Copyright ©The Institute of Electronics,
Information and Communication Engineers

SCIS 2013 The 30th Symposium on
Cryptography and Information Security
Kyoto, Japan, Jan. 22 - 25, 2013
The Institute of Electronics,
Information and Communication Engineers

暗号ハードウェアに対する効率的なパディングオラクル攻撃

The "Million Message Attack" in 15,000 Messages
— Efficient Padding Oracle Attacks on Cryptographic Hardware

Romain Bardou [*] [†] Riccardo Focardi [‡] [§] Yusuke Kawamoto [¶] [†]
Lorenzo Simionato [‡] [‖] Graham Steel [**] [†] Joe-Kai Tsay [††] [†]

あらまし RSA 暗号標準 PKCS#1 v1.5 に対しては，1998 年に Bleichenbacher によって攻撃が発見されている．この攻撃では，攻撃者は RSA 暗号の秘密鍵を用いることなく，暗号文を復号したり，電子署名を偽造できる．この攻撃は，非常に多くのメッセージを用いることから,「ミリオンメッセージ攻撃」と呼ばれており，あまり実際的な脅威だと見なされていないのか，今日でも数多くの暗号ハードウェアが v1.5 を用いている．

本論文では，PKCS#1 v1.5 に対するより効率的な攻撃について述べ，v1.5 を用いる暗号ハードウェアの脆弱性を指摘する．この攻撃は，Bleichenbacher による攻撃を改良したもので,「パディングオラクル攻撃」のひとつである．パディングオラクルとは，復号した平文そのものを攻撃者に渡すことはないが，攻撃者から受け取ったビット列の復号に失敗したときにエラーメッセージを返すオラクルのことである．v1.5 を用いる暗号ハードウェアは，パディングオラクルとして振る舞うとき，平文の部分情報を漏らすため，攻撃者はパディングオラクルを繰り返し呼び出して平文を得ることができる．

本論文では,「ミリオンメッセージ攻撃」の改良により，攻撃に要するオラクルの呼出し回数が平均 50,000 回，中央値 15,000 回にまで減少することを明らかにする．また，この効率的な攻撃に対して，スマートカード，セキュリティトークン，エストニア eID カードといった多くの暗号ハードウェアが脆弱性を持つことを実証する．

なお，本論文は著者が CRYPTO 2012 で発表した成果の一部である．

キーワード RSA 暗号，PKCS#1 v1.5，パディングオラクル攻撃，暗号ハードウェア

1 はじめに

スマートカードや USB キーなど耐タンパ性を持つハードウェアの利用が急速に拡大している．こういった暗号ハードウェアでは，セキュリティを確保するために，暗号機能や鍵管理などの操作を API を通じて行い，利用者が実行できる操作が制限されている．利用者は，暗号ハードウェアにある秘密鍵を知ることなく，API を通じて，暗号文の復号，署名の生成，鍵の管理などの操作を実行できる．

多くの暗号ハードウェアでは，暗号操作を実現するために，RSA 暗号標準 PKCS#1 v1.5 を用いているが，本論文では，この v1.5 を用いる暗号ハードウェアに対する新たな効率的な攻撃について述べる．この攻撃では，攻撃者は，許されている限られた操作のみを用いて，暗号文を解読することができる．例えば，多くの暗号ハードウェアは，PKCS#11 をサポートしており，RSA 暗号で暗号化された対称鍵を復号してハードウェア内部のみで用いることができるが，本論文で示す新たな攻撃を用いると，攻撃者は，暗号化された対称鍵を解読する（すなわち対称鍵そのものを不正に手に入れる）ことができる．

v1.5 に対する新たな攻撃は，1998 年に Bleichenbacher によって発見された攻撃を改良したもので,「パディングオラクル攻撃」のひとつである．パディングオラクルとは，攻撃者から受け取ったビット列の復号操作に失敗したときにエラーメッセージを返すオラクルのことである．v1.5 を用いる暗号ハードウェアは，パディングオラクルとして振る舞うとき，復号操作で得られる平文の部分情報を漏らしてしまう．そのため，たとえ暗号ハードウェ

