Introduction to Cryptology

5.2 - A Padding Oracle Attack

Federico Pintore
Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

A padding Oracle Attack

In the CBC mode, the number of bits of a message should be a multiple of the block length. Otherwise, the message is padded.

A padding Oracle Attack

In the CBC mode, the number of bits of a message should be a multiple of the block length. Otherwise, the message is padded.

The PKCS#5 padding is a famous and standardised method.

A padding Oracle Attack

In the CBC mode, the number of bits of a message should be a multiple of the block length. Otherwise, the message is padded.

The PKCS#5 padding is a famous and standardised method.

- If $|m| = L$, t is the block length (both in bytes) and $L = rt + d$, then $b = t - d$ bytes need to be padded.

A padding Oracle Attack

In the CBC mode, the number of bits of a message should be a multiple of the block length. Otherwise, the message is padded.

The PKCS#5 padding is a famous and standardised method.

- If $|m| = L$, t is the block length (both in bytes) and $L = rt + d$, then $b = t - d$ bytes need to be padded.
- Therefore $1 \leq b \leq t$.

A padding Oracle Attack

In the CBC mode, the number of bits of a message should be a multiple of the block length. Otherwise, the message is padded.

The PKCS#5 padding is a famous and standardised method.

- If $|m| = L$, t is the block length (both in bytes) and $L = rt + d$, then $b = t - d$ bytes need to be padded.
- Therefore $1 \leq b \leq t$.
- Padding of (multiple copies of) the integer b, represented as a 8-bit string.

A padding Oracle Attack

In the CBC mode, the number of bits of a message should be a multiple of the block length. Otherwise, the message is padded.

The PKCS#5 padding is a famous and standardised method.

- If $|m| = L$, t is the block length (both in bytes) and $L = rt + d$, then $b = t - d$ bytes need to be padded.
- Therefore $1 \leq b \leq t$.
- Padding of (multiple copies of) the integer b, represented as a 8-bit string.
- Examples: if $b = 1$, 00000001 is appended to the end of the message; if $b = 2$, 00000010||00000010 is appended.

A padding Oracle Attack

In the CBC mode, the number of bits of a message should be a multiple of the block length. Otherwise, the message is padded.

The PKCS#5 padding is a famous and standardised method.

- If $|m| = L$, t is the block length (both in bytes) and $L = rt + d$, then $b = t - d$ bytes need to be padded.
- Therefore $1 \leq b \leq t$.
- Padding of (multiple copies of) the integer b, represented as a 8-bit string.
- Examples: if $b = 1$, 00000001 is appended to the end of the message; if $b = 2$, 00000010||00000010 is appended.
- The padded message is called encoded data.

A Padding Oracle Attack

When decrypting, the correctness of the padding of the decrypted message is verified:

- the value b of the last byte is read, and it is checked if it is the value of the last b bytes;
- if the padding is correct, the last b bytes are dropped to get the original plaintext.
- Otherwise, “padding error” is output.

A Padding Oracle Attack

When decrypting, the correctness of the padding of the decrypted message is verified:

- the value b of the last byte is read, and it is checked if it is the value of the last b bytes;
- if the padding is correct, the last b bytes are dropped to get the original plaintext.
- Otherwise, “padding error” is output.

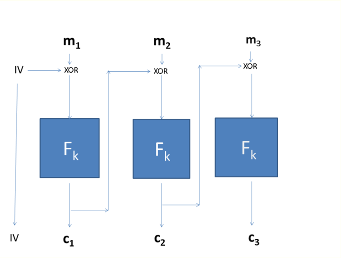
A Padding Oracle Attack

Some deployed protocols return a notification when a ciphertext does not decrypt correctly.

It can be seen as a limited decryption oracle for adversaries, and exploited to recover the entire plaintext from a ciphertext.

