CS61065: Theory And Applications of Blockchain

Blockchain Security

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

Risks

Risk

- **Common Risks**
  - Blockchain operation mechanism (Blockchain 1.0, 2.0)
- **Specific Risks**
  - Development, deployment and execution of smart contract (Blockchain 2.0)

https://www.sciencedirect.com/science/article/pii/S0167739X17318332?casa_token=-Os8D6Mop7AAAAAA:NU-biO_avee1LRnqJ-6Aycw3yoQWhcWle0WdXaha3O-Dl2qPgl6EBZ0laOiJQQPpyi_6lVZ_ig8

Risks

Risk

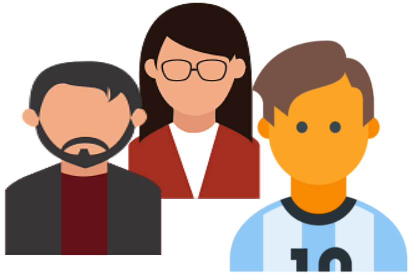
Blockchain operation mechanism

- 51% vulnerability
- Private key security
- Criminal activity
- Double spending
- Transaction privacy leakage

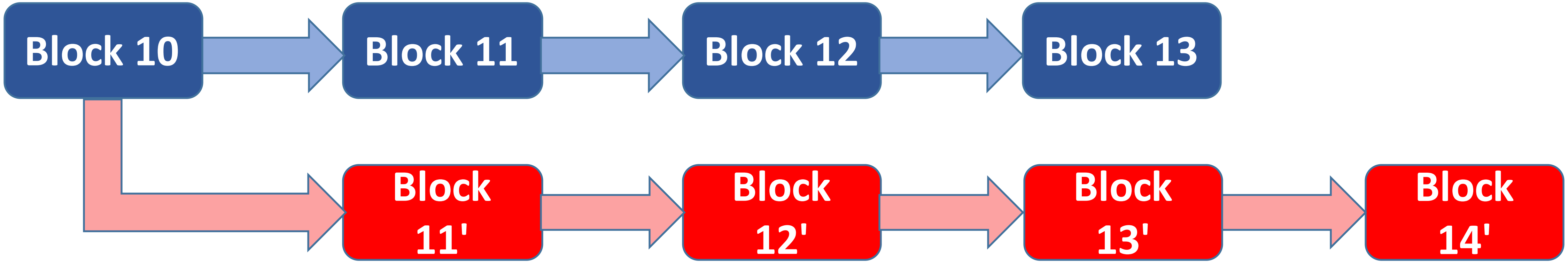
Smart contract execution

- Criminal smart contract
- Vulnerabilities in smart contract
- Under-optimized smart contract
- Under-priced operations

