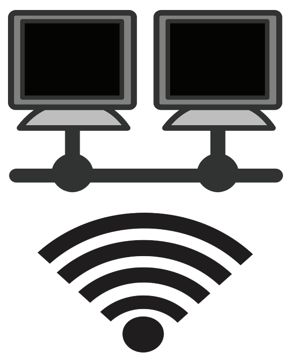
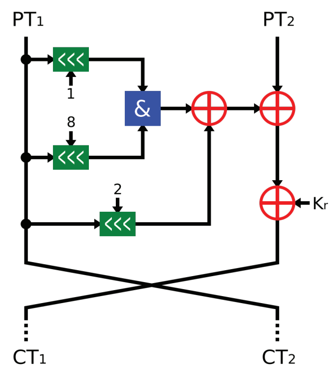
CSE 40567 / 60567: Computer Security





Cryptography 6

Homework #3 has been released. It is due 2/18 at 11:59PM

See **Assignments Page** on the course website for details

Cryptographic Protocols in the Wild

Cryptocurrencies

Goal: decentralize the management of the currency's integrity

Why?

Ensure total transparency with respect to costs, fees, and operations for all users

Need strong cryptography to do this

Application of digital signatures: Bitcoin Blockchain

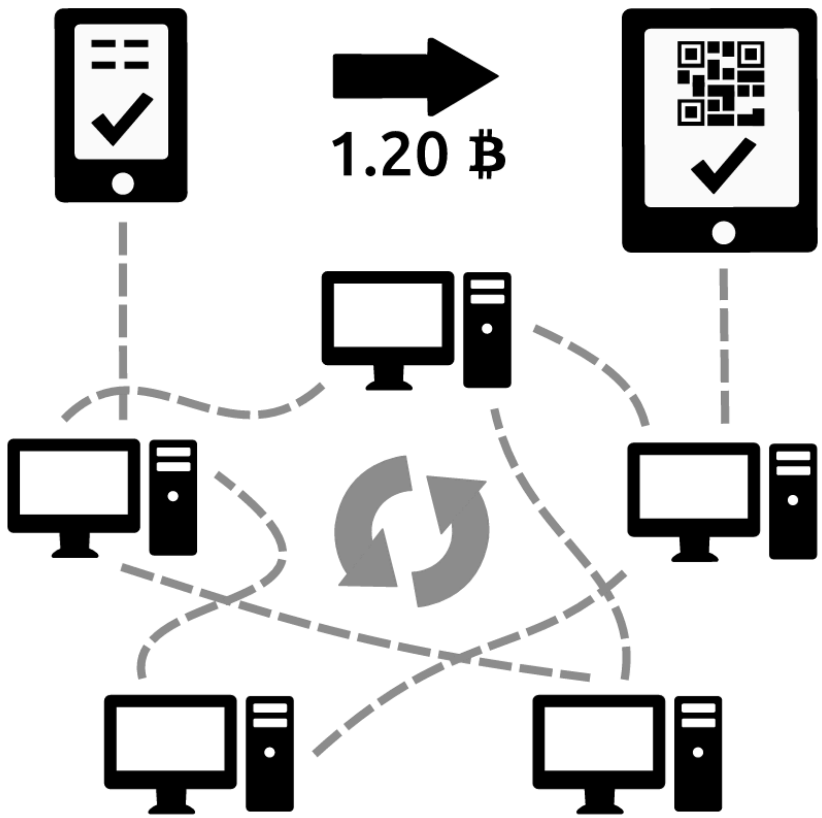


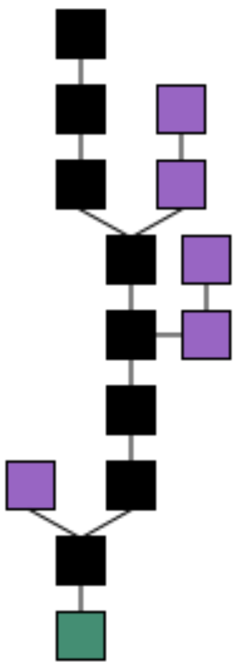
Image credit: Bitcoin Project

- The block chain is a shared public ledger
- All confirmed transactions are included in the block chain

Bitcoin Transactions

- A **transaction** is a transfer of value between bitcoin wallets that gets included in the blockchain
- Each bitcoin wallet has a private key used for signing transactions
  - The signature proves that the transaction came from the owner of the wallet
  - The signature also prevents the transaction from being altered after it has issued

Blocks



Network block formation BY-SA 3.0 Theymos

- Small sets of recorded transactions
- Each block contains a SHA-256 hash of the previous transaction, thus creating the chain
- Blocks are computationally hard to create

Blockchain security

What can an attacker do?

- Refuse to relay valid transactions to other nodes
- Attempt to create blocks that include or exclude specific transactions at will
- Attempt to create a 'longer chain' of blocks that make previously accepted blocks become 'orphans'

https://bitsonblocks.net/2015/09/09/a-gentle-introduction-to-blockchain-technology/

Blockchain security

What can't an attacker do?

- Create bitcoins outside of the legitimate mining process
  - Technically, this is possible, but will be rejected by all other nodes on the network
- Steal bitcoins from another user's account
- Make payments on a user's behalf or attempt to masquerade as another user

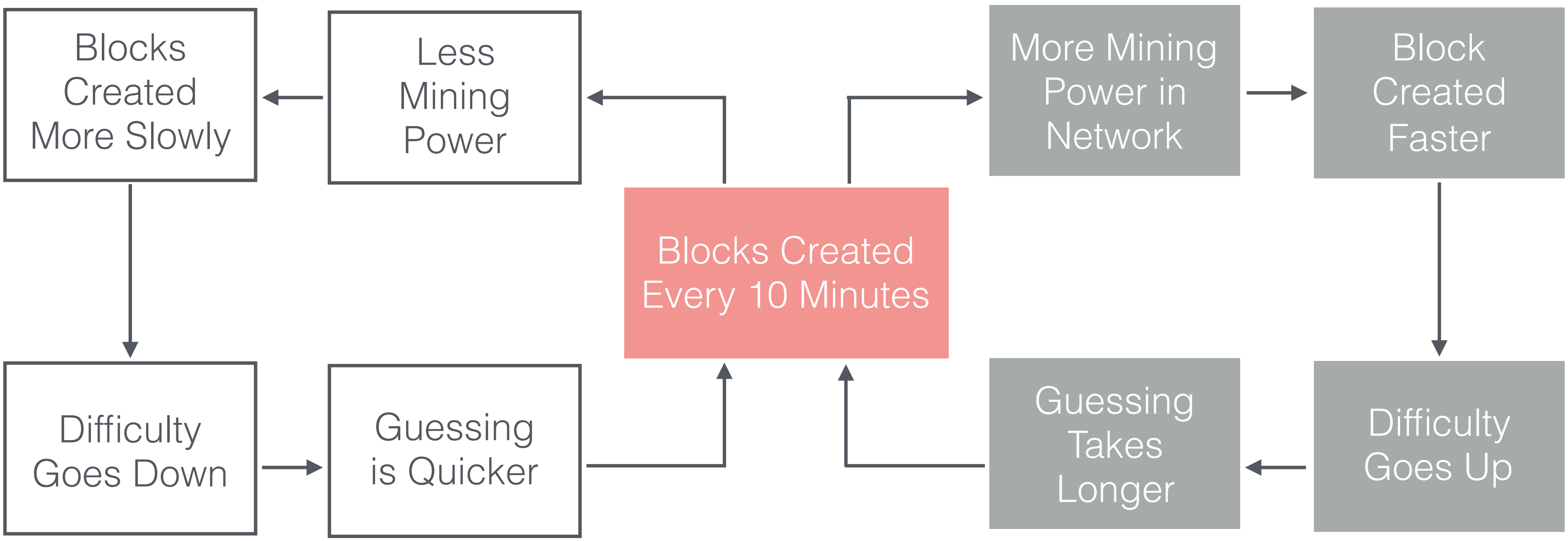
Assume an attacker has created a bogus block

