに，時刻 T 以降の *Alice* の署名生成履歴との整合性がとれていることも必要となる．

このようにヒステリシス署名は，当該エンティティが生成した個々の電子署名の現在に至るまでの順序関係を明らかにすることで，それ以外の電子署名を生成していないこと，すなわち電子署名アリバイを調停者に対して証明可能とするものである．

3.3 履歴交差

履歴交差とは，署名生成履歴の証明力を向上させる手段であり，**図 5** に示すように，あるエンティティの

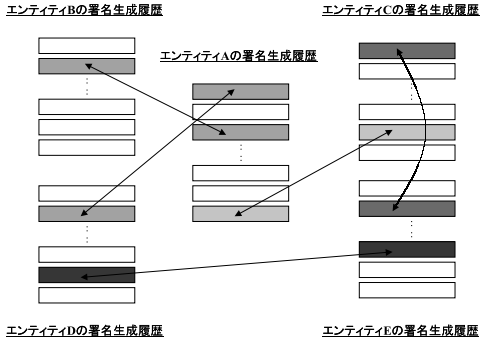

図 5 履歴交差
Fig. 5 Signature history intercrossing scheme.

署名生成履歴と別のエンティティの署名生成履歴とを綴り合せることである．

エンティティ間で履歴交差が繰り返し行われると，各エンティティの署名生成履歴が他エンティティの署名生成履歴の中に反映されることになり，

- 「過去の電子署名」の偽造には，複数のエンティティの署名生成履歴の整合的な偽造が必要となり，不正のための作業量が鼠算的に増える，
- そうした偽造作業に複数のエンティティを巻き込まなくてはならず，何らかの相互抑制効果が働くものと考えられる，

といった効果が得られるため，署名生成履歴の証明力は飛躍的に向上する．

4. 電子署名アリバイ実現機構の具体的な実現方式

本章では，ヒステリシス署名と履歴交差を利用した電子署名アリバイ実現機構の具体的な実現方式の一例を示す．

4.1 表 記 法

$Sign()$：従来の電子署名方式における署名生成処理．
$Verify()$：従来の電子署名方式における署名検査処理．
$h()$：一方向性ハッシュ関数．
$A\|B$：2 つのデータ A，B を連結したデータ．
K_s：*Alice* の署名生成鍵
K_v：*Alice* の署名検査鍵
n：*Alice* がヒステリシス署名生成を行った回数．
IV：初期値．
M_n：n 番目の署名対象メッセージ．
S_n：n 番目のヒステリシス署名付きメッセージ．
R_n：n 番目のヒステリシス署名生成記録．
H_n：n 回目のヒステリシス署名生成を行った後の署名生成履歴（1 回目から n 回目までのヒステリシス署名生成記録を連結したデータ）．

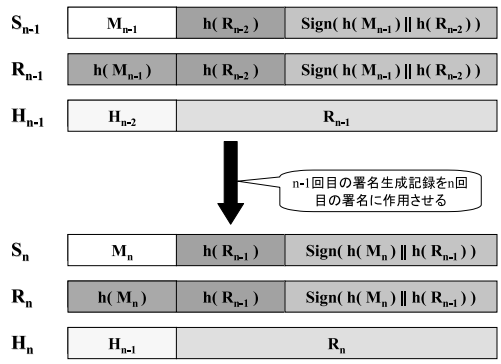

図 6 チェイニング署名
Fig. 6 Chaining signature scheme.

4.2 チェイニング署名

ヒステリシス署名の具体的な実現方式の一例としてチェイニング署名を説明する（**図 6** 参照）．チェイニング署名は，n 回目のヒステリシス署名生成時に $n-1$ 回目のヒステリシス署名生成記録 R_{n-1} のハッシュ値を作用させる署名方式である．

Alice が n 番目のヒステリシス署名付きメッセージ S_n を生成するときの具体的手順，ならびに *Bob* が当該ヒステリシス署名付きメッセージを検査するときの具体的手順を以下に示す．

