

Payment systems

Tuomas Aura
CS-C3130 Information security

Aalto University, autumn 2019

Outline

1. EMV card payment
2. (More card security features)
3. (Anonymous digital cash)
4. Bitcoin

EMV CARD PAYMENT

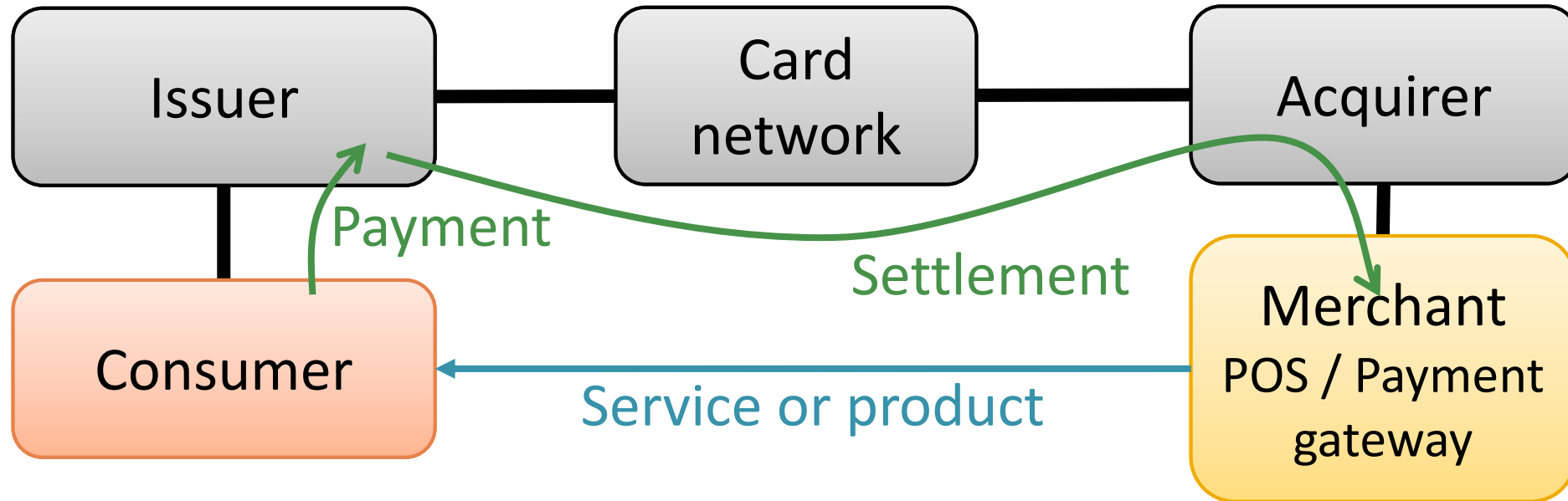
EMV bank cards

- Smart card chip (ICC)
 - Tamperproof ICC stores the **private signature key** and a **certificate**
 - Also, a **shared key** with card issuer
- EMV standard specifies protocols between card, terminal, issuer
 - Message contents between card and issuer left somewhat open (no need for interoperability)
 - Many options, implementations vary between countries



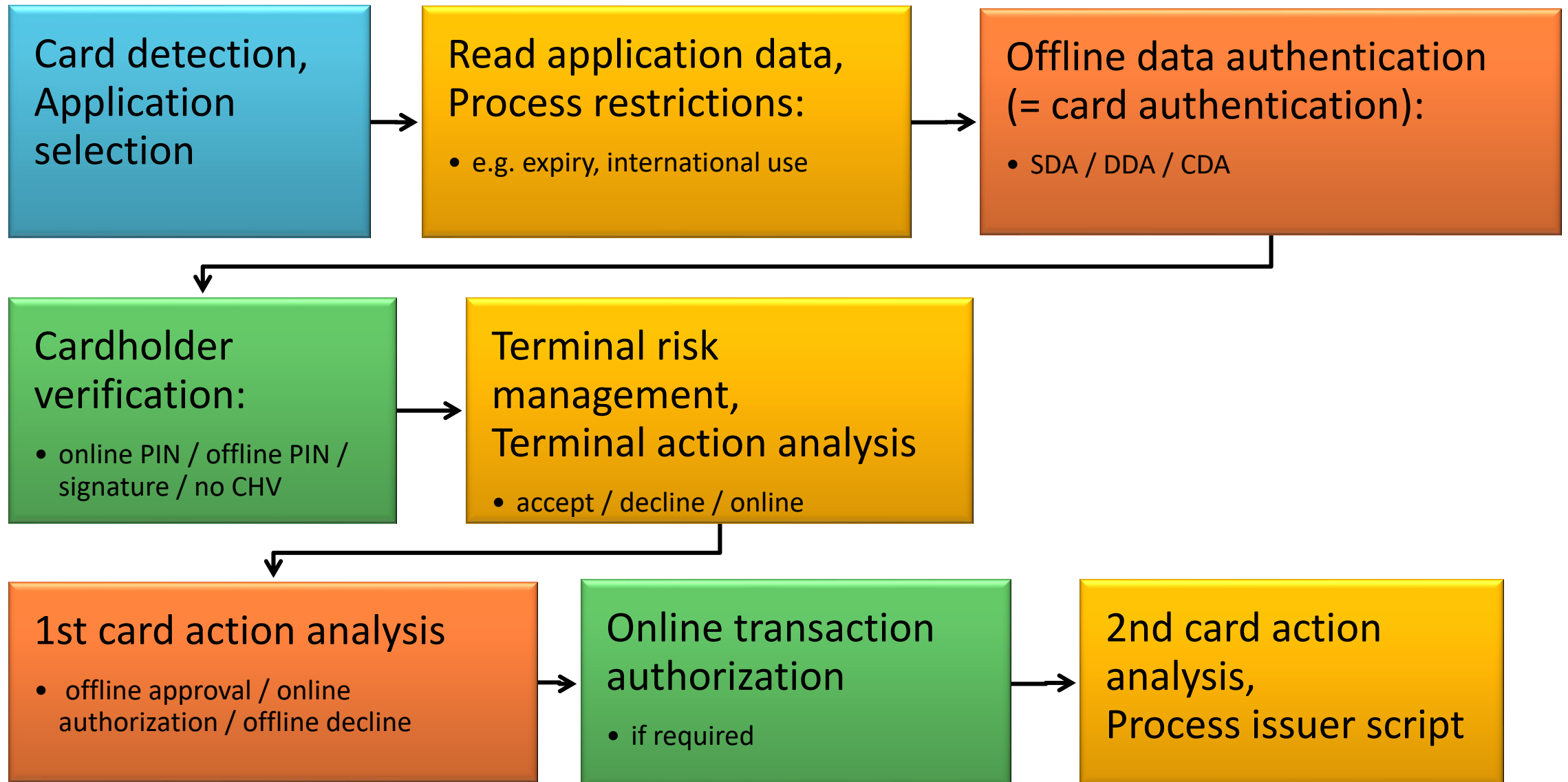
Photos: Mastercard, Visa

Terminology



- Four-corner model i.e. four-party scheme
- Card networks: Mastercard, Visa etc.
- Issuing bank, acquiring bank