Defense: make it computationally expensive (and thus financially expensive) to add blocks

General strategy: creators of blocks need to guess a number

- Number and block contents lead to a hash that is smaller than a separate chosen number
- The chosen number is related to the current processing power of the bitcoin network
- Guessing becomes more difficult as more computers join

Proof of Work



Block Hashing Algorithm

Field	Purpose	Updated when...	Size (Bytes)
Version	Block version number	You upgrade the software and it specifies a new version	4
hashPrevBlock	256-bit hash of the previous block header	A new block comes in	32
hashMerkleRoot	256-bit hash based on all of the transactions in the block	A transaction is accepted	32
Time	Current timestamp as seconds since 1970-01-01T00:00 UTC	Every few seconds	4
Bits	Current target in compact format	The difficulty is adjusted	4
Nonce	32-bit number (starts at 0)	A hash is tried (increments)	4

When mining, the algorithm repeatedly hashes the block header while incrementing the nonce field.

Incrementing the nonce field entails recomputing the merkle tree (i.e., tree of hashes)

https://en.bitcoin.it/wiki/Block_hashing_algorithm

Anatomy of a block

Block #125552

BlockHash 00000000000000001e8d6829a8a21adc5d38d0a473b144b6765798e61f98bd1d

Summary

Number Of Transactions	4
Height	125552 (Mainchain)
Block Reward	50 BTC
Timestamp	May 21, 2011 1:26:31 PM
Mined by	
Merkle Root	2b12fcf1b09288fcaff797d71e950e...
Previous Block	125551

Difficulty	244112.48777433
Bits	1a44b9f2
Size (bytes)	1496
Version	1
Nonce	2504433986
Next Block	125553

https://blockexplorer.com

Transaction details

Transaction

Transaction 51d37bdd871c9e1f4d5541be67a6ab625e32028744d7d4609d0c37747b40cd2d

Summary

Size	135 (bytes)
Received Time	May 21, 2011 1:26:31 PM
Mined Time	May 21, 2011 1:26:31 PM
Included in Block	00000000000000001e8d6829a8a21adc5d38d0a473b144b6765798e61f98bd1d
Coinbase	04f2b9441a022a01

Details

51d37bdd871c9e1f4d5541be67a6ab625e32028744d7d4609d0c37747b40cd2d mined May 21, 2011 1:26:31 PM

No Inputs (Newly Generated Coins) → 15nNvBTUdMaiZ6d3GWCeXFu2MagXL3XM1q 50.01 BTC (S)

320148 CONFIRMATIONS 50.01 BTC

Assume an attacker has modified a block

Defense: validate the cryptographic fields of the block w.r.t. to other blocks in the chain

Example: **Bitcoin address generated using owner's private key**

Address 0 BTC

Address 1HuppjXz7dPrt2a67LqacDW5T4VanFrpqC

Summary confirmed

Total Received	29.5 BTC
Total Sent	29.5 BTC
Final Balance	0 BTC
No. Transactions	2

Signal Protocol

Designed by Open Whisper Systems

https://whispersystems.org/

Widely deployed via the following apps:

Signal

WhatsApp

Facebook Messenger

Google Allo

Signal’s crypto components

Protocol Elements:

Double Ratchet Algorithm

Prekeys

Triple Diffie-Hellman handshake

Cryptographic Primitives:

Curve25519

AES-256

SHA-256

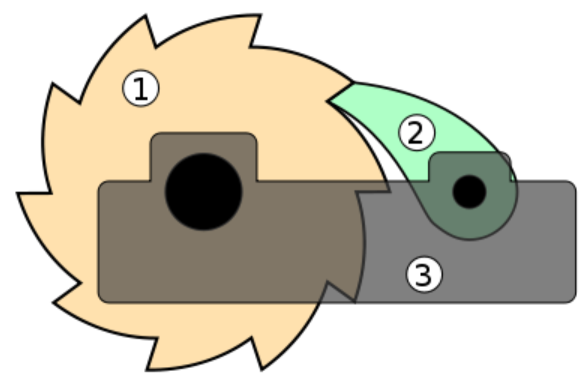
Double Ratchet Algorithm

Developed by Trevor Perrin and Moxie Marlinspike (2013)

Specifically designed for instant messaging

Goal: after initial key exchange, manage short-lived session keys

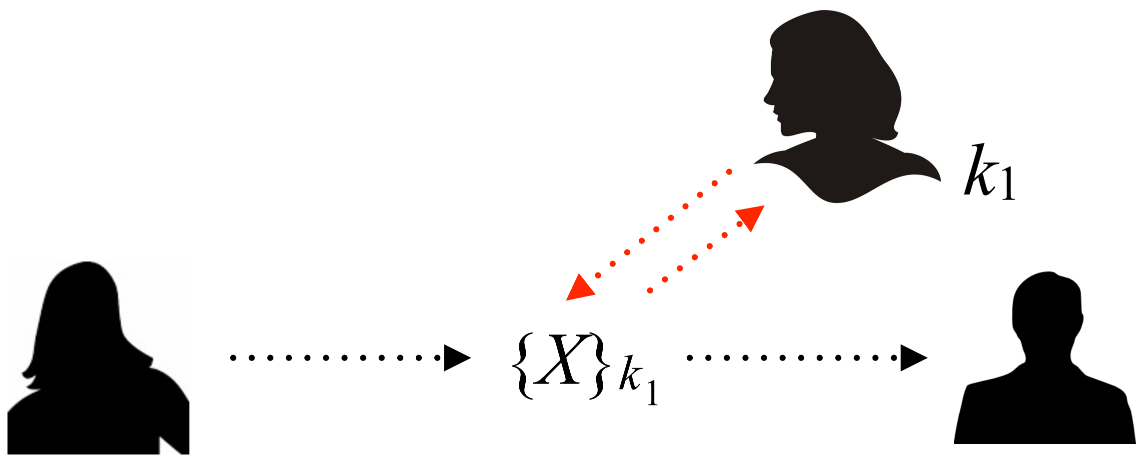
- A cryptographic ratchet is a function that only moves forward
- With a prior state value, all future values can be computed
- Impossible to calculate an older value from any values beyond it



