How to Store and Use Bitcoins

SIMPLE LOCAL STORAGE

To spend a Bitcoin, you need to know:

1. Some info from the public blockchain,

and

2. The owner’s secret signing key

So it’s all about key management.

Goals

Availability: Being able to spend your coins when you want to.

Security: Making sure nobody else can spend your coins.

Convenience: Managing your keys (and thus your coins)

Achieving all the three simultaneously could be a challenge!

Simplest approach

Store key in a file, on your computer or phone

Very convenient.

As available as your device.

- device lost/wiped ⇒ key lost ⇒ coins lost

As secure as your device.

- device compromised ⇒ key leaked ⇒ coins stolen

Wallet software

Software used when bitcoins are stored locally on a device

Wallet software → software that:

- Keeps track of your coins.
- Manage details of your keys.
- Provides nice user interface.

Nice trick: use a separate address/key for each coin.

benefits privacy (looks like separate owners)

wallet can do the book-keeping, user needn't know

Encoding addresses

Encode as **text string**: base58 notation

```
123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz
```

or use **QR code**

HOT AND COLD STORAGE

Hot storage

online

convenient but risky

Cold storage

offline

archival but safer

separate keys

Hot storage

online

hot secret key(s)

cold address(es)

Cold storage

offline

cold secret key(s)

hot address(es)

payments

Hot storage

Cold storage

online

hot secret key(s)

payments

cold address(es)

offline

Problem:

Want to use a new address (and key) for each coin sent to cold

But how can hot wallet learn new addresses if cold wallet is offline?

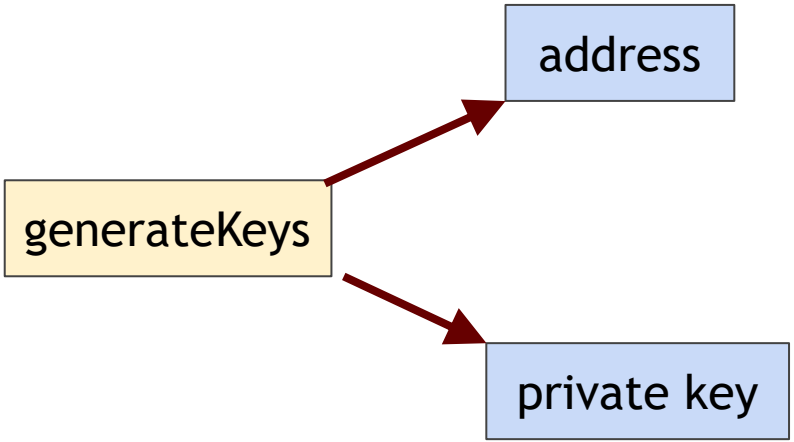
Awkward solution:

Generate a big batch of addresses/keys, transfer to hot beforehand

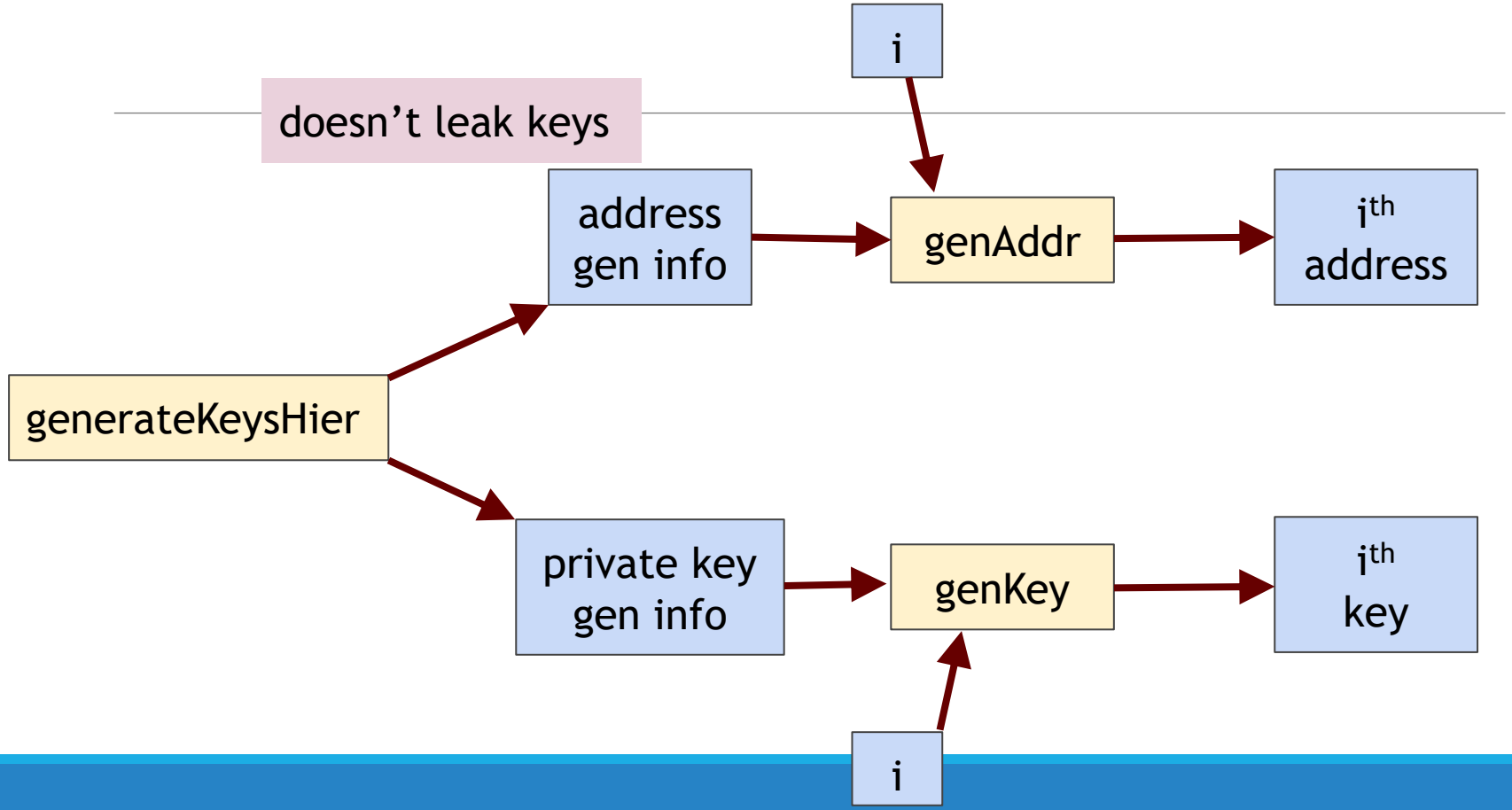
Better solution:

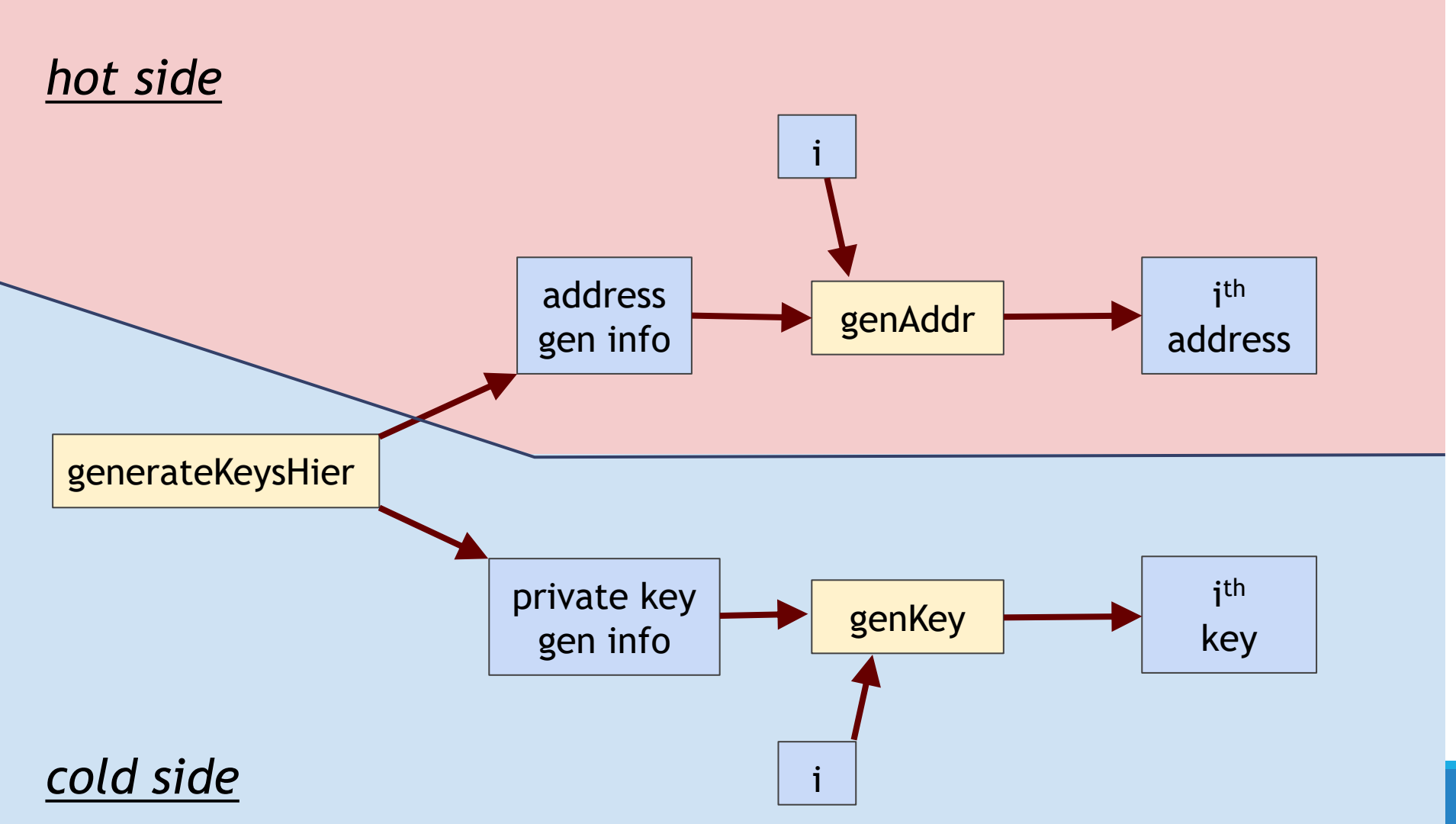
Hierarchical wallet

Regular key generation:



Hierarchical key generation:




hot side
i
address
gen info
genAddr
ith
address
generateKeysHier
private key
gen info
genKey
ith
key
i
cold side

How to store cold info

(1) Info stored in device, device locked in a safe

(2) “Brain wallet”

encrypt info under passphrase that user remembers

(3) Paper wallet

print info on paper, lock up the paper

(4) In “tamperproof” device

device will sign things for you, but won’t divulge keys

In general, a combination of one more more of these techniques can be used

ONLINE WALLETS AND EXCHANGES

Online wallet

Like a local wallet

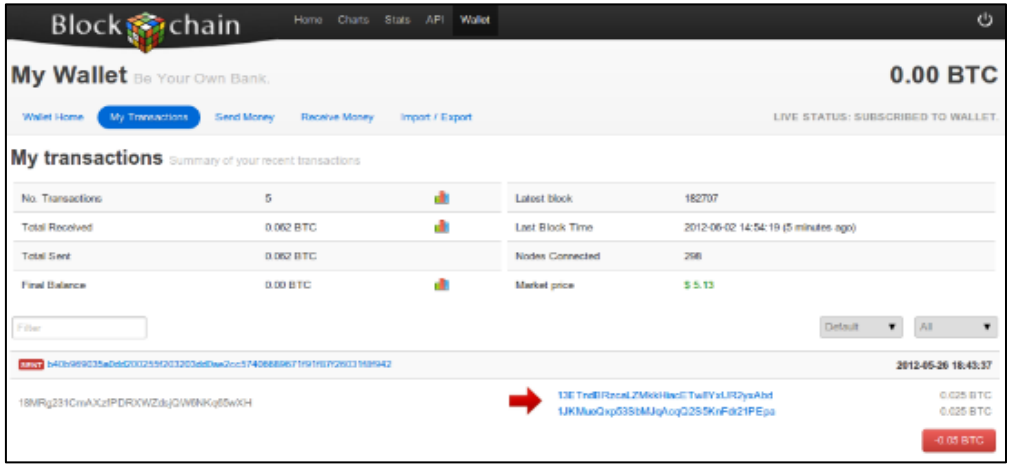
but “in the cloud”

Runs in your browser

Site sends code

Site stores keys

You log in to access wallet



Online wallet tradeoffs

Convenient: nothing to install, works on multiple devices

Security worries: vulnerable if site is malicious or compromised

Ideally, site is run by security professionals

Bank-like services

You give the bank money (a “deposit”)

Bank promises to pay you back later, on demand

Bank doesn’t actually keep your money in the back room

- Typically, bank invests the money