Card payment process



Application selection

Card scheme	RID	Product	PIX	AID
Danmønt (Denmark)	A000000001	Cash card	1010	A0000000011010
Visa	A000000003	Visa credit or debit	1010	A0000000031010
		Visa Electron	2010	A0000000032010
		V Pay	2020	A0000000032020
		Plus	8010	A0000000038010
Mastercard	A000000004	Mastercard credit or debit	1010	A0000000041010
		Mastercard ^[16]	9999	A0000000049999
		Maestro	3060	A0000000043060
		Cirrus ATM card only	6000	A0000000046000
Mastercard	A000000005	Maestro UK (formerly branded as Switch)	0001	A0000000050001
American Express	A000000025	American Express	01	A00000002501
LINK ATM network	A000000029	ATM card	1010	A0000000291010
CB (France)	A000000042	CB (credit or debit card)	1010	A0000000421010
		CB (Debit card only)	2010	A0000000422010
JCB	A000000065	Japan Credit Bureau	1010	A0000000651010
Dankort (Denmark)	A000000121	Debit card	1010	A0000001211010
		Dankort (debit card)	4711	A0000001214711
Consorzio Bancomat (Italy)	A000000141	Bancomat/PagoBancomat	0001	A0000001410001
Diners Club/Discover	A000000152	Diners Club/Discover	3010	A0000001523010
Banrisul (Brazil)	A000000154	Banricompras Debito	4442	A0000001544442
SPAN2 (Saudi Arabia)	A000000228	SPAN	1010	A00000022820101010
Interac (Canada)	A000000277	Debit card	1010	A0000002771010
...				

Card authentication

- Called **offline data authentication**
 - Takes place offline, before deciding whether to contact the issuer online or not
- Three levels of card authentication:
 1. **Static data authentication (SDA):**
 - Certificate verification only, no signature

No longer used in Finland; certificate can be copied
 2. **Dynamic data authentication (DDA):**
 - Card signs a random challenge from terminal with RSA; terminal verifies certificate and signature

Currently the main method
 3. **Combined data authentication (CDA):**
 - Card signs transaction details (in a later step)

Limited advantage over DDA (why?)

Cardholder verification

- Cardholder verification methods (CVM)
 - Online PIN: PIN sent to card issuer for verification
 - Offline plaintext PIN: plaintext PIN verified by card
 - Offline enciphered PIN: PIN sent to the card encrypted with the card's RSA key (2nd key and cert)
 - Handwritten signature
 - No CVM
- Terminal reads the list of methods supported by the card and chooses one
- Online PIN used mainly for debit cards, offline for credit cards



Photo: Wikimedia

Payment authorization – offline

- **Offline approval** for low-risk transactions:
 - Card produces **Transaction Certificate (TC)** in 1st card action analysis phase
 - The online transaction authorization is skipped
- TC contains transaction details and a MAC with the card-issuer shared key
- Card can also decline the transaction offline, or require online approval
- In CDA, card also returns **Signed Dynamic Application Data (SDAD)**

Payment authorization – online

■ Online authorization:

- Card produces **Authorization Request Cryptogram (ARQC)** in 1st card action analysis phase
- Terminal sends ARQC to issuer to approve or decline
- ARQC is authenticated with the card–issuer shared key
- Issuer returns Authorization Response Cryptogram (ARPC), and terminal passes it to the card
- Card produces **Transaction Certificate (TC)** or declines the transaction

Risk management

- Banks and credit-card issuers focus on risk management
- Two separate decisions about online vs. offline processing: PIN verification and transaction authorization
- The main decision is online vs. offline authorization
 - Dynamic risk assessment by both card and terminal
 - Offline risk parameters on the card limit offline transactions
 - Terminal may have its own limits
 - ATM cash withdrawal is always authorized online
 - Some cards (e.g. Visa Electron) only allow online authorization
- After transaction processing:
 - Offline limit is reset after successful online authorization
 - Issuer may also update limits or block the card

Contactless (NFC) payment

- Fast DDA (fDDA)
 - optimized signed message for contactless transactions
- No PIN verification
- Risk parameters for offline use
 - After a certain number of contactless transactions or total amount of money spent, require an online contact transaction with PIN entry
 - Soft and hard limits: after soft limit, online transaction is preferred but not required
 - Hard limits prevent some uses, e.g. busses



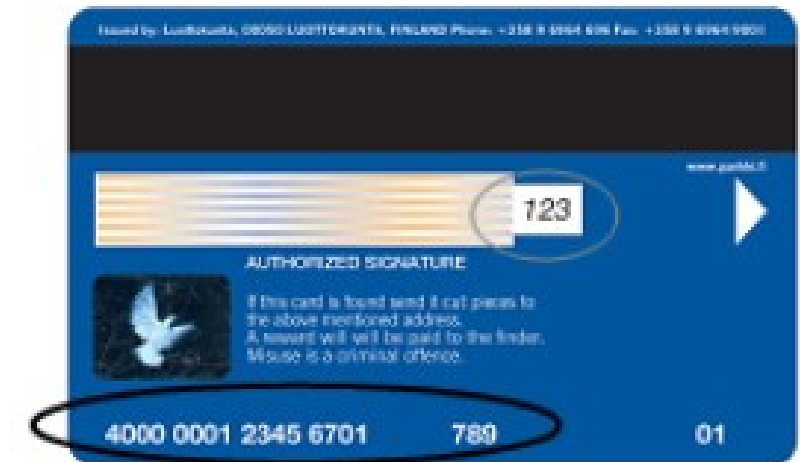
Picture: visa.ca

MORE CARD SECURITY FEATURES

Bank cards

These slides use
Visa terminology
and acronyms

- Credit or debit card features
 - Card number (issuer identification number IIN + primary account number PAN)
 - Card holder name, expiration date, CVV2
 - Magnetic stripe, chip in integrated circuit card (ICC), contactless (NFC) interface
 - Card holder signature, hologram
- PIN
- Terminal types
 - Point of sale (POS)
 - Automated teller machine (ATM) = cash machine



