2006－CSEC－35 (1)
2006／12／8

PKCS#1v1.5 署名の実装における脆弱性の攻撃可能性

†金岡 晃、‡杉本 浩一
†セコム株式会社 IS 研究所 〒181-8528 東京都三鷹市下連雀 8-10-16 セコム SC センター
‡セコムトラストシステムズ株式会社 〒181-8528 東京都三鷹市下連雀 8-10-16 セコム SC センター
E-mail: †a-kanaoka@secom.co.jp、‡ko-sugimoto@secom.co.jp

概要 CRYPTO2006 の Rump Session において Bleichenbacher により、一部の PKCS#1v1.5 署名を実装するソフトウェアに脆弱性が存在することが発表された。その脆弱性は広く利用されているオープンソースソフトウェア openssl にも存在することがわかり、社会に広く影響を与えた。Bleichenbacher により示された攻撃法はいくつかの条件が必要となることが示されているが、本論文ではさらなる解析を行うことで、より詳細な攻撃条件を示した。さらに現実的に利用されている環境を考慮し、その脆弱性の脅威を正確に把握した。

The Threat on PKCS#1v1.5 Signature Implementation Vulnerability

†Akira KANAOKA, ‡Koichi SUGIMOTO
†IS Laboratory, SECOM Co.,Ltd. SC Center, 8-10-16, Simorenjaku, Mitaka, Tokyo, 181-8528 Japan
‡Secom Trust Systems Co.,Ltd. SC Center, 8-10-16, Simorenjaku, Mitaka, Tokyo, 181-8528 Japan
E-mail: †a-kanaoka@secom.co.jp, ko-sugimoto@secom.co.jp

Abstract Bleichenbacher showed that some PKCS#1v1.5 signature implemantation has vulnerability at CRYPTO 2006 Rump Session. That vulnerability enable PKCS#15v1.5 signature forgery without private key under some conditions. Though Bleichenbacher also showed some conditions to attack, there are more detailed one. In this paper, I will show more detailed conditions and consider real threat on that vulnerability.

1 はじめに

CRYPTO2006 の Rump Sessioin において、Bleichenbacher が” Forging some RSA signatures with pencil and paper”と題した発表を行った。そこではある実装のもとで PKCS#1v1.5 署名のデータ検証に問題がある場合に、プライベート鍵なしに署名データを作成することが可能というものだった。その後、オープンソースソフトウェアとして広く利用されている openssl に該当脆弱性が発見され修正版が公開されるなど、広く影響を与えることとなった[1]。

攻撃を実現するために、Bleichenbacher はいくつかの条件を提示し、その具体例も示した。しかし、Bleichenbacher が示した攻撃法は、彼が示した以外にもいくつかの条件が存在する。本論文ではそれら条件をあらためて検証し、現実的に利用されている公開鍵や PKI の状況を踏まえて脆弱性の脅威を考察する。

2 章では当該脆弱性に係わる前提知識とし

てRSAプリミティブとPKCS#1v1.5署名について解説を行う。3章ではBleichenbacherが指摘した脆弱性の詳細と攻撃の原理を解説する。4章では3章の内容を元に、さらに攻撃条件について考察を行い、その脅威を把握する。最後に5章でまとめる。

2 前提知識

本章ではBleichenbacherの示した脆弱性に係わる前提知識を解説する。

2.1 RSA暗号プリミティブ

RSA暗号の暗号化と復号化の基礎は以下のとおり。

$c = m^e \bmod n$

$m = c^d \bmod n$

ここで、mは平文、cは暗号文、nは素数p,qの積、e,dは次の関係を満たす整数値となっている。

$e \cdot d \equiv 1 \bmod LCM(p-1, q-1)$

ここで(e,n)を公開鍵、(d,n)を私有鍵と呼ぶ。eの値として65537が利用されることが多いが、演算処理高速化のため、3が利用されるケースも少なくない。また暗号の鍵長をビット数で表すことが多いが、RSA暗号では、nのビット数を指す。

2.2 RSA署名スキーム

PKCS#1[2]は最新のバージョンは2.1であり、そこでは署名スキームとして、RSASSA-PSSとRSASSA-PKCS1-v1_5の2種類の方法が規定されている。RSASSA-PKCS1-v1_5はバージョン1.5で規定された方式であり、実装としていまだ広く使われているためにバージョン2.1でも併記されている。さらにPKCS#1では署名を行う対象データのエンコード方法が、それぞれの署名スキームに対応してEMSA-PSS、EMSA-PKCS1-v1_5として規定されている。ここではバージョン1.5に対応した2つについて説明する。

2.2.1 EMSA-PKCS1-v1_5

RSA暗号を電子署名として用いる場合、署名を行うデータに直接暗号化を行うわけではなく、署名を行うデータのハッシュ値を取り、ハッシュ値に対して暗号化を行う。RSA署名スキームではさらに、ハッシュ値を求める際に利用したハッシュアルゴリズムを示すデータや、署名対象データを鍵長に合わせるためのパディングの方法がエンコーディング方法として決められている。EMSA-PKCS1-v1_5エンコーディングによるデータ構造は以下の通り。

