ISSN 2229-5518

Attack on RSA Cryptosystem

Sachin Upadhyay

Abstract— The RSA cryptosystem is most widely used cryptosystem it may be used to provide both secrecy and digital signatures and its security is based on the intractability of the integer factorization. The security of RSA algorithm depends on the ability of the hacker to factorize numbers. New, faster and better methods for factoring numbers are constantly being devised. The Trent best for long numbers is the Number Field Sieve. Although the past work has proven that none of the attacks on RSA cryptosystem were dangerous. Indeed most of the dangers were because of improper use of RSA. In this paper what I am trying to do is to analyze the different types of possible attacks on RSA Cryptosystem.

Index Terms— Cryptology, Cryptography, Cryptanalysis, CRT, Decryption, Encryption, RSA.

1 Introduction

The RSA Cryptosystem developed in 1977, by three peoples: Ronald Rivest, Adi Shamir & Len Adleman which is based upon the difficulty of factorization of two large primes. The cryptosystem is most commonly used for providing privacy and ensuring authencity of digital data. These days RSA is deployed in many commercial systems. It is used by web servers and browsers to secure web traffic, it is used to secure login sessions and it is at the heart of electronic credit card payment systems. So we can say that RSA is very frequently used in some or the other applications. The RSA Cryptosystem has been analysed for vulnerability by many researchers. Although the past work has proven that none of the attacks on RSA cryptosystem were dangerous. Indeed most of the dangers were because of improper use of RSA. Our goal is to survey some of these attacks and describe the underlying mathematical tools they use. Throughout the survey we follow standard naming conventions and use Alice and Bob to denote two generic parties wishing to communicate with each other. We use Marvin to denote a malicious attacker wishing to eavesdrop or tamper with the communication between Alice and Bob.

2 Elementary Attacks

We begin by describing some old elementary attacks. These attacks illustrate blatant misuse of RSA. Although many such attacks exist, we are considering few ones

2.1 Common Modulus

To avoid generating a different modulus N = p.q for each user one may wish to fix N once and for all. The same N is used by all users. A trusted central authority could provide user i with a unique pair e_i, d_i from which user i forms a public key (N, e_i) and a secret key (N, d_i).

• *Department of Mathematical Sciences & Computer Applications, Bundelkhand University, Jhansi, UP, India, PH-09452736650. E-mail: sachinupadhyay2010@yahoo.co.in*

At first glance this may seem to work: a cipher text $C = M^{e_a} \bmod N$ intended for Alice cannot be decrypted by Bob since Bob does not possess d_a. However, this is incorrect and the resulting system is insecure. Bob can use his own exponents e_b , d_b to factor the modulus N. Once N is factored Bob can recover Alice's private key d_a from her public key e_a. This observation, due to Simmons, shows that an RSA modulus should never be used by more than one entity.

2.2 Blinding

Let (N, d) be Bob's private key and (N, e) be his corresponding public key. Suppose an adversary Marvin wants Bob's signature on a message $M \in Z^*_N$. Being no fool, Bob refuses to sign M. Marvin can try the following: he picks a random $r \in Z^*_N$ and sets $M' = r^e M \bmod N$. He then asks Bob to sign the random message M′. Bob may be willing to provide his signature S′ on the innocent-looking M′. But recall that $S' = M'^d \bmod N$. Marvin now simply computes $S = S'/r \bmod N$ and obtains Bob's signature S on the original M. Indeed

$$S^e = (S')^e / r^e = (M')^{ed} / r^e \equiv M'/ r^e = M \pmod N \quad (1)$$

This technique, called blinding, enables Marvin to obtain a valid signature on a message of his choice by asking Bob to sign a random "blinded" message. Bob has no information as to what message he is actually signing. Since most signature schemes apply a "one-way hash" to the message M prior to signing, the attack is not a serious concern. Although we presented blinding as an attack, it is actually a useful property of RSA needed for implementing anonymous digital cash.

3 Low Private Exponent

To reduce decryption time (or signature-generation time), one may wish to use a small value of d rather than a random d. Since modular exponentiation takes time linear in $\log_2 d$, a small d can improve performance by at least a factor of 10 (for a 1024 bit modulus). Unfortunately, a clever attack due to M. Wiener shows that a small d results in a total break of the cryptosystem.