A Padding Oracle Attack - Example

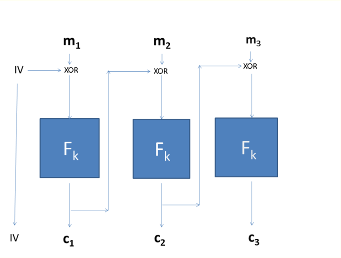
Consider a 3-block ciphertext (IV, c_1, c_2) which corresponds to the message (m_1, m_2) (unknown to the attacker).



- $m_2 = F_k^{-1}(c_2) \oplus c_1$ and it should end with $\underbrace{0\mathrm{x}b \cdots 0\mathrm{x}b}_{b \text{ times}}$.

A Padding Oracle Attack - Example

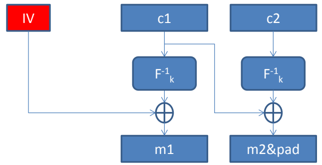
Consider a 3-block ciphertext (IV, c_1, c_2) which corresponds to the message (m_1, m_2) (unknown to the attacker).



- $m_2 = F_k^{-1}(c_2) \oplus c_1$ and it should end with $\underbrace{0\mathrm{x}b \cdots 0\mathrm{x}b}_{b \text{ times}}$.
- Key idea: given $c_1' = c_1 \oplus \Delta$, the decryption of (IV, c_1', c_2) returns (m_1', m_2') where $m_2' = m_2 \oplus \Delta$.

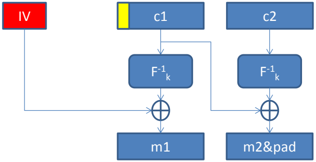
A Padding Oracle Attack

Step 1: learn b (number of padded bytes).



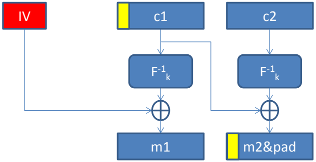
A Padding Oracle Attack

Step 1: learn b (number of padded bytes).



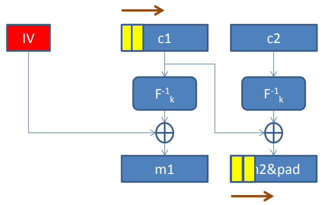
A Padding Oracle Attack

Step 1: learn b (number of padded bytes).



A Padding Oracle Attack

Step 1: learn b (number of padded bytes).



A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation

$$\Delta_n = 0x0 \cdots 0x0{\color{red}0xn}\underbrace{0\mathrm{x}b+1+b \cdots 0\mathrm{x}b+1+b}_{b \text{ times}}.$$

A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation

$$\Delta_n = 0x0 \cdots 0x0 0xn \underbrace{0xb+1+b \cdots 0xb+1+b}_{b \text{ times}}.$$

Let B be the value of the last byte of the unpadded message:

- if a decryption failure is notified, then $0xn \oplus B \neq b+1$;
- when the decryption is valid, it can be deduced that

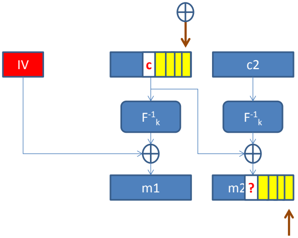
$$0xn \oplus B = b+1 \Leftrightarrow B = 0xn \oplus b+1.$$

A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation (below, $b = 4$)

$$\Delta_n = 0x0 \cdots 0x0 0xn \underbrace{0\text{x}b+1+b \cdots 0\text{x}b+1+b}_{b \text{ times}}$$

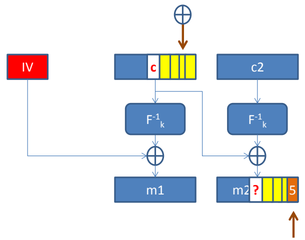


A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation (below, $b = 4$)

$$\Delta_n = 0x0 \cdots 0x0 0xn \underbrace{0\text{x}b+1+b \cdots 0\text{x}b+1+b}_{b \text{ times}}$$