[Picture: www.korttiturvallisuus.fi, Nets Oy]

Historical mag-stripe bank cards

Extra
material

- Magnetic stripe contains card number, holder name, expiration date, service code, PVKI, PVV, CVV1
- CVV1 is a cryptographic MAC of the PAN, name, expiration and service code (based on 3DES)
 - It is possible to copy but not change the mag stripe data
- PIN is a function of data on mag stripe and a secret key
 - offline PIN verification at disconnected POS or ATM
- Offline terminals have a security module to store the card and PIN verification keys
- Service code: e.g. Visa Electron 121 where 2=online only

Outdated
technology but
good to read
for comparison

Mag-stripe Visa PIN verification

Extra
material

- Input from magnetic stripe:
 - Primary account number (PAN) i.e. 15-digit card number
 - PIN verification key indicator (PVKI, one digit 1..6)
 - PIN verification value (PVV, 4 decimal characters)
- Verifier must have
 - PIN verification key (PVK, 128-bit 3DES key)
 - PVKI is an index of PVK to enable key updates for PVK
- Create security parameter (TSP):
 1. Concatenate 11 rightmost digits of PAN, PVKI and PIN
 2. The 16-digit concatenation is one hexadecimal DES block
- PVV generation:
 1. 3DES encryption of TSP with the key PVK
 2. Decimalization of the encryption result to 4-digit PVV
- Decimalization happens by taking the 4 leftmost digits 0..9 from the hexadecimal encrypted block
 - If less than 4 such digits, take 4 first digits A..F and map A=0,B=1,C=3...

[For details see [IBM](#)]

CVV2

- How credit card numbers are mostly stolen:
 - Merchant's customer database hacked and credit card data leaks, e.g. SQL injection attacks (huge problem!)
 - Mag stripe skimmers in payment terminals get the same data
- **CVV2** (card verification value 2) to reduce online fraud
 - 3-4 digits printed on card but not on mag stripe or chip
 - Used in **card-not-present transactions**: web and phone
 - **Merchant verifies CVV2 online** from the issuer
 - **Merchant is not allowed to store CVV2**
 - Code changes when card renewed
 - Still vulnerable to phishing, corrupt merchants, and anyone with physical access to the card



3-D Secure

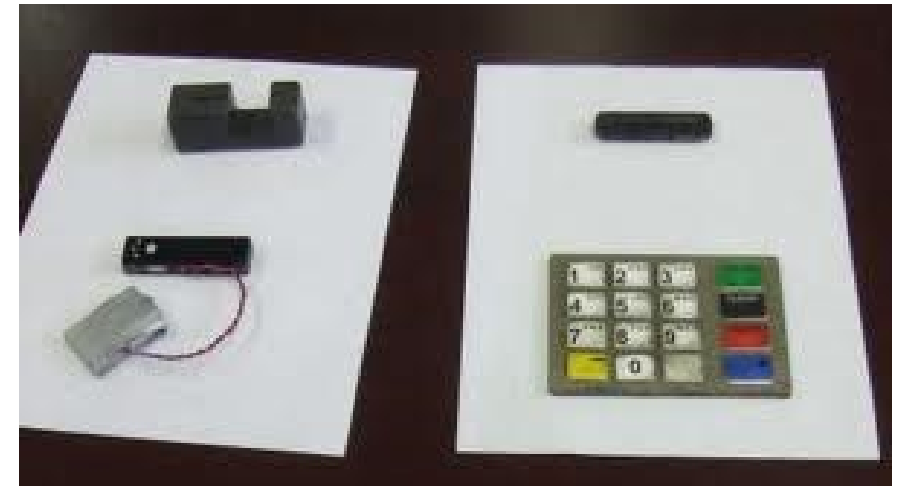
- Web- and XML-based protocol for authorizing payments
 - Merchant opens an iframe at the card issue
 - Card holder authorizes payment by authenticating to the card issuer
- Implementations:
 - Verified by Visa
 - MasterCard SecureCode
- Meets the requirements of EU [Payment Services Directive \(PSD2\)](#): strong customer authentication for most payments

3-D Secure – security analysis

- Advantages:
 - Stolen or copied card cannot be used for online fraud
 - **iframe** and the same-origin policy prevent the merchant from spoofing user input to approve the transaction
- Weaknesses:
 - **iframe** does not allow user to check the bank URL or certificate
 - **MitM attack by merchant**: spoof the card issuer iframe and use input to approve a different transaction
 - Consumers cannot travel with just the credit card; have to carry bank credential with them
- Is regulation better than risk management by financial institutions?
- Anything else?

EMV security compared to mag stripe

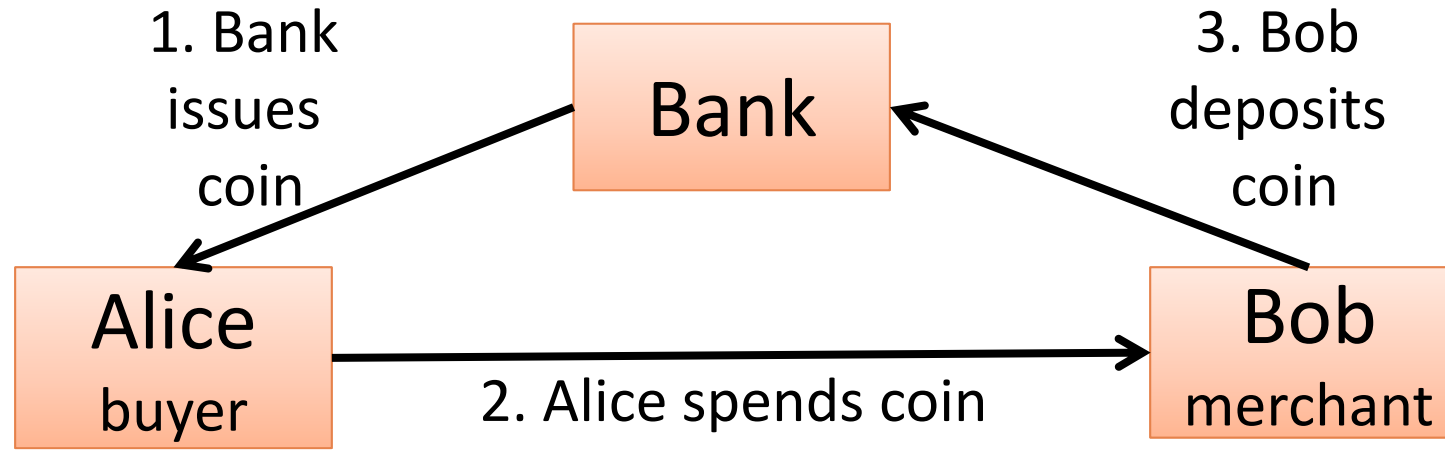
- Not possible to copy the chip
- Mag stripe can still be copied
 - Possible to create a copy of the mag stripe, which can still be used in the USA or as the fallback method after (pretended) chip failure
 - Mag stripe data can also be read from the chip
- PIN used frequently
 - easier to capture
 - Shoulder surfing, skimming, malicious payment terminals



