情報処理学会論文誌 Vol. 51 No. 9 1542–1553 (Sep. 2010)

推薦論文

次世代電子パスポートへの署名偽造攻撃の適用評価

酒見由美[†1] 伊豆哲也[†2] 武仲正彦[†2]
野上保之[†1] 森川良孝[†1]

出入国の厳格かつ迅速な管理を目的として，国際民間航空機関（ICAO）は電子パスポート（e-Passport）の導入を推進しており，日本を含むいくつかの国ですでに発行が開始されている．2009 年 8 月に開催された国際会議において Coron，Naccache，Tibouchi，Weinmann は次世代の e-Passport が使用する ISO/IEC 9796-2 署名の偽造攻撃法と，実際の偽造署名データを発表した．Coron らは計算機実験データをもとに，他の条件下での攻撃コストを予想している．Coron らの署名偽造攻撃では，条件を変更した場合，攻撃計算量の算出に必要なパラメータの計算方法に影響が生じる．しかし，Coron らの評価方法では，その影響について考察されていないため，他の条件下での脅威が判断しにくいという問題がある．本稿では CNTW 攻撃の詳細な計算量を算出・評価するとともに Coron らの署名偽造攻撃を次世代 e-Passport に適用した場合の偽造可能性を議論する．

Evaluating the Forgery Attack against the Next-generation e-Passport

Yumi Sakemi,[†1] Tetsuya Izu,[†2] Masahiko Takenaka,[†2]
Yasuyuki Nogami[†1] and Yoshitaka Morikawa[†1]

For establishing strict and rapid immigration control, ICAO has been spreading the electronic passport (e-Passport), which is introduced in some countries including Japan. Recently, on August 2009, Coron, Naccache, Tibouchi, and Weinmann announced a new forgery attack against the signature scheme ISO/IEC 9796-2, which will be used in the next-generation e-Passport. Using the experimental results, Coron et al. estimated the attack's cost under other conditions, but they did not consider that parameters which used to compute the attack's cost depend on the conditions. Therefore, it is difficult to evaluate the attack's threat by their estimates. In this paper, the detailed cost of the attack is shown. Then, this paper discusses the possibility and the effect when the attack is applied to the next-generation e-Passport.

1. はじめに

出入国における厳格かつ迅速な管理を目的として，ICAO（国際民間航空機関；International Civil Aviation Organization）は電子パスポート（e-Passport）の普及を積極的に推進している[5]．2004 年 10 月に公開された e-Passport の最初の仕様では，電子データ（MRTD；Machine Readable Travel Documents）の完全性保護機能（PA；Passive Authentication）と基本的なアクセス制御機能（BAC；Basic Access Control）が搭載され，いくつかの国ですでに導入されている（日本では 2006 年 3 月に e-Passport の発行が開始された）．さらにクローニング防止機能（AA；Active Authentication）を実現する次世代 e-Passport の仕様策定も進んでおり，ISO/IEC 9796-2 Scheme 1（以下，単に ISO/IEC 9796-2 署名と記す）と呼ばれる素因数分解問題（RSA）をベースとした署名方式が使用される予定となっている[6]．

一方で，2009 年 8 月に開催された暗号技術に関する国際会議 CRYPTO 2009 において，Coron-Naccache-Tibouchi-Weinmann は，ISO/IEC 9796-2 署名に対する偽造攻撃（CNTW 攻撃）を発表し，計算機による偽造実験データを示した[3]．実際，2048 ビット合成数およびハッシュ関数 SHA-1 を用いた ISO/IEC 9796-2 署名に対し，約 2 日で偽造署名が算出できたと報告されている．

Coron らは計算機実験データをもとに，他の条件下での攻撃コストを予想している[3]．CNTW 攻撃では条件を変更すると，攻撃コストの算出に使用するパラメータの計算方法に影響が生じる．攻撃コストの算出時に使用するパラメータは ISO/IEC 9796-2 署名のメッセージエンコーディング関数（以下，パディング関数と呼ぶ）と使用する合成数に依存する．そのため，他の条件下での攻撃コストを算出する際には，ISO/IEC 9796-2 署名において，ハッシュ関数を変更した場合のトレーラの変化と，合成数を変化させた際のパラメータ

†1 岡山大学
Okayama University
†2 株式会社富士通研究所
FUJITSU Lab.
本稿の内容は 2009 年 10 月のコンピュータセキュリティシンポジウム 2009 にて報告され，コンピュータセキュリティ研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

の振舞いについて考察する必要がある．しかし，Coron らの評価方法では，その影響について考慮されていないため，他の条件下での脅威が判断しにくいという問題がある．本稿は，条件の変更時に生じる影響を考慮した CNTW 攻撃の詳細な評価を与える．具体的には，各ハッシュ関数に対応したパディング関数を使用し，10 万個の 2048-bit 合成数に対するパラメータを算出して，その振舞いを調査する．そして，その平均値を使って CNTW 攻撃の計算コストを算出し，評価を行う．その結果，ハッシュ関数を SHA-224 に変更した時点で署名偽造に約 100 万ドルの費用が必要となり，CNTW 攻撃が現実的な費用で適用できるのは SHA-1 以下のハッシュ長（出力長）を持つハッシュ関数を使用する場合であることが判明した．また，使用する合成数により攻撃コストに約 20%の変動が生じることを確認した．加えて，本稿では CNTW 攻撃を次世代 e-Passport に適用した場合の偽造可能性を議論することを目的とする．結論として，確かに次世代 e-Passport は ISO/IEC 9796-2 署名を用いているものの，CNTW 攻撃を適用しても偽造署名を算出できる可能性はきわめて低く，たとえ構成できたとしても現実的な脅威はほとんどないという結果が得られた．しかし e-Passport の有効期限が比較的長期（日本は最長 10 年）であり，CNTW 攻撃が改良される可能性を加味すれば，次世代 e-Passport における ISO/IEC 9796-2 署名の使用は避けるべきである．

本稿の構成は以下のとおりである：2 章で ISO/IEC 9796-2 署名のアルゴリズムを記述するとともに，Coron-Naccache-Tibouchi-Weinmann による署名偽造攻撃（CNTW 攻撃）を説明する．3 章では CNTW 攻撃の詳細な評価結果を示したうえで，4 章で CNTW 攻撃を次世代 e-Passport に適用した場合の偽造可能性について議論する．

2. ISO/IEC 9796-2 署名について

本章では，ISO/IEC 9796-2 署名[7] のアルゴリズムと，2009 年 8 月に提案された Coron-Naccache-Tibouchi-Weinmann による署名偽造攻撃（CNTW 攻撃）[3] を説明する．

2.1 ISO/IEC 9796-2 署名

ISO/IEC 9796 ではメッセージの部分（または完全）復元が可能な署名方式を規定しており，現時点では，素因数分解問題（RSA 署名）を利用した方式 ISO/IEC 9796-2[7] と，離散対数問題を利用した方式 ISO/IEC 9796-3 とに分類される．2002 年に制定された現行の ISO/IEC 9796-2 では 3 つの方式（Scheme 1，2，3）が記述されている．本稿の考察対象は Scheme 1 だけであるため，以下では Scheme 1 を単に ISO/IEC 9796-2 署名と記す．

