RSA lecture summary

M. George

November 5, 2014

1 Public key cryptography

Private key, secret key, or shared key cryptography involve a shared secret between the sender and the recipient. This requires coordination before hand, and is thus not feasible in many ciircumstances.

Public key cryptography works differently: the *recipient* generates a *public/private key pair* and publishes the public key. The keys are mathematically related in such a way that anyone with the public key can encrypt a message that requires the private key to decrypt.

2 Almost RSA

We worked through a simple version of RSA that rests on Fermat's Little Theorem. This algorithm correctly transmits the information, but makes it easy to derive the private key from the public key, and thus doesn't protect the message. We will then generalize the simple version, but it will require more number theory.

The almost algorithm The recipient generates a large random prime number p, and a number k between 1 and p. p and k together form the public key.

The recipient also computes an *inverse of* k *modulo* $(p-1)$. This is a number k^{-1} such that $kk^{-1} \equiv 1 \mod (p-1)$. You will prove on the homework that k^{-1} exists (as long as k and p are relatively prime) and is easy to compute and is unique. This is the *private key.*

The *sender* first converts the *plaintext* message to a number a. If $a \geq p$ then the sender will break up a into multiple independent messages. She will send the *ciphertext* $a^k \mod p$ to the recipient.

Upon receiving the message, the recipient will raise a^k to the k' power. By Fermat's little theorem, we have:

$$
\begin{aligned}
(a^k)^{k^{-1}} &= a^{kk^{-1}} \\
&= a^{1+x(p-1)} && \text{since } kk^{-1} \equiv 1 \mod (p-1) \\
&= a \cdot (a^{(p-1)})^x && \text{simplification} \\
&= a \cdot 1^x && \text{Fermat's little theorem} \\
&= a
\end{aligned}
$$

3 Generalizing to get the actual RSA algorithm

The above algorithm does transmit a to the recipient. Unfortunately, the public key is enough to compute the private key, so anyone can decrypt the message.

To generalize this, we will choose a modulus m that is a product of two primes, instead of a single prime. We assume that multiplication is easy but factoring is hard.

For the algorithm to work, we need a number that I will call $\phi(m)$ that has the same property that Fermat's little theorem gives us: that

$$a^{\phi(m)} = 1 \mod m$$

Computing this number is the subject of future lectures; for now, I will assume it exists.

Now we can modify the above algorithm as follows:

The actual RSA algorithm Recipient chooses large primes p and q, and an exponent k. She publishes the product pq and k, but *not* p and q individually. She also uses p and q privately to compute $k^{-1} \mod \phi(pq)$.

The sender again creates the *plaintext* a, and computes $a^k \mod pq$. She transmits this to the recipient.

The recipient then computes $(a^k)^{k^{-1}}$. As before,

$$
\begin{aligned}
(a^k)^{k^{-1}} &= a^{kk^{-1}} \\
&= a^{1+x\phi(pq)} && \text{since } kk^{-1} \equiv 1 \mod \phi(pq) \\
&= a \cdot (a^{\phi(pq)})^x && \text{simplification} \\
&= a \cdot 1^x && \text{Property of } \phi \\
&= a
\end{aligned}
$$

The difference here is that the recipient can compute $\phi(pq)$ because she knows p and q, but a third party who knows only their product cannot.

4 What is ϕ?

To explain the definition of ϕ and how to compute it, we need a few definitions.

m and n are *relatively prime* if they have no common factors. This is true if and only if their greatest common divisor is 1.

In any set of elements with a reasonable definition of multiplication, we say that an element y is an *inverse* of x if $xy = 1$. An element is a *unit* if it has an inverse.

Examples:

- In the integers, 1 is its own inverse. -1 is also its own inverse. No other element has an inverse. Thus the set of units of the integers are 1 and -1.
- In the rationals, a/b has an inverse b/a as long as $a \neq 0$.
- In the set $\mathbb{Z}_3$ of integers modulo 5, we have $1 \cdot 1 = 1$, so 1 is its own inverse, we have $2 \cdot 3 = 6 = 1$, so 2 and 3 are inverses, and we have $4 \cdot 4 = 16 = 1$ so 4 is its own inverse. Thus all the elements of $\mathbb{Z}_5$ are units, except 0.
- In the set $\mathbb{Z}_{10}$ of integers modulo 10, the units are 1, 3, 7, and 9. The other elements are not units. For example, 6 cannot be a unit, because any number times 6 will always be even, and thus cannot be 1 mod 10.
- In general, x is a unit in $\mathbb{Z}_m$ if x and m are relatively prime. You will prove this on the homework.

We can now define $\phi(m)$: it is the number of units in the set $\mathbb{Z}_m$. We will prove that it has the desired property in the next lecture.