フェーズ 1：ヒステリシス署名生成

(1) 署名対象メッセージ M_n のハッシュ値 $h(M_n)$ を算出する．
(2) 保存してある署名生成履歴 H_{n-1} に含まれる最新の署名生成記録 R_{n-1} のハッシュ値 $h(R_{n-1})$ を算出する．ただし，1 回目のヒステリシス署名生成処理においては，以降の手順でハッシュ値 $h(R_{n-1})$ の代わりに初期値 IV を用いる．
(3) (1)，(2) で算出した 2 つのハッシュ値を連結したデータ $h(M_n)\|h(R_{n-1})$ に対して，署名生成鍵 Ks を用いて従来の署名生成処理を行い，電子署名付きメッセージ $Sign_{K_s}(h(M_n)\|h(R_{n-1}))$ を生成する．
(4) 署名対象メッセージ M_n，最新の署名生成記録のハッシュ値 $h(R_{n-1})$，および電子署名付きメッセージ $Sign_{K_s}(h(M_n\|h(R_{n-1}))$ を連結し，ヒステリシス署名付きメッセージ

$$S_n = M_n\|h(R_{n-1}) \|Sign_{K_s}(h(M_n)\|h(R_{n-1}))$$

を生成する．

フェーズ 2：署名生成履歴の更新

(5) 2つのハッシュ値 $h(M_n)$，$h(R_{n-1})$ と電子署名付きメッセージ $Sign_{K_s}(h(M_n)\|h(R_{n-1}))$ とを連結し，署名生成記録

$$R_n = h(M_n)\|h(R_{n-1}) \|Sign_{K_s}(h(M_n)\|h(R_{n-1}))$$

を生成する．

(6) 保存してある署名生成履歴 H_{n-1} と署名生成記録 R_n とを連結し，署名生成履歴

$$H_n = H_{n-1}\|R_n$$

を生成して保存する．

フェーズ 3：ヒステリシス署名検証

(7) ヒステリシス署名付きメッセージ S_n に含まれる署名対象メッセージ M_n のハッシュ値 $h(M_n)$ を算出する．

(8) (7)で算出したハッシュ値 $h(M_n)$ と，ヒステリシス署名付きメッセージ S_n に含まれるハッシュ値 $h(R_{n-1})$ および電子署名付きメッセージ $Sign_{K_s}(h(M_n)\|h(R_{n-1}))$ と，*Alice* の公開鍵証明書に含まれる署名検査鍵 K_v とを用いて従来の署名検査処理

$$h(M_n)\|h(R_{n-1}) \stackrel{?}{=} Verify_{K_v}(Sign_{K_s}(h(M_n)\|h(R_{n-1}))$$

を行う．

暗号ブレイク後に調停者などがヒステリシス署名付きメッセージ

$$S_m = M_m\|h(R_{m-1}) \|Sign_{K_s}(h(M_m)\|h(R_{m-1}))\ (1\leq m\leq n)$$

の真偽を判定する場合には，上記ヒステリシス署名検証手順だけでは不十分である．なぜなら，すでに暗号ブレイクしているため，上記手順によって検証 OK となるような署名は，正当な署名者である *Alice* 以外にも生成できるからである．したがって，そのような場合には，*Alice* が保存している署名生成履歴 H_n の中に当該ヒステリシス署名付きメッセージに対応する署名生成記録

$$R_m = h(M_m)\|h(R_{m-1}) \|Sign_{K_s}(h(M_m)\|h(R_{m-1}))$$

が含まれているかどうか，さらには，以下の手順により，署名生成履歴 H_n の整合性がきちんと保たれているかということも確認する必要がある．

フェーズ 4：署名生成履歴の整合性照合

(9) ヒステリシス署名付きメッセージ S_m に対して，署名生成履歴 H_n に含まれる署名生成記録 R_m と *Alice* の署名検査鍵 K_v とを用いて従来の署名検査処理を行う．

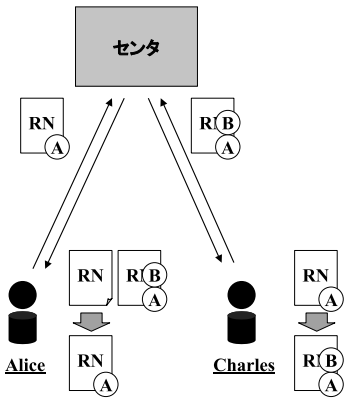

図 7 履歴交差プロトコル
Fig. 7 Protocols of signature history intercrossing.

