Animesh Parab T2-T21 88

Written Assignment 1

1. Explain the padding scheme used in RSA. Why it is used? What is its limitation? (LO2)

Ans :-

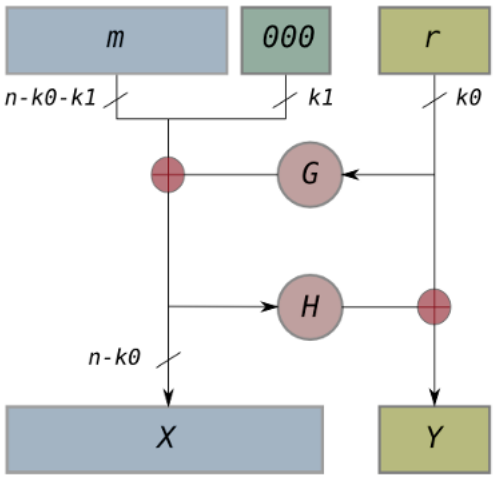
Padding in RSA (Rivest-Shamir-Adleman) is a crucial aspect of the encryption and decryption process. RSA padding schemes are used to add additional bits to the plaintext before encryption and remove them after decryption. Padding is primarily used to address certain security vulnerabilities and limitations of the RSA algorithm.

In cryptography, padding is a number of operations including appending data to anywhere of the plaintext before encryption. The purpose of a padding scheme is to avoid adversary to retrieve information of the primitive, for example, a chosen plaintext attack or an adaptive chosen ciphertext attack in RSA. Optimal Asymmetric Encryption Padding Optimal Asymmetric Encryption Padding(OAEP) was invented by Mihir Bellare and Phillip Rogaway in 1994 and enchanted by Don Johnson and Stephen Matyas in 1996. It was standardized as RSAES-OAEP in PKCS#1 Version 2 and lately republished as RFC 2437. OAEP combined with RSA is good at performance and provides good security especially against adaptive chosen ciphertext attack.

There are two aims of OAEP:

A. Adding random padding to plaintext can convert RSA from a deterministic scheme into a probabilistic one.

B. Prevent leaking any encryption structure information caused by chosen plaintext attack. The padding process of OAEP is shown as below:



Where

• n: the length of bits of RSA modulus

• k0 and k1: numbers defined by OAEP protocol

• m: the plaintext with a length of n−k0−k1 bits

• G and H are two cryptographic hash functions

• ⊕: xor operation

• r: a random generated string of k0 bits

Encoding of OAEP:

a) The plaintext m is padded with k1 zeros appending m to m’ with n−k0 bits length.

b) r is converted into a string of n−k0 bits by a cryptographic hash function G.

c) X=m’⊕G(r).

d) X is reduced to k0 bits by H.

e) Y=r⊕H(X).

f) The result of padding is X and Y.

Decoding of OAEP:

a) r is recovered by r=Y⊕H(X).

b) m’ is recovered by m’=X⊕G(r).

Security of OAEP

The OAEP provides semantic security against chosen ciphertext attack, though Victor Shoup raised doubt about whether OAEP could provide such security. In 2001, Eiichiro Fujisaki’s team proved that RSA-OAEP is semantical secure in the random oracle model.

Both block ciphers and RSA are permutations on a block(RSA's block isn't an integral number of bytes), so it's clear that both of them need some kind of padding if the data size doesn't correspond to the block size.

With block ciphers the padding doesn't do much: It fills up the remainder of the block, and tells you how much padding there was.

With RSA the padding is essential for its core function. RSA has a lot of mathematical structure, which leads to weaknesses. Using correct padding prevents those weaknesses.

For example RSA Encryption padding is randomized, ensuring that the same message encrypted multiple times looks different each time. It also avoids other weaknesses, such as encrypting the same message using different RSA keys leaking the message, or an attacker creating messages derived from some other ciphertexts.

RSA padding should always be used, and it has a minimum size of dozens of bytes, as opposed to a single byte with most block cipher paddings.

