4.2 The RSA Cryptosystem

The Diffie-Hellman key exchange has drawbacks. As discussed in Section 4.1.3, it is susceptible to the man in the middle attack, so one isn't always sure who they are exchanging messages with. Also, it only provides a way to agree on a secret key, not a way to encrypt any information; for that, one must rely on a symmetric-key encryption method. This section is about the RSA public-key cryptosystem of Rivest, Shamir, and Adleman [RSA78], which remedies some of these defects.

In this section we describe the RSA cryptosystem, then discuss several ways to attack it, which we must be aware of in order to implement the cryptosysem without making foolish mistakes.

4.2.1 How RSA works

The fundamental idea behind RSA is to try to construct a so-called "one-way function" on a set X, that is, an invertible function

$$E : X \to X$$

such that it is easy for Nikita to compute E^{-1}, but extremely difficult for anybody else to do so.

Here is how Nikita makes a one-way function E on the set of integers modulo n.

1. Nikita picks two large primes p and q, and lets $n = pq$.
2. It is easy for Nikita to then compute
$$\varphi(n) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1).$$
3. Nikita next chooses a "random" integer e with
$$1 < e < \varphi(n) \text{ and } \gcd(e, \varphi(n)) = 1.$$
4. Nikita uses the algorithm from Section 3.5.2 to find a solution $x = d$ to the equation
$$ex \equiv 1 \pmod{\varphi(n)}.$$
5. Finally, Nikita defines a function $E : \mathbb{Z}/n \to \mathbb{Z}/n$ by
$$E(x) = x^e \in \mathbb{Z}/n.$$
Anybody can compute E fairly quickly using the repeated-squaring algorithm from Section 3.5.2.

Nikita's ***public key*** is the pair of integers (n, e), which is just enough information for people to easily compute E. Nikita knows a number d such that $ed \equiv 1 \pmod{\varphi(n)}$, so, as we will see below, she can quickly compute E^{-1}.

To send Nikita a message, proceed as follows. Encode your message, in some way, as a sequence of numbers modulo n (see Section 4.2.2)

$$m_1, \ldots, m_r \in \mathbb{Z}/n,$$

then send

$$E(m_1), \ldots, E(m_r)$$

to Nikita. (Recall that $E(m) = m^e$.)

When Nikita receives $E(m_i)$, she finds each m_i by using that $E^{-1}(m) = m^d$, a fact that follows from the following proposition.

Proposition 4.2.1. *Let n be an integer that is a product of distinct primes and let $d, e \in \mathbb{N}$ such that $p-1 \mid de-1$ for each prime $p \mid n$. Then $a^{de} \equiv a \pmod{n}$ for all $a \in \mathbb{Z}$.*

Proof. Since $n \mid a^{de} - a$ if and only if $p \mid a^{de} - a$ for each prime divisor p of n, it suffices to prove that $a^{de} \equiv a \pmod{p}$ for each prime divisor p of n. If $\gcd(a,p) \neq 0$, then $a \equiv 0 \pmod{p}$, so $a^{de} \equiv a \pmod{p}$. If $\gcd(a,p) = 1$, then Theorem 3.3.14 asserts that $a^{p-1} \equiv 1 \pmod{p}$. Since $p-1 \mid de-1$, we have $a^{de-1} \equiv 1 \pmod{p}$ as well. Multiplying both sides by a shows that $a^{de} \equiv a \pmod{p}$. □

Thus to decrypt $E(m_i)$ Nikita computes

$$m_i = E^{-1}(E(m_i)) = E(m_i)^d = (m_i^e)^d = m_i.$$

4.2.2 Encoding a Phrase in a Number

In order to use the RSA cryptosystem to encrypt messages, it is necessary to encode them as a sequence of numbers of size less than $n = pq$. We now describe a simple way to do this.

Think of a sequence of capital letters and spaces as a number in base 27 as follows. Let a single space correspond to 0, the letter A to 1, B to 2, ..., Z to 26. Thus, e.g., "HARVARD" denotes a number written in base 27. The corresponding number written in decimal is 1808939906:

$$\begin{aligned}\text{HARVARD} &\leftrightarrow 8 + 27 \cdot 1 + 27^2 \cdot 18 + 27^3 \cdot 22 + 27^4 \cdot 1 + 27^5 \cdot 18 + 27^6 \cdot 4 \\ &= 1808939906\end{aligned}$$

To recover the digits of the number, repeatedly divide by 27 and read off the remainder:

$$\begin{array}{rcrcrl}
1808939906 & = & 66997774 \cdot 27 & + & 8 & \text{“H”} \\
66997774 & = & 2481399 \cdot 27 & + & 1 & \text{“A”} \\
2481399 & = & 91903 \cdot 27 & + & 18 & \text{“R”} \\
91903 & = & 3403 \cdot 27 & + & 22 & \text{“V”} \\
3403 & = & 126 \cdot 27 & + & 1 & \text{“A”} \\
126 & = & 4 \cdot 27 & + & 18 & \text{“R”} \\
4 & = & 0 \cdot 27 & + & 4 & \text{“D”}
\end{array}$$

How many letters can a number hold? If $27^k < n$, then k letters can be encoded in a number $< n$. Put another way,

$$k < \log(n)/\log(27) = \log_{27}(n).$$

4.2.3 Examples

So the arithmetic is easy to follow, we use small primes p and q and encrypt the single letter "X".

