CS 70 Discrete Mathematics and Probability Theory
Fall 2024 Rao, Hug

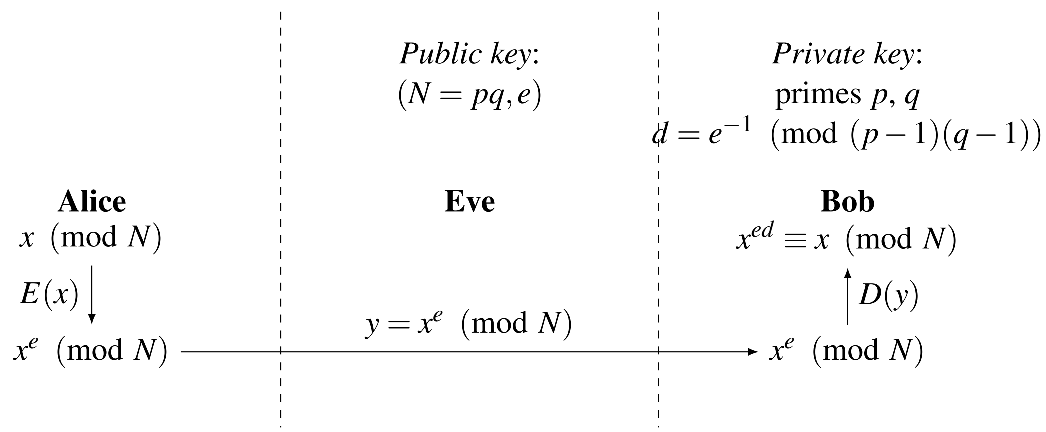
DIS 4B

1 RSA Intro

Note 7

Fermat's Little Theorem: For all primes p, $a^{p-1} \equiv 1 \pmod{p}$ if $a \neq 0$, and $a^p \equiv a \pmod{p}$ for all a.

RSA Scheme:



(a) Evaluate $314159^{30} \pmod{31}$.

(b) Suppose we would like to evaluate $141^{161} \pmod{187}$. If we think about this as an instance of RSA, what are p, q, e, and d? What is the final result of the computation? (*Hint: We know that* $161 = 7 \times 23$ *and* $187 = 11 \times 17$.)

(c) In RSA, we rely on the hardness of two different problems in order to guarantee the security of the scheme. Which two problems are these? If their hardness is not guaranteed, what goes wrong?

Solution:

(a) Since 31 is prime, we know that $a^{30} \equiv 1 \pmod{31}$ for any nonzero a, by FLT. In particular, we have

$$314159^{30} \equiv 1 \pmod{31}.$$

(b) This is an instantiation of the RSA scheme, with $p = 11$, $q = 17$, $e = 7$, and $d = 23$. (p and q could be swapped here, and similarly with e and d.)

In particular, Alice is attempting to send the message $x = 141$, and the encryption is $x^e \pmod{pq}$, or in this case $141^7 \pmod{187}$.

When Bob decrypts the message, he computes $x^{ed} \pmod{pq}$, or in this case $141^{7\times 23} = 141^{161} \pmod{187}$.

Since we know that this scheme will always recover the original message, the resulting quantity must be $141 \pmod{187}$.

(c) RSA relies on the hardness of the following two problems:

1. **Factorizing large numbers**: We assume that it is very hard to find the prime factorization of very large numbers, especially when they are *semiprimes* (i.e. they have exactly two prime factors).
2. **Finding the discrete logarithm**: We assume that is is also very hard to find the *discrete logarithm* of a number. That is, when working under mod N, we assume that it is hard to find $x \pmod{N}$, given e and $x^e \pmod{N}$.

If it is actually easy to factorize large numbers, any attacker can find p and q from $N = pq$ in the public key. With this information, the attacker can compute $d = e^{-1} \pmod{(p-1)(q-1)}$, since they now know all the quantities on the RHS. At this point, the attacker now knows *all* of the private key information, so the security of the scheme is compromised.

