ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Cryptography Basics

COM-402: Information Security and Privacy

(slide credits: Kirill Nikitin, Dusan Kostic, and Bryan Ford)

Outline

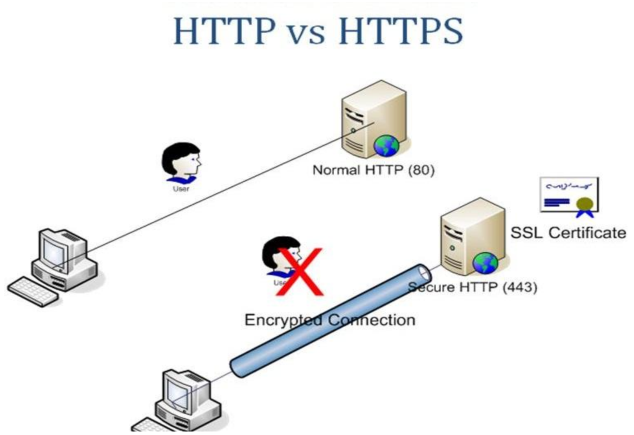
- Introduction
- Symmetric-key cryptography
- Data integrity
- Public-key cryptography
- Key infrastructure
- End-to-end encryption

Outline

- **Introduction**
  - Goals of cryptography
  - Shared-algorithm cryptography, substitution ciphers
  - One-time pad
- Symmetric-key cryptography
- Data integrity
- Public-key cryptography
- Key infrastructure
- End-to-end encryption

Crypto is Everywhere

- Secure communication
  - HTTPS, WPA, SSH, GSM, …
- File and disk encryption
  - VeraCrypt, BitLocker, GPG, …
- Authentication of users and servers
- Cryptocurrencies
- Digital Rights Management
- ...



Introduction

- What is cryptography?
  - A toolbox for many security mechanisms
  - Information security and communication security
  - Secure data at rest and data in motion
- Cryptography is not:
  - The solution to all security problems
  - Reliable unless implemented properly
  - Reliable unless used properly
  - Something you should try to invent or implement yourself

Credit: https://crypto.stanford.edu/cs155/lectures/07-crypto.pdf

Goals of Cryptography

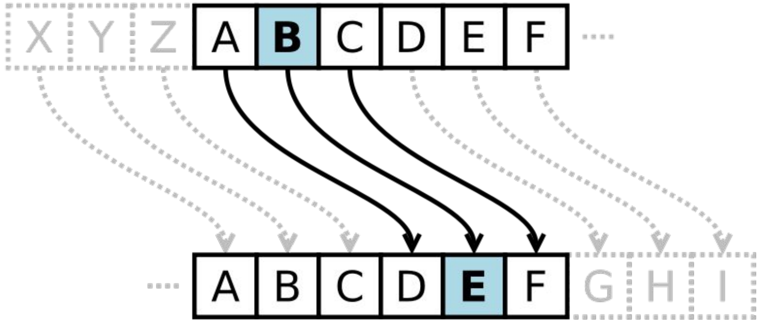
- **Confidentiality:** only users with appropriate key can access information
- **Integrity:** ensure that data is not modified by unauthorized parties
- **Authentication:** provide a way to identify the author of information
- **Accountability:** the responsibility for actions
- **Availability:** achieving other goals shouldn’t impose excessive overhead
- **Timestamping:** a statement is attested for being seen at given time
- ...

Brief History

- 50BC - Caesar’s Cipher - substitution
- 1883. Kerckhoffs’ Principle – A cryptosystem should be secure even if everything about the system, except the key, is public knowledge
- 1917./1918. One-time pad
- Before and during the WW II - Enigma machine
- 70s - Data Encryption Standard (DES)
- 70s - Diffie-Hellman Public Key Crypto
- 70s - Rivest, Shamir, Adleman (RSA)
- 90s - Digital Signature Algorithm (DSA)
- 90s - Secure Hash Algorithm (SHA-1)
- 2001 - SHA-2
- 2001 - Advanced Encryption Standard (AES)
- 2015 - SHA-3

Naive Approach

- Two parties agree on an encryption algorithm (e.g., rot13) and keep it secret
    - 50BC - Caesar's Cipher - substitution
    - Use it to encrypt messages to each other



- 1883. Kerckhoffs' Principle – A cryptosystem should be secure even if everything about the system, except the key, is public knowledge

One-Time Pad

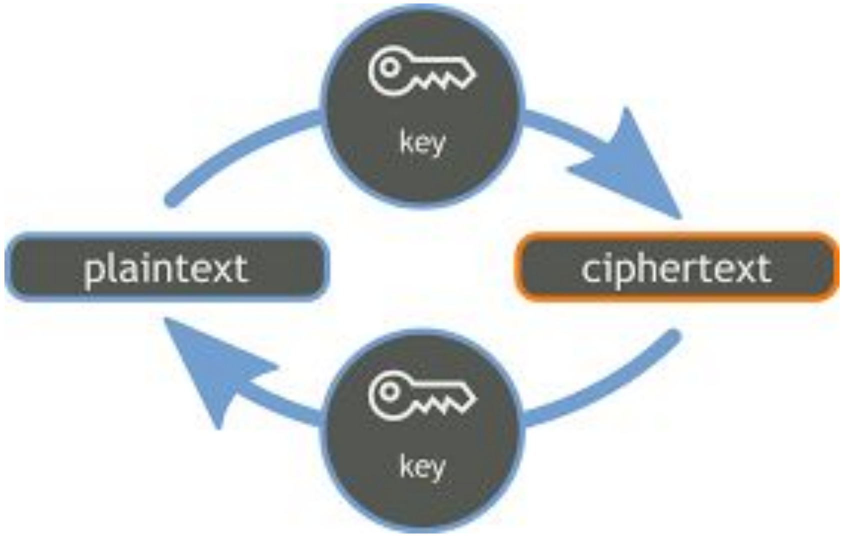
- One-time pad (OTP)
  - c = Encryption(m) = m $\oplus$ key
  - m = Decryption(c) = c $\oplus$ key
  - key is random string *at least* as long as the plaintext
- Provides “perfect” secrecy in principle
- Practical disadvantages:
  - Requires truly uniform random one-time pad values
  - Keys must not be used more than once
  - Key length depends on the message length

Outline

- Introduction
- **Symmetric-key cryptography**
  - Stream ciphers
  - Block ciphers, modes of operation
- Data integrity
- Public-key cryptography
- Key infrastructure
- End-to-end encryption

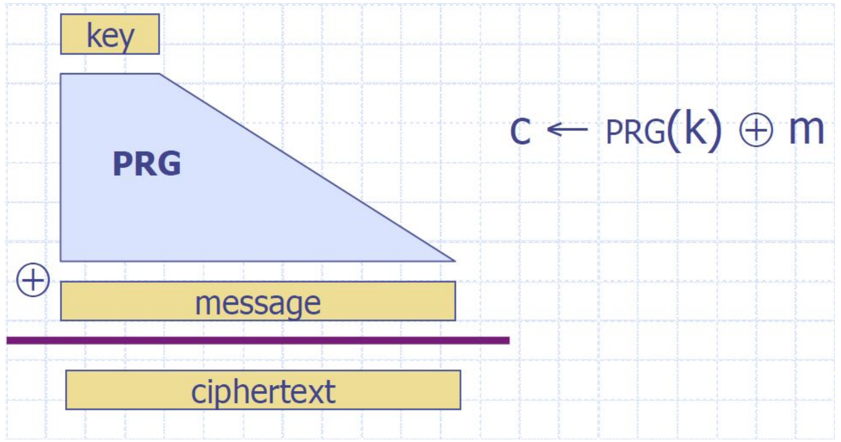
Definition

- Encryption of plaintext and decryption of ciphertext are done using a well-known algorithm and the same key, hence *symmetric* crypto