Common Risk: 51% vulnerability

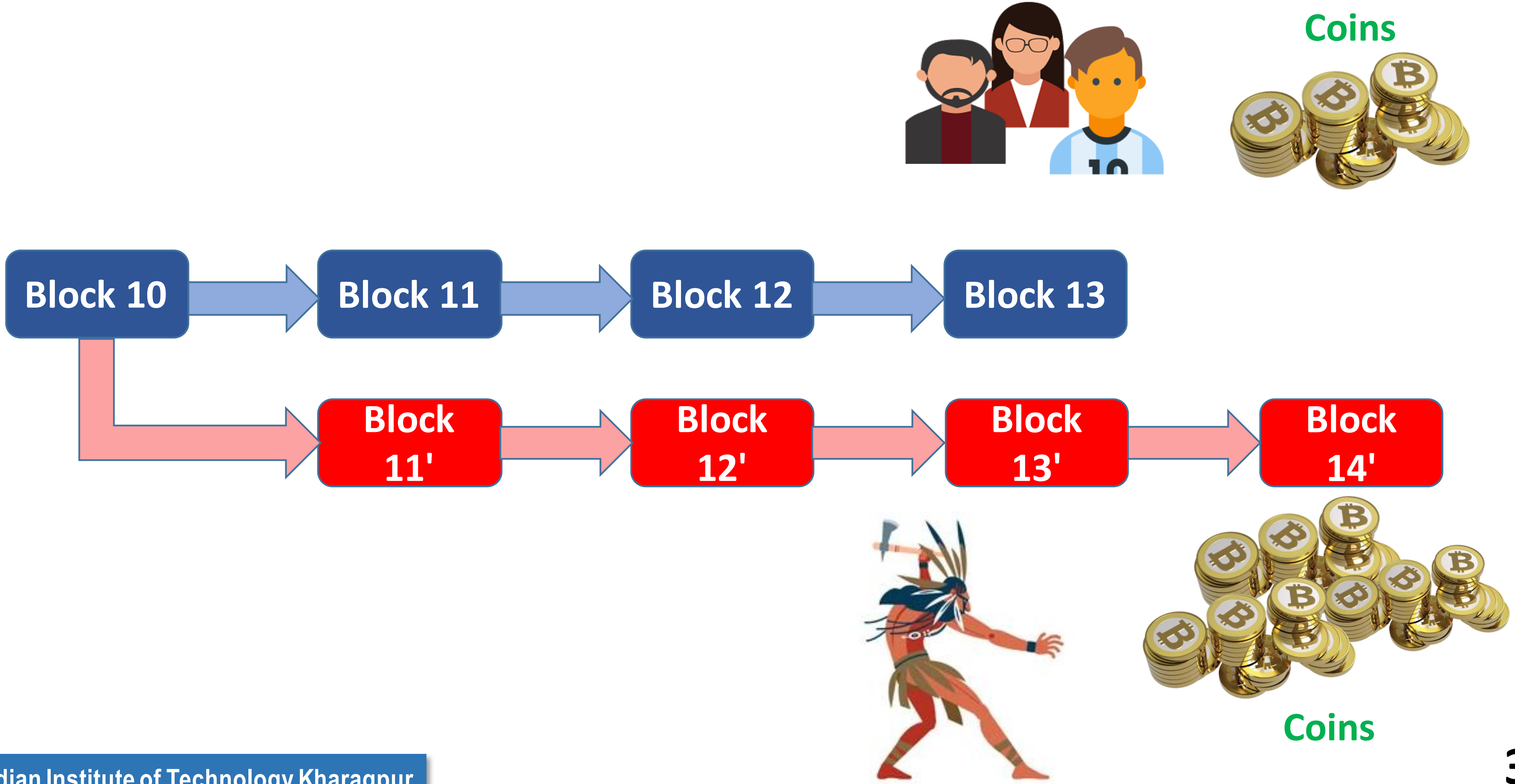


Hashing power

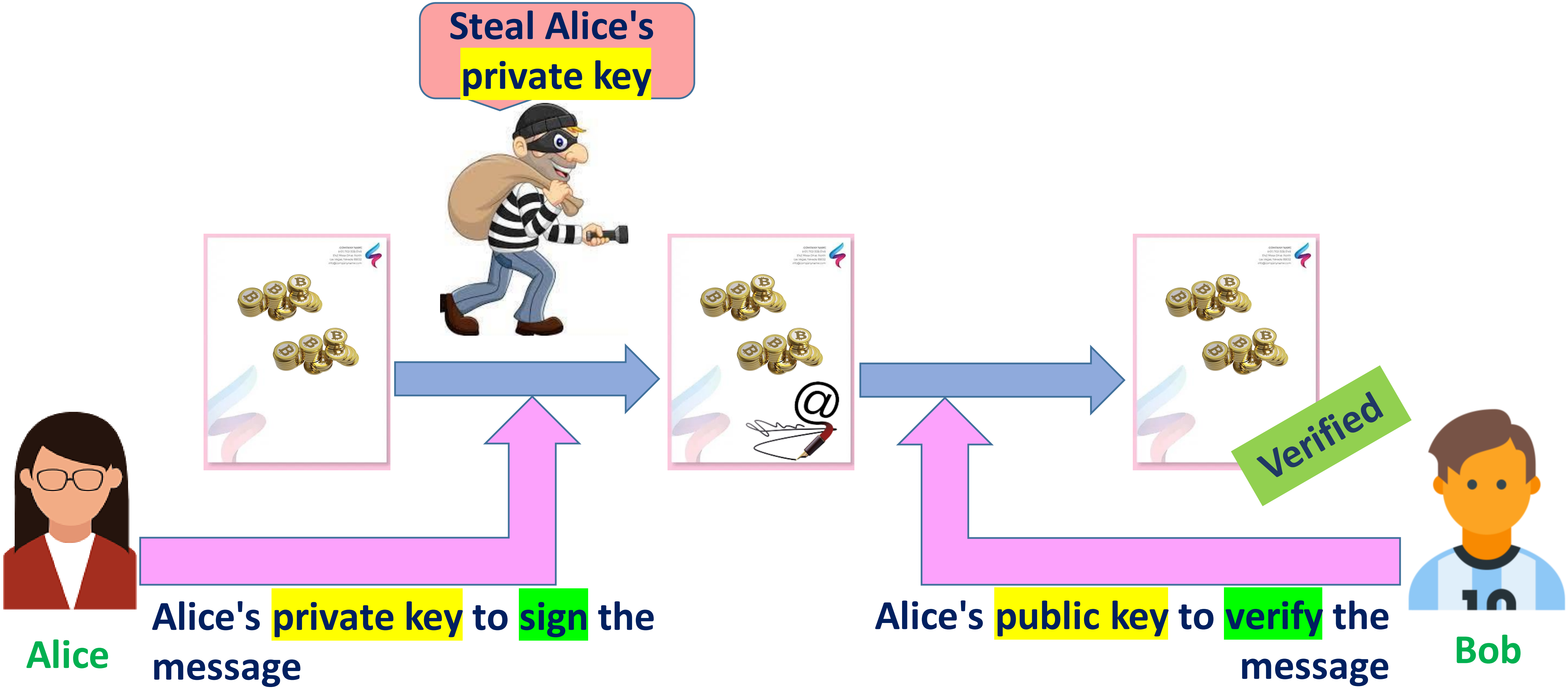


Hashing power

Common Risk: 51% vulnerability

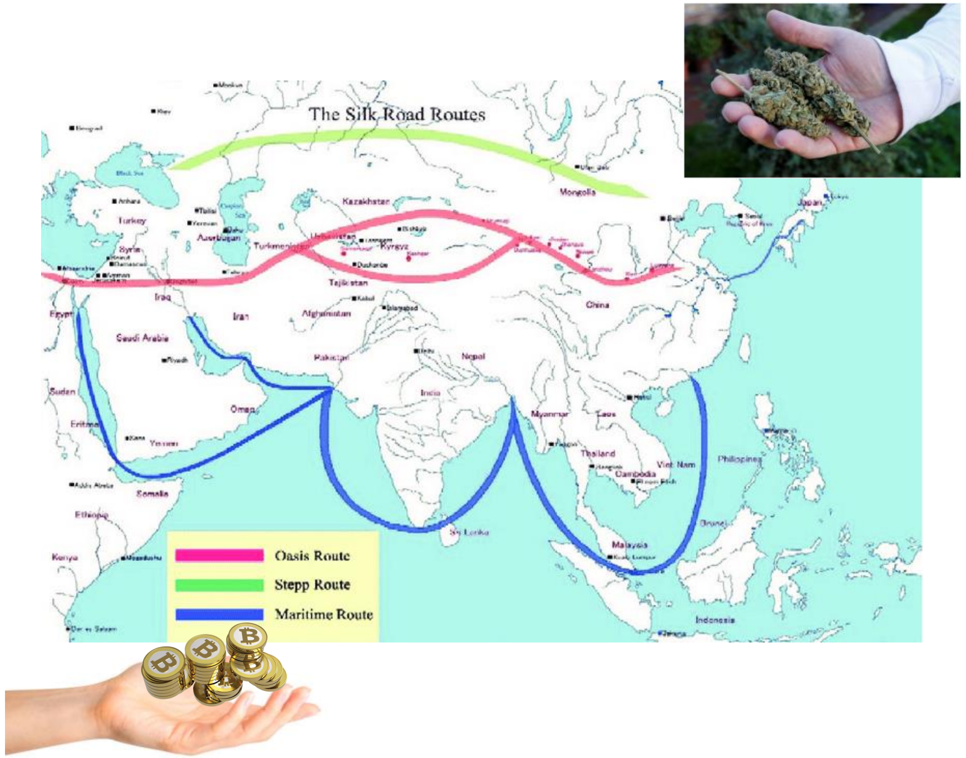


Common Risk: Private key security



Common Risk: Criminal Activity

Ransomware



Underground market

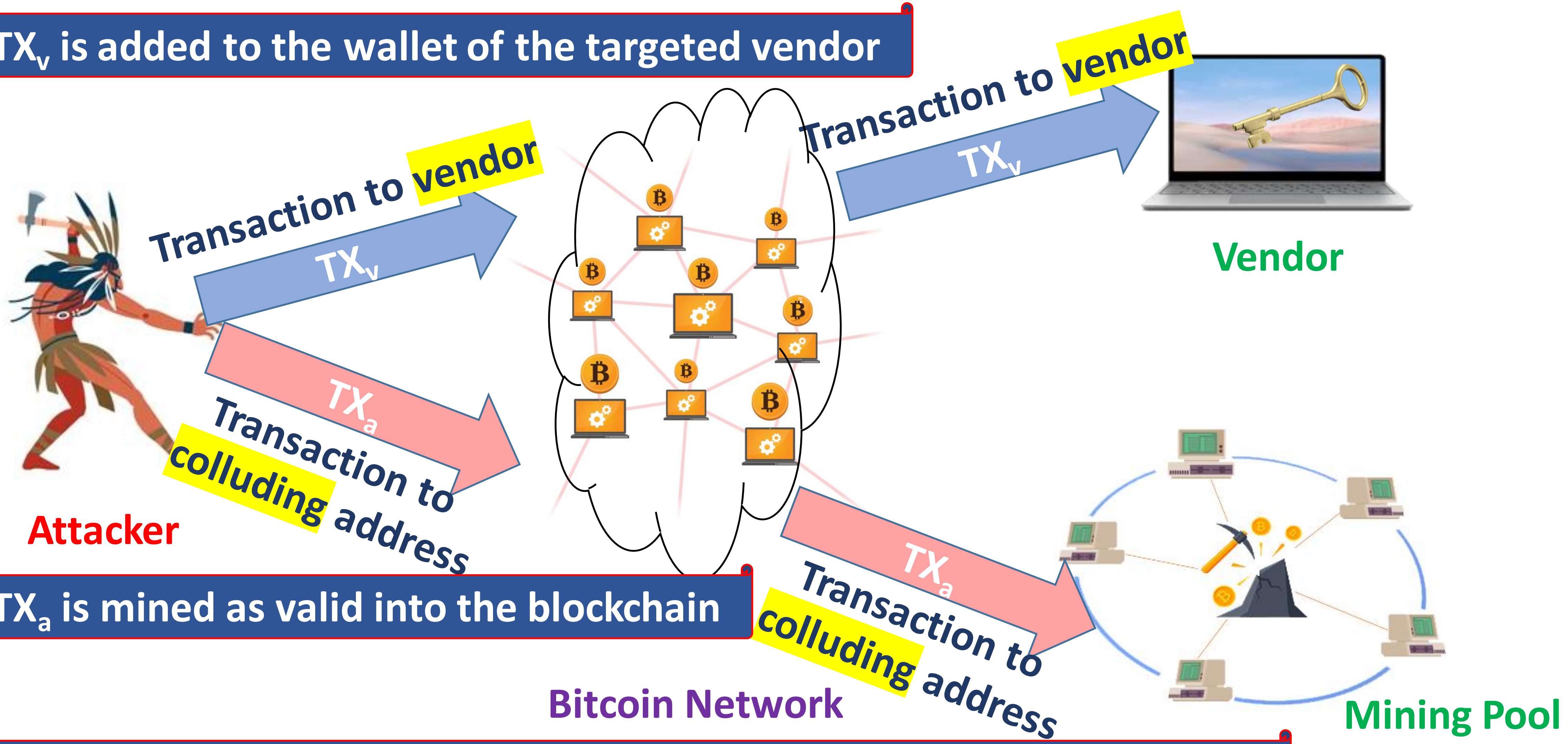
Chaff coin

Valid coin

Money laundering

Common Risk: Double Spending

1. TX_v is added to the wallet of the targeted vendor

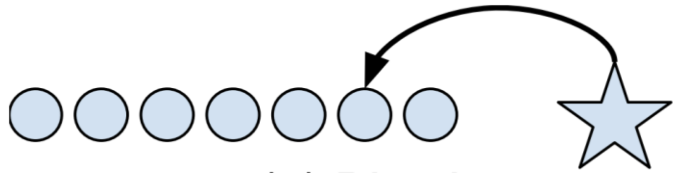


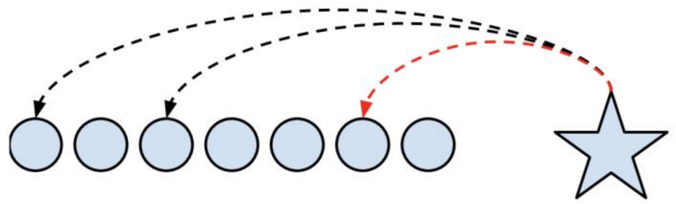
2. TX_a is mined as valid into the blockchain

3. The attacker gets TX_v 's output before the vendor detects misbehavior

Common Risk: Transaction Privacy Leakage

A new transaction (the star) which spends an available coin (the second circle from the right)





Transactions and tracing in Bitcoin

Transactions and tracing in Cryptonote