- Keeps some around to meet withdrawals (“fractional reserve”)

Bitcoin Exchanges

Accept deposits of Bitcoins and fiat currency ($, €, …)

- Promise to pay back on demand

Lets customers:

- Make and receive Bitcoin payments
- Buy/sell Bitcoins for fiat currency
  - typically, match up BTC buyer with BTC seller

What happens when you buy BTC

Suppose my account at Exchange holds $5000 + 3 BTC and I use Exchange to buy 2 BTC for $580 each

- **Result**: my account holds $3840 + 5 BTC

Note: no BTC transaction appears on the blockchain

Only effect: Exchange is making a different promise now

Exchanges: Pros and Cons

Pros:

1. Connects BTC economy to fiat currency economy
2. Easy to transfer value back and forth

Cons:

1. Risk - same kinds of risks as banks


東京で MT.GOX のデモ
へ参加してください。
東京都渋谷区渋谷
２丁目１１−５
MTGOX
WHERE IS
OUR MONEY

Bank Regulation

For traditional banks, government typically:

1. *Imposes minimum reserve requirements*
   - *Must hold some fraction of deposits in reserve*
2. *Regulates behavior, investments*
3. *Insures depositors against losses*
4. *Acts as lender of last resort*

Proof of Reserve

Bitcoin exchange can prove it has fractional reserve.