ANONYMOUS PAYMENTS

This is what university courses taught about digital cash before Bitcoin. The idea was Proposed by David Chaum in 1982. His DigiCash product was never launched

Anonymous digital cash



- **Anonymous payment:** **issuing** and **spending** events should be **unlinkable** by bank and merchant – even if the two collude
- **Non-transferable:** coin must be deposited to bank after payment
- Uses **blind signatures:** bank signs coins without seeing them
- **Double-spending:** if buyer spends a coin twice, identity revealed

Blind signature

- Idea 1: blind signature:

Bank has an RSA signature key pair key (e, d, n) for signing 1€ coins (and different keys for 10€, 100€,...)

1. Alice creates a coin from random “serial number” SN and redundant padding required for the RSA signature;

Alice generates a random number R , computes $\text{coin} \cdot R^e \bmod n$, and sends this to the bank

2. Bank computes $(\text{coin} \cdot R^e)^d \bmod n = \text{coin}^d \cdot R \bmod n$ and sends this to Alice

3. Alice divides with R to get the signed coin $\text{coin}^d \bmod n$

→ Bank has signed the coin without seeing it and cannot link the coin to Alice

- Alice can pay 1€ to Bob by giving him the coin

– Bob deposits coin to bank; bank checks signature and only accepts each coin once

- Problem: double-spending. Buyers are anonymous; if someone pays the same coin to two merchants, who was it?

Double-spending detection

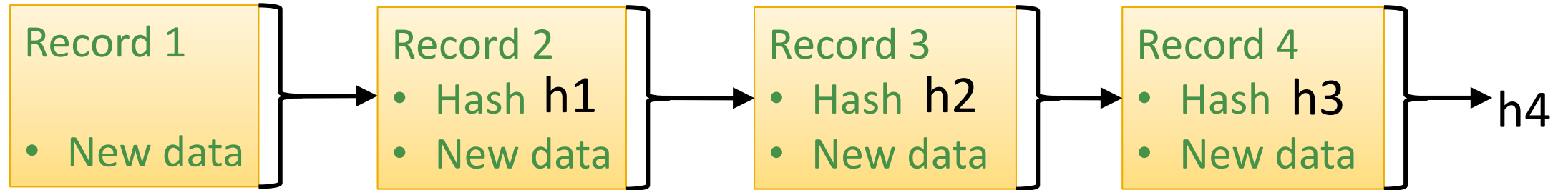
- Idea 2: double-spending detection with secret splitting
 - Alice computes $SN = h(h(A,C) \mid h(A \text{ xor "Alice"}, D))$ where A,C,D is a random number
 - After Alice has given the coin to Bob for bind signing, Bob decides which it wants to see: either $h(A,C), A \text{ xor "Alice"}, D$ or $A, C, h(A \text{ xor "Alice"}, D)$
 - Bob can check that the values are correct by recomputing SN
 - Neither choice reveals the name "Alice", but together they do
 - In double spending, Alice reveals her name with 50% probability
- Make each 1€ coin of k separately signed sub-coins
 - double-spender name revealed with probability $p = 1 - 2^{-k}$
 - Coins will be quite large: $k=128$ with 2048-bit RSA signatures makes 32kB/coin
- One more problem: What forces Alice to compute SN correctly? How can the bank check the contents of the message that it signs blindly?

Cut and choose

- How can the blind signer check correctness of the message?
 - Idea 3: cut and choose
 - Alice creates k pairs of sub-coins for signing
 - Bank asks Alice to reveal N for one sub-coin in each pair and signs the other one
 - probability of detecting malformed coins is $p = 1 - 2^{-k}$
- Alice can make anonymous payments but will be caught with probability $p = 1 - 2^{-k}$ if she tries to create an invalid coin or spend the same coin twice