Why Padding is used in RSA

Padding schemes in RSA (Rivest-Shamir-Adleman) encryption are used to address certain vulnerabilities and limitations associated with the basic RSA algorithm. The most commonly used padding schemes in RSA are PKCS#1 v1.5 padding and OAEP (Optimal Asymmetric Encryption Padding). These padding schemes serve several important purposes:

1. Security: RSA encryption without padding can be vulnerable to attacks like the padding oracle attack, which can reveal information about the plaintext. Padding schemes add randomness and structure to the plaintext before encryption, making it harder for attackers to exploit vulnerabilities.

2. Data Integrity: Padding schemes ensure that the encrypted message can be decrypted correctly. They help distinguish between valid and invalid ciphertexts, preventing errors or tampering during transmission.

3. Preventing Attacks: Padding schemes prevent certain mathematical attacks on the RSA algorithm. Without padding, an attacker could potentially recover the plaintext by analyzing the ciphertext and exploiting patterns in the encryption process.

4. **Randomness**: Padding schemes often include random bytes, adding a level of randomness
to the encryption process. This randomness helps to ensure that encrypting the same plaintext multiple times results in different ciphertexts, improving security.

Two commonly used padding schemes in RSA:

1. PKCS#1 v1.5 Padding:

- PKCS#1 v1.5 padding is an older padding scheme used with RSA encryption.
- It involves adding a specific sequence of bytes to the plaintext before encryption.
- This padding includes a block type byte, random padding bytes, and a message digest.

- PKCS#1 v1.5 padding is still widely supported but is considered less secure than OAEP.

2. OAEP (Optimal Asymmetric Encryption Padding):

- OAEP is a more modern and secure padding scheme.
- It uses a hash function and a random number generator to add padding to the plaintext.
- OAEP padding is designed to provide better security against various cryptographic attacks, including chosen ciphertext attacks.
- It ensures that each encrypted message is unique, reducing the risk of patterns that could be exploited by attackers.

Limitations of Padding in RSA:-

While padding schemes in RSA enhance security, they also come with limitations:

1. Padding Overhead: Padding increases the plaintext's length, resulting in a larger ciphertext. This overhead can be a concern when transmitting data efficiently or when storage space is limited.

2. Compatibility: Different padding schemes exist, and the choice of padding can impact interoperability between different implementations of RSA. It's essential to use a padding scheme that is compatible with the recipient's decryption algorithm.

3. Padding Oracle Attacks: Some padding schemes, like PKCS#1 v1.5, are vulnerable to padding oracle attacks if not implemented correctly. These attacks can leak information about the plaintext.

4. Vulnerabilities : Some padding schemes have been vulnerable to specific attacks. For example, the PKCS#1 v1.5 padding scheme had vulnerabilities in the past that led to practical attacks like the Bleichenbacher attack.

5. Complexity: Implementing padding schemes correctly can be complex and error-prone. Errors in the padding process can lead to security vulnerabilities.

Let's illustrate PKCS#1 v1.5 padding with a simple example using Python:

//python

```
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import padding
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.asymmetric import rsa
# Generate RSA keys (usually done once)
private_key = rsa.generate_private_key(
public_exponent=65537,
key_size=2048
)
public_key = private_key.public_key()
# Message to be encrypted
message = b"Hello, RSA Padding!"
# Encrypt with PKCS#1 v1.5 padding
ciphertext = public_key.encrypt(
message,
padding.PKCS1v15()
)
# Decrypt with PKCS#1 v1.5 padding
decrypted_message = private_key.decrypt(
ciphertext,
padding.PKCS1v15()
)
print(f"Ciphertext: {ciphertext.hex()}")
print(f"Decrypted Message: {decrypted_message.decode()}")
```

In this example, we generate an RSA key pair, encrypt a message using PKCS#1 v1.5 padding,
and then decrypt it. PKCS#1 v1.5 padding ensures that the plaintext length is the same as the RSA modulus size before encryption.