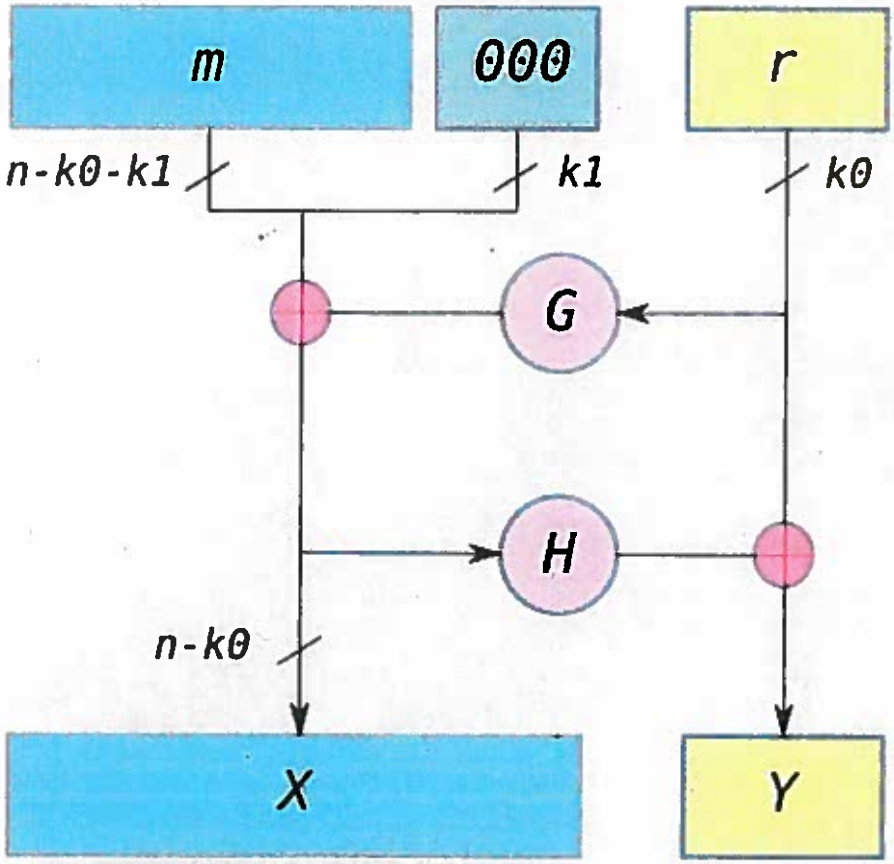
OAEP padding + RSA



optimal assymmetric encryption padding

Optimal asymmetric encryption padding

From Wikipedia, the free encyclopedia

> *"OAEP" redirects here. For the division of the Thailand Ministry of Science Technology and Environment previously known as the Office of Atomic Energy for Peace, see Office of Atoms for Peace.*

In cryptography, **Optimal Asymmetric Encryption Padding** (**OAEP**) is a padding scheme often used together with RSA encryption. OAEP was introduced by Bellare and Rogaway,[1] and subsequently standardized in PKCS#1 v2 and RFC 2437.

The OAEP algorithm is a form of Feistel network which uses a pair of random oracles G and H to process the plaintext prior to asymmetric encryption. When combined with any secure trapdoor one-way permutation f, this processing is proved in the random oracle model to result in a combined scheme which is semantically secure under chosen plaintext attack (IND-CPA). When implemented with certain trapdoor permutations (e.g., RSA), OAEP is also proved secure against chosen ciphertext attack. OAEP can be used to build an all-or-nothing transform.

OAEP satisfies the following two goals:

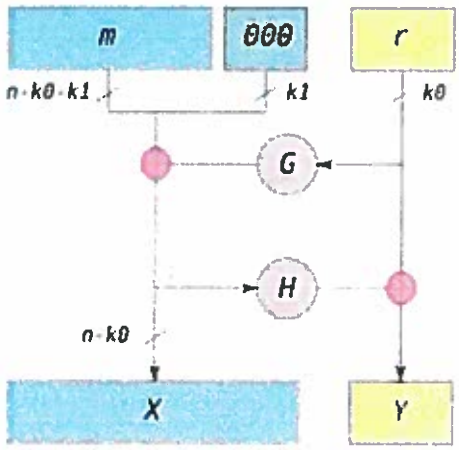
1. Add an element of randomness which can be used to convert a deterministic encryption scheme (e.g., traditional RSA) into a probabilistic scheme.
2. Prevent partial decryption of ciphertexts (or other information leakage) by ensuring that an adversary cannot recover any portion of the plaintext without being able to invert the trapdoor one-way permutation f.

The original version of OAEP (Bellare/Rogaway, 1994) showed a form of "plaintext awareness" (which they claimed implies security against chosen ciphertext attack) in the random oracle model when OAEP is used with any trapdoor permutation. Subsequent results contradicted this claim, showing that OAEP was only IND-CCA1 secure. However, the original scheme was proved in the random oracle model to be IND-CCA2 secure when OAEP is used with the RSA permutation using standard encryption exponents, as in the case of RSA-OAEP.[2] An improved scheme (called OAEP+) that works with any trapdoor one-way permutation was offered by Victor Shoup to solve this problem.[3] More recent work has shown that in the standard model (that is, when hash functions are not modeled as random oracles) it is impossible to prove the IND-CCA2 security of RSA-OAEP under the assumed hardness of the RSA problem.[4][5]

Diagram of OAEP

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 typically some cryptographic hash functions fixed by the protocol.



OAEP Diagram

To encode,

1. messages are padded with k_1 zeros to be $n - k_0$ bits in length.
2. r is a random 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.

To decode,

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

The "all-or-nothing" security is from the fact that to recover m, you must recover the entire X and the entire Y; X is required to recover r from Y, and r is required to recover m from X. Since any changed bit of a cryptographic hash completely changes the result, the entire X, and the entire Y must both be completely recovered.

See also

- Key encapsulation

References

1. M. Bellare, P. Rogaway. *Optimal Asymmetric Encryption -- How to encrypt with RSA*. Extended abstract in Advances in Cryptology - Eurocrypt '94 Proceedings, Lecture Notes in Computer Science Vol. 950, A. De Santis ed, Springer-Verlag, 1995. full version (pdf) (http://www-cse.ucsd.edu/users/mihir/papers/oae.pdf)
2. Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. *RSA--OAEP is secure under the RSA assumption*. In J. Kilian, ed., Advances in Cryptology -- CRYPTO 2001, vol. 2139 of Lecture Notes in Computer Science, SpringerVerlag,