CSE 3550/5000: Blockchain Technology

Lecture 10
Wallets and Key Management

Ghada Almashaqbeh
UConn - Spring 2026

Outline

- Key management in cryptocurrencies.
  - Hot and cold storage.
  - Hierarchical wallets.
  - Cold info storage.
  - Splitting and sharing keys.
  - Online wallets.

Spending Coins

- Recall that coins in any cryptocurrency are virtual.
  - Strings of bits.
- Spending any amount of coins requires:
  - Some public information from the blockchain.
  - The secret key associated with the address (or public key) that owns the coins.
    - To produce a digital signature proving ownership of coins.
- Losing the secret keys means losing these coins; no one will be able to spend them.

Key Management

- Storage and retrieval of secret keys.
  - Involves also when and how to generate new keys for future transactions.
- Goals:
  - Security; only the legitimate owner gets to spend a given coin.
  - Availability; owners can spend their coins whenever they wish.
  - Usability; it is relatively easy for the average user to store/retrieve/use secret keys.
- We will focus on Bitcoin in the slides.
  - The concepts we will study can be applied to any other cryptocurrency.

Wallet Software

- User (or client) software that keeps track of coins that a client owns.
- Provides a convenient user interface to simplify operations.
  - Issuing transactions.
  - Tracking total balance.
  - Generate keys when needed.
  - Bookkeeping of these keys.
- Encode addresses as text strings (Base58) or in the form of QR codes.
  - Simplifies sharing addresses with others.
- A large number of wallets is available.
  - Different flavors; desktop, mobile, web applications, etc.
  - Different vendors; metamask, coinbase, etc.
  - Security is a driving factor for which one to choose.

Naive Solution

- Store the keys in a file on your laptop or smartphone.
  - Security is tied to your device; breaking into the device allows an attacker to steal your keys (and so your coins).
- This is called hot storage.
  - Easy to use but risky.

Hot vs. Cold Storage I

- Hot storage.
  - Storing secret keys on a device that is connected to the Internet and used frequently for variety of applications.
  - Usable/coventient.
- Cold storage.
  - Offline storage, like on a machine or memory device that is stored in some safe location.
  - Less convenient.
- Good practice: The majority of the coins are in the cold storage with a few in the hot storage.

Hot vs. Cold Storage II

- Seperate keys are needed for each.
  - Otherwise, compromising hot storage will compromise cold storage.
- But both should be aware of their addresses to allow transferring currency.
  - Cold storage stores its cold secret keys, cold addresses, and hot public addresses.
  - Hot storage stores hot secret keys and public addresses, as well as cold public addresses.

New Addresses for Cold Storage?

- To break transaction linkability, a good practice in Bitcoin is to create a new address for each new transaction.
- How can a hot wallet learn the addresses of a cold wallet?
  - Remember cold wallet is offline, no internet connectivity.
- Even generating a large chunk of addresses at the beginning will not work.
  - Cold storage needs to connect whenever a new batch of addresses is generated.