(10) 署名生成履歴 H_n に含まれる署名生成記録 R_{m-1} のハッシュ値 $h(R_{m-1})$ を算出する．

(11) 署名生成記録 R_m の中のハッシュ値 $h(R_{m-1})$ と，(10)で算出したハッシュ値 $h(R_{m-1})$ が同じ値であるかどうかを確認する（前の署名生成記録との整合性照合）．

(12) 署名生成記録 R_m のハッシュ値 $h(R_m)$ を算出する．

(13) 署名生成履歴 H_n に含まれる署名生成記録 R_{m+1} の中のハッシュ値 $h(R_m)$ と，(12)で算出したハッシュ値 $h(R_m)$ が同じ値であるかどうかを確認する（後の署名生成記録との整合性照合）．

4.3 履歴交差プロトコル

図 7 は，センタを利用した履歴交差プロトコルの一例である．*Alice* と *Charles* との間で履歴交差を行う場合には以下の手順に従って行う．

(1) *Alice* は，乱数 RN を生成し，保存してある最新の署名生成記録 R_{n-1} や署名生成鍵 Ks などを用いて，ヒステリシス署名付き乱数

$$S_n = RN\|h(R_{n-1}) \|Sign_{K_s}(h(RN)\|h(R_{n-1}))$$

を生成したのち，当該データに対する署名生成記録

$$R_n = h(RN)\|h(R_{n-1}) \|Sign_{K_s}(h(RN)\|h(R_{n-1}))$$

を自分の署名生成履歴 H_n に含ませる．

(2) *Alice* は，ヒステリシス署名付き乱数 S_n をセンタに送付する．

(3) センタは，ランダムに選択した *Charles* に，*Alice* から送られてきたヒステリシス署名付き乱数 S_n を送付する．

(4) *Charles* は，センタから送られてきたヒステリシス署名付き乱数 S_n に対して保存してある最新の署名生成記録 ${R_{n-1}}'$ や署名生成鍵 Ks' などを用いて二者ヒステリシス署名付き乱数（*Alice* と *Charles* の両方のヒステリシス署名がついた乱数）

$$S_n' = S_n \| h({R_{n-1}}') \| Sign_{K_s'}(h(S_n) \| h({R_{n-1}}'))$$

を生成したのち，当該データに対する署名生成記録

$$R_n' = h(S_n) \| h({R_{n-1}}') \| Sign_{K_s'}(h(S_n) \| h({R_{n-1}}'))$$

を自分の署名生成履歴 ${H_n}'$ に含ませる．

(5) *Charles* は，二者ヒステリシス署名付き乱数 S_n' をセンタに返送する．

(6) センタは，*Charles* から返送されてきた二者ヒステリシス署名付き乱数 S_n' を，*Alice* に送付する．

上記手順において，センタを利用する目的は，エンティティ間の結託を防ぐためである．したがって，エンティティ間の結託について考慮する必要がない場合には，エンティティだけで行ってもよい．

このような履歴交差はエンティティが署名するたびに行うものではなくある程度定期的に行えばよいものであって，リアルタイム性は要求されない．また，ここでのセンタはデータ転送の仲介者としての役割のみを果たすものであり，特別な処理を行う必要がない．そのため，負荷集中などの問題はある程度防ぐことができるものと考える．

ところで，履歴交差において特筆すべきは，他のエンティティのために行った行為が自らの利益につながるといった特質を備えていることである．すなわち，履歴交差を行うことで，*Alice* の署名生成履歴の一部が *Charles* の署名生成履歴の中に取り込まれるだけでなく，*Charles* の署名生成履歴の一部が *Alice* に送られることにもなり，*Charles* の署名生成履歴の証拠性も著しく向上する．

5. 考　　察

5.1 安全性に関する評価

3 章で述べたように，我々が提案する電子署名アリバイ実現機構では，電子署名付きメッセージに関するデータが，当該電子署名付きメッセージを生成したとされるエンティティの署名生成履歴に含まれているか否かで，正当な電子署名付きメッセージと偽造されたものとを判別する．そのため，耐タンパ性を備えたモジュールに署名生成履歴を格納するとともに，ヒステリシス署名や履歴交差を併用することでその完全性を確保するようにしている．以下で，ヒステリシス署名や履歴交差の安全性についてより詳細に考察する．

