GPG

The government is watching you...

Josef “Jeff” Sipek <jeffpc@josefsipek.net>

What is cryptography?

Cryptography, n.

The science which studies methods for encoding messages so that they can be read only by a person who knows the secret information required for decoding, called the key

Little History

Little History

- Caesar cipher

Little History

- Caesar cipher
- ROT13

Little History

- Caesar cipher
- ROT13
- Symmetric cipher

Little History

- Caesar cipher
- ROT13
- Symmetric cipher
  - Key used to encrypt & decrypt

Little History

- Caesar cipher
- ROT13
- Symmetric cipher
  - Key used to encrypt & decrypt
  - AES

Little History

- Caesar cipher
- ROT13
- Symmetric cipher
  - Key used to encrypt & decrypt
  - AES
  - Blowfish

Problems with Symmetric Ciphers

- “Pre-shared secret”

Asymmetric Ciphers

Asymmetric Ciphers

- 2 keys!

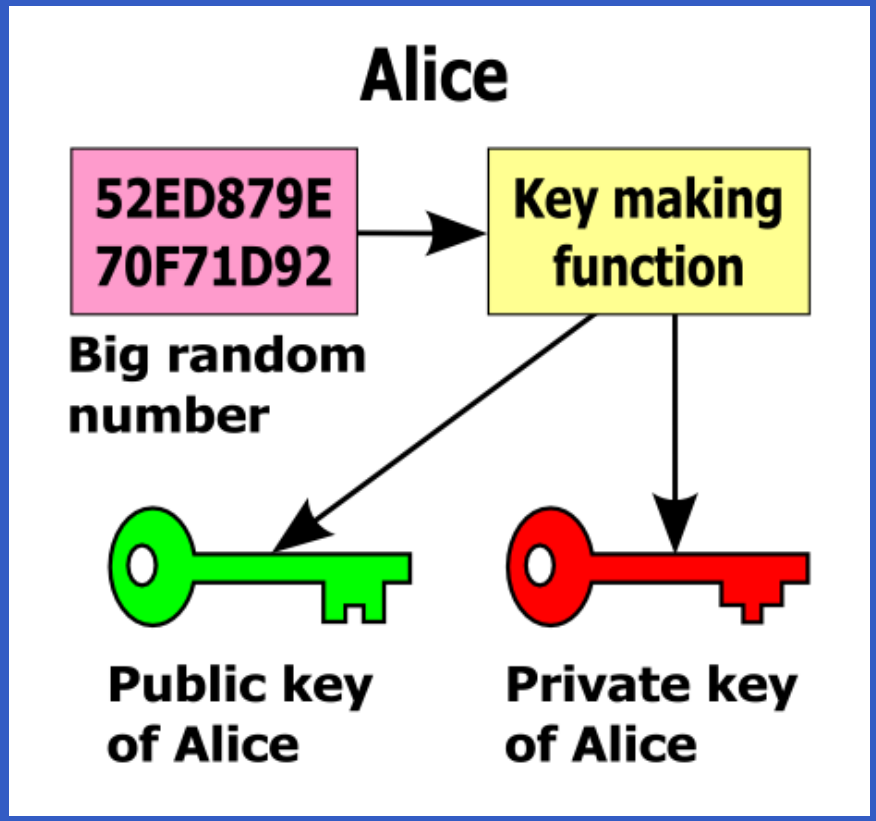
Asymmetric Ciphers

- 2 keys!
  - Public

Asymmetric Ciphers

- 2 keys!
  - Public
  - Private

Asymmetric Ciphers



Asymmetric Ciphers

- Encrypt/Decrypt

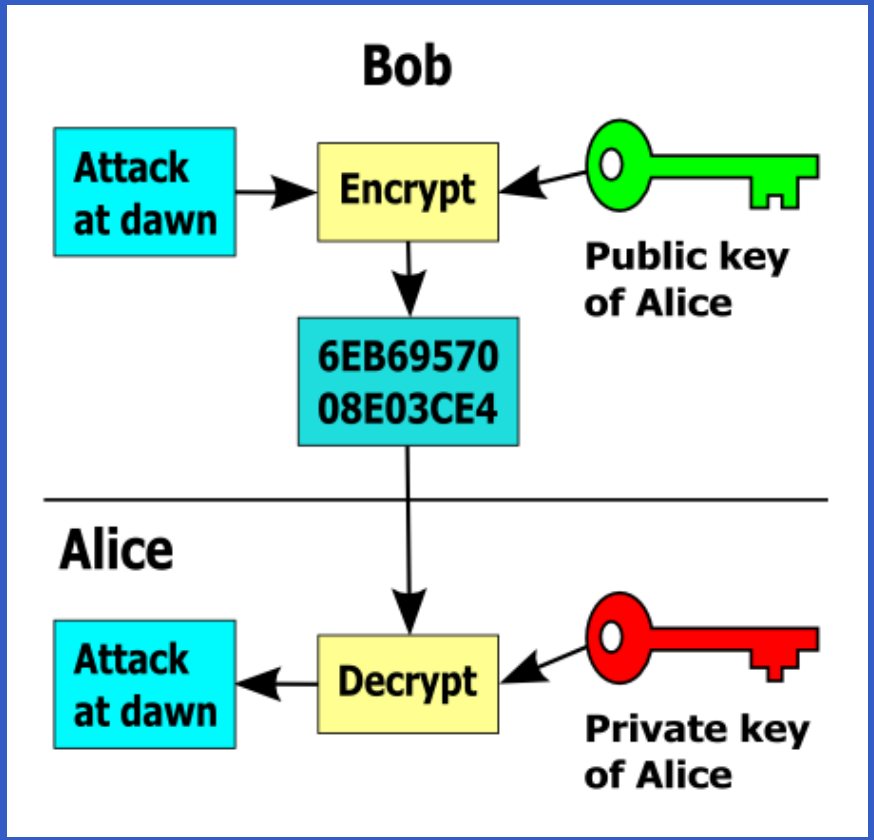
Asymmetric Ciphers

- Encrypt/Decrypt
  - Encrypt with public key

Asymmetric Ciphers

- Encrypt/Decrypt
  - Encrypt with public key
  - Decrypt with private key

Asymmetric Ciphers



Asymmetric Ciphers

- Sign/Verify signature

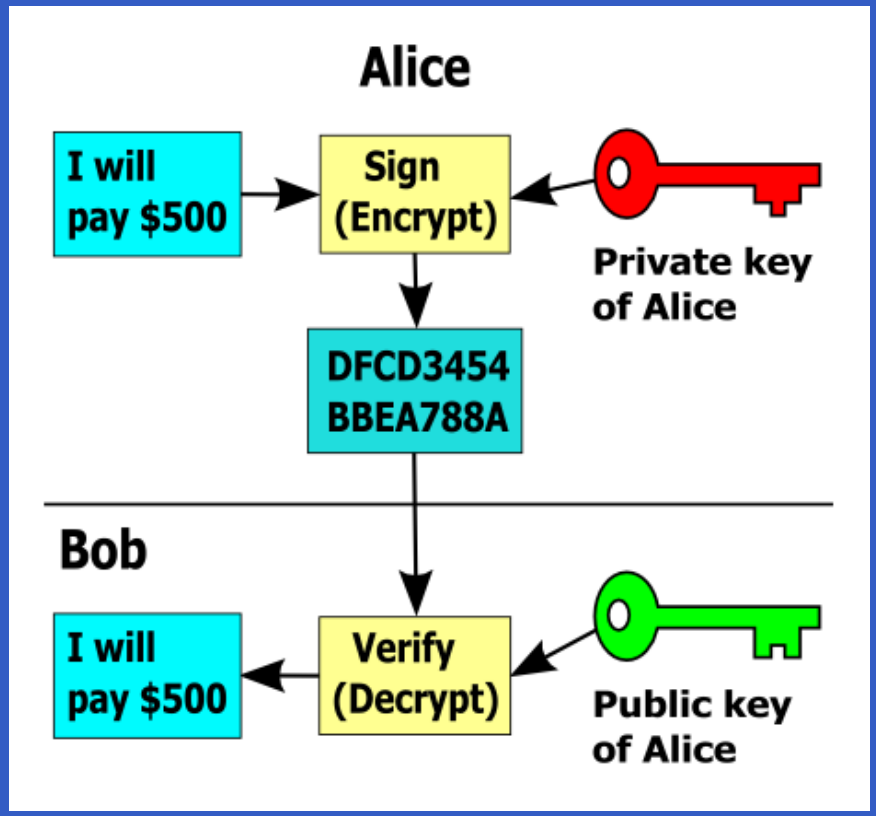
Asymmetric Ciphers

- Sign/Verify signature
  - Sign with private key

Asymmetric Ciphers

- Sign/Verify signature
  - Sign with private key
  - Verify with public key

Asymmetric Ciphers



What, Why, How?

What, Why, How?

- What? — GPG

