Possible Attacks on RSA Signature

KEYWORDS: Cryptography, Public Key, RSA, Cryptanalysis

G. Jai Arul Jose
Research Scholar, Sathyabama University, Chennai

Dr. C. Suyambulingom
Professor (Rtd.), Department of Mathematics, Tamil Nadu Agricultural University, Coimbatore

ABSTRACT *Cryptographic techniques, such as encipherment, digital signatures, key management and secret sharing schemes, are important building blocks in the implementation of all security services. In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature is analogous to the handwritten signature. A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message. In this paper we present a brief account on RSA signature and some possible attacks on it.*

1. Integer Factorization

The problem of integer factorization is one of the oldest in number theory and the advents of computers have stimulated considerable progress in recent years. However, the security of many cryptographic techniques depends upon the intractability of the integer factorization problem. A partial list of such schemes includes the RSA public-key encryption scheme and the RSA signature scheme. This section focuses on the knowledge on algorithms for the integer factorization problem.

Definition: The integer factorization problem is the following: given a positive integer ν, find its prime factorization; i.e., write $n = p_1^{e1} p_2^{e2} \dots p_k^{ek}$ where the pi are pair wise distinct primes and each $e_i \geq 1$.

This problem is believed to be hard for general ν when ν is large. Some ingenious methods have been devised in an attempt to factorize large composite numbers ν. The three methods that are most effective on very large numbers are the quadratic sieve, the elliptic curve method and the number field sieve. Other well-known methods that were precursors include Pollard's rho-method and p - 1 method, Williams's p + 1 method, the continued fraction algorithm, and of course, trial division.

2. The RSA problem

The intractability of the RSA problem forms the basis for the security of the RSA public-key encryption scheme and the RSA signature scheme.

Definition: The RSA problem is the following: given a positive integer n that is a product of two distinct odd primes p and q, a positive integer e such that gcd(e, (p - 1)(q - 1)) = 1, and an integer c, find an integer m such that m^e °C (mod n).

Clearly the RSA problem is no more difficult then factorization, since if p and q can be found then it is simple to find m.

3. RSA Publickey Cryptosystem

The RSA public-key cryptosystem was introduced in 1978, and may be used for both secrecy and digital signatures. The cryptosystem works in Z_n, where n is the product of two large primes p and q, and its security is based on the difficulty of factoring n, that is, the integer factorization problem.

To use the RSA public-key cryptosystem, a user A first generates their public and secret keys by

(i) Generating two large distinct primes p and q.
(ii) (ii) Computing n = pq and $\phi(n) = (p - 1)(q - 1)$, where $\phi(n)$ is Euler Totient Function.
(iii) Choosing a random integer e such that $0 < e < \phi(n)$, and $\gcd(e, \phi(n)) = 1$,
(iv) Using the Euclidean Algorithm to compute the unique integer d, where $0 < d < \phi(n)$, such that e^d ° 1 (mod $\phi(n)$), and
(v) Publishing the pair (n, e) as the public key, and keeping d as the private key.

RSA is an example of block cipher, that is, a message is encrypted by being broken down into blocks (or strings) of a fixed length, and each block is encrypted individually. The plaintext and the ciphertext space are $P = C = Z_n$. To encrypt a message block m for user A, a user B

(i) Obtains A's authentic public key (n, e),
(ii) Represents the message m as an integer in the range 0, …, (n – 1),
(iii) Computes the ciphertext $E_K(m) = c = m^e \bmod n$, and
(iv) transmits the ciphertext c to user A.

To decrypt the ciphertext c, user A computes $D_k(c) = c^d = m \bmod n$.

RSA has the property that for any two distinct messages m_1 and m_2 with ciphertexts c_1 and c_2 respectively, the ciphertext of $m = m_1 . m_2 \bmod n$ is

$$c \equiv m^e \equiv (m_1 . m_2)^e \equiv m_1^e m_2^e \equiv c_1 . c_2 \pmod{n}.$$

This is often referred to as the homomorphic property of RSA.

4. RSA Signature

The RSA public-key cryptosystem can be used to provide digital signatures by reversing the roles of encryption and decryption as follows:

A user A also generates their public and private keys exactly as in the RSA publickey cryptosystem. The set of users of signatures also need to agree on a hash function h. Then to generate a signature of a message m, user A

(i) Computes M = h(m),
(ii) Computes $s = M^d \bmod n$, and
(iii) Outputs s as the signature of m.

To verify the signature, a user B
(i) Obtains A's authentic public key (n, e),
(ii) Verifies that $s \leq n$; if not, then reject the signature
(iii) Computes $M' = s^e \bmod n$,
(iv) Accepts the signature s if and only if $M' = M$.

A. The RSA signature with message recovery

A user A also generates their public and private keys exactly as in the RSA publickey cryptosystem. The set of users of signatures agree on a redundancy function R. Then to generate a signature of a message m, user A

(i) Computes M = R(m),
(ii) Computes s = M^d mod n, and
(iii) Outputs s as the signature of m.

To verify the signature, a user B
(i) Obtains A's authentic public key (n, e),
(ii) Computes $M' = s^e$ mod n,
(iii) Verifies that M' has the required redundancy, and
(iv) Recovers the message m = $R^{-1}(M')$.

Note that due to the homomorphic property of RSA, for any two distinct message m_1 and m_2 with corresponding signatures s1 and s2 respectively, the signature of m = $m_1 . m_2$ mod n is

$s = (m_1 . m_2)^d \equiv m_1^d . m_2^d \equiv s_1 . s_2$ (mod n).

