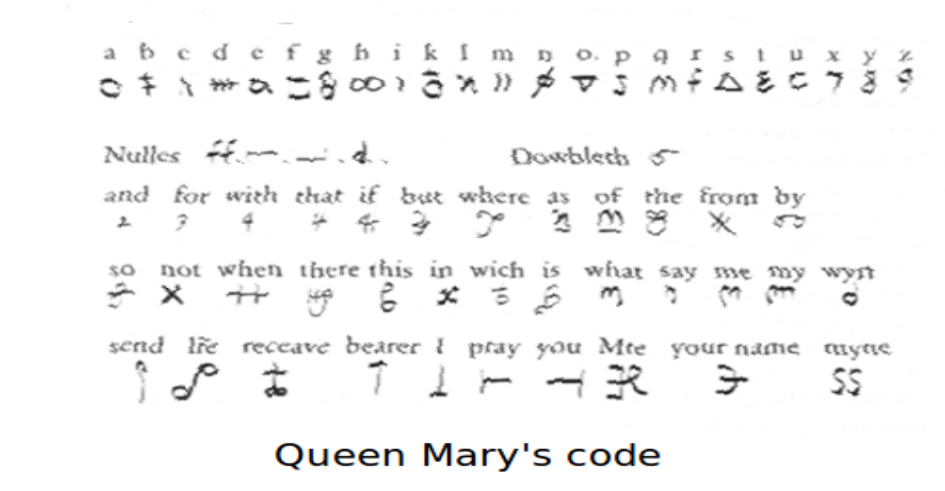
Security

What do I mean?

- Security has many contexts but:
  - For this talk I'm going to concentrate on keeping data communications and also our PCs secure
- This comes down to two topics:
  - Cryptography/Cryptanalysis
  - Weaknesses in wireless LAN technology
- *First, a few definitions.....*

Codes or ciphers?



