Desarrollo de un programa criptográfico

CURSO CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA

María del Carmen Cámara
José María de Fuentes
Daniel Garzón
Lorena González
Ana Isabel González-Tablas
Pablo Martín
Antonio Nappa

uc3m | Universidad Carlos III de Madrid

COSEC

Cifrado asimétrico

Tenemos que responder a las siguientes preguntas:

- ¿Para qué utiliza el cifrado asimétrico?
- ¿Qué algoritmos ha utilizado y por qué?
- ¿Cómo gestiona las claves?

Ana I. González-Tablas Ferreres, José M. de Fuentes García-Romero de Tejada, Lorena González Manzano, Pablo Martín González

Cifrado asimétrico

Ejemplo en Python con PyCryptodome:

```
# Ciframos usando RSA-OAEP encryption scheme (RSA with PKCS#1 OAEP padding):
msg = b'A message for encryption'
encryptor = PKCS1_OAEP.new(pubKey)
encrypted = encryptor.encrypt(msg)
print("Encrypted:", binascii.hexlify(encrypted))


# Descriframos
decryptor = PKCS1_OAEP.new(keyPair)
decrypted = decryptor.decrypt(encrypted)
print('Decrypted:', decrypted)
```


Ana I. González-Tablas Ferreres, José M. de Fuentes García-Romero de Tejada, Lorena González Manzano, Pablo Martín González

Cifrado asimétrico

In the diagram,

- n is the number of bits in the RSA modulus.
- k_0 and k_1 are integers fixed by the protocol.
- m is the plaintext message, an $(n - k_0 - k_1)$-bit string
- G and H are mask generation functions based on chosen cryptographic hash functions
- $\oplus$ is an xor operation.

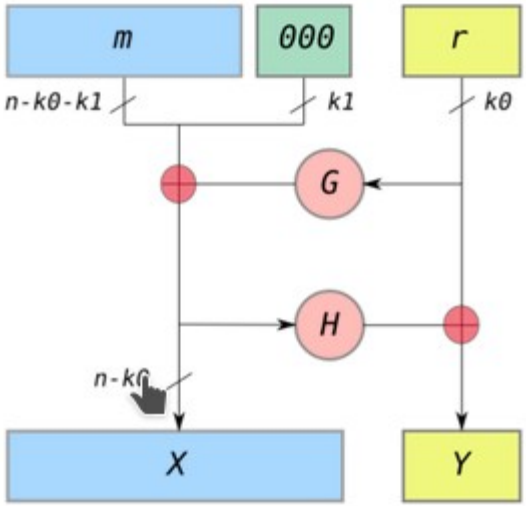
To encode,

1. messages are padded with k_1 zeros to be $n - k_0$ bits in length.
2. r is a randomly generated k_0-bit string
3. G expands the k_0 bits of r to $n - k_0$ bits.
4. $X = m00...0 \oplus G(r)$
5. H reduces the $n - k_0$ bits of X to k_0 bits.
6. $Y = r \oplus H(X)$
7. The output is $X \,||\, Y$ where X is shown in the diagram as the leftmost block and Y as the rightmost block.

Usage in RSA: The encoded message can then be encrypted with RSA. The deterministic property of RSA is now avoided by using the OAEP encoding.

To decode,

1. recover the random string as $r = Y \oplus H(X)$
2. recover the message as $m00...0 = X \oplus G(r)$



Ana I. González-Tablas Ferreres, José M. de Fuentes García-Romero de Tejada, Lorena González Manzano, Pablo Martín González

Firma

- ¿Para qué utiliza la firma digital?
- ¿Qué algoritmos ha utilizado y por qué?
- ¿Cómo gestiona las claves?
- ¿Cuál es la PKI que ha desarrollado?

Ana I. González-Tablas Ferreres, José M. de Fuentes García-Romero de Tejada, Lorena González Manzano, Pablo Martín González

Firma

Ejemplo en Python con PyCryptodome:

```
from Crypto.PublicKey import RSA


# Generate a 1024-bit RSA key-pair
keyPair = RSA.generate(bits=1024)
print(f"Public key:  (n={hex(keyPair.n)}, e={hex(keyPair.e)})")
print(f"Private key: (n={hex(keyPair.n)}, d={hex(keyPair.d)})")


msg = b'Mensaje a ser firmado'
from hashlib import sha512
hash =
signature = pow(hash, keyPair.d, keyPair.n) #firmo con la privada
print("Signature:", hex(signature))


msg = b'Mensaje a ser firmado'
hash =
hashFromSignature = pow(signature, keyPair.e, keyPair.n) #valido firma con la publica
print("Signature valid:", hash == hashFromSignature)
```


Ana I. González-Tablas Ferreres, José M. de Fuentes García-Romero de Tejada, Lorena González Manzano, Pablo Martín González

CURSO CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA

COSEC

uc3m | Universidad Carlos III de Madrid