Queen Mary
University of London

MAS 335 **Cryptography**

Notes 9: Public-key cryptography **Spring 2008**

Diffie–Hellman key exchange

The functions used for the RSA cipher can also be used to implement the key-exchange protocol that we discussed at the very beginning of our discussion of public-key cryptography. This system of key exchange actually predates the RSA cipher.

Assume that Alice wants to send a secret message to Bob. Alice and Bob agree on a modulus p, a prime number. (They must share the prime p, so they must assume that Eve can get hold of it!) Each of them chooses a number coprime to $\lambda(p) = p-1$, and computes its inverse. These numbers are not revealed. Alice chooses d_A and e_A, Bob chooses d_B and e_B. Note that our commutation condition is satisfied:

$$T_{d_A}T_{d_B}(x) = x^{d_A d_B} \bmod p = T_{d_B}T_{d_A}(x).$$

In terms of our analogy, T_{e_A} is Alice putting on her padlock, while T_{d_A} is Alice removing her padlock.

Now Alice takes the message x and applies T_{e_A}; she sends $T_{e_A}(x)$ to Bob. Bob applies T_{e_B} and returns $T_{e_B}T_{e_A}(x)$ to Alice. Alice applies T_{d_A} and returns

$$T_{d_A}T_{e_B}T_{e_A}(x) = T_{d_A}T_{e_A}T_{e_B}(x) = T_{e_B}(x)$$

to Bob, who then applies T_{d_B} and recovers $T_{d_B}T_{e_B}(x) = x$, the original message.

Nobody has yet discovered a weakness in this protocol like the weakness we found using one-time pads. In other words, even if Eve intercepts all the messages $T_{e_A}(x)$, $T_{e_B}T_{e_A}(x)$ and $T_{e_B}(x)$ that pass to and fro between Alice and Bob, there is no known easy algorithm for her to discover x (even given the modulus p).

Contrast this with the standard RSA protocol: First, it allows a pair of users to communicate securely, whereas RSA allows any two users in a pool to communicate; secondly, three messages have to be sent, rather than just one; thirdly, what is secret and what is public are different in this case (the prime is public but the exponent is secret).

The security of this protocol depends on the fact that, if $y = x^e \pmod{p}$, then knowledge of x and y does not allow an easy calculation of e. For suppose that Eve could solve this problem. Recall that Eve knows x^{e_A}, x^{e_B} and $x^{e_A e_B}$ (the three messages exchanged during the protocol). If she could use x^{e_A} and $x^{e_A e_B}$ to discover e_B, she could find its inverse d_B modulo $p-1$ and then calculate $(x^{e_B})^{d_B} \bmod p = x$, the secret message.

Thus, the security of this method depends on the fact that the following problem is hard:

> Given x, y, and a prime p such that $y \equiv x^e \bmod p$, find e.

This is known as the *discrete logarithm problem*, since in a sense e is the logarithm of y to base x (where our calculations are in the integers mod p, rather than in the real numbers as usual). This problem is believed to be at least as difficult as factorisation, although (like factorisation) it is not known to be NP-complete.

If it happens that the order of x mod p is small (so that there are only a few distinct powers of x mod p), then e can be found by exhaustive search. So, to make the problem hard, the order of x should be as large as possible. Ideally, choose x to be a primitive root mod p (an element of order $\lambda(p) = p-1$).

Example Suppose that $p = 30491$ and $x = 13$. Then $x^2 = 169$, $x^3 = 2197$, $x^4 = 28561$, and $x^5 \equiv 1 \pmod{p}$. So the discrete logarithm problem is easily solved. On the other hand, 2 is a primitive root mod 30491, so all the powers $2^0, 2^1, 2^2, \ldots, 2^{30489}$ are distinct, and finding which one is a particular element y will be very laborious.

How do we check that 2 is a primitive root mod 30491, without actually working out all these powers? We know that $2^{30490} \equiv 1 \pmod{30491}$, by Fermat's Little Theorem. So the order of 2 must be a divisor of 30490. We factorise 30490 into prime factors: $30490 = 2 \cdot 5 \cdot 3049$. So any *proper* divisor would have to divide the product of two of these primes. So we check that none of $2^{2\cdot5}$, $2^{2\cdot3049}$ and $2^{5\cdot3049}$ is congruent to 1 mod 30491. So in this case we only have to check three powers of 2; but it is necessary to factorise $p-1$.

Diffie–Hellman key establishment is a slight variation on the above protocol. In this case, Alice and Bob agree on a prime p and a primitive root g modulo p. These are regarded as public. Alice picks a random exponent a (for best results she should choose a with $\gcd(a, p-1) = 1$), and Bob picks a random exponent b (also with $\gcd(b, p-1) = 1$). Now Alice sends $g^a \bmod p$ to Bob, and Bob sends $g^b \bmod p$ to Alice. (Notice that it doesn't matter who sends their message first.) Then Alice calculates $(g^b)^a \bmod p$ using her secret number a and the number $g^b \bmod p$ sent by Bob. Similarly, Bob calculates $(g^a)^b \bmod p$. Therefore they can use $k = g^{ab} = (g^a)^b = (g^b)^a \bmod p$ as their secret key.