[*] INRIA SecSI, LSV, CNRS & ENS-Cachan, France
[†] Work partially carried out while at INRIA SecSI, LSV, CNRS & ENS-Cachan, France
[‡] DAIS, Università Ca' Foscari, Venezia, Italy
[§] Work partially supported by the RAS Project *TESLA: Techniques for Enforcing Security in Languages and Applications*
[¶] School of Computer Science, University of Birmingham, United Kingdom y.kawamoto at cs.bham.ac.uk
[‖] Now at Google Inc.
[**] INRIA Project ProSecCo, Paris, France
[††] Department of Telematics, NTNU, Norway

アが平文そのものを攻撃者に渡さなくても，攻撃者はパディングオラクルを繰り返し呼び出して平文を得ることができる．

Bleichenbacher による攻撃は，パディングオラクルを非常に多くの回数呼び出すことから，「ミリオンメッセージ攻撃」と呼ばれているが，本論文では，これを改良することにより，攻撃に要するオラクルの呼出し回数が中央値で約 15,000 回（平均値で約 50,000 回）にまで減少することを明らかにする．また，いくつかの暗号ハードウェアの実装が PKCS#1 v1.5 に正しく従っておらず，オラクルの呼出し回数が中央値で約 4,000 回（平均値で約 10,000 回）で済むことを明らかにする．

本研究の貢献をまとめると，以下のとおりである．

- 2 節にて，PKCS#1 v1.5 に対する新たな効率的な攻撃を示す．この攻撃は，Klima らによって提案された技法 [4] と組み合わせることにより，元の Bleichenbacher 攻撃と比較して，オラクルの呼出し回数が中央値で約 10 分の 1（平均値で約 4 分の 1）に減少する．

- 3 節にて，この効率的な攻撃に対して，スマートカード，セキュリティトークン，エストニア eID カードといった多くの暗号ハードウェアが脆弱性を持つことを示す．具体的には，我々が調査した暗号ハードウェアが，どのような種類のパディングオラクルとして機能し，攻撃にどのぐらいの時間を要するのかを示す．特に，最も脆弱なハードウェアに対しては，オラクルの呼出し回数が中央値で約 4,000 回（平均値で約 10,000 回）で済むことを明らかにする．

なお，本論文は著者が CRYPTO 2012 で発表した成果の一部をまとめたものである．より詳しい内容はテクニカルレポート [1] を参照されたい．

2 パディングオラクル攻撃

ビット列が正しいフォーマットの平文に復号されうるかどうかを教えてくれるオラクルを「パディングオラクル」という．パディングオラクルに様々なビット列を送ることによって暗号文を解読する攻撃のことを「パディングオラクル攻撃」という．

本論文では，RSA 暗号標準 PKCS#1 v1.5 に対するパディングオラクル攻撃を扱う．

2.1 PKCS#1 v1.5

n, e を RSA 公開鍵とし，d を対応する秘密鍵とする．これらは $n = pq$，$ed \equiv 1 \ (mod\ \phi(n))$ を満たす．n のバイト長を k とすると，$2^{8(k-1)} \leq n < 2^{8k}$ である．

PKCS#1 v1.5 において l バイトのメッセージ D を暗号化する手順は以下のとおりである．まず，$k-3-l$ バイトの疑似乱数列 PS を生成する．この際，PS の各バイトが 0 でないものとし，疑似乱数列の長さを 8 バイト以上とする．（したがって，データブロックのバイト長 l は $k-11$ 以下となる．）次に，以下のフォーマットのブロックを生成する．

$$\texttt{0x00}, \texttt{0x02}, PS, \texttt{0x00}, D$$

（疑似乱数列 PS とメッセージ D は 0x00 で区切られている．）このようにして得られたブロック $EB_1, EB_2, \cdots, EB_k$ を整数

