EULER CIRCLE PAPER: MISCELLANEOUS ATTACKS ON RSA

MEHANA ELLIS

Abstract. The RSA (Rivest-Shamir-Adleman), which was introduced in 1977, is used most often when we need to protect digital data and keep it secure. I will be referencing Dan Boneh's article "Twenty Years of Attacks on the RSA Cryptosystem" [B$^+$], as well as articles by Hinek et al [HLT02], Dujella [Duj09], and Cryptography by Simon Rubinstein-Salzedo [RS18] (some theorems and definitions I will be paraphrasing/citing, when I paraphrase I will also put a citation). It is important to understand how RSA is attacked in order to be able to further secure RSA. I will assume basic knowledge of RSA, but I will explain some of the basics in the beginning. We will also look at multi-prime RSA, a type of RSA where the modulus is equal to a product of more than two primes. In this paper, I hope to investigate a variety of attacks on some of the uses of RSA.

1. Introduction

To begin, we will see a brief explanation of how to generate keys for the RSA algorithm [RSA78]:

Definition 1.1. Begin by choosing two primes p and q with around the same number of digits. Alice[1] will compute $n = pq$, then $\phi(n) = (p-1)(q-1)$. Now Alice picks some integer e, where $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$ [2]. Now she computes $d = e^{-1} \pmod{\phi(n)}$. This gives us the private key, d, and the public key consisting of n and e. [RS18]

We see here that, like cryptosystems such as ElGamal (using Diffie-Hellman), we have a public and private key, but in RSA only one person must create a key, which makes RSA more efficient than other cryptosystems.

Definition 1.2. Alice's message m can be encrypted by computing $m^e \pmod{n} = c$. Generally, to help further secure the message, several digits are appended to m. To decrypt the message, Bob computes $c^d \pmod{n}$. [B$^+$]

It is worth mentioning that this appending of random digits to the end of the plaintext before encrypting it is called *padding*. Although it doesn't immediately seem to be related to actually attacking RSA, understanding how messages are secured is a big part of attacking the cryptosystem. There are two general types of padding, namely, randomized and deterministic, but these are not too difficult to understand and I won't focus much on padding in the rest of this paper.

Date: December 9, 2019.

[1] I will use the standard characters Alice, Bob, and Eve in this paper, where Alice and Bob are the protagonists and Eve is the antagonist.

[2] A common choice of e is $e = 2^{16} + 1$.

2. Some attacks on RSA

Now that we've seen the basics of RSA encryption, we can take a closer look at the ways of attacking it. We will be in the perspective of Eve, who wishes to decrypt the messages between Alice and Bob by attacking RSA. Of course, the first way of attacking RSA that comes to mind is to factor n. Eve then only has to find the primes p and q with product n. With this, she can easily find the decryption exponent and decrypt the message. However, this will be rather hard, because factoring n for very large numbers is challenging. So we will assume that we don't know the factorization of n and examine the ways in which we can still break RSA without knowing p and q. First, let's look at some important theorems:

Theorem 2.1 (Wiener)**.** *Let $N = pq$ with $q < p < 2q$. Let $d < \frac{1}{3}N^{\frac{1}{4}}$. If we have the public key consisting of N and e with $ed \equiv 1 \pmod{\phi(N)}$, it is easy for an attacker to recover d.* [B$^+$]

Here is the most well-known proof of Wiener's theorem, based on continued fractions:

Proof Notice that $ed - a\phi(N) = 1$ for some a, because $ed \equiv 1 \pmod{\phi(N)}$, so we have

$$\left|\frac{e}{\phi(N)} - \frac{a}{d}\right| = \frac{1}{d\phi(N)}.$$

In other words, $\frac{e}{\phi(N)}$ is approximately equal to $\frac{a}{d}$. We now notice that $|N - \phi(N)| < 3\sqrt{N}$, because $\phi(N) = N - p - q + 1$ and $p + q - 1 < 3\sqrt{N}$. Now let us replace $\phi(N)$ with N:

$$\left|\frac{e}{N} - \frac{a}{d}\right| = \left|\frac{ed - a\phi(N) - aN + a\phi(N)}{Nd}\right|.$$