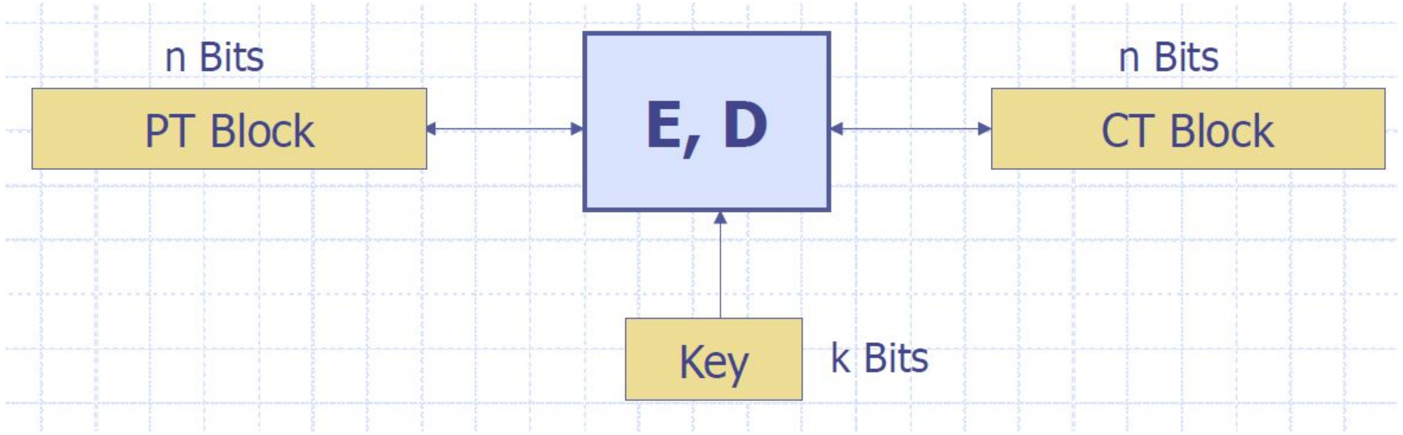
Stream Ciphers

- Making OTP practical (and less secure)
- Require Pseudo Random Generators
- Examples: RC4 (used in HTTPS and WEP), CSS (used for DVD encryption), E0 (used in Bluetooth), A5/1,2 (used in GSM encryption), Salsa20/ChaCha, …



Block Ciphers

- Encrypt blocks of data of fixed size
- Modes of operation handle variable length data

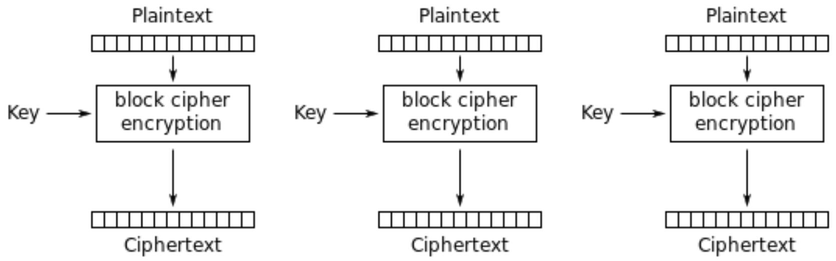


Examples of Block Ciphers

- Data Encryption Standard (DES):
  - Block size 64 bits
  - Key size 56 bits
  - Deprecated
- Advanced Encryption Standard (AES):
  - Block size 128 bits
  - Key size 128/192/256 bits
  - Hardware support in Intel and AMD processors

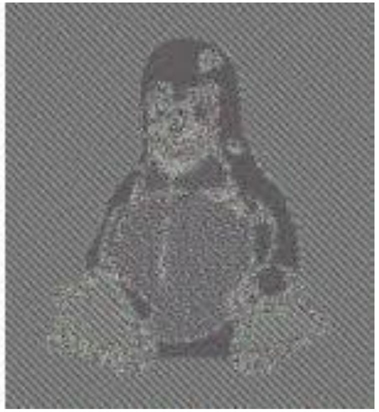
Modes of Operation - Electronic Code Book

- Electronic Code Book (ECB)
- Example of a *bad* mode of operation (insecure, obsolete): same plaintext blocks encrypt to same ciphertext blocks



Electronic Codebook (ECB) mode encryption

Original

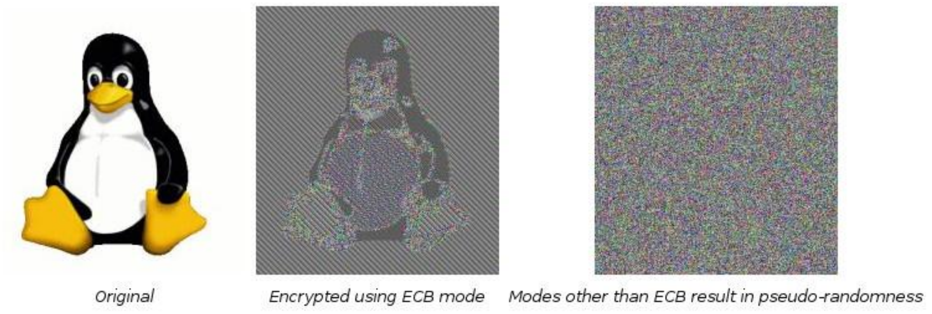


ECB-encrypted image

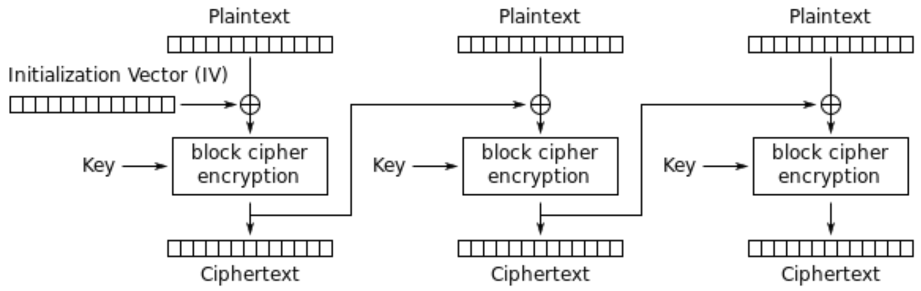
en.wikipedia.org

Modes of Operation - Cipher Block Chaining

- Cipher Block Chaining (CBC)
- A good, secure mode of operation (when used correctly)



en.wikipedia.org



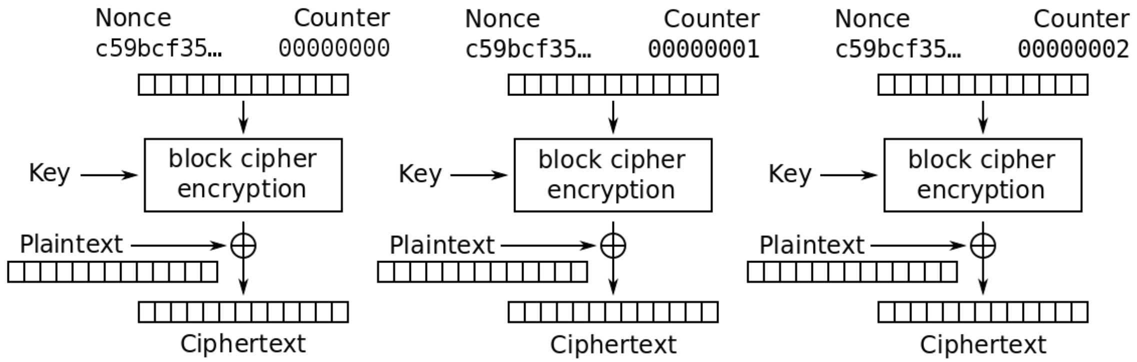
Cipher Block Chaining (CBC) mode encryption