$$x = EB_1 \cdot 2^{8(k-1)} + EB_2 \cdot 2^{8(k-2)} + \cdots + EB_k \cdot 2^0$$

に変換する．暗号文に対応する整数 $x^e \ mod\ n$ を計算し，これを暗号文ブロックに変換する．

暗号文ブロックを復号するには，暗号文ブロックに対応する整数 c に対して，$c^d \ mod\ n$ を計算し，これを変換して平文を得る．

2.2 Bleichenbacher 攻撃

RSA 暗号標準 PKCS#1 v1.5 に対するパディングオラクル攻撃 [2] は，1998 年に Bleichenbacher によって発見された．この攻撃では，秘密鍵 d を知らなくとも，任意の暗号文 c に対して，平文 $m = c^d \ mod\ n$ を得ることができる．攻撃者は，整数 s を選んで，$c' = c \cdot s^e \ mod\ n$ を計算し，c' をパディングオラクルに送る．仮に c' が正しいフォーマットの平文に復号される場合，$m \cdot s$ の最初の 2 バイトが 0x00, 0x02 であるため，$B = 2^{8(k-2)}$ とおくと，$2B \leq m \cdot s \ mod\ n < 3B$ となり，平文 m に関する部分情報を得ることができる．Bleichenbacher 攻撃では，攻撃者が，パディングオラクルの振る舞いに応じて，様々な整数 s を選んでパディングオラクルに送り，最終的に平文の候補をひとつに絞り込む（すなわち暗号文 c を解読する）ことができる．

平文 m が正しいフォーマットに従っている場合，

$$2B \leq m \ mod\ n < 3B$$

が成り立つ．そこで，平文の候補となる区間の集合 M_0 を $\{[2B, 3B-1]\}$ とおく．まず，$c \cdot s_1^e \ mod\ n$ が正しいフォーマットに復号されるような整数 s_1 を見つける．次に，この s_1 を用いて，平文の候補となる区間の集合 M_0 を絞り込んで M_1 を得る．

$$M_1 \leftarrow \bigcup_{(a,b,r)} \left\{ \left[\max\left(a, \left\lceil \frac{2B+rn}{s_1} \right\rceil\right), \min\left(b, \left\lfloor \frac{3B-1+rn}{s_1} \right\rfloor\right)\right] \right\}$$

ここで，M_1 は，$[a,b] \in M_0$ と $\frac{as_1-3B+1}{n} \leq r \leq \frac{bs_1-2B}{n}$ を満たす任意の (a,b,r) に対して．和を取ったものであ

る．同様に，$s_2, s_3, s_4, \cdots$ に対しても以上のプロセスを繰り返すことにより，最終的に平文の候補をひとつに絞り込むことができる．

整数 s_i を探索する方法は以下のとおりである．

(a) $i = 1$ のとき
$c \cdot (s_1)^e \ mod\ n$ が正しいフォーマットに復号されるような最小の正の整数 $s_1 \geq n/(3B)$ を見つける．

(b) $i > 1$ かつ $|M_{i-1}| > 1$ のとき
$c \cdot (s_i)^e \ mod\ n$ が正しいフォーマットに復号されるような最小の正の整数 $s_i > s_{i-1}$ を見つける．

(c) $i > 1$ かつ $|M_{i-1}| = 1$ のとき
$M_{i-1} = \{[a, b]\}$ とおく．

$$r_i \geq 2\frac{bs_{i-1} - 2B}{n} \quad \text{かつ} \quad \frac{2B + r_i n}{b} \leq s_i < \frac{3B + r_i n}{a}$$

を満たし，$c \cdot (s_i)^e \ mod\ n$ が正しいフォーマットに復号されるような小さな r_i, s_i を見つける．

直観的には，$c \cdot (s_i)^e \ mod\ n$ が正しいフォーマットに復号されるとき，ある r_i が存在して

$$2B \leq ms_i - r_i n < 3B$$

が成り立つので，$a \leq m \leq b$ より，s_i に関する条件が導かれる．

一般に，上記の (a) は 1 回だけ実行され，(b) は非常に少ない回数しか実行されない．(我々の実験では，(b) はおよそ 9 割で高々1 回だけ実行され，およそ 3 割では全く実行されない．) しかし，(a) と (b) では，s_i を見つけるために要するパディングオラクルの呼出し回数は非常に膨大なものとなる．一方，(c) の段階では，非常に効率的に s_i を見つけることができる．