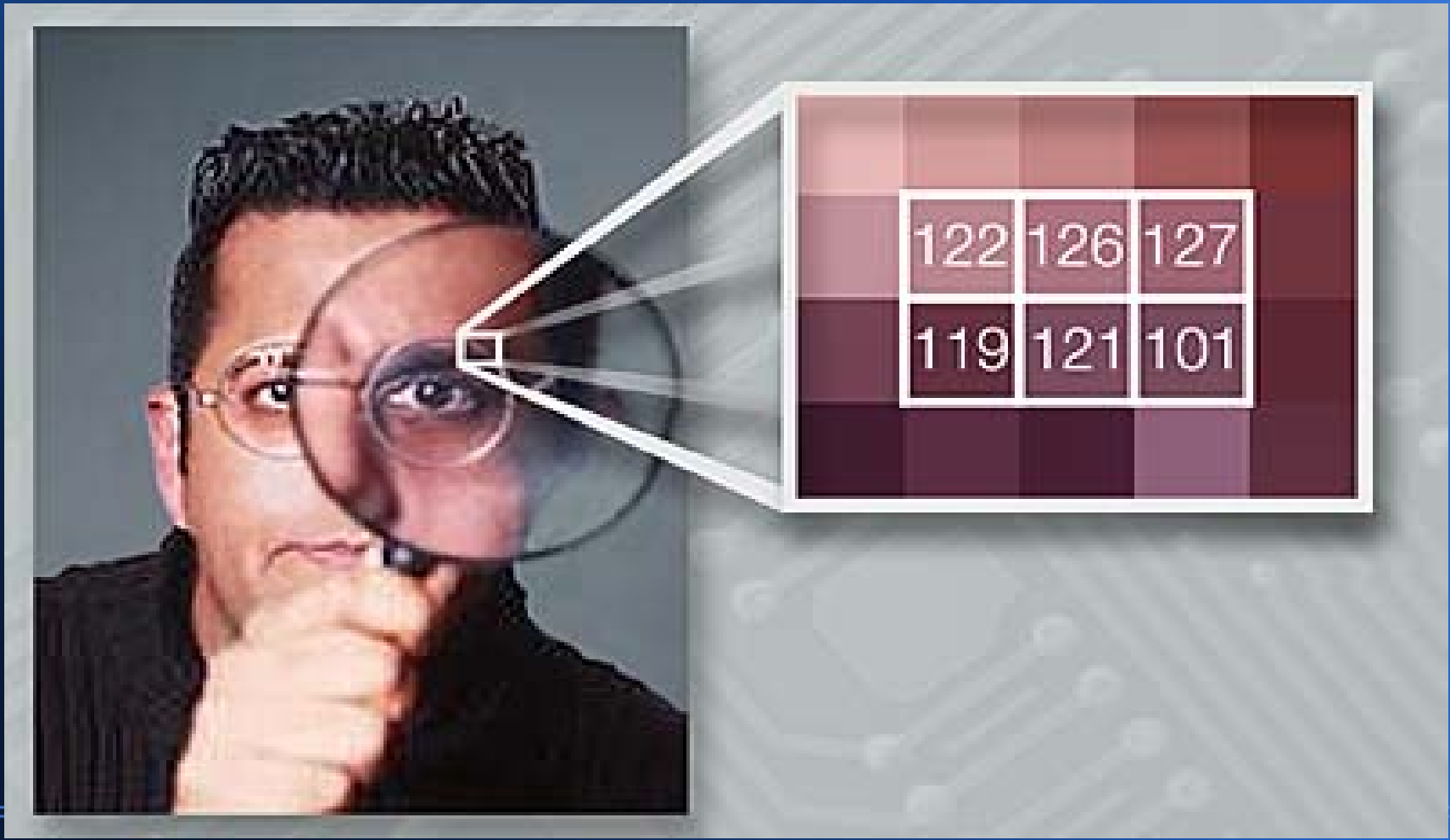
Queen Mary's code

Key	CANADABRAZILEGYPTCUBA
Plaintext	*themeetingisatthedock*
Ciphertext	VHRMHEUZNFQDEZRWXFIDK

A Viginiere cipher

- Codes are complete substitutes for whole words,sentences or entities
- Ciphers substitute or transpose more atomic entities like characters according to a 'key'

Steganography?



Cipher implementations

- Block cipher
  - Where the data is processed as blocks e.g. 2048 or 8192 or 16384 entities at a time
- Stream cipher
  - Where data is output through a shift register, with or without feedback loops. Good for communications work because it is potentially faster.. Potentially stronger because of the 'roll up' of data history created by the feedback. In reality often weaker because of poorly done

What's important ?

- The two most important areas for the average computer user are: secure ordering and banking on-line; and people hijacking their wireless LAN.
- To understand these areas we will have to look at the underlying ciphers and signatures for the SSL (secure sockets-https) and also the mechanisms behind WEP (Wired Equivalent Privacy),WAP, and WAP2 (Wireless Application Protocol) – *First, what ciphers or codes.....?*

Cipher types

- Symmetric ciphers
  - The same key is used by sender and receiver of the data to encrypt and decrypt. Key distribution is a real problem
- Asymmetric ciphers
  - The problem is finding two keys that are related so that one decrypts what the other encrypts and yet retains total security
  - The answer lies in 'one way' mathematical functions

Why are codes and most ciphers no good?

- Codes require both ends to have the same code book -problems changing them! OUT!
- Symmetric ciphers require both ends to have the same 'key'. Easier, but key distribution is still a problem. Keys need to be longer than the message for the cipher to be robust!
- Steganography – both ends need to know where to hide/find the data cells. Very complex for textual data