BITCOIN

Background info: hash chain



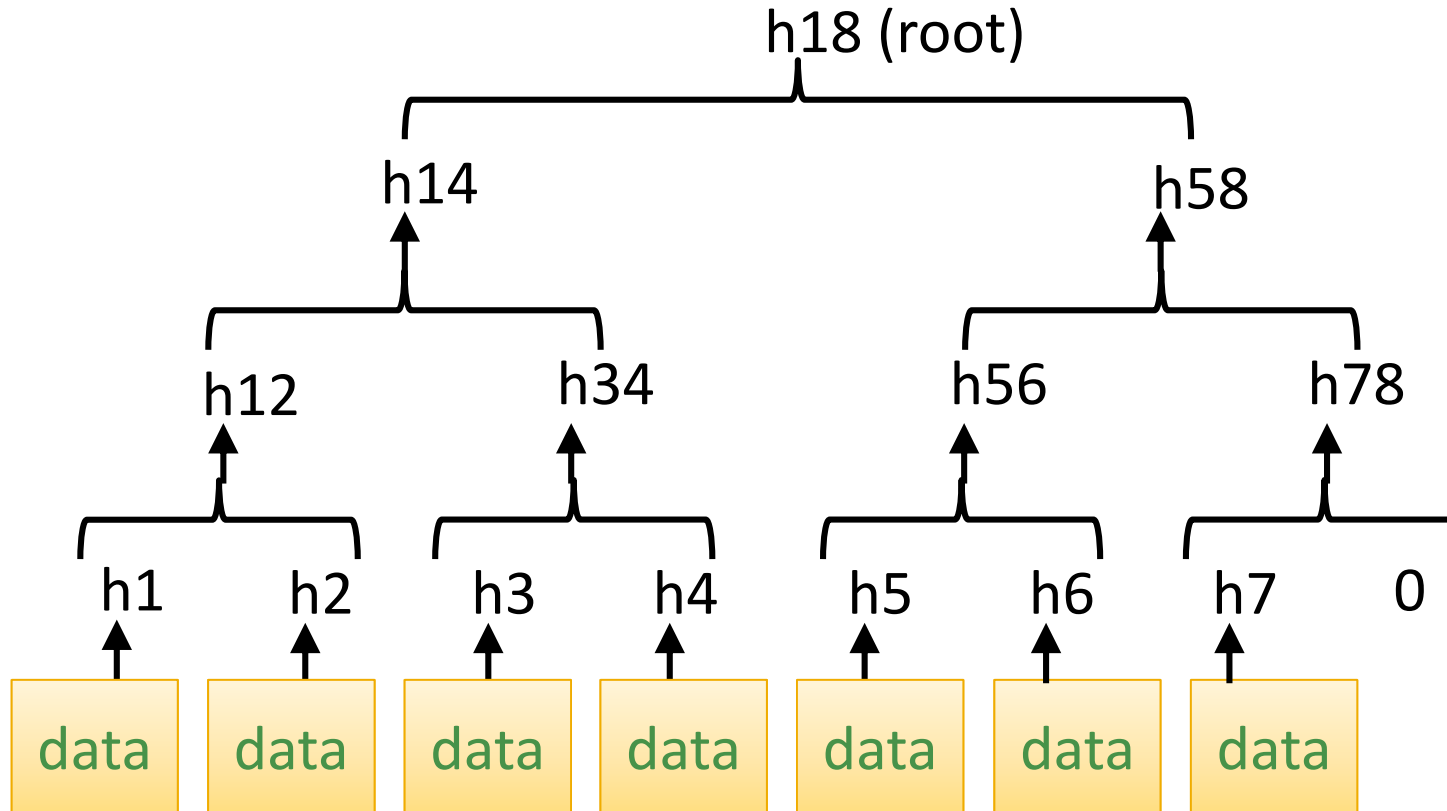
$h1 = h(\text{data1}, 0)$
 $h2 = h(\text{data2}, h1)$
 $h3 = h(\text{data3}, h2)$

...

- Cumulative hash of data: backward-linked list, with a hash value as the unambiguous reference to the previous records
- Verifying that some data is in the chain costs $O(N)$
- Appending a data item costs $O(1)$, but updating the hash costs $O(N)$ if earlier data changes

Background info: Merkle hash tree

- Binary or **n-ary tree of hash values**
- Verifying presence of data costs $O(\log N)$ in computation and communication
- Adding or updating data costs $O(\log N)$

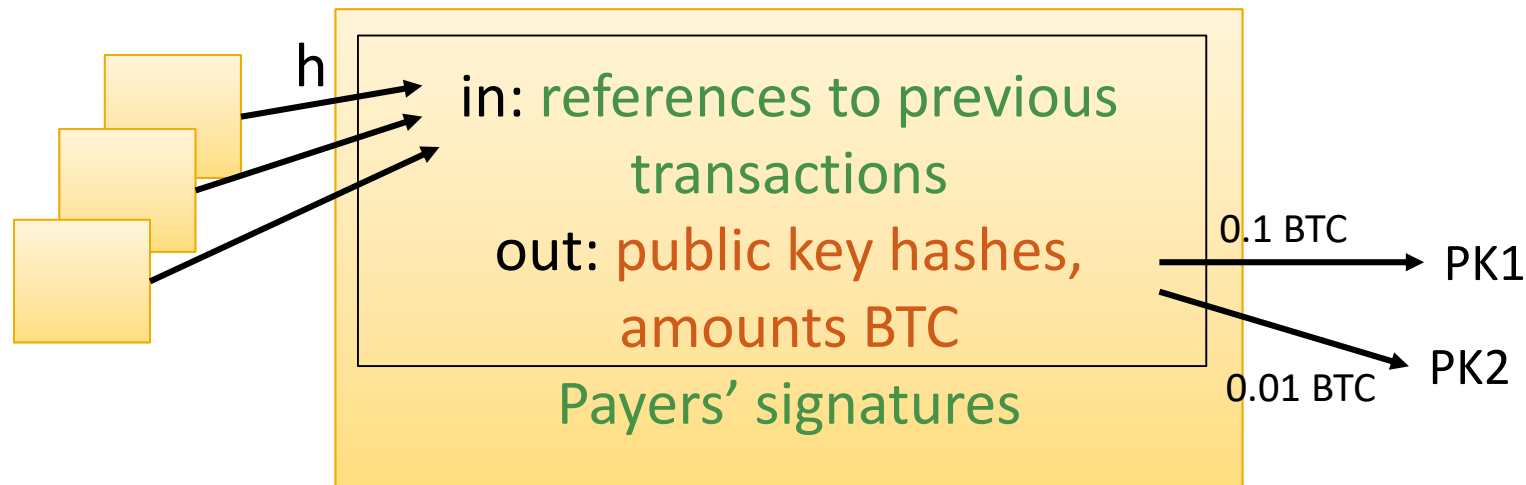


Bitcoin

- Created in 2008 by pseudonym Satoshi Nakamoto
- Transferable digital money
 - Based on cryptographic signatures and hash functions
- Public signature keys used as identities
- P2P system, no central bank or trusted issuer
- Competitive mechanism for the initial issue
 - Money distributed in proportion to wasted CPU power

Bitcoin transaction

- Direct transactions between public key pairs:
 - **Transaction** record contains (A) inputs, (B) outputs
 - **Inputs**: references to **earlier transactions** where payer received the money
 - **Outputs**: **payee public key hashes and amounts**
 - Total inputs from previous transactions must be \geq total outputs



