Journal of Information Processing Vol. 16 122–129 (Sep. 2008)

Regular Paper

Extending Bleichenbacher's Forgery Attack[⋆1]

Tetsuya Izu,[†1] Takeshi Shimoyama[†1]
and Masahiko Takenaka[†1]

In 2006, Bleichenbacher presented a new forgery attack against the signature scheme RSASSA-PKCS1-v1_5. The attack allows an adversary to forge a signature on almost arbitrary messages, if an implementation is not proper. Since the example was only limited to the case when the public exponent is 3 and the bit-length of the public composite is 3,072, the potential threat is not known. This paper analyzes Bleichenbacher's forgery attack and shows applicable composite sizes for given exponents. Moreover, we extend Bleichenbacher's attack and show that when 1,024-bit composite and the public exponent 3 are used, the extended attack succeeds the forgery with the probability $2^{-16.6}$.

1. Introduction

In the rump session of CRYPTO 2006, held on August 2006, Bleichenbacher presented a new forgery attack [1] against the signature scheme RSASSA-PKCS1-v1_5 (PKCS#1v1.5 for short) defined in PKCS#1 [7] and RFC 3447 [9], a cryptographic standard developed and maintained by RSA Laboratories [7]. The attack allows an adversary to forge a valid signature on an (almost) arbitrary message in a very simple way, if an implementation of the signature scheme is loose, namely, a format check in the verification is not adequate. In fact, several implementations of PKCS#1v1.5 including OpenSSL, Firefox2 and Sun's JRE (Java Runtime Environment) library had this vulnerability. In response to Bleichenbacher's attack, US-CERT published a vulnerability note on September 2006 [2], and these implementations resist the attack now.

Since Bleichenbacher's presentation was limited to the case when the bit-length of the public composite n (denoted by $|n|$) is 3,072 and the public exponent e is 3, applicability to other parameters was unclear. Though Tews showed the applicability of the extended forgery attack when $|n| = 1{,}024$ and $e = 3$ [10], other cases such as $e = 17, 65{,}537$ have not been discussed yet.

In this paper, we analyze Bleichenbacher's forgery attack and show applicable composite sizes for given exponents. Then we propose the extended attack assuming the same implementational error, which is a generalization of the original attack and Tew's extended attack. For fixed n and e, the success probability of the proposed attack is $2^{(|n|-15)/e-353}$ in the random oracle model. When $|n| = 1{,}024$ and $e = 3$, the proposed attack succeeds the forgery with the probability $2^{-16.6}$ which coincides with Tew's experiment [10].

The rest of this paper is organized as follows: in Section 2, Bleichenbacher's forgery attack against PKCS#1v1.5 and analytic results are described. Then the extended attack is proposed in Section 3. Some numerical examples of forged signatures are in the appendix.

2. Bleichenbacher's Attack

This section describes Bleichenbacher's forgery attack [1] against RSASSA-PKCS1-v1_5 (PKCS #1v1.5 for short) with the loose implementation. Let n be an RSA composite whose size is denoted by $|n|$ (in bit). In the following, a variable in the `typewriter font` denotes an octet string and a variable in the Roman font denotes an integer. Two variables in the same letter correspond to each other, namely, `A` is an octet representation of an integer A, and vice versa.

2.1 RSASSA-PKCS1-v1_5

Let us introduce the signature scheme RSASSA-PKCS1-v1_5 defined in PKCS#1 [7] and RFC 3447 [9]. For a given public composite n (a product of two large primes with the same size) such that $|n|$ is a multiple of 8, a message m (to be signed) is encoded to an integer M, an integer representation of an octet string `M` defined by $\mathtt{M} = \mathtt{00}||\mathtt{01}||\mathtt{PS}||\mathtt{00}||\mathtt{T}||\mathtt{H}$ (PKCS#1v1.5 message format) where `PS` is an octet string with `ff` such that $|\mathtt{M}| = |n|$ (and $|\mathtt{PS}| \geq 64$), `T` is an identifier of the signature scheme and the hash function (**Table 1**), and `H` is an octet representation of the hash value $H(m)$. Then, a signature s is generated by $s = M^d \bmod n$ for the signer's secret integer d.

On input the original message m, its signature s and the signer's public expo-

---

†1 FUJITSU LABORATORIES Ltd.

⋆1 A part of this paper was published at the 1st International Workshop on Advances in Information Security (WAIS 2007) [4].

Table 1 Identifiers of the algorithm and the hash function [7].