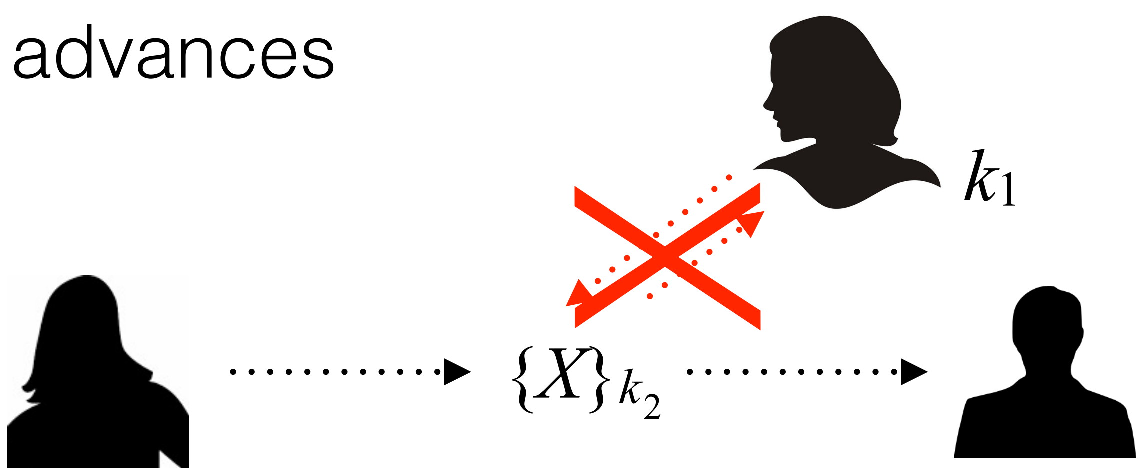
Drawing of a ratchet

https://whispersystems.org/blog/advanced-ratcheting/

What happens when Mallory steals a session key?



Ratchet advances



“Self-Healing”

How the primitives map to the algorithm

DH ratchet:
Elliptic curve Diffie–Hellman (ECDH) with Curve25519

Message authentication codes (MAC, authentication):
Keyed-Hash Message Authentication Code (HMAC) based on SHA-256

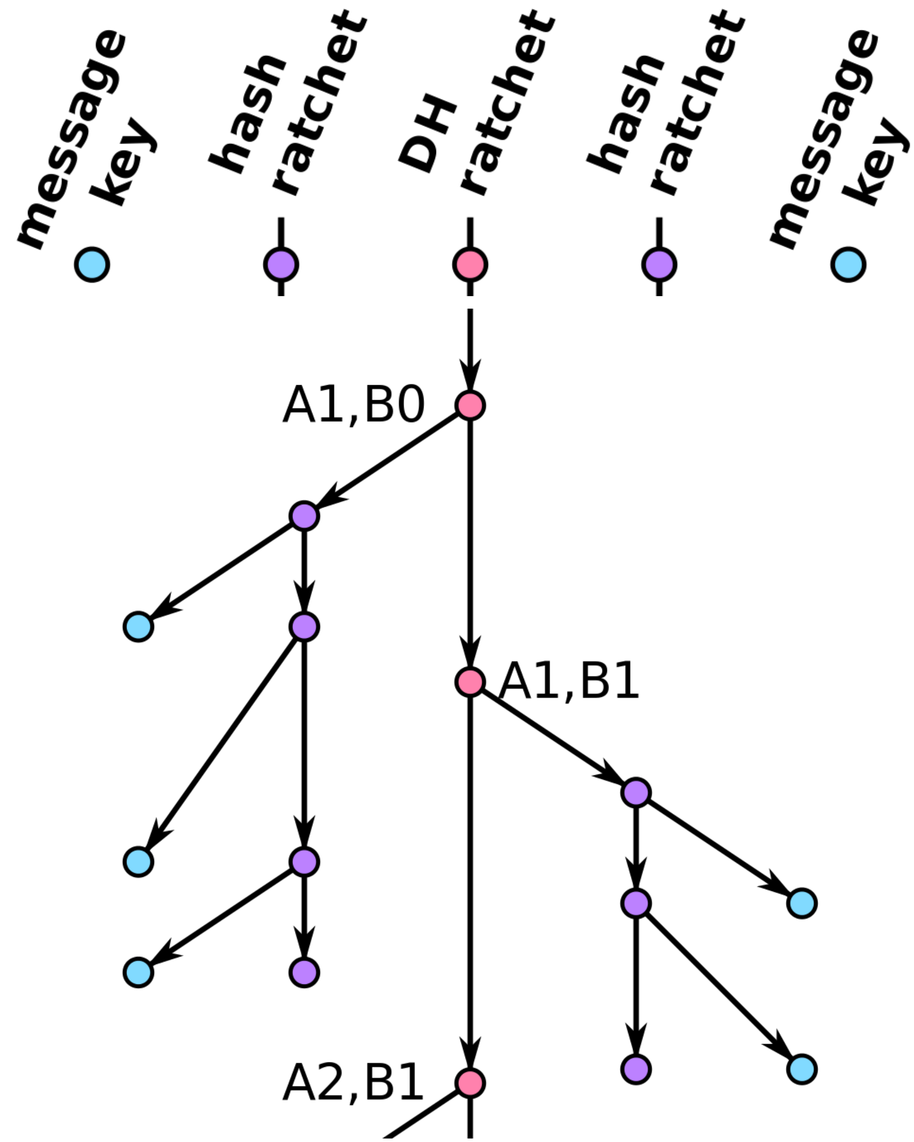
Symmetric encryption:
AES, partially in Cipher Block Chaining mode (CBC) with padding as per PKCS #5 and partially in Counter mode (CTR) without padding

Hash ratchet:
HMAC

Double Ratchet Algorithm

- Client advances one of two hash ratchets (one for sending, one for receiving)
    - Both are seeded with a common secret from a DH ratchet
- Continually provide the remote host with a new public DH value and advance the DH ratchet whenever a new DH value from the remote host arrives
- As soon as a new common secret is established, a new hash ratchet is initialized

