Attacks On the RSA Cryptosystem

Prepared for *SE 4C03* Class Project, Winter 2005 Instructor: Dr. Kartik Krishnan

Prepared by: Xiao-lei Cui, 0140976

Report Due Date: April 5th 2005

Last Revision Date: April 4th, 2005

1. Overview of RSA Encryption / Decryption

In public-key cryptography, each individual has a pair of keys, (e, d), where e is the public key known to the others, and d is the private key known merely to the owner. The public key is used to encrypt the message sent (or signing the message), and the private key is used to decrypt the ciphertext (or verifying the message). Likewise to secret key algorithms, public key algorithms take a plain message and perform a irreversible transformation on it. RSA, namely after its three inventors, Rivest, Shamir, and Adleman [6], is a public key cryptographic algorithms that may perform both encryption and decryption. RSA is frequently used in applications such as e-mail, e-banking, remote login, etc, where security of digital data is a primary concern. Over years, numerous attacks on RSA illustrating RSA's present and potential vulnerability have brought our attention to the security issues of RSA cryptosystem. We will investigate some essential attacks in later section.

Before looking at the attacks, we firstly describe a simplified version of RSA algorithm. Let N be the product of two large prime numbers, N = p*q, where p, q are of the same size in term of bits in binary representation, and N is called the RSA modulus. Let e, d be two integers, such that $e*d = 1 \mod \Phi(N)$. $\Phi(N) = (p-1)*(q-1)$ is the number of primes in the interval of [1..N-1]. Now, we obtained the public key, $\langle N, e \rangle$, which is used for encryption; and the private key, $\langle N, d \rangle$, which is known only to the recipient of the encrypted messages.

Here is how RSA encryption and decryption works. To encrypt a message M (<N), one computes: $C := M^{h} \in M$

To decrypt the ciphertext C, the receiver (owner of d) computes:

 $M := C^d = M^{(ed)} = \mod N$

Using the above equality, RSA function is defined as $x \rightarrow x^{h} \mod N$. If d is known, RSA function can be easily inverted. The term, *breaking RSA*, refers to inverting RSA function without any notion of d. Throughout this report, we use "Alice" to denote the message sender, "Bob" to denote the legitimate receiver, and "Marvin" for the attacker.

2. Two Categories of Attacks On RSA

There is a straight method, to enumerate all element in the multiplicative group of N until M is found, but such method results in an exponential running time, $O(n^e)$. Therefore, we are interested mostly in 'efficient' algorithms with a substantial lower running time. During the past years of attacking on RSA, such efficient algorithms can be classified into two categories: Mathematical Attacks and Implementation Attacks.

2.1 Mathematical Attacks on RSA

Mathematical attacks focus on attacking the underlying structure of RSA function. The first intuitive attack is the attempt to factor the modulus N. Because knowing the factorization of N, one may easily obtain $\Phi(N)$, from which d can be determined by $d = 1/e \mod \Phi(N)$. However, at present, the fastest factoring algorithm runs in exponential time. Our objective is to survey RSA attacks that decrypts message without directly factoring N.

a) Elementary attacks

Generally speaking, Elementary attacks revealed blatant misuse of RSA. One common example of such misuse would be choosing common modulus N to serve multiple users. Let's assume the same N is used by all users, and Alice is sending a message M to Bob, which has been encrypted by the RSA function, $C = M^{A}(e_{b}) \mod N$. It looks like Marvin can not decrypt C since he does not know db.

However, in fact, Marvin is able to use his own keys, em and dm, to factor N, and in turn recover Bob's private key, db. So the resulting system is no longer secure!

b) Small Private Key attacks

To improve the RSA decryption performance in the matter of running-time, Alice might tend to use a small value of da, rather than a large random number. A small private key indeed will improve performance dramatically, but unfortunately, a attack posed by M.Wiener [5] shows that a small d leads to a total collapse of RSA cryptosystem. This break of RSA is base on **Wiener's Theorem**, which in general provides a lower constraint for d. Wiener has proved that Marvin may efficiently find *d* when $d < \frac{1}{3} \cdot \frac{N}{14}$.

In addition to his success in RSA-attack, Wiener also discovered a number of techniques that enable fast decryption and are not susceptible to his attack. Two sample techniques are illustrated as the following.

Choosing large public key: Replacing *e* by e', where $e' = e + t * \Phi(N)$ for some large t. When e' is sufficient large, i.e. $e' > N^{0.5}$, then Weiner's attack can not be mounted regardless of how small *d* is.