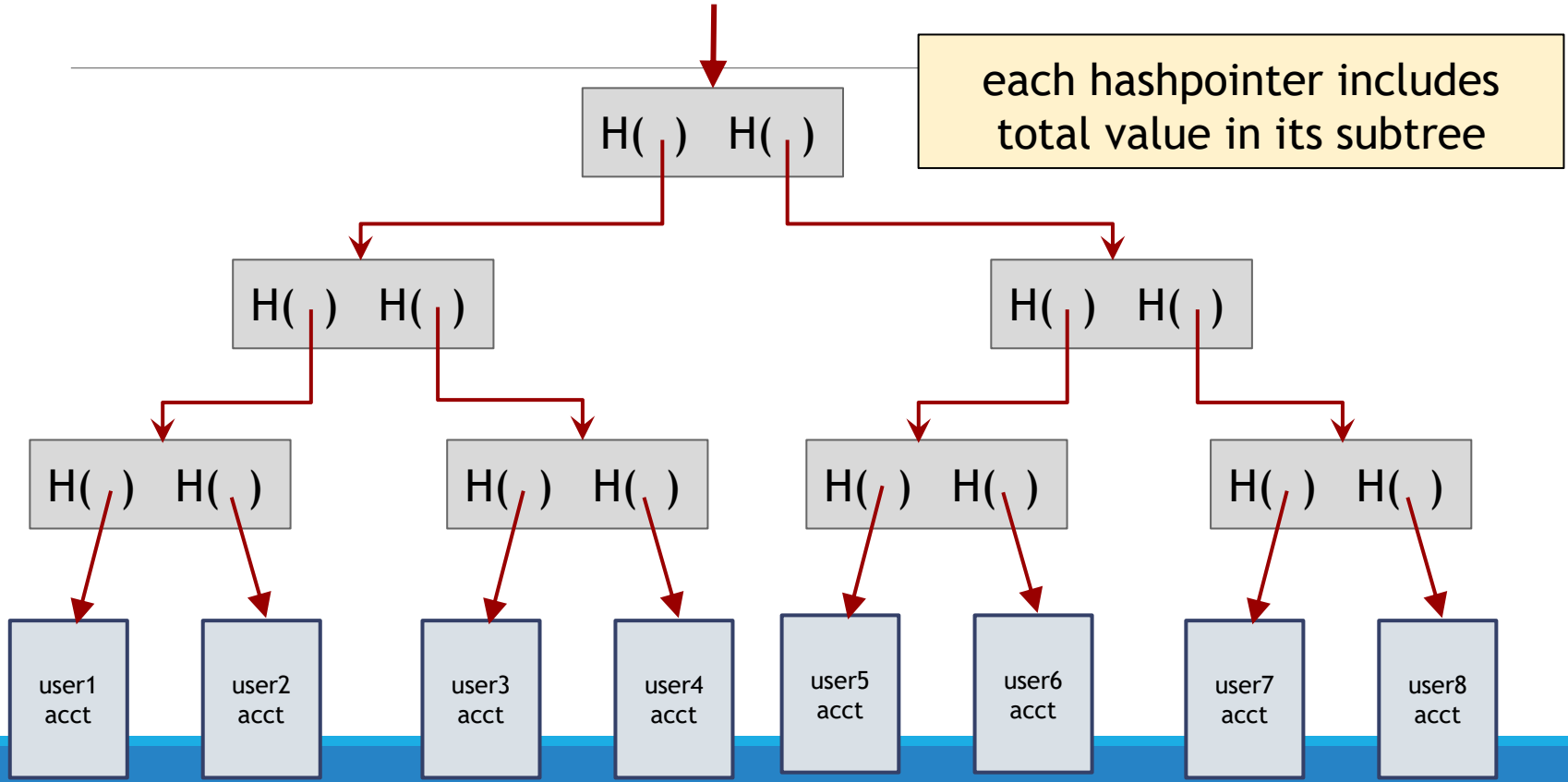
- Fraction can be 100%

Prove how much reserve you're holding:

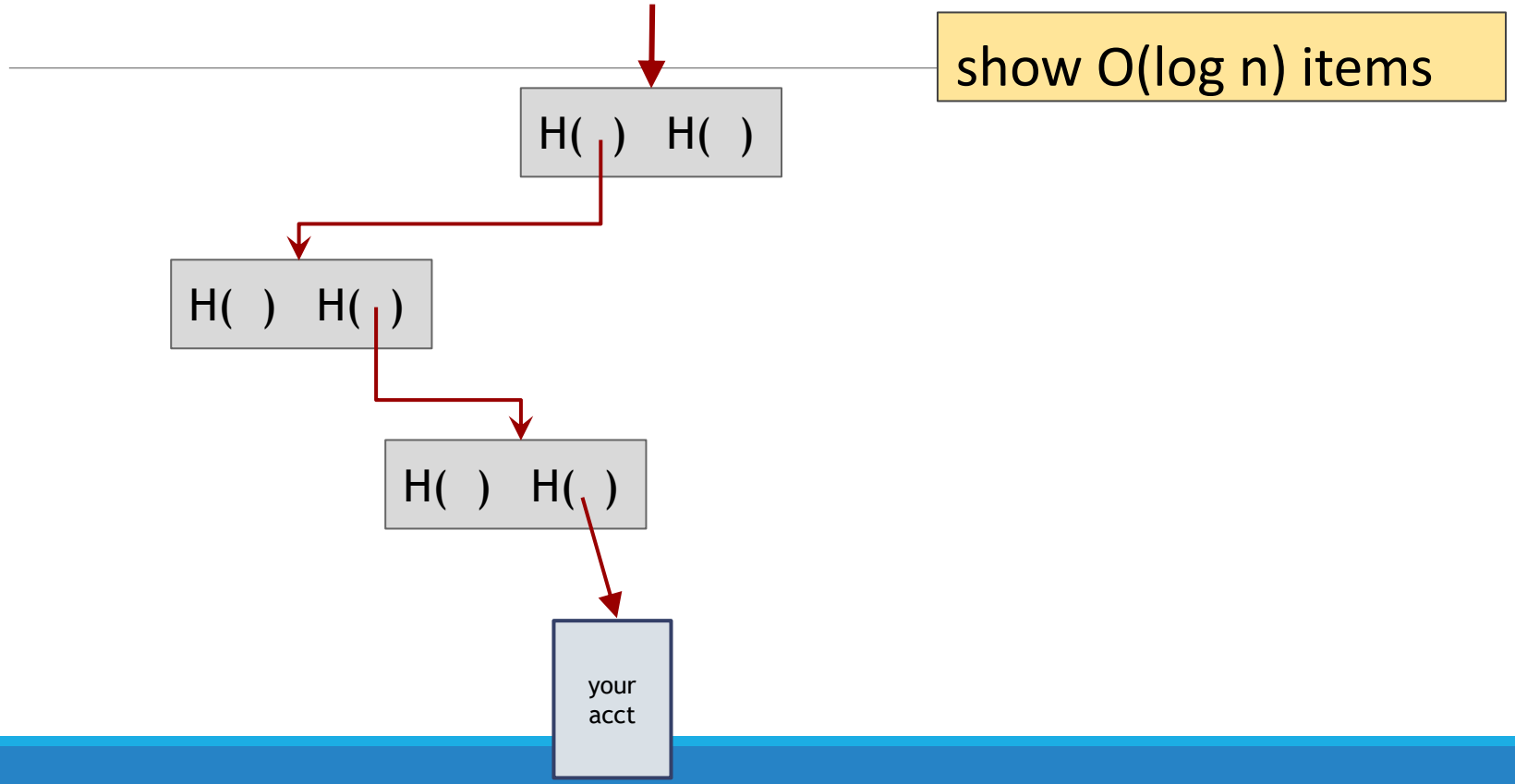
- Publish valid payment-to-self of that amount
- Sign a challenge string with the same private key

Prove how many demand deposits you hold: ...

