Efficient Padding Oracle Attacks On Cryptographic Hardware

or The Million Message Attack in 15 000 Messages

Graham Steel
joint work with R. Bardou, R. Focardi, Y. Kawamoto, L. Simionato, J. Kai-Tsay

BLUF (Bottom Line Up Front)

We've been researching the security properties of cryptographic hardware APIs for some time (see e.g. CCS'10)

BLUF (Bottom Line Up Front)

We've been researching the security properties of cryptographic hardware APIs for some time (see e.g. CCS'10)

One barrier to satisfactory results on existing hardware is their use of RSA PKCS#1v1.5 for encrypted key import

BLUF (Bottom Line Up Front)

We've been researching the security properties of cryptographic hardware APIs for some time (see e.g. CCS'10)

One barrier to satisfactory results on existing hardware is their use of RSA PKCS#1v1.5 for encrypted key import

Perhaps Bleichenbacher's 'Million Message Attack' is not considered a practical threat?

BLUF (Bottom Line Up Front)

We've been researching the security properties of cryptographic hardware APIs for some time (see e.g. CCS'10)

One barrier to satisfactory results on existing hardware is their use of RSA PKCS#1v1.5 for encrypted key import

Perhaps Bleichenbacher's 'Million Message Attack' is not considered a practical threat?

We devised a way to execute the MMA in a median of 15 000 messages

Perhaps this will encourage the removal of PKCS#1v1.5 padding from standards

PKCS#1 v1.5 Encryption

PKCS#1 v1.5 Encryption

Let n, e be an RSA public key and d be the corresponding private key, i.e. $n = pq$ and $ed \equiv 1 \ (mod\ \phi(n))$.

Let k be the byte length of n, so $2^{8(k-1)} \leq n < 2^{8k}$.

PKCS#1 v1.5 Encryption

Let n, e be an RSA public key and d be the corresponding private key, i.e. $n = pq$ and $ed \equiv 1 \ (mod \ \phi(n))$.

Let k be the byte length of n, so $2^{8(k-1)} \leq n < 2^{8k}$.

Suppose we want to encrypt plaintext P of length $l (< k - 11)$.

Generate $k - l - 3$ pseudorandom non-zero padding bytes PS

PKCS#1 v1.5 Encryption

Let n, e be an RSA public key and d be the corresponding private key, i.e. $n = pq$ and $ed \equiv 1 \ (mod\ \phi(n))$.

Let k be the byte length of n, so $2^{8(k-1)} \leq n < 2^{8k}$.

Suppose we want to encrypt plaintext P of length $l\,(< k - 11)$.
Generate $k - l - 3$ pseudorandom non-zero padding bytes PS

Padded block for encryption is

$$\mathrm{0x00}, \mathrm{0x02}, PS, \mathrm{0x00}, P$$

Bleichenbacher Attack (CRYPTO'98)

Want to attack ciphertext c and discover $m = c^d \ mod \ n$
Assume access to a padding oracle.

Bleichenbacher Attack (CRYPTO'98)

Want to attack ciphertext c and discover $m = c^d \ mod \ n$
Assume access to a padding oracle.

Choose integers s, send $c' = c \cdot s^e \ mod \ n$, to the padding oracle.

Oracle will decrypt to give $m' = m \cdot s$

Bleichenbacher Attack (CRYPTO'98)

Want to attack ciphertext c and discover $m = c^d \ mod \ n$
Assume access to a padding oracle.

Choose integers s, send $c' = c \cdot s^e \ mod \ n$, to the padding oracle.

Oracle will decrypt to give $m' = m \cdot s$

If m' is valid, the first two bytes of $m \cdot s$ are $\texttt{0x00}, \texttt{0x02}$.

Let $B = 2^{8(k-2)}$, then we have

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

Narrowing Plaintext Range

Initial interval M_0 is $[a, b] = [2B, 3B - 1]$

After s_i is found, let

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

for all $[a, b] \in M_{i-1}$ and $\frac{as_i - 3B + 1}{n} \leq r \leq \frac{bs_i - 2B}{n}$.

Narrowing Plaintext Range

Initial interval M_0 is $[a, b] = [2B, 3B - 1]$

After s_i is found, let

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

for all $[a, b] \in M_{i-1}$ and $\frac{as_i - 3B + 1}{n} \leq r \leq \frac{bs_i - 2B}{n}$.

Intuition: solve $m \cdot s_i = r \cdot n + t$ where $2B \leq t < 3B$

Original Attack Algorithm

Step 2.a If $i = 1$, then search for the smallest positive integer $s_1 \geq \lceil (n + 2B)/b \rceil$ such that $c_0 \cdot s_1^e \ mod \ n$ is PKCS conforming.