Hash Function	Length (bit)	Octet String
MD2	144	`3020300c06082a864886f70d020205000410`
MD5	144	`3020300c06082a864886f70d020505000410`
SHA-1	120	`3021300906052b0e03021a05000414` $(= \mathtt{T}_{\mathrm{SHA1}})$
SHA-256	152	`3031300d060960864801650304020105000420`
SHA-384	152	`3041300d060960864801650304020205000430`
SHA-512	152	`3051300d060960864801650304020305000440`

nent e, a verifier obtains an octet string $\mathtt{M}'$ representing an integer $M' = s^e \bmod n$ and checks whether it satisfies the format

$$\mathtt{M}' = \mathtt{00}||\mathtt{01}||\mathtt{PS}||\mathtt{00}||\mathtt{T}||\mathtt{H}'.$$

Then the verifier obtains a value H', an integer representation of the octet string $\mathtt{H}'$, and compares whether $H' = H(m)$. If this equation holds, the signature is accepted by the verifier.

In the implementation level, a part of the format check is sometimes inadequate for various reasons. For example, when an octet string

$$\mathtt{00}||\mathtt{01}||\mathtt{PS}||\mathtt{00}||\mathtt{T}||\mathtt{H}'||\mathtt{garbage}$$

(a garbage data is followed) is obtained by a verifier as a decoded message, it should be rejected because it is in the illegal format. However, some implementations accept the string because they do not properly check the number of `ff` (only check $|\mathtt{PS}| \geq 64$) and they stop the scan at the end of `H'` (namely, they do not notice the existence of the garbage). Such loose implementation is the target of Bleichenbacher's forgery attack described in the next subsection.

2.2 Outline of Bleichenbacher's Attack

Next, let us introduce Bleichenbacher's forgery attack [1] against PKCS#1v1.5. Here we assume that the hash function SHA-1 and parameters $|n| = 3{,}072$ and $e = 3$ are used [3]. In the attack, an adversary chooses a message $\bar{m}$ with arbitrary bit-length such that

$$a = 2^{288} - (T \times 2^{160} + H(\bar{m}))$$

is divisible by 3, where T is an integer representation of the octet string $\mathtt{T}_{\mathrm{SHA1}}$ (as in Table 1). Note that such $\bar{m}$ can be obtained by generating $\bar{m}$ randomly (3 trials are required on average). The adversary also computes two integers

$$\begin{cases} g = a^2/3 \times 2^{1087} - a^3/27 \times 2^{102}, \\ \bar{s} = 2^{1019} - a/3 \times 2^{34}. \end{cases}$$

Observe that

$$\begin{aligned} \bar{s}^e &= (2^{1019} - a/3 \times 2^{34})^3 \\ &= 2^{3057} - a \times 2^{2072} + a^2/3 \times 2^{1087} - a^3/27 \times 2^{102} \\ &= 2^{3057} - 2^{2360} + T \times 2^{2232} + H(\bar{m}) \times 2^{2072} + g \\ &= (2^{985} - 2^{288} + T \times 2^{160} + H(\bar{m})) \times 2^{2072} + g. \end{aligned}$$

Since an integer $2^{985} - 2^{288} + T \times 2^{160} + H(\bar{m})$ corresponds to an octet string $\mathtt{00}||\mathtt{01}||\mathtt{ff...ff}||\mathtt{00}||\mathtt{T}||\mathtt{H}'$ (the number of `ff` is different from that of the original `PS`), $\bar{s}$ is a forged signature on the message $\bar{m}$, if an implementation of the verification ignores the number of `ff` and the garbage g. In the forgery, the adversary only requires to compute $H(\bar{m})$, a and $\bar{s}$. This is why the attack is called "the pencil and paper attack" [1]. Note that the adversary does not use modulus computations and thus integers n, d are not required in the forgery.

A numerical example of Bleichenbacher's forgery attack with a 3,072-bit composite and $e = 3$ is shown in Table 5 in the appendix.

2.3 Analysis

This subsection analyzes Bleichenbacher's forgery attack with general parameters. Only SHA-1 is considered in the following, however, similar attacks and analysis can be obtained for other hash functions. For simplicity, we consider the public composite n with arbitrary length (rather than a multiple of 8).

Firstly, we consider the case with general n but $e = 3$. Since the padding $\mathtt{00}||\mathtt{T}_{\mathrm{SHA1}}$ is 128-bit and the hash value is 160-bit, we use the same a as in the original attack, namely $a = 2^{288} - (T \times 2^{160} + H(\bar{m}))$ such that $3|a$. Let

$$\bar{s}(\alpha, \beta) = 2^{\alpha} - a/3 \times 2^{\beta},$$

be a forged signature. Then, we have

$$\bar{s}(\alpha, \beta)^3 = 2^{3\alpha} - a \times 2^{2\alpha+\beta} + g(\alpha, \beta)$$

for the garbage $g(\alpha, \beta) = a^2/3 \times 2^{\alpha+2\beta} - a^3/27 \times 2^{3\beta}$. Since $\bar{s}(\alpha, \beta)^3$ should be in the PKCS#1v1.5 format, we have $3\alpha = |n| - 15$, namely, $\alpha = (|n| - 15)/3$ and $|n|$ should be divisible by 3. On the other hand, since the garbage should be smaller than $2^{2\alpha+\beta}$, we have $2\alpha+\beta > 576+\alpha+2\beta-\log_2 3$, namely, $\beta < |n|/3-581+\log_2 3$. By substituting $\beta \geq 0$ in this inequality, we have a condition on n that