Merkle tree with subtree totals



Checking that you're represented in the tree



Proof of Reserve

1. Prove that you have at least X amount of reserve currency

2. Prove that customers have at most Y amount deposited

So reserve fraction ≥ X / Y

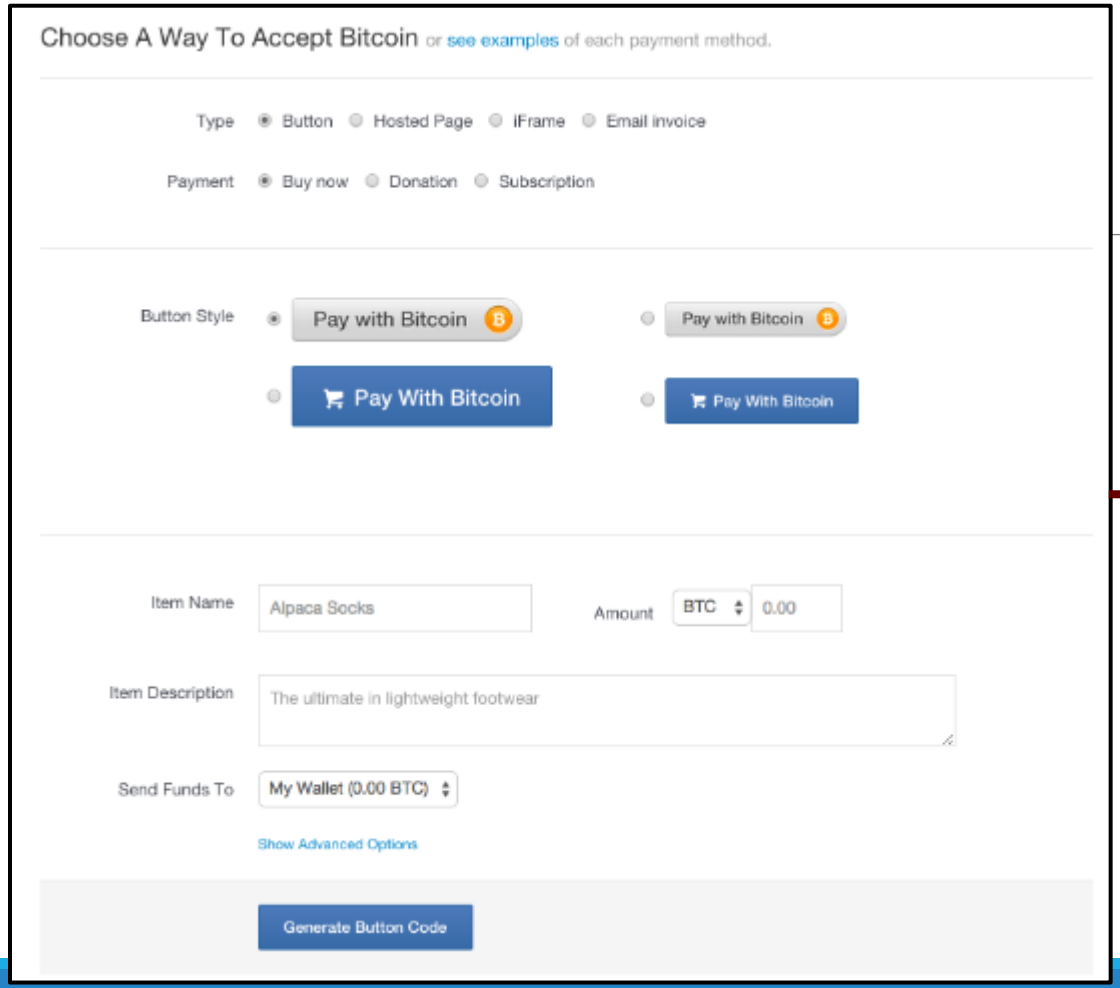
PAYMENT SERVICES

Scenario: merchant accepts BTC

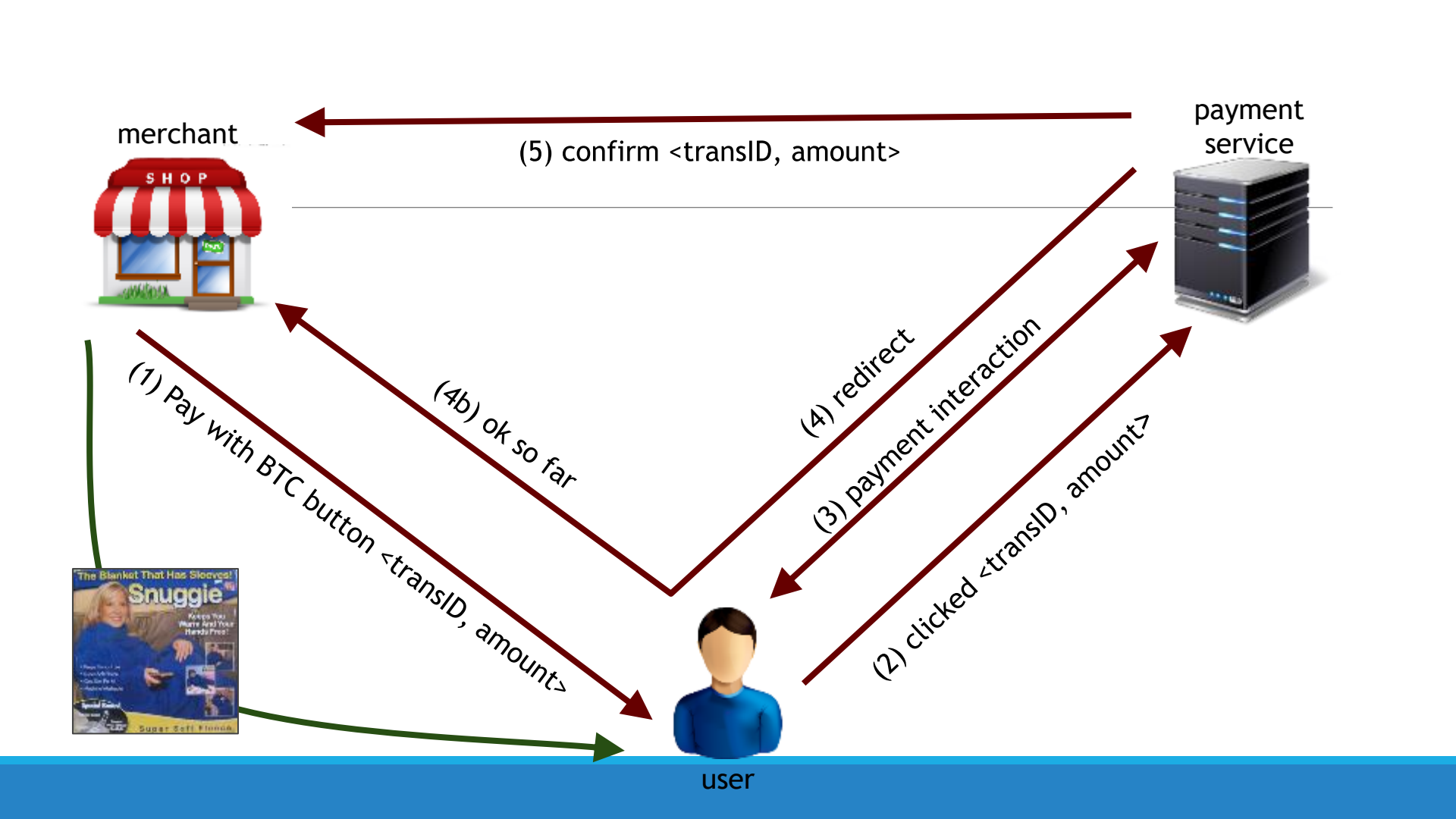
Customer wants: to pay with Bitcoin

Merchant wants:

- To receive dollars
- Simple deployment
- Low risk (tech risk, security risk, exchange rate risk)



HTML for payment button


merchant
payment service
(5) confirm <transID, amount>
(4b) ok so far
(4) redirect
(3) payment interaction
(2) clicked <transID, amount>
(1) Pay with BTC button <transID, amount>
user

End result

customer: pays Bitcoins
merchant: gets dollars, minus a small percentage
payment service:
 gets Bitcoins
 pays dollars (keeps small percentage)
 absorbs risk: security, exchange rate
 needs to exchange Bitcoins for dollars, in volume

TRANSACTION FEES

Transaction Fees

Recall:

- Transaction fee = value of inputs - value of outputs
- Fee goes to miner who records the transaction

Interesting economics, discussed in later lecture

How are transaction fees set today?

Transaction Fees

It costs resources for

- Peers to relay your transaction
- Miner to record your transaction
- Transaction fee compensates for (some of) these costs

Generally, higher fee means transaction will be forwarded and recorded faster!

Consensus Fee Structure

Current consensus fees:

No fee if:

- Transaction less than 1000 bytes in size,
- All outputs are 0.01 BTC or larger, and
- Priority is large enough

Priority = (sum of inputAge*inputValue) / (trans size)

Otherwise fee is 0.0001 BTC per 1000 bytes

Approx transaction size: $148\ N_{inputs} + 34\ N_{outputs} + 10$

Consensus Fee Structure

Most miners enforce the consensus fee structure.

If you don’t pay the consensus fee, your transaction will take longer to be recorded!