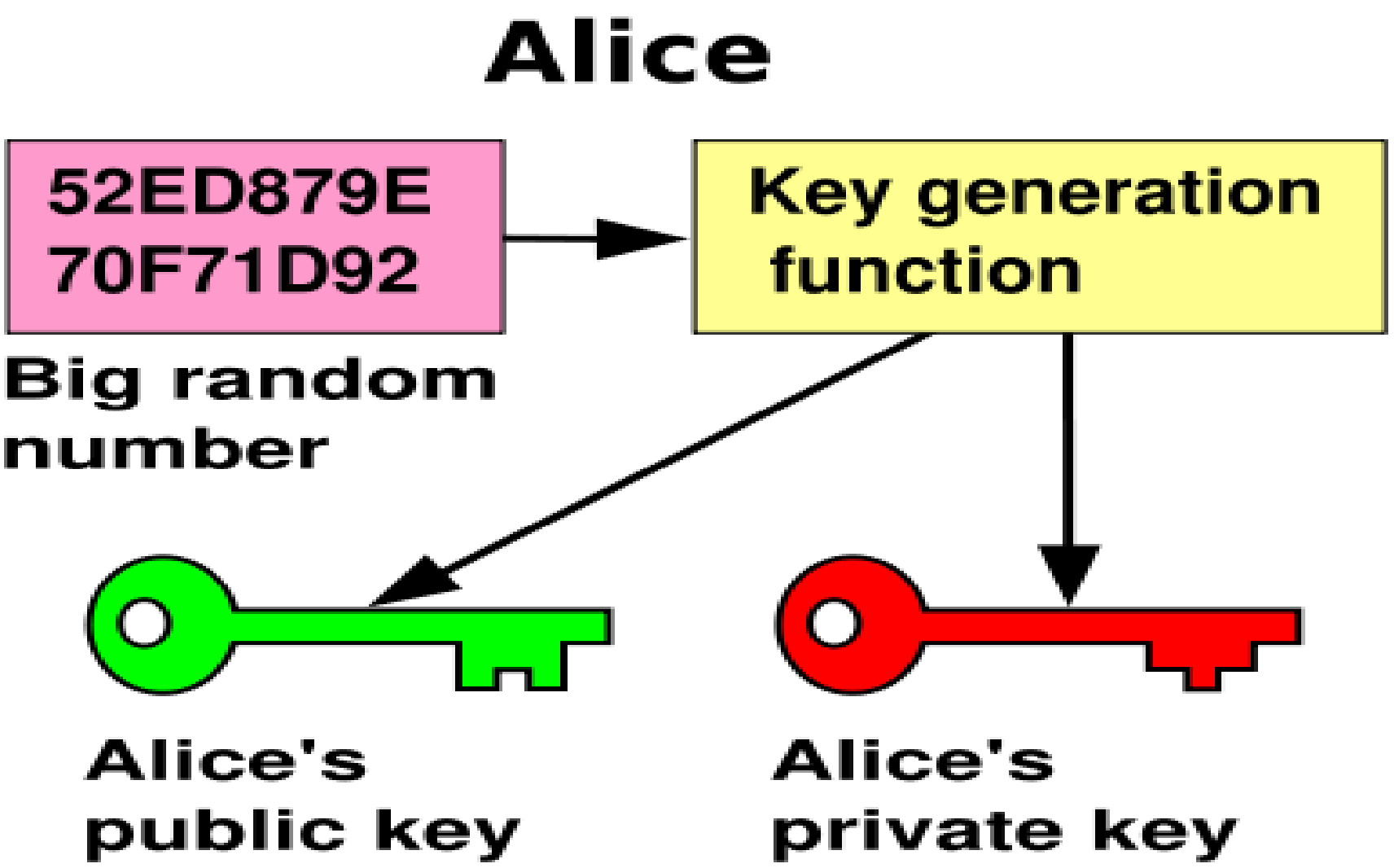
What would be ideal?

- An asymmetric cipher, where different keys are used to encrypt at one end and decrypt at the other.....that somehow they are related!
- That knowing the encryption key does not give any chance whatever of deriving the decryption key.....or if it did it would take longer than the universe has existed to do it!
- A tall order? No. Several versions have been developed (GCHQ, Diffie and Hellman,RSA,DSA)

