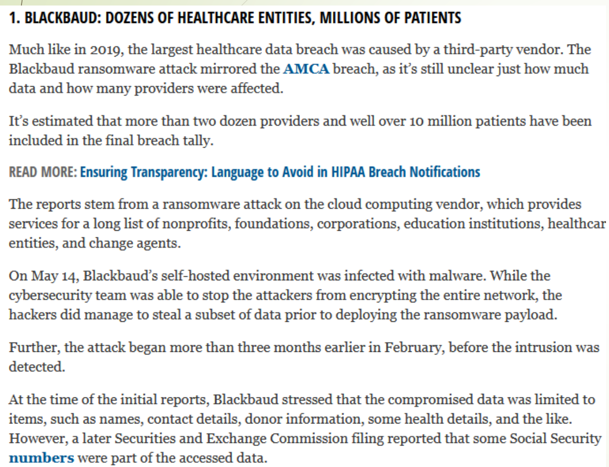
Biomedical Security

Erwin M. Bakker

Some Security News



1. BLACKBAUD: DOZENS OF HEALTHCARE ENTITIES, MILLIONS OF PATIENTS

Much like in 2019, the largest healthcare data breach was caused by a third-party vendor. The Blackbaud ransomware attack mirrored the **AMCA** breach, as it's still unclear just how much data and how many providers were affected.

It's estimated that more than two dozen providers and well over 10 million patients have been included in the final breach tally.

READ MORE: Ensuring Transparency: Language to Avoid in HIPAA Breach Notifications

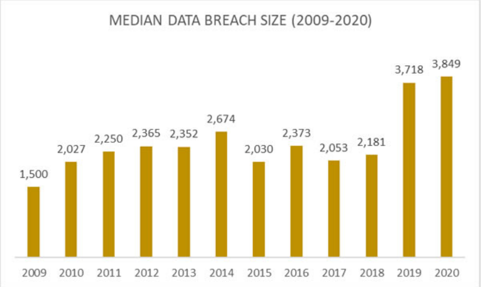
The reports stem from a ransomware attack on the cloud computing vendor, which provides services for a long list of nonprofits, foundations, corporations, education institutions, healthcar entities, and change agents.

On May 14, Blackbaud's self-hosted environment was infected with malware. While the cybersecurity team was able to stop the attackers from encrypting the entire network, the hackers did manage to steal a subset of data prior to deploying the ransomware payload.

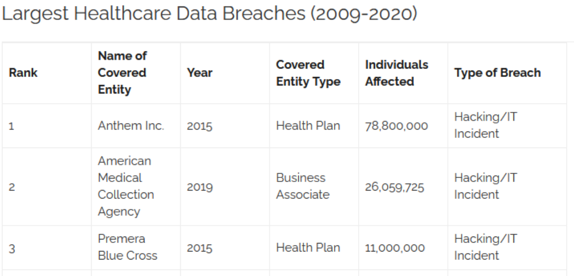
Further, the attack began more than three months earlier in February, before the intrusion was detected.

At the time of the initial reports, Blackbaud stressed that the compromised data was limited to items, such as names, contact details, donor information, some health details, and the like. However, a later Securities and Exchange Commission filing reported that some Social Security **numbers** were part of the accessed data.

https://healthitsecurity.com/news/the-10-biggest-healthcare-data-breaches-of-2020



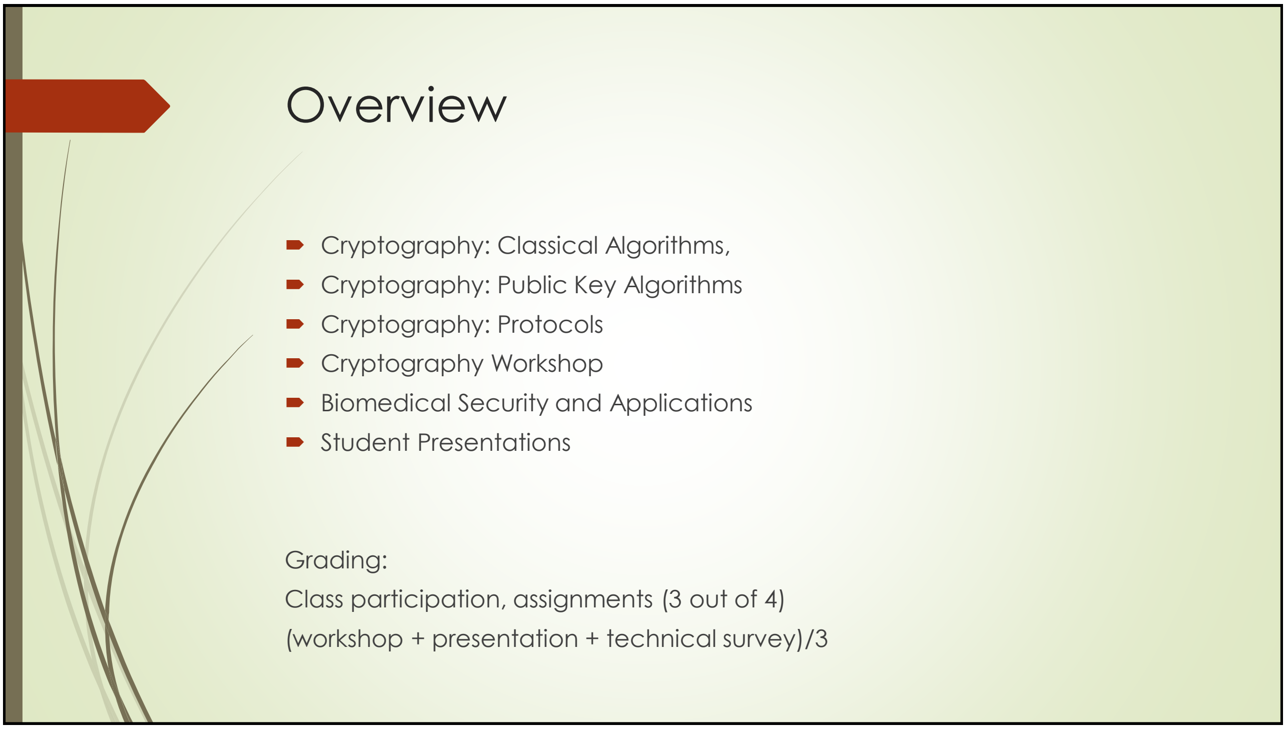
MEDIAN DATA BREACH SIZE (2009-2020)



Largest Healthcare Data Breaches (2009-2020)

Rank	Name of Covered Entity	Year	Covered Entity Type	Individuals Affected	Type of Breach
1	Anthem Inc.	2015	Health Plan	78,800,000	Hacking/IT Incident
2	American Medical Collection Agency	2019	Business Associate	26,059,725	Hacking/IT Incident
3	Premera Blue Cross	2015	Health Plan	11,000,000	Hacking/IT Incident

From: https://www.hipaajournal.com/healthcare-data-breach-statistics/



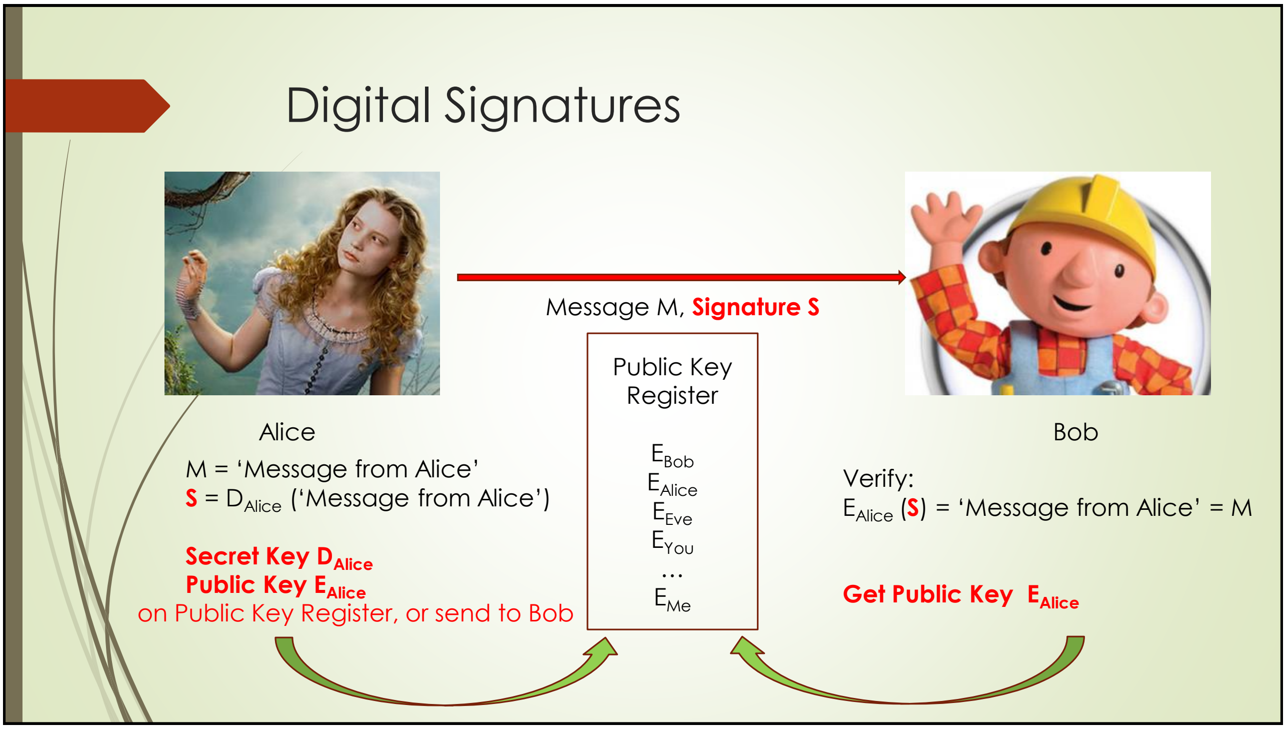
Overview

- Cryptography: Classical Algorithms,
- Cryptography: Public Key Algorithms
- Cryptography: Protocols
- Cryptography Workshop
- Biomedical Security and Applications
- Student Presentations

Grading:

Class participation, assignments (3 out of 4)

(workshop + presentation + technical survey)/3



Digital Signatures

Message M, **Signature S**

Public Key Register

E_{Bob}
E_{Alice}
E_{Eve}
E_{You}
…
E_{Me}

Alice

M = ‘Message from Alice’