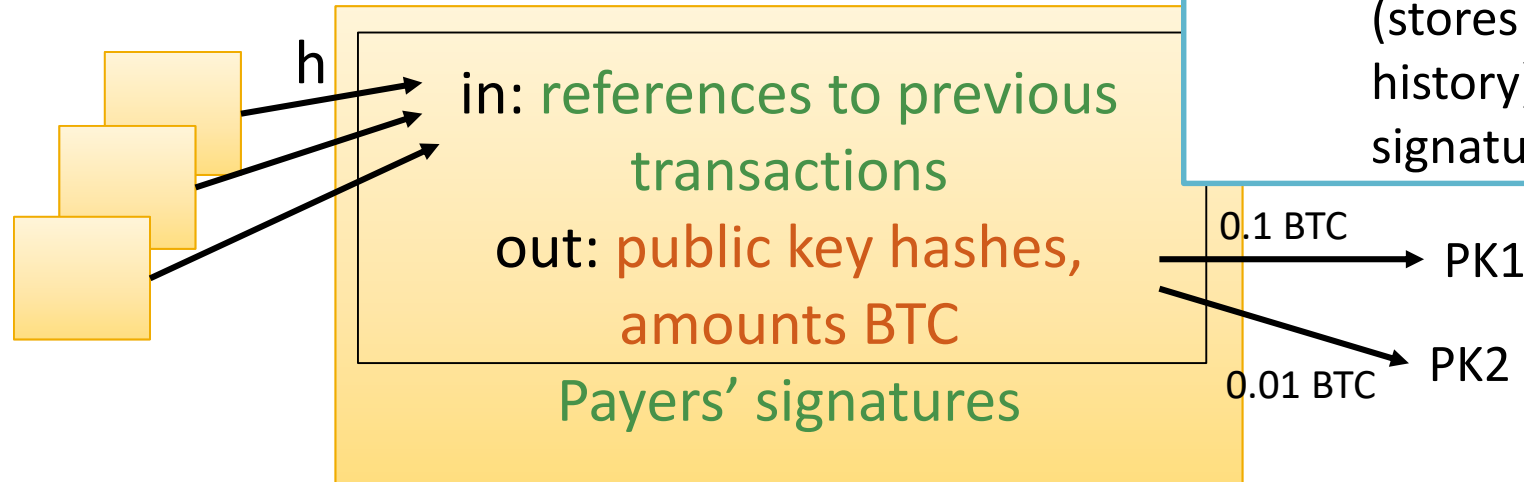
Bitcoin transaction

- Direct transactions between public key pairs:

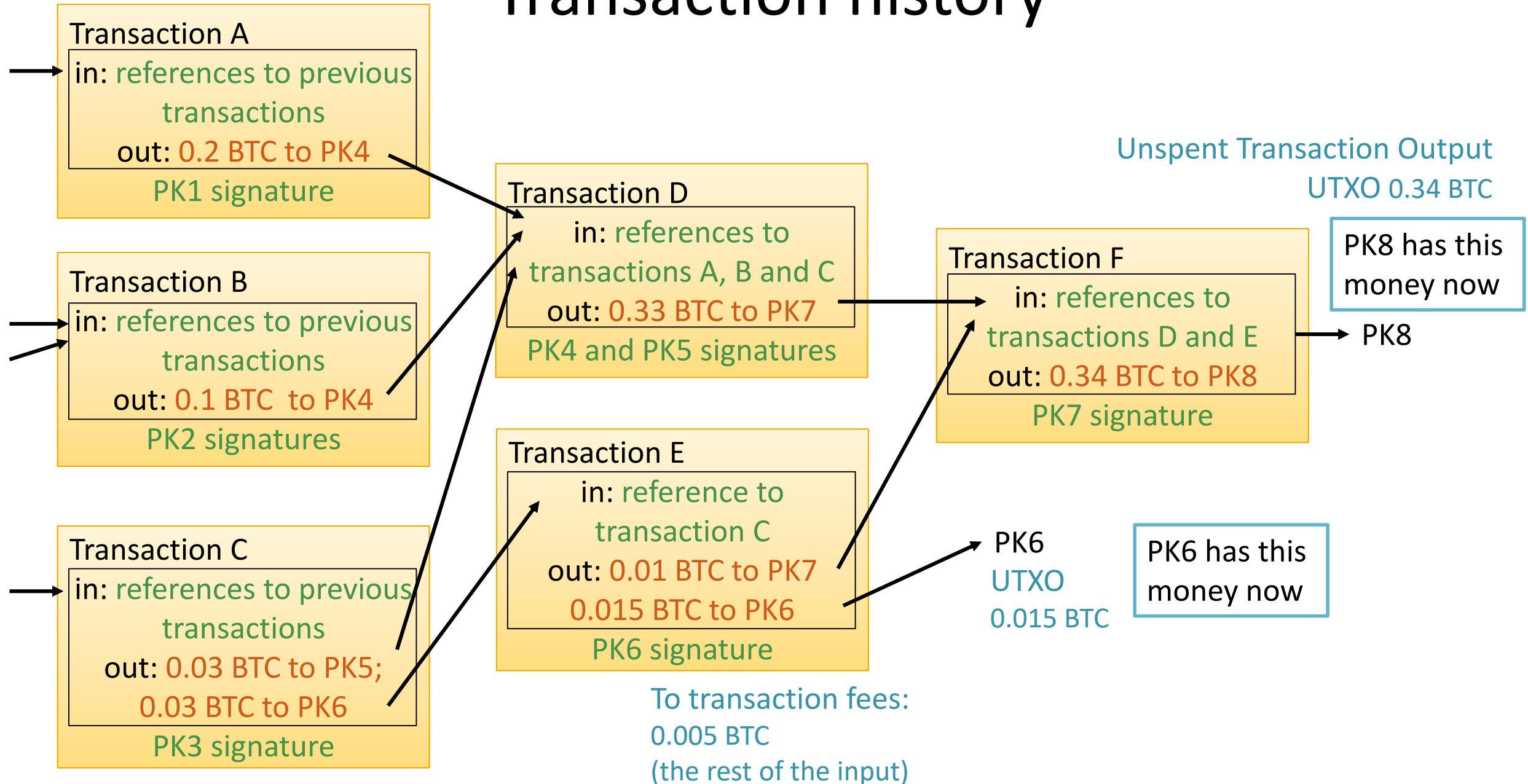
- **Transaction** record contains (A) inputs, (B) outputs
- **Inputs**: references to **earlier transactions** where p
- **Outputs**: **payee public key hashes and amounts**
- Total inputs from previous transactions must be ≥

Questions:

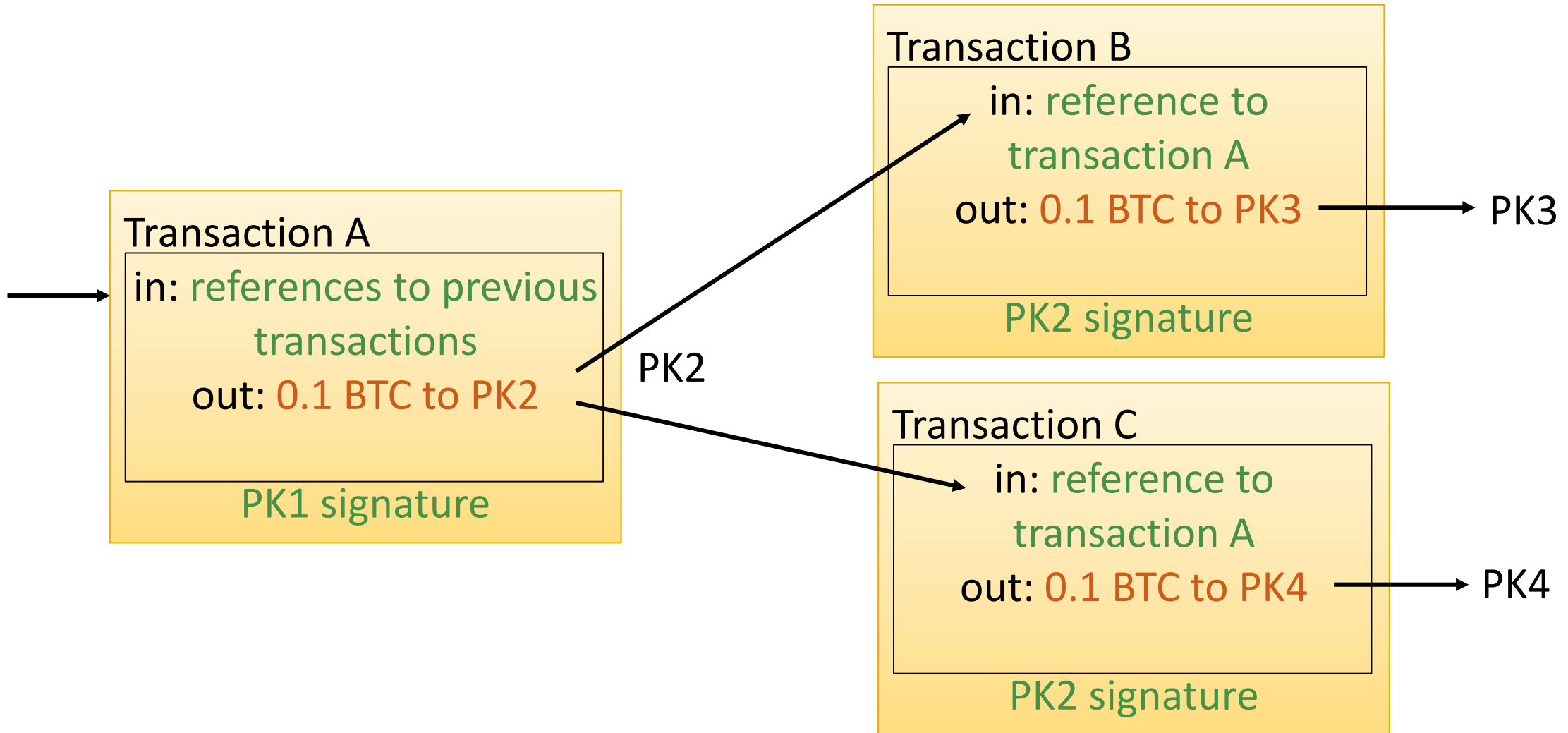
- How to bundle received small outputs, or to get change for a large input?
- What happens if outputs < inputs?
- Who keeps the books (stores the transaction history) and checks the signatures?



Transaction history



Double spending



How to prevent double spending?

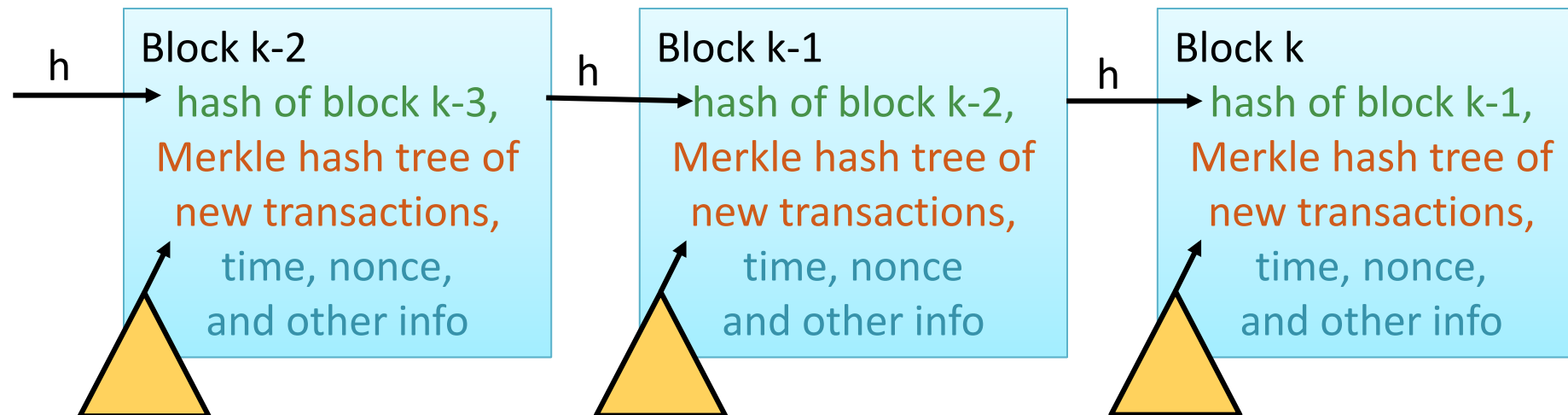
Public ledger

Public ledger: public log of all past transactions

- **Verifiable**: includes transaction signatures
 - Anyone download and verify the full history or parts of it
 - **Global consistency**: everyone agrees on what is in the log
 - **Immutable history**: nobody can change the log later
 - **Double spending** prevention: the first one of conflicting transactions accepted, others rejected
-
- **Q: Who can be trusted to maintain the ledger?**
In Bitcoin, a global P2P network

Block chain

- Public ledger in Bitcoin is implemented as a **block chain**
 - Hash chain: **block** contains hash of the previous block and **Merkle hash tree** of new transactions
- New block is added every ~10 minutes (ledger size grows)
- The latest block is a cumulative hash of all transactions ever



