拜占庭叛将指南

区块链漏洞新类型分享

分享人：于晓航/xhyumiracle 长亭科技区块链安全负责人

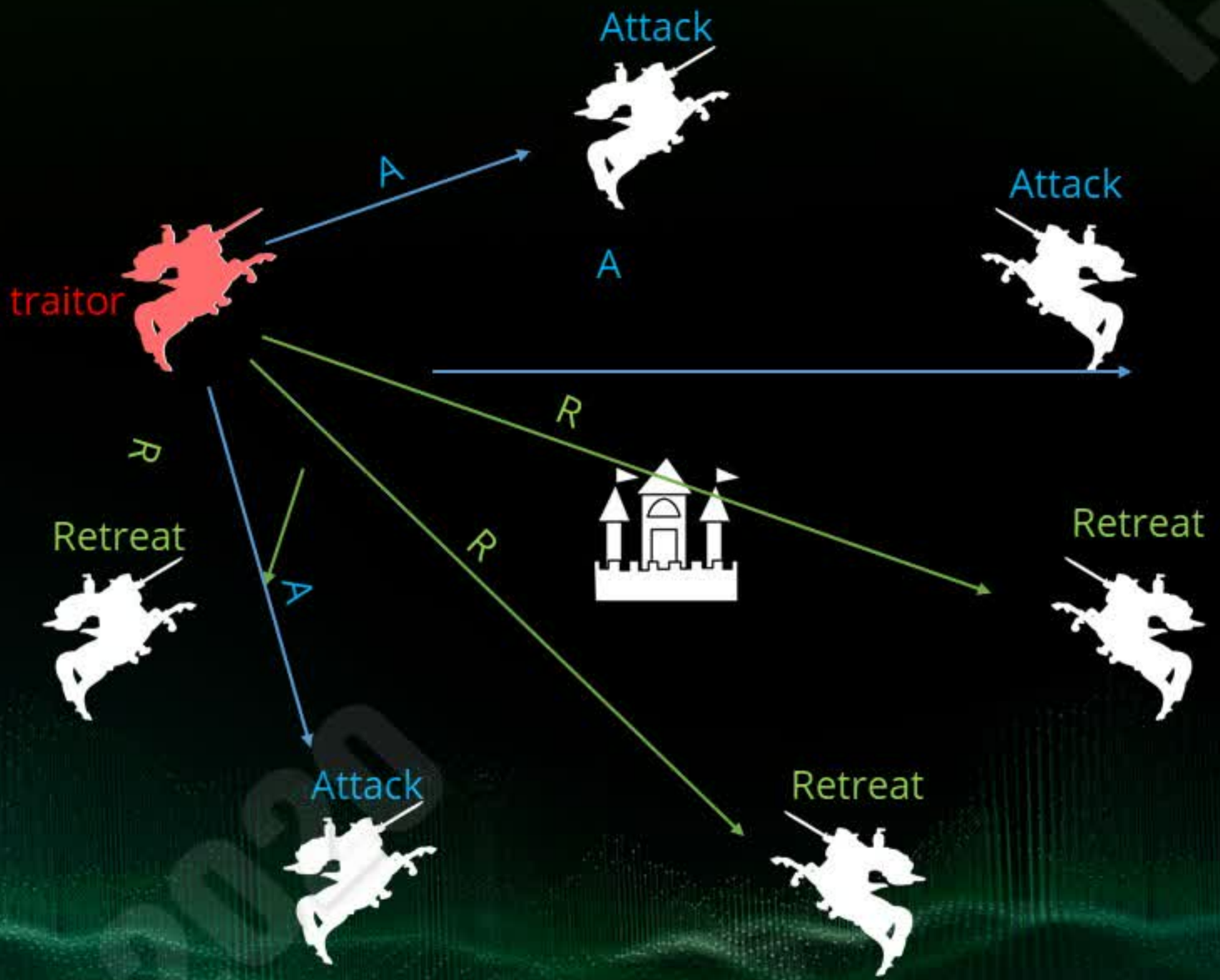
于晓航/xhyumiracle

- 长亭科技区块链安全负责人
- 公链、联盟链、智能合约、交易所、钱包等漏洞挖掘
- N*100k 行区块链项目源码审计
- ≈100 漏洞
- DEFCon, RWCTF, BCCon, MiiXCon...
- 区块链安全生存指南、区块链生态安全服务解决方案、区块链漏洞定级标准
- EVM Opcode JOP