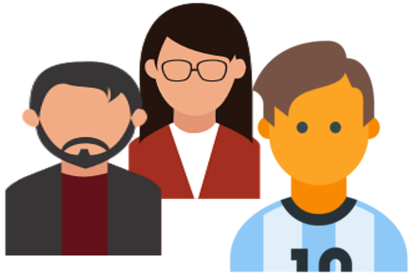
- ✓ Each transaction input explicitly identifies the coin being spent, thus forming a linkage graph

- ✓ Each transaction input identifies a set of coins, including the real coin along with several chaff coins called "mixins."
  - ✓ Many mixins can be ruled out by deduction
  - ✓ The real input is usually the "newest" one

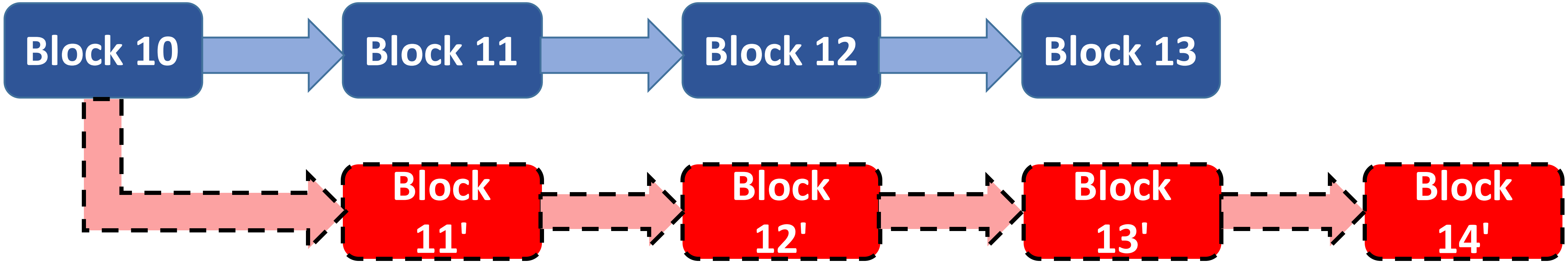
Selfish Mining Attack

Is bitcoin mining protocol incentive-compatible?

Selfish Mining Attack



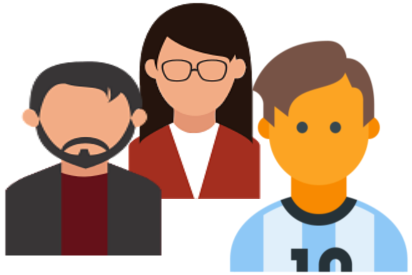
Others



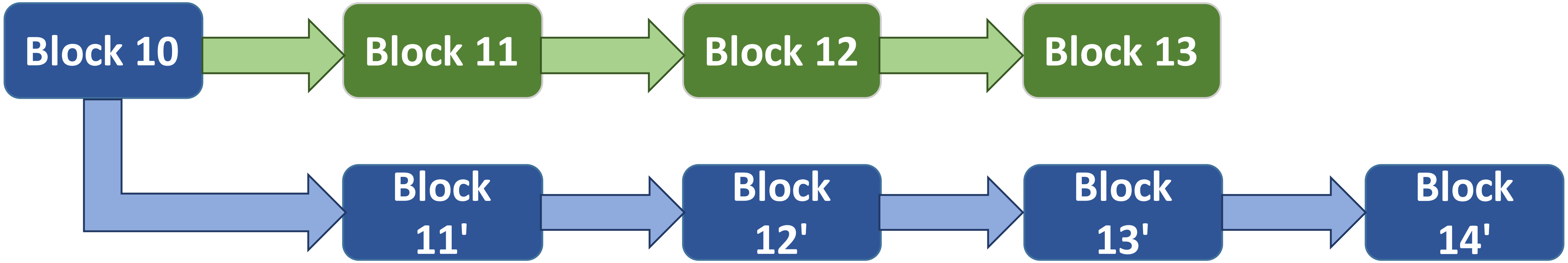
Mining Pool

https://link.springer.com/chapter/10.1007/978-3-662-45472-5_28

Selfish Mining Attack



Others



Mining Pool

Selfish Mining Attack

Pool intentionally forking the chain for keeping discovered blocks private

The honest nodes continue to mine on the public chain
The pool mines on its own private branch