Modes of Operation - Counter Mode (CTR)

- Turns a block cipher into a stream cipher
- Generates the next keystream block by encrypting values of a "counter"

➔ (+) Parallelizable, software and hardware efficient, random access to blocks, provable security, simplicity, message of arbitrary bit length

➔ (–) No integrity, error propagation, stateful encryption



Counter (CTR) mode encryption

Figure credit: Wikipedia

Outline

- Introduction
- Symmetric-key cryptography
- **Data integrity**
  - Cryptographic hash functions
  - Message Authentication Codes
  - Authenticated encryption
- Public-key cryptography
- Key infrastructure
- End-to-end encryption

Data integrity

- Goal: provide integrity, not confidentiality
- Integrity: ensure that data is not modified by unauthorized parties
- Symmetric crypto doesn’t provide integrity by itself
    - Example: flipping a bit in OTP cipher text results in flipping the same bit in the plaintext after decryption:

$$(M_1 \oplus K) \oplus M_2 = (M_1 \oplus M_2) \oplus K$$

    - Examples for block ciphers are a bit more complicated, but neither block ciphers provide integrity by default

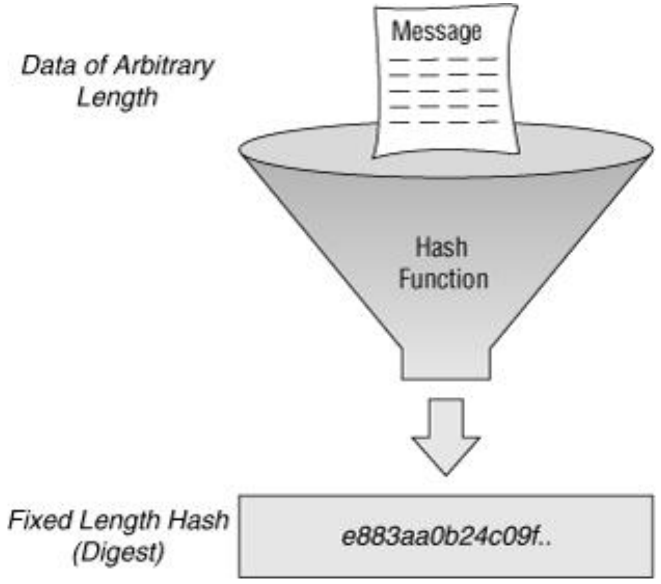
Cryptographic hash functions

- Map bit-strings of any length to a fixed-length output in a deterministic way
- Desirable properties of hash functions (informal definitions):
  - **One-way:** given y it is infeasible

    to find any x such that y = h(x)
  - **Collision-resistance:**

    infeasible to find x and x' such that h(x)=h(x')
  - **Pseudo-randomness:**

    indistinguishable from a random oracle



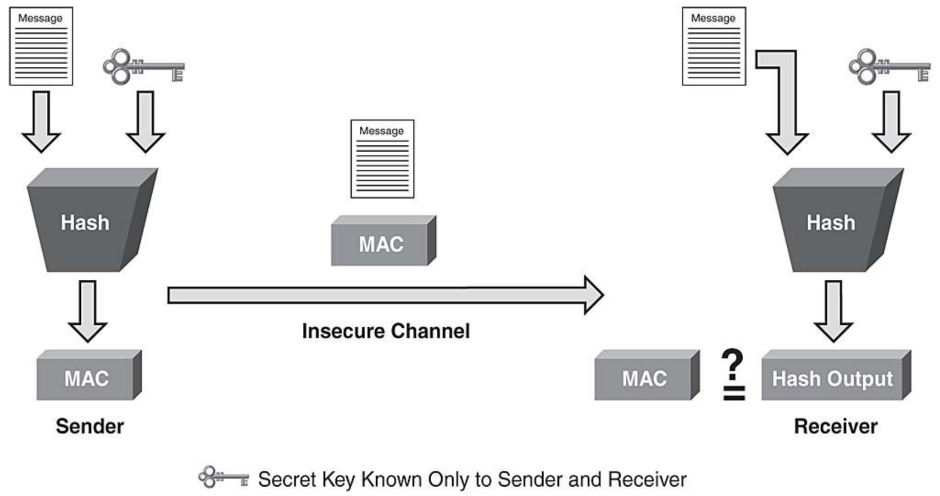
[source: Cisco]

Cryptographic hash functions

- **Password storage:** store only the hash of a password and compare it with the hashed value of user's input
  - Linux's /etc/shadow, user credentials on websites
  - Use of salt to increase entropy
- **Files/Messages integrity verification:** message authentication codes
- **Proof-of-work:** consensus mechanism in Bitcoin
- **Key derivation:** high-entropy secret from user's low-entropy input
- **Commitments:** committing to some value and accountably revealing it later
- **Blockchains:** record the hash of a previous block to ensure immutability
- …

Message Authentication Code (MAC)

- Sender sends a message to receiver, such that receiver is able to verify the origin of the message and that the message is not modified in transit
- Sender and receiver have a shared secret key
- MAC is a small (constant-size) digest of any arbitrarily large message



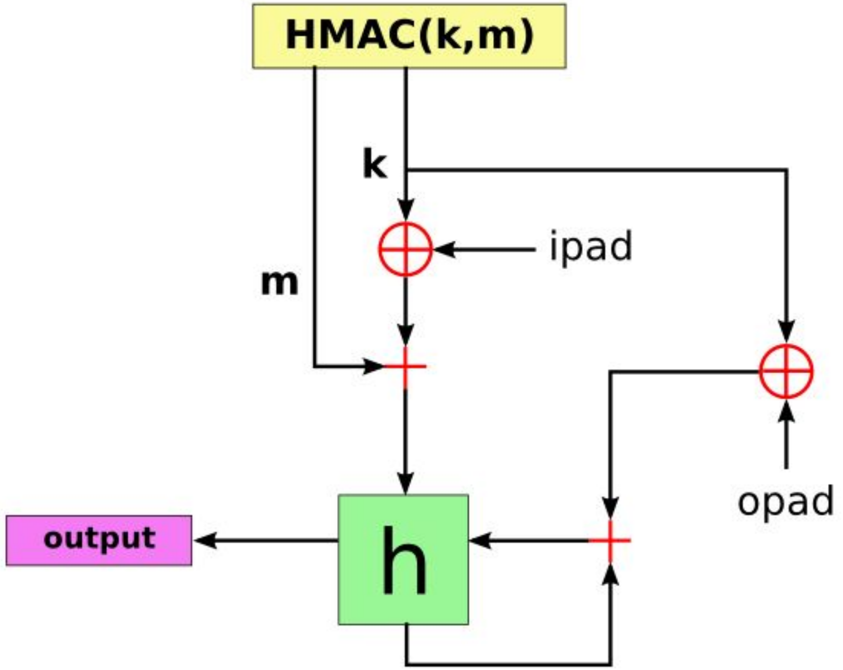
Source: NetworkWorld

Keyed-Hash Message Authentication Codes

- Takes an input message and a key, yields a message digest that depends on both
- Hard to derive message or key from resulting hash
- Hard to find any relationship between hashes w/ different keys
- Naive implementation: just use unkeyed hash on key + message, for subtle cryptographic reasons, not the best design