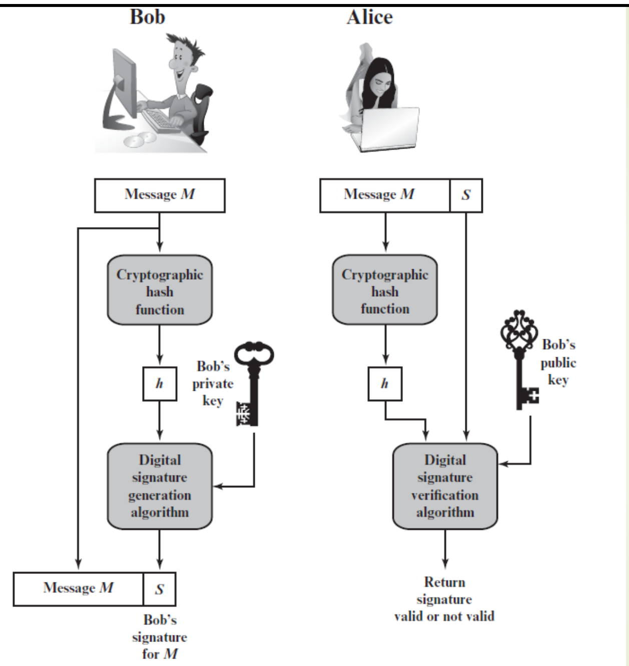
S = D_{Alice} (‘Message from Alice’)

Secret Key D_{Alice}
Public Key E_{Alice}
on Public Key Register, or send to Bob

Bob

Verify:
E_{Alice} (**S**) = ‘Message from Alice’ = M

Get Public Key E_{Alice}


Bob
Alice
Message M
Cryptographic hash function
h
Bob's private key
Digital signature generation algorithm
Message M
S
Bob's signature for M
Message M
S
Cryptographic hash function
h
Bob's public key
Digital signature verification algorithm
Return signature valid or not valid

(a) Bob signs a message

(b) Alice verifies the signature

Figure 13.1 Simplified Depiction of Essential Elements of Digital Signature Process

Public Key Crypto System ElGamal

- Public Key

  p a prime

  $g < p$

  $y = g^x \bmod p$

- Private Key

  $x < p$

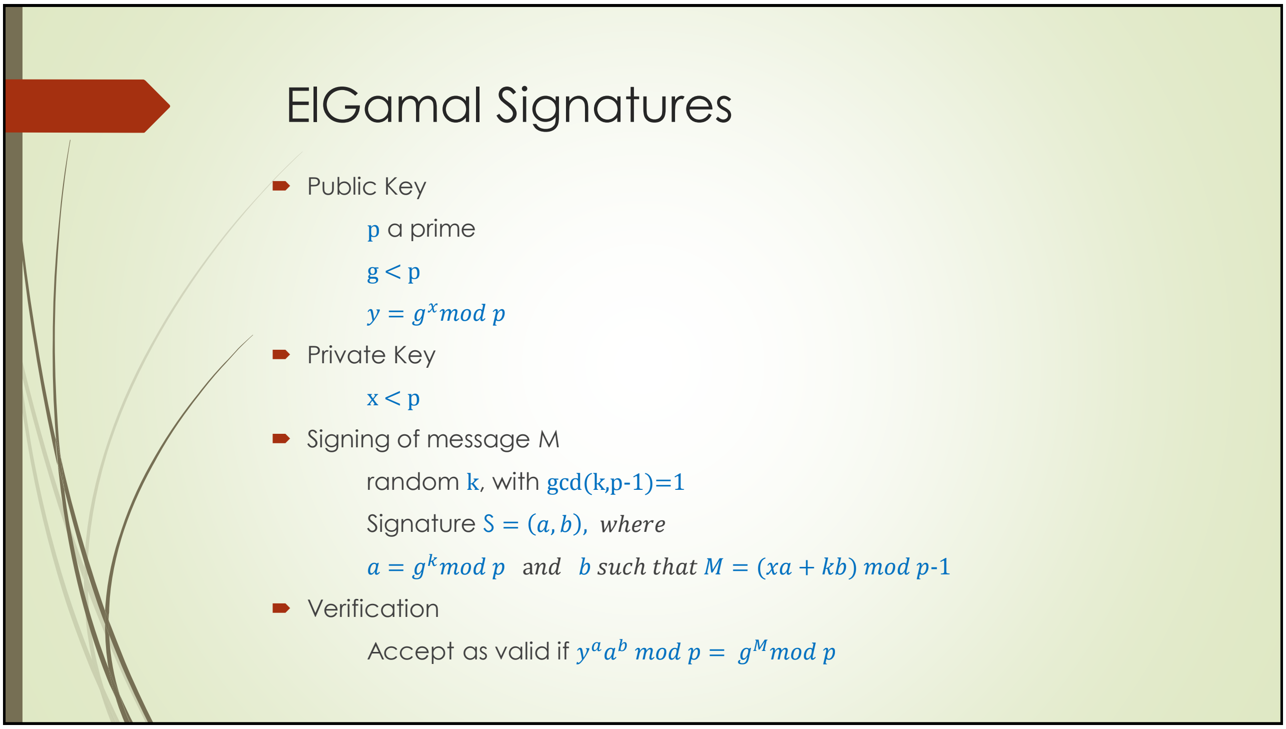
- Encryption of message M

  random k, with $\gcd(k, p-1)=1$

  $C = (a, b),\ where\ \ a = g^k \bmod p$ and $b = y^k M \bmod p$

- Decryption

  $M = b/a^x \bmod p$ (Note: $y^k M a^{-x} \bmod p = g^{kx} M g^{-xk} \bmod p = M$)



ElGamal Signatures

- Public Key

  p a prime

  $g < p$

  $y = g^x mod\ p$

- Private Key

  $x < p$

- Signing of message M

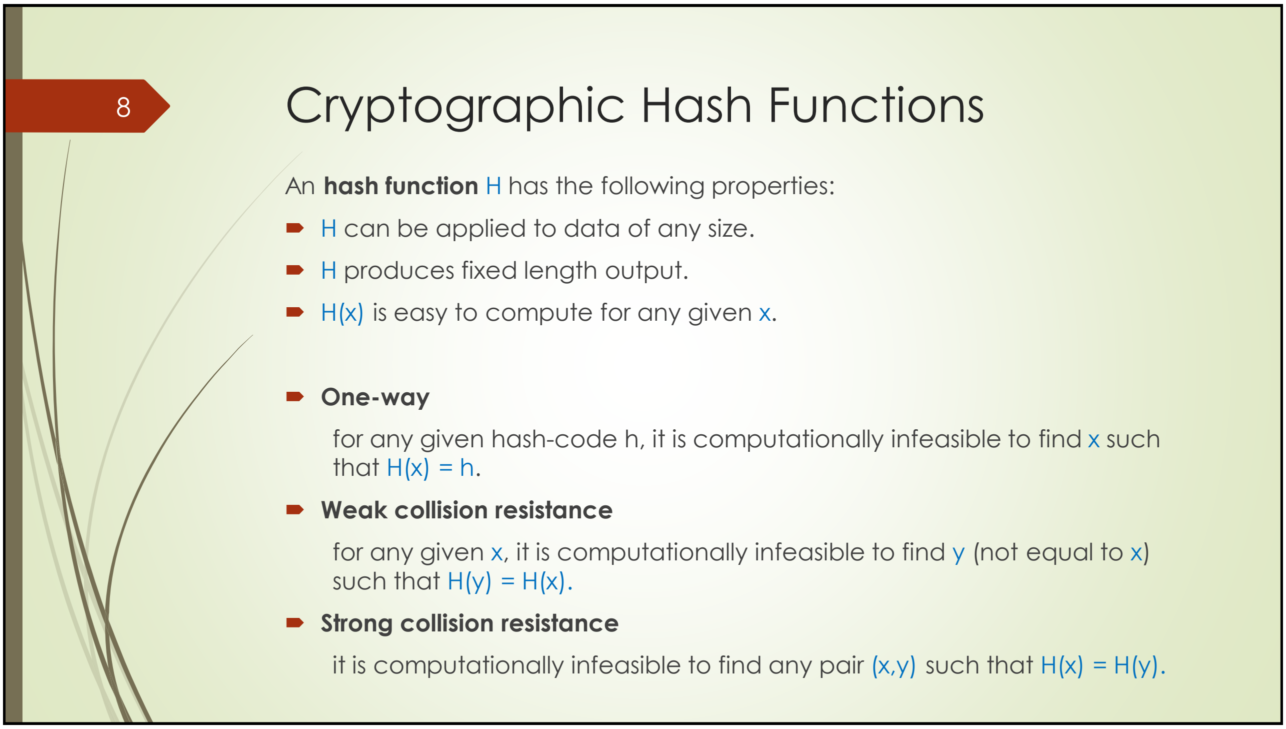
  random k, with $gcd(k,p\text{-}1)=1$

  Signature $S = (a, b),\ where$

  $a = g^k mod\ p \quad and \quad b\ such\ that\ M = (xa + kb)\ mod\ p\text{-}1$

- Verification

  Accept as valid if $y^a a^b\ mod\ p =\ g^M mod\ p$



8

Cryptographic Hash Functions

An **hash function** H has the following properties:

- H can be applied to data of any size.
- H produces fixed length output.
- H(x) is easy to compute for any given x.

- **One-way**

  for any given hash-code h, it is computationally infeasible to find x such that H(x) = h.

- **Weak collision resistance**

  for any given x, it is computationally infeasible to find y (not equal to x) such that H(y) = H(x).

- **Strong collision resistance**

  it is computationally infeasible to find any pair (x,y) such that H(x) = H(y).

9

Cryptographic Hash Functions

An **hash function** H has the following properties:

- H can be applied to data of any size.
- H produces fixed length output.
- H(x) is easy to compute for any given x.