$$|n| > 1{,}743 - 3\log_2 3 = 1{,}738.24....$$

Consequently, Bleichenbacher's attack with $e = 3$ is applicable to the case with $|n| \geq 1{,}739$ with $|n|$ is divisible by 3. More precisely, $|n|$ can be parameterized by $|n| = 3k$ for $k \geq 580$ and β is in a form $\beta = 8\ell + 2$ $(0 \leq \ell \leq 55)$ since `PS` is a repetition of the octet string `ff`.

Next, let us discuss with general n and e. Similar to the above discussion, we set $\bar{s}(\alpha, \beta) = 2^{\alpha} - a/e \times 2^{\beta}$ for $a = 2^{288} - (T \times 2^{160} + H(\bar{m}))$ such that $e|a$ and $\alpha = (|n| - 15)/e$. Then, we have

$$\bar{s}(\alpha, \beta)^e = 2^{e\alpha} - a \times 2^{(e-1)\alpha+\beta} + g(\alpha, \beta)$$

for the garbage $g(\alpha, \beta) = a^2(e-1)/(2e) \times 2^{(e-2)\alpha+2\beta} + \ldots$. By the same discussion, we have conditions on n that

$$|n| > 576e + 15 - e\log_2\left(\frac{2e}{e-1}\right)$$

and $|n| - 15$ is divisible by e. Also, we have $0 \leq \beta < |n|/3 - 581 + \log_2 3$ and $\beta \equiv 2 \pmod 8$ on β. Especially, we have $|n| = 17k + 15$ $(k \geq 575)$ for $e = 17$, and $|n| = 65{,}537k + 15$ $(k \geq 1{,}061)$ for $e = 65{,}537$. Consequently, Bleichenbacher's attack for general e is far from feasible. Even if $e = 3$, Bleichenbacher's attack cannot be applicable to 1,024-bit (since 1,024 is smaller than 1,739) or 2,048-bit composites (since $n - 15 = 2{,}033$ is not divisible by 3, 17, 65,537).

2.4 Oiwa, et al.'s Variant

In 2007, Oiwa, et al. proposed a variant of Bleichenbacher's attack [6]. In the message format `00||01||PS||00||T||H`, `T||H` can be described by `{{OID,PF},H}` in ASN.1 language, where $\{, \ldots, \}$ denotes the enumerate type, `OID` is the hash object ID and `PF` is the parameter field. In PKCS#1, `PF` is defined as NULL. When `PF` is replaced by non-null data, though the message format is not accepted in PKCS#1, it is acceptable by an ASN.1 parser. An idea of a variant attack by Oiwa, et al. [6] is to insert the garbage into the parameter field rather than at the end of the message format. If the message format is checked by generic ASN.1 parser, the forgery will be successful. In fact, they actually forged a signature and found the vulnerability in GNUTLS ver 1.4.3 and earlier (though they are resistant to Bleichenbacher's attack).

By the same analysis, it is easily shown that Oiwa, et al.'s variant has the same ability with regard to Bleichenbacher's attack. Moreover, the same extension proposed in the next section can be possible.

3. Extending Bleichenbacher's Attack

The security of PKCS#1v1.5 relies on the hardness of factoring n and computing the e-th root mod n. A key idea of Bleichenbacher's forgery is to set the forged signature $\bar{s}$ in the special form so that upper bits of $\bar{s}^e$ are in the PKCS#1v1.5 message format by using the garbage g. In this scenario, an adversary computes the e-th power only, however, because of the speciality of the forged signature, the public composites should be large as described in the previous section. In this section, we extend Bleichenbacher's attack by using computers rather than pencils and papers. Our strategy is to obtain a forged signature in non-special forms. To do so, for a given hash value $H(\bar{m})$, we search $\bar{s}^e$ such that the e-th root over integer exists, by computing the e-th root over real numbers (note that the e-th root computation over real number is easy with computers).