セキュリティパラメータ k に対し，$(\mathsf{sk}, \mathsf{pk})$ を署名者の秘密鍵・公開鍵ペアとする．ただし $\mathsf{sk} = (p, q, d)$，$\mathsf{pk} = (N, e)$，p，q は $k/2$-bit 素数，$N = p \times q$ は k-bit 合成数，d，e は $de \equiv 1 \pmod{(p-1)(q-1)}$ を満たす整数とする．このときメッセージ m に対する署名 σ は，パディング関数 $\mu(\cdot)$ を用いて

$$\sigma = \mu(m)^d \bmod N$$

と表せる．ここで，パディング関数 $\mu(\cdot)$ は

$$\mu(m) = \texttt{0x6A}||m[1]||H(m)||\texttt{0xBC} \tag{1}$$

と定められている．また，$H(\cdot)$ は出力長 k_H（≥ 160）bit のハッシュ関数，$m[1]$ はメッセージ m の上位 $(k - k_H - 16)$-bit を示す．さらに，0x6A はパディングが ISO/IEC 9796-2（部分復元）であることを示すヘッダ，0xBC はハッシュ関数が SHA-1 であることを示すトレーラである．このとき，任意のメッセージ m に対し，$\mu(m)$ はつねに $k-1$ ビットとなっている．

署名 σ を受け取った検証者は，$\overline{m} = \sigma^e \bmod N$ によってパディングされたメッセージ $\overline{m}$ を構成し，フォーマットを満たしているかをチェックする．

このとき，メッセージ m が $k - k_H - 16$ ビット以下ならば $m[1] = m$ となるので，ISO/IEC 9796-2 署名はメッセージを完全に回復できている．また $k - k_H - 16$ ビットより長ければ，ISO/IEC 9796-2 署名はメッセージを部分的に回復できることになる．

2.2 CNTW 攻撃

2009 年 8 月に開催された暗号に関する国際会議 CRYPTO 2009 において，Coron-Naccache-Tibouchi-Weinmann は，ISO/IEC 9796-2 署名の新しい偽造方法（CNTW 攻撃）を提案し，実際に偽造が可能であることを計算機実験によって確認した[3]．本節では CNTW 攻撃の概要を説明する．

CNTW 攻撃（およびベースとなった攻撃）の目標は，偽造メッセージを m^* とするとき，L 個のメッセージ $m_1, m_2, \ldots, m_L$ による積表現

$$\mu(m^*) = \delta^e \mu(m_1)^{e_1} \mu(m_2)^{e_2} \cdots \mu(m_L)^{e_L} \bmod N \tag{2}$$

の係数 δ と各指数 $e_1, e_2, \ldots, e_L$（$1 \leq e_1, e_2, \ldots, e_L < e$）を導出することである（係数 δ の算出方法は文献 3) に示されているが，本稿ではその記述を省略する）．このとき，それぞれのメッセージに対応する署名の間では，

$$\sigma^* = \delta \cdot \sigma_1^{e_1} \sigma_2^{e_2} \cdots \sigma_L^{e_L} \bmod N \tag{3}$$

という関係式が成立するため，攻撃者が $\sigma_1, \sigma_2, \ldots, \sigma_L$ を入手できれば，偽造署名 σ^* を実際に導出することができる．

上のようなメッセージ間の積表現を算出するために，Desmedt-Odlyzko は $\mu(m_i)$ の素因数分解を利用した方法を提案した[4]．しかし，実際に署名を偽造するには，$\mu(m_i)$ は 200-bit 以下でなければならず，ISO/IEC 9796-2 には適用できなかった．そこで Coron-Naccache-Stern は $\mu(\cdot)$ の代わりに，$\mu(\cdot)$ から算出される別のパディング関数

$$\nu_{a,b}(\cdot) = a \cdot \mu(\cdot) - b \cdot N \tag{4}$$

の素因数分解を利用した方法を提案した（CNS 攻撃）[2]．パラメータ a，b を適切に設定することで $\nu_{a,b}(\cdot)$ はたかだか $(k_H + 16)$-bit となるため，偽造に必要な計算量は，$k_H = 128$ のときで 2^{54}，$k_H = 160$ のときで 2^{61} となり，ISO/IEC 9796-2 署名が理論的には偽造可能であることが示された．当時の ISO/IEC 9796-2 署名では $k_H \geq 128$ と定められていたが，Coron-Naccache-Stern の攻撃により，$k_H \geq 160$ に変更された．

しかし ISO/IEC 9796-2 署名の実際の偽造には至っておらず，理論的な偽造可能性を示すに留まっていた．

2009 年 8 月に開催された CRYPTO 2009 において，Coron-Naccache-Tibouchi-Weinmann はパラメータ a，b を最適化し，出力値がたかだか $(k_H + |a|)$-bit（$|a|$ は数ビットのパラメータ）になることを示した．また Coron-Naccache-Stern アルゴリズムの各処理の実装を高速化し，（実質的に）$(k_H + |a| - 8)$-bit 以下となるメッセージだけを扱うことで，署名に関する積表現の実際の導出に成功した[3]．具体的には，N が 2048-bit の合成数，$e = 2$，ハッシュ関数が SHA-1，$|a| = 10$ の場合に約 2 日間で偽造署名が算出できることを示した．このとき $\nu_{a,b}(m)$ が実質的に 162-bit 以下となるようなメッセージ m だけを扱っている．

計算環境には Amazon のクラウドコンピューティングサービス EC2 を利用し，約 800 ドル分の計算リソースを使用した．

2.3 CNTW 攻撃の影響

CNTW 攻撃は署名に関する積表現 (3) を用いて偽造署名 σ^* を導出するため，攻撃者は L 個の正当な署名 $\sigma_1, \sigma_2, \ldots, \sigma_L$ を必要とする．しかしハッシュ関数として SHA-1 を用いた場合，L は膨大（上の実験では $L = 2^{18.7}$）なため，攻撃者がこれら署名を入手するのは現実的には困難である．

また攻撃者が署名を入手して偽造署名 σ^* を導出できたとしても，対応する偽造メッセージ m^* の上位 $(k - k_H - 16)$-bit は偽造過程で自動的に定められるため，m^* の上位ビットはほぼランダムとなり，m^* が意味のあるデータとなる可能性はきわめて低い．

以上のことから，確かに CNTW 攻撃は ISO/IEC 9796 -2 署名に対する偽造署名の算出に成功しているものの，現実社会における影響は無視できるほどに小さい．そのため，何らかのシステムが ISO/IEC 9796-2 署名を使用しているからといってただちに別の署名方式に移行する必要性は薄いと考えられる．しかし攻撃アルゴリズムのさらなる改良を加味すれば，これから新たに構築するシステムでは，ISO/IEC 9796-2 署名（Scheme 1）の使用は避けるべきである（実際，Scheme 1 は既存システムとの互換性維持を目的とされており，新システムでは Scheme 2，3 の使用が推奨されている）．

3. CNTW 攻撃の詳細評価

Coron-Naccache-Tibouchi-Weinmann は，2.2 節で紹介した計算機実験のデータをもとに，他の条件下での攻撃計算量を予想している[3]．しかし ISO/IEC 9796-2 署名においてハッシュ関数 SHA-2 を使用した場合にトレーラが 16-bit になること，また，合成数 N を変化させたときのパラメータ a の振舞いが考察されていないことから，他の条件下での脅威が判断しにくいという問題がある．そこで本章では，CNTW 攻撃の詳細な計算量を算出・評価する．