- **One-way**

  for any given hash-code h, it is computationally infeasible to find x such that H(x) = h.
- **Weak collision resistance**

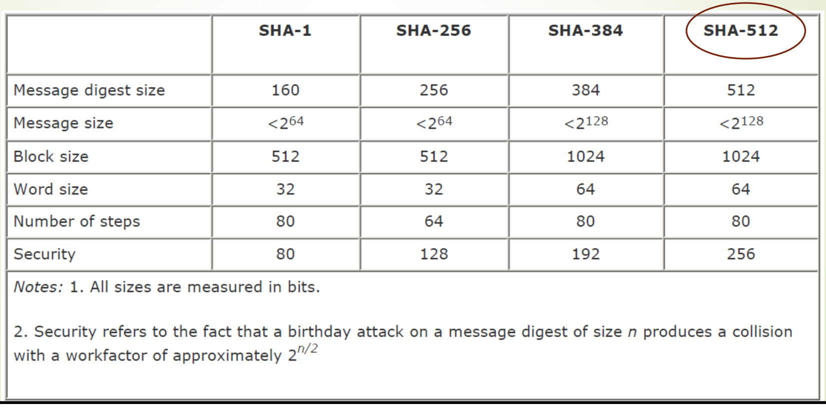
  for any given x, it is computationally infeasible to find y (not equal to x) such that H(y) = H(x).
- **Strong collision resistance**

  it is computationally infeasible to find any pair (x,y) such that H(x) = H(y).

Could you use DES to implement a Cryptographic Hash Function?

Secure Hash Algorithm (SHA)

- Developed by NSA in 1993. Based on MD4.
- In 20002 a revised version by NIST. In 2005 SHA-1 started to be phased out by NIST. By 2010 SHA-256, SHA-384, and SHA-512.
- Around 2005 an attack were 2 different message could be found using 2^{69} operations yielding the same SHA-1 hash! (2^{80} operations were expected to be necessary)

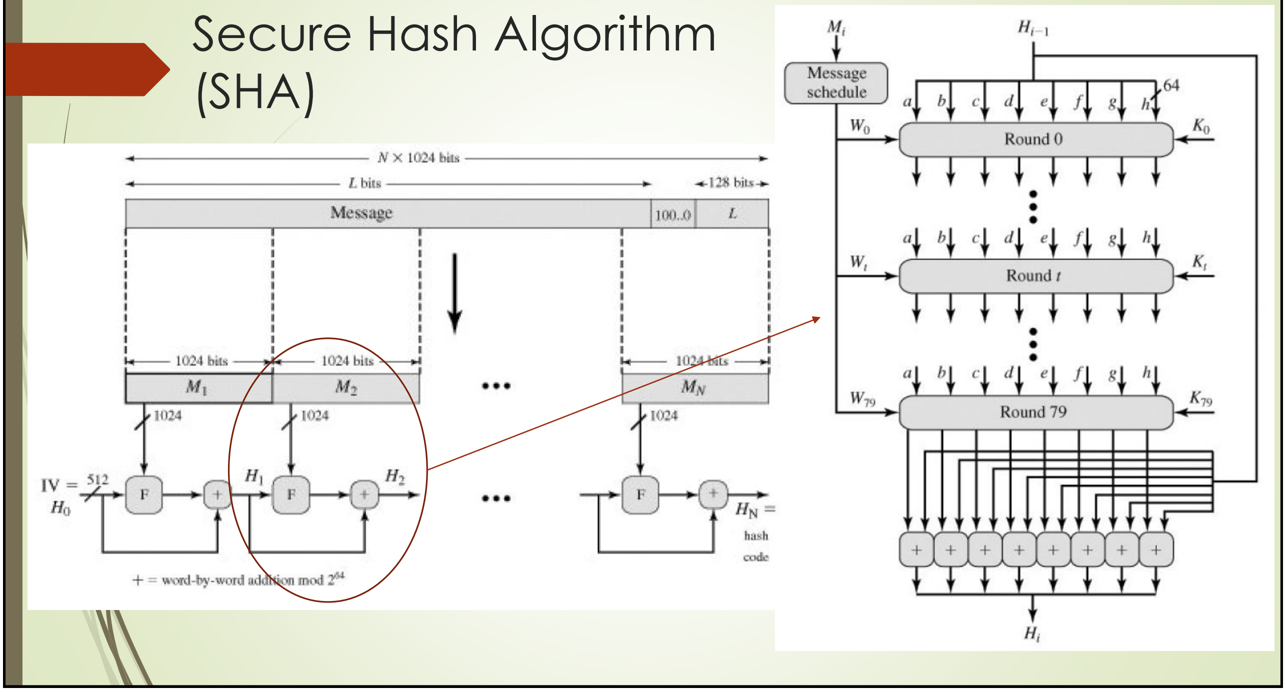


	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	64	80	80
Security	80	128	192	256

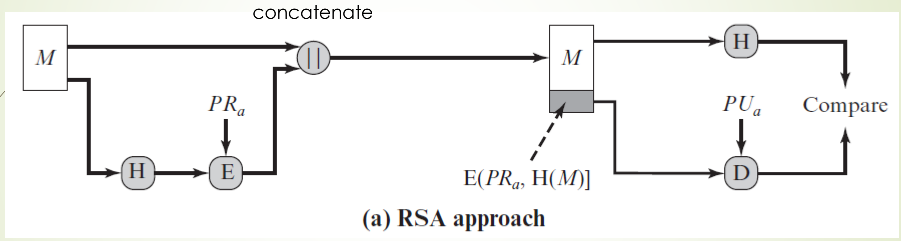
Notes: 1. All sizes are measured in bits.

2. Security refers to the fact that a birthday attack on a message digest of size n produces a collision with a workfactor of approximately $2^{n/2}$

Secure Hash Algorithm (SHA)



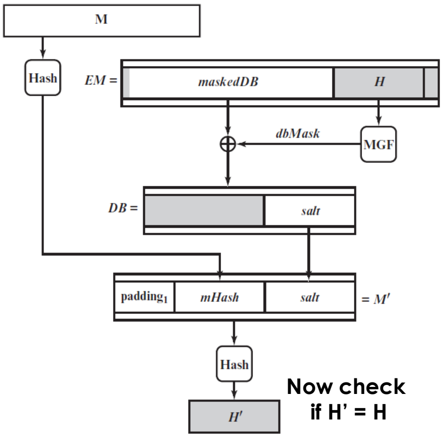
RSA Digital Signatures



(a) RSA approach

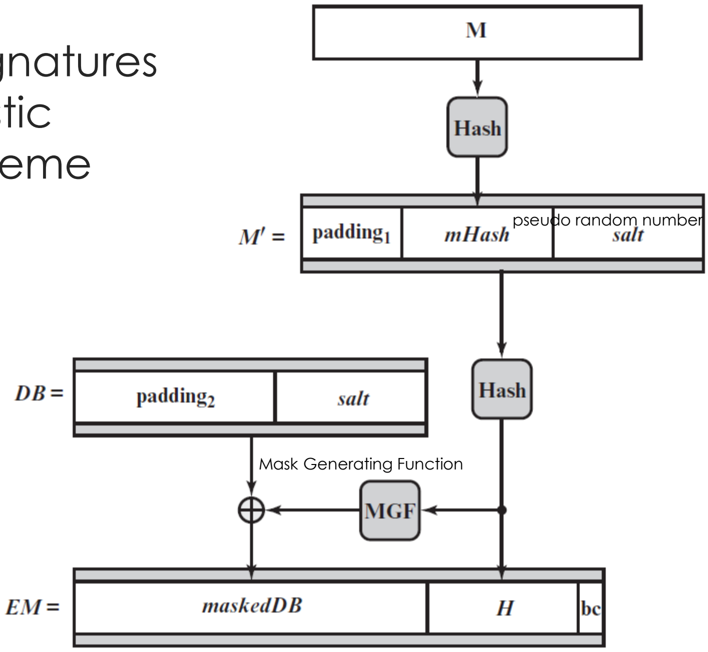
H = Secure Hash Code of fixed Length
Pr_a = Private Key of Alice
Pu_a = Public Key of Alice

RSA Digital Signatures RSA Probabilistic Signature Scheme



Now check if H' = H

Figure 13.7 RSA-PSS EM Verification



pseudo random number

Mask Generating Function

Figure 13.6 RSA-PSS Encoding

Authentication: Passwords

Password file protected by:

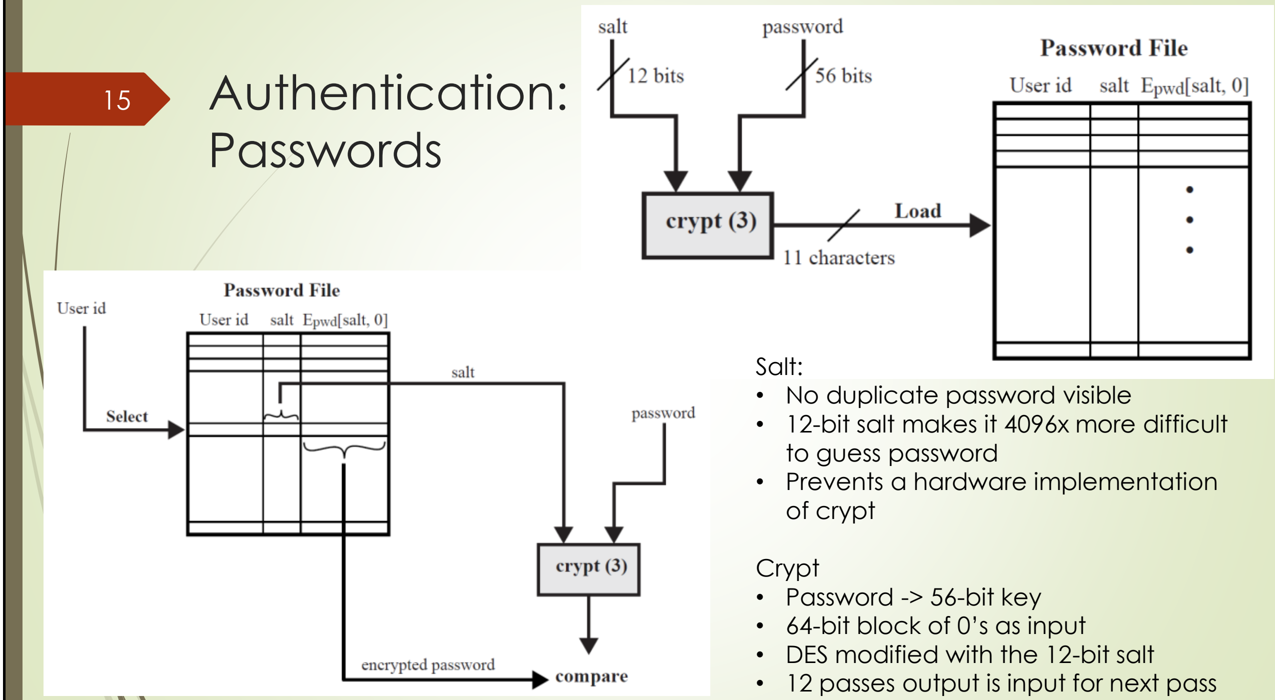
- One-way encryption: only encrypted passwords are stored
- Access control: password files only accessed only by a few accounts

Passwords cracked by:

- Trying default passwords, all short passwords.
- Dictionary attacks, license plates, etc.
- User's information: pets, names, children names etc.
- Trojan Horse to bypass access restrictions
- Line tap between user and system.