Hash Message Authentication Code

- Hash Message Authentication Code (HMAC)
- Examples: HMAC_SHA1, HMAC_SHA256, …
- Used in TLS, IPsec, …



Source: Wikipedia

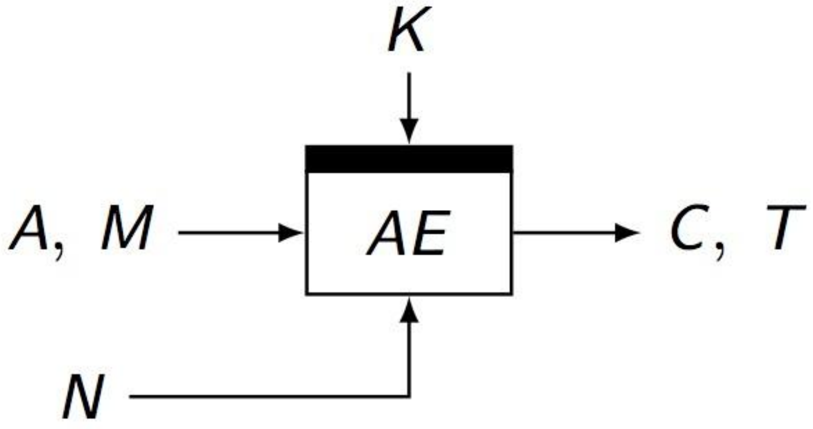
Authenticated encryption

- Goal – ensure confidentiality and integrity together
- Authenticated Encryption (AE) abstraction introduced in 2000
  - Less error-prone primitive to use as a “black box”
- Two main ways to construct AE schemes:
  1. Combine MAC and unauthenticated encryption schemes:
     a. MAC-then-Encrypt - Original SSL
     b. Encrypt-and-MAC - Original SSH
     c. Encrypt-then-MAC - IPSec
        i. The latest versions of all SSL, SSH, IPSec use Encrypt-the-Mac
  2. Create fresh AE schemes “from scratch” for this purpose

Authenticated Encryption [w/ Additional Data]

Parameters:

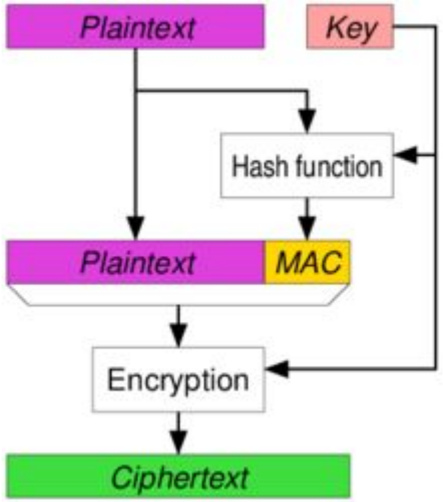
- *K*: key
- *A*, *M*: additional data and message
- *N*: nonce
- *C*, *T*: ciphertext and tag



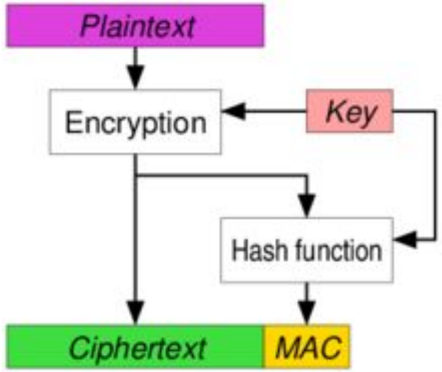
Properties:

- Message M is **both** authenticated (by tag T) **and** encrypted (to ciphertext C)
- Additional data A is **only** authenticated (by tag T) **but not** encrypted

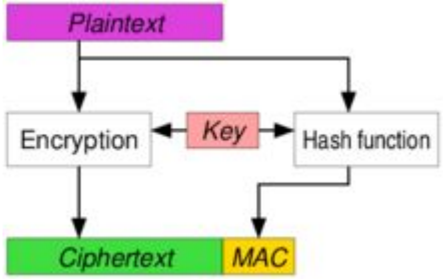
Combination Approaches - Detail



MAC-then-Encrypt



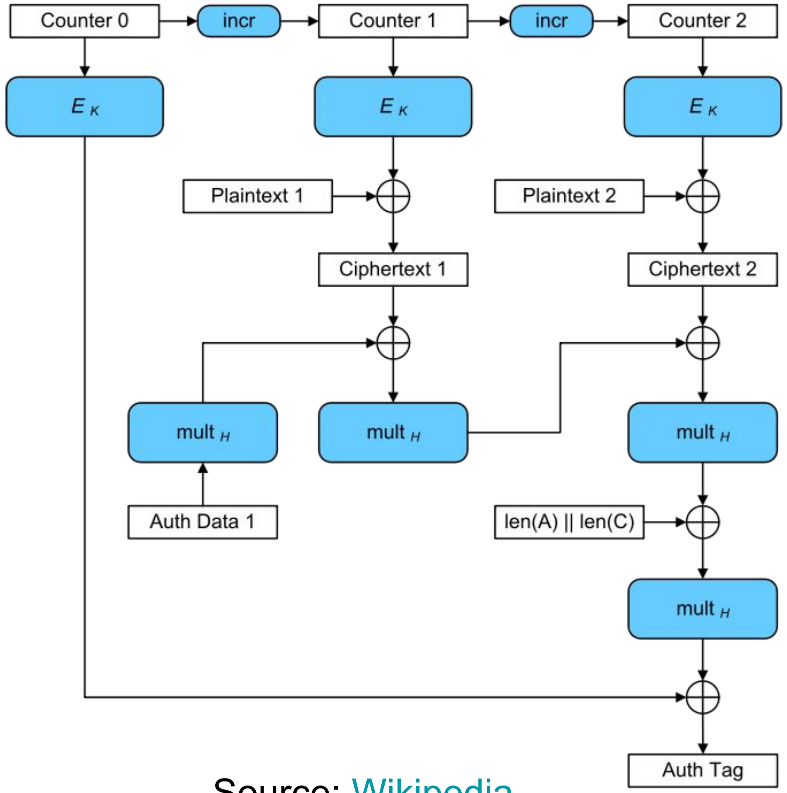
Encrypt-then-MAC



Encrypt-and-MAC

Galois/Counter Mode (GCM)

- Authenticated Encryption with Associated Data (AEAD) mode
- Based on a block cipher E_k (usually AES)



Source: Wikipedia

Outline

- Introduction
- Symmetric-key cryptography
- Data integrity
- **Public-key cryptography**
    - Interactive key exchange
    - Digital signatures
- Key infrastructure
- End-to-end encryption

Public-key Cryptography

- Solves the problem of having to agree on a pre-shared key
- (Public, private) key pair instead
- Public & private key mathematically related
  - use large-number arithmetic
  - leverages computations believed to be difficult
- "Owner" of identity holds private key secret, distributes public key to communication partners

First Approach of Asymmetric Crypto

- 1975. Diffie and Hellman in “New directions in cryptography” describe the idea of asymmetric (public key) cryptography:

We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals.