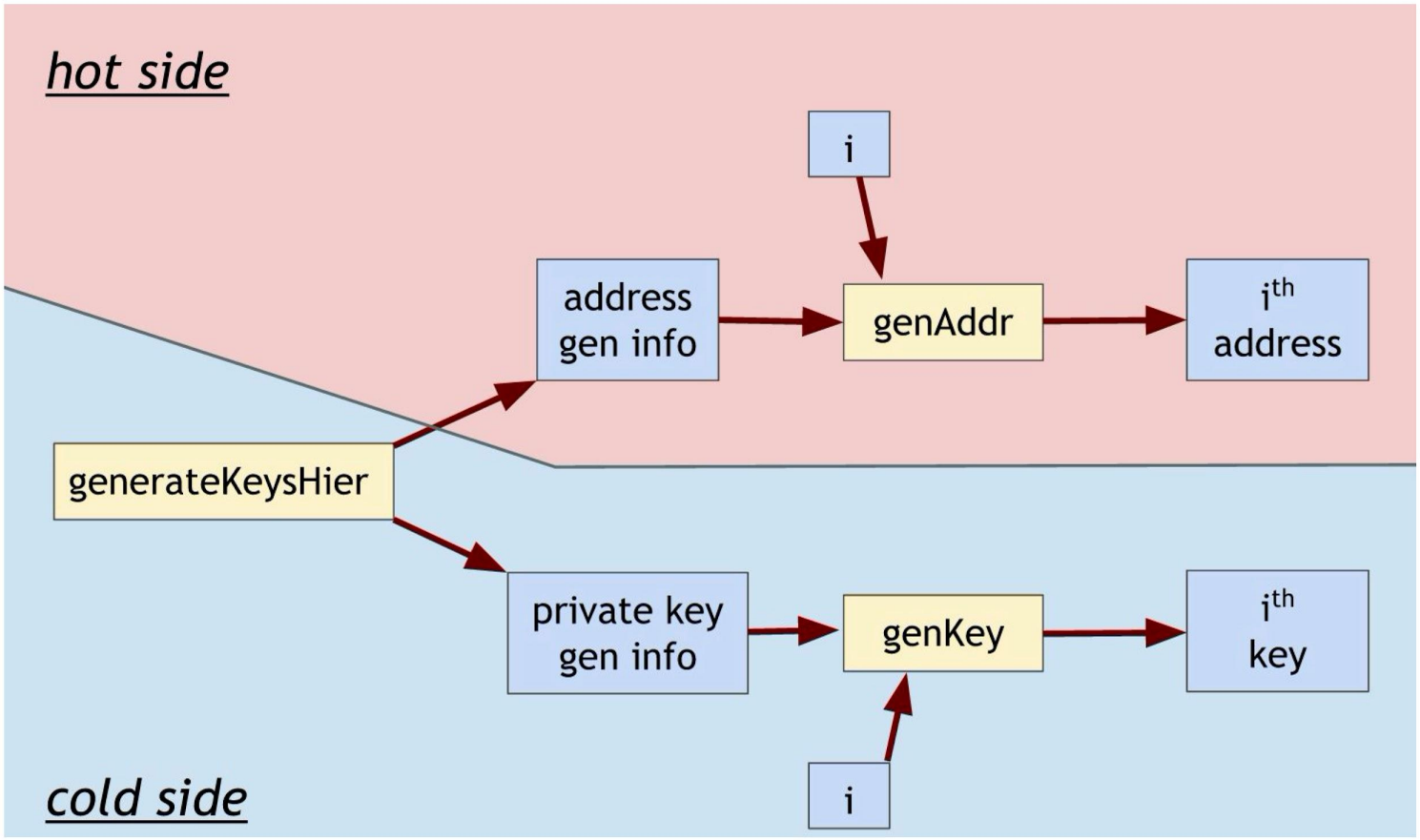
Hierarchical Key Generation I

- Allows cold storage to generate an infinite number of key pairs (or addresses).
- Usual keyGen algorithms generate a single key pair; public and private.
- Hierarchical keyGen instead generates some info that allows generating all future keys in a deterministic way.
  - Usually called master public key and master secret key.
- The master public key does not reveal any info about the master secret key or all future generated secret keys.

Hierarchical Key Generation II

- By having the operation indexed with some integer *i*, the master key allows generating the i^{th} future key.
  - Works for both public and secret keys.
- Hot storage has a copy of the master public key, while the master secret key is known only to the cold storage.
- Not all digital signature schemes support this hierarchical approach.
  - ECDSA supports that, details come later.

Hierarchical KeyGen - Pictorially



*From Chapter 5, Bitcoin and Cryptocurrency Technologies book.

Hierarchical KeyGen in ECDSA (BIP32)

- A group G, in which DDH is believed to be hard, of order p (where p is some large prime number) and a group generator g.
- KeyGen is extended into 3 algorithms (H below is a hash function modeled as a random oracle):
  - keyGenHer(1^n): msk = x, mpk = g^x, Hash function H.
    - n is the security parameter.
    - x is some integer selected at random from Z_p.
  - addrGen(mpk, i): r = H(i || mpk), pk_i = mpk . g^r = g^{x+r}, $addr_i$= H(pk_i)
  - keyGen(msk, i): r = H(i || mpk), sk_i = msk + r = x+r
- A hot wallet will be able to generate the ith address (or public key) and a cold storage will be able to generate the ith private key corresponding to this address without any interaction.

Security Issue in the Previous Scheme

- No forward secrecy.
  - Given i, sk_i, mpk, it is easy to determine msk.
    - Assume an attacker compromised the hot wallet (got hold of mpk), and one secret key of the cold storage has been leaked.
    - This allows the attacker to compute msk, and hence, all previous as well as future secret keys.
    - ***Do you see why?***
- Source of vulnerability; the way the randomness r is computed.
- Many modifications were proposed to fix this problem.
  - Out of scope.
- This shows how proposing security countermeasure is not an easy task!
  - Put differently, the attacker capability must be carefully defined and achieving security under a restricted attacker model offers a limited security level.