15

Authentication: Passwords



Salt:

- No duplicate password visible
- 12-bit salt makes it 4096x more difficult to guess password
- Prevents a hardware implementation of crypt

Crypt

- Password -> 56-bit key
- 64-bit block of 0's as input
- DES modified with the 12-bit salt
- 12 passes output is input for next pass

16

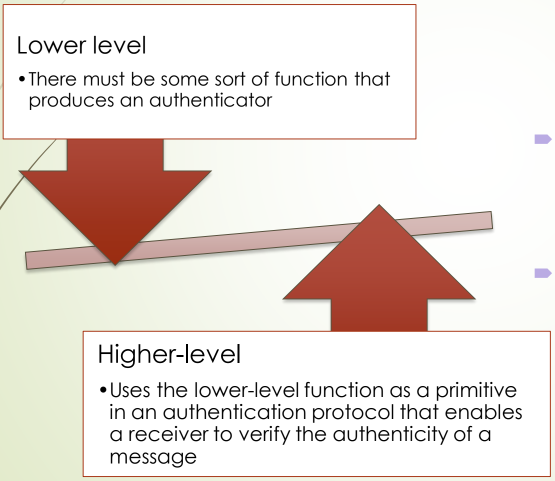
Message Authentication Challenges

- Disclosure
  - Release of message contents to any person or process not possessing the appropriate cryptographic key
- Traffic analysis
  - Discovery of the pattern of traffic between parties
- Masquerade
  - Insertion of messages into the network from a fraudulent source
- Content modification
  - Changes to the contents of a message, including insertion, deletion, transposition, and modification
- Sequence modification
  - Any modification to a sequence of messages between parties, including insertion, deletion, and reordering
- Timing modification
  - Delay or replay of messages
- Source repudiation
  - Denial of transmission of message by source
- Destination repudiation
  - Denial of receipt of message by destination

17

Message Authentication Functions

- Two levels of functionality:



Lower level

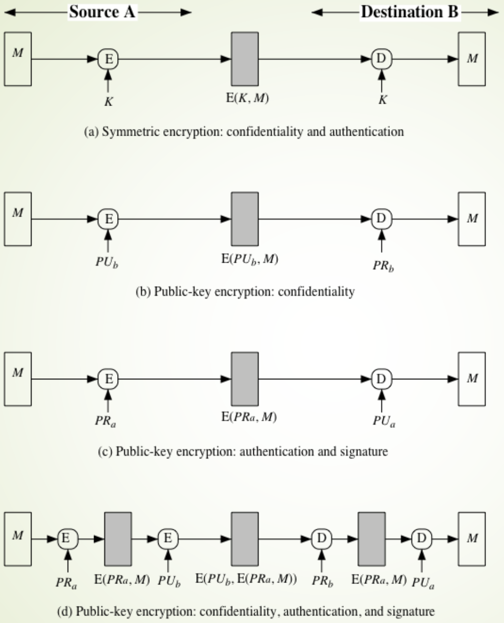
- There must be some sort of function that produces an authenticator

Higher-level

- Uses the lower-level function as a primitive in an authentication protocol that enables a receiver to verify the authenticity of a message

- Hash function
  - A function that maps a message of any length into a fixed-length hash value which serves as the authenticator
- Message encryption
  - The ciphertext of the entire message serves as its authenticator
- Message authentication code (MAC)
  - A function of the message and a secret key that produces a fixed-length value that serves as the authenticator

18



(a) Symmetric encryption: confidentiality and authentication

(b) Public-key encryption: confidentiality

(c) Public-key encryption: authentication and signature

(d) Public-key encryption: confidentiality, authentication, and signature

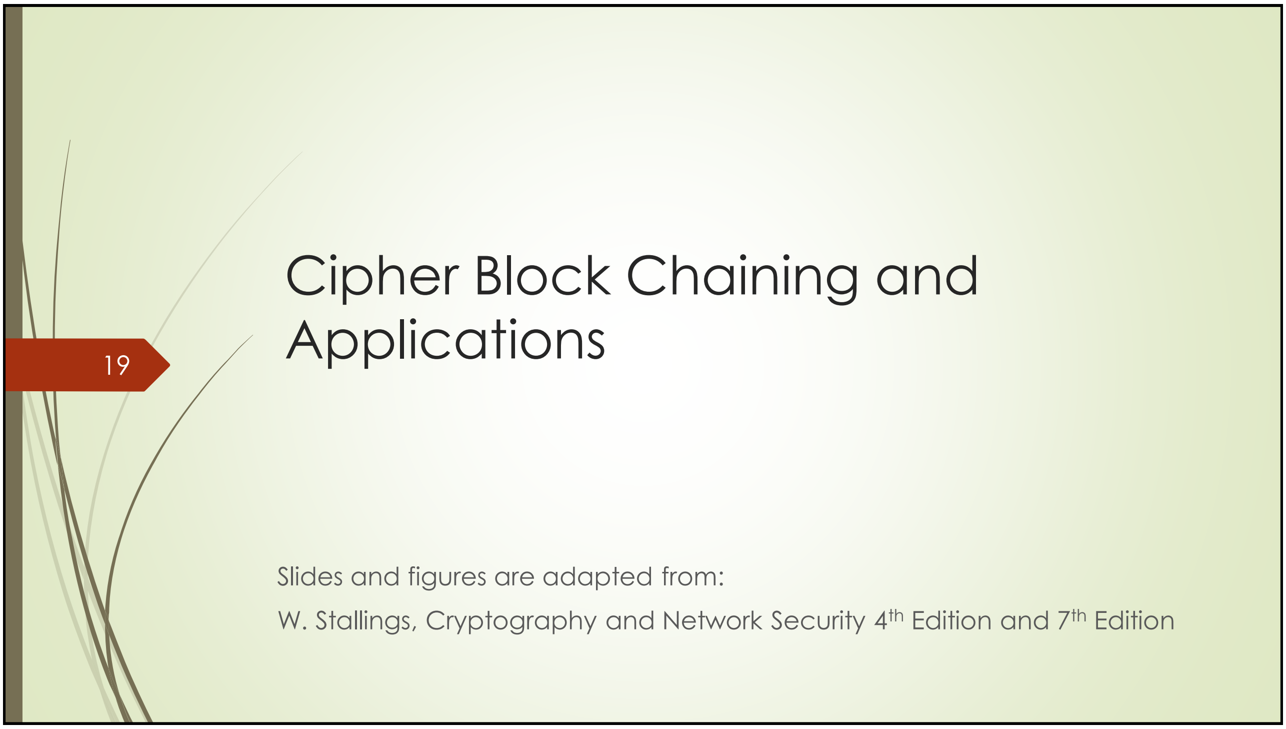
Figure 12.1 Basic Uses of Message Encryption

Scenario: Message M send by Specialist Alice to Specialist Bob

Secrecy and Authentication

- Secret shared key K

- PU_a is PUblic key from Alice
- PR_a is PRivate key from Alice

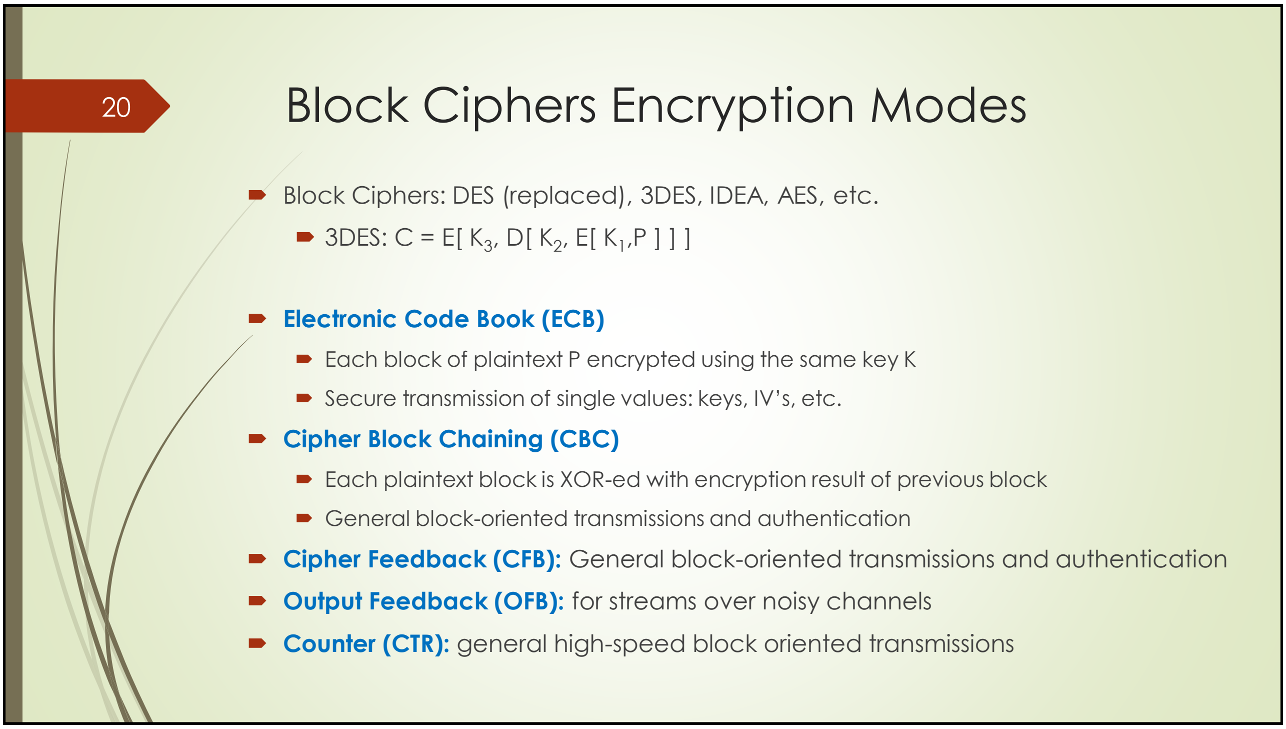


19

Cipher Block Chaining and Applications

Slides and figures are adapted from:

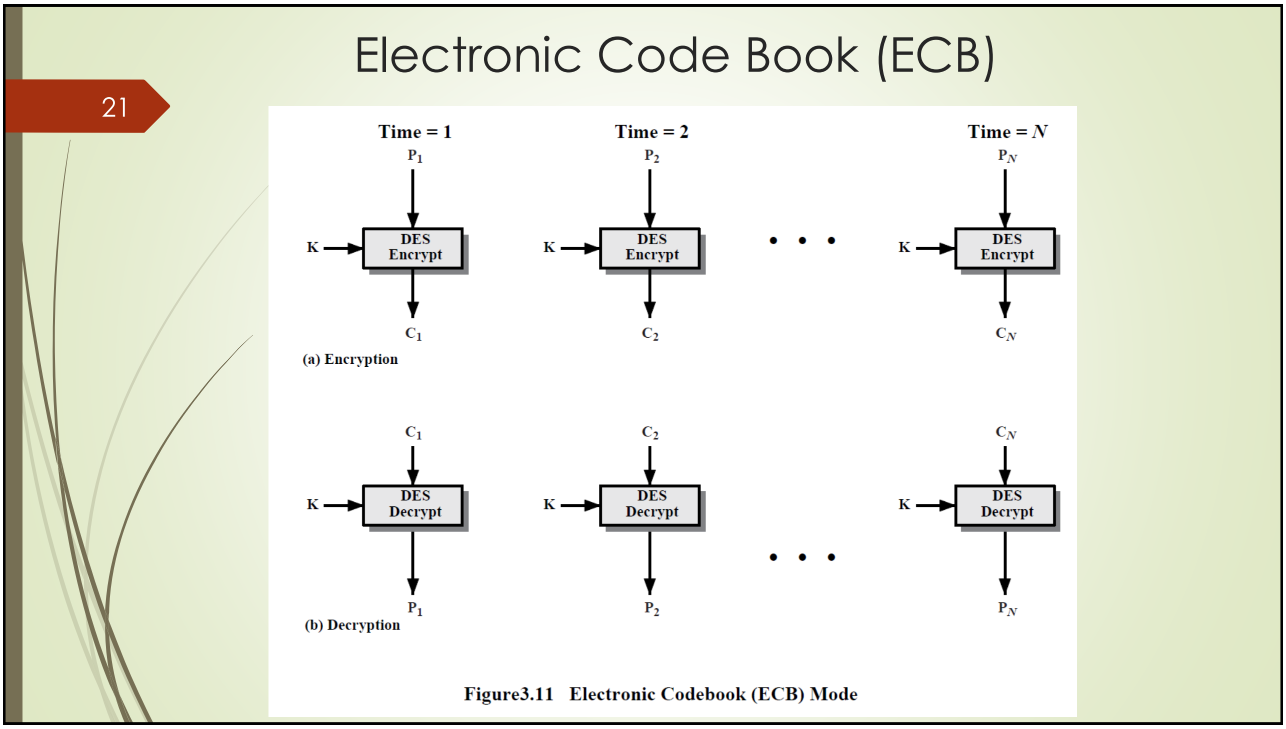
W. Stallings, Cryptography and Network Security 4th Edition and 7th Edition



20

Block Ciphers Encryption Modes

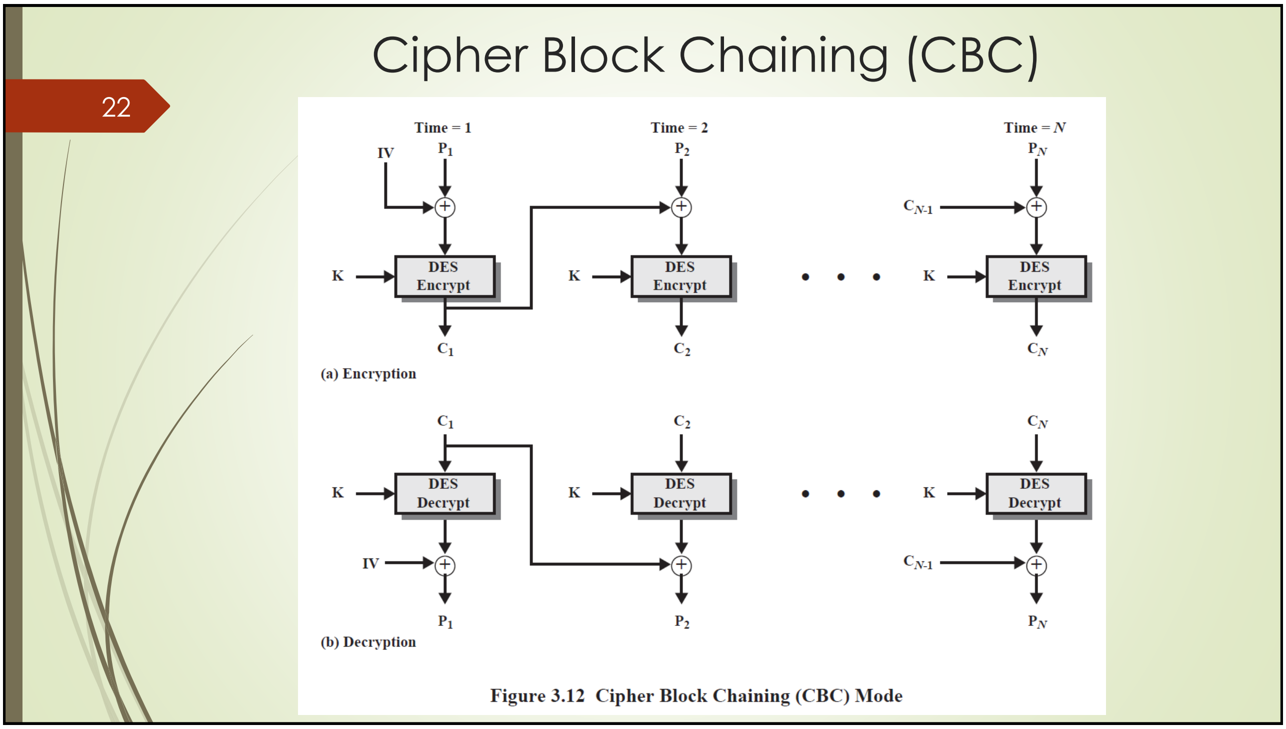
- Block Ciphers: DES (replaced), 3DES, IDEA, AES, etc.
  - 3DES: $C = E[K_3, D[K_2, E[K_1,P]]]$
- **Electronic Code Book (ECB)**
  - Each block of plaintext P encrypted using the same key K
  - Secure transmission of single values: keys, IV's, etc.
- **Cipher Block Chaining (CBC)**
  - Each plaintext block is XOR-ed with encryption result of previous block
  - General block-oriented transmissions and authentication
- **Cipher Feedback (CFB):** General block-oriented transmissions and authentication
- **Output Feedback (OFB):** for streams over noisy channels
- **Counter (CTR):** general high-speed block oriented transmissions



Electronic Code Book (ECB)

21

Figure3.11 Electronic Codebook (ECB) Mode



Cipher Block Chaining (CBC)

22

Figure 3.12 Cipher Block Chaining (CBC) Mode

23

Cipher Block Chaining (CBC) IV should be unpredictable

- K and IV are known to both Alice and Bob
- IV must be unpredictable, communicated using ECB
- If Eve knows IV and/or is able to let Bob use *IV'*, then

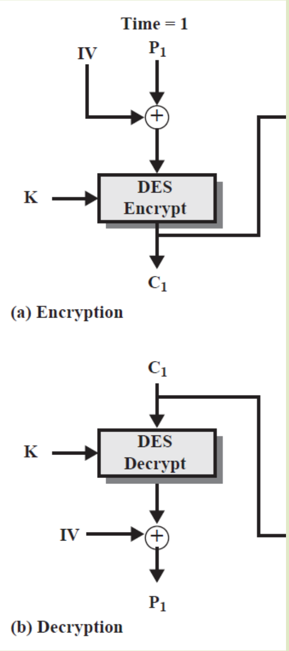
$$C_1 = E_K[\, IV \otimes P_1 \,]$$

$$P_1 = IV + D_K[C_1]$$

$$\Rightarrow P_1[i] = IV[i] + D_K[C_1][i] \quad \text{, i-th bit}$$

$$\Rightarrow P_1'[i] = IV'[i] + D_K[C_1][i] \quad \text{, i-th bit}$$

- Eve can change bits in the plaintext P_1^1 that Bob thinks he received.



(a) Encryption

(b) Decryption

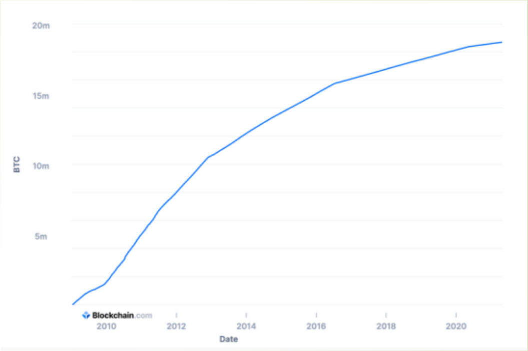
24

Bitcoin: A Peer-to-Peer Electronic Cash System Satoshi Nakamoto, 2008 (http://bitcoin.org/bitcoin.pdf)

- Developed by a person or group under the pseudonym **Satoshi Nakamoto** in 2008
- Decentralized (peer-to-peer) e-cash
- No trust or third party but cryptographic proof against double spending.
- The P2P-network timestamps transactions by hashing them into an ongoing chain of **hash-based proof-of-work**.
- Blocks cannot be changed without redoing the proof-of-work.
- The longest chain serves as proof:
  - of the sequence of transactions/events, and
  - proof that it came from the largest pool of CPU power.
  - As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers
- Operational since early 2009

25

Bitcoins



- Digital Wallet
- Transactions
  - Between owner and receiver
  - Broadcasted on P2P-network (public, ‘anonymous’)
  - Mining nodes collect the transactions into blocks
- Proof of Work
- Mining

26

Bitcoins Transactions

- Transactions
  - Between owner and receiver
  - Broadcasted on P2P-network (public, ‘anonymous’)
  - Mining nodes collect the transactions into blocks
- A transaction block is a full page in a ledger book
- A block contains info of transaction info and (cryptographically) links to previous blocks.
- Links can be followed to the first block of the Bitcoin Network.
- The Block Chain file is maintained at every node of the network.