El-Gamal

The El-Gamal cryptosystem is a rival to RSA and is widely used. Its security is based on the difficulty of the discrete logarithm. It works as follows.

Bob chooses a prime number p and a primitive root g mod p. (Remember that this is an element such that the powers $g^0, g^1, \ldots, g^{p-2}$ are all distinct modulo p, and include all the non-zero congruence classes mod p. We saw in Theorem 21 that primitive roots exist for any prime p.) He also chooses an integer $a \in \{1, \ldots, p-2\}$, and computes $h = g^a \pmod{p}$. His public key is (p, g, h); the number a is kept secret.

Alice wants to send a plaintext x to Bob, encoded as an integer in the range $\{1, \ldots, p-1\}$. She chooses a random number k, also in this range, and computes $y_1 = g^k \pmod{p}$ and $y_2 = xh^k \pmod{p}$. The ciphertext is the pair (y_1, y_2).

Note that

- the ciphertext is twice as long as the plaintext;
- there are $p-1$ different ciphertexts for each plaintext, one for each choice of the random number k.

Bob receives the message (g^k, xh^k) mod p. He knows the number a such that $h = g^a$ mod p; so he can compute

$$h^k \equiv (g^a)^k \equiv (g^k)^a \bmod p$$

without knowing Alice's secret number k. Now he can find x by "dividing" $y_2 = xh^k$ by h^k; more precisely, he uses Euclid's algorithm to find the inverse of h^k mod p and multiplies y_2 by this inverse to get the plaintext x.

Eve, intercepting the message, is faced with the problem of finding either

- the number a for which $h \equiv g^a \pmod{p}$, so that she can then use the same decrypting method as Bob; or
- the number k for which $y_1 \equiv g^k \pmod{p}$, so that she can find h^k directly and hence find x.

Either approach requires her to solve the Discrete Logarithm problem, and so may be assumed to be difficult. No better way of trying to break the cipher is known.

Note that, if Eve does have the computational resources to solve a discrete logarithm problem, she should employ them on the first of the above problems. For if this is solved, then she knows Bob's private key and can read all his mail. Solving the second only gives her Alice's random number k, which will be different for each message, so the same job would have to be done many times.

Here is a brief example. Suppose that Bob chooses the prime $p = 83$, the primitive root $g = 2$, and the number $a = 30$, so that $h = 2^{30} \bmod 83 = 40$. Bob's public key is $(83, 2, 40)$. Suppose that Alice's plaintext is $x = 54$ and her random number is $k = 13$. Then Alice's ciphertext is

$$(g^k, xh^k) \bmod p = (58, 71).$$

Bob computes $58^{30} \bmod 83 = 9$. By Euclid's algorithm, the inverse of 9 mod 83 is 37; and so the plaintext is $37 \cdot 71 \bmod 83 = 54$.

El-Gamal signatures

Using the El-Gamal scheme for digital signatures is a bit more complicated than using, say, RSA. This is because, as we saw, the ciphertext in El-Gamal is twice as long as the plaintext, and depends on the choice of a random number k. So, to sign a message, Alice cannot simply pretend that the message is a cipher and decrypt it with her private key! Instead, she adds further data whose purpose is to authenticate the message.

Suppose that Alice's El-Gamal public key is (p, g, h), where p is prime and g is a primitive root mod p. Then $h \equiv g^a \pmod{p}$, where the number a is known only to Alice.

To sign a message $x \in \{1, \ldots, p-1\}$, Alice chooses a random number k satisfying $\gcd(k, p-1) = 1$. Then using Euclid's algorithm, she computes the inverse l of k mod $p-1$. Now she computes

$$\begin{aligned} z_1 &= g^k \bmod p, \\ z_2 &= (x - az_1)l \bmod p-1 \end{aligned}$$

The signed message is (x, z_1, z_2). Just as in the case of encryption, note that it is longer than (in this case, three times as long as) the unsigned message, and it depends on a random number k. Alice then encrypts this message with Bob's public key and sends it to Bob.

On receipt, Bob decrypts the message, and finds three components. The first component is the plaintext x. The second and third components comprise the signature. Bob accepts the signature as valid if

$$h^{z_1} z_1^{z_2} \equiv g^x \pmod{p}.$$

We have to show that

- if Alice follows the protocol correctly, this condition will be satisfied;

- Eve cannot forge the signature (i.e. produce (x, z_1, z_2) satisfying this condition) without solving a discrete logarithm problem.