署名を偽造したい場合は，上述の Bleichenbacher 攻撃の冒頭に，blind step と呼ばれる操作を行う．具体的には，メッセージ m' に対して，異なるランダムな整数 s_0 を選んで $m' \cdot (s_0)^e \ mod\ n$ をパディングオラクルに渡して正しくフォーマットに従っているかどうかを調べる．最初に成功した s_0 に対して，$c := m' \cdot (s_0)^e \ mod\ n$ とおいて，Bleichenbacher 攻撃を行う．なお，正しくフォーマットに従っている暗号文 m' を解読する場合，$s_0 = 1$ であり，blind step は不要である．

Bleichenbacher 攻撃のアルゴリズムの詳細については [2] を参照されたい．

2.3 Bleichenbacher 攻撃の改良

Bleichenbacher 攻撃の新たな改良について述べる．

2.3.1 Trimming

Bleichenbacher 攻撃ではメッセージに整数 s を掛ける操作を扱っていたが，ここでは，まず，メッセージを整数 t で割る操作（厳密には t^{-1} を掛ける操作）について考えたい．

仮に平文 m が整数 t で割り切れるならば，$m \cdot t^{-1} \ mod\ n$ は m/t であるが，割り切れないならば，一般に m/t にはならない．

命題 1 互いに素な正の整数 u，t に対し，$u < \frac{3}{2}t$ と $t < \frac{2n}{9B}$ が成り立つものとする．もし，m と $m \cdot u \cdot t^{-1} \ mod\ n$ が正しいフォーマットに従っているならば，m は t で割り切れる．

(証明) $mu < m \cdot \frac{3}{2}t < 3B \cdot \frac{3}{2}t < n$ より，$m \cdot u \ mod\ n = m \cdot u$ である．$x = mut^{-1} \ mod\ n$ とおく．x は正しいフォーマットに従っているので，$x < 3B$ である．よって，$xt < 3Bt < n$ より，$xt = xt \ mod\ n$ である．すると，$xt = xt \ mod\ n = mu \ mod\ n = mu$ より，m は t で割り切れる． □

命題 1 より，m と $m \cdot u \cdot t^{-1} \ mod\ n$ が正しいフォーマットに従っているならば，$mut^{-1} \ mod\ n = m \cdot \frac{u}{t}$ となる．したがって，$2B \leq m \cdot \frac{u}{t} < 3B$ より，$2B \cdot \frac{t}{u} \leq m < 3B \cdot \frac{t}{u}$ となる．すなわち，平文の候補の区間を狭めることができる．なお，$2B \cdot \frac{t}{u} \leq m < 3B$ より，$\frac{u}{t} > \frac{2}{3}$ を満たすように u, t を取る必要がある．

我々の攻撃では，分数 $\frac{u}{t}$ に対して，$cu^e t^{-e} \ mod\ n$ をパディングオラクルに送ることにより，$m \cdot u \cdot t^{-1} \ mod\ n$ が正しいフォーマットに従っているかどうかを調べる．これにより，平文の候補の区間を狭めることができる．

2.3.2 Trimming における分数 $\frac{u}{t}$ の選び方

平文 m が正しいフォーマットに従っている場合，一般に，分母 t が小さいほど，$m \cdot u \cdot t^{-1} \ mod\ n$ が正しいフォーマットに従っている可能性が高い．これは，ランダムな平文 m が t で割り切れる確率が $1/t$ となるからである．したがって，分母 t としては $2, 3, 4, \cdots$ といった具合に小さい順に調べていく方が効率が良い．

一方，平文の候補の区間をより狭めるには，$|\frac{u}{t} - 1|$ が大きくなるように u, t を取るのが良い．これは，前述のとおり，$2B \cdot \frac{t}{u} \leq m < 3B \cdot \frac{t}{u}$ が成り立つからである．

$|\frac{u}{t} - 1|$ を大きくするには，分母 t が大きいほど良い．そこで，我々の攻撃では，まず，(平文 m を割り切るような) できるだけ大きな分母 t を見つけてから，$|\frac{u}{t} - 1|$ ができる大きくなるような分子 u を見つけるという手順を取る．

分母 t を大きく取るためには，平文 m を割り切る分母 $t_1, t_2, \cdots, t_p$ を見つけてから，これらの最小公倍数を t と

する．なぜなら，平文 m が $t_1, t_2, \cdots, t_p$ で割り切れるならば，これらの最小公倍数でも割り切れるからである．

こうして得られた分母 t を用いて，最大の分子 u_h と最小の分子 u_l を求めることにより，$2B \cdot \frac{t}{u_l} \leq m < 3B \cdot \frac{t}{u_h}$ が得られる．