3.1 CNTW 攻撃計算量の算出方法

CNTW 攻撃（および，そのベースである CNS 攻撃）では，署名に関する積表現 (3) を算出するために，パディング値が p_L-smooth となるようなメッセージ（とその素因数分解）を L 個以上収集する必要がある（ここで p_L は L 番目の素数であり，ある自然数が p_L-smooth であるとは，その自然数が p_L 以下の素数で素因数分解できることを指す）．このメッセージ探索の計算量は署名偽造全体の計算量の大部分を占めるため，メッセージ探索の改良が必須となる．そこで CNTW 攻撃は，メッセージ探索を，(1) p_L-smooth 判定テスト，(2)（p_L-smooth と判定された場合に）素因数分解，の 2 段階に分割することでメッセージ探索を高速化した．特に p_L-smooth 判定テストに Bernstein's smoothness detection algorithm（BSDA）を用いた場合，試し割り算法に比べて約 1,000 倍の高速化を実現した．

BSDA を用いた場合，n 個の x-bit の自然数が p_L-smooth であるかを判定するために必要な計算量は $O(t \cdot \log^2 t \cdot \log\log t)$ となる．ここで，t は n 個の x-bit 自然数のリストおよび p_L 以下の素数リストのサイズを表しており，$t = n \cdot x + L \cdot \log_2 L$ である[3]．ランダムな x-bit の自然数が p_L-smooth である確率を α とすると，パディング値が p_L-smooth であるメッセージを L 個収集するためには $n = L/\alpha$ 個のメッセージに対する smoothness 判定テスト（BSDA）が必要となる．CNTW 攻撃のように n が非常に大きくなる場合，$n' = n/k$ 個のメッセージに対する BSDA 処理を k 回行う方が効率が良い．最適な n' を選択した場

合，メッセージ探索に必要な計算量は次式で与えられる：

$$C_{\mathrm{BSDA}} = n \cdot s \cdot \log^2 t' \cdot \log\log t', \tag{5}$$

ここで $t' = (u \log u)/2$，$u = L \cdot \log_2 L$ である．

さらに，CNTW 攻撃では Large Prime Variant と呼ばれるテクニックを利用して，BSDA の処理を施すメッセージの総数 n を削減している．具体的には，メッセージ探索処理 (1) の p_L-smooth 判定テストの代わりに (p_L, p_{L2})-semismooth 判定テストを行う．ここで，ある自然数が (p_L, p_{L2})-semismooth であるとは，その自然数の最大素因数を除くすべての素因数が p_L 以下であり，かつ最大素因数が p_{L2} 以下であることを指す．メッセージ探索処理において，p_L-smooth でないと判定されたメッセージの中には，最大素因数を除くすべての素因数は p_L 以下であるが，最大素因数のみが p_L より大きいために，おしくも p_L-smooth とならなかったメッセージも存在する．そのようなメッセージを偽造処理において有効に利用することにより，BSDA の処理を施すメッセージの総数を削減できる．Large Prime Variant を適用した際の BSDA の処理を施すメッセージの総数の算出方法については，文献 3) の付録 E で詳細に議論されている．

3.2　CNTW 攻撃のパディング関数

CNTW 攻撃が使用するパディング関数 $\nu_{a,b}(\cdot) = a \cdot \mu(\cdot) - b \cdot N$ の任意の入力値に対する出力値ができるだけ小さくなるような a，b の求め方を説明する．なお，ここではハッシュ関数に SHA-1 を用いる場合について述べる．$\mu(\cdot)$ と N がほぼ同じサイズであることと，$\mu(\cdot)$ の最上位および最下位の各 8-bit が固定値であることから，a，b を適切に定めることで，$\nu_{a,b}(\cdot)$ の最上位，最下位の各 8-bit を 0 にすることができる．

具体的に，そのようなパラメータ a，b は次の条件を満たす[3]．

$$0 < b \cdot N - a \cdot C_\mu < a \cdot 2^{k-8} \tag{6a}$$

$$b \cdot N - a \cdot C_\mu \equiv 0 \bmod 2^8 \tag{6b}$$

ここで，C_μ はパディング関数 $\mu(\cdot)$ の $m[1]$ 部分とハッシュ値部分を 0 に設定したもので，次式で与えられる．

$$C_\mu = \mathtt{0x6A} \cdot 2^{k-8} + \mathtt{0xBC} \tag{7}$$

さらにメッセージの上位 $(k - k_H - 16)$-bit の値 $m[1]$ を適切に定めることで，$\nu_{a,b}(\cdot)$ の上位 $9 - (k - k_H - 8)$-bit についても 0 にできる．その結果 $\nu_{a,b}(\cdot)$ の出力長を $(k_H + |a|)$-bit に削減した．

CNTW 攻撃の計算量は $\nu_{a,b}(\cdot)$ の出力長に強く依存するため，a のサイズが小さいほど攻撃計算量を大幅に低減させることができることとなる．

文献 3) では，合成数 N として RSA2048[9] を使用した場合に，式 (6) を満たす最小の a が 10-bit となることが報告されている．このとき BSDA の処理対象は 170-bit となる．

さらに CNTW 攻撃では，ハッシュ計算コストが，smoothness 判定テスト（BSDA）のコストに比べて十分に小さいことから，メッセージ m に対するパディング値 $\nu_{a,b}(m)$ の先頭 8-bit が 0 になるようなメッセージを選択している．これによって，BSDA の処理対象となるメッセージの大きさを実質的に $170 - 8 = 162$-bit に削減した．

3.3　パディング関数の詳細評価

3.2 節で述べたように，CNTW 攻撃の計算量を決めるパディング関数 $\nu_{a,b}(\cdot)$ の出力値は，使用する合成数 N から決まる a の大きさに依存する．したがって，使用する合成数 N に対する a の大きさが重要となる．しかし，文献 3) では，RSA2048[9] に対する a，b だけしか議論されておらず，使用する N を変化させた場合の a，b の振舞いが不明である．すなわち，RSA2048 以外の N を使用した際の CNTW 攻撃の脅威が予測困難である．そこで 2048-bit の RSA 型合成数を 10 万個準備し，SHA-1 を用いた場合の a の値とその分布を求めた（**図 1**，**表 1**）．ここで，実験には，ランダムに生成した 2 つの 1024-bit 素数の積からなる 2048-bit 合成数を用いた．**アルゴリズム 1** に合成数 N に対する a の算出アルゴリズムを示す．ここで，**アルゴリズム 1** におけるパラメータ Trl はパディング関数 $\mu(\cdot)$ のトレーラ，|Trl| は Trl のビット数を意味しており，SHA-1 の場合はそれぞれ $\mathtt{Trl} = \mathtt{0xBC}$，$|\mathtt{Trl}| = 8$ となる．**アルゴリズム 1** では，a の初期値を 1 として，1 ずつ増加させていき，その a に対して式 (6) を満たすような b を探索する．そして，最初に所望の b が見つかった時点での a が式 (6) を満たす最小の a となる．ここで，**アルゴリズム 1** の *step* 7 および *step* 19 で b を探索する際の初期値を $\lfloor a \cdot C_\mu / N \rfloor$ にしているが，これは式 (6) を満たす b が $\lfloor a \cdot C_\mu / N \rfloor$ 付近に存在するためである．

