Lecture 04: Public and Private key and Double Spending

Links

- https://blockchair.com/
- https://bitinfocharts.com/top-100-richest-bitcoin-addresses.html
- https://mempool.space/
- https://www.blockchain.com/explorer/mempool/btc

💰 Double Spend Problem (Blockchain Concept)

🤔 What is Double Spend?

The **Double Spend Problem** happens when someone tries to spend the *same digital money* more than once.

Since digital money is just data, it can be **copied** or **reused** if there's no proper system to prevent it.

This is a big risk in digital payment systems.

📲 Example: UPI / Card Payments

- When you pay with **UPI** or **card**, the system **locks** the money for the first person who claims it.
- The person who's **"fastest" (rich first)** gets the money, and others get rejected.

✅ This is called a **locking mechanism** — prevents the same money from being spent twice.

⛓️ How Blockchain Solves It?

Blockchain uses:

- 📜 **Ledger**: A public, unchangeable record of all transactions.
- ⏳ **Order**: Transactions are confirmed in blocks one after another.
- 🔐 **Consensus**: Everyone in the network agrees which transaction came first.

So, even if someone tries to double spend:

- Only **one transaction is validated**.
- The other is rejected as invalid.

🚀 Why is it Important?

✅ Keeps the currency **trustworthy**.

✅ Prevents fraud.

✅ Makes digital currency (like Bitcoin) work like real cash — can't spend the same note twice.

📚 Analogy:

🎮 Imagine you have 1 gaming coin and try to play 2 machines at the same time.

Only 1 machine accepts the coin. The other rejects it because the coin is already spent.

<————BITCOIN————>

1 BTC Send→ 1. OM 2. Jerry

Tom

🌳 Merkle Root & Hashcode in Blockchain

📖 What is Merkle Root?

- A **Merkle Root** is a **single hash value** that represents all the transactions in a block.
- It is created by **combining all transaction hashes** in pairs and hashing them again and again until only one hash remains — the Merkle Root.

✅ **Why?**

- Saves space: No need to store all transactions in the block header.
- Verifies transactions quickly and securely.

🔗 Block Header Structure

A **block header** in blockchain contains:

```
mathematica
CopyEdit
HashCode = Prefix + Nonce
          ↓
HashCode = Nonce + Previous Block Hash + Merkle Root
```

🧱 Block Header Fields:

1️⃣ **Block Number** — Position of the block in the chain.

2️⃣ **Nonce** — Random number used in mining (Proof of Work).

3️⃣ **Previous Hash** — Hash of the previous block (ensures chain integrity).

4️⃣ **Merkle Root** — Combined hash of all transactions in this block.

🌲 How Merkle Root is Computed?

🪄 Steps:

- Take hash of each transaction (tx1, tx2, ...).
- Combine them in pairs and hash again.
- Repeat until **one hash remains** → Merkle Root.

📊 Example:

```
less
CopyEdit
tx1   tx2   tx3   tx4
 |     |     |     |
h1    h2    h3    h4
 |_____|
   h12       h34
    |_______|
      Merkle Root
```