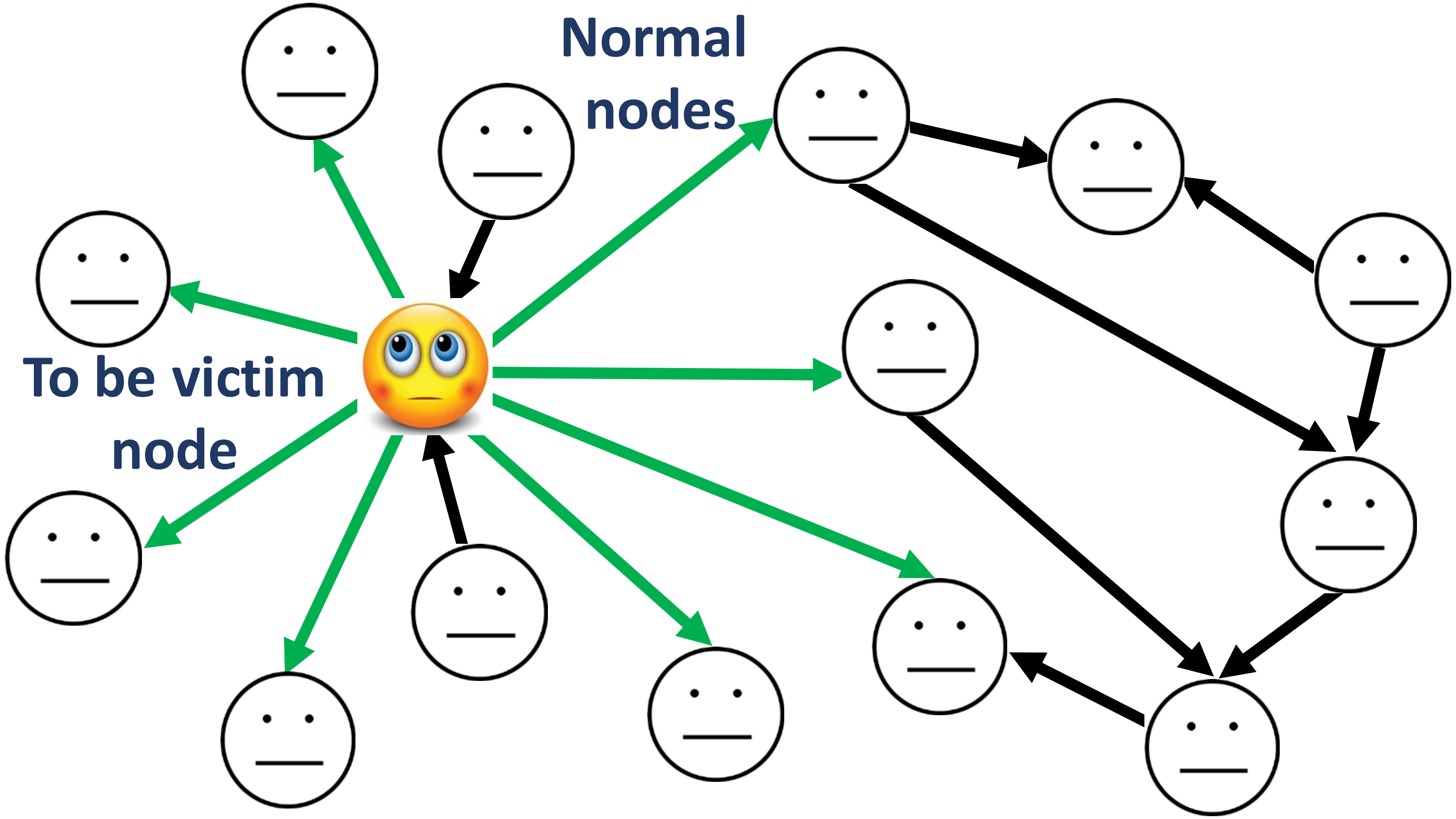
Discovering more blocks by pool develops a longer lead on the public chain, and continues to keep these new blocks private

When the public branch approaches the pool's private branch in length, the selfish miners reveal blocks from their private chain to the public