文献 3) では**アルゴリズム 1** における N に RSA2048 のみを使用し，得られた a を評価に用いている．それに対し，本実験では準備した 10 万個の N それぞれに対して**アルゴリズム 1** を実行することで得られる平均的な a を評価に用いる．図 1 より，a の平均値は 8.02-bit となり，ほぼ (8 ± 2)-bit 内に収まることが判明した．

他方で文献 3) では，ハッシュ関数に SHA-256 を用いた場合でも，合成数 N が RSA2048，ハッシュ関数が SHA-1 の場合に得られた a を評価に使用している．すなわち，パディング関数 $\mu(\cdot)$ のトレーラを SHA-1 と同様の 8-bit として評価している．しかし SHA-256 を用

表 1　10 万個の 2048-bit 合成数 N，SHA-1 使用時の a のビットサイズに対する N の度数分布表
Table 1　Freguency distribution of 2048-bit composite N's with regard to bit-size of a when experimenting using 100,000 Ns (SHA-1).

a（bit）	2	3	4	5	6	7	8	9	10	11	12	13	14
N の個数（個）	0	87	102	599	2,108	9,045	34,463	42,122	8,606	2,058	630	180	0

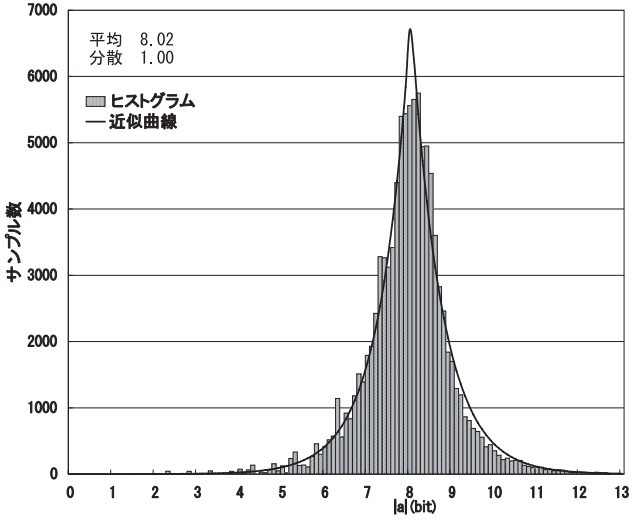

図 1　N が 2048-bit，SHA-1 使用時の a 値の分布
Fig. 1　Distribution of a-values for 2048-bit composite N's (SHA-1).

いた場合，トレーラは 16-bit 値 0x34CC となるため，次式のようなパディング関数 $\mu(\cdot)$ を用いることとなる．

$$\mu(m) = \texttt{0x6A}||m[1]||\texttt{SHA}_{256}(m)||\texttt{0x34CC} \tag{8}$$

この場合，$m[1]$ はメッセージ m の上位 $(k-256-24)$-bit である．パラメータ a，b は合成数 N とパディング関数 $\mu(\cdot)$ から決まるため，SHA-256 を用いる場合には算出方法が変わってくる．したがって，文献 3) の評価結果を用いて SHA-256 を用いた場合の CNTW 攻撃を詳細に評価することは不可能である．そこで，SHA-256 の場合に対応したパラメータ a の算出方法を示すとともに，その算出方法を用いて a の評価を行う．

SHA-256 を用いる場合，ヘッダの値が 8-bit であり，トレーラの値が 16-bit であることから，$\nu_{a,b}(\cdot)$ の最上位 8-bit と最下位 16-bit が 0 となるような a，b を定めることが可能となる．よって，SHA-256 を用いる場合には，a，b の条件は式 (6) およびトレーラの値から次のように与えられる．

$$0 < b \cdot N - a \cdot C_\mu < a \cdot 2^{k-8} \tag{9a}$$
$$b \cdot N - a \cdot C_\mu \equiv 0 \bmod 2^{16} \tag{9b}$$

ここで，C_μ はパディング関数 $\mu(\cdot)$ の $m[1]$ 部分とハッシュ値部分を 0 に設定したもので，次式で与えられる．

$$C_\mu = \texttt{0x6A} \cdot 2^{k-8} + \texttt{0x34CC} \tag{10}$$

SHA-1 の場合と同様に，10 万個の 2048-bit RSA 型合成数 N に対して a を算出し，その分布を求めた（**図 2**，**表 2**）．ただし，トレーラの値の変化にともない，SHA-256 の場合には**アルゴリズム 1** における Trl を 0x34CC とする必要がある．文献 3) では，Trl を 8-bit 値として評価している．図 2 よりトレーラの値の変化を加味すると a の平均値は実際には 12.0-bit となり，ほぼ (12 ± 2)-bit 内に収まることが判明した．同様に，ハッシュ関数に SHA-2 を用いた場合のトレーラはすべて 16-bit となるため，パディング関数 $\nu_{a,b}(\cdot)$ の出力値は平均 $(k_H + 12)$-bit となることが分かる．

3.3.1　パラメータ a の分布に関する考察

図 1 および図 2 より，不特定の N に対して a はほぼ (平均値 $\pm$ 2)-bit 以内に収まることが判明した．しかし一方で，SHA-1 および SHA-256 使用時に a が 3-bit ときわめて小さくなるような合成数 N の存在も確認された．SHA-1 および SHA-256 使用時に a が 3-bit となる合成数 N とそれに付随する各パラメータをそれぞれ式 (17) および式 (18) に示す．先述したように，パラメータ a の大きさが小さくなるほど，攻撃計算量は大幅に削減される．そのため，大きさが (平均値 $\pm$ 2)-bit を下回るような a の生起確率を詳細に評価する必要がある．そこで，実験により得られたパラメータ a の分布から近似式を求め，得られた近似式から a の生起確率について詳細に評価する．

図 1 と図 2 を比較すると，SHA-256 使用時の方が滑らかな分布となっていることが分かる．そこで，SHA-256 を用いた際の実験結果を用いて，最小二乗法による近似を行った．そ

表 2　10 万個の 2048-bit 合成数 N，SHA-256 使用時の a のビットサイズに対する N の度数分布表
Table 2　Freguency distribution of 2048-bit composite N's with regard to bit-size of a when experimenting using 100,000 Ns (SHA-256).

a（bit）	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
N の個数（個）	0	1	0	2	10	30	119	541	2,239	8,769	34,348	42,897	8,463	1,929	478	132	34	7	0	1	0

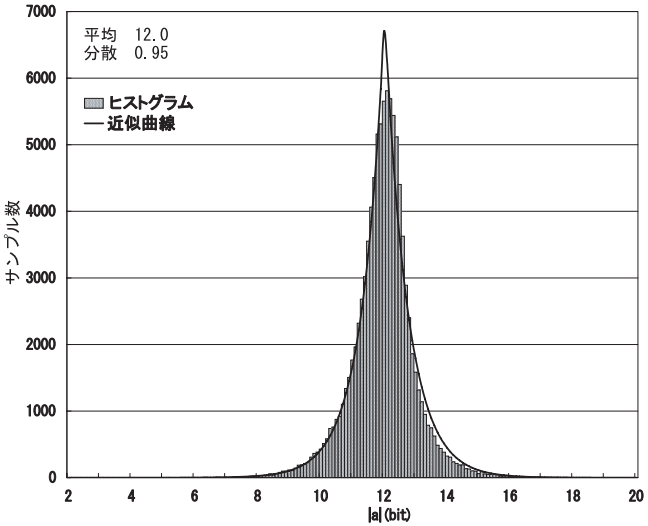

図 2　N が 2048-bit，SHA-256 使用時の a 値の分布
Fig. 2　Distribution of a-values for 2048-bit composite N's (SHA-256).

