Lecture 21

Diffie-Hellman key exchange, RSA and public key crypto

Diffie-Hellman key exchange

One problem with symmetric systems is the need for Alice and Bob to share a secret key. How exactly should this be accomplished? In 1976, Diffie and Hellman (two people, not one) showed how this could be done heralding a new era in cryptography.

Alice and Bob communicate in the clear over their communication channel. They agree on a very large prime number p. From now on, all their calculations will be in the field $\mathbb{Z}_p$ or more accurately the cyclic group U_p. Next Alice and Bob agree on a generator g of U_p. Thus $\langle g \rangle = U_p$.

Now Alice chooses a secret number a ($1 \leq a \leq p-1$) and Bob chooses a secret number b.

Alice computes g^a and sends it to Bob.

Bob computes g^b and sends it to Alice.

Eve, who's been listening in, knows p, g, g^a and g^b.

Now Comes the Clever bit.

- Alice has received g^b from Bob and computes $(g^b)^a = g^{ab}$.
- Bob has received g^a from Alice and computes $(g^a)^b = g^{ab}$.

> Thus both Alice and Bob are now in receipt of the shared information g^{ab}.

You might think that because Eve knows g^a and g^b she could compute a and b and so g^{ab}. The question is how? She knows g so she can only find a by calculating g^x for $x = 1, 2, 3, \ldots$ until she hits upon g^a

This is believed to be an intractable problem.

Alice and Bob can now use the shared information g^{ab} (which Eve doesn't share) to work out a secret key between them and to set up a symmetric cryptosystem between them.

You now have the information to read Diffie and Hellman's original paper bearing in mind the following points:

- The field $\mathbb{Z}_p$ is denoted $GF(p)$ ("Galois field of order p") and a generator is called a <u>primitive element</u>.

- The problem of finding a given g^a is called the <u>discrete logarithm problem</u>.

This is by analogy with <u>ordinary arithmetic</u> in $\mathbb{R}$: if $x = 10^y$ then $y = \log_{10}(x)$.

RSA Diffie-Hellman is not actually a cryptosystem - it is a solution to the problem of key-exchange. RSA (Rivest, Shamir, Adleman) is a cryptosystem but quite different from symmetric ones in that the key is split in two — with one being made public and only the other being kept secret. It was published in 1978.

Details of their system now follow.

The basis of this system is a pair of large distinct primes p and q which are generated randomly and independently.

The product $n = pq$ is computed.

We shall be working in the multiplicative monoid $\mathbb{Z}_n$.

Choose an integer e such that

$$1 < e < \varphi(n) \quad \text{and} \quad \gcd(e, \varphi(n)) = 1.$$

The second condition implies by Bézout's theorem that there are integers $x, y \in \mathbb{Z}$ such that

$$ex + \varphi(n)\, y = 1.$$

Thus we may find a natural number d — which is either x or its reduction modulo $\varphi(n)$ — such that

$$1 < d < \varphi(n) \quad \text{and} \quad de \equiv 1 \pmod{\varphi(n)}$$

We can now set up our cryptosystem.

The public key is (n, e)

The private key is (n, d) i.e. essentially d

n:	RSA modulus
e:	encryption exponent
d:	decryption exponent

Before we show how this information is used to construct ciphers, we do a numerical example.

Example

$p = 11$ and $q = 23$

$\therefore n = pq = 11 \times 23 = 253$

$(p-1)(q-1) = 10 \times 22 = 220$

We need to choose $1 < e < 220$ where $\gcd(220, e) = 1$.

The smallest possible is $e = 3$.

We now compute d using Blankinship's algorithm

$$\left(\begin{array}{cc|c} 1 & 0 & 220 \\ 0 & 1 & 3 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} 1 & -73 & 1 \\ 0 & 1 & 3 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} 1 & -73 & 1 \\ -3 & 220 & 0 \end{array}\right)$$

$$220 \times 1 + (-73) \times 3 = 1.$$

$\therefore \quad (-73) \times 3 \equiv 1 \pmod{220}.$

Thus $(220-73) \times 3 \equiv 1$

and so $147 \times 3 \equiv 1 \pmod{220}$

Thus we put $d = 147$.

253	:	RSA modulus
3	:	encryption exponent
147	:	decryption exponent

We shall now show how to use (n, e, d) to encrypt and decrypt data.

To begin with, our plaintext space will be $\mathbb{Z}_n$.

Let $0 \leq M < n$, where M is our cleartext.

We encrypt M by putting

$$C = M^e \bmod n$$

Example With the parameters as above

$$M = 165, \quad C = 165^3 \bmod 253$$
$$= 110$$

I claim that $M = c^d$. Let's see this in our special case. We need to compute

$$110^{147} \bmod 253.$$

To calculate this, we shall use repeated squaring

$$147 = 128 + 16 + 2 + 1$$

Thus $110^{147} = 110^{128}\ 110^{16}\ 110^{2}\ 110^{1}$

	mod 253
110	110 *
110^2	209 *
110^4	165
110^8	154
110^{16}	187 *
110^{32}	55
110^{64}	242
110^{128}	121 *

$$\therefore 110^{147} = 121 \cdot 187 \cdot 209 \cdot 110$$

$$\equiv 165 = M$$ ✓

Let's now prove that this always works.

Theorem Let (n, e, d) be the parameters of an RSA Cryptosystem. Then for any $m \in \mathbb{Z}_n$, we have that

$$(m^e)^d = m.$$

Proof By assumption, $ed \equiv 1 \mod (p-1)(q-1)$.

Thus $ed = 1 + \ell(p-1)(q-1)$ for some ℓ.

Now

$$(m^e)^d = m^{ed} = m^{1+\ell(p-1)(q-1)} = m\, m^{\ell(p-1)(q-1)}.$$

We have that $m^{\ell(p-1)(q-1)} = (m^{p-1})^{\ell(q-1)}$. If $p \nmid m$ then $m^{p-1} \equiv 1 \mod p$ by Fermat's Little Theorem.

Similarly, if $q \nmid m$ then $m^{q-1} \equiv 1 \mod q$.

Thus if $p, q \nmid m$ we have that

$$(m^e)^d \equiv m \mod p$$

$$(m^e)^d \equiv m \mod q$$

But p and q are coprime and so $(m^e)^d \equiv m \mod pq = n$, as required. We still have a couple of cases to deal with.

- Suppose $p \nmid m$ but $q \mid m$. Then

$$(m^e)^d \equiv m \mod p \quad \text{and} \quad (m^e)^d \equiv 0 \equiv m \mod q$$

- Suppose $p \mid m$ but $q \nmid m$. Then

$$(m^e)^d \equiv 0 \equiv m \mod p \qquad (m^e)^d \equiv m \mod q.$$

Thus in all cases the result holds. ∎