3.1 Description of Proposed Forgery Attack

Let $\bar{m}$ be a message and $H(\bar{m})$ be its hash value by SHA-1. For given bit-length of the composite $|n| \geq 369$, define f as a function of $\bar{m}$ by

$$\begin{aligned} f &= 2^{|n|-15} + 15 \times (2^{|n|-23} + \ldots + 2^{|n|-79}) + T \times 2^{|n|-208} + H(\bar{m}) \times 2^{|n|-368} \\ &= (2^{192} - 2^{128} + T) \times 2^{|n|-208} + H(\bar{m}) \times 2^{|n|-368} \end{aligned}$$

where T denotes an integer representation of the octet string $\mathtt{T}_{\mathrm{SHA1}}$ (Table 1). Note that the integer $2^{192} - 2^{128} + T$ in the above equation represents an octet string `00||01||ffffffffffffffff||00||T`. Next, compute the e-th root $\sqrt[e]{f}$ as a real number and its ceiling $\lceil \sqrt[e]{f} \rceil$. If the difference $g = \lceil \sqrt[e]{f} \rceil^e - f$ is smaller than $2^{|n|-368}$, the forgery succeeds, since the forged signature $\bar{s} = \sqrt[e]{f+g} = \lceil \sqrt[e]{f} \rceil$ is valid on the message $\bar{m}$ with the garbage g. If g is not small, change the message $\bar{m}$ until the forgery succeeds (see **Fig. 1**).

Let us analyze the proposed forgery attack with general n and e. Assume that upper $(160-t)$ bits of $H(\bar{m})$ and the corresponding part in $\lceil \sqrt[e]{f} \rceil^e$ coincides each other (here, t is a parameter determined by n and e later). Since f is $(|n|-15)$-bit,

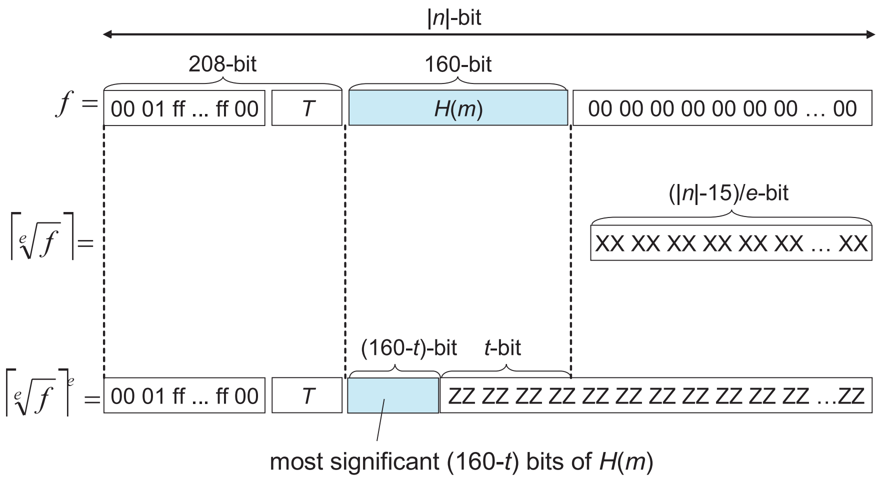

Fig. 1 Outline of the proposed forgery attack.

the integer part of $\sqrt[e]{f}$ is $(|n|-15)/e$-bit and $\lceil\sqrt[e]{f}\rceil^e$ is $e\cdot(|n|-15)/e$-bit. By the above assumption, we have a condition $|n| > 208+(160-t)+(e-1)(|n|-15)/e$, namely,

$$|n| > (353-t)e+15. \quad (1)$$

Here we implicitly used the random oracle assumption. Especially, when $|n| =$ 1,024 and $e = 3$, this condition implies that $t > 50/3 \approx 16.6$. That is, in order to forge a signature with 1,024-bit composites, the proposed forgery attack succeeds with the probability $2^{-16.6}$, namely, $2^{16.6}$ messages are required to forge a signature, which is feasible in practice. Note that the proposed attack is a generalization of the extension by Tews [10] in which $275{,}992 \approx 2^{18.1}$ messages are required in the experiment.

In the above construction, the number of the octet `ff` in f was fixed to 8 (this is minimum for the forgery). Similar construction is possible with more octets than 8, but requires larger composites instead.

As an example of the proposed forgery attack, a forged signature on the message $\bar{m}$ ="`00002e36`" (as a binary data with big endian) with $|n| = 1{,}024$ and $e = 3$ is shown in **Table 2**, where underlined octets correspond to the hashed value $H(\bar{m})$ and masked octets correspond to the garbage g. Here, the messages

Table 2 A forged signature by the extended forgery attack (1,024-bit, $e = 3$).

$\bar{m}$	"00002e36"
$H(\bar{m})$	701f0dd6 f28a0bab 4b647db8 ddcbde40 1f810d4e
f	0001ffff ffffffff ffff0030 21300906 052b0e03 021a0500 0414701f 0dd6f28a 0bab4b64 7db8ddcb de401f81 0d4e0000 00000000
$\bar{s}$	00000000 0001428a 2f98d728 ae220823 1fc5cff6 ac440735 9b078378 24846240 4cebfc71 5690f34c 7119d1da 99227fd0
$\bar{s}^e$	0001ffff ffffffff ffff0030 21300906 052b0e03 021a0500 0414701f 0dd6f28a 0bab4b64 7db8ddcb de401f81 0d4e06dd 391b3fd4 ace323ee de903694 dd78887f 5f8a73e0 5ea698ae 72a6bdfa cb7c359e 1f78cbee 96939eea 4d9b8f3e 47aebae3 90f4fe61 73ef7535 80c4cb88 edd95623 84b7e5ed ccc19fa3 ca64c0a2 a37e5000

were incrementally generated (as integers) from "`00000000`", and $0x00002e36 =$ $11{,}830 = 2^{13.53}$ messages were generated until the forgery succeeds.