の結果，パラメータ a のビットサイズを x，度数を S として次の近似式が得られた．

$$S = \frac{1}{2 \cdot \dfrac{1}{\alpha \log_e 2}} \cdot \exp\left(\frac{-|x-\bar{x}|}{\dfrac{1}{\alpha \log_e 2}}\right) \tag{11}$$

ここで，$\bar{x}$ は分布の平均値であり，SHA-256 の場合，$\bar{x} = 12.0$ である．また，α は定数であり，実験結果を用いて算出すると $\alpha = 2.07$ となった．式 (11) はちょうど，期待値が $\bar{x}$，分散が $\frac{2}{(\alpha \log_e 2)^2}$ のラプラス分布である．

式 (11) を式変形すると，次式が得られる．

$$\begin{aligned} S &= \frac{\alpha \log_e 2}{2} \cdot \exp\big(-\alpha \log_e 2 |x-\bar{x}|\big) \\ &= \frac{\alpha \log_e 2}{2} \cdot \Big(\exp\big(\log_e 2\big)\Big)^{-\alpha|x-\bar{x}|} \\ &= \frac{\alpha \log_e 2}{2} \cdot 2^{-\alpha|x-\bar{x}|} \end{aligned} \tag{12}$$

式 (12) より，a の分布は 2 を底とする指数関数となることが分かる．ここで，パラメータ a，b を求める際のパディング関数 $\nu_{a,b}(\cdot)$ の振舞いについて考察する．$\nu_{a,b}(\cdot)$ において，a が 1-bit 増えると，b も 1-bit 増えることから，自由度は 2-bit ずつ増加（探索空間が 2^2 倍増加）する．そのため，パラメータ a のビットサイズに対する度数は a が 1-bit 増えるごとに，2^2 倍ずつ増加すると考えられる．したがって，パディング関数 $\nu_{a,b}(\cdot)$ の振舞いを考慮すると，a の分布は 2^2 の指数関数に従うと予想できる．そこで，実験で得られた $\alpha = 2.07$ を 2 として近似を行うと，次式が得られる．

$$S \sim \log_e 2 \cdot 2^{-2|x-\bar{x}|} \tag{13}$$

式 (13) から得られる SHA-1 および SHA-256 使用時の a の分布に対する近似曲線を，それぞれ図 1 と図 2 に示す．ただし，平均値 $\bar{x}$ は SHA-1 の場合は 8.02，SHA-256 使用時は 12.0 となる．図 1 および 図 2 において，実験結果であるヒストグラムと近似曲線では，周辺部分で度数がほぼ一致している．本稿では，周辺部分の a について議論を行うため，式 (13) を用いて近似を行うこととする．

式 (13) から a が (平均値 ± 2)-bit となる確率は以下のように求められる．

$$\begin{aligned} &\int_{\bar{x}-2}^{\bar{x}+2} \log_e 2 \cdot 2^{-2|x-\bar{x}|} dx \\ &= 2 \log_e 2 \int_{\bar{x}-2}^{\bar{x}} \exp\big(\log_e 2^{2(x-\bar{x})}\big) dx \\ &= 2 \log_e 2 \int_{\bar{x}-2}^{\bar{x}} \exp\big(2(x-\bar{x}) \log_e 2\big) dx \\ &= \Big[\exp\big(2(x-\bar{x}) \log_e 2\big)\Big]_{\bar{x}-2}^{\bar{x}} \\ &= 1 - 2^{-4} = 0.937 \end{aligned} \tag{14}$$

アルゴリズム 1：パディング関数 $\mu(\cdot)$ のトレーラが `Trl` である場合の合成数 N に対するパラメータ a を算出するアルゴリズム

Input:　N
Output:　a, b

1. $D_1 \leftarrow \texttt{0x6A} \cdot 2^{k-8} + \texttt{Trl}$　　// D_1 に C_μ を設定する
2. $a \leftarrow 1$
3. For:
4. 　$D_2 \leftarrow a \cdot D_1$.　　//$D_2$ に $a \cdot C_\mu$ を格納する
5. 　$D_3 \leftarrow a \cdot 2^{k-8}$.
6. 　$D_4 \leftarrow \lfloor D_2/N \rfloor$.
7. 　$b \leftarrow D_4$
8. 　For:
9. 　　$D_5 \leftarrow b \cdot N$
10. 　　If D_5 is larger than D_2, then:
11. 　　$D_6 \leftarrow D_5 - D_2$.　　//D_6 に $b \cdot N - a \cdot C_\mu$ を格納する
12. 　　If D_6 is smaller than D_3, then:
13. 　　If D_6 & $(2^{|\texttt{Trl}|} - 1)$ is equal to 0, then
14. 　　goto *step* 31
15. 　　else
16. 　　goto *step* 19
17. 　　$b \leftarrow b + 1$
18. 　End for:
19. 　For $b \leftarrow D_4 - 1$ down to 1:
20. 　　$D_5 \leftarrow b \cdot N$
21. 　　If D_5 is larger than D_2, then:
22. 　　$D_6 \leftarrow D_5 - D_2$.　　//D_6 に $b \cdot N - a \cdot C_\mu$ を格納する
23. 　　If D_6 is smaller than D_3, then:
24. 　　If D_6 & $(2^{|\texttt{Trl}|} - 1)$ is equal to 0, then
25. 　　goto *step* 31.
26. 　　else
27. 　　goto *step* 29.
28. 　End for:
29. 　$a \leftarrow a + 1$
30. End for:
31. Return a, b

式 (14) は全体の約 94%の N に対するパラメータ a が (平均値 ±2)-bit となることを意味している．これは，表 1 および表 2 より，10 万個の N のうち，SHA-1 使用時には 96,344 個，SHA-256 使用時には 96,716 個の合成数 N に対する a が (平均値 ± 2)-bit となっており，実験結果とも一致しているため，妥当といえる．したがって，全体の約 94%の N に対するパラメータ a が (平均値 ± 2)-bit となることが示された．

表 3　各ハッシュ関数に対する CNTW 攻撃の計算量
Table 3　Complexity of CNTW attack with regard to hash functions.