5.1.1 チェイニング署名

具体的な実現方式であるチェイニング署名により，ヒステリシス署名の安全性を検討する．

チェイニング署名において，ハッシュ関数の一方向性，すなわち，$h(x)$ が与えられているときに，$h(y) = h(x)$ を満足するような y を算出することが困難である，という性質が保たれているならば，4.2 節のフェーズ 4 に関して次の命題が成り立つ．

[命題]

4.2 節のフェーズ 4 の手順によって，署名生成記録 R_n の整合性が確認されたとき，R_{n+1} が偽造された署名生成記録でなければ R_n も偽造された署名生成記録ではない．ただし，「R_i が偽造された署名生成記録である」とは，署名生成記録 R_i の整合性がフェーズ 4 の手順で確認され，かつ，署名生成記録 R_i の中の，署名対象メッセージのハッシュ値 $h(M_i)$ があるべき位置に，正しい署名対象メッセージ M_i のハッシュ値とは異なるデータがあることとする．

[証明]

R_{n+1} が偽造された署名生成記録ではなく，かつ ${R_n}'$ が偽造された署名生成記録であると仮定する．本来の署名対象メッセージを M_n とすると，それに対応する署名生成記録 R_n は

$$R_n = h(M_n) \| h(R_{n-1}) \| Sign_{K_s}(h(M_n) \| h(R_{n-1}))$$

である．R_{n+1} は偽造された署名生成記録ではないので，署名対象メッセージ M_{n+1} と署名生成履歴 R_n とを用いて

$$R_{n+1} = h(M_{n+1}) \| h(R_n) \| Sign_{K_s}(h(M_{n+1}) \| h(R_n))$$

と表すことができる．一方，${R_n}'$ は偽造された署名生成記録なので，署名生成記録 ${R_n}'$ は，フェーズ 4 の手順を満足し（署名生成記録 R_{n+1} に含まれるハッシュ値 $h(R_n)$ と，偽造された署名生成記録 ${R_n}'$ から算出したハッシュ値 $h({R_n}')$ とが一致し），かつ，ハッシュ値 $h(M_n)$ があるべき位置に異なるデータ $h(M_n)'$ を含んでいることになる．すなわち，$h(R_n) = h({R_n}')$，かつ，$R_n \neq {R_n}'$ ということなり，これは，ハッシュ関数 h の一方向性に反する．したがって，R_{n+1} が偽造された署名生成記録でなければ R_n も偽造された署名生成記録ではない．

[**系**]
任意の $n(>m)$ に対し，署名生成記録 R_n が偽造されたものではなく，かつ，すべての $m \leq i < n$ について，署名生成記録 R_i がフェーズ 4 の手順を満足すれば，R_m は偽造された署名生成記録ではない．

[**証明**]
命題を繰り返し適用すればよい．

この系から，確かにあるエンティティによって生成されたことが分かっているヒステリシス署名付きメッセージに対応する署名生成記録を基準とし，そこから整合性を保ったまま過去の方向に遡ることができる範囲にある署名生成記録に対応するヒステリシス署名付きメッセージは，すべて当該エンティティによって生成されたことが証明される．

一方，これとは逆に「あるヒステリシス署名付きメッセージが，自分が過去に生成したものではないこと（後になって他人に偽造されたものであること）」を示すためには,「当該エンティティがそれ以前に生成した個々のヒステリシス署名付きメッセージに対応する署名生成記録は，すべて現在保存されている署名生成履歴の中に含まれている」ということを示すことができればよい．具体的には，各エンティティは自分の署名生成履歴を調停者に提出し，その中に当該ヒステリシス署名付きメッセージに対応する署名生成記録がないことを示すことにより，自分が生成したものではないということを証明できる．