2.3.3 Skipping Holes

2.2 節で述べた Bleichenbacher 攻撃の (a) では，s_1 の探索を $n/(3B)$ から開始し，$c \cdot (s_1)^e \ mod \ n$ が正しいフォーマットに復号されるような s_1 が見つかるまで，s_1 の値を増やしていく．これは，$m \cdot s_1 \geq n + 2B$ と $m < 3B$ より，$s_1 \geq n/(3B)$ が成り立つためである．

より一般には，$c \cdot (s_1)^e \ mod \ n$ が正しいフォーマットに復号されるような s_1 に対しては，ある整数 j が存在し，

$$\frac{2B + jn}{3B} \leq s_1 < \frac{3B + jn}{2B}$$

が成り立つ．$j = 1, 2$ に関しては，

$$\frac{3B + n}{2B} < \frac{2B + 2n}{3B}$$

が成り立つ．s_1 は $\frac{3B+n}{2B}$ と $\frac{2B+2n}{3B}$ の間の値を取ることはないので，この間に関しても s_1 の探索をスキップすることができる．(n の値に依るが，これにより，数千回のオラクル呼出しを削減できる．このことは [2] には書かれていない．) 一方，$j \geq 3$ の場合に関しては，

$$\frac{3B + jn}{2B} > \frac{2B + (j+1)n}{3B}$$

となり，s_1 の探索をスキップすることができない．

ところが，2.3.1 節で述べた trimming を行い，平文の候補の区間が $[2B, 3B-1]$ から $[a, b]$ に狭まっている場合，s_1 の探索をさらにスキップすることができる．$c \cdot (s_1)^e \ mod \ n$ が正しいフォーマットに復号されるような s_1 に対しては，ある整数 j が存在し，

$$\frac{2B + jn}{b} \leq s_1 < \frac{3B + jn}{a}$$

が成り立つ．区間 $[a, b]$ が狭まるにつれて，b が小さくなるため $\frac{2B+jn}{b}$ が大きくなり，a が大きくなるため $\frac{3B+jn}{a}$ が小さくなる．すると，十分小さな $j \geq 3$ に対しても，

$$\frac{3B + jn}{a} < \frac{2B + (j+1)n}{b}$$

となる．s_1 は $\frac{3B+jn}{a}$ と $\frac{2B+(j+1)n}{b}$ の間の値を取ることはないので，この間[1]に関しても s_1 の探索をスキップすることができる．

なお，Bleichenbacher 攻撃では，s_1 を見つける段階がオラクルを最も多く呼び出す部分であり，我々の考案した trimming と skipping holes の組み合わせは，Bleichenbacher 攻撃におけるオラクルの呼出し数を大幅に削減する．

[1] これを “hole” と呼ぶことにする．平文の区間 $[a, b]$ が小さいほど，hole の数が増え，ひとつの hole が大きくなる．

2.4 様々なパディングオラクルと攻撃に要するオラクル呼出し回数

実際の暗号ハードウェアは，必ずしも PKCS#1 v1.5 を正しく実装しておらず，様々な種類のパディングオラクルに対応している．

これまで説明してきたパディングオラクルは，復号結果がフォーマットに正しく従っていないときに必ずエラーを返すが，以下では，エラーを返さない場合があるパディングオラクルについても考えたい．

- FFT オラクル: 復号結果がフォーマットに正しく従っていないときに必ずエラーを返す．(これまで説明してきたパディングオラクル)
- FTT オラクル: パディングの疑似乱数列が 8 バイトよりも短いときであっても，エラーを返さない．
- TFT オラクル: パディングの疑似乱数列 PS とメッセージを区切るバイト `0x00` が存在しない場合であっても，エラーを返さない．
- TTT オラクル: パディングの疑似乱数列が 8 バイトよりも短いときであっても，エラーを返さず，区切りのバイト `0x00` が存在しない場合であっても，エラーを返さない．