This simplifies to

$$\left|\frac{1 - a(N - \phi(N))}{Nd}\right| \le \frac{3a}{d\sqrt{N}}.$$

We know that $a\phi(N) < ed$ and $e < \phi(N)$, so $a < d < \frac{1}{3}N^{\frac{1}{4}}$. Finally, we have

$$\left|\frac{1 - a(N - \phi(N))}{Nd}\right| \le \frac{1}{dN^{\frac{1}{4}}} < \frac{1}{2d^2}.$$

Therefore, we have recovered d efficiently given the information in Wiener's Theorem, which is what we wanted. Boneh notes that this is essentially an algorithm that takes $O(n)$ time.

Boneh frequently mentions the Coppersmith Theorem, because it is useful when working with a composite modulus. First we will see the definition of the Lenstra-Lenstra-Lovász algorithm, which is mentioned in the Coppersmith Theorem:

Definition 2.2 (Lenstra-Lenstra-Lovász)**.** Let there be a basis $B = \{b_0, b_1, b_2, \dots, b_n\}$ such that its Gram-Schmidt process[3] orthogonal basis is $B* = \{b_0^*, b_1^*, b_2^*, \dots, b_n^*\}$ and the Gram-Schmidt coefficients are $\mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{langleb_j^*, b_j^* \rangle}$ for all $1 \le j < i \le n$. If there is a parameter δ in the interval $\left(\frac{1}{4}, 1\right)$ such that $1 \le i < j \le n : |\mu_{i,j}| \le \frac{1}{2}$ (or, adding another condition called Lovász's condition, for $k = 1, 2, \dots, n : \delta||b_{k-1}^*||^2 \le ||b_k^*||^2 + \mu_{k,k-1}^2||b_{k-1}^*||^2$), then B is

[3] The Gram-Schmidt process is a process in inner product space where we orthonormalize a set of vectors. See Wikipedia's page on the LLL algorithm for more info.

LLL-reduced. Note that the larger the value of δ, the stronger the basis reduction. [Wik19]

A consequence of this algorithm is as follows: Let L be a lattice spanned by $\langle v_1, v_2, \dots, v_w\rangle$[4]. If we input this span, we will get a point $v \in L$ that satisfies

$$||v|| \leq 2^{\frac{w}{4}} \det(L)^{\frac{1}{w}}.$$

Theorem 2.3 (Coppersmith)**.** *Let n be an integer and $f \in \mathbb{Z}[x]$ be a monic polynomial of degree d. Let $X = n^{\frac{1}{d}-\epsilon}$ for some $\epsilon \geq 0$. Then, given $\langle n, f\rangle$, an attacker easily can find all integers $|x_0| < X$ satisfying $f(x_0) \equiv 0 \pmod{N}$. We can determine the running time by the time it takes to run the Lenstra–Lenstra–Lovász (LLL) lattice basis reduction algorithm on a lattice of dimension $O(w)$ with $w = \min(\frac{1}{\epsilon}, \log_2 n)$.* [B$^+$]

Coppersmith's Theorem has many applications such as Hastad's attack and a consequent stronger version, which we will see later on. Here are some more (miscellaneous) theorems we will use for the different attacks:

Theorem 2.4 (Chinese Remainder Theorem/CRT)**.** *Suppose we want to find the smallest value of x leaving a remainder of y_1 when divided by d_1, a remainder of y_2 when divided by d_2, etc., y^n when divided by d_n, where these values $d_1, d_2, \dots, d_n$ are relatively prime. Also, let $m = d_1 d_2 \cdots d_n$, and $b_i = \frac{m}{d_i}$. If, for all $1 \leq i \leq n$, $a_i b_i \equiv 1 \pmod{d_i}$ is satisfied by the numbers a_i, then we have*

$$x = \sum_{i=1}^{n} a_i b_i y_i \pmod{m}$$

(AoPS Wiki).