If it is actually easy to find the discrete logarithm of a number, then any attacker can compute x when given $y = x^e \pmod{N}$, since N and e are part of the public key. This means that the attacker now gains knowledge of the original message we were trying to hide, so the security of the scheme is compromised.

As a note here, we don't *actually* know whether these problems are hard to solve—however, we have not yet found any polynomial-time algorithms for either of these problems, despite many decades of mathematicians and computer scientists attempting to find them. This leads us to *suspect* that the problems are indeed hard to solve, which is why so much of cryptography is built around these concepts.

2 RSA Warm-Up

Note 7

Consider an RSA scheme with modulus $N = pq$, where p and q are distinct prime numbers larger than 3.

(a) What is wrong with using the exponent $e = 2$ in an RSA public key?

(b) Now suppose that $p = 5$, $q = 17$, and $e = 3$. What is the public key?

(c) What is the private key?

(d) Alice wants to send a message $x = 10$ to Bob. What is the encrypted message $E(x)$ she sends using the public key?

(e) Suppose Bob receives the message $y = 19$ from Alice. What equation would he use to decrypt the message? What is the decrypted message?

Solution:

(a) To find the private key d from the public key (N,e), we need $\gcd(e,(p-1)(q-1)) = 1$. However, $(p-1)(q-1)$ is necessarily even since p,q are distinct odd primes, so if $e=2$, $\gcd(e,(p-1)(q-1)) = 2$, and a private key does not exist. (Note that this shows that e should more generally never be even.)

(b) $N = p \cdot q = 85$ and $e = 3$ are displayed publicly. Note that in practice, p and q should be much larger (512-bit) numbers. We are only choosing small numbers here to allow manual computation.

(c) We must have $ed = 3d \equiv 1 \pmod{64}$, so $d = 43$. Reminder: we would do this by using extended gcd with $x = 64$ and $y = 3$. We get $\gcd(x,y) = 1 = ax + by$, and $a = 1$, $b = -21$.

(d) We have $E(x) = x^3 \pmod{85}$, where $E(x)$ is the encryption function. $10^3 \equiv 65 \pmod{85}$, so $E(x) = 65$.

(e) We have $D(y) = y^{43} \pmod{85}$, where $D(y)$ is the decryption function, the inverse of $E(x)$.

$$x \equiv 19^{43} \pmod{85}$$

From CRT we know that for coprime numbers p and q if

$$x \equiv a \pmod{p}$$
$$x \equiv b \pmod{q}$$

then

$$x = aqq_1 + bpp_1 \pmod{pq}$$

where $p_1 = p^{-1} \pmod{q}$ and $q_1 = q^{-1} \pmod{p}$.

In our case we have $p = 5$ and $q = 17$. So

$$x \equiv 19^{43} \equiv (-1)^{43} \equiv -1 \equiv 4 \pmod{5}$$

and

$$\begin{aligned} x &\equiv 19^{43} \pmod{17} \\ x &\equiv (2)^{43} \pmod{17} \\ x &\equiv (2^4)^{10} \cdot 2^3 \pmod{17} \\ x &\equiv (-1)^{10} \cdot 8 \pmod{17} \\ x &\equiv 8 \pmod{17} \end{aligned}$$

Hence

$$x = a = 4 \pmod{5} \qquad x = b = 8 \pmod{17}$$

and

$$p_1 = p^{-1} \pmod{17} = 5^{-1} \pmod{17} = 7$$
$$q_1 = q^{-1} \pmod{5} = 17^{-1} \pmod{5} = 3$$

So we have

$$x \equiv aqq_1 + bpp_1 \pmod{pq}$$
$$x \equiv 4 \cdot 17 \cdot 3 + 8 \cdot 5 \cdot 7 \pmod{85}$$
$$x \equiv 4 \cdot 17 \cdot 3 + 280 \pmod{85}$$
$$x \equiv 17 \cdot (12) + 280 \pmod{85}$$
$$x \equiv 17 \cdot (10 + 2) + 280 \pmod{85}$$
$$x \equiv 34 + 25 \pmod{85}$$
$$x \equiv 59 \pmod{85}$$

so $D(y) = 59$.