目录

拜占庭将军问题
Byzantine Generals Problem

拜占庭将军问题


Attack
A
traitor
A
Attack
R
R
Retreat
A
R
Retreat
Attack
Retreat

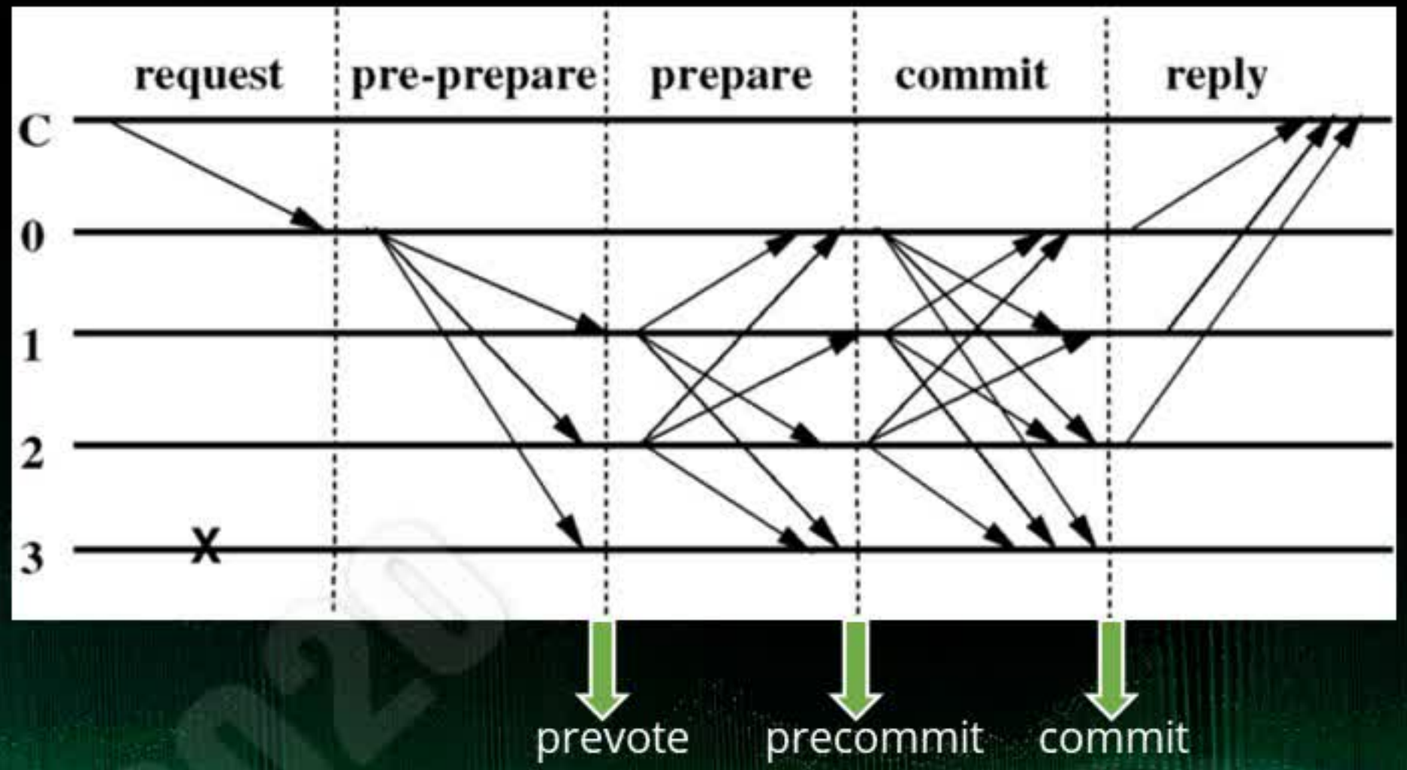
拜占庭将军问题

实用拜占庭容错算法 Practical Byzantine Fault Tolerance

Votes > 2/3
2 Rounds

拜占庭将军问题

实用拜占庭容错算法 Practical Byzantine Fault Tolerance



拜占庭将军问题

Proof of Work (PoW)

Proof of Stake (PoS)

Practical Byzantine Fault Tolerance(PBFT)

Delegate PoS (DPOS)

Hybrid Consensus

...