以上をまとめると，チェイニング署名を用いて電子署名アリバイを実現するためには，以下の要件を満足していればよい．

(a) 各エンティティの署名生成履歴が欠損のない完全な状態で保存されている．
(b) 各エンティティの署名生成履歴に含まれる最新の署名生成記録について，その正当性（確かに本人が生成したヒステリシス署名付きメッセージに対応する署名生成記録であること）が確認可能である．
(c) 各エンティティが生成したすべてのヒステリシス署名付きメッセージに対応する署名生成記録が署名生成履歴の中に存在している (利用者は署名生成履歴に残らないような方法で署名できない）ことが保証されている．

上記要件 (a) は，適切な記録媒体を使ったデータのバックアップなど，通常のデータ保管技術によって実現可能である．また，プライバシや機密保護の観点からみて特に問題がなければ，オーソリティに預けるようにしてもよい（この処理はリアルタイムに行う必要がないものであり，またこのようにしても電子署名アリバイ実現機構の安全性には何ら影響しない）．

また，上記要件 (b)，(c) を満足する手段の 1 つとして，3 章でも述べたように耐タンパ性を備えたハードウェア，またはソフトウェア・モジュールの利用が考えられる．そのようなモジュールに求められる機能を以下に示す．

- 各エンティティの署名生成鍵はモジュール内で生成され，外部からは読み出せない（エンティティ自身も自分の署名生成鍵の正しい値を知らない）．
- 署名した場合には，その記録が最新の署名生成記録として必ずモジュール内に残る．
- モジュール内に保存されている最新の署名生成記録は外部から書き換えたり，消去したりできない．
- 必要に応じてモジュール内に保存された最新の署名生成記録を確認（読み出し）できる．
- 新たに署名を行う場合には必ずその時点での最新の署名生成記録が使用される．

さて，2.1 節で署名生成者自身が行う不正行為の可能性について言及したが，ヒステリシス署名（チェイニング署名）技術を利用することにより，署名生成者自身であっても過去に遡って電子署名を偽造することが困難であることは自明である．また，故意にモジュールを紛失するなどといった不正も考えられるが，履歴交差を行っていれば，当該エンティティの署名生成履歴に関する情報が他のエンティティの署名生成履歴の中にも部分的に含まれているので，それらを参照し，前後の整合性を確認することなどより，何らかの証拠となるデータを再構築することができる可能性もある．

ただし，エンティティが署名偽造者と同様の手段，たとえば，電子署名付きメッセージや署名検査鍵から署名生成鍵を算出するなどの手段によって自分自身の署名生成鍵の正しい値を知った場合，その署名生成鍵を用いて，モジュール外部でヒステリシス署名付きメッセージを生成可能となる．すなわち，当該ヒステリシス署名付きメッセージに関するデータは当然署名生成履歴に反映されないので，当該エンティティは二重帳簿的な不正を行うことが可能であり，その結果として，ヒステリシス署名付きメッセージを受け取った署名検証者が不利益を被る恐れがある．

5.1.2 履歴交差と信頼ポイント

履歴交差がヒステリシス署名の証拠性を高める手段であることはすでに述べたが，この履歴交差はそれ以外にも有用な特徴を持っている．

5.1.1 項で電子署名アリバイを実現するためには，各エンティティの署名生成履歴が欠損のない完全な状態

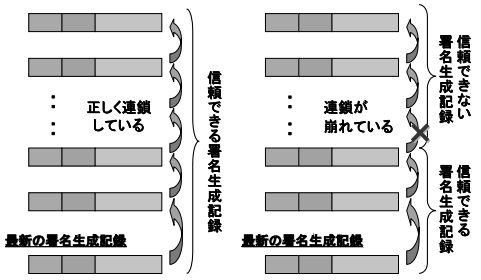

図 8 署名生成記録の欠落による証拠性の喪失
Fig. 8 Loss of evidence because signature history is damaged.

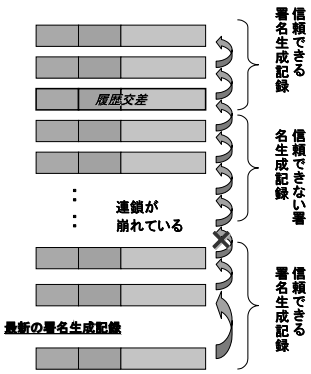

図 9 信頼ポイント
Fig. 9 Addition of trust point using signature history intercrossing.