3 RSA with Multiple Keys

Note 7

Members of a secret society know a secret word. They transmit this secret word x between each other many times, each time encrypting it with the RSA method. Eve, who is listening to all of their communications, notices that in all of the public keys they use, the exponent e is the same. Therefore the public keys used look like $(N_1, e), \ldots, (N_k, e)$ where no two N_i's are the same. Assume that the message is x such that $0 \le x < N_i$ for every i.

Further, in all of the subparts, you may assume that Eve knows the details of the modified RSA schemes (i.e. Eve knows the format of the N_i's, but not the specific values used to compute the N_i's).

(a) Suppose Eve sees the public keys $(p_1q_1, 7)$ and $(p_1q_2, 7)$ as well as the corresponding transmissions. Can Eve use this knowledge to break the encryption? If so, how? Assume that Eve cannot compute prime factors efficiently. Think of p_1, q_1, q_2 as massive 1024-bit numbers. Assume p_1, q_1, q_2 are all distinct and are valid primes for RSA to be carried out.

(b) The secret society has wised up to Eve and changed their choices of N, in addition to changing their word x. Now, Eve sees keys $(p_1q_1, 3)$, $(p_2q_2, 3)$, and $(p_3q_3, 3)$ along with their transmissions. Argue why Eve cannot break the encryption in the same way as above. Assume $p_1, p_2, p_3, q_1, q_2, q_3$ are all distinct and are valid primes for RSA to be carried out.

(c) Let's say the secret x was not changed ($e = 3$), so they used the same public keys as before, but did not transmit different messages. How can Eve figure out x?

Solution:

(a) Normally, the difficulty of cracking RSA hinges upon the believed difficulty of factoring large numbers. If Eve were given just p_1q_1, she would (probably) not be able to figure out the factors.

However, Eve has access to two public keys, so yes, she will be able to figure it out. Note that $\gcd(p_1q_1, p_1q_2) = p_1$. Taking GCDs is actually an efficient operation thanks to the Euclidean Algorithm. Therefore, she can figure out the value of p_1, and from there figure out the value of q_1 and q_2 since she has p_1q_1 and p_1q_2.

(b) Since none of the N's have common factors, she cannot find a GCD to divide out of any of the Ns. Hence the approach above does not work.

(c) Eve observes $x^3 \pmod{N_1}$, $x^3 \pmod{N_2}$, $x^3 \pmod{N_3}$. Since all N_1, N_2, N_3 are pairwise relatively prime, Eve can use the Chinese Remainder Theorem to figure out $x^3 \pmod{N_1N_2N_3}$. However, once she gets that, she knows x, since $x < N_1$, $x < N_2$, and $x < N_3$, which implies $x^3 < N_1N_2N_3$, so she can directly take the cube root of the result from CRT. Uh oh!

4 RSA for Concert Tickets

Note 7

Alice wants to tell Bob her concert ticket number, m, which is an integer between 0 and 100 inclusive. She wants to tell Bob over an insecure channel that Eve can listen in on, but Alice does not want Eve to know her ticket number.

(a) Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's ticket number is. How did she do it?

(b) Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m? (You may assume that r is coprime to N.)

Solution:

(a) There are only 101 possible values for Alice's ticket number, so Eve can try encrypting all 101 values with Bob's public key and find out which one matches the one Alice sent.

(b) Alice sends $x = r^e \pmod{pq}$, as well as $y = (rm)^e = r^e m^e = xm^e \pmod{pq}$. We can find $x^{-1} \pmod{N}$ using the Extended Euclidean Algorithm, and multiplying this value by y gives us $m^e \pmod{N}$. Now we proceed as in the previous part to find m.

Another approach is to compute xm^e for all 101 values of m, and compare the value to y, checking which one matches.