Double Ratchet Algorithm



Zero Knowledge Proofs

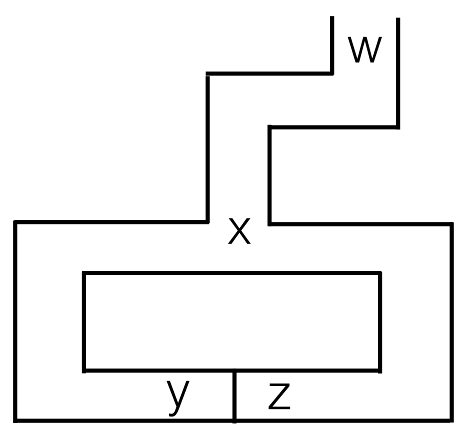
“I know that you know something, without knowing what that something is”

- The usual way for Alice to prove something to Bob is to tell him what that something is.
    - But then he knows it
    - And he can tell others
    - Alice cannot prevent this from happening once she divulges her secret

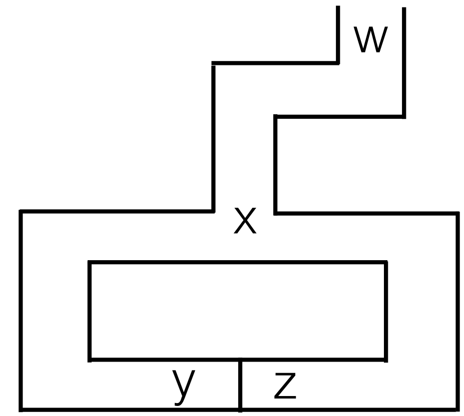
Zero-knowledge proofs: using one-way functions, Alice can prove to Bob that she knows something without divulging it

Basic protocol

- Assume the following (somewhat ridiculous) scenario:
    - There is a cave, which contains a secret
    - Someone who knows the magic words can open the secret door between y and z
    - For everyone else, both passages lead to a dead end
    - Alice knows the secret, and wants to prove this to Bob, without telling him what the secret is



Basic Protocol



1. Bob stands at point w
2. Alice walks to either point y or z
3. After Alice disappears into the Cave, Bob walks to point x
4. Bob shouts to Alice, asking her to either:
   a. come out of the left passage, or
   b. come out of the right passage
5. Alice complies, using the magic words to open the secret door if she has to
6. Alice and Bob repeat steps 1 through 5 n times

Why does this work?

- There is no way Alice can repeatedly guess which side Bob will ask her to come out of
    - She has a 50% chance of fooling him in one round
    - A 25% chance of fooling him in two rounds
    - A 1 in 65,536 chance of fooling him in 16 rounds
    - **A 1 in 2^n chance of fooling him in all rounds**

A more realistic protocol

1. Alice uses her information and a random number to transform the hard problem into another hard problem, one that is *isomorphic* to the original
    - She then solves this new instance of the hard problem
2. Alice commits to the solution of the new instance, using a bit-commitment scheme
3. Alice reveals the new instance to Bob. He cannot use this new problem to get any information about the original instance or its solution

A more realistic protocol

4. Bob asks Alice to either:

    a. prove to him that the old and new instances are isomorphic, or

    b. open the solution she committed to in step 2 and prove that it is a solution to the new instance

5. Alice complies.

6. Alice and Bob repeats steps 1 through 5 *n* times

Hard problems that can be used for zero knowledge proofs

- Graph Isomorphism
  - If two graphs are identical except for the names of the points, they are isomorphic
  - For a large graph, finding whether two graphs are isomorphic is an NP-complete problem
- Hamiltonian Cycles
  - e.g., Alice knows a circular, continuous path along the lines of a graph that passes through each point exactly once
  - This is another computationally hard problem
  - Bob can know the graph, but not its Hamiltonian Cycle

Applications

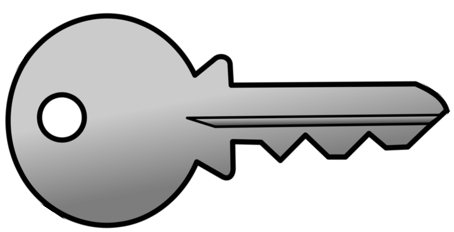
- Electronic voting systems
- Digital signatures
- Cryptocurrency
- Nuclear disarmament discussions

Not commonly used because trust violations in technology are rampant

Public Key Infrastructure

Where do we get a key from?

- We've mostly assumed that actors have had access to needed public keys
  - Keys need to come from a (trusted?) source
  - Distribution should be public
  - The basic public key protocols don't tell us how to do this



Answer: PKI

Digital Certificates

▼ **Details**

Subject Name	
Country	US
Postal Code	46556
State/Province	IN
Locality	Notre Dame
Street Address	Information Technology Center
Organization	University of Notre Dame
Organizational Unit	EIS
Common Name	nps1-prod-v.cc.nd.edu
Issuer Name	
Country	US
Organization	Internet2
Organizational Unit	InCommon
Common Name	InCommon Server CA
Serial Number	38 3C 92 27 DE 47 B8 39 D9 B4 A8 FC EC 5E F8 2A
Version	3
Signature Algorithm	SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)
Parameters	none
Not Valid Before	Sunday, August 25, 2013 at 8:00:00 PM Eastern Daylight Time
Not Valid After	Thursday, August 25, 2016 at 7:59:59 PM Eastern Daylight Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	none
Public Key	256 bytes : AF D7 E5 BF 97 0D 63 A0 …
Exponent	65537
Key Size	2048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 3E D3 B4 24 2C 4D DE E8 …

Certificates are: a signed message

Certificates contain:

- Owner's Name
- Public Key
- Algorithm Identifiers

x.509 standard

RFC 5280

- Certificate
  - Version Number
  - Serial Number
  - Signature Algorithm ID
  - Issuer Name
  - Validity period
    - Not Before
    - Not After
  - Subject name
  - Subject Public Key Info
    - Public Key Algorithm
    - Subject Public Key
  - Issuer Unique Identifier (optional)
  - Subject Unique Identifier (optional)
  - Extensions (optional)
    - ...
- Certificate Signature Algorithm
- Certificate Signature