In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

Interesting Facts

- DH key exchange was published in 1976
- RSA scheme was published in 1977

Rewind...

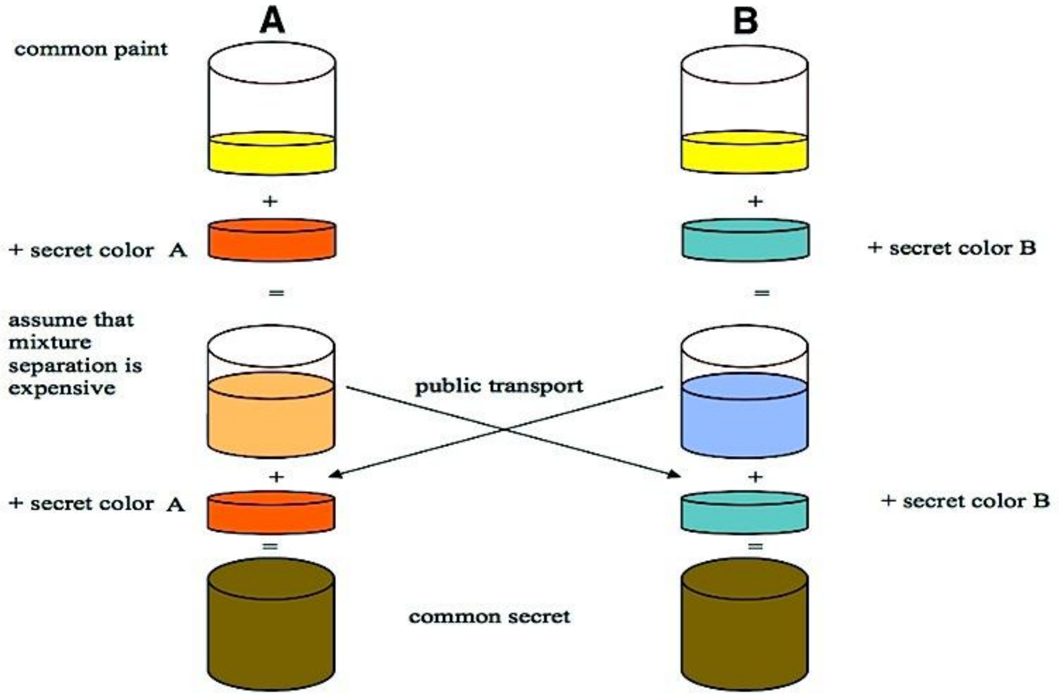
- 1970. James Ellis at UK GCHQ realises that the idea of public key crypto is possible, in a secret report “The possibility of Non-secret Encryption”
- 1973. Clifford Cocks at UK GCHQ discovers a workable mathematical formula for non-secret encryption in a secret report “A note on Non-Secret Encryption”. His formula was a special case of the RSA algorithm
- 1974. Malcolm Williamson at UK GCHQ describes a key exchange method in a secret report “Non-Secret Encryption Using a Finite Field”. His method was similar to the one discovered by Diffie and Hellman

Primitives

- Public and private key
  - Two keys (numbers), *public* is distributed widely, *private* is kept secret
  - Easy: f(private) -> public
  - Hard: f(public) -> private
- Encryption and Decryption
  - Encrypt with public key, decrypt with private key
  - Hard to decrypt without the private key
- Interactive key exchange
  - Create a shared *private* key over an insecure communication channel
- Digital signatures
  - Sign with the private key, verify the signature using the public key
  - Hard to create a signature if only public key is known

Interactive Key Exchange

- Diffie-Hellman Key Exchange



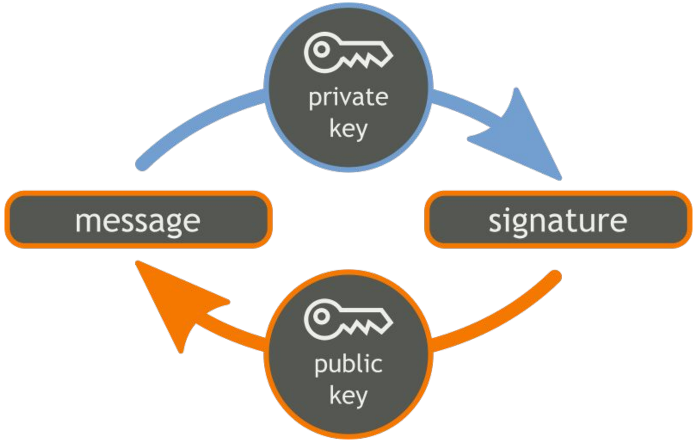
Credit: A.J. Han Vinck, Introduction to public key cryptography

Diffie-Hellman Key Exchange

- Steps
  - Alice and Bob agree on a finite algebraic group *G* with generator *g*
  - Alice picks a random *a* and sends *g^a* to Bob
  - Bob picks a random *b* and sends *g^b* to Alice
  - Alice computes *s = (g^b)^a*
  - Bob computes *s = (g^a)^b*
- Security of DH relies on hardness of the Discrete Logarithm Problem
- DLP is not hard in all groups: must choose appropriately
- Application example: Handshake protocol in TLS

Digital signatures

- Public-key counterpart of MACs (with important differences)
- Provides a way for a signer *S* with a key pair (*pk, sk*) to sign a message in such a way that any other party who knows *pk* can verify that *S* is the author of the message and that the message has not been not modified

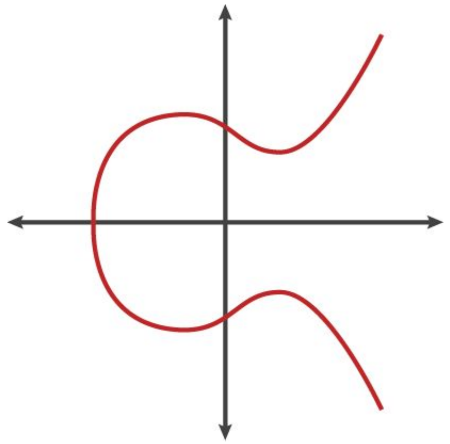


Digital signatures - Usage examples

- Software-update distribution
  - Verification that a software update indeed comes from the releasing company and that it has not been modified on its way
- Authenticated email
  - Provide a way for Alice to verify that the email she received from Bob is indeed from Bob and not from someone else
  - Symmetric crypto already solves this but users must share a secret key
- Public-key certificates
- ...