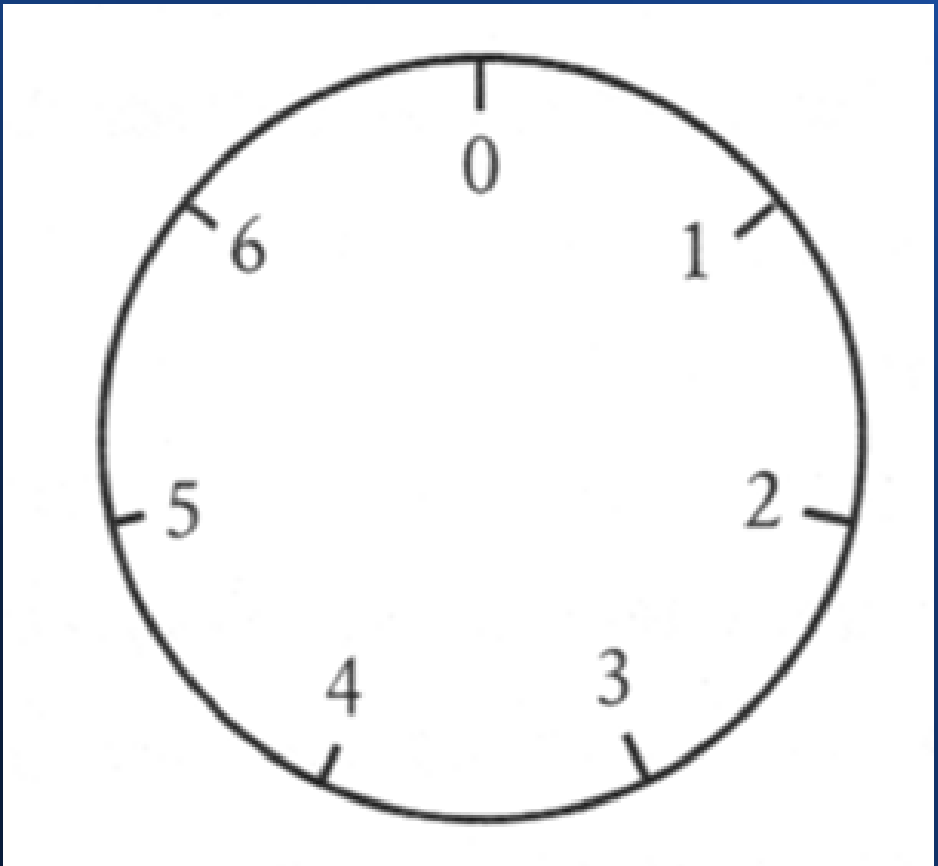
Who is 'Alice'

- There is Alice, and Bob and also Eve
- In cryptography these names are used as 'place-holders'
- They could equally have different names but Alice represents correspondent 'A', Bob represents correspondent 'B', and Eve is a proxy name for an 'eavesdropper' or someone who can intercept communications between 'A' and 'B'

Public key encryption



Key generation I



- Involves Modulo arithmetic – counting to the base n and forgetting carries!
- 6+5 Mod 7 = 4
- 9+7 Mod 13 = 3
- 9+12 Mod 13 =8
- Apply this to powers!