区块链漏洞画像

Blockchain Vulnerability Portraits

漏洞层次划分

- 业务
- 协议
- 架构
- 算法
- 实现

漏洞成因坐标

- 共识机制
- 账本逻辑
- 经济模型
- RPC服务
- P2P协议
- 合约虚拟机
- 数据库
- 语言特征
- ...

区块链漏洞定级标准

危害程度

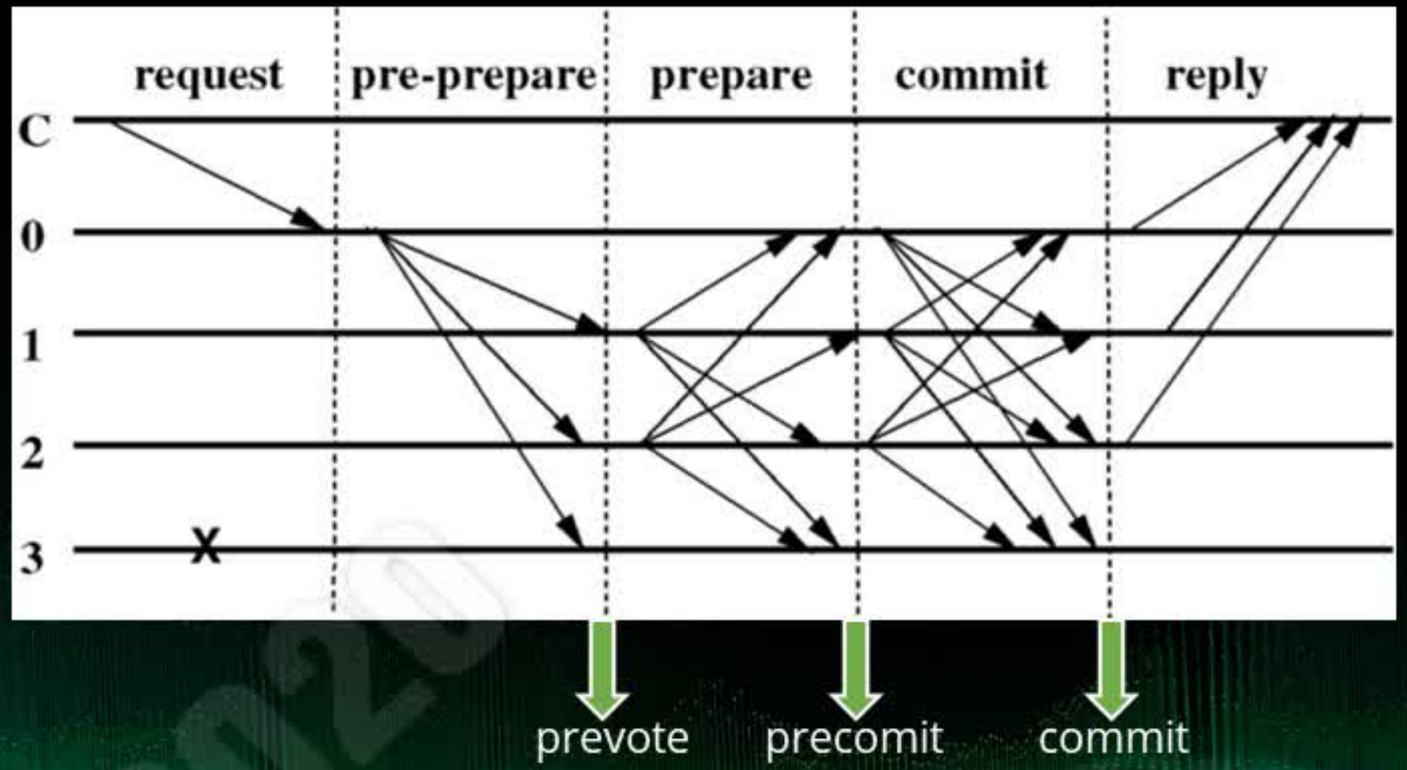
利用难度	严重危害	高危害	中危害	低危害
低难度	严重	高危	中危	低危
中难度	严重	中危	中危	低危
高难度	高危	低危	低危	低危
极高难度	低危	N/A	N/A	N/A