1. Choose p and q: Let $p = 17$, $q = 19$, so $n = pq = 323$.
2. Compute $\varphi(n)$:
$$\begin{aligned}\varphi(n) &= \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1) \\ &= pq - p - q + 1 = 323 - 17 - 19 + 1 = 288.\end{aligned}$$
3. Randomly choose an $e < 288$: We choose $e = 95$.
4. Solve
$$95x \equiv 1 \pmod{288}.$$
Using the GCD algorithm, we find that $d = 191$ solves the equation.

The public key is $(323, 95)$, so the encryption function $E : \mathbb{Z}/323 \to \mathbb{Z}/323$ is defined by
$$E(x) = x^{95},$$
and the decryption function is $D(x) = x^{191}$.

Next, we encrypt the letter "X". It is encoded as the number 24, since X is the 24th letter of the alphabet. We have
$$E(24) = 24^{95} = 294 \in \mathbb{Z}/323.$$

To decrypt, we compute E^{-1}:
$$E^{-1}(294) = 294^{191} = 24 \in \mathbb{Z}/323.$$

This example illustrates RSA but with bigger numbers. Let
$$p = 738873402423833494183027176953, \; q = 3787776806865662882378273.$$
Then
$$n = p \cdot q = 2798687536910915970127263606347911460948554197853542169$$
and
$$\begin{aligned}\varphi(n) &= (p-1)(q-1) \\ &= 2798687536910915970127262867470721260308194351943986944.\end{aligned}$$
We somehow randomly chose
$$e = 1483959194866204179348536010284716655442139024915720699.$$
Then
$$d = 2113367928496305469541348387088632973457802358781610803$$

Since $\log_{27}(n)/\log(27) \approx 38.04$, we can encode then encrypt single blocks of up to 38 letters. Let's encrypt "HARVARD", which is encoded as $m = 1808939906$. We have

$$E(m) = 625425724974078486559370130768554070421628674916144724.$$

It is also interesting to note that changing the input message even slightly completely changes the encrypted version. For example, "HARVAHD" is encoded as $m' = 1665450836$ and encrypted as

$$E(m') = 437968760439188600589414766639328726464015666686231875.$$

4.3 Attacking RSA

Nikita's public key is (n, e). If we compute the factorization of $n = pq$, then we can compute $\varphi(n)$ and hence deduce her secret decoding number d. Thus attempting to factor n is a way to try to break an RSA public-key cryptosystem. In this lecture we consider several approaches to "cracking" RSA, and relate them to the difficulty of factoring n.

4.3.1 Factoring n Given $\varphi(n)$

Suppose $n = pq$. Given $\varphi(n)$, it is very easy to compute p and q. We have

$$\varphi(n) = (p-1)(q-1) = pq - (p+q) + 1,$$

so we know both $pq = n$ and $p + q = n + 1 - \varphi(n)$. Thus we know the polynomial

$$x^2 - (p+q)x + pq = (x-p)(x-q)$$

whose roots are p and q. These roots can be found using the quadratic formula.

Example 4.3.1. The number $n = pq = 31615577110997599711$ is a product of two primes, and $\varphi(n) = 31615577098574867424$. We have

$$\begin{aligned} f &= x^2 - (n + 1 - \varphi(n))x + n \\ &= x^2 - 12422732288x + 31615577110997599711 \\ &= (x - 3572144239)(x - 8850588049), \end{aligned}$$

where the last step is easily accomplished using the quadratic formula:

$$\begin{aligned} \frac{-b + \sqrt{b^2 - 4ac}}{2a} &= \frac{12422732288 + \sqrt{12422732288^2 - 4 \cdot 31615577110997599711}}{2} \\ &= 8850588049. \end{aligned}$$

We conclude that $n = 3572144239 \cdot 8850588049$.

4.3.2 When p and q are Close

Suppose that p and q are "close" to each other. Then it is easy to factor n using a factorization method of Fermat.