Values of the function 3^x

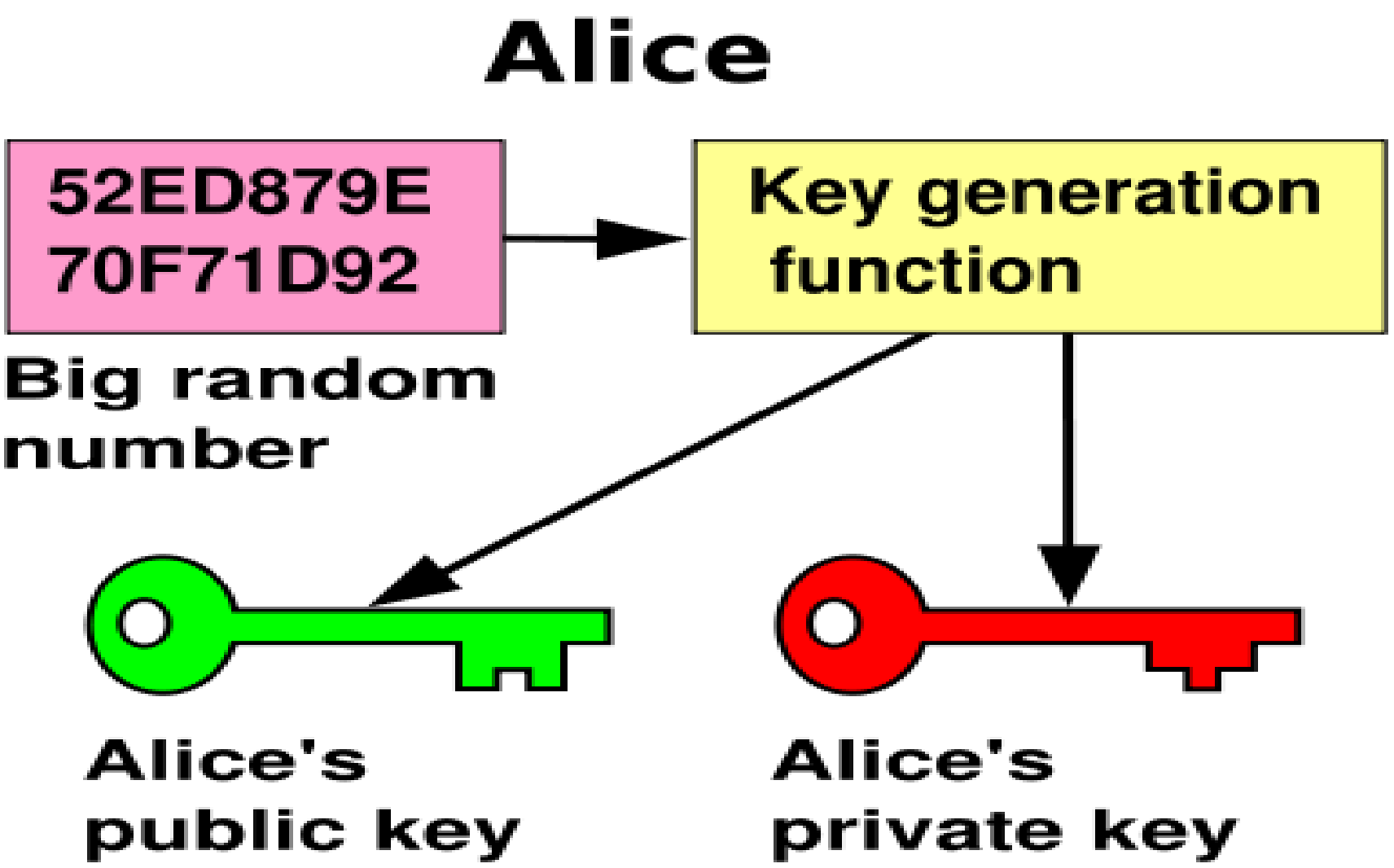
x	1	2	3	4	5	6
3^x	3	9	27	81	243	729
$3^x \pmod 7$	3	2	6	4	5	1

- x increases regularly
- 3^x increases regularly
- $3^x \pmod 7$ is all over the place – relatively unpredictably

Key generation II

- Involves a 'one way function'
- The RSA algorithm is based on the difficulty of factoring large numbers – current estimates say that factoring a 500 digit number requires 10^{21} years (2GHz processor)
- This large number is, in practice, created by choosing two large prime numbers each in the order of $>10^{100}$, and multiplying them together
- There is no known way to solve this except by 'brute force' *see handout sheet for worked example*