区块链漏洞案例
Blockchain Vulnerability Study Cases

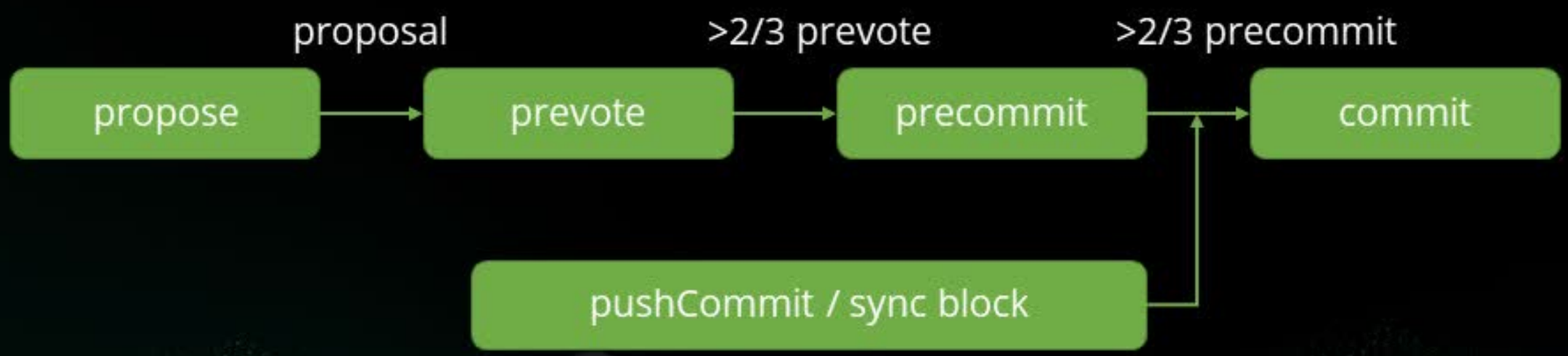
耳食漏洞

耳食漏洞

实用拜占庭容错算法 Practical Byzantine Fault Tolerance



耳食漏洞


proposal
>2/3 prevote
>2/3 precommit
propose
prevote
precommit
commit
pushCommit / sync block

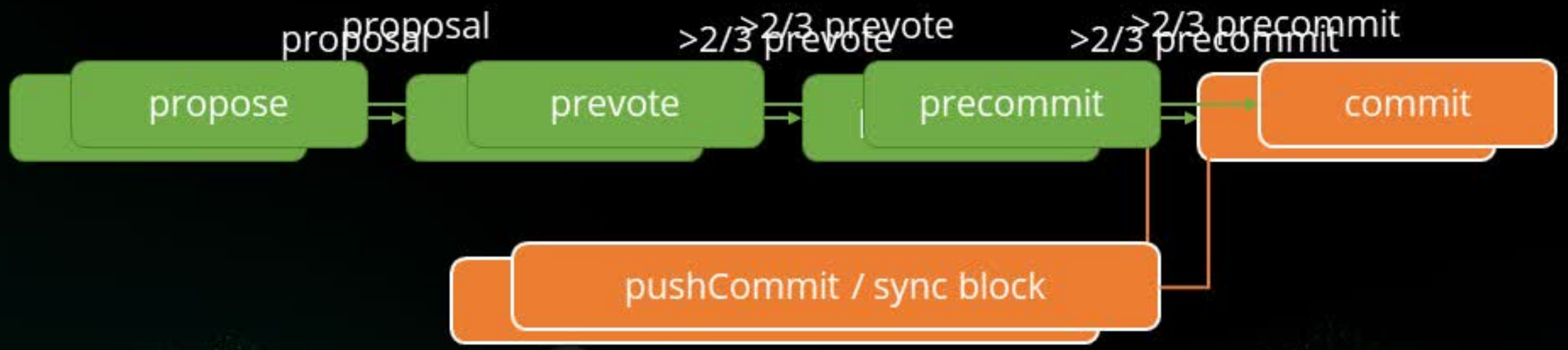
耳食漏洞


```
func Push(msg interface{}) {
    switch m := msg.(type) {

    case message.Commit:
        go func() {
            bft.cmtC <- *m
        }()
    }
}
```

