

◇ Security in computer networks ◇

Everything we have seen so far in CSM14 involves *symmetric-key cryptography*.

Linking machines and using symmetric-key cryptography is all very well, but:

1. Are they all to use the same key (increasing the security risk) or different keys (making key management more difficult)?
2. How are they to agree on a key to use?

◇ Security risk and key management ◇

If all n machines use the same key:

1. A security breach at any of these n points will break the entire system. All communication will be wide open.

If they use different keys, there are two possibilities:

1. Everything is modelled as client/server. This may be acceptable but often is not: there may be a bottleneck at the server.
2. Pairwise communication is allowed. In this case, each pair of machines will need a key. They will need $O(n^2)$ keys, which will quickly become awkward as n increases. Adding a machine is also a pain.

◇ Key distribution ◇

How is a new machine going to get to know the network key? Or, how is a new machine going to agree on n new keys with the other n machines?

Ideas:

- ◇ Post the key on a disk?
- ◇ Use the phone?
- ◇ Courier it? (US's COMSEC)
- ◇ Meet in person to agree a key?
- ◇ Encrypt the key and send it over the network?

The system is only as secure as its weakest link.

◇ Solving the problem ◇

The problem was believed impossible for a long time.

Eventually a solution was published in the 1970s by Whitfield Diffie, Martin Hellman, and Ralph Merkle.

But it seems that it was discovered a short while before this by Malcolm Williamson at GCHQ, and kept classified.

◇ A thought experiment ◇

Alice wants to send a message to Bob, but is worried that the postman will intercept it and read it.

She can put it in a box and padlock it, and send it to Bob. But Bob doesn't have the key...

She could put the key in another box and padlock that, but...

Suppose that Alice has a padlock and corresponding key, and Bob has a different padlock and corresponding key.

Can you think of a way to get the secret message from Alice to Bob?

◇ The solution ◇

1. Alice puts her message in the box, and padlocks it. She sends this to Bob.
2. Bob then puts his padlock on it as well, and sends it back to Alice.
3. Alice removes her padlock, and sends it to Bob again.
4. Bob removes his padlock and reads the message.

What assumptions have we made here?

◇ Commutativity of encryption ◇

The padlock approach does not translate immediately to the world of cryptography. We are looking for

1. Encryption under K_A
2. Encryption under K_B
3. Decryption with K_A
4. Decryption with K_B

to get us back to our starting point. Usually, however, it doesn't.

We would need a *commutative* encryption algorithm: one in which

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

◇ Non-commutativity of a simple cipher ◇

Even simple monoalphabetic substitution doesn't work. (A Caesar cipher does—why?)

Alice's encryption key

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
H	F	S	U	G	T	A	K	V	D	E	O	Y	J	B	P	N	X	W	C	Q	R	I	M	Z	L

Bob's encryption key

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
C	P	M	G	A	T	N	O	J	E	F	W	I	Q	B	U	R	Y	H	X	S	D	Z	K	L	V

Message	m	e	e	t	m	e	a	t	n	o	o	n
Encrypt with Alice's key	Y	G	G	C	Y	G	H	C	J	B	B	J
Encrypt with Bob's key	L	N	N	M	L	N	O	M	E	P	P	E
Decrypt with Alice's key	Z	Q	Q	X	Z	Q	L	X	K	P	P	K
Decrypt with Bob's key	w	n	n	t	w	n	y	t	x	b	b	x

◇ A second thought experiment ◇

Alice, Bob and Eve (the baddie) each have a pot containing one litre of yellow paint. Alice and Bob are far apart, but can send pots of paint to each other.

Alice and Bob want to agree on a secret 'key'—in this case, a secret colour.

They each have access to unlimited quantities of paint in whatever colours they like. But how can Alice and Bob end up having a pot each of the same colour, without Eve being able to duplicate the colour?