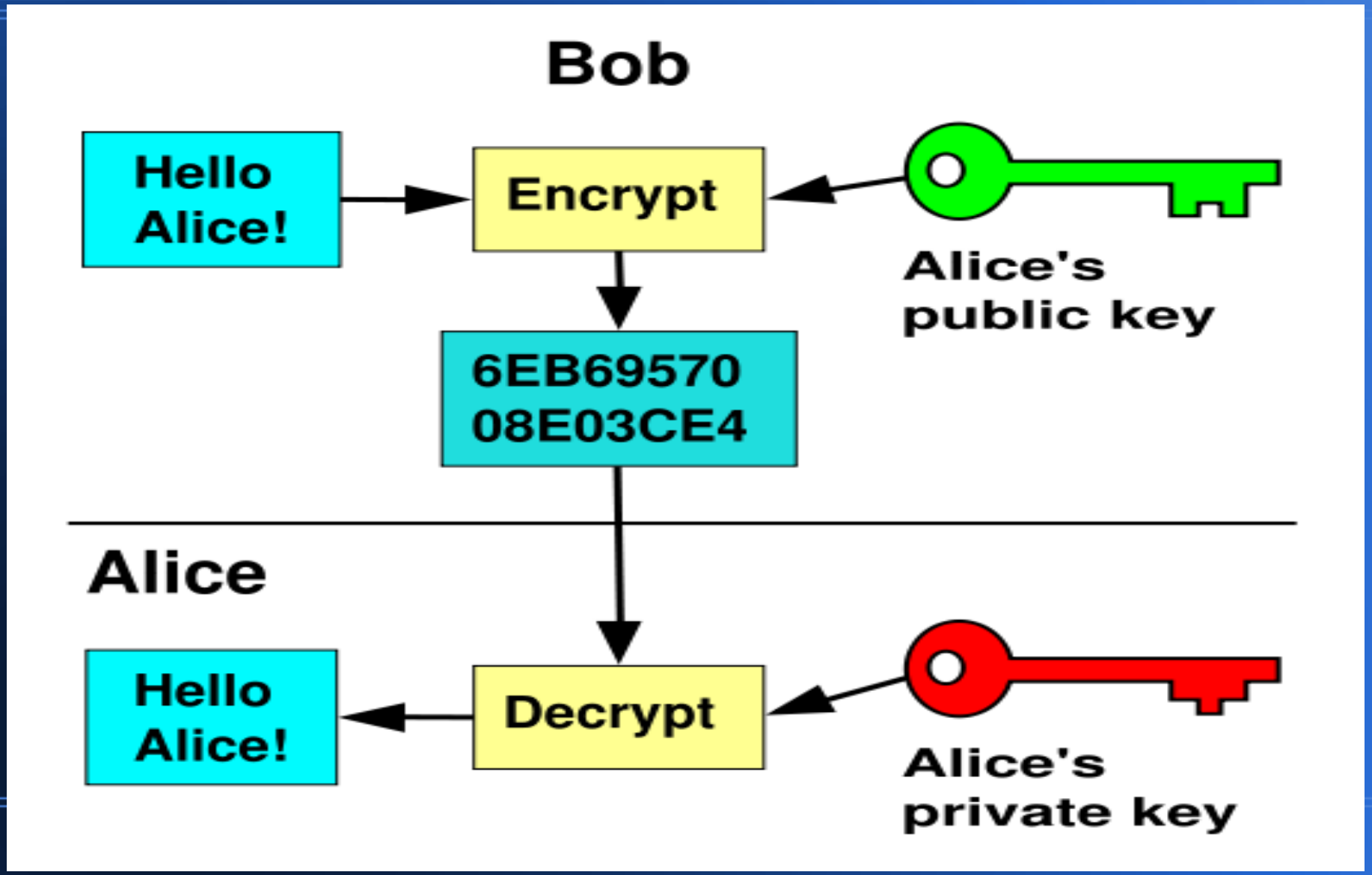
Public key encryption



The encryption mechanism

- Alice could publish her public key on the web or in a directory or through an https server.
- If you want to send a message, securely, to Alice, you use Alice's public key to encrypt the plaintext and then send it to Alice... she can decrypt it using her private key
- The private key is the only one that will work – even the sender cannot decrypt the message once he has encrypted it with the public key!

It looks like this



Back to Symmetry?

- Public key is time consuming and computing needs fast transactions
- There is nothing inherently weak about symmetric key ciphers except key distribution
- They can be fast
- Best of both? A really strong symmetric cipher with key distribution handled by public key encryption/decryption!