Theorem: (M.Wiener) Let N = p.q with $q < p < 2q$. Let $d < 1/3 N^{1/4}$. Given (N, e) with $e.d = 1 \bmod \varphi(N)$, Marvin can efficiently recovered

$$\left|\frac{e}{\varphi(N)}-\frac{k}{d}\right|\le\frac{1}{d\varphi(N)}$$

Using N in place of φ (N), we obtained:

$$\left|\frac{e}{N}-\frac{k}{d}\right|=\left|\frac{ed-k\varphi(N)-kN+k\varphi(N)}{Nd}\right|$$
$$=\left|\frac{l-k(N-N+\varphi(N))}{Nd}\right|\le\left|\frac{3k\sqrt{N}}{Nd}\right|=\frac{3k}{d\sqrt{N}}$$

Now k φ(N) = ed-1< ed.
Since e < φ (N), we see that k < d ½ $N^{1/4.}$ Hence we obtained

$$\left|\frac{e}{N}-\frac{k}{d}\right|\le\frac{1}{dN^{1/4}}\le\frac{1}{2d^2} \qquad (2)$$

This is a classic approximation relation. The number of fractions k/d with d < N approximating e/N so closely is bounded by $\log_2 N$. In fact, all such fractions are obtained as convergent of the continued fraction expansion of e/ N. All one has to do is compute the logN convergent of the continued fraction for e/N One of these will equal k/d. Since ed- kψ (N) = 1, we have gcd (k, d) = 1, and hence k/d is a reduced fraction. This is a linear-time algorithm for recovering the secret key d.

Since typically N is 1024 bits, it follows that d must be at least 256 bits long in order to avoid this attack. This is unfortunate for low-power devices such as "smartcards", where a small d would result in big savings. All is not lost however. Wiener describes a number of techniques that enable fast decryption and are not susceptible to his attack.

4 Low Public Exponent

To reduce encryption or signature-verification time, it is customary to use a small public exponent e. The smallest possible value for e is 3, but to defeat certain attacks the value e = 2^16 +1 = 65537 is recommended. When the value 2^16 +1 is used, signature verification requires 17 multiplications, as opposed to roughly 1000 when a random e < ψ (N) is used. Unlike the attack of the previous section, attacks that apply when a small e is used are far from a total break.

4.1 Hastad's Broadcast Attack

As a first application of Coppersmith's theorem, we present an improvement to an old attack due to Hastad. Suppose Bob wishes to send an encrypted message M to a number of parties $P_1, P_2, \ldots P_k$. Each party has its own RSA key (N_i, e_i) we assume M is less than all the N_i. Naively, to send M, Bob encrypts it using each of the public keys and sends out the i^{th} ciphertext to P_i. An attacker Marvin can eavesdrop on the connection out of Bob's sight and collect the k transmitted cipher texts. For simplicity, suppose all public exponents e_i are equal to 3. A simple argument shows that Marvin can recover M if k >= 3. Indeed, Marvin obtains C_1, C_2, C_3, where

$C_1 = M^3 \bmod N_1 \quad C_2 = M^3 \bmod N_2 \quad C_3 = M^3 \bmod N_3$

We may assume that gcd $(N_i, N_j) = 1$ for all $i \neq j$ since otherwise Marvin can factor some of the N_i's. Hence, applying the Chinese Remainder Theorem (CRT) to C_1, C_2, C_3 gives a $C' \in Z_{N1N2N3}$ satisfying C′ = M^3 mod $N_1N_2N_3$. Since M is less than all the N_i's, we have M^3 < $N_1N_2N_3$. Then C′ = M^3 holds over the integers. Thus, Marvin may recover M by computing the real cube root of C. More generally, if all public exponents are equal to e, Marvin can recover M as soon as k ≥ e. The attack is feasible only when a small e is used.

Hastad describes a far stronger attack. To motivate Hastad's result, consider a naive defence against the above attack. Rather than broadcasting the encryption of M, Bob could "pad" the message prior to encryption. For instance, if M is m bits long, Bob could send $M_i = i2^m + M$ to party P_i. Since Marvin obtains encryptions of different messages, he cannot mount the attack. Unfortunately, Hastad showed that this linear padding is insecure. In fact, he proved that applying any fixed polynomial to the message prior to encryption does not prevent the attack.

4.2 Franklin-Reiter Related Message Attack

Franklin and Reiter found a clever attack when Bob sends Alice related encrypted messages using the same modulus. Let (N, e) be Alice's public key. Suppose $M_1, M_2 \in Z^*_N$ are two distinct messages satisfying $M_1 = f(M_2) \bmod N$ for some publicly known polynomial. To send M_1 and M_2 to Alice, Bob may naively encrypt the messages and transmit the resulting cipher texts C_1, C_2. We show that given C_1, C_2, and Marvin can easily recover M_1, M_2. Although the attack works for any small e, we state the following lemma for e = 3 in order to simplify the proof .Coppersmith's Short Pad Attack The Franklin-Reiter attack might seem a bit artificial. After all, why should Bob send Alice?