◇ The solution ◇

1. Alice chooses a colour (C_A) at random. She pours a litre of it into her pot of yellow (Y), mixes it thoroughly, and sends it to Bob.
2. Meanwhile, Bob also chooses a random colour (C_B), pours a litre of it into his own pot of yellow, mixes it, and sends it to Alice.
3. When Alice receives Bob's pot ($Y + C_B$), she pours a litre of her own chosen colour into it, and mixes it. This gives her three litres of 'secret' paint ($Y + C_B + C_A$).
4. Meanwhile, when Bob gets Alice's pot ($Y + C_A$), he mixes in a litre of his own colour. This gives him three litres of secret paint ($Y + C_A + C_B$) too.

Even if Eve intercepts the pots, she cannot learn their key. She can get hold of what Alice sends ($Y + C_A$) and what Bob sends ($Y + C_B$), but she has no way of removing the yellow from either pot, and no way of combining the two to get the secret colour. (If she mixes them, she will end up with $2Y + C_A + C_B$, which is altogether too yellow.)

◇ One-way functions ◇

Diffie, Hellman and Merkle found a way to get a similar effect by computer.

The essential point of the paint analogy is that some things are easy to do but very hard to reverse—mixing paint, baking a cake, smashing a plate, multiplying numbers...

1. What is 23×29 ?
2. What two numbers have been multiplied together to get 527?

◇ Modular arithmetic ◇

Usually, when we add two numbers up, we end up with something bigger than either of the numbers we started with, because we conceive of numbers as ‘going on for ever’.

There are situations, however, in which we go against this, and allow the numbers to ‘wrap round’: if we start something at 10 o’clock, and it takes 5 hours, it finishes at 3 o’clock, not 15 o’clock. Once we get past 12, we go back to 1 again.

Arithmetic that uses all of the numbers from 0 to $n - 1$, and wraps back round to 0 instead of reaching n , is known as *arithmetic modulo n* . Examples:

$$\diamond 13 + 19 \pmod{26} = 32 \pmod{26} = 6$$

$$\diamond 7 \times 11 \pmod{5} = 77 \pmod{5} = 2$$

$$\diamond 3^7 \pmod{11} = 2187 \pmod{11} = 9$$

◇ Modular exponentiation ◇

Two problems to try:

1. If $3^\alpha = 531441$, what is α ?
2. If $3^\beta = 6 \pmod{17}$, what is β ?

What makes it much easier to search for α than to search for β ?

◇ Monotonicity ◇

When we try to solve for α , we can take advantage of the fact that if $f(x) = 3^x$ then f is *monotonic*—that is, the bigger the input, the bigger the output.

We have no such luck when working modulo 17.

Powers of 3								
x	0	1	2	3	4	5	6	7
3^x	1	3	9	27	81	243	729	2187
$3^x \pmod{17}$	1	3	9	10	13	5	15	11
x	8	9	10	11	12	13	14	15
3^x	6561	19683	59049	177147	531441	1594323	4782969	14348907
$3^x \pmod{17}$	16	14	8	7	4	12	2	6

The numbers are all over the place!

◇ A few mathematical laws ◇

The first few laws essentially say that if you are working modulo n , you can more or less put ‘(mod n)’ wherever you like and it won’t make any difference:

$$(a + b) \pmod{n} = ((a \pmod{n}) + b) \pmod{n} \quad (1)$$

$$(ab) \pmod{n} = ((a \pmod{n})b) \pmod{n} \quad (2)$$

$$a^x \pmod{n} = (a \pmod{n})^x \pmod{n} \quad (3)$$

$$(a^x \pmod{n})^y \pmod{n} = (a^x)^y \pmod{n} \quad (4)$$

Some laws about exponentiation. All of these hold with or without modular arithmetic:

$$(ab)^x = a^x b^x \quad (5)$$

$$a^{(x+y)} = a^x a^y \quad (6)$$

$$a^{xy} = (a^x)^y = (a^y)^x \quad (7)$$

The key equations are 4 and 7.