00	01	FF	FF	・・・	FF	00	T

エンコード後のデータ長が鍵長と同じサイズになるようにFFの数を決定する。ここで、FFは少なくとも8オクテットが必要とされる。またTはハッシュ値とハッシュアルゴリズムを含んだデータであり、ASN.1構造をDERエンコーディングしたものとなっている。ハッシュアルゴリズムがSHA-1の場合、下にしめすようなデータになる。

30	21	30	09	06	05	2B	0E	03	02	1A	05	00	04	14	H

ここでHは署名を行う対象データのハッシュ値である。

2.2.2 RSASSA-PKCS1-v1_5

署名スキームでは、前小節で示したEMSA-PKCS1-v1_5を利用して署名を行う。署名の順序を以下に示す。

① 署名対象のデータをEMSA-PKCS1-v1_5でエンコードし、エンコードデータEMを得る

② エンコードデータEMを整数値に変換する

③ 変換した整数値に、RSA暗号プリミテ

ィブの復号化作業を行う

④ 得られた整数値をデータ化し、署名データを得る

3 脆弱性の詳細と攻撃の原理

Bleichenbacher が指摘した脆弱性は、前章で説明した EMSA-PKCS1-v1_5 の実装における、構造チェックの甘さに関するものである。署名データを検証する際、公開鍵で復号されたデータが EMSA-PKCS1-v1_5 の形式であるかをチェックしなければならない。考えられる構造チェック方法に以下のものがある。

① パディングのヘッダ（00 01）
② パディング中の ff の長さが 8 オクテット以上
③ パディングのフッタ（00）
④ DER エンコードされた ASN.1 構造
⑤ ハッシュアルゴリズム
⑥ ハッシュ値データサイズ
⑦ 全体のデータ長
⑧ T の後に余分なデータがないか

実装によってはこれらすべてを行っておらず、特に上記の⑦や⑧が実装されていない場合に、ある攻撃が可能になるために脆弱である、というのが Bleichenbacher の示したものである。

⑦や⑧がない場合、たとえば以下のエンコードデータもチェックを通過してしまう。

00	01	FF	FF	···	FF	00	T	余分なデータ

ここでは、ff の数を少なくし T の後に余分なデータを挿入している。

正式なエンコードデータ EM の整数値表現は以下のようになる。

$$EM = 2^{8N-15} - 2^{8(TLength+1)} + T$$

ここで、$TLength$ は T のオクテット長を示す。また、N は n のオクテット長であり、鍵長 $KeyLength$（ビット）は以下のようになる。

$$8(N-1)+1 \le KeyLength \le 8N$$

一方で、不正なエンコードデータ EM' は以下のように表現される。

$$EM' = 2^{8N-15} - 2^{8(TLength+1)+8DP} + T \cdot 2^{8DP} + G$$

ここで G は余分なデータを示す。また DP は T を配置するデータ上の位置を示し、下位から DP オクテットの部分に T が入っていることを意味する。

以下の理解を容易にするため、ここで $R = 2^{8(TLength+1)} - T$ と置き、EM' の表現を以下のように置き換える。

$$EM' = 2^{8N-15} - R \cdot 2^{8DP} + G$$

公開鍵指数 e が 3 の場合、プライベート鍵指数 d を知らずに、検証時（公開鍵による復号時）に EM' の形式になるデータ Z を生成すること、つまり署名データの作成が可能である。

$$Z^3 = EM'$$

Bleichenbacher の方法では、ここで以下の公式を利用する。

$$(a-b)^3 = a^3 - 3a^2b + 3ab^2 - b^3$$

ここで $a^3 = 2^{KeyLength-15}$ と置き a を求め、その a と $3a^2b = N \cdot 2^{DP}$ から b を求めると以下のようになる。

$$a = 2^{\frac{8}{3}N-5}$$

$$b = \frac{R}{3} \cdot 2^{8DP-\frac{16}{3}N+10}$$

求めた a,b を使い、$Z = a - b$ とすることで署名データの偽造が可能となる。

Blechenbacher は、鍵長が 3072 ビット、R の配置位置を下位から 259 オクテット目（$DP = 259$）としたときの例として以下の Z を提示している。

$$Z = 2^{1019} - \frac{R}{3} \cdot 2^{34}$$

ここで、Blechenbacher は R が 3 の倍数で

なければならない、という攻撃の条件も示している。

4　攻撃の実現可能性

Blechenbacher が示した攻撃手法において、条件とされたのが公開鍵指数 e が 3 であることと、R が 3 の倍数を持つこととなっている。しかし、攻撃を実現させるための条件はほかにも存在する。本章では 4.1 節にその条件を示し、4.2 節においてさらに現実的な実現性を考察する。