Miners prioritize transactions based on fees and the priority formula.

CURRENCY EXCHANGE MARKETS

Basic Market Dynamics

Market matches buyer and seller

Large, liquid market reaches a consensus price

Price set by supply (of BTC) and demand (for BTC)

Supply of Bitcoins

Supply = coins in circulation (+ demand deposits?)

Coins in circulation: fixed number, currently ~13.1 million

When to include demand deposits?

- When they can actually be sold in the market.

Demand for Bitcoins

BTC demanded to mediate fiat-currency transactions

- Alice buys BTC for $
- Alice sends BTC to Bob
- Bob sells BTC for $

BTC “out of circulation” during this time

BTC demanded as an investment

- If the market thinks demand will go up in future

Demand for Bitcoins

Simple model of transaction-demand

T = Total transaction value mediated via BTC ($ / sec)

D = Duration that BTC is needed by a transaction (sec)

S = Supply of BTC (not including BTC held as long-term investments)

$\frac{S}{D}$ Bitcoins become available per second

$\frac{T}{P}$ Bitcoins needed per second

Equilibrium:

$$P = \frac{TD}{S}$$

References

CS 4593/6463 – Bitcoins and Cryptocurrencies, Prof. Murtuza Jadliwala, University of Texas, San Antonio

Note: most of the slides used in this course are derived from those available for the book "Bitcoins and Cryptocurrencies Technologies – A Comprehensive Introduction", Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller & Steven Goldfeder, 2016, Princeton University Press.

SPLITTING AND SHARING KEYS

Current key storage schemes

Problem: single point of failure

Trivial solution: make multiple copies or backup

- **Advantage**: Availability improves
- **Disadvantage**: Security of stored keys is worst (multiple avenues to steal)

Question: Can we improve both availability and security?

Answer: Surprisingly, yes! Using some cute cryptographic / mathematical tricks → Secret Sharing!