Elliptic Curve Cryptography

- Elliptic curve cryptography (ECC) is based on the algebraic structure of elliptic curves over finite fields.
- Smaller keys for equivalent security than traditional crypto (e.g., 256-bit for ECC comparable to 2048-bit RSA)
  - Faster operations
  - Smaller public keys -> smaller certificates and less data
- Popular secure curves (over GF(p), p prime):
  - Curve25519
  - Curve448



Elliptic Curve Digital Signature Algorithm (ECDSA)

- A variant of the Digital Signature Algorithm (DSA) using elliptic curve crypto
- For 80-bit security (2^{80} operations), ECDSA public key – 160 bits, DSA – 1024 bits; whereas signature size is the same
- Used in Bitcoin to authenticate transactions and every Bitcoin address is a cryptographic hash of an ECDSA public key
- iMessages and iCloud keychain syncing rely on ECDSA

Elliptic Curve Digital Signature Algorithm (ECDSA)

- Requires random or unpredictable data as input, unique for each signature
- 2010: Recovery of the ECDSA private key used by Sony to sign software for the PlayStation 3 due to the lack of randomness
- 2013: Loss of funds in Android Bitcoin Wallet due to a faulty random number generator

Outline

- Introduction
- Symmetric-key cryptography
- Data integrity
- Public-key cryptography
- **Key infrastructure**
  - Key distribution
  - Public-key infrastructure
- End-to-end encryption

Key Distribution

- For both symmetric and asymmetric crypto we have to distribute keys
- Symmetric cryptosystems require the exchange of secret keys
  - Need for a secret/confidential channel
- Asymmetric cryptosystems require the exchange of public keys
  - Need for a trusted/integrity protected channel
- Authorities trusted to provide secret / trustworthy keys:
  - Key Distribution Centers (KDC)
  - Certification Authorities (CA)

Using Hierarchy of Trust

- One KDC/CA is not enough to serve all users
- KDCs/CAs are organized into hierarchies or peer networks

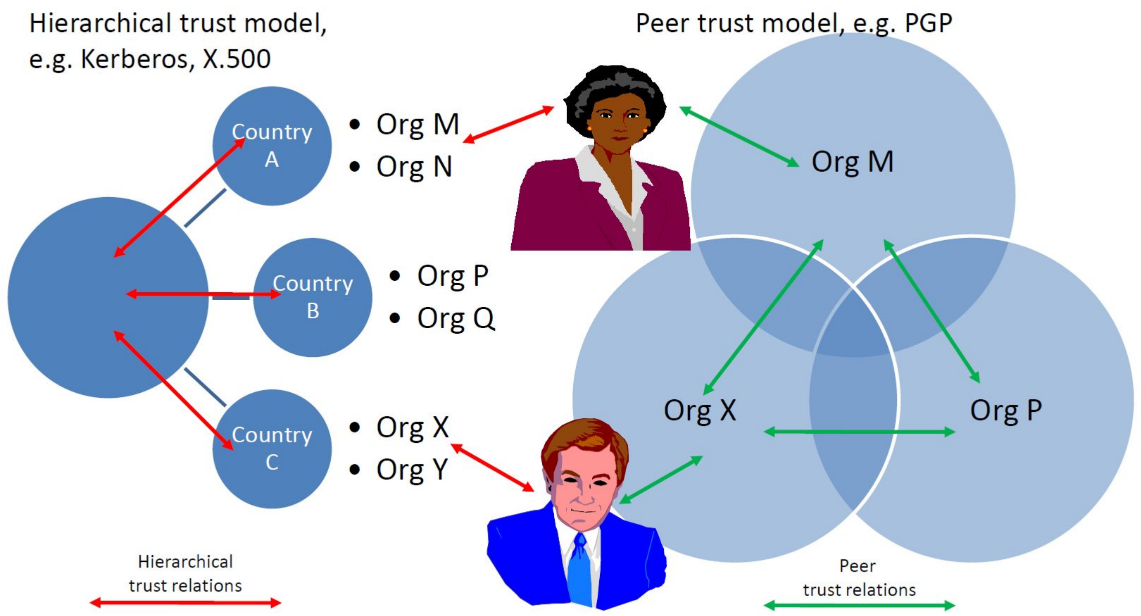


Figure Credit: P. Janson "IT Security Engineering" course

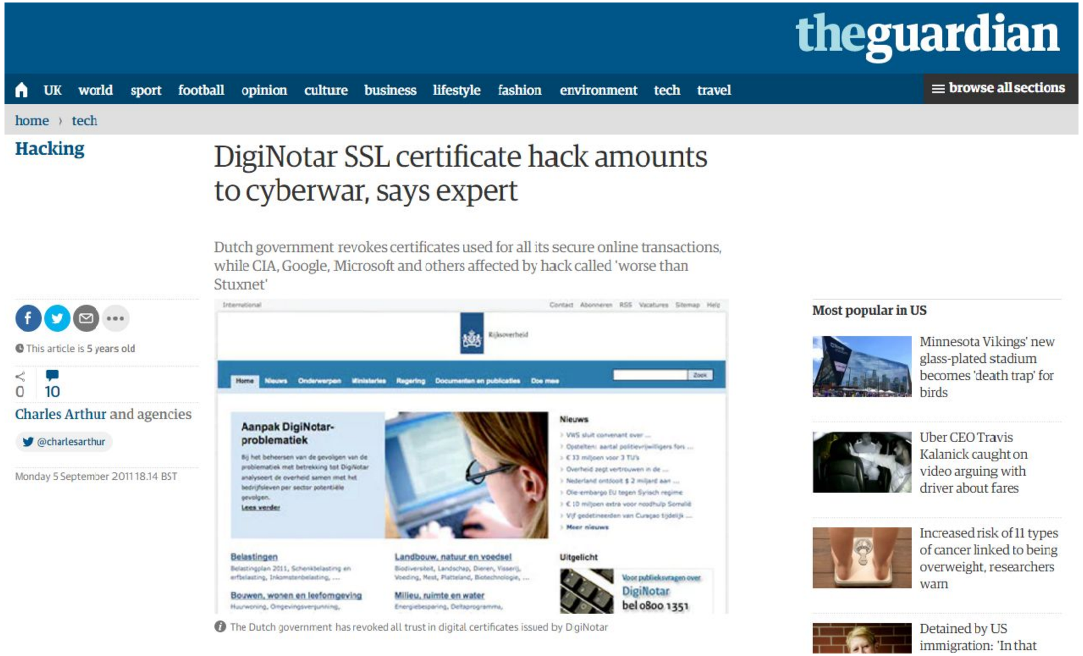
Public Key Infrastructure

PKI binds *public keys* with their *owners*

- Certificate Authority (CA): stores, issues and signs the digital certificates
- Root certificate public keys are embedded in web browser
- Few Root CAs sign "delegation" certificates declaring that other CAs are also trusted to sign server certs
- Subsidiary "issuing" CA signs a cert for, e.g., Google servers
- When your browser connects to Google server via SSL, the server sends its server-side cert and the "chain" of signatures down from the root CA

Attacks

- Huge problem when CA gets compromised (Comodo, DigiNotar)



theguardian

UK world sport football opinion culture business lifestyle fashion environment tech travel ≡ browse all sections

home › tech

Hacking

DigiNotar SSL certificate hack amounts to cyberwar, says expert

Dutch government revokes certificates used for all its secure online transactions, while CIA, Google, Microsoft and others affected by hack called 'worse than Stuxnet'

This article is 5 years old

0 10

Charles Arthur and agencies

@charlesarthur

Monday 5 September 2011 18.14 BST

International Contact Abonneren RSS Vacatures Sitemap Help

Rijksoverheid

Home Nieuws Onderwerpen Ministeries Regering Documenten en publicaties Doe mee Zoek

Aanpak DigiNotar-problematiek

Bij het beheersen van de gevolgen van de problematiek met betrekking tot DigiNotar analyseert de overheid samen met het bedrijfsleven per sector potentiële gevolgen.

Lees verder

Nieuws

- VWS sluit convenant over ...
- Opstelten: aantal politievrijwilligers fors ...
- € 33 miljoen voor 3 TU's
- Overheid zegt vertrouwen in de ...
- Nederland ontdooit $ 2 miljard aan ...
- Olie-embargo EU tegen Syrisch regime
- € 10 miljoen extra voor noodhulp Somalië
- Vijf gedetineerden van Curaçao tijdelijk ...
- Meer nieuws

Belastingen

Belastingplan 2011, Schenkbelasting en erfbelasting, Inkomstenbelasting, ...