Step 2.b - Searching with more than one interval left If $i > 1$ and $|M_{i-1}| > 1$, then search for the smallest integer $s_i > s_{i-1}$ such that $c_0 \cdot s_i^e \ mod \ n$ is PKCS conforming.

Step 2.c - Searching with one interval left If $i > 1$ and $|M_{i-1}| = 1$, i.e., $M_{i-1} = \{[a, b]\}$, then choose small integers r_i, s_i such that

$$r_i \geq 2\frac{bs_{i-1} - 2B}{n}$$

$$\frac{2B + r_i n}{b} \leq s_i < \frac{3B + r_i n}{a}$$

until $c_0 \cdot s_i^e \ mod \ n$ is PKCS conforming.

Step 3 - Narrowing the set of solutions (as above)

Step 4 - Computing Solution If $M_i = [a, a]$, then set $m \leftarrow a$, and return m as solution of $m \equiv c^d \ mod \ n$. Otherwise, set $i \leftarrow i + 1$ and continue with Step 2.b or Step 2.c.

Complexity and Existing Optimisations

Bleichenbacher estimated 2^{20} steps (hence name of attack) for arbitrary plaintexts

Complexity and Existing Optimisations

Bleichenbacher estimated 2^{20} steps (hence name of attack) for arbitrary plaintexts

In case m already valid plaintext, we obtained mean 215k, median 163k with original algorithm (1024 bit modulus).

Complexity and Existing Optimisations

Bleichenbacher estimated 2^{20} steps (hence name of attack) for arbitrary plaintexts

In case *m* already valid plaintext, we obtained mean 215k, median 163k with original algorithm (1024 bit modulus).

Observation: in step 2c find hits much faster than 2b or 2a

Complexity and Existing Optimisations

Bleichenbacher estimated 2^{20} steps (hence name of attack) for arbitrary plaintexts

In case *m* already valid plaintext, we obtained mean 215k, median 163k with original algorithm (1024 bit modulus).

Observation: in step 2c find hits much faster than 2b or 2a

Existing optimisation due to Klima, Pokorny & Rosa:
in step 2b, use 2c formula in parallel on each interval

Complexity and Existing Optimisations

Bleichenbacher estimated 2^{20} steps (hence name of attack) for arbitrary plaintexts

In case *m* already valid plaintext, we obtained mean 215k, median 163k with original algorithm (1024 bit modulus).

Observation: in step 2c find hits much faster than 2b or 2a

Existing optimisation due to Klima, Pokorny & Rosa:
in step 2b, use 2c formula in parallel on each interval

Our idea: try to use 2c like reasoning on step 2a.
Problem: bounds collapse.

Proposition

Let u and t be two coprime integers such that $2t < u < 3t$ and $1 < t < n/(9B)$. If m and mut^{-1} mod n are PKCS conforming, then m is divisible by t.

Proposition

Let u and t be two coprime integers such that $2t < u < 3t$ and $1 < t < n/(9B)$. If m and mut^{-1} *mod* n are PKCS conforming, then m is divisible by t.

Proof

We have $mu < m3t < 3B3t < n$.

Thus, $mu \; mod \; n = mu$.

Proposition

Let u and t be two coprime integers such that $2t < u < 3t$ and $1 < t < n/(9B)$. If m and mut^{-1} mod n are PKCS conforming, then m is divisible by t.

Proof

We have $mu < m3t < 3B3t < n$.
Thus, mu mod $n = mu$.

Let $x = mut^{-1}$ mod n.

We know $x < 3B$ since it is conforming.
Thus $xt < 3Bt < n$ and so xt mod $n = xt$.
Now, $xt = xt$ mod $n = mu$ mod $n = mu$
which implies t divides m.

Using the Proposition

If we find u and t such that for a PKCS conforming m, mut^{-1} mod n is also conforming

Then we know that m is divisible by t and mut^{-1} mod $n = mu/t$.

As a consequence

$$2Bt/u \leq m < 3Bt/u.$$

Using the Proposition

If we find u and t such that for a PKCS conforming m, mut^{-1} mod n is also conforming

Then we know that m is divisible by t and mut^{-1} mod $n = mu/t$.

As a consequence

$$2Bt/u \leq m < 3Bt/u.$$

Note can test with $c' = c \cdot u^e \cdot t^{-e}$ mod n

Holes

For a successful s we must have $2B \leq m \cdot s - r \cdot n < 3B$ for some natural number r.

Holes

For a successful s we must have $2B \leq m \cdot s - r \cdot n < 3B$ for some natural number r.