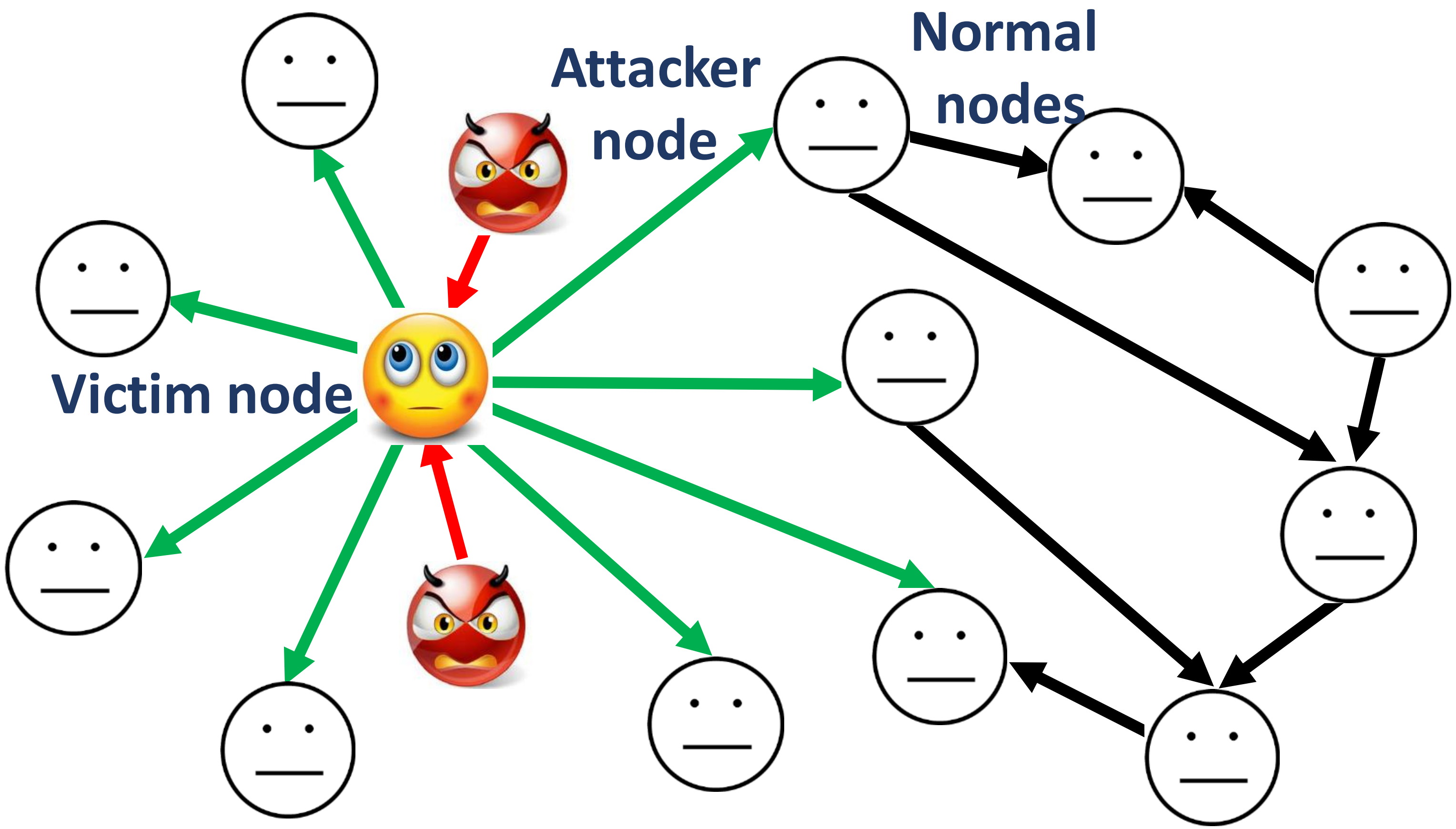
Eclipse Attack

Is the bitcoin peer-to-peer **network safe?**

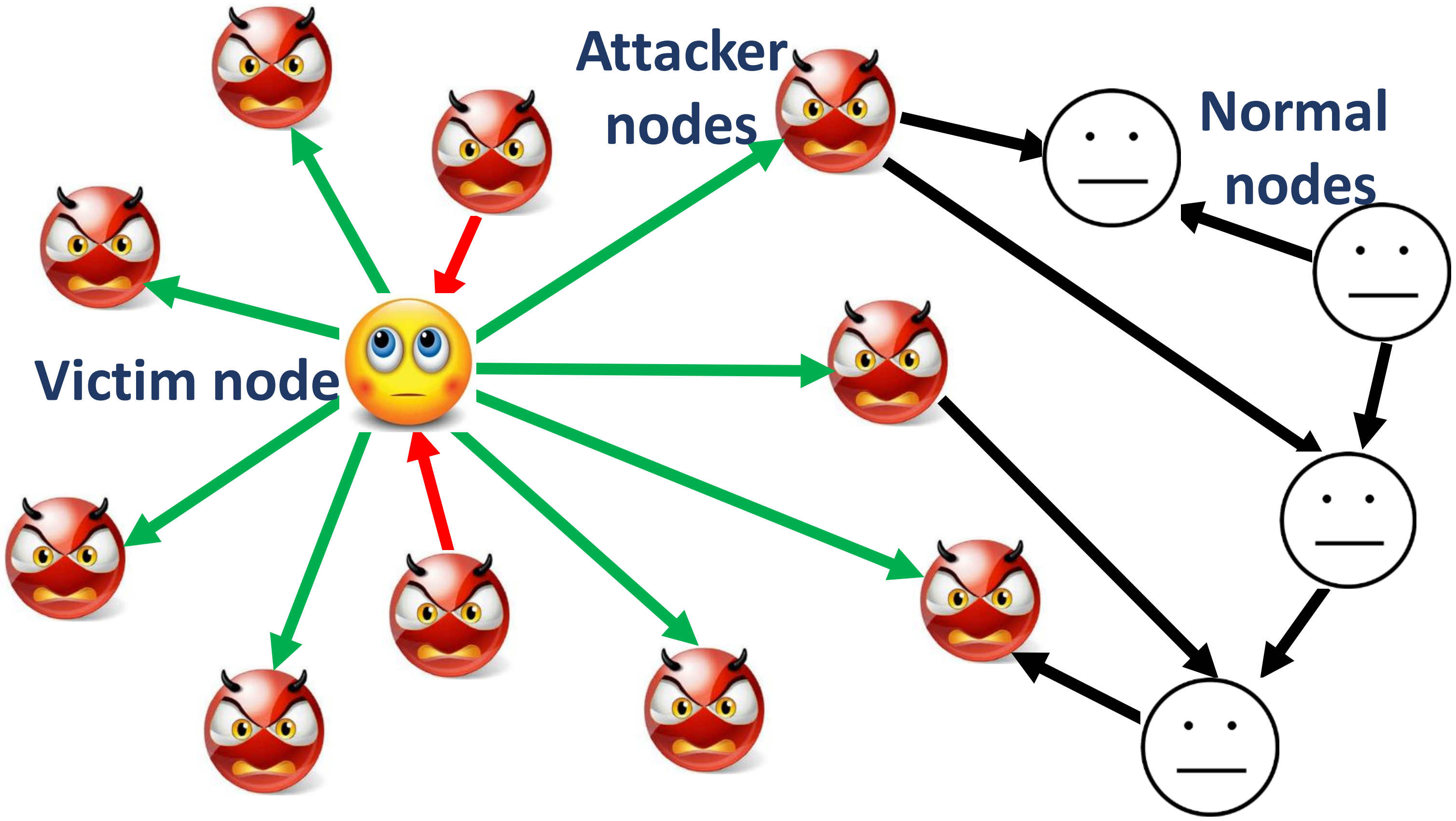
Eclipse Attack



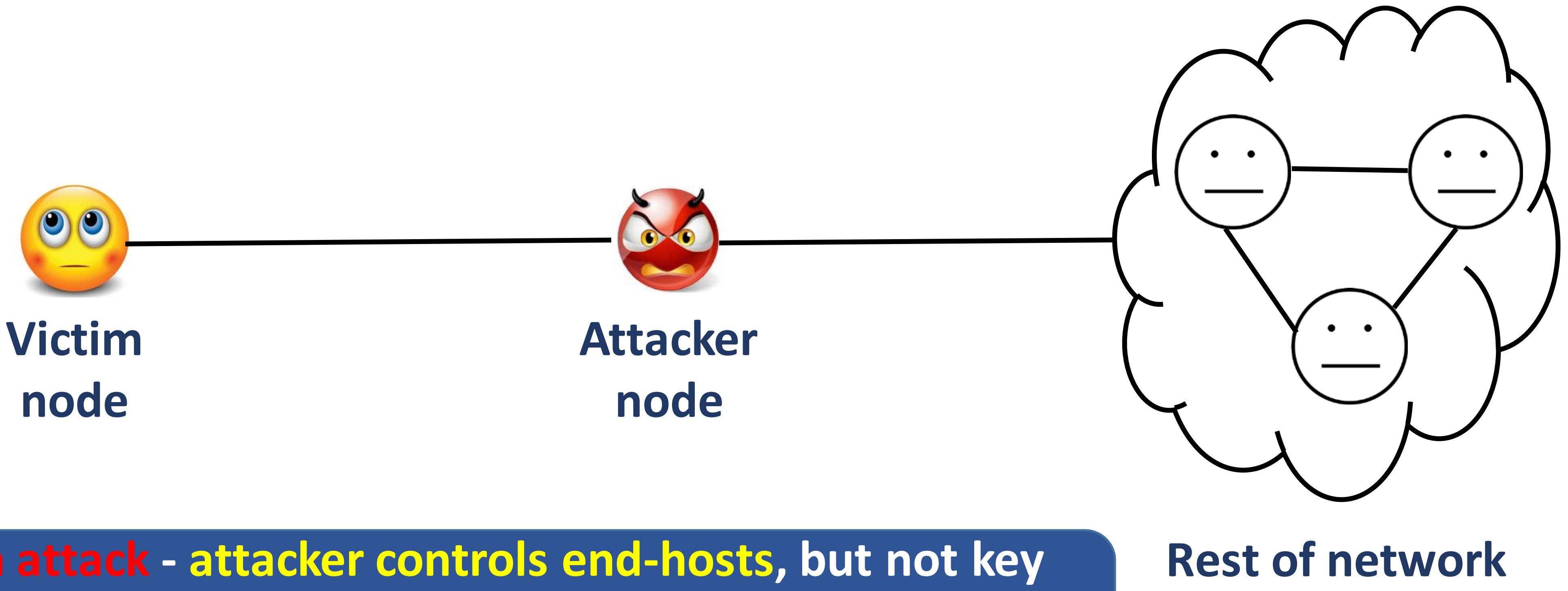
Eclipse Attack



Eclipse Attack



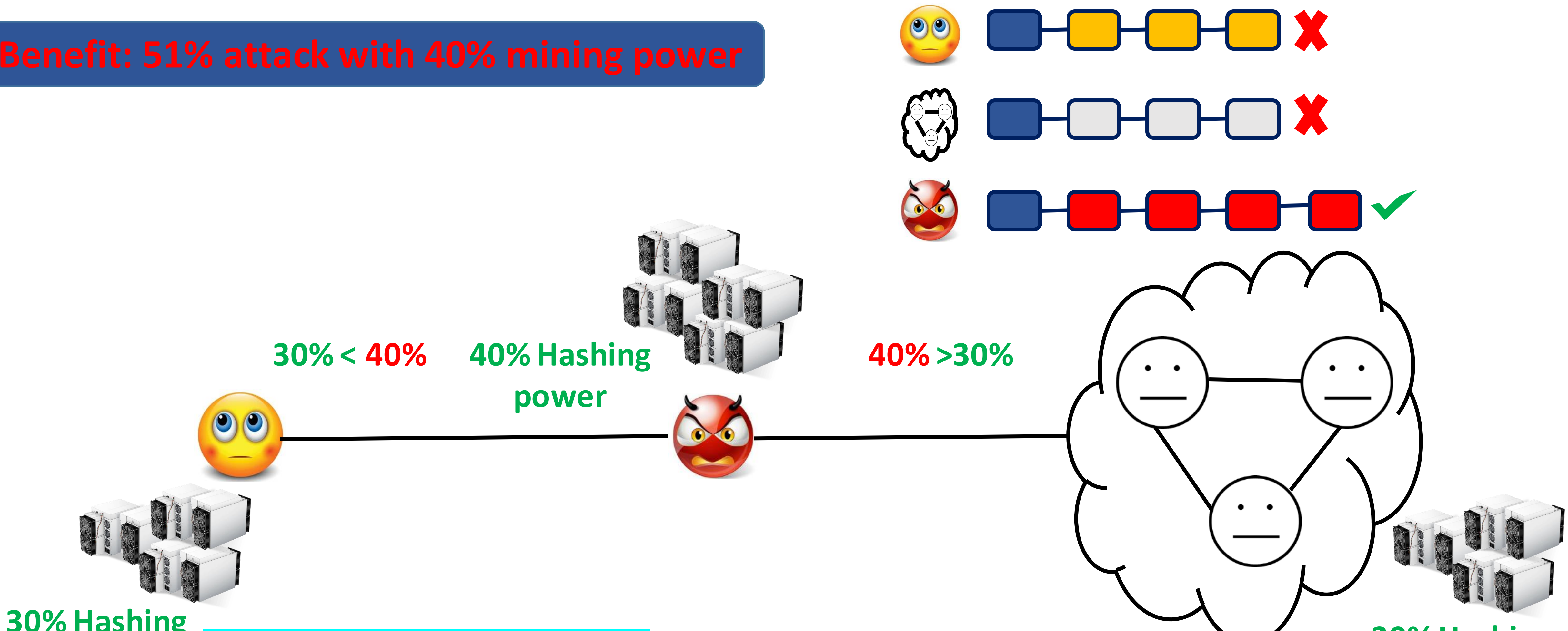
Eclipse Attack



Off-path attack - attacker controls end-hosts, but not key network infrastructure between the victim and the rest of the bitcoin network

