



NTNU

Norwegian University of
Science and Technology

PADDING ORACLES: RSA ENCRYPTION

TTM4205 – Lecture 8

Tjerand Silde

08.09.2026

Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary



Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary

Reference Material

These slides are based on:

- ▶ The referred papers in the slides
- ▶ JPA: parts of chapter 10
- ▶ DW: parts of chapter 6

Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary



Padding Oracles

By this we mean, on a high level, an API that allows an adversary to learn if some input is correctly formed.

We limit ourselves to inputs with a particular padding.

Padding Oracles

We will look at symmetric and asymmetric padding schemes:

- ▶ in depth on the CBC block cipher mode (last time)
- ▶ extension attack against hashing (last time)
- ▶ padding attacks against the RSA scheme (today)

Several of which are relevant to the weekly problems.

We will also look at some mitigations to these issues.

Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary

Textbook RSA

The plain RSA encryption scheme works as follows:

KGen:

- ▶ Samples primes p and q of appropriate size and entropy
- ▶ Use fixed e and compute $d \equiv e^{-1} \bmod \text{lcm}(p-1, q-1)$
- ▶ Output the key pair $\text{pk} = (e, n = p \cdot q)$ and $\text{sk} = (d, p, q)$

Textbook RSA

The plain RSA encryption scheme works as follows:

Enc:

- ▶ Takes as input a message m and public key $pk = (e, n)$
- ▶ Computes the ciphertext $c \equiv m^e \bmod n$ and outputs c

Textbook RSA

The plain RSA encryption scheme works as follows:

Dec:

- ▶ Takes as input a ciphertext c and secret key $sk = (d, p, q)$
- ▶ Computes the message $m \equiv c^d \bmod n$ and outputs m

Textbook RSA

Question: Why is not the textbook RSA scheme secure?

Textbook RSA

The following things make the RSA scheme insecure:

- ▶ It is not randomized, so not even IND-CPA secure
- ▶ Given a ciphertext you can search for the message
- ▶ It is homomorphic (you can multiply ciphertexts)

Solution: structured, but randomized padding



Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary

RSA-PKCS#1v1.5

Let n be of k bytes. Given a message m of $\ell \leq k - 11$ bytes, the padded messages $\bar{m}$ of length k bytes is constructed as follows:

00 02 {at least 8 non-zero random bytes} 00 $\{m\}$

Quite simple, not proven secure, not secure in practice...

A bad couple of years for the cryptographic token industry



SafeNet eToken PRO Anywhere

Figure: <https://blog.cryptographyengineering.com/2012/06/21/bad-couple-of-years-for-cryptographic>

Optimal Asymmetric Encryption Padding

OAEP: more complex, proven secure, and what you should use.

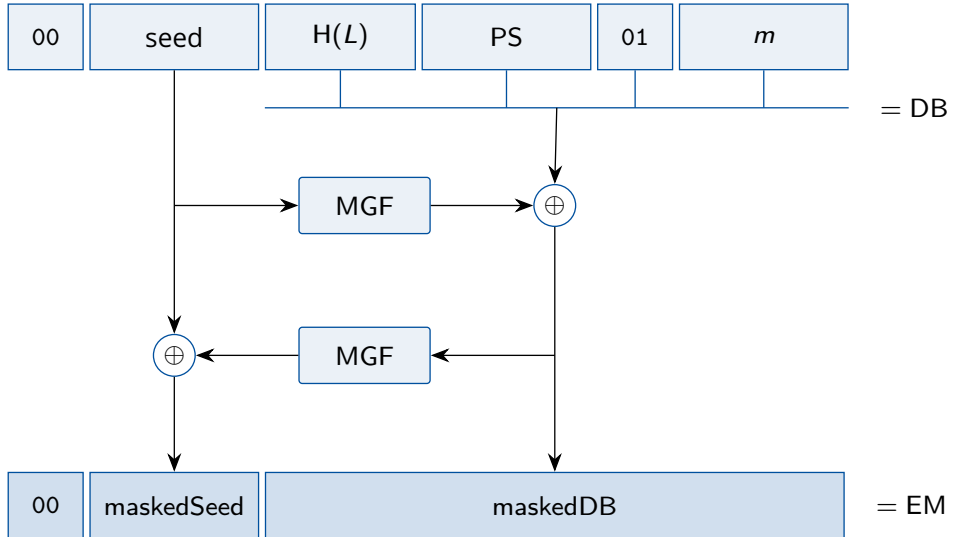
- ▶ Let n be of k bytes and the message m be of ℓ bytes.
- ▶ Let H be a hash function with output h bytes, and let L be a label of arbitrary length, which can be set to the empty string.
- ▶ Let seed be an ephemeral random string of h bytes.
- ▶ Let MGF, the *mask generation function*, be a pseudorandom generator: it takes a seed and returns as many bytes as we ask for.
- ▶ Let PS, the *padding string*, be $k - \ell - 2h - 2$ zero bytes.