で保存されていることが必要だと述べたが，いくら安全な状態で保存しておいたとしても，ハードウェアの故障なども含め，何らかの原因でデータの一部が欠落してしまう恐れはつねに存在する．チェイニング署名を利用している環境においてそのようなことが起きた場合，**図 8** に示すとおり，1 カ所の署名生成記録の欠落が，それ以前のすべての署名生成記録の検証を（欠落せずに残っていたとしても）不可能としてしまうことになる．

そのための対策として，信頼ポイントの追加ということが考えられる．信頼ポイントとは，その署名生成記録に対応するヒステリシス署名付きメッセージが，確かに当該利用者によって生成されたということが保証された署名生成記録のことである．このような信頼ポイントを適宜追加しておくことにより，欠落した署名生成記録以前に署名されたものの真偽も検証可能になる（**図 9** 参照）．

そのような信頼ポイントを追加するためには，オーソリティに署名生成履歴の一部を預託する方法などが考えられるが，システムを利用するエンティティどうしが協力し合うことによってシステム全体の信頼性を高めていく履歴交差も信頼ポイントを追加するための有効な方法の 1 つである．

5.1.3 ハッシュ関数のブレイク

「ハッシュ関数のブレイク」とは，以下に示すような 4 つの場合のいずれかの状態になることだとする．

(a) ハッシュ値が等しくなるような 1 組のメッセージが見つかる（衝突が 1 つ見つかる）．
(b) 与えられたハッシュ値になるようなメッセージを 1 つ見つけることができる（一方向性が崩れる）．
(c) 与えられたハッシュ値を与えるようなメッセージをいくつでも見つけることができる．
(d) 与えられたハッシュ値を与えるような意味のある（都合のよい）メッセージを見つけることができる．

これらのうち，(a) のようなブレイクは，本稿で提案しているヒステリシス署名にとって脅威とはならない．(b) の場合については，多少の脅威となる可能性はあるが，非常に限定的なケースであり，提案方式そのものの利用を妨げるようなものではない．提案方式にとっての脅威は，(c) または (d) のようなタイプのブレイクである．そのような状態になった場合，電子署名アリバイを実現する手段としてヒステリシス署名は有効ではない．

ところで，コンピュータの処理能力の向上によるブレイク，すなわち計算量的なブレイクを考えた場合，解読に 2 の 80 乗程度の計算量を必要とするくらいの安全性を保持するためには，

電子署名方式：1,024 ビット（RSA 方式）
ハッシュ関数：160 ビット （SHA-1 方式）

のパラメータが必要だといわれている．これらの数字は，それぞれ，電子署名方式：署名検査鍵から署名生成鍵を算出するため，ハッシュ関数：衝突を 1 つ見つけるために必要な計算量から見積もったものであり，上記 (c) や (d) のようなブレイクにはさらなる計算量が必要となる．

以上のようなことから，本稿で提案しているヒステリシス署名にとって脅威となるようなハッシュ関数のブレイクは，本稿で問題としている暗号ブレイクより困難であると考えられる．

5.2 従来方式との関係

2.2 節で述べた電子署名を偽造される要件に対する対策としてタイムスタンプや電子公証を利用した場合においても，電子署名アリバイ実現機構と同様の効果が得られる場合がある．また，提案方式で利用している耐タンパ性を備えたモジュールは，論理的には，タ

イムスタンプや電子公証におけるオーソリティの役割を果たすものを各エンティティが所有している，と考えることもできる．しかし，従来方式と提案方式とは以下の点で異なっている．

(1) 証拠となるデータの保管場所

従来方式では，電子署名の偽造が行われたかどうかを判別するための証拠をオーソリティが保管するのに対して，提案方式では，各エンティティが所有する耐タンパ性を備えたモジュールに保管している．

実際の運用なども含めて考えた場合，署名生成履歴を当該モジュールではなく，従来のオーソリティにそのつど保管してもらうようなシステムでは，2.3 節でも述べたように，サービス依頼時にすでに不正が行われている場合に対処できなかったり，トラフィックの増加や負荷の集中などにより必要とするときにサービスを利用できなかったりするなどといった課題を解決しなければならず，特にヒステリシス署名のようにすべての署名生成履歴を安全に保管しておくことが必要な場合には，現状では実現困難である．