PKI

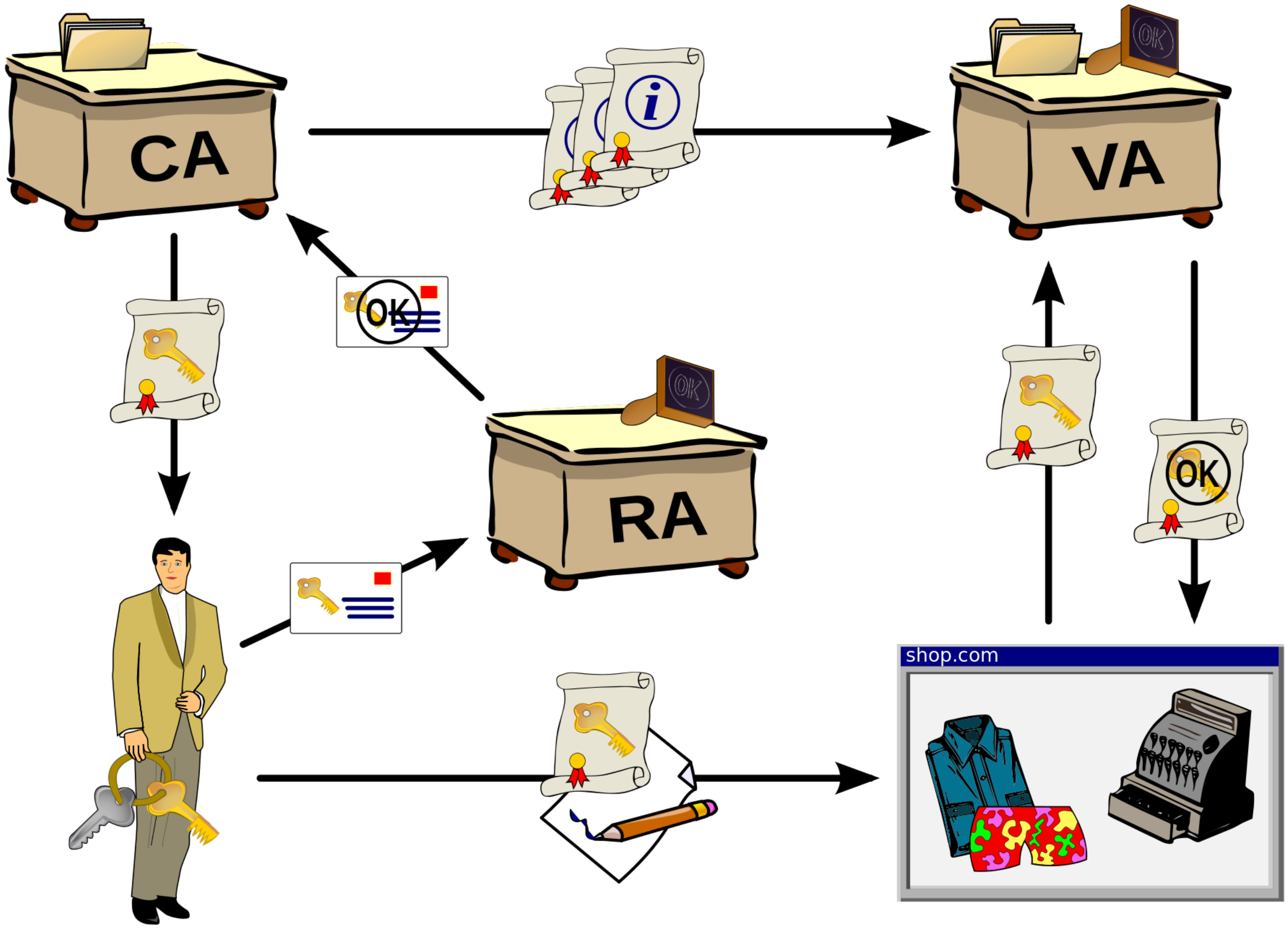
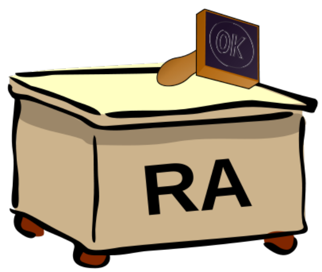


Diagram of Public Key Infrastructure

Registration Authority

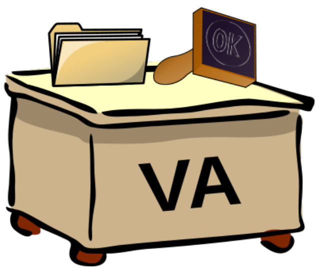


- Verifies the identities of users requesting certificates
- Tells CA to issue certificate if user is validated

RFC 4210:

"The functions that the registration authority may carry out will vary from case to case but MAY include personal authentication, token distribution, revocation reporting, name assignment, key generation, archival of key pairs, et cetera."

Validation Authority



- Verifies the digital certificate of a user
- Serves as a trusted third party

Q. Why use a VA?

A. Facilitates scalability by providing clients with one point of access for CA discovery

PKI and Trust

When you use a certificate, you are relying on the trustworthiness of the issuer

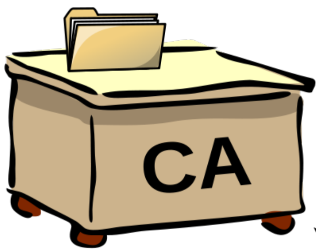
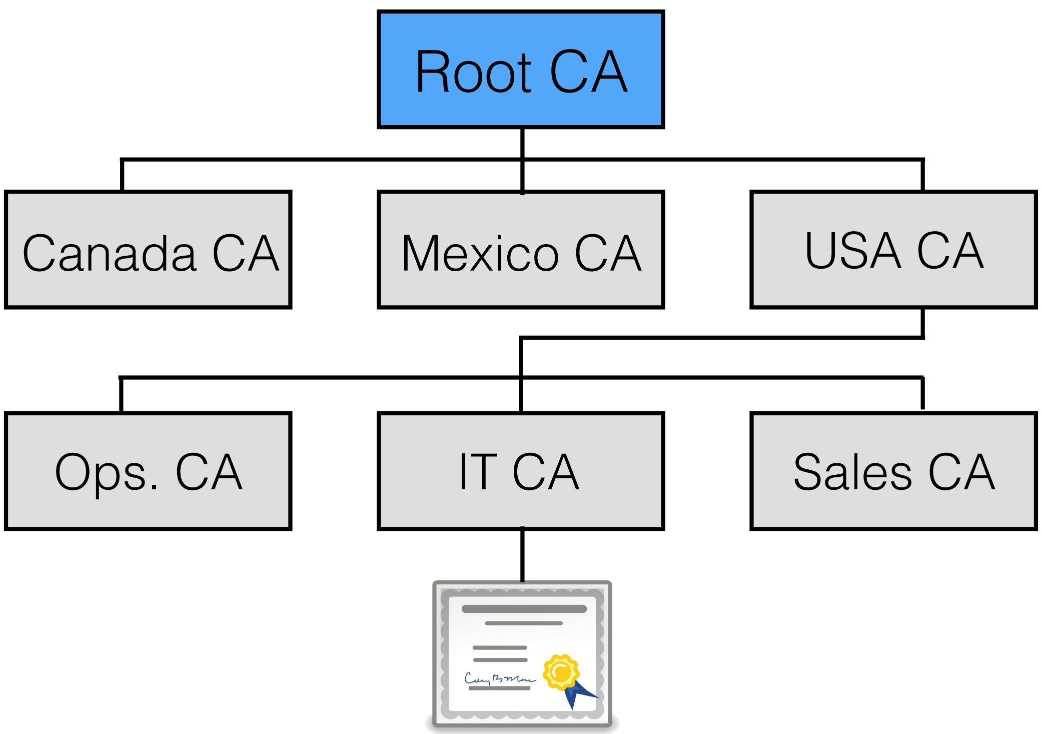


Diagram of Public Key Infrastructure (cc) BY-SA 3.0 Chrkl

- CA issues and signs certificates
- Certificates may be for end users
- Or, they may be for **sub-CAs**



Gnome Application Certificate (cc) BY-SA 3.0 GNOME icon artists

CAs and Sub-CAs

- Web browsers come with a large set of CAs built in
  - Your vendor trusts them, but do you?
  - Are they honest?
  - Competent?
  - Does their threat model match yours?
- What is a sub-CA allowed to do?
  - Does it issue certificates for its own jurisdiction, or for *any* domain?
- What authority, if any, is embodied by a CA?