Lemma (FR) Set e = 3 and let (N, e) be an RSA public key. Let $M_1 \neq M_2 \in Z^*_N$ satisfy $M_1 = f(M_2) \bmod N$ for some linear polynomial $f = ax + b \in Z_N[x]$ with b≠0. Then given (N, e, C_1, C_2, f), Marvin can recover M_1, M_2 in time quadratic in log N.

Proof: To keep this part of the proof general, we state in using an arbitrary e (rather than restricting to e = 3). Since $C_1 = M_1^e \bmod N$, we know that M_2 is a root of the polynomial $g_1(x) = f(x)^e - C_1 \in Z_N[x]$, Similarly M_2 is a root of $g_2(x) = f(x)^e - C_2 \in Z_N[x]$. The linear factor $x-M_2$ divides both polynomials. Therefore, Marvin may use the Euclidean algorithm to compute the gcd of g_1 and g_2. If the gcd turns out to be linear, M_2 is found. The gcd can be computed in quadratic time in c and log N.

We show that when e = 3 the gcd must be linear. The polynomial $x^3 - C_2$ factors, modulo both p and q into a linear factor and an irreducible quadratic factor (recall that gcd(e,ψ(N))=1 and hence $x^3 - C_2$ has only one root in Z_N). Since g_2 cannot divide g_1, the gcd must be linear, for e > 3 the gcd is almost al-

ways linear. However for some rare M_1, M_2 and f, it is possible to obtain a nonlinear gcd, in which case the attack fails.

For e > 3 the attack takes time quadratic in e. Consequently, it can be applied only when a small public exponent e is used. For large e the work in computing the gcd is prohibitive. It is an interesting question to device such an attack for arbitrary e. In particular, can the gcd of g_1 and g_2 above be found in time polynomials in log e?

4.3 Partial Key Exposure Attack

Let (N, d) be a private RSA key, suppose by some means Marvin is able to expose a fraction of the bits of d, say a quarter of them. Can he reconstruct the rest of d? Surprisingly, the answer is positive when the corresponding public key is small. Recently Boneh, Durfee, and Frankel showed that as long as e < sqrt N, it is possible to reconstruct all of d from just a fraction of its bits. These results illustrate the importance of safeguarding the entire private RSA key.

Theorem: (Coppersmith) Let N = pq be an n-bit RSA modulus. Then given the n/4 least significant bits of p or the n/4 most significant bits of p, one can efficiently factor N.
By definition of e and d, there exists an integer k such that

$$ed-k\ (N-p-q+1)=1 \qquad (3)$$

Since $d< \varphi(N)$, we must have $0 \le k \le e$. Reducing the equation modulo $2^{n/4}$ and setting $q = N/p$, we obtain

$$(ed)\ p - kp\ (N - p+1)+ kN = p\ (\text{mod } 2^{n/4}) \qquad (4)$$

Since Marvin is given the n/4 least significant bits of d, he knows the value of ed mod $2^{n/4}$. Consequently, he obtains equations an equation in k and p. For each of the e possible values of k, Marvin solves the quadratic equation in p and obtains a number of candidate values for p mod $2^{n/4}$. For each of these candidate values, he runs the algorithms of above theorem in order to factor N. One can show that the total number of candidate values for p mod $2^{n/4}$ is at most e $\log_2 e$. Hence after at most e $\log_2 e$ attempts, N will be factored.
Finally when the encryption exponent e is small, the RSA system leaks half the most significant bits of the corresponding private key d. To see this consider once again the equation ed-k (N-p-q+1) = 1 for an integer $0 \le k \le e$. Given k, Marvin may easily compute

$$d' = [(kN + 1)/e]$$

Then

$$|d'-d| \le k\ (p+q)/e \le 3k\sqrt{N}/e < 3\sqrt{N} \qquad (5)$$

Hence, d' is a good approximation for d. The bound shows that, for most d, half the most significant bits of d' are equal to those of d. Since there are only e possible values for k, Marvin can construct a small set of size e such that one of the element in the set is equal to half the most significant bits of d. The case e = 3 is especially interesting. In this case one can show that always k = 2 and hence the system completely leaks half the most significant bits of d.

5 Implementation Attack

Now we are moving to a completely different type of attacks. Rather than attacking the underlying structure of the RSA function, these attacks focus on the implementation of RSA

5.1 Random Faults

Implementation of RSA decryption and signature frequently use the CRT to speed up the computation of M^d mod N. Instead of working modulo N, the signer Bob first computes the signatures modulo p and q and then combines the results using the CRT. More precisely, Bob first computes

$$C_p = M^{d_p} \bmod p \quad \text{and} \quad C_q = M^{d_q} \bmod q$$

Where $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$. He then obtains the signature C by setting

$$C = T_1 C_p + T_2 C_q \ (\text{mod } N) \qquad (6)$$

Where

$$T_1 = \begin{cases} 1 \bmod p \\ 0 \bmod q \end{cases} \quad \text{and} \quad T_2 = \begin{cases} 0 \bmod p \\ 1 \bmod q \end{cases}$$

The running time of the last CRT step is negligible compared to the two exponentiations. Note that p and q are half the length of N. Since simple implementations of multiplication take quadratic time, multiplication modulo p is four times faster than modulo N. Furthermore, d_p is half the length of d and consequently computing $M^{d_p} \bmod p$ is eight times faster than computing M^d mod N. Overall signature time is thus reduced by a factor of four. Many implementations use this method to improve performance.