OAEP is a two-round Feistel network run on the padded message:

- ▶ build the *data block* $DB = H(L) \parallel PS \parallel 01 \parallel m$, of $h + (k - \ell - 2h - 2) + 1 + \ell = k - h - 1$ bytes,
- ▶ mask it, $maskedDB = DB \oplus MGF(seed, k - h - 1)$,
- ▶ mask the seed back, $maskedSeed = seed \oplus MGF(maskedDB, h)$,
- ▶ encrypt $00 \parallel maskedSeed \parallel maskedDB$, which is k bytes, to get ctx .

Every bit of the encoding depends on the whole message and on the seed, so no part of it can be changed in a controlled way.

OAEP Encoding



OAEP Decoding

Decryption computes $EM = ctx^d \bmod n$ and then unwinds the two rounds:

- ▶ split $EM = 00 \parallel \text{maskedSeed} \parallel \text{maskedDB}$,
- ▶ recover $\text{seed} = \text{maskedSeed} \oplus \text{MGF}(\text{maskedDB}, h)$,
- ▶ recover $\text{DB} = \text{maskedDB} \oplus \text{MGF}(\text{seed}, k - h - 1)$,
- ▶ check that DB starts with $H(L)$, skip the zero bytes to the 01 separator, and return the rest as m .

Optimal Asymmetric Encryption — How to Encrypt with RSA

MIHIR BELLARE*

PHILLIP ROGAWAY†

November 19, 1995

Figure: <https://cseweb.ucsd.edu/~mihir/papers/oaep.pdf>

Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary



The Bleichenbacher Attack

However, many implementations (still) use RSA-PKCS#1v1.5 or similar padding schemes (note that this is version 1.5).

Recall: $00\ 02\ \{\text{at least 8 non-zero random bytes}\}\ 00\ \{m\}$.

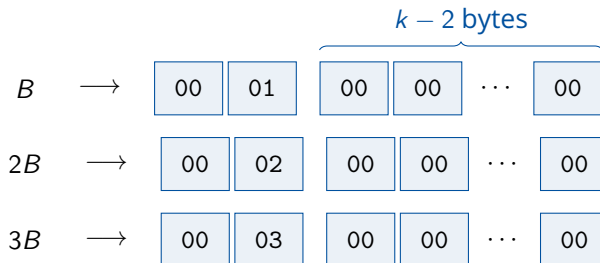
Bleichenbacher showed at CRYPTO 1998 that the first two bytes are enough: the original attack needed about a million queries.

Question: Assuming no integrity check of RSA ciphertexts, how could you attack this scheme?

The Bleichenbacher Attack

- ▶ Recall that RSA is homomorphic: $\bar{m}^e \cdot s^e \equiv (\bar{m} \cdot s)^e \pmod{n}$.
- ▶ We can multiply a ciphertext c by a chosen $s^e \pmod{n}$.
- ▶ We learn if padding is valid (error message or timings).
- ▶ We know the value s since we could choose it ourselves.
- ▶ Then we learn that unknown $\bar{m}$ times s is valid encoding.
- ▶ Let $B = 2^{8(k-2)}$. If valid then $2 \cdot B \leq \bar{m} \cdot s \pmod{n} < 3 \cdot B$.
- ▶ Repeat for fresh values s until we have a unique $\bar{m}$ left.

What the Bound B Means



With $B = 2^{8(k-2)}$, the interval $[2B, 3B)$ is exactly the set of integers whose top two bytes are 00 02. A conforming ciphertext therefore tells us that $\bar{m} \cdot s \bmod n$ lands in that interval, which is a range of about $1/256$ of all values.

Narrowing the Interval

A conforming answer for the multiplier s_i means that, for some integer k ,

$$2B \leq s_i \cdot \bar{m} - k \cdot n \leq 3B - 1.$$

Solving for the unknown $\bar{m}$ turns each answer into a range:

$$\frac{2B + k \cdot n}{s_i} \leq \bar{m} \leq \frac{3B - 1 + k \cdot n}{s_i}.$$

Every new s_i gives another such constraint, and the intersection of the ranges shrinks. After enough queries a single value of $\bar{m}$ remains; the plaintext.

Combining the Constraints

We keep a set M of intervals known to contain $\bar{m}$, starting from the one the original ciphertext gives us, $M = \{[2B, 3B - 1]\}$.