What, Why, How?

- What? — GPG
- Why?

What, Why, How?

- What? — GPG
- Why?
  - Encrypt/decrypt documents

What, Why, How?

- What? — GPG
- Why?
  - Encrypt/decrypt documents
  - Sign/verify documents

What, Why, How?

- What? — GPG
- Why?
  - Encrypt/decrypt documents
  - Sign/verify documents
- How?

Demo Time!

Demo Time!

- Generate a key

Demo Time!

- Generate a key
- Sign a file

Demo Time!

- Generate a key
- Sign a file
- Verify the signature on a file

Demo Time!

- Generate a key
- Sign a file
- Verify the signature on a file
- Encrypt a file

Demo Time!

- Generate a key
- Sign a file
- Verify the signature on a file
- Encrypt a file
- Decrypt a file

Demo Time!

- Generate a key
- Sign a file
- Verify the signature on a file
- Encrypt a file
- Decrypt a file
- Nasty, eh?

Before Next Meeting

Before Next Meeting

- Generate key pair

Before Next Meeting

- Generate key pair
- Send the *public* key to me

Before Next Meeting

- Generate key pair
- Send the *public* key to me
- I will compile a list of keys

Before Next Meeting

- Generate key pair
- Send the *public* key to me
- I will compile a list of keys
- I will put it on the web

Before Next Meeting

- Generate key pair
- Send the *public* key to me
- I will compile a list of keys
- I will put it on the web
- You will verify that your key is correct

Before Next Meeting

- Generate key pair
- Send the *public* key to me
- I will compile a list of keys
- I will put it on the web
- You will verify that your key is correct
  - If it is *NOT*, let me know *ASAP*

Before Next Meeting

- Generate key pair
- Send the *public* key to me