3.2 Special Cases

Let us consider two special cases of the proposed forgery attack, namely $t = 0$ or $t = 160$ cases.

When we set $t = 0$, the forgery attack always succeeds. In this case, the condition (1) implies

$$|n| > 353e+15. \quad (2)$$

Even when $e = 3$, this condition implies that $|n| > 1{,}074$ which is beyond 1,024. Also, we have $|n| > 6{,}017$ for $e = 17$ and $|n| > 23{,}134{,}577$ for $e = 65{,}537$. Since this case only uses the garbage space, it allows a forgery on arbitrary chosen messages with smaller composites than the original attack. In particular, this attack does not require a condition on the target message $\bar{m}$ and the attack always succeeds. As a numerical example, a forged signature on the message $\bar{m}$ ("pkcs-1v2-1.doc" [8]) with $|n| = 1{,}152$ and $e = 3$ in Table 6 in the appendix, which succeeds a forgery for $e = 3$. Note that Bleichenbacher's original attack cannot forge for 1,152-bit composites nor the exponent $e = 3$.

On the other hand, when we set $t = 160$, the attack becomes most powerful but the adversary can not control the hash value at all. In this case, the condition (1) implies

Table 3 A comparison of forgery attacks.

	Bleichenbacher's Attack	Proposed Attack		
		$t = 0$	General t	$t = 160$
$M(e)$	$576e + 15 - e\log_2\left(\frac{2e}{e-1}\right)$ $\lvert n\rvert - 15$ is divisible by e	$353e + 15$	$(353 - t)e + 15$	$193e + 15$
$M(3)$	1,740	1,075	$1{,}075 - 3t$	595
$M(17)$	9,790	6,017	$6{,}017 - 17t$	3,297
$M(65{,}537)$	37,683,790	23,134,577	$23{,}134{,}577 - 65{,}537t$	12,648,657
Success Probability	1	1	2^{-t}	2^{-160}

$$|n| > 193e + 15 \tag{3}$$

which is obtained by substituting $t = 160$ into the condition (1). Consequently, we have $|n| > 595$ for $e = 3$, $|n| > 3{,}297$ for $e = 17$ and $|n| > 12{,}648{,}657$ for $e = 65{,}537$. However, since the adversary can not control the hash value, the success probability (in the sense that the adversary obtains the target message $\bar{m}$) is 2^{-160} which is beyond feasible. Another forged signature on the hash value

H = `7fa66ee7 e5cc4a9f bd6e13a8 11d298c2 6b9b3302`

with $|n| = 4{,}096$ and $e = 17$ in Table 7 in the appendix, which succeed a forgery for $e = 17$, however, the success probability is 2^{-113}. Note that Bleichenbacher's original attack cannot forge for 1,024-bit composites nor the exponent $e = 17$.

3.3 Comparison

A comparison between the original and proposed attacks are shown in **Table 3**, where $M(e)$ denotes the minimum bit-length of the composites to which the attack succeeds with a general exponent e. Since exponents $e = 3$, 17, 65,537 are widely used, corresponding values $M(3)$, $M(17)$, $M(65{,}537)$ are also included in the comparison. As in the table, the proposed attack with $t = 160$ forges with smallest composites. In particular, it only forges for $|n| = 1{,}024$ (with $e = 3$).

4. Concluding Remarks

This paper analyzes Bleichenbacher's forgery attack against the signature scheme RSASSA-PKCS1-v1_5 (PKCS#1v1.5) with the implementation error, and proposed an extended attack. Searching loose implementations which accept these forgeries is also required.

Table 4 RSA parameters (3,072-bit).