Cold Info Storage - different types

- (1) Store secret keys on a device, and lock that device in a safe.
- (2) Brain wallets:
  - Encrypt under a password that the user memorizes.
  - Security issue: offline password cracking.
- (3) Paper wallet:
  - Print the key material and store the paper in a safe.
- (4) Tamper resistant hardware device:
  - Either the device generates the secret key and stores it, or just use it to store the private key.
  - The device signs the transactions; it does not allow retrieving the secret key at all.

Splitting and Sharing Keys

- Distribute the trust of storing a key among multiple devices/locations instead of one.
- Works by creating shares of the key and store each share in a different place (or with a different party).
  - Usually a threshold is used, meaning that having t out of n shares allows constructing the secret key.
- Good for both availability and security.
  - Any t shares can be used.
  - As long as less than t shares are revealed, no information will be leaked about the secret key.
    - Much better than having a single key stored at a single location.

Additive Secret Sharing

- It is a full threshold scheme (meaning all shares are needed to reconstruct the secret).
- Example 2 out of 2 or (2,2)-AddSS (all operations are integers within some finite group):
  - **Share:** pick a random value as the first share, so s1 = r, then set the second share to be s2 = s-r.
  - **Reconstruct:** given s1 and s2, compute s = s1 + s2
- This can be extended to any full threshold (3,3), (4,4), …
- Full threshold secret sharing distribute trust but lacks robustness.
  - If a single share is lost, the whole secret is lost.
- Threshold secret sharing (t-out-of-n secret sharing) using, e.g., Shamir secret sharing (see next) supports both trust distributions and robustness.

Shamir Secret Sharing I

- Supports any threshold t out of n.
- The basic idea is:
  - Generate a random polynomial of degree (t-1) with the free coefficient set as the secret key.
  - Each share is simply the evaluation of the polynomial at the share index.
  - Reconstruct is computing the free coefficient by applying lagrange interpolation over t shares.
- Example: (3,5)-SSS over Z_{11}
  - $f(x) = S + a_1x + a_2x^2 \pmod{11}$, shares are (i, f(i)), and the secret is S
  - Say S = 4, choose $f(x) = 4 + 2x + 3x^2 \pmod{11}$
  - Shares are (1, 9), (2, 9), (3, 4), (4, 5), (5, 1)
  - Any 3 shares uniquely define the original polynomial, and so the secret which is f(0).

Shamir Secret Sharing II

- **So the secret S is:**

$$S = P(0) = \sum_{j=1}^{t} y_j L_j(0)$$

- **where:**

$$L_j(0) = \prod_{\substack{m=1 \\ m \neq j}}^{t} \frac{-x_m}{x_j - x_m}$$

- **Exercise:** pick any 3 shares from the example on the previous slide and show that the secret S can be recovered.

Mulitsig and Threshold Signatures

- Multisig:
  - Simply generate multiple keys instead of one; so one signature per device/party.
  - To authorize a transaction, all devices (or t of them) must sign.
  - If collective signing is used, these multi-signatures can be aggregated into one signature under an aggregated public key.
- Threshold signatures:
  - A signing key (that corresponds to one verification/public key) is divided among several parties such that any subset of them of size t can jointly produce a signature (each produces a partial signature such that t partials signatures can be combined into one valid signature that will verify under the public key).
  - There is increasing interest of threshold signatures these days across a large number of cryptocurrency systems (e.g., the BLS scheme).

Online Wallets

- A wallet in the form of a web app.
  - The site stores keys.
  - The site issues transactions when the users asks for that.
  - The user logs in using some credentials.
- Great usability; login using any device connected to the Internet.
- Site compromised, wallet is compromised.
- Usually used when trading on exchanges.

Cryptocurrency Exchanges

- Pretty much a centrally managed bank system.
  - Accept deposits in cryptocurrency or fiat currency.
    - With a promise to pay back when a client ask for any withdrawal.
  - Allow customers to pay in cryptocurrency, receive payments, and trade cryptocurrencies.
    - Mainly match buyers with sellers at the exchange rate along with charging fees.
  - Nothing goes to the network, it is just that the exchange makes a different promise to the customers.
    - Everything is local.

Cryptocurrency Exchanges - Issues

- Highly convenient.
  - Several services at the same place.
  - Very close to traditional banking systems already in use.
  - Gives a direct value for cryptocurrency in terms of fiat currency.
- High risk.
  - Requires pre-identification - know your customer policy.
    - Any anonymity aspect promised by a cryptocurrency is taken away!
  - Security risks - a target for attackers.
    - Exchanges accumulate currency from all of their customers - the decentralization aspect is taken away!
- Regulation issues.
  - Regulations about cryptocurrencies are still evolving, as compared to traditional banking institutions which are highly regulated.