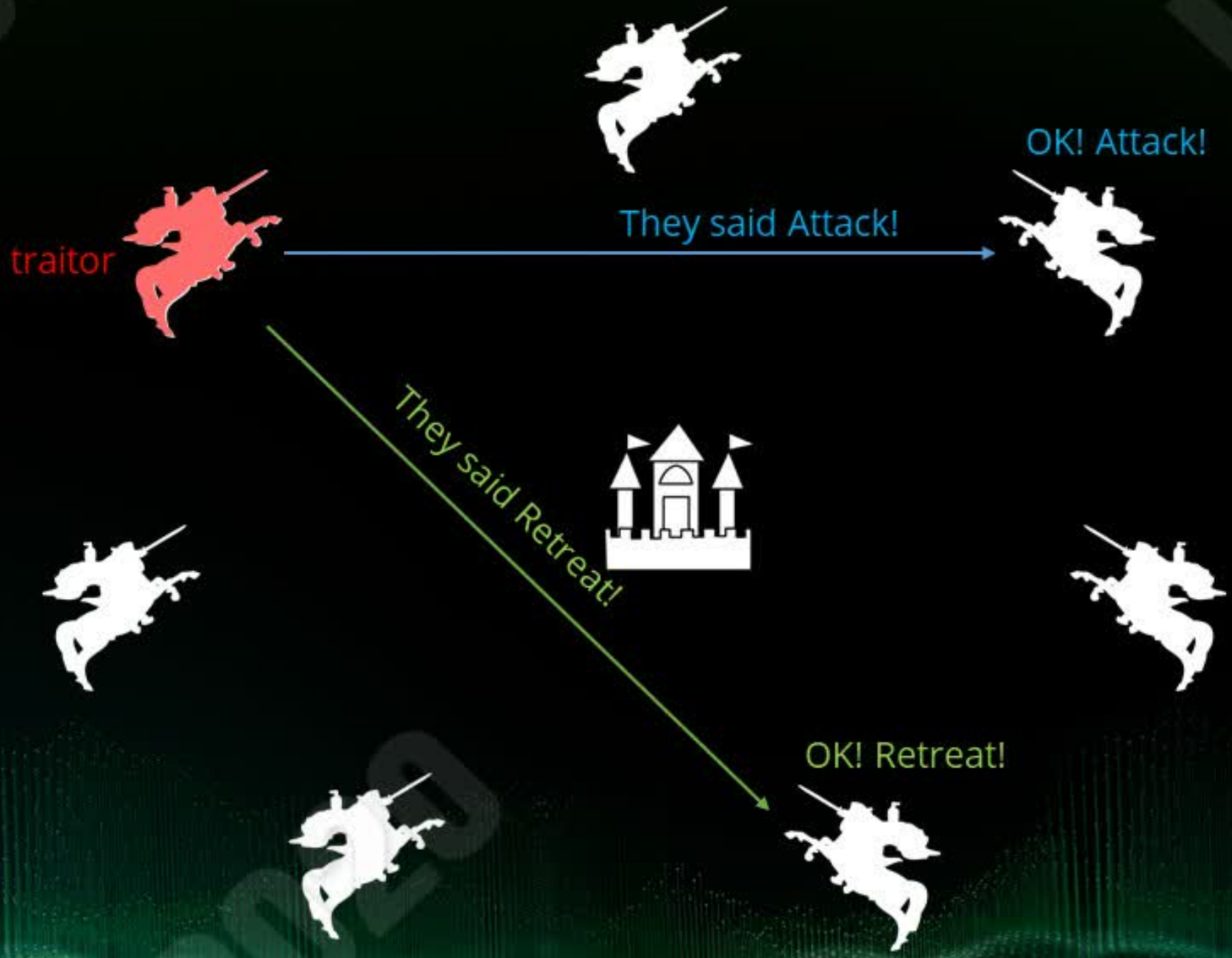
耳食漏洞


```
func (bft *BFT) start() {
    ...
    for {
        select {
        ...
        case cmt := <-bft.cmtC:
            bft.handleCmtRecords(&cmt)
        }
      }
}


func (bft *BFT) handleCmtRecord(record *message.Commit) {
    if bft.checkCmtExist(record) {
        return
    }
    ...
    bft.Commit(record)
}
```