Landbouw, natuur en voedsel

Biodiversiteit, Landschap, Dieren, Visserij, Voeding, Mest, Platteland, Biotechnologie, ...

Bouwen, wonen en leefomgeving

Huurwoning, Omgevingsvergunning,

Milieu, ruimte en water

Energiebesparing, Deltaprogramma,

Uitgelicht

Voor publieksvragen over DigiNotar bel 0800 1351

The Dutch government has revoked all trust in digital certificates issued by DigiNotar

Most popular in US

Minnesota Vikings' new glass-plated stadium becomes 'death trap' for birds

Uber CEO Travis Kalanick caught on video arguing with driver about fares

Increased risk of 11 types of cancer linked to being overweight, researchers warn

Detained by US immigration: 'In that

Methods Used for PKI

- Certificate Authorities (CAs):
- Web of Trust (WOT):
  - PGP, GnuPG
  - Self-signed certificates
- Simple Public-Key Infrastructure (SPKI):
  - experimental, tries to overcome the complexities of CAs and WOT
- CONIKS: Key Transparency

Outline

- Introduction
- Symmetric-key cryptography
- Data integrity
- Public-key cryptography
- Key infrastructure
- **End-to-end encryption**
  - TLS
  - HTTPS

End-to-end encryption

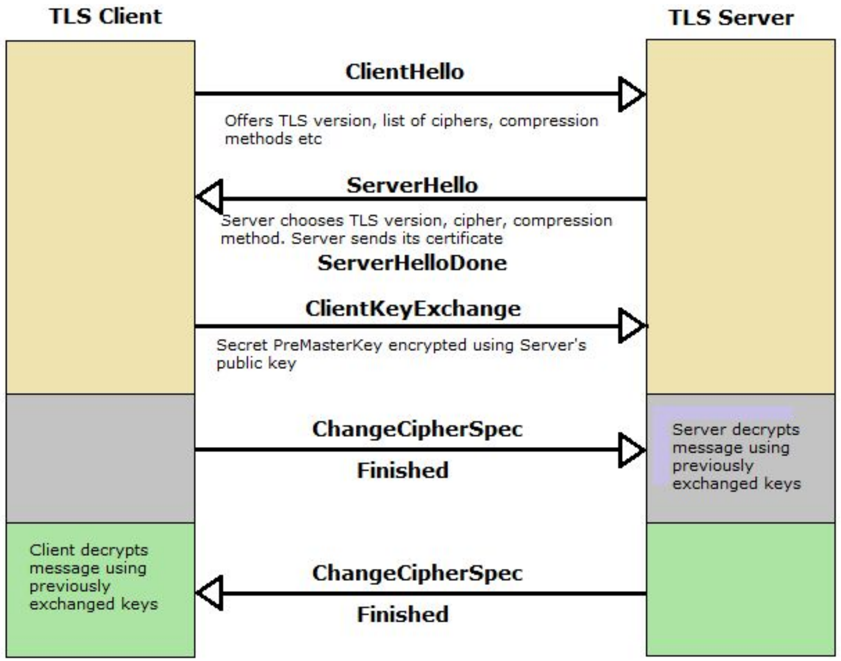
- Only the communicating users can read the messages
- In principle, prevents potential eavesdroppers – including telecom and Internet providers – from being able to decrypt the conversation
- Examples: TLS, Signal Protocol, ...

TLS: Definition

- Transport Layer Security / Secure Sockets Layer
- “TLS protocol aims primarily to provide privacy and data integrity between two communicating computer applications”
- Widely used: Web browsing, email, VoIP, instant messaging, etc.
- History overview:
    - SSL 1.0, 2.0, 3.0 - developed at Netscape in 95/96 (deprecated)
    - TLS 1.0 - defined in 1999 in RFC 2246
    - TLS 1.1 - defined in 2006 in RFC 4346
    - TLS 1.2 - defined in 2008 in RFC 5246
    - TLS 1.3 - working draft since July 2016

TLS: Details

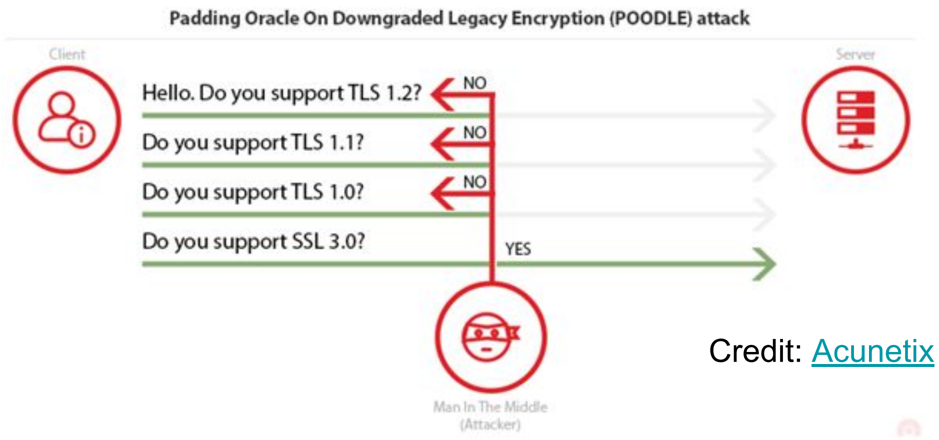
- Client and server run the TLS handshake protocol to negotiate a connection


TLS Client
TLS Server
ClientHello
Offers TLS version, list of ciphers, compression methods etc
ServerHello
Server chooses TLS version, cipher, compression method. Server sends its certificate
ServerHelloDone
ClientKeyExchange
Secret PreMasterKey encrypted using Server's public key
ChangeCipherSpec
Finished
Server decrypts message using previously exchanged keys
Client decrypts message using previously exchanged keys
ChangeCipherSpec
Finished

Source: KeywordSuggest

Some Historical TLS Vulnerabilities

- Downgrade attack:
    - Trick the server into using an insecure version of TLS
    - Negotiate a connection with weak keys (FREAK, 2014.)
    - Force the server to do downgrade to weak DH groups (Logjam, 2015.)