Eclipse Attack

Benefit: 51% attack with 40% mining power



Attacker partitions miners so they can't build on each other blocks

The attacker then outcompetes each partitioned miner

Eclipse Attack

Attacker populates the victim node's peer tables with attacker's IP addresses

Victim node restarts and loses current outgoing connections

The victim establishes all new outgoing connections to attacker IP addresses

Eclipse Attack

1. Populating of IP addresses

- ✓ Each node picks its peers from IP addresses stored in two tables
  - ➢ **New table:** IPs the node has heard about
  - ➢ **Tried table:** IPs the node peered with some point
- ✓ The tables also store a timestamp for each IP
- ✓ Each table stores the IPs in buckets
- ✓ To find an IP to make an outgoing connection to:
  1. Choose new or tried table to select from
  2. Select an IP with newest timestamp
  3. Attempt an outgoing connection to that IP

Attacker populates tables with attacker IPs so that the victim node only connects to the attacker IPs

Selection Bias: Attacker ensures its IPs are the newer one

Eclipse Attack

2. Restarting node event is natural?

- ✓ Software/security updates
- ✓ Packets of death/DoS attacks
- ✓ Power/network failures
- ✓ ISP outages

3. Bucket eviction

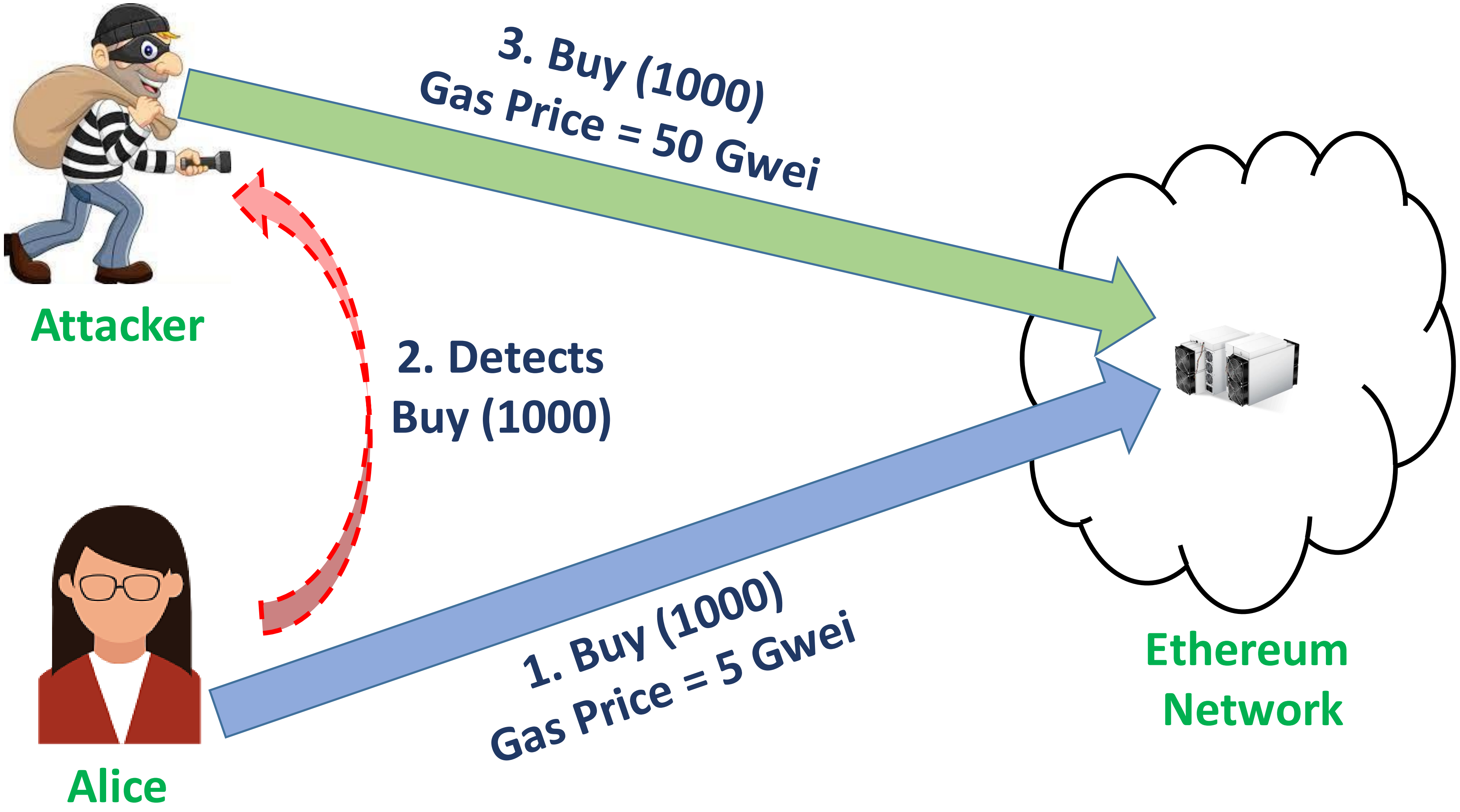
✓ The bucket is full, and an IP is inserted into it

1. Randomly selects 4 IPs
2. Delete oldest IP
3. Insert new IP

Eviction Bias: Attacker IPs will always have the **most recent timestamps**

Try-Try-Again: If an attacker IP replaces another attacker IP, the evicted IP is resend and eventually replaced by honest IP

Front-running Attack



Front-running Attack

Front-running Attack

- Displacement
  - After Attacker's run, Alice's function call is not important to the attacker
- Insertion
  - After Attacker's run, the state of the contract is changed. Alice's function call is needed to run on this modified state
- Suppression
  - After Attacker's run, she tries to delay Alice's function call. After the delay, Alice's function call is indifferent to the attacker

Front-running Attack

Front-running Attack (Displacement / Insertion / Suppression)

- Asymmetric
  - Alice is trying to cancel an offer, and Attacker is trying to fulfill it first
- Bulk
  - Attacker is trying to run a large set of functions and Alice is trying to buy a limited set of shares offered by a firm on a blockchain

Front-running Attack

Markets and Exchanges:
Spotting a profitable cancellation transaction

(Unordered) mempool

[...]
Cancel (order 100)
Breed (cryptokittie 1, cryptokittie 2)
Buy (order 101)
Bid (B_j)
Register (B_i)
[...]