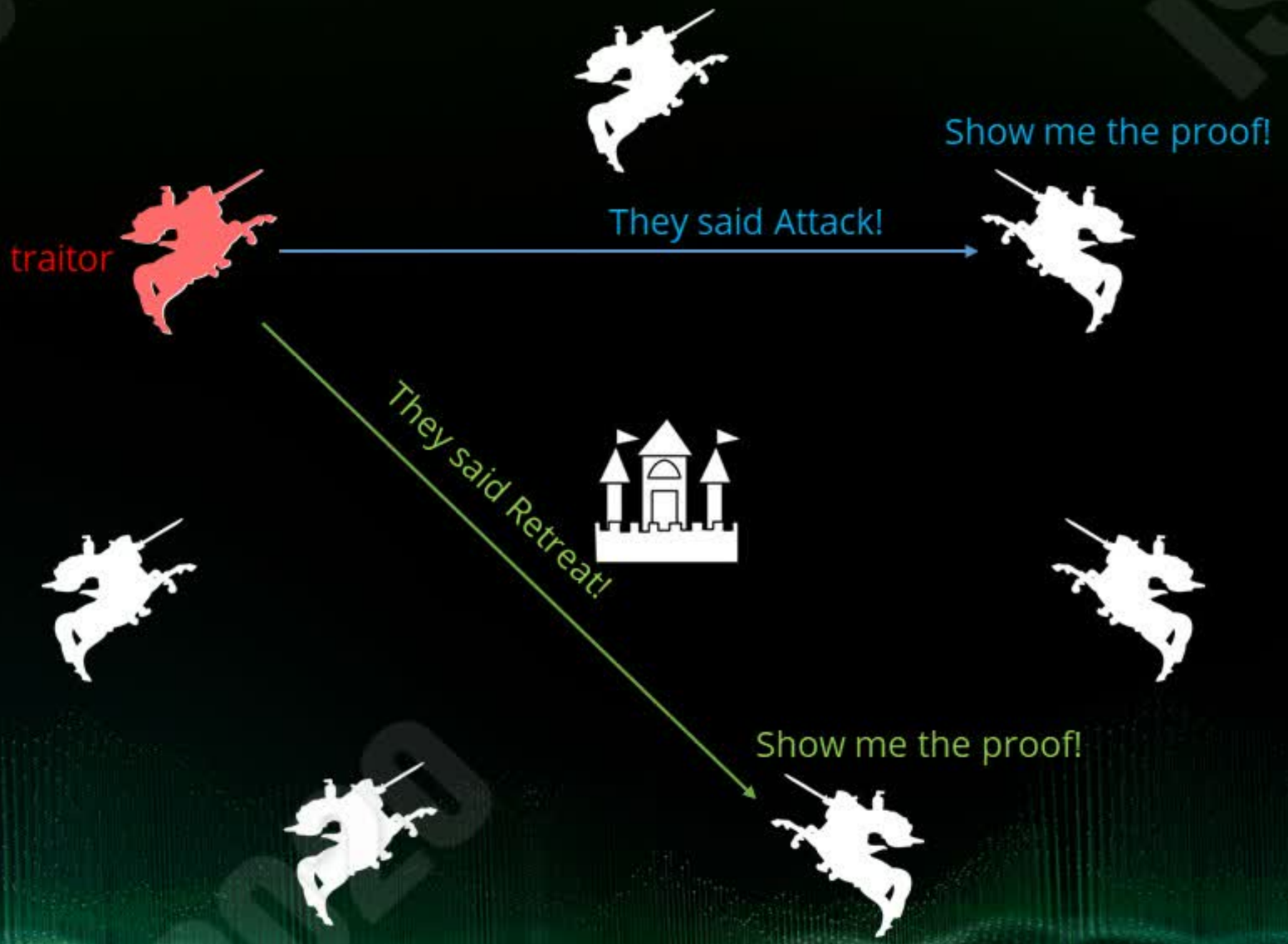
耳食漏洞


proposal
>2/3 prevote
>2/3 precommit
propose
prevote
precommit
commit
pushCommit / sync block

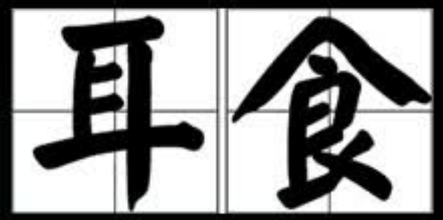
耳食漏洞


traitor
They said Attack!
OK! Attack!
They said Retreat!
OK! Retreat!

耳食漏洞


traitor
They said Attack!
Show me the proof!
They said Retreat!
Show me the proof!