How many CAs does your browser trust?

AAA Certificate Services
Root certificate authority
Expires: Sunday, December 31, 2028 at 6:59:59 PM Eastern Standard Time
This certificate is valid

Name	Kind	Expires	Keychain
AAA Certificate Services	certificate	Dec 31, 2028, 6:59:59 PM	System Roots
Actalis Authentication Root CA	certificate	Sep 22, 2030, 7:22:02 AM	System Roots
AddTrust Class 1 CA Root	certificate	May 30, 2020, 6:38:31 AM	System Roots
AddTrust External CA Root	certificate	May 30, 2020, 6:48:38 AM	System Roots
AddTrust Public CA Root	certificate	May 30, 2020, 6:41:50 AM	System Roots
AddTrust Qualified CA Root	certificate	May 30, 2020, 6:44:50 AM	System Roots
Admin-Root-CA	certificate	Nov 10, 2021, 2:51:07 AM	System Roots
AdminCA-CD-T01	certificate	Jan 25, 2016, 7:36:19 AM	System Roots
AffirmTrust Commercial	certificate	Dec 31, 2030, 9:06:06 AM	System Roots
AffirmTrust Networking	certificate	Dec 31, 2030, 9:08:24 AM	System Roots
AffirmTrust Premium	certificate	Dec 31, 2040, 9:10:36 AM	System Roots
AffirmTrust Premium ECC	certificate	Dec 31, 2040, 9:20:24 AM	System Roots
America Onli...cation Authority 1	certificate	Nov 19, 2037, 3:43:00 PM	System Roots
America Onli...cation Authority 2	certificate	Sep 29, 2037, 10:08:00 AM	System Roots
ANF Global Root CA	certificate	Jun 5, 2033, 1:45:38 PM	System Roots
Apple Root CA	certificate	Feb 9, 2035, 4:40:36 PM	System Roots
Apple Root CA - G2	certificate	Apr 30, 2039, 2:10:09 PM	System Roots
Apple Root CA - G3	certificate	Apr 30, 2039, 2:19:06 PM	System Roots
Apple Root Certificate Authority	certificate	Feb 9, 2025, 7:18:14 PM	System Roots

- Chrome: 198 CAs
- IE: 320 CAs
- Firefox: 150 CAs

Lifetime of keys

Certificates expire at some set point. There are three reasons for this:

1. Sense that after some time, the likelihood of compromise is unacceptably high
    - Unclear what that time period is
2. Algorithms age
    - Recall our discussions about md5 and 1024-bit public / private key pairs
3. Certificates expire to ease bookkeeping with respect to revocations
    - No need to keep track of the revocation status of an expired certificate (maybe)

Certificate Revocation

1. Private key compromise
    - Example: CA DigiNotar was hacked in 2011, allegedly by government sponsored actors (*whose* government remains an open question)
2. Suspected or actual misbehavior by the holder of the private key
    - Example: CA certifies a phishing site, thus side-stepping security warnings about a suspicious certificate
3. Cryptographic algorithm is broken or used with an insufficient key size
    - Example: md5

Certificate revocation procedures

A client accepting a certificate can check for revocation in two ways:

1. The original mechanism places the revoked certificate on the Certificate Revocation List (CRL)
   - File of revoked certificates signed by the issuing CA
   - URL of this list is included in each certificate
   - Client checks if the current cert. is on the list before accepting it

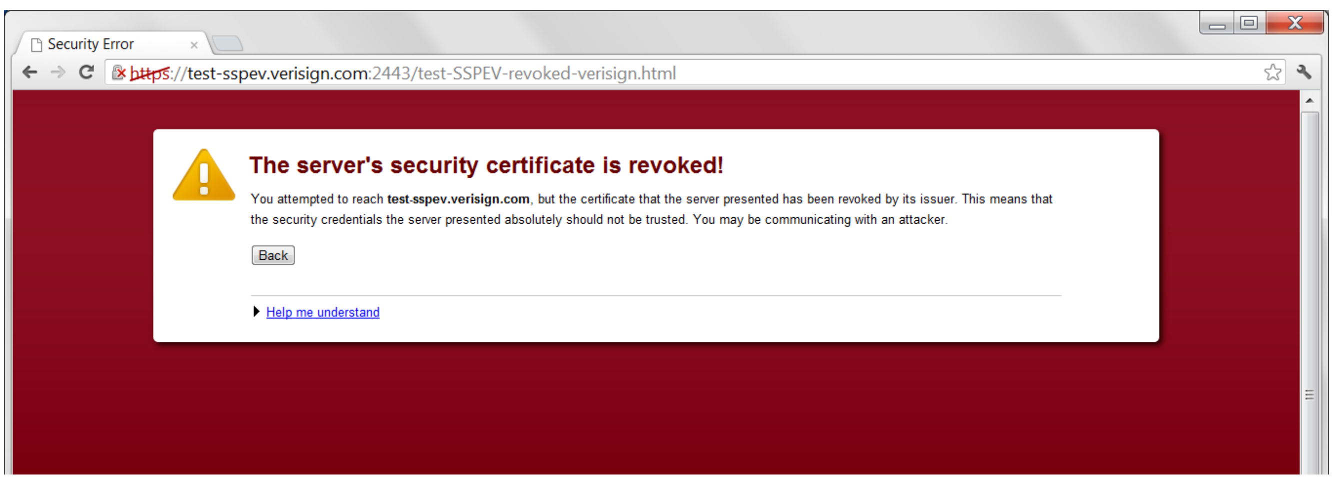


Image credit: http://unmitigatedrisk.com/?p=222

Certificate revocation procedures

2. Online Certificate Status Protocol (OCSP)

   - Verifies the continuing validity of a certificate, rather than whether it was ever valid
   - Returns Valid, Invalid, or Unknown status codes
     - What happens if a server returns Unknown?
   - Perceived advantage over CRL: reduces time between compromise and revocation