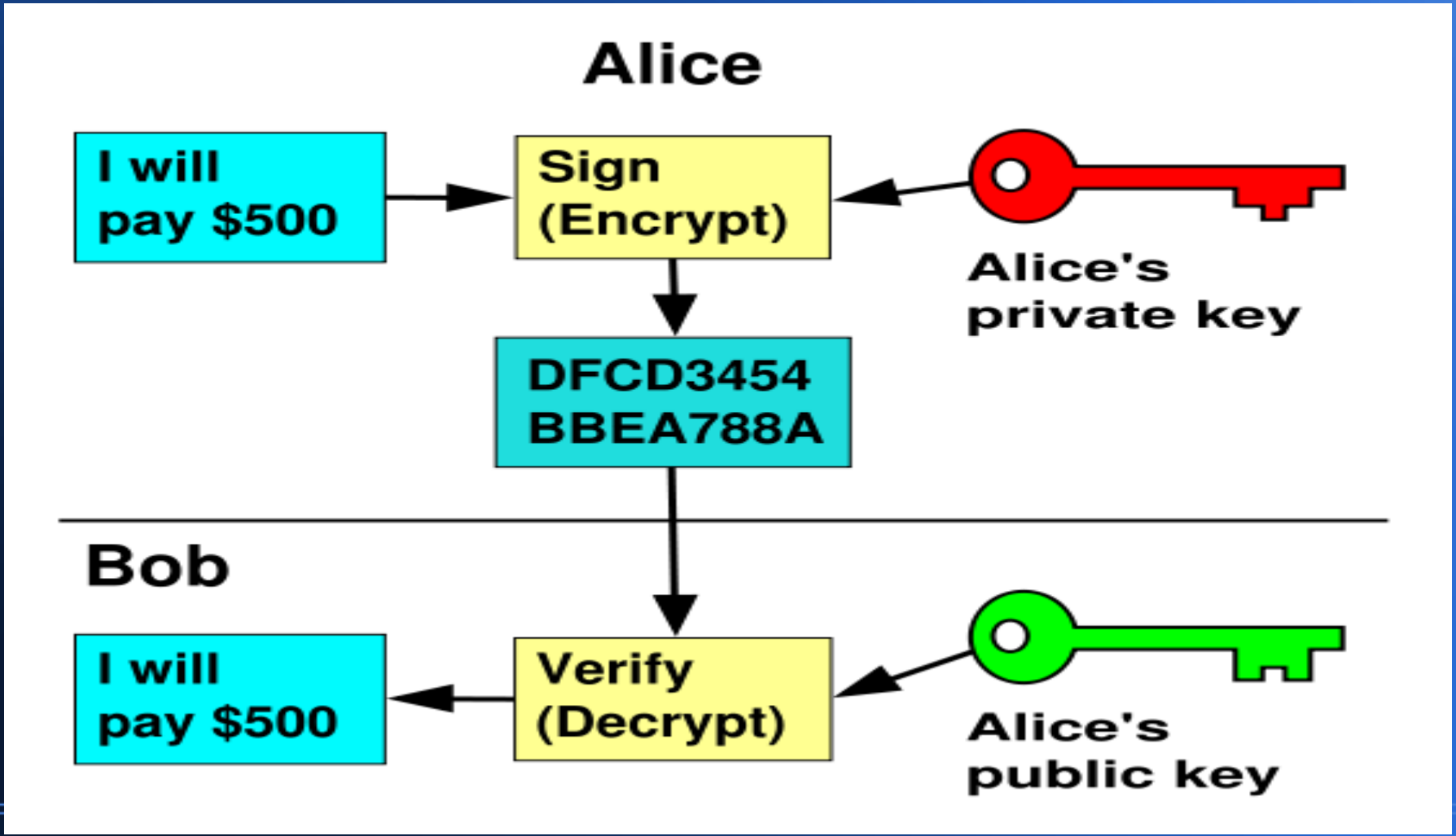
Where does it fit in SSL?

- The heart of encryption in SSL is a symmetric cipher called 'Blowfish'. (a 64 bit 16 round Feistel cipher). [like a 16 rotor software Enigma]
- It is, in itself, 'strong' encryption. There is no known cryptanalysis for a 16 round Feistel cipher (2010). The only potential weakness is key exchange. Blowfish is a FAST process.
- RSA public key encryption is used in an initial exchange for blowfish key derivation

Public key has further uses.....

- Digital signatures
  - Electronic payments can be signed
  - Your private key is used to encrypt your signature
  - Your public key is used by the recipient to decrypt your signature.
  - The private key is kept utterly secret – only data produced by applying it is transmitted
  - Because they are a related key pair it works both ways!

The signature process



Reassurance on that then!

- The encryption methods used are 'strong'
- Your encrypted messages cannot be broken by conceivable effort applied over conceivable time
- Your private keys MUST be kept securely by pass-phrase or better methods e.g. encrypted!
- Don't worry, most systems do this! e.g. PGP
- *What about people hi-jacking your Wi-Fi?*