耳食漏洞



耳食，汉语词汇，常见于文言文，谓不加省察，徒信传闻；也可作名词指传闻。

区块链漏洞定级标准

危害程度

利用难度

	严重危害	高危害	中危害	低危害
低难度	严重	高危	中危	低危
中难度	严重	中危	中危	低危
高难度	高危	低危	低危	低危
极高难度	低危	N/A	N/A	N/A

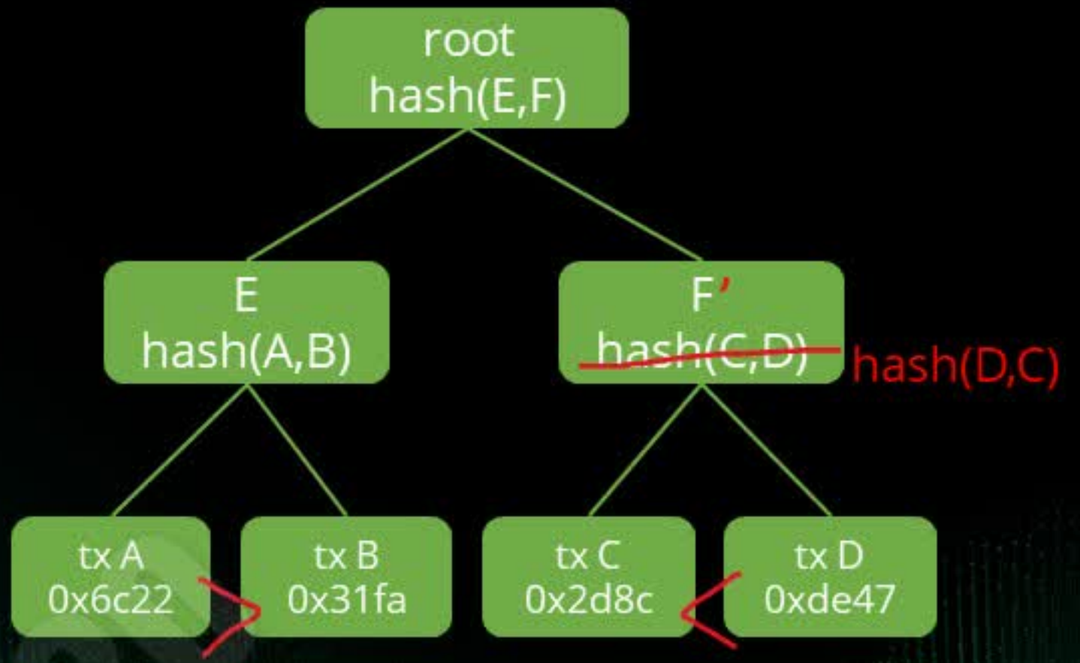
双生树漏洞

什么是Merkle tree?

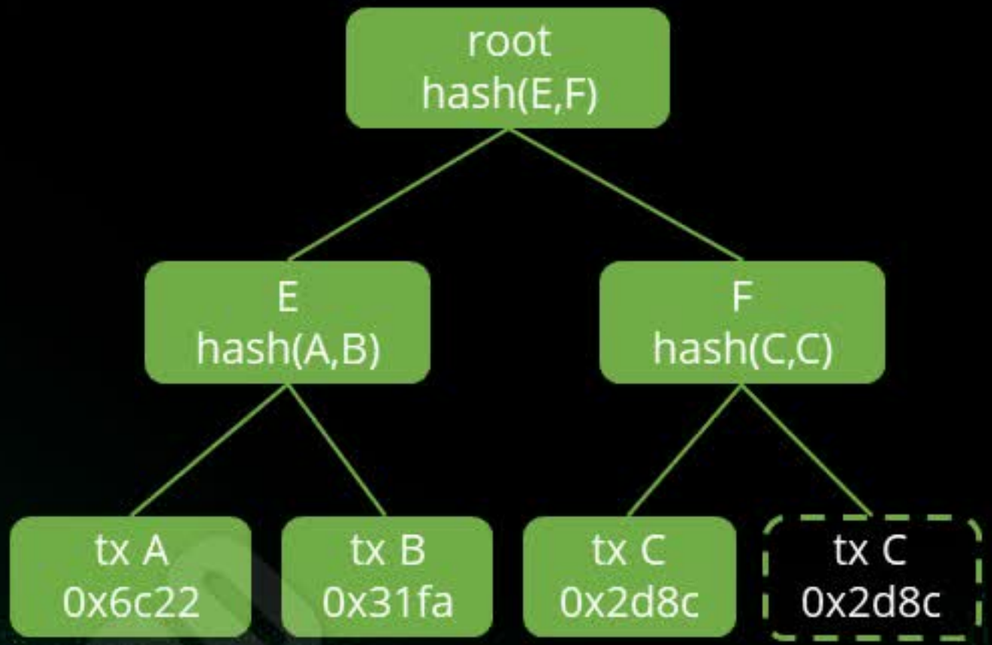
Merkle tree 实例伪码

```
if hashList[i+1] == None :
    hashList[p] = sha3(hashList[i]+hashList[i])
else :
    if hashList[i] > hashList[i+1] :
        hashList[p] = sha3(hashList[i]+hashList[i+1])
    else:
        hashList[p] = sha3(hashList[i+1]+hashList[i])
```