Theorem 2.5 (Boneh-Durfee-Frankel)**.** *Let $N = pq$ be an n-bit RSA modulus with $N = 3 \pmod 4$. Let $1 \leq e,\ d \leq \phi(N)$ satisfy $ed \equiv 1 \pmod{\phi(N)}$ and $e^{2^{\frac{n}{4}-3}}$. Then there is an algorithm that given $\langle N, e\rangle$ and the $\frac{n}{4}$ least significant bits of d computes all of d in polynomial time in n and e.* [BDF98]

Note that there is a different, modified way of writing the latter theorem mentioned in the article "Exposing an RSA Private Key Given a Small Fraction of its Bits" by Boneh, Durfee, and Frankel.

Now that we have most of the information needed to attack RSA in some ways, we can look at a few of the attacks. The first one we'll see is called Wiener's attack, which is based off of Wiener's theorem, except that the modulus is slightly different than in the theorem:

Wiener's Attack. Let $\lambda(N) = \frac{\phi(N)}{G}$, where $G = \gcd(p-1, q-1)$. We know that $ed \equiv 1 \pmod{\lambda(N)}$; there must exist some integer K such that $ed = \frac{K(p-1)(q-1)}{G} + 1$. Let $k = \frac{K}{\gcd(K,G)}$ and $g = \frac{G}{\gcd(K,G)}$. Substituting these values into the previous equation for ed gives us $ed = \frac{k(p-1)(q-1)}{g} + 1$. Now we can be clever and divide by dpq to get $\frac{e}{pq} = \frac{k}{dg}(1 - \frac{p+q-1-(g/k)}{pq})$. We can generally assume that $ed > pq$, but this may not always be the case. If we assume that this is the case, however, then we can write the previous equation as $egd = k(p-1)(q-1) + g$, or $egd = k(\phi(n)) + g$. In all, the algorithm will find $\frac{k}{dG}$.

[4] Some notes on the terminology: The span of a lattice is the linear space spanned by its vectors, and det is the determinant. The determinant is a scalar value that can be obtained by certain computations on elements of a square matrix.

Now let's put ourselves in the minds of Alice and Bob, where we want to protect our messages from Eve's decryption. Logically, as the value of e gets larger and larger, the encryption time will become longer and longer. This means that if we choose e to be some very large number, it will take a very long time for Eve to decrypt our message. The problem with Wiener's attack is that there are some instances in which it is ineffective.

Let's look at the significance of Coppersmith's theorem in attacks on RSA. With this theorem, it is easier to find solutions to the monic polynomial f of degree d. Generally we shouldn't use this formula with a prime modulus, but the power of this theorem is that we can find small polynomial roots working with a composite modulus n. Another reason for the importance of Coppersmith's theorem is that N being composite lends itself nicely to RSA, where N is the product of two primes. There is one other theorem by Coppersmith that is worth mentioning:

Theorem 2.6 (Coppersmith)**.** *Let $N = pq$ be an n-bit RSA modulus. Then given the $\frac{n}{4}$ least significant or most significant bits*[5] *of p, one can efficiently factor N.* [B$^+$]

According to Boneh, we can factor N in at most $e \log_2 e$ attempts. This is overall pretty useful, even if e gets fairly large, because assuming we have a computer that can do large computations, this is not too challenging for an attacker to do, although it is rather time consuming. Recall that all we really need to do to attack RSA is to factor the modulus N. It is not hard to do so, and from there we can discover the key $\langle N, d\rangle$. This is due to the latter theorem, which leads us to think that perhaps the private key d can be easily found. It turns out that we can indeed find this key and generalize the theorem: If we have the $\frac{n}{4}$ least or most significant bits of p (one of the two prime factors of N), we can factor N. From this, we know what $\phi(n) = (p-1)(q-1)$ is, and d is the same as $e^{-1} \pmod{\phi(n)}$. This is somewhat related to the Boneh-Durfee-Frankel theorem in that we put the least (and most, in Coppersmith's theorem) significant bits to use in order to compute d or anything that will tell us what d is.