A conforming s_i does not tell us which multiple of n was subtracted, so we take every r that is possible for an interval $[a, b] \in M$,

$$\frac{a \cdot s_i - 3B + 1}{n} \leq r \leq \frac{b \cdot s_i - 2B}{n},$$

and for each such r we intersect the old interval with the new one:

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

Combining the Constraints

The first queries leave several candidate intervals, since several r fit. As s_i grows, all but one become empty and are discarded.

Once a single interval $[a, b]$ is left, we deliberately choose the next s_i so that a conforming answer splits it: each further query roughly halves $b - a$.

After about $\log_2 n$ successful queries the interval has collapsed to a single value, and $\bar{m} = a$. Removing the padding leaves the message. Bleichenbacher needed around a million oracle calls in total, most of them spent *finding* conforming s_i rather than narrowing the interval.

The Million Message Attack

Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1

Daniel Bleichenbacher

Bell Laboratories
700 Mountain Ave.
Murray Hill, NJ 07974
E-mail: bleichen@research.bell-labs.com

Figure: <https://spar.isi.jhu.edu/~mgreen/bleichenbacher.pdf>



Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary

Improving the Attack

1. *Trimmers*: multiply by a fraction t/u that keeps the padding valid, which shrinks the interval before the search even starts
2. Skip values of s that cannot produce a conforming message
3. Parallelization, threading, and pre-computation
4. Adapt the search to how strict the padding check is

Padding Checks

The efficiency depends on how strict the padding check is:

1. FFF: padding is 'ok' only if correctly padded and the plaintext is of a specific length (e.g., it's a 128-bit AES key and nothing else).
2. FFT: padding is 'ok' only if correctly padded, but plaintext is any length.
3. FTT: same as above, but also allows 0s in the "non-zero random bytes".
4. TFT: same as above, but 'ok' even if there are no zeros after the first byte.
5. TTT: padding is 'ok' as long as it starts with 00 02.

Queries Needed per Oracle Type

Oracle	Original algorithm		Modified algorithm			
	Mean	Median	Mean	Median	Trimmers	Mean skipped
FFF	-	-	18 040 221	12 525 835	50 000	7 321
FFT	215 982	163 183	49 001	14 501	1 500	65 944
FTT	159 334	111 984	39 649	11 276	2 000	61 552
TFT	39 536	24 926	10 295	4 014	600	20 192
TTT	38 625	22 641	9 374	3 768	500	18 467

Table 1: Performance of the original and modified algorithms.

Figure: <https://eprint.iacr.org/2012/417.pdf>

Efficient Padding Oracle Attacks on Cryptographic Hardware*

Romain Bardou¹, Riccardo Focardi^{2**}, Yusuke Kawamoto^{3***},
Lorenzo Simionato^{2†}, Graham Steel^{4***}, and Joe-Kai Tsay^{5***}

¹ INRIA SecSI, LSV, CNRS & ENS-Cachan, France

² DAIS, Università Ca' Foscari, Venezia, Italy

³ School of Computer Science, University of Birmingham, UK

⁴ INRIA Project ProSecCo, Paris, France

⁵ Department of Telematics, NTNU, Norway

Figure: <https://eprint.iacr.org/2012/417.pdf>

Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary

Twenty Years Later: ROBOT

In 2017 the attack was still there. The ROBOT paper found that many TLS stacks leaked a usable oracle again, through slightly different error codes, timeouts, or TCP behaviour, in products from several major vendors.

The countermeasure in the TLS standard is not to reject at all: on a padding failure, continue the handshake with a *random* premaster secret, so that the connection fails later and for a reason the adversary cannot distinguish.

Getting that right in constant time is hard, which is why TLS 1.3 removed RSA key transport altogether.



Protection

- ▶ Use OAEP rather than PKCS #1 v1.5 for encryption
- ▶ Never let the failure be observable: same error, same timing, or a random premaster secret
- ▶ Encrypt-then-authenticate, so malformed ciphertexts never reach the decryption
- ▶ Note that OAEP is not automatically safe either: Manger's attack in 2001 broke implementations that distinguish an integer that is too large from a decoding error
- ▶ Best of all: do not use RSA for encryption, and use an ephemeral key exchange with an AEAD

Contents

Introduction

Padding Oracles

Recall: RSA Encryption

RSA Padding Schemes

The Bleichenbacher Attack

Improved Bleichenbacher Attack

RSA Padding Oracle Mitigations

Summary

What to Remember

- ▶ Textbook RSA is deterministic and homomorphic, so it needs padding, and the padding needs care
- ▶ One bit of feedback is enough: the adversary walks an interval down to a single message
- ▶ PKCS #1 v1.5 has been broken since 1998 and keeps coming back through error handling
- ▶ OAEP has a proof, but the proof is about the scheme, not about your implementation
- ▶ As last lecture: authenticate first, and never let a failure be distinguishable

Questions?