Suppose $n = pq$ with $p > q$, say. Then

$$n = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2.$$

Since p and q are "close",

$$s = \frac{p-q}{2}$$

is small,

$$t = \frac{p+q}{2}$$

is only slightly larger than $\sqrt{n}$, and $t^2 - n = s^2$ is a perfect square. So we just try

$$t = \lceil\sqrt{n}\rceil, \quad t = \lceil\sqrt{n}\rceil + 1, \quad t = \lceil\sqrt{n}\rceil + 2, \ldots$$

until $t^2 - n$ is a perfect square s^2. (Here $\lceil x \rceil$ denotes the least integer $n \geq x$.) Then

$$p = t + s, \qquad q = t - s.$$

Example 4.3.2. Suppose $n = 23360947609$. Then

$$\sqrt{n} = 152842.88\ldots.$$

If $t = 152843$, then $\sqrt{t^2 - n} = 187.18\ldots$.
If $t = 152844$, then $\sqrt{t^2 - n} = 583.71\ldots$.
If $t = 152845$, then $\sqrt{t^2 - n} = 804 \in \mathbb{Z}$.
Thus $s = 804$. We find that $p = t + s = 153649$ and $q = t - s = 152041$.

4.3.3 Factoring n Given d

In this section, we show that cracking RSA is, in practice, at least as difficult as factoring n. We give a probabilistic algorithm that given a decryption key determines the factorization of n.

Suppose that we crack an RSA cryptosystem with modulus n and encryption key e by somehow finding an integer d such that

$$a^{ed} \equiv a \pmod{n}$$

for all a. Then $m = ed - 1$ satisfies $a^m \equiv 1 \pmod{n}$ for all a that are coprime to n. As we saw in Section 4.3.1, knowing $\varphi(n)$ leads directly to a factorization of n. Unfortunately, knowing d does not seem to lead easily to a factorization of n. However, there is a probabilistic procedure that, given an m such that $a^m \equiv 1 \pmod{n}$, will find a factorization of n with high probability.

Algorithm 4.3.3 (Probabilistic Algorithm to Factor n Given d). In the description of this algorithm, a always denotes an integer coprime to n. Given an integer $m > 1$ such that $a^m \equiv 1 \pmod{n}$ for all a, this probabilistic algorithm factors n with high probability.

1. *If* $a^{m/2} \equiv 1 \pmod{n}$ *for all* a*, replace* m *by* $m/2$. Note that m is even since $(-1)^m \equiv 1 \pmod{n}$. It is not practical to determine whether or not $a^{m/2} \equiv 1 \pmod{n}$ for all a, because it would require doing a computation for too many a. Instead, we try a few random a; if $a^{m/2} \equiv 1 \pmod{n}$ for the a we check, we divide m by 2.

   Note that if there exists even a single a such that $a^{m/2} \not\equiv 1 \pmod{n}$, then at least half the a have this property, since $a \mapsto a^{m/2}$ is a nontrivial homomorphism $(\mathbb{Z}/n)^\times \to \{\pm 1\}$ and the kernel can have size at most $\phi(n)/2 = \#(\mathbb{Z}/n)^\times/2$.

   Keep replacing m by $m/2$ until we find an a such that $a^{m/2} \not\equiv 1 \pmod{n}$.

2. *Try to factor* n *by computing gcd's.* Assume that we have found an m such that $a^m \equiv 1 \pmod{n}$ for all a coprime to n, but there is an a such that $a^{m/2} \not\equiv 1 \pmod{n}$. (That $x^2 \equiv 1 \pmod{p}$ implies $x = \pm 1 \pmod{p}$ follows from Proposition 5.1.1 in the next chapter.) Since $(a^{m/2})^2 \equiv 1 \pmod{n}$, we also have $(a^{m/2})^2 \equiv 1 \pmod{p}$ and $(a^{m/2})^2 \equiv 1 \pmod{q}$, so $a^{m/2} \equiv \pm 1 \pmod{p}$ and $a^{m/2} \equiv \pm 1 \pmod{q}$. Since $a^{m/2} \not\equiv 1 \pmod{n}$, their are three possibilities for these signs, so with probability $2/3$,

   $$a^{m/2} \equiv +1 \pmod{p} \qquad \text{and} \qquad a^{m/2} \equiv -1 \pmod{q}$$

   or

   $$a^{m/2} \equiv -1 \pmod{p} \qquad \text{and} \qquad a^{m/2} \equiv +1 \pmod{q}.$$

   (The only other possibility is that both signs are -1.) In the first case,

   $$p \mid a^{m/2} - 1 \qquad \text{but} \qquad q \nmid a^{m/2} - 1,$$

   so $\gcd(a^{m/2} - 1, pq) = p$, and we have factored n. Similarly, in the second case, $\gcd(a^{m/2} - 1, pq) = q$, and we again factor n.

   Keep trying a's until one of these two cases occurs.

Example 4.3.4. Somehow we discover that the RSA cryptosystem with

$$n = 32295194023343 \qquad \text{and} \qquad e = 29468811804857$$

has decryption key $d = 11127763319273$. Let's use this information to factor n. We have

$$m = ed - 1 = 327921963064646896263108960.$$

For each $a \le 20$ we find that $a^{m/2} \equiv 1 \pmod{n}$, so we replace m by

$$\frac{m}{2} = 163960981532323448131554480.$$

Again, we find with this new m that for each $a \le 20$, $a^{m/2} \equiv 1 \pmod{n}$, so we replace m by $81980490766161724065777240$. Yet again, for each $a \le 20$,