moz://a

TLS 101

Diane Hosfelt | @avadacatavra

August 20, 2017

What is cryptography?

Cryptography is a way to secure communications

- Secrecy
- Authentication

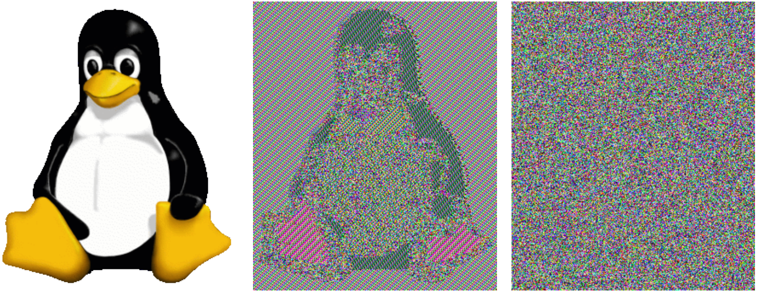


Figure 1: Try this! https://github.com/pakesson/diy-ecb-penguin

Classical vs. Modern Cryptography

Classical:

- Security through obscurity
- Relied on secure channels for key exchange
- Substitution ciphers, codebooks

Modern:

- Kerckhoff's Principal
- Computers + Internet
- Public key cryptography allows insecure channels
- Encryption standards (DES, AES)

Classical crypto: Enigma

- Polyalphabetic cipher

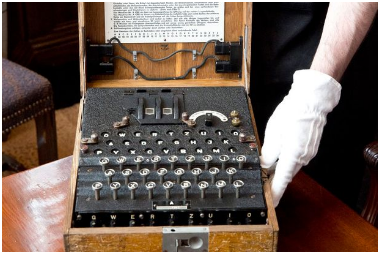


Figure 2: The Enigma machine, used by the Germans in WW2 to perform a complex polyalphabetic cipher

Classical crypto: Enigma

- Polyalphabetic cipher
- rotors

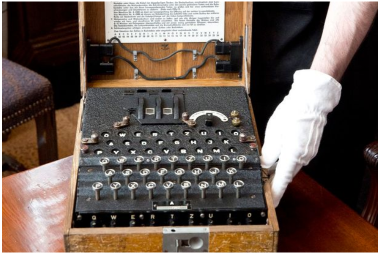


Figure 2: The Enigma machine, used by the Germans in WW2 to perform a complex polyalphabetic cipher

Classical crypto: Enigma

- Polyalphabetic cipher
- rotors
- operator error

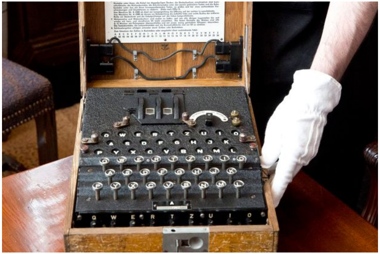


Figure 2: The Enigma machine, used by the Germans in WW2 to perform a complex polyalphabetic cipher

Classical crypto: Enigma

- Polyalphabetic cipher
- rotors
- operator error
- Marian Rejewski

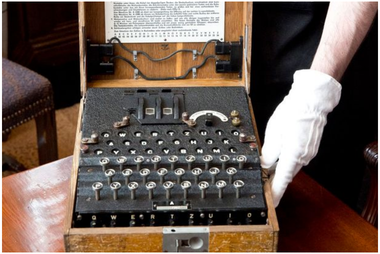


Figure 2: The Enigma machine, used by the Germans in WW2 to perform a complex polyalphabetic cipher

Classical crypto: Enigma

- Polyalphabetic cipher
- rotors
- operator error
- Marian Rejewski
- Cribbing - 26x26x26 = 17576 trials to brute force

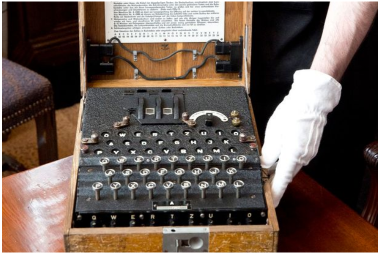


Figure 2: The Enigma machine, used by the Germans in WW2 to perform a complex polyalphabetic cipher

How has cryptography changed?

Kerckhoff's Principal

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

One time pad $\oplus$

- Vernam's cipher
- One-time pre-shared random key
- key length $\geq$ message length
- Perfect secrecy

Different types of cryptography

Symmetric Alice and Bob both know one secret key s_k

- What do you think is a problem with this method?

Asymmetric Two keys: one for encryption, one for decryption

- One way function

Example

Alice and Bob each have public and private keys, which they use to compute a shared secret key

- Diffie-Hellman exchange published in 1976
- RSA invented in 1978
- What ideas do you have about how they can make their secret key?

Public key cryptography

How can you pass a secure message to someone you've never met?

Public key cryptography!