FFT オラクルは，平文 m が $2B \leq m < 3B$ を満たしていたとしても，m がフォーマットに従っていなければエラーを返すのに対して，TTT オラクルは，平文 m が $2B \leq m < 3B$ を満たしていればエラーを返さない．したがって，攻撃に要するオラクル呼出し回数は，TTT オラクルの方が FFT オラクルよりもはるかに少なくて済む．

鍵の長さが 1024 ビットのときに攻撃に要するオラクル呼出しの回数を表 1 に示す．この結果は，毎回異なるランダムな鍵と 16 バイトのランダムな平文を生成して，攻撃を 1000 回実行した結果である．我々の攻撃に要するオラクル呼出し回数の中央値[2]は，FFT の場合に約 15,000，TTT の場合には約 4,000 である．

表 1 から分かるが，中央値が平均値よりも小さい．これは，非常に多くのオラクル呼出しを要する実行がわずかに存在する一方で，ほとんどの実行ではオラクル呼出しが平均よりも少なくて済むからである．

なお，この結果では，我々の trimming と skipping holes の最適化に加えて，Klima らによって提案された以下の改良方法 [4] を併用している．

- *Parallel thread*: parallel thread 法では，節 2.2 で述べた Bleichenbacher 攻撃の (b) において，$|M_{i-1}| > 1$ の場合であっても，(c) を適用する．その際，M_{i-1}

[2] 攻撃の半数が要するオラクル呼出し数は，中央値よりも少ない．

オラクルの種類	Bleichenbacher 攻撃		我々の攻撃	
	平均値	中央値	平均値	中央値
FFF	-	-	18,040,221	12,525,835
FFT	215,982	163,183	49,001	14,501
FTT	159,334	111,984	39,649	11,276
TFT	39,536	24,926	10,295	4,014
TTT	38,625	22,641	9,374	3,768

表 1: 攻撃に要するオラクル呼出しの回数

に含まれる各区間に対して，別々のスレッドを走らせて，s_i を探索する．この手法は，[4] でも述べられているように，s_i の探索を大幅に効率化する．

- *Tighter bounds*: Bleichenbacher 攻撃では，最初の平文の候補の区間として $[2B, 3B]$ を用いていたが，パディングの疑似乱数列の各バイトが 0 でないという事実を利用して，[4] では $[2B, 3B]$ よりも少し狭い区間を用いている．しかし，実験結果を見る限り，この手法は s_i の探索をほとんど効率化させていない．

なお，Klima らの改良方法だけを用いる場合，元の Bleichenbacher 攻撃と比較して，オラクル呼出し回数の中央値が 52% （平均値は 38% ）改善する．

パディングオラクルが強力になるにつれ（表の上から下へ行くにつれて）parallel thread 法の効果が低下し，我々の skipping holes 法が効果を増すことが実験から分かっている．これは，TTT などのより強力なオラクルでは，もともと Bleichenbacher 攻撃の (b) を実行しないことが多く[3]，parallel thread 法を適用する機会がほとんどないためである．

3 暗号ハードウェアに対する攻撃

前節で述べた攻撃による，スマートカード，セキュリティトークン，エストニア eID カードの脆弱性について述べる．各暗号ハードウェアの詳細については，テクニカルレポート [1] を参照されたい．

3.1 スマートカードとセキュリティトークン

我々が入手したスマートカードとセキュリティトークンが，どのような種類のパディングオラクルとして機能し，攻撃にどのぐらいの時間を要するのかを表 2 に示す．

これらの暗号ハードウェアのうち，RSA Securid だけが PKCS#1 v2.1 （RSA OAEP）もサポートしている．一般に，PKCS#1 v1.5 の代わりに v2.1 を用いれば，我々の攻撃を防ぐことができる．しかし，RSA Securid では v2.1 と v1.5 で同じ鍵を使えるため，v2.1 で生成された暗号文を，v1.5 に基づくパディングオラクルを用いて攻撃することが可能[4]である．

Feitian と SATA DKey に関しては，どちらの実装も PKCS#1 v1.5 に従っておらず，ブロックの最上位 2 バイトが `0x00, 0x02` でない場合であっても，エラーを返さないことがある．そのため，我々の攻撃を適用できない．