Key continuity failure messages

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
c9:d7:93:67:29:03:fd:6f:a9:ba:2e:58:63:34:fd:3a.
Please contact your system administrator.
Add correct host key in /home/wjs3/.ssh/known_hosts to get rid of this message.
Offending key in /home/wjs3/.ssh/known_hosts:6
RSA host key for alfred has changed and you have requested strict checking.
Host key verification failed.
```

How often have you ignored messages like this?

Users should not become accustomed to clicking through such messages

What could be better about PKI?

Biggest problem: too many CAs, very little oversight

Possible Solutions

- Certificate transparency: every CA logs all of the certificates it issues (Google proposal)
  - Easy to detect multiple certificates for the same site
  - Problem: Requires universal buy-in
- DNS-based Authentication of Named Entities (DANE)
  - Allows x.509 certificates to be bound to DNS names using Domain Name System Security Extensions (DNSSEC)
  - Problem: an attacker with access to a DNS server can replace certificates

What could be better about PKI?

- Usability issue: browser warnings are routinely ignored
  - Training users helps somewhat, but only goes so far
  - Need better protocols and UI

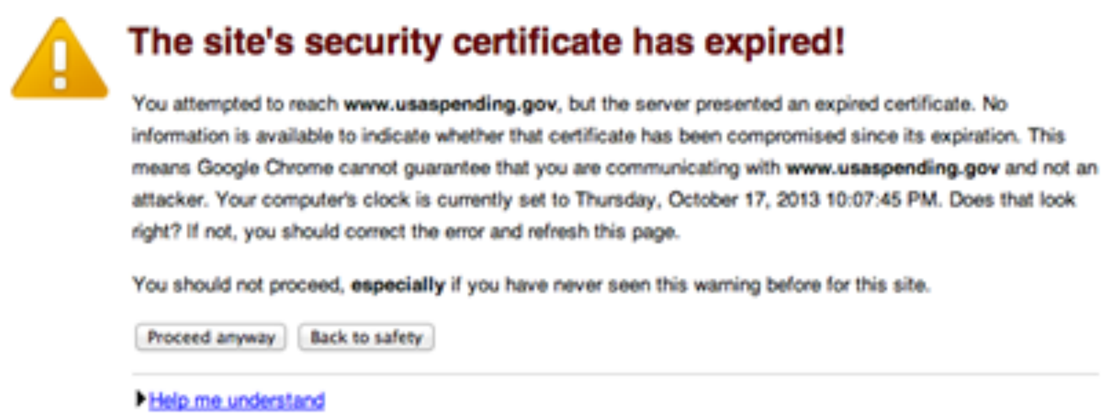


Image Credit: http://www.itnews.com.au/news/us-govt-left-vulnerable-by-expired-ssl-certs-360936

Cryptanalysis and Brute Force Attacks

Forms of cryptanalysis

- Known-plaintext attack
  - Some portion of the plaintext for the given ciphertext is known
    - Typically not useful these days
- Chosen-plaintext attack
  - Any plaintext can be encrypted with a given cryptosystem and key, but the (private) key itself cannot be analyzed
- Related-key attack
  - Observe operation of cipher under several different keys whose values are initially unknown, but a relationship between the keys can be discerned

Forms of cryptanalysis

- Ciphertext-only attack
  - No access to other information directly involved in the cryptosystem, but might leverage some information related to the plaintext
    - The statistics of the language the plaintext is written in
- Side-channel attack
  - Information gained from the physical implementation of a cryptosystem, rather than brute force attack or theoretical weaknesses in the algorithm(s)

WEP Attack

- Classic example of Related Key attack
- Client adapters and APs share WEP key; encryption provided by RC4, a stream cipher
  - Same key can't be used twice
  - WEP includes a 24-bit IV in each packet
  - RC4 key for a packet is IV concatenated with the WEP key

Wireless icon (cc) BY-SA 3.0 RRZEicons

S. Fluhrer, I. Mantin, A. Shamir, SAC, 2001

WEP Attack