Ethereum Network

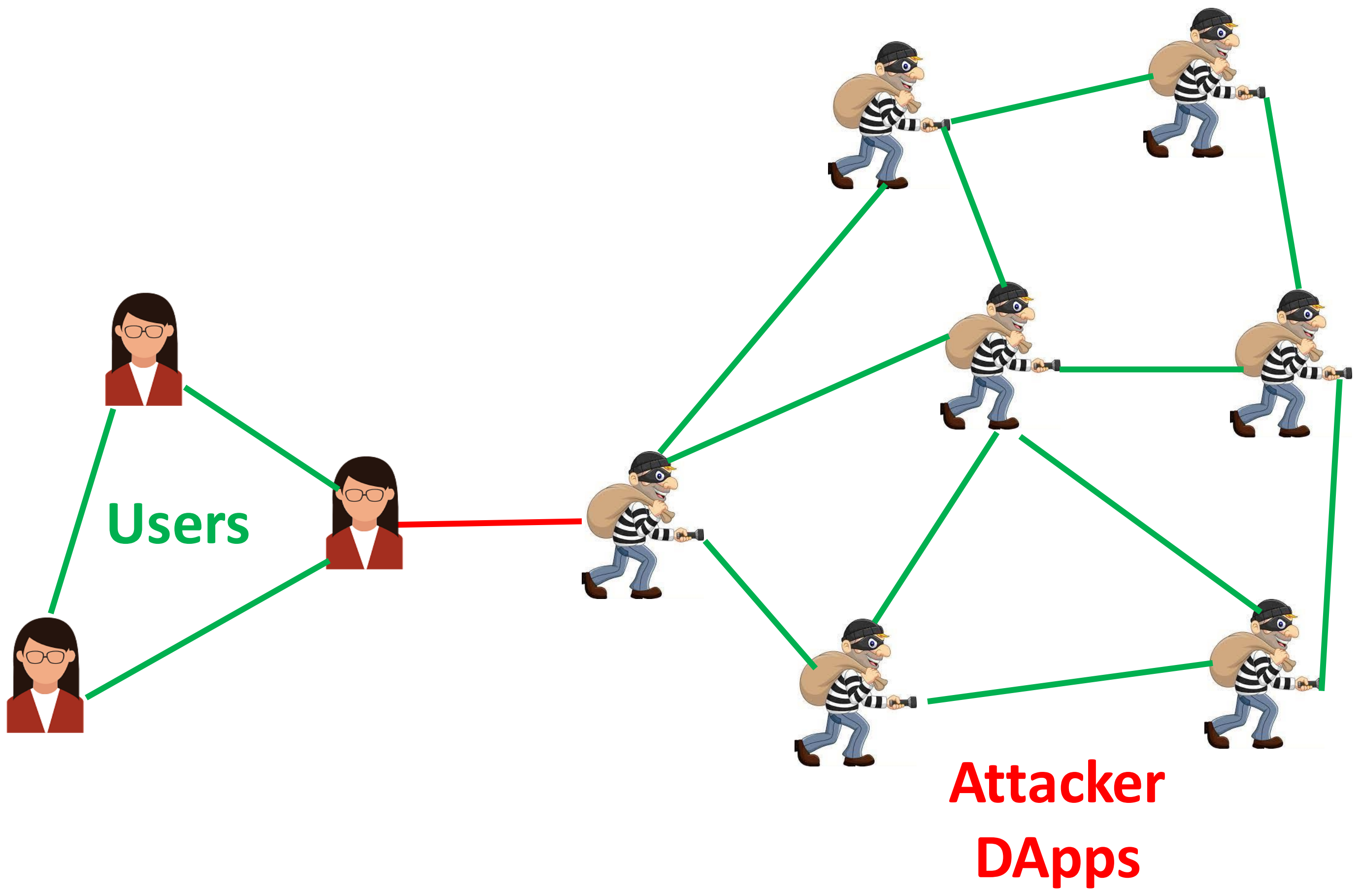
Adversarial miner mines a block with preferred order

Reordered Block

Block Height #N
Register (B_i)
Buy (order 100)
Cancel (order 100)
Buy (order 101)
Bid (B_j)
Breed (cryptokittie 1, cryptokittie 2)

Front-running Attack

Gambling:
Bribing miners for prioritizing themself



- ✓ When the timer of Fomo3D game reached about 3 minutes, the winner bought 1 ticket and then sent multiple high gasPrice transactions to her own DApps
- ✓ Transactions congested the network
- ✓ Bribed miners to prioritize them ahead of any new ticket purchases in Fomo3D

Tesseract - Secure Real Time Crypto Currency Exchange (ACM CCS ’19)

- Centralized exchange designs are vulnerable to theft of funds
- Decentralized exchanges cannot offer real-time cross-chain trades

- Tesseract - Real time - as fast as centralized exchanges
- Can observe the buy (a.k.a. “bid”) and sell (a.k.a. “ask”) prices in real time and **allow parties to modify their trading positions in milliseconds**.
- Supports **cross chain asset exchanges**
- Secure against theft by exchange operators

Tesseract (CCS ’19)

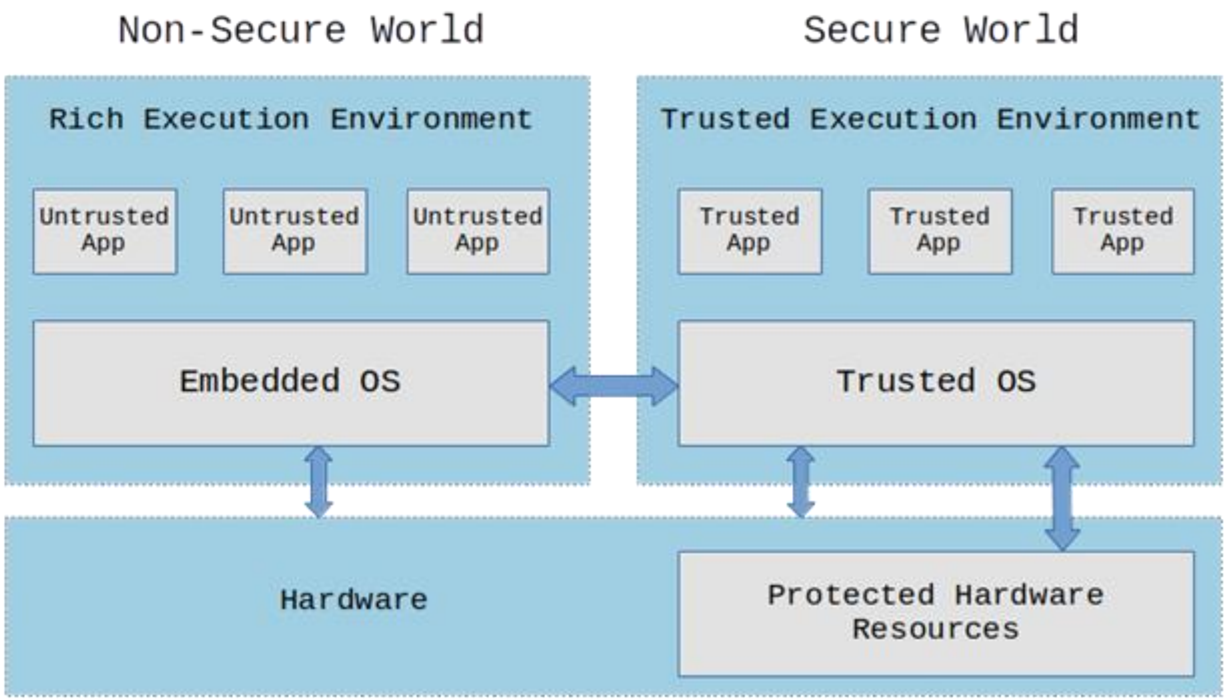
- Tesseract relies on a Trusted Execution Environment (**TEE**)
- **TEE enables Tesseract to behave like a trusted third party**, controlling funds without exposing them to theft while preventing frontrunning by the exchange operator
- Ensures ***all-or-nothing settlement***

Tesseract (CCS '19)

TEE is a **secure area of a main processor** (Refer to Intel SGX – Software Guard Extensions)

- Guarantees code and data loaded inside to be protected with respect to:
    - **Isolated execution**
    - **Integrity of applications**
    - **Confidentiality of their assets**

- **Provides attestation that an output represents the result of such an execution in the TEE**