Given that we have trimmed the first interval M_0 to the range $[a, b]$, this gives us a series of bounds

$$\frac{2B + r \cdot n}{b} \leq s < \frac{3B + r \cdot n}{a}$$

Holes

For a successful s we must have $2B \leq m \cdot s - r \cdot n < 3B$ for some natural number r.

Given that we have trimmed the first interval M_0 to the range $[a, b]$, this gives us a series of bounds

$$\frac{2B + r \cdot n}{b} \leq s < \frac{3B + r \cdot n}{a}$$

If

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

we have a 'hole' of values where a suitable s cannot possibly be.

Can skip these holes in search.

Performance of Modified Algorithm

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

Performance of Modified Algorithm

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

Oracle	Original algorithm		Optimised algorithm	
	Mean	Median	Mean	Median
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

Results on Hardware

Device	PKCS#1 v1.5 Attack		CBC-PAD Attack	
	Token	Session	Token	Session
Aladdin eTokenPro	✓	✓	✓	✓
Feitian ePass 2000	×	×	N/A	N/A
Feitian ePass 3003	×	×	N/A	N/A
Gemalto Cyberflex	✓	N/A	N/A	N/A
RSA Securid 800	✓	N/A	N/A	N/A
Safenet Ikey 2032	✓	✓	N/A	N/A
SATA DKey	×	×	×	×
Siemens CardOS	✓	✓	N/A	N/A

Timings

Device	Token		Session	
	Oracle	Time	Oracle	Time
Aladdin eTokenPro	FTT	21m	FTT	17m
Gemalto Cyberflex	FFT	92m	N/A	N/A
RSA Securid 800	TTT	13m	N/A	N/A
Safenet Ikey 2032	FTT	88m	FTT	17m
Siemens CardOS	TTT	21m	FFT	89s


EESTI VABARIIK
ISIKUTUNNISTUS
REPUBLIC OF ESTONIA IDENTITY CARD
MÄNNIK
MARI-LIIS
N/F
EST
01.01.1971
47101010033
AA1234567
AA1234567
09.01.2016
SPECIMEN


EESTI / EST
09.01.2011
IDESTAA12345678<<<<<<<<<<<<<<<
7101010F1601091EST<<<<<<<<<<<8
MAENNIK<MARI<LIIS<<<<<<<<<<<<<

Estonian ID Card

Contains 2 RSA keypairs

One can be used for signature only

One for signature and encryption/decryption

Uses PKCS#1v1.5 padding, FFT oracle

Digidoc software puts padding errors into world-readable logfile

Countermeasures

OAEP has been in PKCS#1 since v2.0 1998 - recommended for all new applications since v2.1 (2002)

Countermeasures

OAEP has been in PKCS#1 since v2.0 1998 - recommended for all new applications since v2.1 (2002)

Only device in our list supporting OAEP is the RSA SecureID - which allows PKCS#1v1.5 on the same key.

Countermeasures

OAEP has been in PKCS#1 since v2.0 1998 - recommended for all new applications since v2.1 (2002)

Only device in our list supporting OAEP is the RSA SecureID - which allows PKCS#1v1.5 on the same key.

Note `UnwrapKey` with symmetric key (CBC-PAD) is also a problem in PKCS#11 - GCM/CCM appear only in v2.30 (still in draft)

Countermeasures

OAEP has been in PKCS#1 since v2.0 1998 - recommended for all new applications since v2.1 (2002)

Only device in our list supporting OAEP is the RSA SecureID - which allows PKCS#1v1.5 on the same key.

Note `UnwrapKey` with symmetric key (CBC-PAD) is also a problem in PKCS#11 - GCM/CCM appear only in v2.30 (still in draft)

PKCS#1v1.5 still being used in current standards for XML encryption, TLS,... - our results can also be used there

Countermeasures

OAEP has been in PKCS#1 since v2.0 1998 - recommended for all new applications since v2.1 (2002)

Only device in our list supporting OAEP is the RSA SecureID - which allows PKCS#1v1.5 on the same key.

Note `UnwrapKey` with symmetric key (CBC-PAD) is also a problem in PKCS#11 - GCM/CCM appear only in v2.30 (still in draft)

PKCS#1v1.5 still being used in current standards for XML encryption, TLS,... - our results can also be used there

Manufacturer reaction has been varied - some very positive, some less so..

Pro Tips

If you would like to try improving the attack algorithm:

- (obvious?) you don't need to implement encryption/decryption!
- Pay close attention to floor/ceiling bounds in original algorithm

Thanks

Attacks included in our tool
for security analysis of device interfaces

(ask me or see `tookan.gforge.inria.fr` for a demo video)