ハッシュ関数	$s = \lvert\nu_{a,b}(\cdot)\rvert$	$\log_2 L$	$\log_2$ cost	計算時間	$\log_2 \tau$	Amazon EC2 cost [US $]
SHA-1	160	21	55.5	0.5 年	38	454
SHA-224	228	27	66.7	1,286 年	48	1,110,777
SHA-256	260	29	71.5	34,874 年	52	30,131,119
SHA-384	388	38	88.3	3,902,163,409 年	68	3,371,469,185,653
SHA-512	516	46	102.6	84,083,141,453,024 年	81	72,647,834,215,413,100

また，同様にして実験結果より得られた最小の a（3-bit）が発生する確率を求めた．SHA-1 および SHA-256 使用時に a が 3-bit 以下となる確率は SHA-1 使用時には 9.8×10^{-2}%，SHA-256 使用時には 3.8×10^{-4}%となり，きわめて低い確率であることが判明した．この場合の攻撃計算量については，次節で評価を行う．

3.4　CNTW 攻撃の計算量評価

文献 3) では，特定の合成数（`RSA2048`）を用いた偽造実験の計算時間等の情報を用いて，他の条件下での攻撃コストが評価されている．しかし 3.3 節で述べたように，a 値の分布やハッシュ関数の違いを考慮していないため，CNTW 攻撃の脅威が正確に判断できないという問題がある．そこで本節では，これらの差異を考慮した攻撃コストを算出する．

3.1 節で述べたように，CNTW 攻撃では，署名に関する積表現 (3) の算出に必要なメッセージ探索の計算コストが署名偽造全体の計算量の大部分を占める．そのため，本稿ではメッセージ探索に必要な計算コストを CNTW 攻撃の計算コストと考える．前節までの議論をふまえた攻撃コストの算出結果を**表 3** に示す．ここで s は各ハッシュ関数を用いた場合の実質的なパディング関数 $\nu_{a,b}(\cdot)$ の出力値の大きさであり，3.2 節で述べたメッセージ選択による改良も加味して，次式で与えられる．

$$s = k_H + |a| - 8 \tag{15}$$

また，L は式 (5) の攻撃計算量が最小となる値である．τ は Large Prime Variant を適用した際の BSDA の処理を施す必要のあるメッセージ数である．cost は L および τ を用いた場合の攻撃計算量であり，式 (5) により次式で与えられる．

$$\text{cost} = \tau \cdot s \cdot \log^2 t' \cdot \log\log t', \tag{16}$$

ここで $t' = (u \log u)/2$，$u = L \cdot \log_2 L$ である．

また，計算時間はシングルコア 2.4 GHz の PC 1 台を用いた場合であり，積表現 (3) の算

表 4　Coron らによる各 s 値に対する CNTW 攻撃の計算量[3)]
Table 4　Complexity of CNTW attack with regard to s-values by Coron, et al.

ハッシュ関数	$s = \lvert\nu_{a,b}(\cdot)\rvert$	$\log_2 L$	計算時間	$\log_2 \tau$	Amazon EC2 cost [US $]
–	64	11	15 秒	20	negligible
–	128	19	4 日	33	10
–	160	21	0.5 年	38	470
SHA-1	170	22	1.8 年	40	1,620
–	176	23	3.8 年	41	3,300
–	204	25	95 年	45	84,000
–	232	27	1,900 年	49	1,700,000
–	256	30	32,000 年	52	20,000,000

出に必要な時間である．計算時間の評価には，Coron-Naccache-Tibouchi-Weinmann が算出した「パディング関数 $\nu_{a,b}(\cdot)$ の出力値が 170 ビットの場合にはシングルコア 2.4 GHz の PC 1 台で 1.8 年の計算時間が必要となる」という情報を用い，各 s に対する cost の比から算出した．さらに Amazon EC2 cost は得られた計算時間から見積もった Amazon EC2 の使用料金である．また，比較のために文献 3) の評価結果を**表 4** に示す．

表 3 と表 4 から分かるとおり，ハッシュ関数に SHA-1 を使用する場合，Coron らの予想では署名偽造に 1,620 ドルの費用が必要であるのに対し，本計算量評価では 454 ドルとなった．すなわち，a 値の分布やハッシュ関数の違いを加味すると，実際には Coron らの予想よりも安価で署名偽造が可能であるということが分かる．また，表 3 から分かるとおり，ハッシュ関数として SHA-2 を使用する場合，SHA-224 の時点で署名偽造に約 100 万ドルの計算リソースを必要とする．計算時間という観点からも署名偽造に約 1,000 年必要であることから，本攻撃による署名偽造は難しいことが分かる．すなわち，CNTW 攻撃によって実際に署名偽造を成功できるのは，SHA-1 以下のハッシュ長（出力長）を持つハッシュ関数の場合である．次に，SHA-1 および SHA-256 に対し 3.3 節で考察した a 値のゆらぎに対する攻撃計算量の違いを**表 5** にまとめる．この表から分かるとおり，合成数 N によって，s 値が変動し，攻撃計算量が約 20%程度増減する．また，3.3.1 項で示されたように，全体の約 94%の N に対する a は (平均値 ± 2)-bit となるが，確率がきわめて小さいながらも a が 3-bit となるような N が確認された．この場合，実質的なパディング関数 $\nu_{a,b}(\cdot)$ の出力値は最大でも SHA-1 使用時には 155-bit，SHA-256 使用時には 251-bit となる．そのため，平均的なパディング関数 $\nu_{a,b}(\cdot)$ の出力値と比較すると，SHA-1 使用時には 5-bit，SHA-256 使用時には 9-bit 削減されることとなる．**表 6** に a が 3-bit の場合の CNTW 攻撃の攻撃計算量を示す．表 6 より，a が 3-bit となるような N を合成数に使用した場合，平均的な攻撃計算量と比較して，SHA-1 使用時には約 46%，SHA-256 使用時には約 60%削減されることとなる．しかし，表 3 より，そのような N の場合でも SHA-256 使用時の署名の偽造は現実的でないことが分かる．したがって，ハッシュ関数として SHA-2 を使用する場合，s 値を数ビット削減できる改良やパラメータ a がきわめて小さい値をとる合成数 N を使用したとしても，CNTW 攻撃による現実的な署名偽造の実行は困難であるという結論が得られる．

表 5　SHA-1，SHA-256 の場合の計算コスト
Table 5　Complexity comparison between SHA-1 and SHA-256.

SHA-1		SHA-256	
$\lvert s = \nu_{a,b}(\cdot)\rvert$	$\log_2$ cost	$s = \lvert\nu_{a,b}(\cdot)\rvert$	$\log_2$ cost
158	55.1	258	71.2
160	**55.5**	**260**	**71.5**
162	55.9	262	71.8

表 6　a が 3-bit となる場合の SHA-1，SHA-256 使用時の計算コスト
Table 6　Complexity of CNTW attack with 3-bit a-value (SHA-1, SHA-256).

SHA-1		SHA-256	
$\lvert s = \nu_{a,b}(\cdot)\rvert$	$\log_2$ cost	$s = \lvert\nu_{a,b}(\cdot)\rvert$	$\log_2$ cost
155	54.6	251	70.2

4. 次世代電子パスポートへの適用

本章では，ISO/IEC 9796-2 署名の利用が予定されている，次世代の電子パスポートについて，その機能概要を述べる．そして，ISO/IEC 9796-2 署名が利用される Active Authentication （以下 AA と記す）に対する CNTW 攻撃の適用について検討を行う．

4.1 電子パスポート

電子パスポート（e-Passport）は，出入国の厳格かつ迅速な管理を目的として，ICAO（国際民間航空機関；International Civil Aviation Organization）が標準化を積極的に推進している[5)]．2004 年 10 月に公開された電子パスポート（e-Passport）の最初の仕様では，基本的なアクセス制御機能（BAC；Basic Access Control）と電子データ（MRTD；Machine Readable Travel Documents）の完全性保護機能（PA；Passive Authentication）が搭載され，いくつかの国ですでに導入されている．日本では，2005 年 6 月に IC 旅券（電子パ

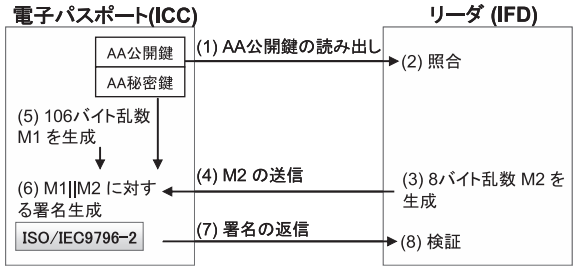

図 3　AA プロトコル
Fig. 3　AA protocol.