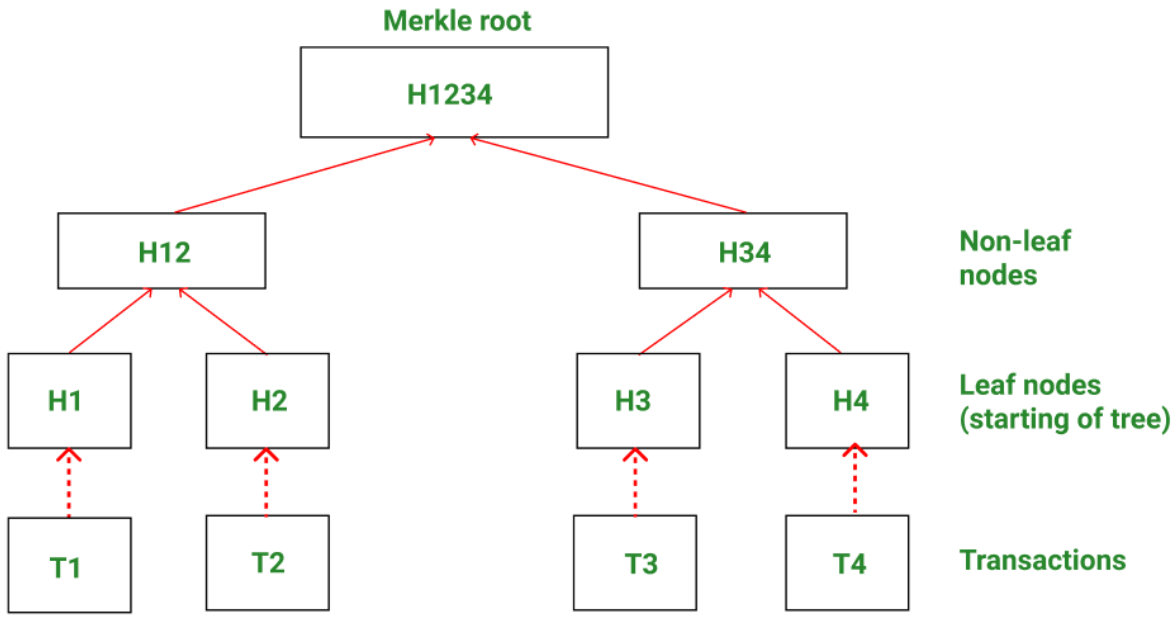
🚀 Why is it important?

- 🧪 **Verification**: Easy to check if a transaction is part of a block.
- 🔐 **Security**: Protects against tampering.
- 📦 **Efficiency**: Reduces data needed to store/verify transactions.

📚 Reference:

GeeksForGeeks: Blockchain Merkle Trees

https://www.geeksforgeeks.org/software-engineering/blockchain-merkle-trees/



🔐 Private Key & Public Key (Asymmetric Cryptography)

🧾 What is Asymmetric Cryptography?

- It is a **method of encryption** where two keys are used:

  1️⃣ **Public Key** — Shared with everyone.

  2️⃣ **Private Key** — Kept secret.

They work as a **pair** — what one key encrypts, the other can decrypt.

This ensures **security & authenticity**.

📦 How it works?

📝 Example:

```
makefile
CopyEdit
```

```
Message: hello
```

✅ Encrypt with **Private Key**:

```
pgsql
CopyEdit
hello — [Private Key] → Encrypted: sfewrAesfrrF
```

✅ Decrypt with **Public Key**:

```
pgsql
CopyEdit
Encrypted ← [Public Key] → hello
```

📲 Analogy: UPI Example

- Your **UPI ID** → like your **Public Key**
  - Anyone can see it & send you money.
- Your **UPI PIN** → like your **Private Key**
  - Only you know it & use it to authorize the payment.

So even if everyone knows your UPI ID (public), they can't spend your money without your PIN (private).

🎯 Why use Asymmetric Cryptography?

✅ Secure communication.

✅ No need to share private keys.

✅ Used in Blockchain, SSL certificates, digital signatures, etc.

✍️ Digital Signature & Public Key Validation

🔐 What is a Digital Signature?

A **digital signature** is a special kind of encrypted code that proves:

✅ The sender really sent the message.

✅ The message was not changed (integrity).

It's like a handwritten signature — but in digital form!

📜 How it works?

🔷 Step by Step:

1️⃣ Sender **signs** the message using their **Private Key**.

2️⃣ The receiver **validates** the signature using the sender's **Public Key**.

🔷 Why Public Key?

Because the Public Key is available to everyone, anyone can check if the signature is valid — but only the person with the Private Key could have created it.

✅ **Only the sender's Private Key can create a valid signature.**
✅ **Only the sender's Public Key can verify it.**

📲 Analogy:

Imagine:

- 🔑 Sender locks a box with their **special lock (private key)**.
- 👮 Anyone with the sender's **public key** can check that the lock is genuine.

So it proves the sender's identity!

🚀 Why use Digital Signatures?

✅ Authenticate the sender.

✅ Ensure data is not tampered with.

✅ Used in Blockchain, UPI, Emails, and secure websites.