Using Chinese Remainder Theorem: Suppose one chooses d such that both $d_p = d \mod (p-1)$ and $dq = d \mod (q-1)$ are small, then a fast decryption of C can be carried out as follows: first compute $Mp = C^{d}p \mod p$ and $Mq = C^{d}q \mod q$. Then use the **CRT** to compute the unique value $M \in \mathbb{Z}\mathbb{N}$ satisfying $M = Mp \mod p$ and $M = Mq \mod q$. The resulting M satisfies $M = C^{d} \mod N$ as required. The point is that the attack by **Wiener's Theorem** does not apply here because the value of d mod $\Phi(N)$ can be large.

c) Small Public Key Attacks

Similar to the private key preferences, to reduce encryption time, it is customary to use a small public key (e), but unlike the previous situation, attacks on small e turn out to be much less effective. The most powerful attacks on small e are based on **Coppersmith's Theorem** [3]. This theorem provides an algorithm for efficiently finding all roots of N that are less than $x = N^{(1/d)}$. For brevity reason, we will bypass the details of **Coppersmith's Theorem**, rather focus on its impact. One example of applications based on this theorem is known as "Hastad's Broadcast Attack"[4].

Hastad's Broadcast Attack

Suppose Bob wishes to send an encrypted message M to a number of parties P1; P2; ...; Pk. Each party has its own RSA key, < Ni, ei >. Hastad showed that a linear-padding to M prior to encryption is insecure, and further more, by eavesdropping Marvin learns Ci = fi (M)^Aei mod Ni for i = 1..k, if enough parties are involved, Marvin can recover the plaintext Mi from all the ciphertext [4]. His discovery stands on the mathematical analysis on solving system of equations: gi (M) = 0 mod Ni (1). He proved that a system of univariate equations modulo relatively prime composites, such as (1), could be efficiently solved if sufficiently many such equations are provided.

2.2 Implementation Attacks on RSA

Securely implementing RSA is not a trivial task. Attacks falling into this category take on the implementation pitfalls of RSA cryptosystems. A clever attack posed by Kocher, known as "Timing Attacks"[2], is a typical example of attacks on the RSA implementation.

Suppose a smartcard that stores a private RSA key is used, and Marvin may not be able to examine its contents and expose the key. However, by precisely measuring the *time* it takes the smartcard to perform an RSA decryption, Marvin can quickly discover the private decryption exponent *d*. This is

referred to as "Timing Attacks". Marvin can attack against a simple implementation of RSA using the "repeated squaring algorithm".

The algorithm works as follows: Let d = dndn-1, ..., d0Set z equal to M and C equal to 1. For (i = 0 to n) do these steps: 1. If di = 1, set C equal to $Cz \mod N$. 2. Set z equal to $z^2 \mod N$. At the end, C has the value $Md \mod N$.

To mount the attack, Marvin asks the smartcard to generate signatures on a large number of random messages $M1, ..., Mk \in$ multiplicative group of N, and measures the time *Ti* it takes the card to generate each of the signatures.

The attack recovers bits of d one at a time. Since we knew that d is prime, d must be odd number, thus the least significant bit do must be 1. The following description illustrates how Marvin can actually find out what d is bit-by-bit.

```
Marvin begins with the least significant bit, d0 = 1
For i = 2 to n
If the measure on {ti} and {Ti} are correlated
di = 1
else di = 0
```

Finally, Marvin recovers all di, where i = 1, ..., n.

3. Summary on the RSA Defends

- RSA modulus N, should not be used by more than one entity.
- One can improve decryption performance by using large e, or using CRT to decrypt C, both of which may stand against Weiner's attack.
- Linear padding to M prior to encryption can not form a defense against Hastad's Broadcast Attack. One must use a *randomized* pad [1] rather than a fixed one.
- Adding appropriate time-delay when generating each signatures so that modular exponentiation always takes a fixed amount of time, will prevent against Timing Attack.

4. Conclusion

Ever since RSA's initial publication (1977), the past twenty-seven years of research in breaking RSA has produced some insightful attacks, yet no devastating attacks have been found by now. From these attacks, we learned how to avoid the major pitfalls when implementing RSA, and derive methodologies to defend RSA cryptosystems against attacks. The ongoing research that might bring in new security issues and challenges to RSA become our essential tool to enhance the degree of security of RSA cryptosystems.

Reference:

- [1] M. BELLARE and P. ROGAWAY, Optimal asymmetric encryption, *EUROCRYPT '94*, Lecture Notes in Computer Science, vol. 950, Springer-Verlag, Berlin and New York, 1994, pp. 92-111.
- [2] P. KOCHER, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, *CRYPTO '96*, Lecture Notes in Computer Science, vol. 1109, Springer-Verlag, 1996, pp. 104–113.
- [3] D. Boneh, Twenty Years of Attacks on the RSA Cryptosystem, http://www.ams.org/notices/199902/boneh.pdf.
- [4] J. HASTAD, Solving simultaneous modular equations of low degree, SIAM J. Comput. 17 (1988), 336-341.
- [5] M. WIENER, Cryptanalysis of short RSA secret exponents, *IEEE Trans. Inform. Theory* 36 (1990), 553– 558.
- [6] C. KAUFMAN, R. PERLMAN, "Network Security –private communication in a public world", 2nd edition, Prince Hall PTR, 2002.