さらに，従来のオーソリティに署名生成履歴を保管する場合，署名生成したにもかかわらずその情報をオーソリティに知らせないなどといった不正が生じる恐れもある．

これに対し，提案方式では，電子署名の生成場所と署名生成履歴の保管場所とを耐タンパ性を備えたモジュール内に閉じ込めることで，その間での不正が介在する余地をなくすとともに，サービスの可用性を確保している．ただし，各エンティティがトラブル発生時の証拠となる署名生成履歴を保管することになるので，安全に保管するためのコストが発生するだけでなく，署名生成履歴の喪失の危険性も従来方式より高くなってしまう．

(2) エンティティの利便性

従来方式は，元々，ある時点で当該電子データが存在していたことを証明しようとするものであるのに対し，提案方式は，ある時点で当該電子データが存在していなかったこと証明しようとするものである．

したがって，従来方式を利用した場合，すべての電子署名付きメッセージにタイムスタンプを押してもらったり，副署してもらったりするとが必要となるため，署名生成者のみで処理を完了することができない．また，タイムスタンプを押していない電子署名付きメッセージはすべて無効だという考え方自体が，各エンティティにとって受け入れがたいものだと考える．

(3) *Linking* の目的

タイムスタンプの偽造を困難とするためにある時点のタイムスタンプとその前後のタイムスタンプとの間に相関を持たせる *Linking* と呼ばれる手法が提案されているが[5]，我々が提案するチェイニング署名も，署名生成履歴に含まれるある署名生成記録 $R2$ とその前後の署名生成記録 $R1$，$R3$ とを関連付けることにより，$R1-R2$ 間や $R2-R3$ 間に別の署名生成記録を後から追加したり，$R2$ を削除したりすることを困難にするために同様の手法を用いている．

ただし，我々は，そのような理由に加えて，IC カードのような内部記憶容量に限りのあるモジュールを用いるような場合に，署名生成履歴をモジュール外部に安全に出力・保管できるようにする目的でも *Linking* を利用している．すなわち，Linking を行わない単純な署名生成履歴では，そこに含まれる個々の署名生成記録間に相関がないので，耐タンパモジュール外に出力すると改ざんされても分からないが，Linking を行っておけば，署名生成記録間の相関をチェックすることで改ざんの有無を検出することができる．

6. おわりに

本稿では，暗号ブレイクに対する対策として電子署名アリバイ実現機構を提案した．さらに，電子署名アリバイ実現機構で利用する署名生成履歴の完全性を確保する手段として，ヒステリシス署名と履歴交差と呼ぶ 2 つのコンセプトを説明し，安全性などの評価を行った．

来るべき電子社会において，多くの人が安心感を持って生活を送るためには，何かトラブルが起こったときのための証拠性基盤の確立が必要である．暗号ブレイクをもたらすかもしれない各種技術が急速に進展する中で，ヒステリシス署名や履歴交差といった技術は，長期にわたって証拠性を保つための基本技術の 1 つだと考える．今後は，プロトタイプを開発し，性能なども含めたシステム実現に向けた評価を行う予定である．また，本人が自己の署名生成鍵を知りえた場合に生じる，署名生成履歴に残らない手段での署名生成などといった課題の解決も図っていく．

謝辞 洲崎誠一は，通信・放送機構の委託研究「次世代証拠基盤技術の研究開発」として研究を行った．また，松本勉は，文部科学省科学研究費補助金特定領域研究 13224040（松本勉）の支援を受けて研究を行った．本研究自体の方向性に対して数々のご助言を賜るとともに，具体的な方式などに関して活発にご討論いただいた早稲田大学の岩村充教授，東京電機大学の佐々木良一教授，(株) 日立製作所の豊島久氏，松木武氏，宝木和夫氏，吉浦裕氏，宮崎邦彦氏の各氏に対して，つつしんで感謝の意を表する．