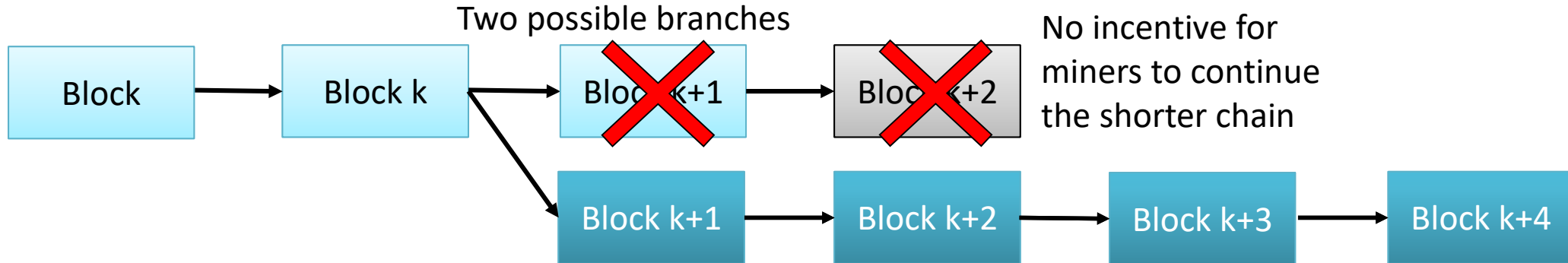
Mining

- Global P2P network propagates all transaction data and maintains the block chain
- Anyone can add blocks to the block chain
 - Must perform **proof-of-work** i.e. solve a cryptographic puzzle:
Brute-force search a nonce (any number) such that the SHA-256 hash of the block is smaller than a target value
$$h(\text{block header, nonce}) \leq \text{target 256-bit value}$$
 - The first to find a solution gets a **reward**, and everyone moves to search for the next block
 - The **difficulty** of the puzzle is adjusted to keep the average block generation time at 10 minutes

How new coins are issued

- **Reward** for generating a new block is the **transaction fees** of included transactions **plus some new BTC**
 - This is how the new coins are issued!
- **Maximum 21M BTC** to be issued over time
 - Initially 50 BTC per block
 - Currently 12.5 BTC per block
 - Halved every 210k blocks i.e. every ~4 years

Security of the block chain



- Global consistency = how to avoid chain branching?
- Rule: if the chain branches, the longest branch wins
 - If there are two conflicting blocks, the next block determines the winner
 - Payee should wait for 6 new blocks (~1 hour) before considering the transaction final
- If someone controls more than 50% of the global hash rate, they can modify the history and double spend

Summary of Bitcoin data structures

- **Block chain**: linear hash chain of blocks
 - History starts at **genesis block**; new block added every 10 minutes
 - Each block contains a **Merkle hash tree**: hashes of all (or many) transactions since the previous block
 - However, block size is limited to 1 MB; if block is full, miners select which transactions to include
- **Transaction history**: directed acyclic graph of signed transactions
 - Constructed by any node in the P2P network from the transactions in the blockchain to determine who has the coins now
 - Everyone should download, keep up to date, and verify the entire global transaction history

Bitcoin philosophy

Extra
material

- **Anonymous transactions** like cash money?
 - Signature keys, not users, own the money
 - Not exactly anonymous: input and output are linkable
- **Digital equivalent of gold:**
 - Limited supply on earth
 - Cannot be controlled or inflated by any government
 - Different from current monetary policy (e.g. quantitative easing)
 - However, competing electronic currencies are easy to create

Potential problems with Bitcoin

Extra
material

- Volatile exchange rate with other currencies
- Unlike precious metals, new competing currencies can be created
- Security is based on wasting hardware and energy in hash computation (for proof of work)
- Security reduced if there are many such currencies (mining capacity may move around)
- Favorite currency for drug trade and other crime – but maybe this is why it will succeed?
- Slow global transaction rate limits applications
- Tax authorities will be interested
- Speculative bubble?

Security issues with Bitcoin

- **Block size limit** caps global transaction rate to ~7 per second; small transaction may never succeed (e.g. buying coffee)
- 10..60 minutes wait for transaction confirmation
- A few **large Chinese mining pools** dominate the P2P network
- **Malware** attacks against wallets, **bank robbery** by hackers
- **No way to reverse a transaction** without the payee's cooperation
 - Stolen BTC are gone for good
- Governments may want to take control
- Competing digital currencies are easy to create and could have stronger business and political support

Why would anyone use Bitcoin?

Extra
material

Even the most dysfunctional
money is better than not
having a means for economic
exchange



Blockchains beyond Bitcoin

Extra
material

- Cryptocurrencies enabling financial services:
 - Speeding up international money transfers
 - Local special-purpose currencies
- Blockchains have other applications:
 - Public event logs for accountability
 - Global consensus / consistency as a service
 - Highly distributed trust model
- Consensus without wasting all that energy:
 - Proof of stake, proof of elapsed time
 - Byzantine consensus in managed systems

SUMMARY

List of key concepts

- EMV, issuer, card holder, ICC, offline card authentication, SDA, DDA, CDA, cardholder verification, online vs. offline PIN, offline approval or authorization, online authorization, contactless payment, NFC, risk management, soft and hard limits, CVV2, card skimmer, mag stripe
- Anonymous payment, unlinkability, transferable money, double spending
- Hash chain, Merkle tree,
- Bitcoin, transaction, inputs, outputs, transaction fee, transaction history, public log, global consistency, block chain, block, P2P network, mining, proof of work, cryptographic puzzle, puzzle difficulty, chain branching, longest chain, transaction confirmation, hash rate

Reading material

- Ross Anderson: Security Engineering, 2nd ed., chapter 10
- Interesting reading online:
 - University of Cambridge Security Group:
<http://www.cl.cam.ac.uk/research/security/banking/>
 - EMV in nutshell, <https://www.cs.ru.nl/E.Poll/papers/EMVtechreport.pdf>
 - BitCoin wiki: https://en.bitcoin.it/wiki/Main_Page

Exercises

- What are the main threats in

- a) online card transactions?
- b) POS transactions?
- c) ATM cash withdrawals?

What differences are there in the way credit cards and bank debit cards address these threats?

- Could you (technically) use bank cards or credit cards

- a) as door keys?
- b) as bus tickets?
- c) for strong identification of persons on the Internet?

(This question may require quite a bit of research.)

- How could a malicious merchant perform a man-in-the-middle attack against EMV chip-and-PIN transactions?
- When a fraudulent bank transaction occurs, who will suffer the losses? Find out about the regulation and contractual rules on such liability.
- Bank security is largely based on anomaly detection and risk mitigation. In what ways could a bank reduce the risk of fraud in mag-stripe or chip-and-PIN payments?
- Even though DigiCash coins are unlinkable, what ways are there for the merchant or bank (or them together) to find out what Alice buys?
- How anonymous are Bitcoin payments?
- Find a Bitcoin block explorer web site with the full transaction record and browse around. Find the latest blocks and transactions, and the first block ever. See how the mining difficulty has changed over time.