n	d9057e4d 2e231c66 f0a35c2c b7eddb75 04b181d6 535b81b3 83eb4765 1d76950d
	76c0c513 9efc0933 16255a5a a958007c 1b698c4c 2641418a dab6419f 8c8cf6a9
	ac799a12 7b0ec916 b5837e9c 0ecb3dc3 9629427c 08b9b076 1014d3fb c2d6d26f
	aade8a49 7aa8b03a 8e0fa396 6f6b54bd 2735a972 85cbaaed 4760ff5c 7c8b4fe2
	3d6c053c 69d0fa64 ef3ec8ad 4fa03c16 9b8e5a68 466f7dbb 1f05f6ec caf9706c
	d524b148 c41ccb67 512bcf40 b6456321 1a420f22 fedeaf1a 44ff940d eeec2117
	9ce14bec 73b5b294 f0723d03 3a810ac3 a98dc56a a9e94eca 798c2033 3fa79eb8
	ea10d25b cca36cc2 b14f4c53 3c42560c aafbb7c6 5d524591 68f8b4e0 99351f23
	8f5fbf52 ee002fb8 240f7323 938207e3 59a17330 b7df56ef e8660f9a 5cc319ce
	d3d93f25 84f5e42a 80f0acdd dec65d4d 629e2250 cbbb06f5 7ceab655 b22216d7
	9120bdc9 216310be 4c3b81ea 92017a0b 8205e92d afb9c402 9b0f4603 2a847f67
	ba0c271e a3c8f60d 5c48f4fe 22e0d3e9 3b72e9ce 1e5191bc 6167decd cde29c89
d	90ae5433 74176844 a06ce81d cff3e7a3 5876568e e23d0122 57f22f98 be4f0e08
	f9d5d8b7 bf52b0cc b96e3c3c 70e555a8 12465d88 1980d65c 91ced66a 5db34f1b
	c8511161 a75f30b9 ce57a9bd 5f32292d 0ec62c52 b07bcaf9 600de2a7 d739e19f
	c73f06db a71b2027 095fc264 4a478dd3 6f791ba1 ae87c748 da40aa3d a85cdfec
	28f2ae28 468b5198 9f7f3073 8a6ad2b9 bd09919a d99fa927 6a03f9f3 31fba048
	8e187630 82bddcef 8b728a2b 242e4216 11815f6c a9e9ca11 83550d5e 9f48160e
	83a1eb18 fd99ce23 eb14095c 333f0375 747bec29 cbe110e8 4aee7d3b 98b0e20a
	53586ce9 319c9857 50fd3c8f 7cc6613f 773748a6 9aa5550c fb691771 f5921b52
	dedacd6c cbcee703 1663f656 2019fcd7 2fcd66f5 2f6b5d86 f148f420 5eed94b1
	170937ea 8cd536d2 932435c3 adb9d529 98ab1613 8a24f1e2 c9b1c7ad cd57713c
	c08f4ccf d8a2dc47 681ef8c0 3fe709d9 52dd12ca edbba76c 21629613 fe8e0343
	193e73a5 26533256 aedda14e 6517c092 52a66013 4a2acb98 2c5e2ec1 9fdd7cab
e	03

References

1) Bleichenbacher, D.: Forging Some RSA Signatures with Pencil and Paper, presentation in the rump session, *CRYPTO 2006* (Aug. 2006).
2) US-CERT: Multiple RSA Implementations Fail to Properly Handle Signatures, Vulnerability Note VU#845620 (Sep. 5, 2006). http://www.kb.cert.org/vuls/id/845620
3) Finney, H.: Bleichenbacher's RSA Signature Forgery Based on Implementation Error, e-mail (Aug. 27, 2006). http://www.imc.org/ietf-openpgp/mail-archive/msg14307.html
4) Izu, T., Takenaka, M. and Shimoyama, T.: Analysis on Bleichenbacher's Forgery Attack, *WAIS 2007*, pp.1167–1174, IEEE Computer Society (Apr. 2007).
5) NTT Communications, Certificates for the Internal Credit Application CA (in Japanese). http://www.ntt.com/creditca/x509.pdf
6) Oiwa, Y., Kobara, K. and Watanabe, H.: A New Variant for an Attack Against

Table 5 Valid and forged signatures (3,072-bit, $e = 3$).