We can now use Coppersmith's Theorem (2.3) for some attacks. The first attack we will examine is called Hastad's attack, involving "eavesdropping" and intercepting ciphertexts:

Hastad's Attack. Suppose Alice would like to send an encrypted message m to multiple people, $p_1, p_2, p_3, \ldots, p_k$. Everyone has eir own key, namely $\langle N_i, e_i\rangle$, where $N_i > m$ for all N_i. Alice, knowing that the ciphertext she sends will be different depending on the recipient, sends person p_j the j^{th} ciphertext for $j \in \{1, 2, 3, \ldots, k\}$. If she eavesdrops, Eve is able to intercept all k of the ciphertexts, and from this she can figure out m. [B$^+$]

This is due to the fact that Eve is able to find m if every public exponent is the same (when the public exponent e is small, if it is large then we will need more ciphertexts). There is also Hastad's Theorem, which gives us a more generalized explanation of his attack:

Theorem 2.7 (Hastad)**.** *Suppose $N_1, N_2, N_3, \ldots, N_k$ are pairwise relatively prime (i.e., every pair is relatively prime) integers. Let $N_{\min} = \min i(N_i)$, and let $g_i \in \mathbb{Z}_{N_i}[x]$ be k polynomials of maximum degree d. Suppose $g_i(m)$ is a multiple of N_i for all $i \in 1, 2, 3, \ldots, k$, where $m < N_{\min}$. If $k > d$, it is not too hard to find m for $\langle N_i, g_i\rangle_{i=1}^{k}$.* [B$^+$]

Because we are dealing with polynomials here, these seem to hint at using Coppersmith's theorem in the proof of Hastad's Theorem. The proof is partly based off of CRT, but we can

[5] The most significant bit of a binary string is the first bit of the string, and the least significant bit is the last bit of the string.

actually learn what the Chinese Remainder coefficients are by looking at the proof based on Boneh's proof of this:

Proof Let $N = n_1 \times n_2 \times n_3 \times \ldots \times n_k$. Notice that if we multiply the polynomials g_i by some power of x, all of them will have degree d (see footnote)[6]. To get something that looks similar to CRT, we can let

$$g(x) = \sum_{i=1}^{k} T_i g_i(x),$$

where T_i is either equivalent to $1 \pmod{n_a}$ if $i = a$ or $0 \pmod{n_a}$ if $i \neq a$. We know now that $g(x)$ must be monic with degree d, because $g(x)$ is monic modulo each n_i, and thus we have $g(m) \equiv 0 \pmod{N}$. Because $m < n_{\min} \leq N^{\frac{1}{k}} < N^{\frac{1}{d}}$, the required restrictions, we know that Hastad's Theorem (2.7) follows from Coppersmith's Theorem (2.3).

Next, let's think about the Chinese Remainder Theorem and how it can be applied to RSA. We might wonder how this popular theorem in number theory relates to cryptography, but notice that it involves a lot of modular arithmetic, which is useful because it can help make the computation time of $a^d \pmod{N}$ quicker. I will try to elaborate on using the Chinese Remainder Theorem for RSA. Given the pair $(a^d \pmod{p}, a^d \pmod{q})$, we know by CRT that we can find a value of $a^d \pmod{N}$, where $0 \leq a^d \pmod{N} \leq N-1$. After some guesses as to what formula we should use next, I found that using Euler's Formula (I will assume knowledge of this) on $a^d \pmod{p}$ and $a^d \pmod{q}$ will give us $a^{d \pmod{\phi(p)}}$ and $a^{d \pmod{\phi(q)}}$, respectively. These become $a^{d \pmod{p-1}} \pmod{p}$ and $a^{d \pmod{q-1}} \pmod{q}$, respectively. We then find that $\phi(p) = p-1$ and $\phi(q) = q-1$, which is how we come up with $\phi(n) = (p-1)(q-1)$ due to the fact that $n = pq$.

3. Less common attacks for a different type of RSA

We will now look at attacks on a special type of RSA, namely, multi-prime RSA. One of the first questions that might have come to mind when we were first learning about RSA encryption was, "What would happen if we had a system of RSA where the modulus N was equal to a product of more than two primes?" Luckily, I found a paper by Hinek et al, in which the authors tell us about such a thing: multi-prime RSA. The main thing to notice here is that the higher the value of r in r-prime RSA, this value will not necessarily determine whether we are able to easily find a factor of $N = p_1 p_2 p_3 \ldots p_r$. First, let's introduce a general type of algorithm:

Definition 3.1. A Las Vegas algorithm is a randomized algorithm (an algorithm that is based partly on randomness) that only gives a correct output (it will either say the correct output or let us know of a failure).