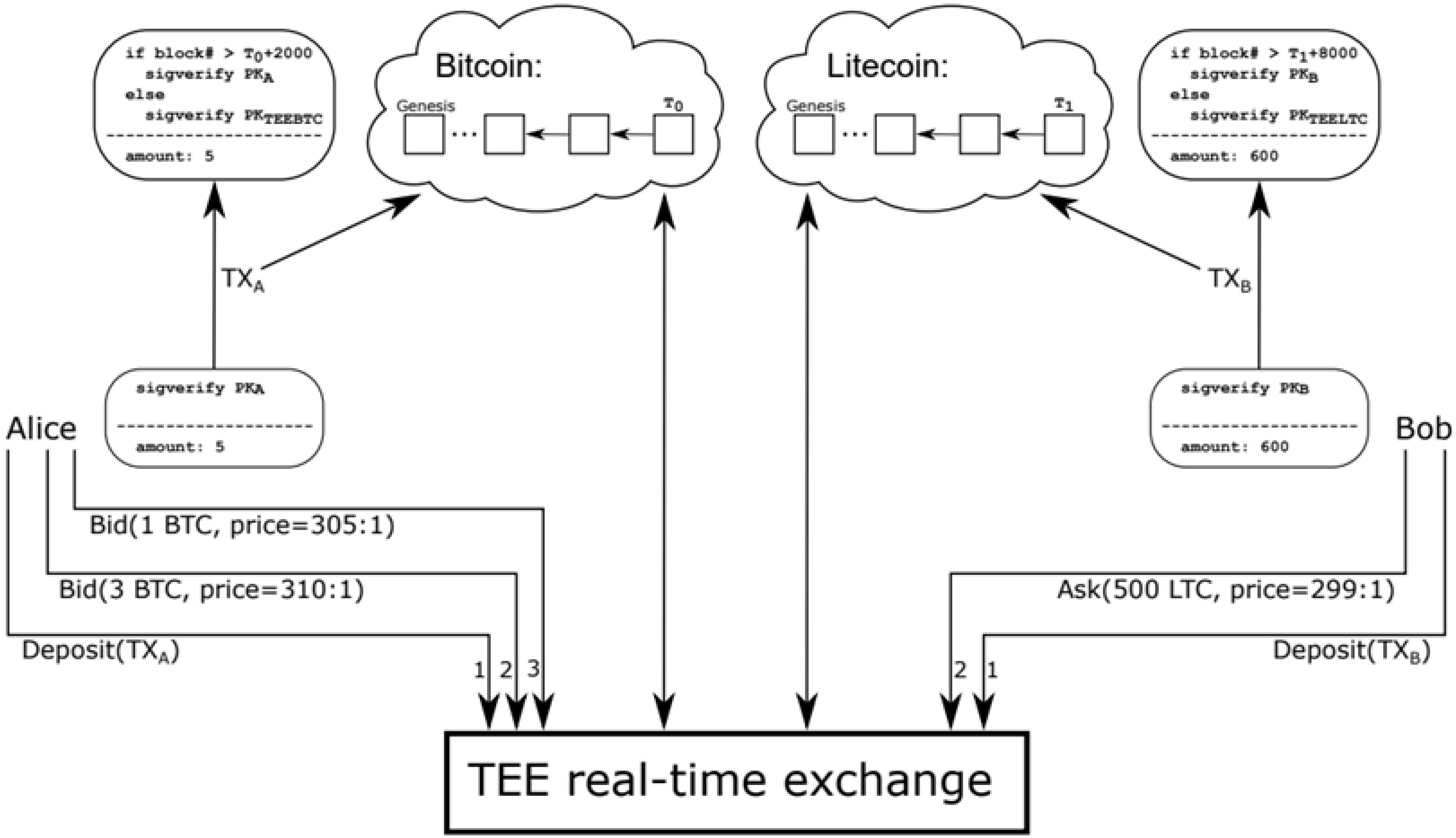


https://embeddedbits.org/introduction-to-trusted-execution-environment-tee-arm-trustzone/

Tesseract (CCS ’19)

- Assumes the security of TEE and *remote attestation*
  - Code that runs inside the TEE enclave can neither be observed nor tampered with
- Strong network adversary that can gain complete OS and network control of the host in which the funds are stored
- Adversary’s goal is to maximize her profit

Tesseract (CCS ’19)



Tesseract configured with Bitcoin and Litecoin networks.
The TEE forms the secure real-time exchange with which the parties interact to carry out asset swaps.

Tesseract (CCS ’19)

1 **Initialization of exchange**

1. **Tesseract enclave** is running light blockchain clients (**SPV** – Simplified Payment Verification clients).
2. During initialization, the **enclave fetches the latest blockchain state**.
3. The enclave then invokes a **key generation** procedure to create a keypair **(sk, pk) for each supported cryptocurrency**.
4. The enclave will then **attest pk** as its deposit address, for each cryptocurrency.
5. The **attested pk** keys are published through multiple services - (such as websites, IPFS, and even Bitcoin and other blockchains).
6. The **pk** also forms the depositor address for Tesseract exchange in the corresponding cryptocurrency.

https://en.bitcoinwiki.org/wiki/Simplified_Payment_Verification

Tesseract (CCS ’19)

2 User Initialization

1. A **new user opens a Tesseract account by depositing a significant amount** into a deposit address of the exchange.
2. The **proof of deposit is sent to the enclave** in the form of the *transaction*, *index of block*, and ***Merkle tree***.
3. Tesseract credits the user’s account (in the enclave) after verifying the deposit.
4. Tesseract also **protects against replay attacks**, by requiring strictly increasing block indices for the user’s deposits.
5. Each user account is identified by the user’s public key and hence the user can use its secret key to generate signed messages to interact with Tesseract.

Tesseract (CCS ’19)

Asset Exchange

3

1. Users post buy or sell orders along with their exchange rate. Orders are recorded in the enclave.
   Eg: Bob bids to **sell** LTC for the price of **299 LTC per BTC**
   Alice bids to **buy** LTC price of **305 LTC per BTC**.
2. Tesseract server publishes an anonymized version of the order book with remote attestation.
3. Realtime trading is executed by matching the buy and sell orders.
4. The trade results do not reflect in the blockchain networks immediately, and are only reflected in the tesseract exchange sandbox.
5. Periodically, *settlement* transactions are issued by tesseract in the blockchain networks to update users’ balance.
6. Tesseract exchange can be configured to collect a proportional fee for each successful trade (e.g 0.1% of amount). This will help in exchange owners to maintain their infrastructure.

Tesseract (CCS ’19)

Resistance against Eclipse Attacks is the most interesting aspect of Tesseract.

- Tesseract enclave is **preloaded with the current difficulty parameter** of each PoW-based blockchain.
- This can be further verified by users before using the Tesseract exchange.
- Assume adversary **A controls < ½ of computational power.**
- **A has total control over the network**

Eclipse Attack:

- **A** cuts the enclave off from the Bitcoin network and presents it with a fake blockchain containing a deposit transaction TX_{fake}
- Tesseract monitors the rate at which blocks are being mined in the network. (e.g. ~10 minutes for bitcoin)
- Fixed **difficulty parameter** -> **A** cannot change the difficulty, and forced to mine at the current difficulty.
- **A** cannot mine at the required rate (since **A** controls < ½ of computational power of main network).
- TEE has a trusted clock. **Attack is detected by Tesseract**.

References

- https://www.sciencedirect.com/science/article/pii/S0167739X17318332?casa_token=-Os8D6Mop7AAAAAA:NU-biO_avee1LRnqJ-6Aycw3yoQWhcWle0WdXaha3O-Dl2qPgl6EBZ0laOiJQQPpyi_6lVZ_ig8
- https://ieeexplore.ieee.org/abstract/document/8369416?casa_token=Y4DlRAdrVA0AAAAA:AsbmBNQugd6EcGIN_JKgcqI8AThFdIyhSCIkAiwO3JL6nVWrAeZDHyj4oyqxQ475ZO8saES-ReI
- https://link.springer.com/chapter/10.1007/978-3-662-45472-5_28
- https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-heilman.pdf
- https://link.springer.com/chapter/10.1007/978-3-030-43725-1_13