m	"pkcs-1v2-1.doc" [8]
$H(m)$	f7497dac 551ec010 2f0da8f1 bc8cad52 f93476c3
s	8e33fd97 65de866e 6af1c2ee 0beea1fc 26f7207c 3c9881ef f37876a0 6332d88c
	526f8102 93d21d6e 392c248a 1d2b0d6f 2f8ade54 29420bdb 78bd384c 7ef5a52f
	2249759e 1edef3f3 88f5d67f c53e8e68 f3dcb403 59716aca 1c3d911d 73fb031d
	8cb7b0d3 c3b4a378 02ad1ad5 595859e9 1bd61f51 95e7c275 cc0bfe93 96aee5d2
	69474578 7f8b2488 95fd7676 d1dbd964 50cf6ad6 10869c65 aa1520df 508a4376
	354b27b5 49677f28 5bcd54e3 b4c3aaa9 1225a955 7e630201 3343b6f8 56de4cbd
	af8e227e 4c755675 71c86627 af4ea910 8ecc1d1f 00331169 597d31b5 2028877c
	3904b4c1 03077f11 fe4cf28a 79e41bf3 473083ce af4039ae aa92ac62 2826fc90
	aef29c49 66bfc99c 01421130 d2b6313d 07031652 1862e9d5 fb3715e7 00fc168b
	abc17ac4 c3b1a83c abe59ab6 34e29539 0c51fafa 685aeeb9 c53aa717 c2cb3960
	eae314b8 ba09ef93 bef18bea 59502641 08e31ffc 569ed6aa b3f145f8 d0e82466
	8d2ca851 e6a279c7 474387ea 3d300923 dbbaa193 a0baf928 2668fa60 469ecc14
$s^e \bmod n$	0001ffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff 00302130
	0906052b 0e03021a 05000414 f7497dac 551ec010 2f0da8f1 bc8cad52 f93476c3
$\bar{s}$	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	07ffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff fffffffe aaead6ea b6b2b18e
	bd595822 b1555ac6 9f0ca790 717e556a e9678bec fb663c6e a19b4904 00000000
$\bar{s}^e$	0001ffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ff003021 30090605
	2b0e0302 1a050004 14f7497d ac551ec0 102f0da8 f1bc8cad 52f93476 c3000000
	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
	00000000 00000000 00000000 00000000 2a9aa11c bb60cb35 cb569ddd 576c2729
	34a1298d 905793b0 24ba9a39 7f041398 a7622310 78e8099f 87faed46 0fbb8f46
	67ace20c a1940f81 bced58bf 9ac3671c a2551f73 4cb80ec1 7fffffff ffffffff
	ffffffff fffffffd a285694c d9347ab7 528d15f9 d0dbf0cc 704f592f da3facc6
	210397ee 5d034b6d 269467e8 329d478c 53a8e99d 80f0732a 05d709d4 00e7ada7
	7ddc41a8 e640296f b2a8eae6 f4888211 591f0578 a07d6ec4 f147f08e ccb06340
	4439cb38 fc8144b0 cb0e382b 65583078 a7e9b040 00000000 00000000 00000000

RSA Signature Verification using Parameter Field, *EUROPKI 2007*, LNCS 4582, pp.143–153, Springer-Verlag (June 2007).

7) RSA Laboratories: RSA PKCS #1 v2.1: RSA Cryptography Standard (June 14, 2002).

8) RSA Laboratories: RSA PKCS #1 v2.1: RSA Cryptography Standard (in Word) (June 14, 2002). ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.doc

9) RSA Laboratories: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, RFC 3447 (Feb. 2003). http://www.ietf.org/rfc/rfc3447.txt

10) Tews, E.: Real World Exploit for Bleichenbacher's Attack on SSL, e-mail submitted to the Cryptography Mailing List (Sep. 14, 2006). http://www.imc.org/ietf-openpgp/mail-archive/msg14307.html

Appendix

A.1 Numerical Example of Bleichenbacher's Forgery

We show a numerical example of Bleichenbacher's original forgery attack [1]. For obtaining a valid signature, a 3,072-bit composite n and a secret key d were generated by OpenSSL as in **Table 4**. We used the hash function SHA-1 and a public key $e = 3$ and chose a digital file "pkcs-1v2-1.doc" [8] as a message m (whose corresponding a is divisible by 3, fortunately).

A valid signature s on the message m, and a forged signature $\bar{s}$ on the same message m are shown in **Table 5**, where underlined octets correspond to the hashed value $H(m)$ and masked octets correspond to the garbage g. Comparing s^e and $\bar{s}^e$ in Table 5, all octets are same except the number of the octet `ff` and the garbage. Thus, if an implementation ignores these differences, the forged

Table 6 A forged signature by the extended forgery attack (1,152-bit, $e = 3$, $t = 0$).

$\bar{m}$	"pkcs-1v2-1.doc" [8]
$H(\bar{m})$	f7497dac 551ec010 2f0da8f1 bc8cad52 f93476c3
$\bar{s}$	00000000 07ffffff ffffffff feaaead6 eab6b2b1 8e848b2b fc6229a1 298029f9 27529629 bb642126 87226bf8 913ab27d 52295002
$\bar{s}^e$	0001ffff ffffffff ffff0030 21300906 052b0e03 021a0500 0414f749 7dac551e c0102f0d a8f1bc8c ad52f934 76c30000 008e30ab d25ce35d 65cd0c25 1fc29df3 37419efd 4d08694d f3b45d86 42970cbe ef3cb225 c0e88433 552da1d0 dc35aaa1 73f1189f e0b341fc 56d5c5ea 45db5483 15e79d2a 71b6235a 44891287 00bb02f9 ffabe940 83af15c8 eabb0c30 2fefc008

signature $\bar{s}$ is accepted in the verification. Actually, OpenSSL 0.9.7f accepts the forged signature $\bar{s}$ on the message m.

A.2 Numerical Example of Proposed Forgery with $t = 0$

As an example of the proposed attack with $t = 0$, we show a forged signature on the message $\bar{m}$ ("pkcs-1v2-1.doc" [8]) with $|n| = 1{,}152$ and $e = 3$ in **Table 6**. Note that this case succeeds a forgery for 1,152-bit composites while the original attack cannot. Also note that a certificate with $|n| = 1{,}152$ and $e = 3$ is used in practice [5].