27

Transactions

- Transfer of coin between owner and recipient
- Double spending still possible

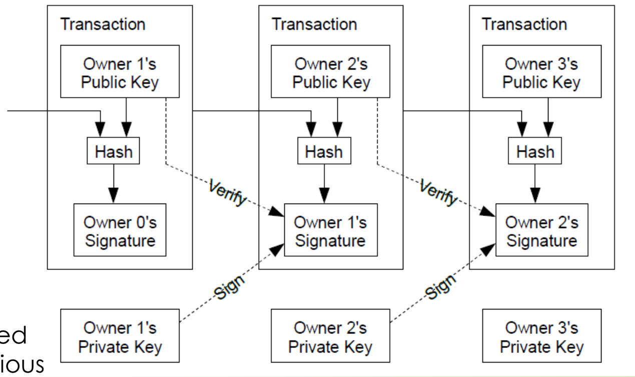
Solution:

- whole history will be maintained
- Block of transaction timestamped and hashed
- Chained by including previous timestamp/hash

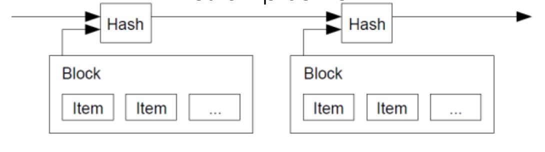
⇒ absence of a transaction can be checked

⇒ Each new timestamp reinforces the previous ones

Owner 1 transfers to Owner 2



Timestamp Server



28

Bitcoins Transactions

A Distributed Timestamp Server

- Each Block caries a Proof of Work
  - Scanning for (by incrementing) a Nonce such that when added to the block the hash starts with a number of zero bits (i.e. hash < Threshold)
- This starts a new block that is linked to the block chain.
- The machine that generated the solution is rewarded with a new Bitcoin. [In the future it can also be transaction fees.]
- The first transaction in a block consist of the new coin owned by the creator of the block.
- This new block chain status is broadcasted on the P2P network. The longest chain will be taken (majority vote/most work).

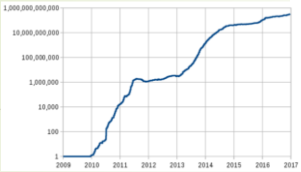
29

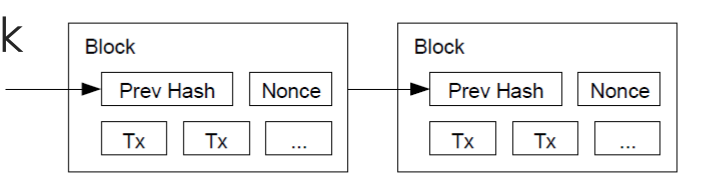
Running the Network

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

30

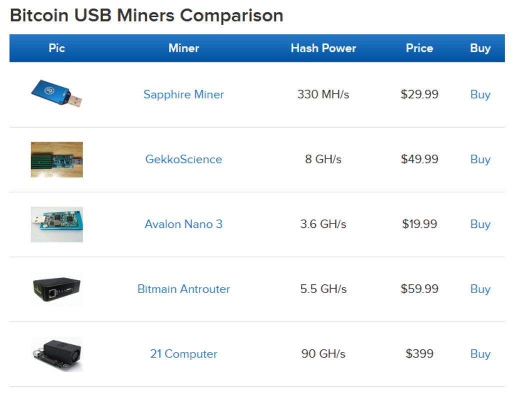
Proof of Work





- **Miners** keep the blockchain consistent, complete, and unalterable by repeatedly verifying and collecting newly broadcast transactions into a new group of transactions called a block.
- Each block contains a cryptographic hash of the previous block, using the SHA-256 hashing algorithm
- A new block must contain a so-called **proof-of-work, a nonce, such that when the block content is hashed along with the nonce, the result is numerically smaller than the network's difficulty target.**
- To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour.
- Between 1 March 2014 and 1 March 2015, the average number of nonces miners had to try before creating **a new block** increased from **16.4 x 10^{18} to 200.5 x10^{18} .**
- an attacker must modify all subsequent blocks in order for the modifications of one block to be accepted and increases as time passes.

Bitcoin Miner Hardware



Bitcoin USB Miners Comparison

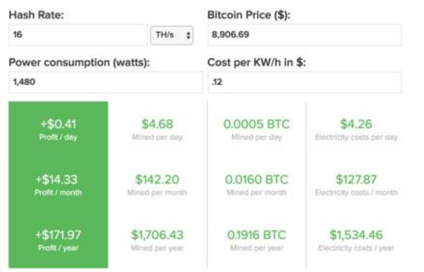
Pic	Miner	Hash Power	Price	Buy
	Sapphire Miner	330 MH/s	$29.99	Buy
	GekkoScience	8 GH/s	$49.99	Buy
	Avalon Nano 3	3.6 GH/s	$19.99	Buy
	Bitmain Antrouter	5.5 GH/s	$59.99	Buy
	21 Computer	90 GH/s	$399	Buy

https://www.buybitcoinworldwide.com/mining/hardware/

Since it's now impossible to profitably mine Bitcoin with your computer, you'll need specialized hardware called ASICs.

Here's what an ASIC miner looks like up close:

The Dragonmint 16T miner.



Hash Rate: 16 TH/s

Bitcoin Price ($): 8,906.69

Power consumption (watts): 1,480

Cost per KW/h in $: .12

+$0.41 Profit / day	$4.68 Mined per day	0.0005 BTC Mined per day	$4.26 Electricity costs per day
+$14.33 Profit / month	$142.20 Mined per month	0.0160 BTC Mined per month	$127.87 Electricity costs / month
+$171.97 Profit / year	$1,706.43 Mined per year	0.1916 BTC Mined per year	$1,534.46 Electricity costs / year

Note that is appears profitable even with high electricity costs ($0.12 per KW/h). With $0.03 / KW/h it's even more profitable:

Hash rates vs Difficulty

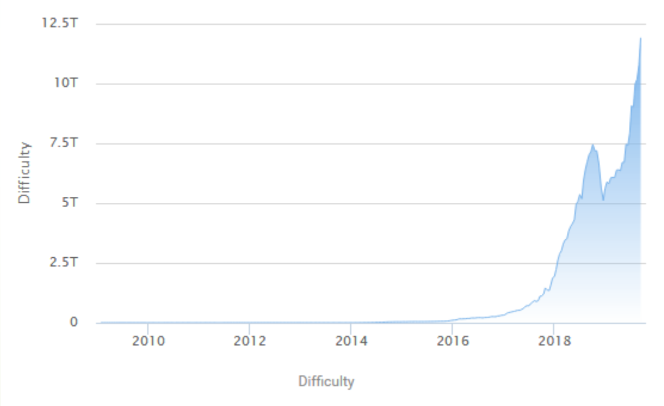
90.38×10^{18} H/s

Current difficulty level is such that a 16TH/s mining machine takes about 100 years on average to find a block.

12.5 bitcoins per block
=> Reward of ~1000 euro/year
…

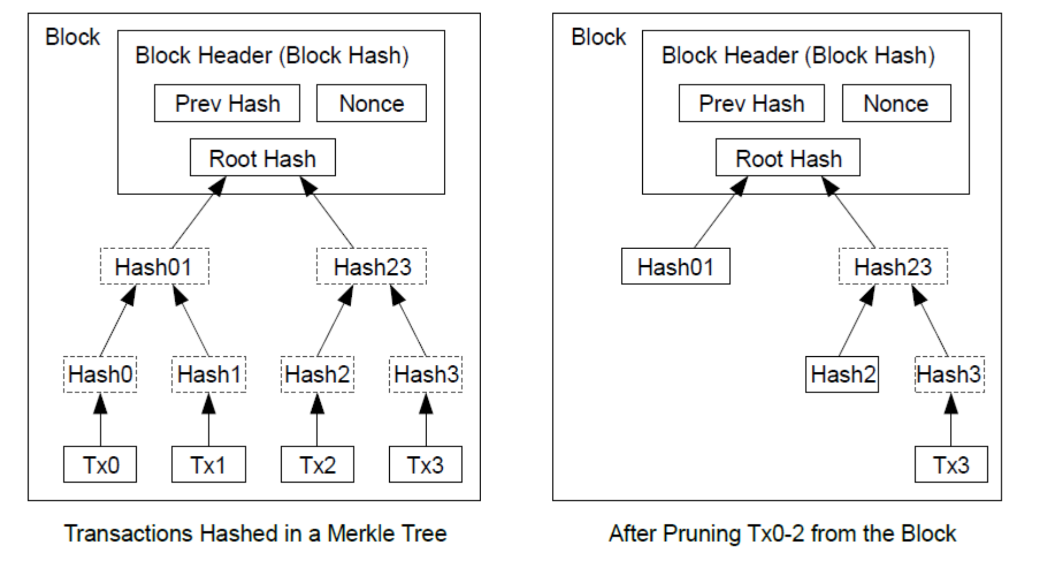
Hashrate 90.38 EH/s

Difficulty 11,890,594,958,795 - **11.89 T**



33

At some point the spent transactions before can be discarded if buried under enough new blocks.
Pruning transactions 0, 1, and 2:



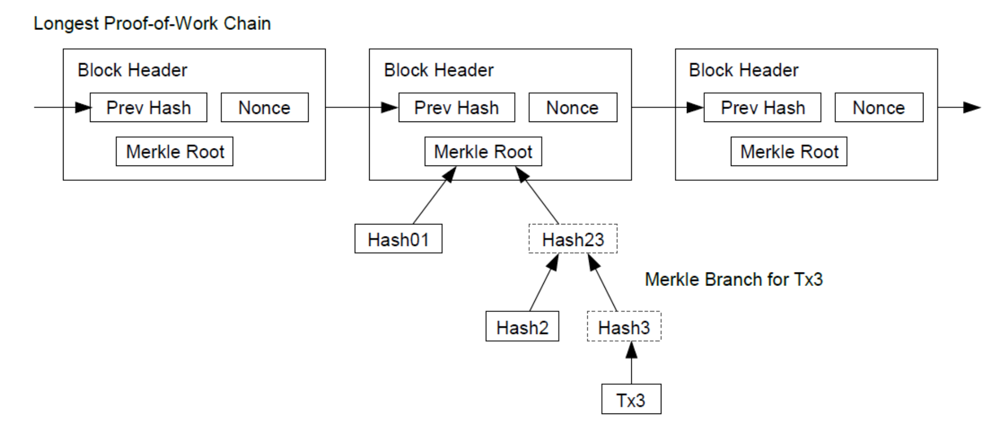
Transactions Hashed in a Merkle Tree

After Pruning Tx0-2 from the Block

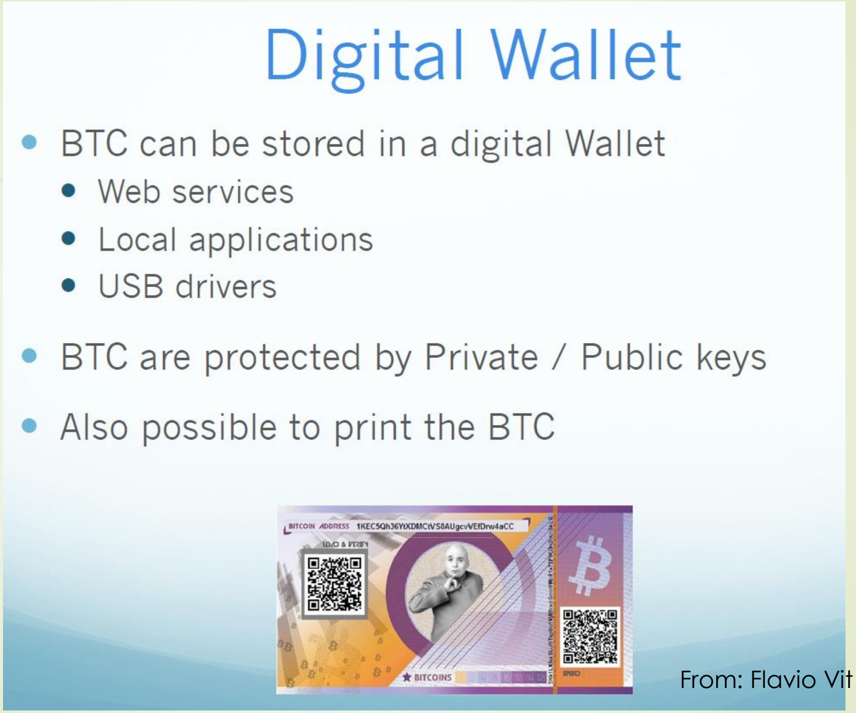
34

Verify Payments:

- Keep a copy of the block headers of the longest proof of work chain
  Obtain this by querying network nodes until convinced that it is the longest.
- Link the transaction, Tx3 for example, to the block it's timestamped in.
- => proof that a network node accepted Tx3
- => blocks added after it further confirm the network has accepted it.



35



Digital Wallet

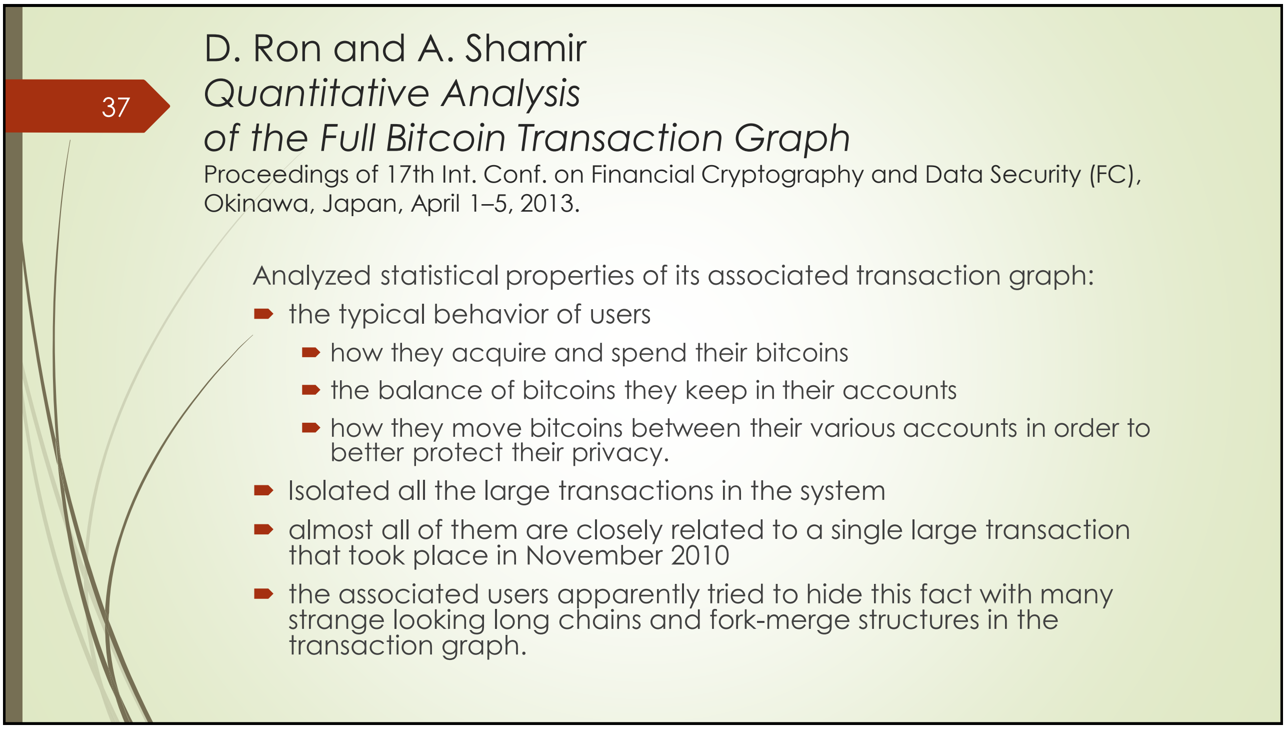
- BTC can be stored in a digital Wallet
  - Web services
  - Local applications
  - USB drivers
- BTC are protected by Private / Public keys
- Also possible to print the BTC

From: Flavio Vit presentation

36

Digital Wallet

- Bitcoins fraction => the smallest fraction:
- 1 Satoshi is 0.00000001 BTC
- Losing your private key => losing yours BTCs …
- Forever gone from BTC economy
- BTC is deflationary!



37

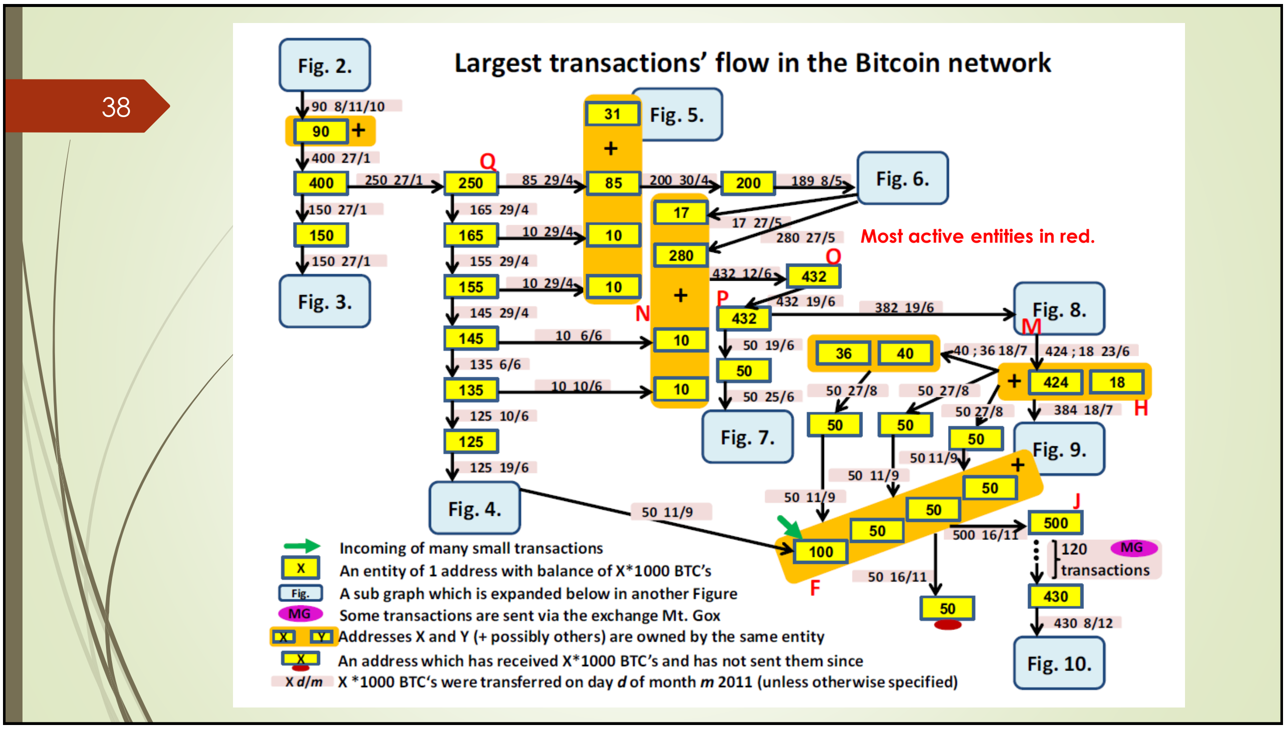
D. Ron and A. Shamir

Quantitative Analysis
of the Full Bitcoin Transaction Graph

Proceedings of 17th Int. Conf. on Financial Cryptography and Data Security (FC), Okinawa, Japan, April 1–5, 2013.

Analyzed statistical properties of its associated transaction graph:

- the typical behavior of users
  - how they acquire and spend their bitcoins
  - the balance of bitcoins they keep in their accounts
  - how they move bitcoins between their various accounts in order to better protect their privacy.
- Isolated all the large transactions in the system
- almost all of them are closely related to a single large transaction that took place in November 2010
- the associated users apparently tried to hide this fact with many strange looking long chains and fork-merge structures in the transaction graph.



38

Largest transactions’ flow in the Bitcoin network

Most active entities in red.