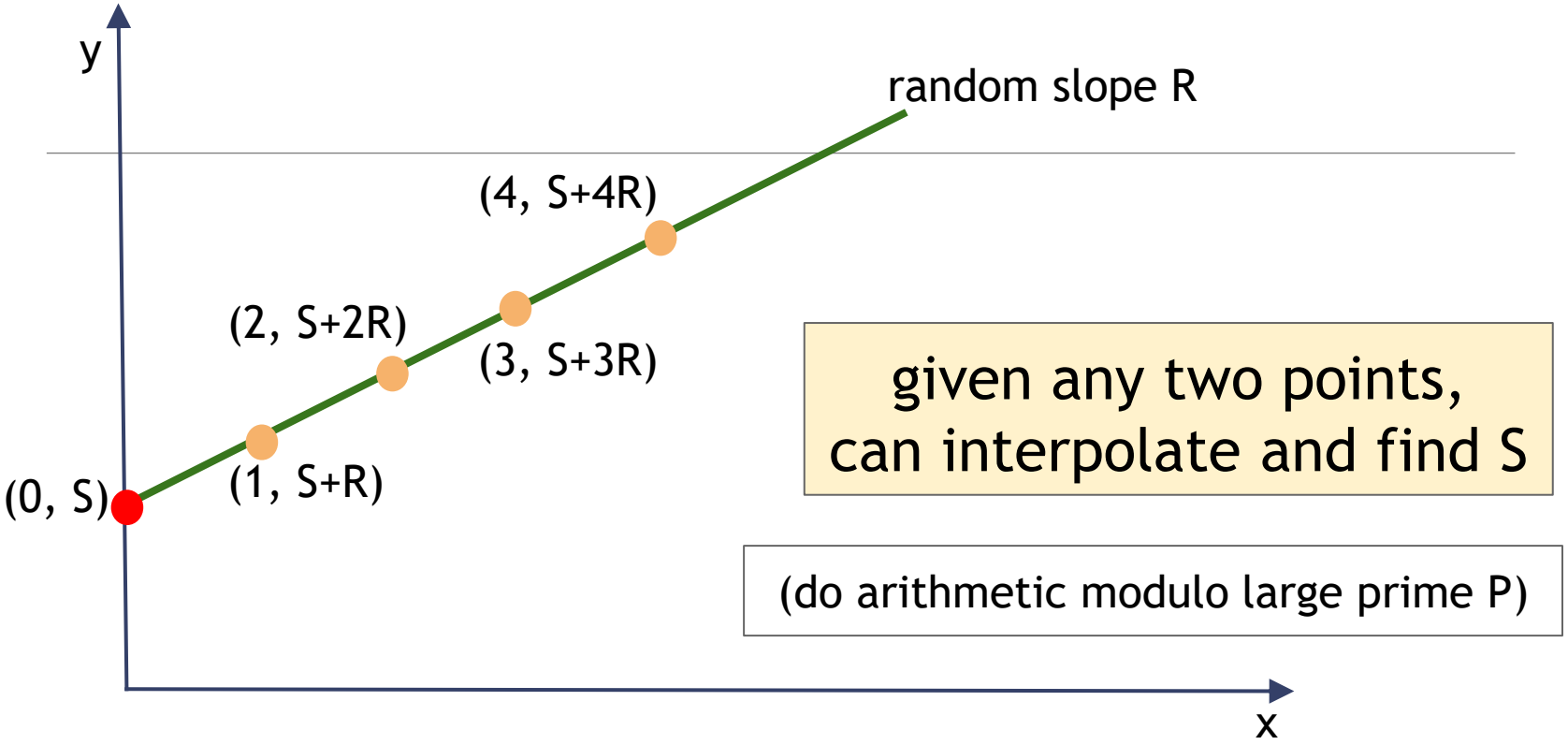
Secret sharing

Idea: split secret into N pieces, such that
given any K pieces, can reconstruct the secret
given fewer than K pieces, don't learn anything

Example: N=2, K=2
P = a large prime
S = secret in [0, P)
R = random in [0, P)

split:
$X_1 = (S+R) \bmod P$ $X_2 = (S+2R) \bmod P$

reconstruct:
$(2X_1-X_2) \bmod P = S$



given any two points,
can interpolate and find S

(do arithmetic modulo large prime P)

Secret sharing

Equation	Random parameters	Points needed to recover S
$(S + RX) \bmod P$	R	2
$(S + R_1X + R_2X^2) \bmod P$	R_1, R_2	3
$(S + R_1X + R_2X^2 + R_3X^3) \bmod P$	R_1, R_2, R_3	4

etc.

support K-out-of-N splitting,
for any K, N

Secret sharing

Good: Store shares separately, adversary must compromise several shares to get the key.

Bad: To sign, need to bring shares together, reconstruct the key. ⇐ vulnerable

Multi-sig

Recall multi-sig from Lecture 3.

Lets you keep shares apart, approve transaction without reconstructing key at any point.

Example

Andrew, Arvind, Ed, and Joseph are co-workers.
Their company has lots of Bitcoins.

Each of the four generates a key-pair,
puts secret key in a safe, private, offline place.

The company's cold-stored coins use multi-sig, so that three of the four keys must sign to release a coin.