- Instead of one key, have two
- Private key: only you know the private key
- Public key: anyone can see this

Public key Exchange

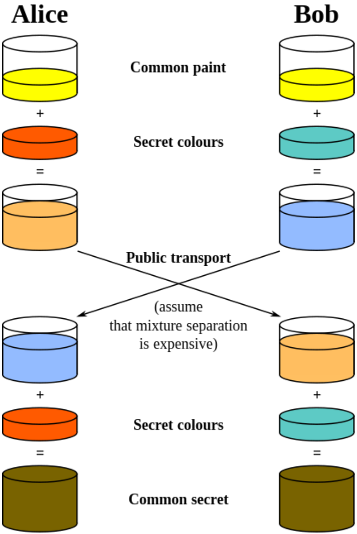
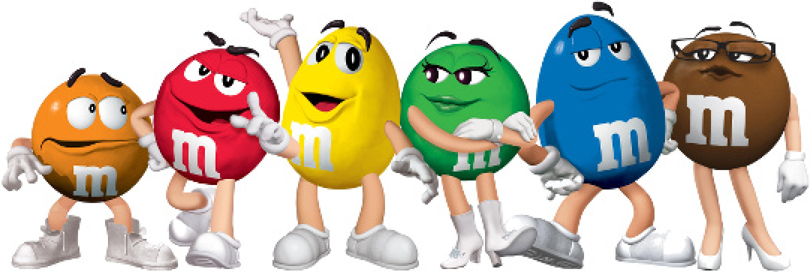


Figure 3: Image by A.J. Han Vinck, University of Duisburg

Chocolate Key Exchange



Secure sockets layer

- Invented by Netscape
- SSL v2 in 1995
- SSL v3 in 1996
  - deprecated in June 2015
  - Vulnerable to POODLE (block ciphers)
  - RC4 sucks (only non-block cipher in v3)
- TLS 1.0 in 1999
- TLS 1.1 in 2006
- TLS 1.2 in 2008

TLS

Implementations

- OpenSSL
- NSS
- BoringSSL
- LibreSSL
- SecureChannel

privacy secure and reliable

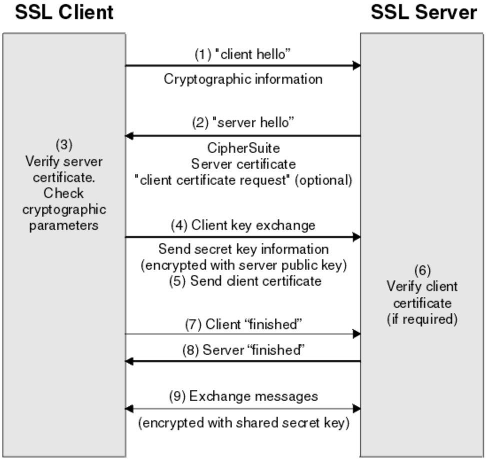
authentication required for at least one party (server)

forward secrecy future key disclosure can't decrypt past comms

The Handshake

What we need to do:

- Agree on protocol version and algorithms
- Authenticate each other
- Establish a shared symmetric key



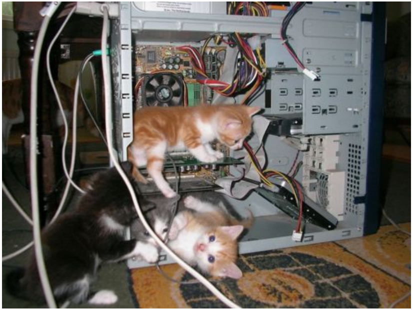
SSL Client
SSL Server
(1) "client hello"
Cryptographic information
(2) "server hello"
CipherSuite
Server certificate
"client certificate request" (optional)
(3) Verify server certificate. Check cryptographic parameters
(4) Client key exchange
Send secret key information
(encrypted with server public key)
(5) Send client certificate
(6) Verify client certificate (if required)
(7) Client "finished"
(8) Server "finished"
(9) Exchange messages
(encrypted with shared secret key)

TLS attacks

- Downgrade attacks
- Replay attacks
- POODLE
- Heartbleed
- BEAST
- Lucky13

Challenges of TLS

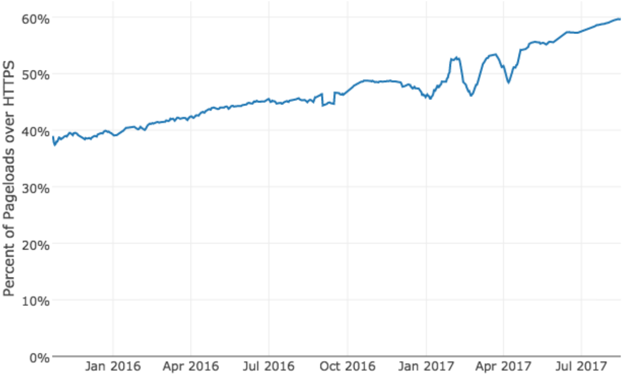
- encryption is hard
- implementation is hard
- memory is hard
- configuration is hard



TLS and HTTPS and you

- piggybacks HTTP on top of TLS
- trust via certificate authorities
- not E2E

Figure 4: Percentage of webpages loaded by Firefox over HTTPS



The Future

- TLS 1.3
  - Remove support for MD5 and other terrible things
  - HKDF
  - 1-RTT and 0-RTT
- formal verification
- constant time asm compilers?
- memory safe languages?

Learn more

- Overview of TLS handshake
- TLS1.3 vs TLS 1.2
- Matt Green's blog

Questions?