In particular, for any message m_1 with signature s_1, the signature of m = $-m_1$ mod n is s = $-s_1$ mod n. It is important, therefore, that the redundancy function R is not multiplicative, that is, $R(m_1 . m_2) \neq R(m_1) . R(m_2)$.

5. Possible Attacks on RSA Signatures

The security of RSA signatures is based on the intractability of the integer factorization problem. RSA can be used as the basis of digital signatures with and without message recovery. Three possible attacks on the RSA signature scheme are as follows:

5.1. Factorization

If an adversary is able to factor the public modulus n of some entity A, then the adversary can compute ϕ(n) and then, using the extended Euclidean algorithm, deduce the private key d from ϕ(n) and public exponent e by solving $e^d \equiv 1$ (mod ϕ(n)). This constitutes a total break of the system. To guard against this, one must select p and q so that factoring n is a computationally infeasible task.

A lot of algorithm has been proposed regarding factorization, the Pollard rho algorithm [7], and the Pollard (π-1) algorithm [8], Brent's method [9], are probabilistic, and may not finish, even for small values of N, but Trial division algorithm and proposed method can finish all trivial and nontrivial values of N, shown in Table 1. This method is not probabilistic. To break RSA in to two prime numbers we should have the product of that prime numbers is equal to N. Factorization of N is very difficult to find that prime number. MFF can factors of N, which is P and Q, are its respective prime factors. Various steps involved in the method are as follows:

1. Let N = P*Q.
2. Compute X =ceil (sqrt (N)).
3. Compute Y =sqrt (X^2 – N).
4. If Y is integer
5. Compute P =X – Y and Q =X + Y.
   Stop.
6. Else X = X +1, X+ 2,…. , X + 2*X, .., X+N.
7. Continue step 3 to 6, till Y is integer.

Example 1:
Let N=95
Decimal number = 2
Number of bits = 7
Let factors = P, Q
Compute X_n =10
Compute Y = 2.236 (is not integer number)
Go to step six.
X_n =11
Y = 5.09 (is not integer number)
Go to step six
X = 12
Y = 7 (is an integer number)
P = 5
Q = 19

Example 2:
Let N=99400891
Decimal number = 8
Number of Bits = 28
Compute X_n = 9970
Compute Y= 3
P = 9967
Q = 9973

Example 3:
Let N= 2320869986411928544793
Decimal number = 22
Number of Bits = 73
Let factors = P, Q
Compute X = 48175408524
Compute Y = 219488.97 (is not integer number)
Go to step six
X = X + 1, …, X + 17073029192103
X = 17121204600627
Y = 17121136822844
P = 67777783
Q = 34242341423471

5.2. Existential forgery

The basic idea behind RSA signatures is to compute s = M^d (mod n) where M is (some function of) the message. This means that an adversary can choose an arbitrary s* and compute m* = $(s^*)^e$ (mod n) and claim s* is a valid signature on m*.

This is one reason why RSA signatures are always either of the form

(a) s = $(h(m))^d$ (mod n), where h is a one-way collision resistance hash function, giving a signature with appendix, or
(b) s = $(R(m))^d$ (mod n), where R is a redundancy-adding function, giving a signature with message recovery for a message m of limited length.

5.3. Multiplicative property of RSA

The RSA signature scheme (as well as the encryption scheme) has the following multiplicative property, sometimes referred to as the homomorphic property. If $s_1 = m_1^d$ mod n and $s_2 = m_2^d$ mod n are signatures on messages m_1 and m_2, respectively (or, more properly, on messages with redundancy added), then s = s_1s_2 mod n has the property that s = $(m_1m_2)^d$ mod n. If m = m_1m_2 has the proper redundancy, then s will be valid signature for it. Hence, it is important that the redundancy function R is not multiplicative, i.e., $R(m_1\ m_2) \neq R(m_1)\ R(m_2)$. Alternatively this homomorphism weakness of RSA can be eliminated by applying some one-way hash-function h to m before signing m, as long as h is not multiplicative.

Conclusion

The security of RSA signatures is based on the intractability of the integer factorization problem. RSA can be used as the basis of digital signatures with and without message recovery. We have described general types of attack against RSA signature. For RSA signatures the homomorphism property could only be used by a forger to forge a signature.

REFERENCE

[1] William Stallings, “Cryptography and Network Security: Principles and Practice”. PHI. | [2] Bruce Schneier, Applied Cryptography”, John Willy and Sons. | [3] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. “Handbook of Applied Cryptography”. CRC Press, 1996. | [4] T. Nagell. “Introduction to number theory”. Chelsea Publishing Company, 1981. | [5] D.R. Stinson. “Cryptography theory and practice”. CRC Press, 1995. | [6] Y. Zheng. “Digital signcryption or how to achieve cost (signature+encryption) cost (signature) + cost(encryption)”. In Advances in Cryptology – Proceedings of Crypto ’97, pages 165–179. Springer-Verlag, 1997. | [7] J. Pollard, "Monte Carlo methods for index computation (mod p)",Math. Comp., Vol. 32, pp.918-924, 1978. | [8] J. Pollard, "Theorems on factorization and primality testing", Proc. Cambridge Philos.Soc., Vol. 76, pp.521-528, 1974. | [9] R. P. Brent, “An improved Monte Carlo factorization algorithm”, BIT 20 (1980), 176-184. MR 82a:10007, Zbl 439.65001. rpb051. |