A.3 Numerical Example of Proposed Forgery with $t = 160$

As an example of the proposed attack with $t = 160$, we show a forged signature on the hash value $H(\bar{m})$ with $|n| = 4{,}096$ and $e = 17$ in **Table 7**. Note that this case succeeds a forgery for $e = 17$ while the original attack cannot. However, the adversary cannot obtain the message $\bar{m}$ in practice.

(Received November 27, 2007)
(Accepted March 4, 2008)
(Released September 10, 2008)

Table 7 A forged signature by the extended forgery attack (4,096-bit, $e = 17$, $t = 160$).

$H(\bar{m})$	7fa66ee7 e5cc4a9f bd6e13a8 11d298c2 6b9b3302
$\bar{s}$	00000000 00010aa7 58ccbbf7 7d970c35 9e1c3dc0 f20d32ad 2cf9e18a 463ea7c6 346e7f90
$\bar{s}^e$	0001ffff ffffffff ffff0030 21300906 052b0e03 021a0500 04147fa6 6ee7e5cc 4a9fbd6e 13a811d2 98c26b9b 3302448a 78e5e262 89a4190f 7d18916a 7aaaf897 feeb1e94 5866a030 208c1f48 2c906901 5f70eb66 97253c87 49790ff7 c175fc06 bddf8bb4 d2ba1cdd c626336a dda2165c dc3f425a 12cc59bc be11883e bbccc73a 0d130b94 83ac2a29 19850778 f066ff4f 374e7a96 f4fb3343 fd397d9c f7a1b8ce 16340da6 f9876f1f cca76cb4 7bfb368b a95a5842 e99c0bfb a2de62cf dbf2c635 c2c268f3 2dc228f7 2f0ebfe2 776dae35 3b82b9d9 474777ed c85eed79 e147fa2b 7500f1d4 23189a7b 9b08abb6 0df908f0 7c1c0fbb 528b3e22 df358b24 8bef05b8 f2449d0b f3fb6dc6 31a809ed 31000210 3df7ae2e 80f3f822 ae5a9f69 2948a2b5 a4529bf0 2b30fc99 1874a25f 28b5de4d 4f9c76cc 419a6848 4536e2fe 2771af8b 989e5fef 1a3aaeea f1694ebf 36e8685c 7f65eff8 b99d956b 676b5a5d f68c4519 330b4b7b 82037bd7 502d7823 4e952ba7 b9662cc2 e4389d00 76e16a47 e3dad8af e7f86e37 f164aa90 b377dfbc 9d5cc1a4 e1a966fe 3902fea5 2526240b 99ecf6b3 ced8e16e 2d085131 e5ca1676 25459ca0 0821ff8e 03cde17d 3509de96 cbe40f6f 97d5dd5b b7c977fa be2be4f5 79abcbf7 7093ad52 c346371b 5b2708fc 8b831412 9a023cfc 6b2ff020 105db3ac ef80a605 e3c1ea94 d0af9790 00000000 00000000

Tetsuya Izu received his B.S. and M.S. degrees in mathematics from the University of Tokyo in 1992, and from Rikkyo University in 1994 respectively. He received his Ph.D. in engineering from the University of Electro-Communications in 2007. He has been engaged in research on cryptography and information security at FUJITSU LABORATORIES Ltd. since 1997. He was a visiting researcher at the University of Waterloo, Canada, in 2001. He received the Paper Prizes of the Symposium on Cryptography and Information Security (SCIS) in 1999 and the Computer Security Symposium (CSS) in 2002. He was awarded the Young Scientists' Prize by the Minister of Education, Culture, Sports, Science and Technology Research on side channel attacks and countermeasures in information security in 2007. He received the IPSJ Kiyasu special industrial achievement award in 2008. He is a member of IACR, IEICE, IPSJ and JSIAM.

Takeshi Shimoyama received his B.S. and M.S. degrees in mathematics from Yokohama City University in 1989 and 1991, respectively, and D.E. degree in electrical engineering from Chuo University in 2000. He has been a research engineer of FUJITSU LABORATORIES Ltd since 1991. He joined the Research Project of Info Communication Security under the Telecommunications Advancement Organization of Japan from 1996 to 1998. His current research interests are in cryptanalysis and information security. He was awarded the SCIS paper prize in 1997, the IWSEC paper prize in 2007, and the OHM Technology Award in 2007. He attained the world record in integer factoring by GNFS in 2006.

Masahiko Takenaka received his B.E. and M.E. degrees in electronic engineering in 1990, 1992 respectively from Osaka University, Osaka, Japan. Since 1992, he has been engaged in research and development on cryptography, side channel analysis and network security at FUJITSU LABORATORIES Ltd. He is currently a senior researcher. He was awarded the CSS 2002 Paper Prize in 2002, and the OHM Technology Award in 2005. He is a member of IEICE.