我々は 2011 年 5 月までにこれらの暗号ハードウェアの脆弱性を製造元に報告し，一部から回答を得た．SafeNet 社は PKCS#1 v2.1 を用いることを計画している．RSA 社は，PKCS#1 v1.5 に正しく従うように修正することを計画している．（これにより，RSA Securid が TTT オラクルとして機能することを防ぎ，攻撃を遅くすることができる．しかし，FFT オラクルとして機能するため，攻撃自体をなくすことはできない.）Siemens 社は，最新版で PKCS#1 v1.5 に正しく従うよう修正している[5]．

3.2 エストニア eID カード

エストニアでは，現在 100 万枚以上の電子 ID（eID）カードが使われており，広く普及している．eID カードでは 2 組の RSA 鍵ペアが使われている [3]．一つ目の鍵ペアは，メッセージの暗号化と署名生成（例 : TLS/SSL における相互認証）に用いられ，二つ目の鍵ペアは，法的拘束力を持つ署名の生成に用いられる．

eID カードは，一つ目の鍵ペアに関して FFT オラクルとして機能するため，我々の攻撃を適用して暗号文を解読したり，署名を生成することができる．一方，二つ目の鍵ペアに関してはパディングオラクル攻撃を適用できない．

鍵長が 2048 ビットの場合，暗号文の解読に要するオラクル呼出し数の中央値は約 1 万回（平均値は約 3 万回）で，攻撃時間の中央値は約 10 時間（平均値は約 27 時間）である．鍵長が 1024 ビットの場合の攻撃時間の

[3] というのも，強力なオラクルでは，成功する s_1 を見つけるのにかかるオラクル数がもともと少なく，s_1 が小さいため，$|M_1| = 1$ となることがより多い．そのため，(b) を実行しないことが多い．

[4] 署名を偽造する際の blind step と同様に，暗号文に s_0^e を掛けて v1.5 のフォーマットに従う暗号文を作ってから，我々の攻撃を適用すればよい．

[5] CardOS は現在，Atos 社の製品となっている．

暗号ハードウェア	Token 鍵		Session 鍵	
	オラクル	攻撃時間	オラクル	攻撃時間
Aladdin eTokenPro	FTT	21 分	FTT	17 分
Feitian ePass 2000	×	×	×	×
Feitian ePass 3003	×	×	×	×
Gemalto Cyberflex	FFT	92 分	N/A	N/A
RSA Securid 800	TTT	13 分	N/A	N/A
Safenet Ikey 2032	FTT	88 分	FTT	17 分
SATA DKey	×	×	×	×
Siemens CardOS	TTT	21 分	FFT	89 秒

表 2: 様々な暗号ハードウェアのオラクルの種類と攻撃時間（中央値）

推定値は，中央値で約 3.5 時間（平均値で約 11.5 時間）である．また，署名の偽造に要する攻撃時間の中央値は，2048 ビット鍵で約 65 時間，1024 ビットで約 30 時間である．

4 むすび

RSA 暗号標準 PKCS#1 v1.5 に対する Bleichenbacher 攻撃を改良することにより，「ミリオンメッセージ攻撃」に要するオラクルの呼出し回数が中央値 15,000 回（平均値 50,000 回）にまで減少することを明らかにした．また，この効率的な攻撃に対して，スマートカード，セキュリティトークン，エストニア eID カードといった多くの暗号ハードウェアが脆弱性を持つことを実証した．

参考文献

[1] Romain Bardou, Riccardo Focardi, Yusuke Kawamoto, Lorenzo Simionato, Graham Steel, and Joe-Kai Tsay. Efficient padding oracle attacks on cryptographic hardware. Cryptology ePrint Archive, Report 2012/417, 2012. `http://eprint.iacr.org/2012/417.pdf`.

[2] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard. In *Advances in Cryptology: Proceedings of CRYPTO '98*, volume 1462 of *LNCS*, pages 1–12, 1998.

[3] ID Süsteemide AS. EstEID specification v2.01. `http://www.id.ee/public/EstEID_Spetsifikatsioon_v2.01.pdf`.

[4] Vlastimil Klíma, Ondrej Pokorný, and Tomás Rosa. Attacking RSA-based sessions in SSL/TLS. In *5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2003)*, pages 426 – 440. Springer-Verlag, 2003.