→ Incoming of many small transactions

X An entity of 1 address with balance of X*1000 BTC’s

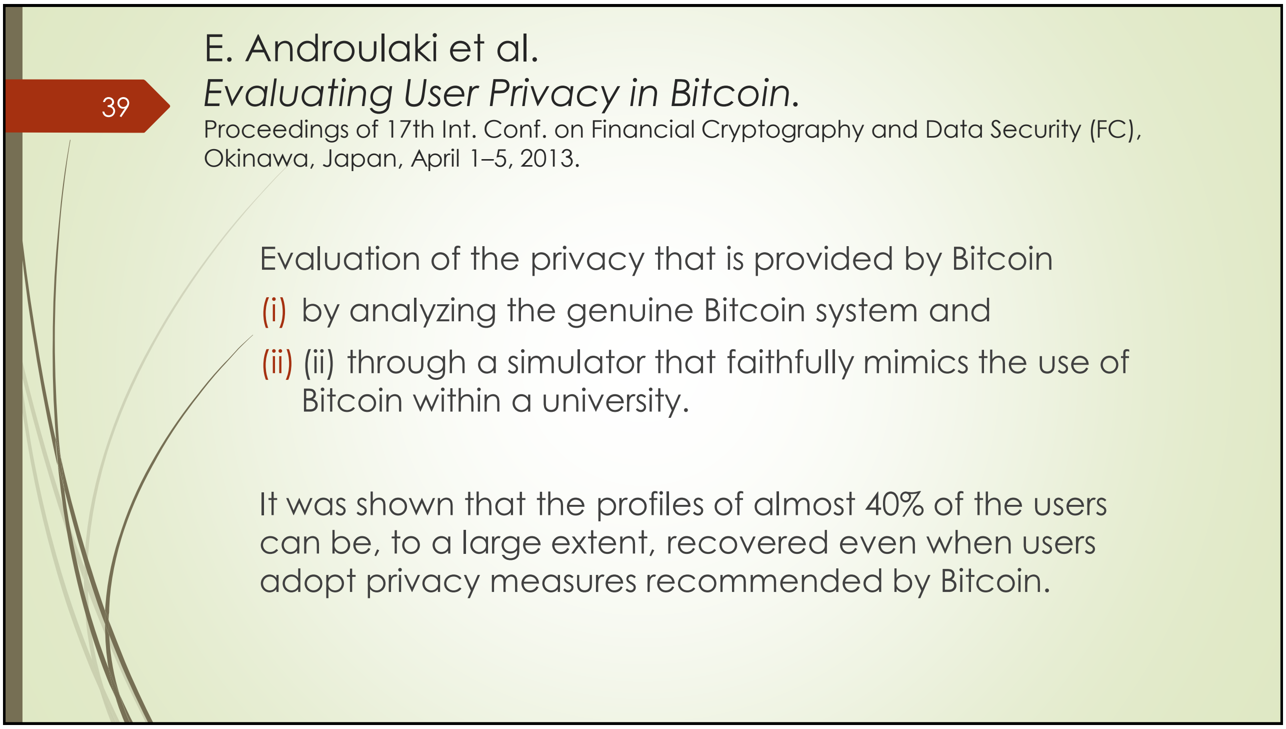
Fig. A sub graph which is expanded below in another Figure

MG Some transactions are sent via the exchange Mt. Gox

X Y Addresses X and Y (+ possibly others) are owned by the same entity

X An address which has received X*1000 BTC’s and has not sent them since

X *d*/*m* X *1000 BTC‘s were transferred on day *d* of month *m* 2011 (unless otherwise specified)



39

E. Androulaki et al.

Evaluating User Privacy in Bitcoin.

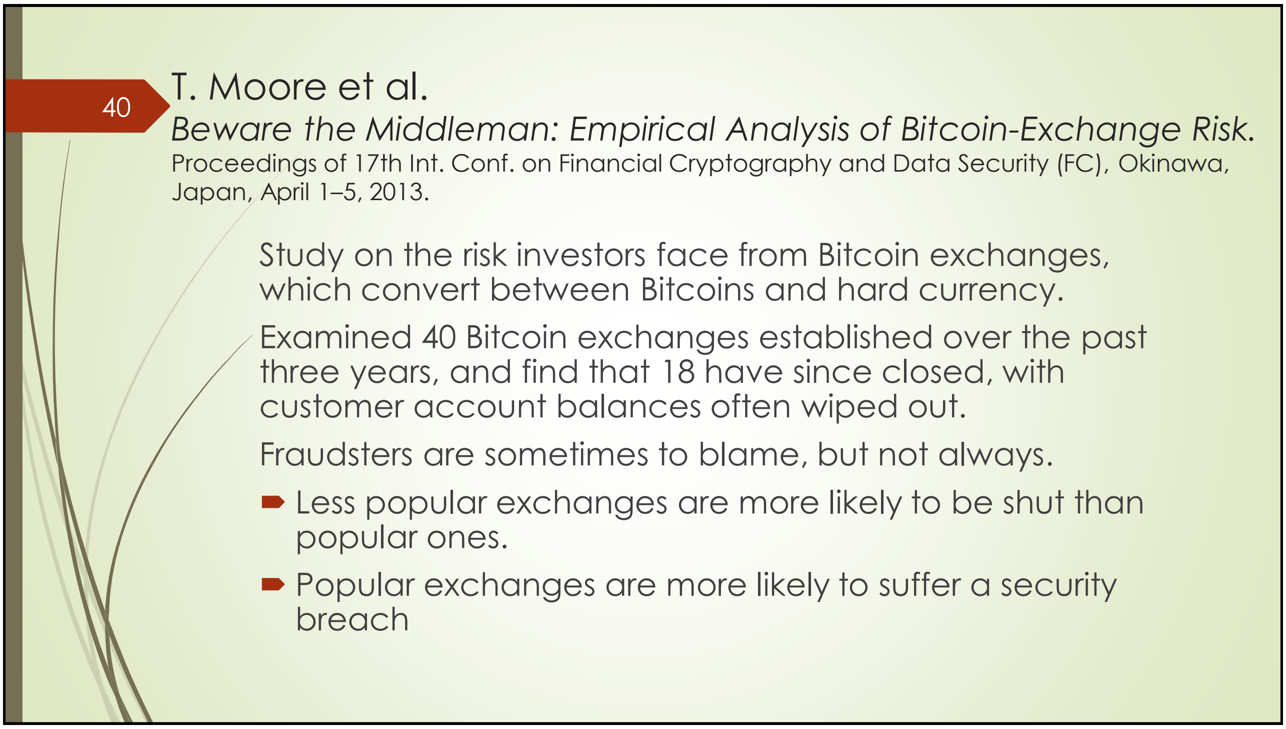
Proceedings of 17th Int. Conf. on Financial Cryptography and Data Security (FC), Okinawa, Japan, April 1–5, 2013.

Evaluation of the privacy that is provided by Bitcoin

(i) by analyzing the genuine Bitcoin system and

(ii) (ii) through a simulator that faithfully mimics the use of Bitcoin within a university.

It was shown that the profiles of almost 40% of the users can be, to a large extent, recovered even when users adopt privacy measures recommended by Bitcoin.



40

T. Moore et al.

Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk.

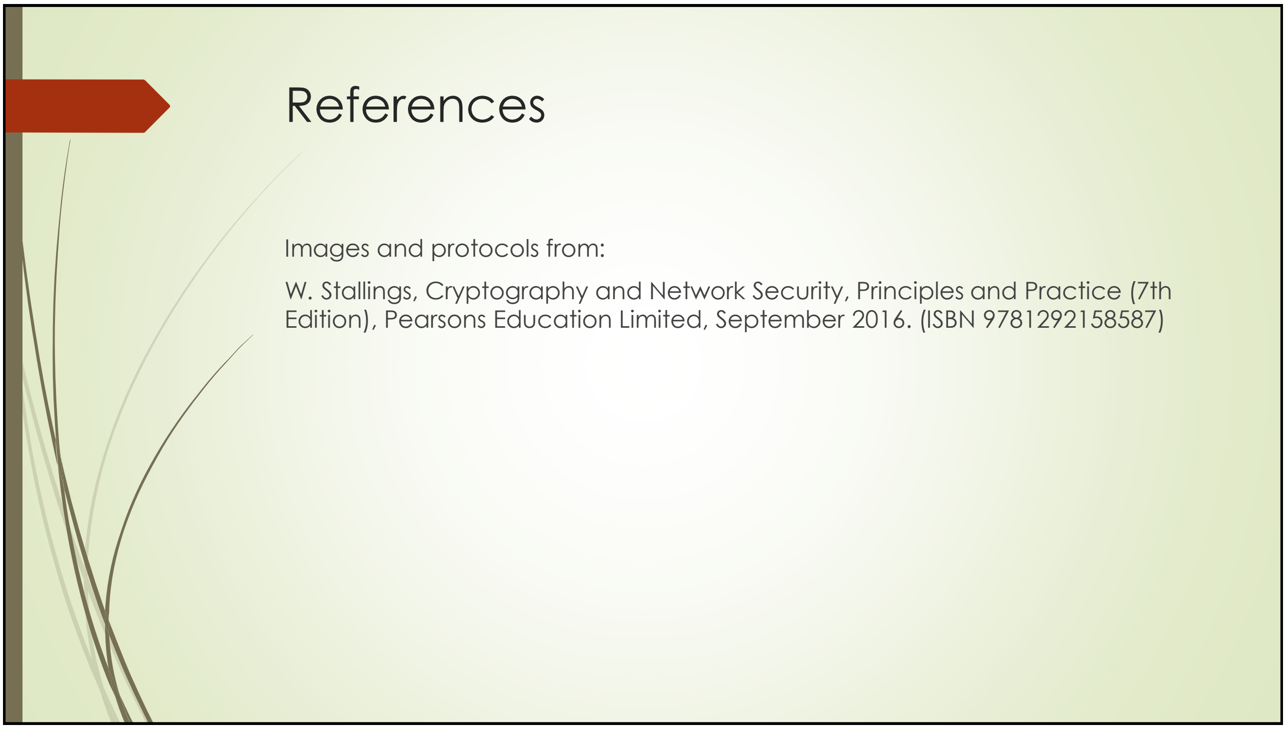
Proceedings of 17th Int. Conf. on Financial Cryptography and Data Security (FC), Okinawa, Japan, April 1–5, 2013.

Study on the risk investors face from Bitcoin exchanges, which convert between Bitcoins and hard currency.

Examined 40 Bitcoin exchanges established over the past three years, and find that 18 have since closed, with customer account balances often wiped out.

Fraudsters are sometimes to blame, but not always.

- Less popular exchanges are more likely to be shut than popular ones.
- Popular exchanges are more likely to suffer a security breach



References

Images and protocols from:

W. Stallings, Cryptography and Network Security, Principles and Practice (7th Edition), Pearsons Education Limited, September 2016. (ISBN 9781292158587)