スポート）の導入を定めた改正旅券法が公布され，2006 年 3 月に発行が開始された．

この仕様では，IC チップ内に保存された MRTD の改ざんと，IC チップとリーダ間の通信の盗聴は防止できるが，IC チップ内のデータの他の IC チップへのコピー（クローニング）は対策できていない．実際，2006 年の BlackHat USA において，Grunwald が，ブランクのスマートカードを用い，電子パスポートのクローニングに成功している[8]．また，2009 年の BlackHat Asia では，リアルタイムで日本の IC 旅券をクローニングするデモが行われた[1]．

これに対し，次世代 e-Passport では，クローニング防止機能（AA；Active Authentication）を実現する仕様策定が進んでいる．日本でも，2009 年 4 月に情報処理通信機構（IPA）より，次世代 IC 旅券に対するセキュリティ要件書（PP；Protection Profile）が公開され[6]，次世代 e-Passport 導入の準備が進められている．

4.2　CNTW 攻撃の適用検討

次世代 e-Passport のクローニング防止機能 AA には ISO/IEC 9796-2 署名が使用される予定となっている[6]．本節では，AA に対し CNTW 攻撃の適用可能性を検討する．

AA のプロトコルを**図 3** に示す．AA では，まずパスポートリーダ（IFD；InterFace Device）が電子パスポート（ICC；IC Card）から AA 用の公開鍵を読み出す．IFD は，その公開鍵の正当性を PA により検証する．次に IFD は 8-byte の乱数 $M2$ を生成し，ICC に対し送信する．ICC は 106-byte の乱数 $M1$ を生成し，$M1$ と $M2$ を連結したメッセージ（$M = M1||M2$）を生成する．次に ICC は，生成したメッセージ M に対し，内部に保存している AA 用秘密鍵を用いて ISO/IEC 9796-2 署名を生成し，IFD に返信する．最後に IFD は，AA 用公開鍵を使用し，返信された署名を検証する．

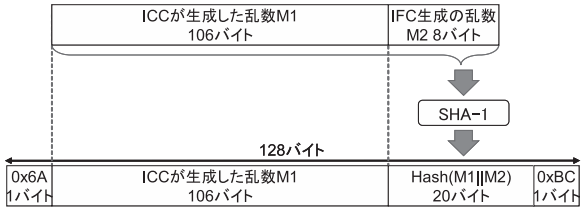

図 4　メッセージエンコード処理[6]
Fig. 4　Message encoding.

ISO/IEC 9796-2 署名のメッセージエンコード処理を**図 4** に示す[6]．文献 6) ではインターオペラビリティのために，メッセージエンコードには 1024-bit 合成数，ハッシュ関数として SHA-1 が使用されることとなっている．本仕様では，ISO/IEC 9796-2 署名は，メッセージが部分的に復元されるモード（ヘッダが `0x6A`）で使用され，復元されるメッセージは ICC が生成した乱数 $M1$ となる．これにより，乱数 $M1$ を署名と別に送信することなく，IFD で署名検証を実施することが可能となっている．

次に，AA に対して，CNTW 攻撃を適用し，AA 機能の偽造が可能かどうかを検討する．2.2 節で述べたように，CNTW 攻撃では，攻撃者はメッセージを自由に選択することができないという限界がある．CNTW 攻撃を本仕様に適用した場合，メッセージの復元される部分（$M1$ 部分）が固定されるため，攻撃者はメッセージのハッシュ値と重なる部分（$M2$ 部分）を操作し，攻撃に利用可能なメッセージを探索する．しかし AA では，耐タンパ性が期待できる ICC が $M1$ を生成するため，攻撃者が CNTW 攻撃で使用する固定値を $M1$ に設定することは困難である．以上の議論により，CNTW 攻撃を用いた AA の偽造は不可能であり，CNTW 攻撃の次世代 e-Passport への現実的な影響は無視できるという結論を得る．

5. ま と め

本稿は，ISO/IEC 9796-2 署名に対する CNTW 攻撃の詳細な評価を与えるとともに，次世代 e-Passport のクローニング防止機能 AA（Active Authentication）への適用可能性を議論した．詳細な評価のために，本評価では各ハッシュ関数に対応したパディング関数を使用し，10 万個の 2048-bit 合成数に対してパラメータ a を算出し，その平均値を使って得られた攻撃コストを使用した．結果として，CNTW 攻撃が現実的な費用で適用できるのは，ハッシュ関数が SHA-1 以下の場合であり，SHA-224 に変更した時点で CNTW 攻撃によ

る署名偽造は困難であることが判明した．また，CNTW 攻撃の AA への適用可能性については，CNTW 攻撃によって偽造署名を実際に算出できる可能性は低く，次世代 e-Passport に与える影響はきわめて小さいという結論が得られた．しかし CNTW 攻撃が改良される可能性を加味すれば，次世代 e-Passport における ISO/IEC 9796-2 署名の使用は避けるべきであろう．

2010 年問題等，他の観点から次世代 e-Passport の安全性を考えた場合，ISO/IEC 9796-2 署名の鍵長が 1024-bit である点と，ハッシュ関数 SHA-1 を使用する点は見直しが必要である．具体的には，(Scheme 1 を使用するにしても）鍵長は 2048 ビット以上，ハッシュ関数は出力長が 224 ビット以上のものを使用すべきである．

参 考 文 献

1) Beek, J.: ePassports reloaded, *Black Hat Japan 2008* (2008). Available at http://www.blackhat.com/presentations/bh-jp-08/bh-jp-08-vanBeek/BlackHat-Japa-n-08-Van-Beek-ePassports.pdf
2) Coron, J., Naccache, D. and Stern, J.: On the Security of RSA Padding, *Proc. CRYPTO 1999*, LNCS 1666, pp.1–18, Springer-Verlag (1999).
3) Coron, J., Naccache, D., Tibouchi, M., Weinmann, R.-P.: Practical Cryptanalysis of ISO/IEC 9796-2 and EMV Signatures, *Proc. CRYPTO 2009*, LNCS 5677, pp.428–444, Springer-Verlag (2009).
4) Desmedt, Y. and Odlyzko, A.: A Chosen Text Attack on the RSA Cryptosystem and Some Discrete Logarithm Schemes, *Proc. CRYPTO 1985*, LNCS 218, pp.516–522, Springer-Verlag (1986).
5) International Civil Aviation Organization (ICAO): Machine Readable Travel Documents (Doc 9303) – Part 1: Machine Readable Passport – Volume 2: Specification for Electrically Enabled Passports with Biometric Identification Capability, 6th edition (2006).
6) 情報処理推進機構：IC 旅券用プロテクションプロファイル解説書 (2009). Available at http://www.ipa.go.jp/security/fy20/reports/epassport/PP-guideVer1.0.pdf
7) International Organization for Standardization (ISO): Information Technology – Security Techniques – Digital Signature Schemes Giving Message Recovery – Part 2: Integer Factorization based Mechanisms (2002).
8) Grunwald, L.: Cloning ePassports without Active Authentication, *BlackHat USA 2006* (2006). Available at http://www.wired.com/science/discoveries/news/2006/08/71521
9) RSA Laboratories: RSA numbers. Aavailable at http://en.wikipedia.org/wiki/RSA_numbers