4.1　DP, N への条件

前章で求めた $Z(=a-b)$ を EM' の形に展開すると G の部分は以下のようになる。

$$G=\frac{R^2}{3}\cdot 2^{16DP-(8N-15)}-\frac{R^3}{27}\cdot 2^{24DP-2(8N-15)}$$

ここで、G に関しては EM' を構成するにあたりいくつかの条件がある。もっとも基本的なものとして、G は 0 以上の整数でなければならないことがあるが、ここでは以下の条件に注目する。

$$G<2^{8DP}$$

上式は、G の値は配置位置より大きくなってはいけないことを示している。データとして上式を見た場合、G のビット数が $8DP$ 以下でなければならない、と言い換えることができる。

G のビット数を求めるために、$\frac{R^2}{3}$ のビット数を求めると、$R=2^{8(TLength+1)}-T$ より以下の値になる。

$$\left|\frac{R^2}{3}\right|=16\cdot TLength+15$$

G を構成する 2 つの項を G_A, G_B とすると、前者の G_A は後者の G_B より大きくなければならないことから G 全体のビット数は G_A のビット数でとなるので、以下の値になる。

$$|G|=\ 16\cdot TLength+15 +16DP-(8N-15)$$

これが $8DP$ より小さいことが必要であるため、以下の式を満たす DP と N が求められる。

$$DP<N-2\cdot TLength-\frac{15}{4}$$

次に、DP の最小値の条件を求める。まず、G を以下のように変形することができる。

$$G=\frac{R^2}{3}\cdot 2^{24DP-2(8N-15)}\left(3\cdot 2^{8N-15-8DP}-\frac{R}{3}\right)$$

このことより、以下の条件が成立する。

$$24DP-2(8N-15)\geq 0$$

これより、DP の条件として

$$DP\geq\frac{2}{3}N-\frac{5}{4}$$

が得られる。

DP の条件は以下にまとめられる。

$$\frac{2}{3}N-\frac{5}{4}\leq DP<N-2\cdot TLength-\frac{15}{4}$$

さらに上式から、N の条件も下のように求められる。

$$N>6TLength+\frac{15}{2}$$

これで N の最小値の要件が求められた。また N に関しては、最小値の要件以外にも満たさなければいけない条件がある。攻撃を可能にするデータという視点で考慮すると、a 自身も整数値でなければ攻撃は事実上不可能と言ってよい。そして a が整数であるためには、N が 3 の倍数でなければならない。

4.2　現実的な実現性

4.1 節において求められた DP, N の条件について、現実的な実現性を考察する。

まず、N の最小値は $TLength$ に依存する。その $TLength$ は PKCS の仕様により、ハッシュ

関数ごとに決定される。そのため、N は利用されるハッシュ関数に依存した長さが必要となり、また N が 3 の倍数であることを加え考慮し、最小のオクテット数が決定する。またそれに従い鍵長である *KeyLength* の最小値も決定する。
それらの関連を表 1 に示す。

表 1 ハッシュ関数別の最小鍵長

ハッシュ関数	*TLength*	最小 N	最小鍵長
MD5	34	213	1697
SHA-1	35	219	1745
SHA-256	51	315	2513
SHA-384	67	411	3281
SHA-512	83	507	4049

現実的に世の中でもっとも利用されている RSA 暗号の鍵長は 1024 ビットであるが、この攻撃方法は 1024 ビットではいずれのハッシュ関数でも実現不可能なことがわかる。次に利用されている鍵長として 2048 ビットがあるが、2048 ビット（256 オクテット）の鍵は N に対する要件である 3 の倍数を満たさず、この攻撃は実現不可能となる。さらに、一部の証明書などでは 4096 ビット（512 オクテット）の RSA 鍵が利用されているが、これも 3 の倍数ではないために実現不可能である。

RSA で利用する鍵は 2 の累乗のオクテット数が多く、それらはすべて 3 の倍数ではない。

5 まとめ

本論文では、Bleichenbacher が示した RSA 署名を実装するソフトウェアの一部に存在する脆弱性の現実的な攻撃可能性を考察した。その結果、Bleichenbacher が示した以外にも攻撃には条件が存在することがわかった。その条件は以下の通りである。

- $N > 6TLength + \frac{15}{2}$
- $\frac{2}{3}N - \frac{5}{4} \le DP < N - 2 \cdot TLength - \frac{15}{4}$

またそれらの新条件を加えて、各ハッシュ関数ごとの最小鍵長を求め、現実的な攻撃可能性を考察した。

参考文献

[1] Openssl Security Advisory, “RSA Signature Forgery (CVE-2006-4339)”, Sep. 2006, http://www.openssl.org/news/secadv_20060905.txt

[2] RSA Laboratories, “PKCS #1 v2.1: RSA Cryptography Standard”, June, 2002, http://www.rsasecurity.com/rsalabs/node.asp?id=2125