参 考 文 献

1) 首相官邸：ミレニアム・プロジェクト（新しい千年紀プロジェクト）について，平成 11 年 12 月 19 日内閣総理大臣決定 (1999).
2) 通商産業省：申請・届出等手続の電子化推進のための基本的枠組み，平成 12 年 3 月 31 日行政情報システム各省庁連絡会議了承 (2000).
3) Rivest, R., Shamir, A. and Adeleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Comm. ACM*, No.21, pp.120–126 (1978).
4) Lenstra, A.K. and Verheul, E.R.: Selecting Cryptography Key Sizes, *Journal of Cryptology*, No.14(4), pp.255–293 (2001).
5) Haber, S. and Stornetta, W.S.: How To Time-Stamp a Digital Document, *Journal of Cryptology*, Vol.3, No.2, pp.99–111 (1991).
6) Bayer, D., Haber, S. and Stornetta, W.: Improving the Efficiency and Reliability of Digital Time-Stamping, *Sequences II, Methods in Communication, Security and Computer Science*, pp.329–334, Springer-Verlag (1993).
7) Buldas, A., Lipmaa, H. and Schoenmakers, B.: Optimally Efficient Accountable Time-Stamping, *Proc. PKC2000*, pp.293–305, LNCS 1751 (2000).
8) Adams, C., Sylvester, P., Zolotarev, M. and Zuccherato, R.: Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols, IETF PKIX-WG RFC3029 (2001).
9) 法務省民事局：公証制度に基礎を置く電子公証制度の導入について.
http://www.moj.go.jp/MINJI/minji24.html
10) Pedersen, T. and Pfitzmann, B.: Fail-stop Signatures, *SIAM J. COMPUT*, Vol.26, No.2, pp.291–330 (1997).
11) Anderson, R.: *Two remarks on public-key cryptology*, Manuscript (2000).
12) Bellare, M. and Milner, S.: A Forward-Secure Digital Signature Scheme, *Proc. CRYPTO'99*, pp.431–448 (1999).
13) 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 CSEC 研究会 (2000).
14) 洲崎誠一，宮崎邦彦，宝木和夫，松本 勉：暗号ブレイク対応電子署名アリバイ実現機構（その 2)—詳細方式，情報処理学会 CSEC 研究会 (2000).
15) 洲崎誠一，松本 勉：電子署名の偽造に関する一考察，情報処理学会コンピュータセキュリティシンポジウム 2001 (2001).
16) Pinkas, D., Ross, J. and Pope, N.: *Electronic Signature Formats for Long Term Electronic Signatures*, IETF S/MIME-WG RFC 3126 (2001).
17) 電子商取引推進協議会（ECOM）：電子署名文書長期保存に関する中間報告，認証・公証 WG (2001).
18) 四方順司，花岡悟一郎，Y. Zheng，今井秀樹：情報量的安全性に基づくメッセージ復元型署名方式の構成，第 23 回情報理論とその応用シンポジウム（SITA2000）予稿集，pp.579–582 (2000).

(平成 13 年 12 月 4 日受付)
(平成 14 年 6 月 4 日採録)

洲崎 誠一（正会員）

1991 年 3 月横浜国立大学電子情報工学科卒業．同年 4 月（株）日立製作所システム開発研究所に入所．以来，情報セキュリティ技術の研究開発に従事．2001 年 4 月横浜国立大学大学院環境情報学府入学．現在，同大学院環境情報学府博士課程後期に在学するとともに，システム開発研究所第 7 部（セキュリティシステム研究部）研究員．1996 年情報処理学会第 52 回全国大会優秀賞，平成 12 年度山下記念研究賞受賞．

松本 勉（正会員）

1958 年生．1986 年東京大学大学院博士課程（電子工学）修了，工学博士．同年横浜国立大学工学部専任講師．現在，同大学大学院環境情報研究院教授．1981 年より主として暗号や情報セキュリティの研究・教育に従事.「明るい暗号研究会」を数人の仲間とともに創り研究を始めた．国際暗号学会 IACR 理事．ASIACRYPT '96 プログラム委員長．ASIACRYPT 2000（国際暗号学会主催）実行委員長．電子情報通信学会より「情報セキュリティの基礎理論」への貢献に関して業績賞を受賞．