付　　録

A.1 SHA-1 および SHA-256 に対して最小の a を持つ合成数 N とそれに対応する各パラメータ

3.3 節の実験で使用した 10 万個の合成数 N のうち，パディング関数 $\nu_{a,b}(\cdot)$ の出力値が最も小さくなる合成数とそれに付随する各パラメータを以下に示す．

ハッシュ関数に SHA-1 を用いた場合，最小の a を持つ合成数 N およびそれに対応するパラメータ b，$m[1]$ は以下のようになる．

$$a = 5 \tag{17a}$$

$$b = 4 \tag{17b}$$

$$\begin{aligned} N = {} & 167329285651890867763973122193164504675205560009952918089334 \\ & 396499098261543814823043625495331985171438719446868192924781 \\ & 747725200276621656257506081050897171643367022452429005728048 \\ & 565728009251647647851166767686368170135668801877212787792753 \\ & 565434139929884574263520427913387082932310662513827535895540 \\ & 874115378027413272970003655081549833132051096671983134799136 \\ & 670464735884994010640535488828845356657414918114840526727907 \\ & 359960619374060364953475901703365835287295789391084271520690 \\ & 642793037658449025411204524762484533266773076998237678425101 \\ & 880015487984317741840138884167265420122122468392466874198349 \\ & 57353327532830059 \end{aligned} \tag{17c}$$

$$\begin{aligned} m[1] = {} & 135843954602103378452585739117261698692671512554290716588637 \\ & 616647812560272458698625738484338998880126715394635731040554 \\ & 925706807262683062009099069978285890913666051377499200629265 \\ & 184017635284688046510779099669301137598906029243939062250973 \\ & 319793873695852273091141165777088222297039562778212712295065 \\ & 750653004783044764999038611040004106726861962761438937624380 \\ & 919851488583552287631316943655903439443388239710057200845549 \\ & 007609255596490194584722984791376463608722518371217052102411 \\ & 712269227885707850926238723330144814244303284252275751303121 \\ & 2559961455872614 \end{aligned} \tag{17d}$$

ハッシュ関数に SHA-256 を用いた場合，最小の a を持つ合成数 N およびそれに対応するパラメータ b，$m[1]$ は以下のようになる．

$$a = 7 \tag{18a}$$

$$b = 4 \tag{18b}$$

$$\begin{aligned} N = {} & 23528712541533881062018183263105035432024766578519101716747677067279673656990557508518339318763033285472047320357083402260197735612778451808028526018552437866249720607495555470078687643235051804178428787731465519595113953690282866055260923821169646158729922728883114516722185500089745370961374850328270977062742906782553184585509023010025625002036615929758643721760557838051379026908350167555411664182130549129305168116881755352795392697987933415940177658389837779571107191759013696058080455485293829162898229938797357211111321419392392251574988745858571490324014832567669117734994088429538230940049894043398938205173 \end{aligned} \tag{18c}$$

$$\begin{aligned} m[1] = {} & 83966325233490206088050407877956956749277148249885899878669744047560136655172212684097144027745813046605457096916854120897124472113109772261541944607315185173916285433111538115285590919928652628338748073703873121885384375509802223452531313868946277699183180133526275913759032924541278157482567055824084847700949956598178678371651825339962053440755000204149690503543831814387245466901866752241952856930103176101717619538802269819926721936960709074967722824005399713943788560461123934896894261785103685927675548061475060243046635321533489689426178510368592767554806147506024304663532153 \end{aligned} \tag{18d}$$

(平成 21 年 11 月 30 日受付)
(平成 22 年 6 月 3 日採録)

推　薦　文

本稿は，ISO/IEC 9796-2 署名への偽造攻撃の計算量の評価に基づいて，次世代電子パスポートへの適用の可能性を検討している．実システムへの適用を考慮している点で，有用性が高い．具体的に数値実験は信頼できるものであり，完成度が高い．よって，十分な有用性と新規性を持つ内容であると確信し，本研究会から推薦する．

（コンピュータセキュリティ研究会主査　菊池浩明）

酒見　由美

1985 年生．2008 年岡山大学工学部通信ネットワーク工学科卒業．2009 年岡山大学大学院自然科学研究科電子情報システム工学専攻博士前期課程修了．2009 年より岡山大学大学院自然科学研究科産業創成工学専攻博士後期課程に在学，現在に至る．情報セキュリティ，暗号実装の研究に従事．2009 年コンピュータセキュリティシンポジウム（CSS 2009）学生論文賞受賞．電子情報通信学会学生会員．

伊豆　哲也（正会員）

1967 年生．1992 年東京大学理学部数学科卒業．1994 年立教大学大学院理学研究科数学専攻博士前期課程修了．1997 年立教大学大学院理学研究科数学専攻博士後期課程退学．博士（工学）．1997 年より富士通株式会社および株式会社富士通研究所に勤務，現在に至る．情報セキュリティ，暗号理論の研究に従事．2001 年 Waterloo 大学（カナダ）客員研究員．1999 年暗号と情報セキュリティシンポジウム（SCIS 1999）論文賞受賞．2002 年コンピュータセキュリティシンポジウム（CSS 2002）優秀論文賞受賞．2007 年科学技術分野の文部科学大臣表彰若手科学者賞受賞．2008 年情報処理学会喜安記念業績賞受賞．電子情報通信学会，IACR 各会員．

武仲　正彦

1967 年生．1990 年大阪大学工学部電気工学科卒業．1992 年大阪大学大学院工学研究科電気工学専攻博士前期課程修了．2009 年筑波大学大学院博士課程システム情報工学研究科コンピュータサイエンス専攻修了．博士（工学）．1992 年より富士通株式会社および株式会社富士通研究所に勤務，現在に至る．情報セキュリティ，暗号実装の研究に従事．主任研究員．2005 年電気科学技術奨励賞（オーム技術賞）受賞．電子情報通信学会，デジタル・フォレンジック研究会各会員．

野上　保之

1972 年生．1994 年信州大学工学部電気電子工学科卒業．1999 年信州大学大学院博士後期課程修了．博士（工学）．1999 年岡山大学工学部助手．2010 年岡山大学大学院自然科学研究科准教授，現在に至る．有限体基礎理論，情報セキュリティに関する研究に従事．IEEE，情報理論とその応用学会各会員．

森川　良孝

1945 年生．1971 年大阪大学大学院修士課程工学専攻修了．1971 年松下電器無線研勤務．1973 年岡山大学助手．1998 年岡山大学電気電子工学科教授．2000 年岡山大学通信ネットワーク工学科教授，現在に至る．画像通信特に画像符号化および暗号化の研究に従事．信学会信号処理ハンドブック執筆．IEEE，電子情報通信学会，ITE 各会員．