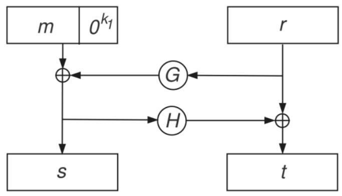
1. OAEP Encryption

We are given the following description of OAEP:



$$c = (s||t)^e \bmod N$$

A. Pseudo-code of OAEP encryption:

Note, when there is a ▷, this means that a comment follows, which gives clarification of something within the line of pseudocode.

Algorithm 1 OAEP encryption

Input: message m

Output: ciphertext c

Choose a random padding string 0^{k_1}

$m||0^{k_1}$ ▷ where $||$ denotes concatenation

Generate a random r

$G\_r \leftarrow G(r)$ ▷ hash r with hash function G

$s \leftarrow G\_r \oplus (m||0^{k_1})$ ▷ where $\oplus$ denotes XOR operation

$H\_s \leftarrow H(s)$ ▷ hash s with hash function H

$t \leftarrow H\_s \oplus (r)$

$c \leftarrow (s||t)^e \bmod N$ ▷ RSA encryption

return c

B. Pseudo-code of OAEP decryption:

Note, when there is a ▷, this means that a comment follows, which gives clarification of something within the line of pseudocode.

Algorithm 2 OAEP decryption

Input: ciphertext c

Output: message m

$(s||t) \leftarrow c^d \bmod N$ ▷ RSA decryption

$H\_s \leftarrow H(s)$ ▷ hash s with hash function H

$r \leftarrow H\_s \oplus (t)$ ▷ where $\oplus$ denotes XOR operation

$G\_r \leftarrow G(r)$ ▷ hash r with hash function G

$(m||u) \leftarrow G\_r \oplus s$

if $u=0^{k_1}$ **then** ▷ 0^{k_1} is a random padding string

    accept and return m

else

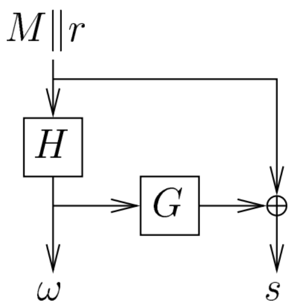
    reject

end

2. PSS Signature

We are given the following description of PSS:

- PSS-R: $\mu(M, r) = \omega \| s$, $\sigma = \mu(M, r)^d \bmod N$



A. Pseudo-code of PSS signature:

Note, when there is a ▷, this means that a comment follows, which gives clarification of something within the line of pseudocode.

Algorithm 3 PSS-R signature

Input: message M
Output: signature σ

Generate a random r
$M \| r$ ▷ where || denotes concatenation
$\omega \leftarrow H(M \| r)$ ▷ hash $M \| r$ with hash function H
$G\_\omega \leftarrow G(\omega)$ ▷ hash ω with hash function G
$s \leftarrow G\_\omega \oplus (M \| r)$ ▷ where ⊕ denotes XOR operation
$\mu(M,r) \leftarrow \omega \| s$ ▷ $\mu(M,r)$ the encoding of M with r
$\sigma \leftarrow \mu(M,r)^d \bmod N$ ▷ PSS-R, i.e., PSS with message recovery scheme
return σ

B. Pseudo-code of PSS verification:

Note, when there is a ▷, this means that a comment follows, which gives clarification of something within the line of pseudocode.

Algorithm 4 PSS-R verification

Input: signature σ
Output: True or False, i.e., whether the verification of signature σ was successful

$\mu_1(M_1,r_1) \leftarrow \sigma^e \bmod N$ ▷ $\mu_1(M_1,r_1)$ is the reconstructed message obtained by RSA verification
$\mu_1(M_1,r_1) \leftarrow \omega_1 \| s_1$ ▷ where || denotes concatenation
$G\_\omega_1 \leftarrow G(\omega_1)$ ▷ hash ω_1 with hash function G
$s_2 \leftarrow G\_\omega_1 \oplus (M \| r)$ ▷ where ⊕ denotes XOR operation
if $s_1 = s_2$ **then**
    return True ▷ $\mu_1(M_1,r_1) = \mu(M,r)$ so the verification of signature σ was successful
else
    return False
end

3. Is RSA Encryption Anonymous?

Bob must send 10 messages $m_1, \ldots, m_{10}$, either to Alice whose RSA public-key is (N_1, e_1), or to Anais whose RSA public-key is (N_2, e_2).

Therefore if Bob sends his 10 messages to Alice, he is going to send the ciphertexts:

$$c_i = (m_i)^{e_1} \mod N_1$$

for $1 \leq i \leq 10$.

Whereas if Bob sends his messages to Anais, he sends the following ciphertexts:

$$c_i = (m_i)^{e_2} \mod N_2$$

An eavesdropper gets the 10 ciphertexts c_i, and also knows the public-key of Alice and Anais, but she doesn't know the messages m_i. How might she be able to determine whether Bob sent his messages to Alice or Anais ?

The eavesdropper knows the public-key of Alice (N_1, e_1), and of Anais (N_2, e_2). She also obtains the 10 cipher texts c_i where $c_i = (m_i)^{e_1} mod N_1$ or $c_i = (m_i)^{e_2} mod N_2$.

We know that $N = p \times q$. If $gcd(N_1, N_2) > 1$, then modulus N_1 and modulus N_2 share a common factor (i.e., either p or q). If they share a common factor, then it's possible that Bob used the same p or q in both modulus N_1 and modulus N_2. This could mean that the messages were sent to the same person. Thus, the eavesdropper could determine whether Bob sent his messages to Alice or Anais. However, given $N = p \times q$, there isn't a known, computationally feasible algorithm that can recover the primes p and q. Furthermore, public modulus N must be very large, at least 1024 bits, to begin with. That is why, it is highly unlikely that the eavesdropper can determine whether Bob sent his messages to Alice or Anais.

If $gcd(N_1, N_2) < = 1$ and $e_1 = e_2$, then it's possible that Bob used the same encryption exponent e for both public-keys (N_1, e_1) and (N_2, e_2). This could mean that the messages were sent to both Alice and Anais. Nevertheless, it would still be difficult for the eavesdropper to decrypt the messages as she needs the private-keys d_1 and d_2 to do that as $m_i = (c_i)^{d_1} mod N_1$ and $m_i = (c_i)^{d_2} mod N_2$. Therefore, she would still need to know the modulus N_1 and modulus N_2, and as explained previously, this is computationally infeasible.

Due to all of this, the RSA encryption is anonymous.