Boneh, DeMillo and Lipton observed that there is an inherent danger in using the CRT method. Suppose that while generating a signature, a glitch on Bob's computer causes it to miscalculate in a single instruction. Say, while copying a register from one location to another, one of the bits is flipped. Given an invalid signature, an adversary Marvin can easily factor Bob's modulus N.

Suppose a single error occurs while Bob is generating a signature. As a result, exactly one of C_p or C_q will be computed incorrectly. Say C_p is correct, but C'_q is not. The resulting signature C', he knows it is a false signature since $C'^e \ne M \bmod N$. However, notice that

$$C'^e = M \bmod p \quad \text{while} \quad C'^e \ne M \bmod q$$

As a result, gcd $(N, C'^e - M)$ exposes a nontrivial factor of N.
For the attack to work Marvin must have full knowledge of M. namely, we are assuming Bob does not use any random padding procedure. Random padding prior to signing defeats the attack. A simpler defense is for Bob to check the generated signature before sending it out to the world. Checking is espe-

cially important when using the CRT speedup method. Random faults are hazardous to many cryptographic systems. Many systems, including a non-CRT implementation of RSA, can be attacked using random faults. However, these results are far more theoretical.

5.2 Bleichenbacher's Attack on PKCS 1

Let N be an n-bit RSA modulus and M be an m-bit message with m < n. Before applying RSA encryption it is natural to pad the message M to n bits by appending random bits to it. An old version of standard known as Public Key Cryptography Standard 1 (PKCS 1) uses this approach. After padding the message looks as follows:

02	Random	00	M

The resulting message is n bits long and is directly encrypted using RSA. The initial block containing "02" is 16 bits long and is there to indicate that a random pad has been added to the message.

When PKCS 1 message is received by Bob's machine, an application decrypts it, checks the initial block, and strips off the random pad. However some applications check for the "02" initial block and if it is not present they send back an error message saying "invalid message". Bleichenbacher showed that this error message can lead to disastrous consequences: using the error message, an attacker Marvin can decrypt cipher text of his choice.

Suppose Marvin intercepts a cipher text C intended for Bob and wants to decrypt it. To mount the attack, Marvin picks a random $r \in Z^*_N$, computes $C' = rC \bmod N$, and sends C′ to Bob's machine. An application running on Bob's machine receives C′ and attempts to decrypt it. It is either responds with an error message or does not respond at all. Hence, Marvin learns whether the most significant 16 bits of the decryption of C′ are equal to 02. In effect, Marvin test whether the 16 bits most significant bits of the decryption of rC mod N are qual to 02, for any r of his choice. Bleichenbacher showed that such an attempt is sufficient for decrypting C.

6 Conclusion

Two decades of research into inverting the RSA function produced some insightful attacks, but no devastating attack has ever been found. The attacks discovered so far mainly illustrate the pitfalls to be avoided when implementing RSA. At the moment it appears that proper implementations can be trusted to provide security in the digital world. These attacks illustrate that a study of the underlying mathematical structure is insufficient. Desmedt and Odlyzko , Joye and Quisquater and deJonge and Chaum describe some additional attacks. Throughout the paper we observed that many attacks can be defeated by properly padding the message prior to encryption or signing.

References

[1] D. Boneh, G.Durfee. New results on cryptanalysis of low private exponent RSA. Preprint, 1998.

[2] D.Bleichenbacher. chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS 1. In CRYPTO '98, volume 1462 of Lecture Notes in Computer Science, pages 1-12, Springer-Verlag, 1998

[3] D. Boneh, R.DeMillo and R. Lipton. On the importance of checking cryptographic protocols for faults. In EUROCRYPT '97, volume 1233 of Lectuer Notes in Computer Science, pages 37 51.Springer-Verlag, 1997.

[4] P.Kocher. Timing attacks on implementations od Diffie-Hellman, RSA, DSS and other systems. In CRYPTO'96, volume 1109 of lecture Notes in Computer Science, pages 104 113 Springer-Verlag, 1996.

[5] J.Hastad Solving simultaneous modular equations of low degree, SIAM J. of Computing, 17:336-341, 1988.