Padding Oracle On Downgraded Legacy Encryption (POODLE) attack
Client
Server
Hello. Do you support TLS 1.2?
NO
Do you support TLS 1.1?
NO
Do you support TLS 1.0?
NO
Do you support SSL 3.0?
YES
Man In The Middle (Attacker)

Credit: Acunetix

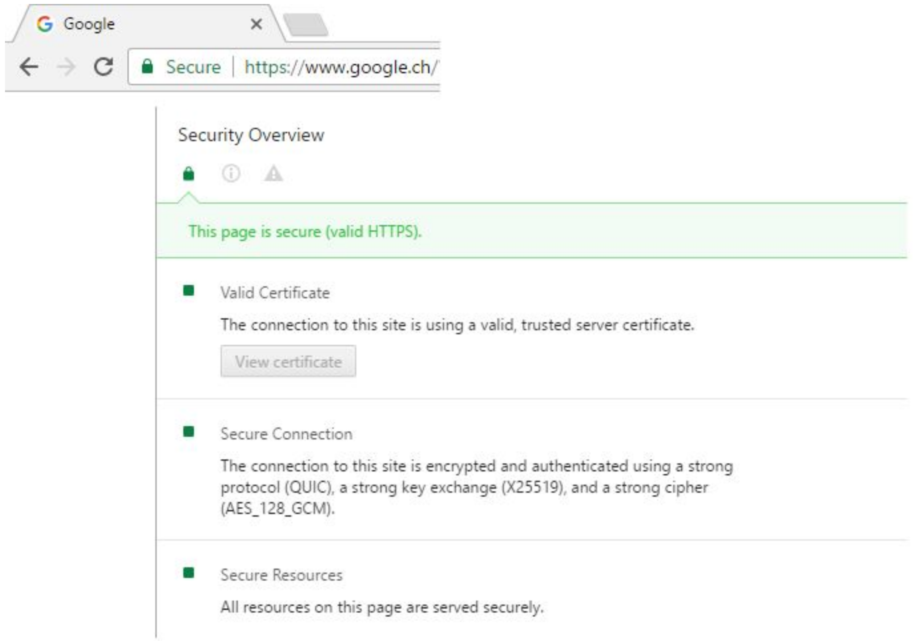
- POODLE, 2014. - SSL 3.0 vulnerability, padding oracle in CBC mode

Some Historical TLS Vulnerabilities

- CRIME, BREACH attacks - exploiting the vulnerability when data compression is used with TLS
- Bugs in Implementations:
  - 2014. Heartbleed in OpenSSL
  - 2017. Cloudbleed - a bug in Cloudflare HTML parser allowed people to read data in memory of programs running on the servers
- TLS attacks explained https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/

HTTPS

- HTTP over TLS, HTTP over SSL, HTTP Secure
- Communication over the HTTP where connection is encrypted by TLS
- Provides the authentication (usually of the web server) and privacy and integrity of data exchanged between the user and the server



Security Overview

This page is secure (valid HTTPS).

Valid Certificate

The connection to this site is using a valid, trusted server certificate.

Secure Connection

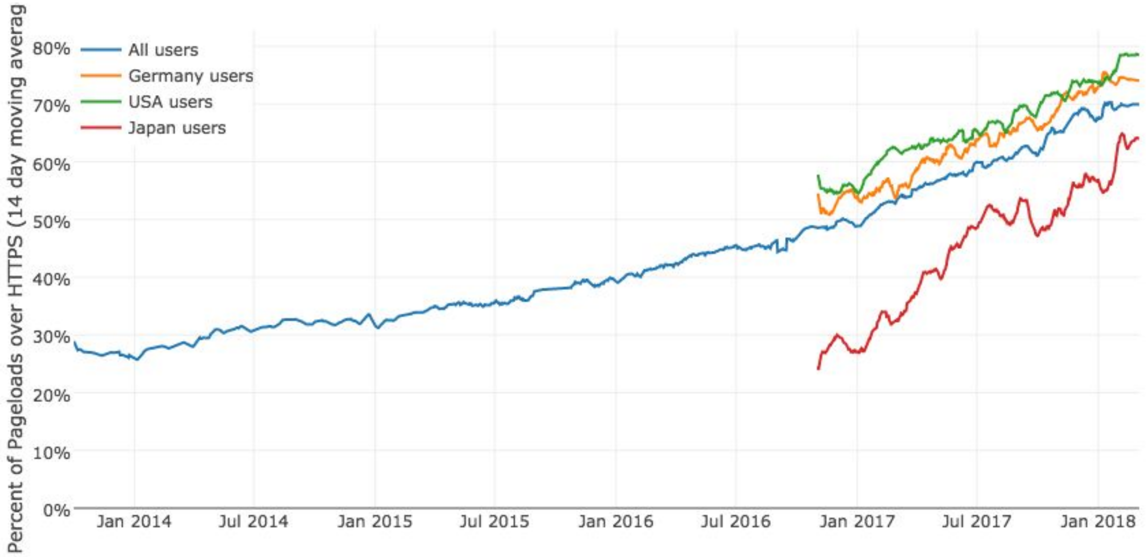
The connection to this site is encrypted and authenticated using a strong protocol (QUIC), a strong key exchange (X25519), and a strong cipher (AES_128_GCM).

Secure Resources

All resources on this page are served securely.

Applications of HTTPS

- Internet banking, payments
- E-commerce, e.g., Amazon.com
- Filling forms with personal/sensitive information
- Public Wi-Fi spots - anyone on the network can sniff packets



Percentage of Web Pages Loaded by Firefox Using HTTPS.
Source: [Let's Encrypt]

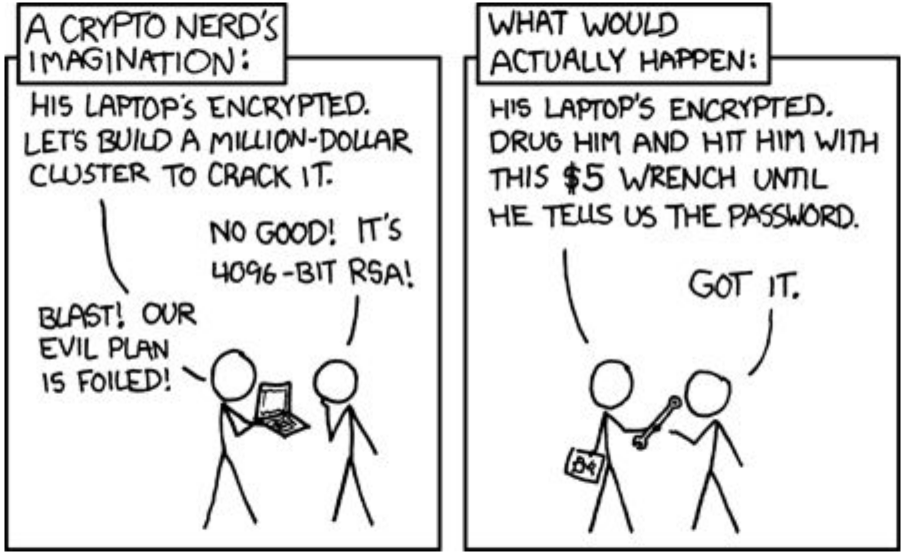
Attacks and Defenses 1/2

- Security of HTTPS relies on the security of TLS
- Web browsers have a pre-installed list of trusted Certificate Authorities
- CAs are a major concern and a potential point of failure to MiTM attacks:
    - e.g. French CA “Tresor public” issued fake certificates impersonating Google to the French government, in order to spy on government employees via MiTM attack
    - Google Certificate Transparency project tries to unveil problems quickly
- For a website to be secure, it must be completely hosted over HTTPS
    - This is a problem for many websites which earn from ads, because some of the ad-networks don’t support HTTPS
    - EPFL is still not 100% HTTPS-compliant

Attacks and Defenses 2/2

- SSL-stripping: MiTM attack which downgrades victim's web connection to the server from HTTPS to HTTP
    - Presented by Moxie Marlinspike at Black Hat 2009
    - User can see that the connection is insecure (over HTTP), but doesn’t know if the connection should be secure (over HTTPS)
    - Browser doesn’t warn about this, so the attack is quite effective
- HTTP Strict Transport Security (HSTS)
    - Policy mechanism used to protect against protocol downgrade attacks and cookie hijacking
    - Protects from SSL-stripping
    - HSTS tells the browser that the connection to the website should always be over HTTPS, so the browser can alert the user if SSL-stripping happened

Conclusion


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.