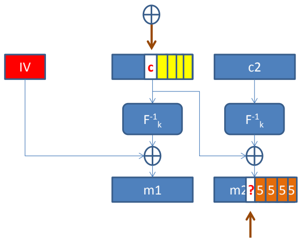


A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation (below, $b = 4$)

$$\Delta_n = 0x0 \cdots 0x0 0xn \underbrace{0\text{x}b + 1 + b \cdots 0\text{x}b + 1 + b}_{b \text{ times}}$$

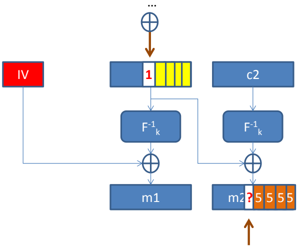


A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation (below, $b = 4$)

$$\Delta_n = 0x0 \cdots 0x00{\color{red}xn}\underbrace{0\text{x}b+1+b \cdots 0\text{x}b+1+b}_{b \text{ times}}$$

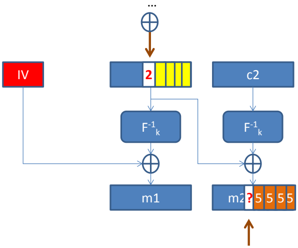


A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation (below, $b = 4$)

$$\Delta_n = 0x0 \cdots 0x0 0xn \underbrace{0\text{x}b + 1 + b \cdots 0\text{x}b + 1 + b}_{b \text{ times}}$$

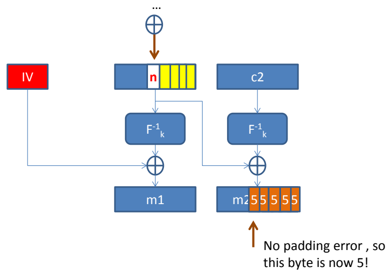


A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation (below, $b = 4$)

$$\Delta_n = 0x0 \cdots 0x00xn \underbrace{0\text{x}b+1+b \cdots 0\text{x}b+1+b}_{b \text{ times}}$$

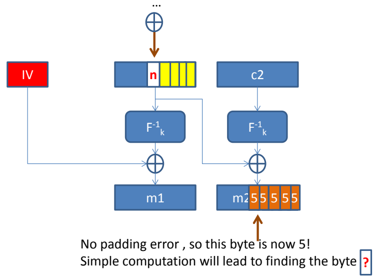


A Padding Oracle Attack

Step 2: recover the plaintext byte by byte.

The adversary modifies c_1 with the perturbation (below, $b = 4$)

$$\Delta_n = 0x0 \cdots 0x0 0xn \underbrace{0\text{x}b + 1 + b \cdots 0\text{x}b + 1 + b}_{b \text{ times}}$$



Further Reading I

Don Coppersmith.
The data encryption standard (DES) and its strength against attacks.
IBM journal of research and development, 38(3):243–250, 1994.

Itai Dinur, Orr Dunkelman, Masha Gutman, and Adi Shamir.
Improved top-down techniques in differential cryptanalysis.
Cryptology ePrint Archive, Report 2015/268, 2015.
http://eprint.iacr.org/.

Further Reading II

Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir.
Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems.
Cryptology ePrint Archive, Report 2012/217, 2012. http://eprint.iacr.org/.

Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir.
New attacks on feistel structures with improved memory complexities.
In Rosario Gennaro and Matthew Robshaw, editors, Advances in Cryptology – CRYPTO 2015, volume 9215 of Lecture Notes in Computer Science, pages 433–454. Springer Berlin Heidelberg, 2015.

Further Reading III

Lov K Grover.
A fast quantum mechanical algorithm for database search.
In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219. ACM, 1996.

Howard M Heys.
A tutorial on linear and differential cryptanalysis.
Cryptologia, 26(3):189–221, 2002.