- I will compile a list of keys
- I will put it on the web
- You will verify that your key is correct
  - If it is *NOT*, let me know *ASAP*
- Print a copy & bring it next month

During Next Meeting

During Next Meeting

- We'll go though each key

During Next Meeting

- We'll go though each key
  - The owner will read it

During Next Meeting

- We’ll go though each key
  - The owner will read it
  - The owner will present sufficient ID

During Next Meeting

- We’ll go though each key
  - The owner will read it
  - The owner will present sufficient ID
  - Rest will verify

After Next Meeting

After Next Meeting

- Go home, and for each key which was verified by *YOU*

After Next Meeting

- Go home, and for each key which was verified by *YOU*
  - Sign it

After Next Meeting

- Go home, and for each key which was verified by *YOU*
  - Sign it
- Send me the signed public keys

After Next Meeting

- Go home, and for each key which was verified by *YOU*
  - Sign it
- Send me the signed public keys
- I'll combine them, and create a LiLUG keyring that everyone can import

References

- Images shamelessly stolen from Wikipedia.
- Wikipedia
  Asymmetric key algorithm
- GNU Privacy Guard
  `http://www.gnupg.org`
- OpenPGP Message Format
  `http://www.ietf.org/rfc/rfc2440.txt`

Q&A

Questions?

Remember: Just because I’m paranoid doesn’t mean they aren’t out to get me.

Caesar Cipher

- To encrypt:
  $e = (c + 3) \, mod \, 26$
- To decrypt:
  $c = (e - 3) \, mod \, 26$

ROT13

- To encrypt:
  $e = (c + 13) \, mod \, 26$
- To decrypt:
  $c = (e - 13) \, mod \, 26 = (e + 13) \, mod \, 26$