Protocol weakness: WEP keys need to be changed manually, which happens infrequently

Attacker assumes that the same WEP key is used to encrypt all packets

- 24-bit IV means ~17M possibilities
- Birthday Paradox leads to a 50% chance of two packets out of every ~5000 sharing the same IV
    - 99% chance after ~12,500 packets
- Once IV is known, attacker can work backwards to recover the WEP key (assuming some known plaintext)

Many other attacks followed… Implemented in aircrack-ng: http://www.aircrack-ng.org/

Bar Mitzvah Attack

- Related attack on RC4
- Invariance weakness
  - Preserves part of the state permutation process throughout the initialization process
  - When processed by the PRGA, determines the least significant bits of the allegedly pseudo-random output stream along a long prefix of the stream

http://www.imperva.com/docs/HII_Attacking_SSL_when_using_RC4.pdf

Bar Mitzvah Attack

L-shaped key pattern in RC4 keys

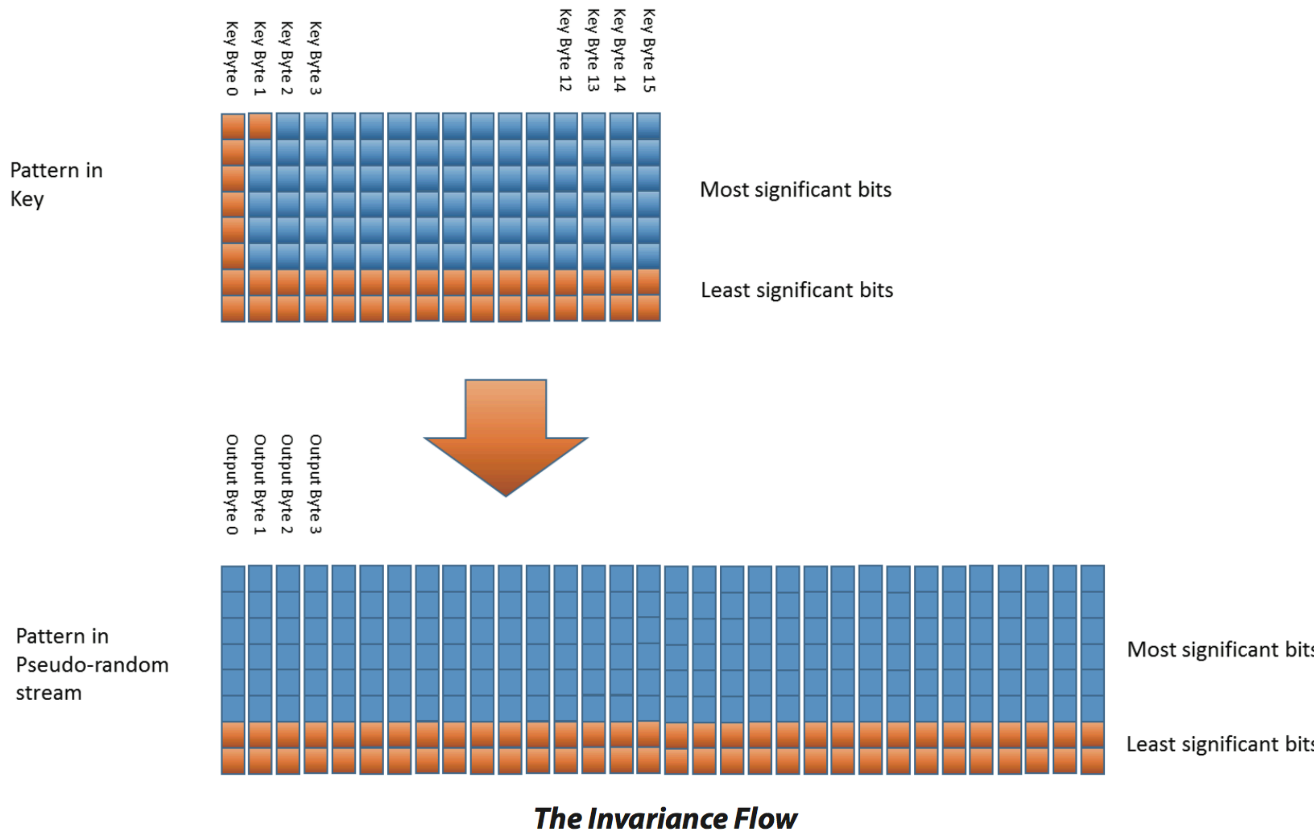


Image credit: Imperva

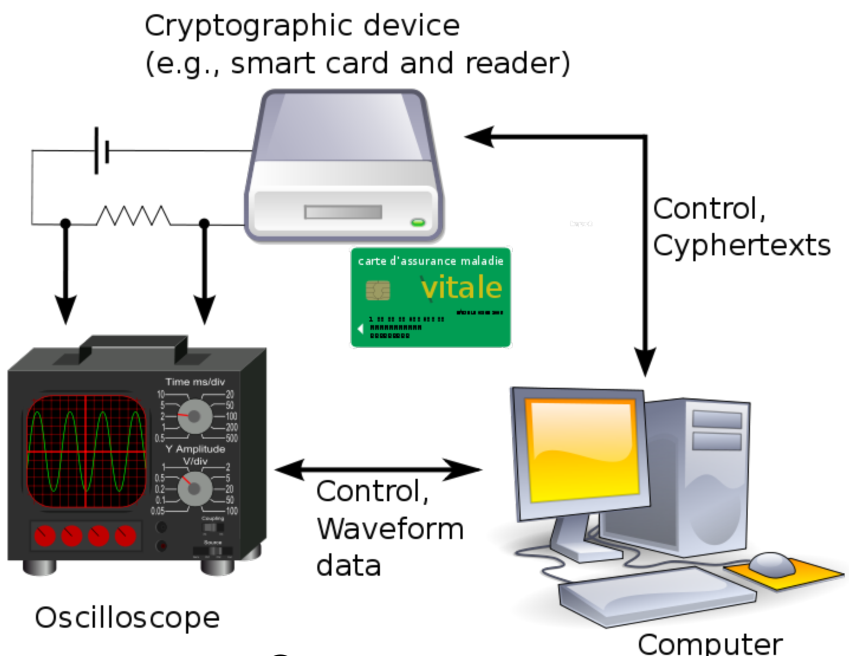
These biased stream bytes are XORed with the plaintext bytes, resulting in significant leakage of plaintext bytes from the ciphertext bytes

Bottom Line: **never use RC4**

Side-channel attacks

Differential Power Analysis

- Different instructions consume different amounts of power
- By measuring the power consumed by the smartcard chip, it may be possible to extract the key

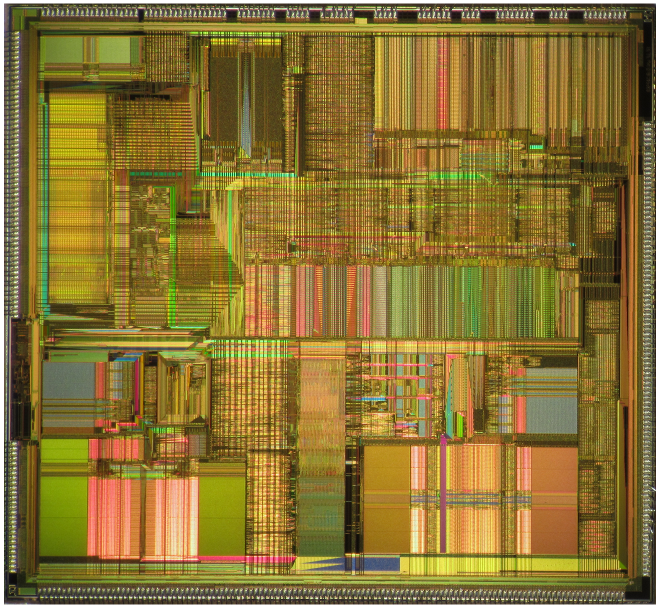


Differential power analysis

Side-channel attacks

Timing Attacks

- If cryptographic operations don't take the same number of clock cycles, they can leak key information
- Profiling cache misses can also reveal key information



Intel Pentium A80501 66MHz SX950 Die Image Pdesousa359

Recommendations for Cryptographic Primitives in 2019

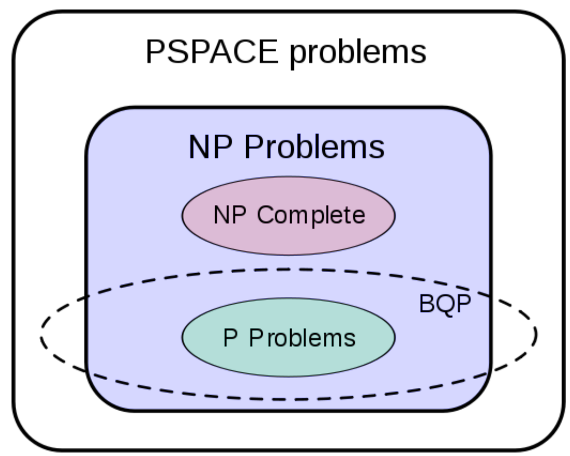
Purpose	Size (bits)
Symmetric cipher key length	128
RSA modulus	2,048
Elliptic curve modulus	256
Hash function (output)	256

Summary: What to use in 2019

Algorithm	**Function**
AES	Block Cipher
Counter mode AES	Stream Cipher
SHA-2-256/384/512 SHA-3-256/384/512	Hash Function
RSA or EC	Public Key Algorithm

Security frontier: Quantum Cryptography

- Canonical problem for quantum computing: prime number factorization
- Shor's algorithm: BQP problem making factoring and discrete logarithm computations easy
  - Given a sufficiently large quantum computer (~4,000 qubits for a 2048-bit key)
  - No such computer exists in 2019



P.W. Shor, "Algorithms for Quantum Computers," FOCS, 1994

Security frontier: Quantum Cryptography

Assuming large quantum computers appear in the next couple of decades, what's a straightforward replacement for public key cryptography?

Symmetric Key Cryptography + Kerberos