In the paper "On Some Attacks on Multi-prime RSA," we are given a few probabilities for finding a factor p_a of N based on a Las Vegas algorithm: The probability of finding p_a is

[6] We assume g_i's are monic (polynomials in which the leading coefficient is 1).

about $\frac{3}{4}$ for 3-prrime RSA and the probability of finding p_a for 4-prime RSA is about $\frac{7}{8}$.

When thinking about attacks on multi-prime RSA, one thing that immediately comes to mind is a modification of Wiener's attack.

Definition 3.2. If $p < q < 2p$, $e < N$, and $d < \frac{\sqrt[4]{N}}{3}$, then d is the denominator of a convergent when we apply the continued fraction expansion to $\frac{e}{n}$. [Duj09]

Modified Wiener's Attack. Let $\lambda(N) = \frac{\phi(N)}{G}$, where $G = \gcd(p-1, q-1)$. We know that $ed \equiv 1 \pmod{\lambda(N)}$; there must exist some integer K such that $ed = \frac{K(p-1)(q-1)}{G} + 1$. Let $k = \frac{K}{\gcd(K,G)}$ and $g = \frac{G}{\gcd(K,G)}$. We have

$$\frac{k}{d} - \frac{e}{N} = \frac{k(N - \phi(N)) - 1}{d \cdot N}.$$

For $d < \frac{N^{1/2r}}{\sqrt{2(2r-1)}}$, $\frac{k}{d}$ will be a convergent when we apply the continued fraction expansion to $\frac{e}{N}$. [HLT02]

Here, Hinek provided the basis to the attack I have shown. To find the private exponent, all we have to do is look at all the convergents of the continued fraction expansion of $\frac{e}{N}$. This will prove to be not very difficult for a determined attacker. This leads us to the final theorem concerning r-prime RSA:

Theorem 3.3 (Hinek-Low-Teske)**.** *Let N be an r-prime modulus and $d < \frac{N^{1/2r}}{\sqrt{2(2r-1)}}$. Given $\langle N, e\rangle$, the decryption exponent can be recovered in polynomial time in $n = \log N$.* [HLT02]

This theorem is a result of the modified Wiener's attack, because the latter attack is in polynomial time.

4. Conclusion

In this paper, we learnt about some miscellaneous attacks on the RSA cryptosystem. Although I was not aiming specifically to prove anything, the most interesting and notable things I found were the r-prime system of RSA and the modified version of Wiener's attack for r-prime RSA. We examined the reasons why some of these attacks are successful under certain conditions, and overall we saw that some of the most important theorems concerning RSA are Coppersmith's Theorems (2.3 and 2.6). Although much of my work in this paper is credited to the authors of the different articles I read, I showed a potential use of the Chinese Remainder Theorem for RSA and examined why the attacks mentioned work.

References

[B$^+$] Dan Boneh et al. Twenty years of attacks on the rsa cryptosystem. *Notices of the AMS*, 46(2):203–213.

[BDF98] Dan Boneh, Glenn Durfee, and Yair Frankel. An attack on rsa given a small fraction of the private key bits. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 25–34. Springer, 1998.

[Duj09] Andrej Dujella. A variant of wiener's attack on rsa. *Computing*, 85(1-2):77–83, 2009.

[HLT02] M Jason Hinek, Mo King Low, and Edlyn Teske. On some attacks on multi-prime rsa. In *International Workshop on Selected Areas in Cryptography*, pages 385–404. Springer, 2002.

[RS18] Simon Rubinstein-Salzedo. *Cryptography.* Springer Undergraduate Mathematics Series. Springer, Cham, 2018.

[Wik19] Wikipedia contributors. Lenstra–lenstra–lovász lattice basis reduction algorithm — Wikipedia, the free encyclopedia, 2019. [Online; accessed 8-December-2019].