The first condition is just a case of checking;

$$h^{z_1} z_1^{z_2} \equiv g^{az_1} g^{kl(x-az_1)} \pmod{p}.$$

Note that $g^{p-1} \equiv 1 \pmod{p}$, so exponents of g can be read modulo $p-1$. Now $kl \equiv 1 \pmod{p-1}$, so $g^{kl(x-az_1)} \equiv g^{x-az_1} \pmod{p}$. Then

$$h^{z_1} z_1^{z_2} \equiv g^{az_1} g^{x-az_1} \equiv g^x \pmod{p}.$$

The second part is a bit more complicated and the argument will not be given here. It is clear that Eve cannot do Alice's computation without knowing a. We have to be sure that there is no other way that she could produce a forgery.

Example Suppose that Alice's public key is $(107, 2, 15)$, with secret number 11, so that 2 is a primitive root mod 107, and $2^{11} \equiv 15 \pmod{107}$. Suppose that Alice wants to send the message 10 to Bob and sign it. She chooses $k = 17$; this number is coprime to 106, and its inverse is 25. The signature is (z_1, z_2), where

$$\begin{aligned} z_1 &= 2^{17} \bmod 107 = 104, \\ z_2 &= (10 - 11 \cdot 104) \cdot 25 \bmod 106 = 58. \end{aligned}$$

So she encrypts the plaintext $(10, 104, 58)$ with Bob's public key and sends it to Bob. (Note that the one number x has now become six numbers in the ciphertext!)

Bob, having decrypted the message, obtains $(10, 104, 58)$. He tests whether

$$15^{104} \cdot 104^{58} \equiv 2^{10} \pmod{107},$$

and, since this is the case, he is assured that the message is from Alice.

Finding primitive roots

The El-Gamal system requires each user to choose a prime p and a primitive root g mod p. How does he find a primitive root? This is a problem which is itself not easy. There are two approaches that have been used.

One approach is to observe that it is not crucial for the operation of the method that g is a primitive root; all we require is that g should have many different powers mod p, so that the discrete logarithm cannot be solved by exhaustive search. So all that Bob has to do is to choose a number g and check that $g^i \not\equiv 1 \pmod{p}$ for all not-too-large i. (If he can factorise $p-1$, he can test whether g is a primitive root in only a few steps by the method of the earlier example; if it is not a primitive root, he can find out what its order actually is by continuing this analysis.)

Another is to observe that there are some special primes for which it is easy to find a primitive root. One way to do this is as follows.

A pair (q,p) of prime numbers is called a *Sophie Germain pair* if $p = 2q+1$. These are so-called because, in 1825, Sophie Germain proved a special case of Fermat's Last Theorem for exponents which are the smaller of a Sophie Germain pair. The important thing is that it appears (though it is not proved yet) that there are lots of such prime pairs. So it is not too inefficient to find a prime q, and then test whether $p = 2q+1$ is also prime.

Now we have the following result.

Proposition 24 *let (q,p) be a Sophie Germain pair. Suppose that $1 < x < p-2$. Then x is a primitive root mod p if and only if $x^q \equiv -1 \pmod{p}$.*

For the order of x mod p divides $p-1 = 2q$ by Fermat's Little Theorem, and is not 1 or 2 (since the only elements with these orders are 1 and $p-1$); so the order is q or $2q$.

Suppose that x is a primitive root (of order $2q$), and let $y = x^q \bmod p$. Then $y^2 \equiv 1 \pmod{p}$, but $y \not\equiv 1 \pmod{p}$; so $x^q \equiv y \equiv -1 \pmod{p}$.

Conversely, suppose that x is not a primitive root; then x has order q, so $x^q \equiv 1 \pmod{p}$.

In our earlier example, $(41,83)$ is a Sophie Germain pair, so to test whether 2 is a primitive root mod 83, we only have to decide whether $2^{41} \equiv -1 \pmod{83}$. This can be done directly, but the calculation can be simplified still further using tools from Number Theory (the so-called Quadratic Reciprocity Law of Gauss). This is beyond the scope of this course, but is discussed in the Number Theory course.

Sophie Germain was the first female mathematician in western Europe. She faced many difficulties in being accepted as a serious mathematician. She communicated by letter with many of the famous mathematicians of the time, such as Gauss and Lagrange, signing her name "Monsieur LeBlanc". Gauss learned that his correspondent was a woman in a curious way.

He lived in Braunschweig in eastern Germany. When Napoleon's armies invaded in 1806, Germain asked the military commander, who was a family friend, to take special care of Gauss. (As a child, she had read the story of how Archimedes had been killed by a Roman soldier during the invasion of Syracuse, and dreaded that Gauss would suffer the same fate.) On asking to whom this special attention was due, Gauss was surprised to learn that "Monsieur LeBlanc" was a woman.