Merkle tree 实例伪码

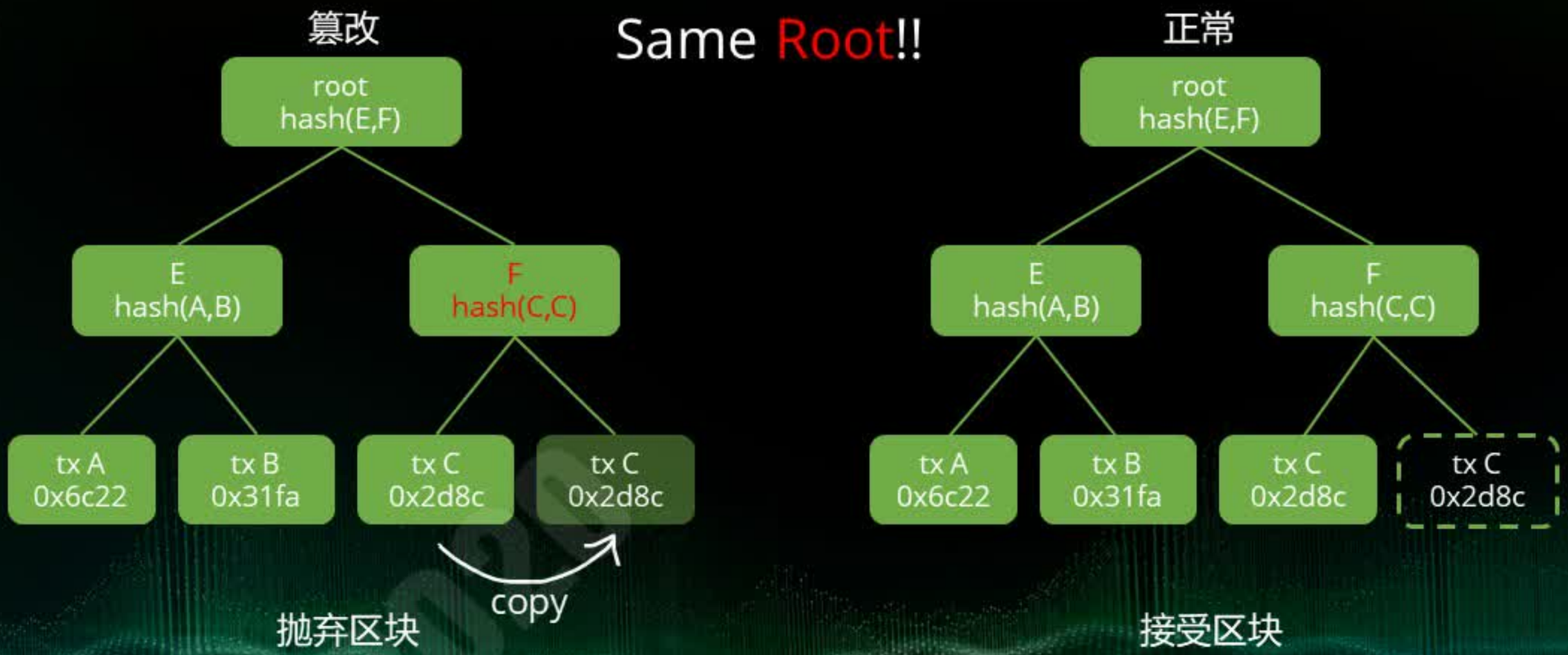


CVE-2012-2459



CVE-2012-2459

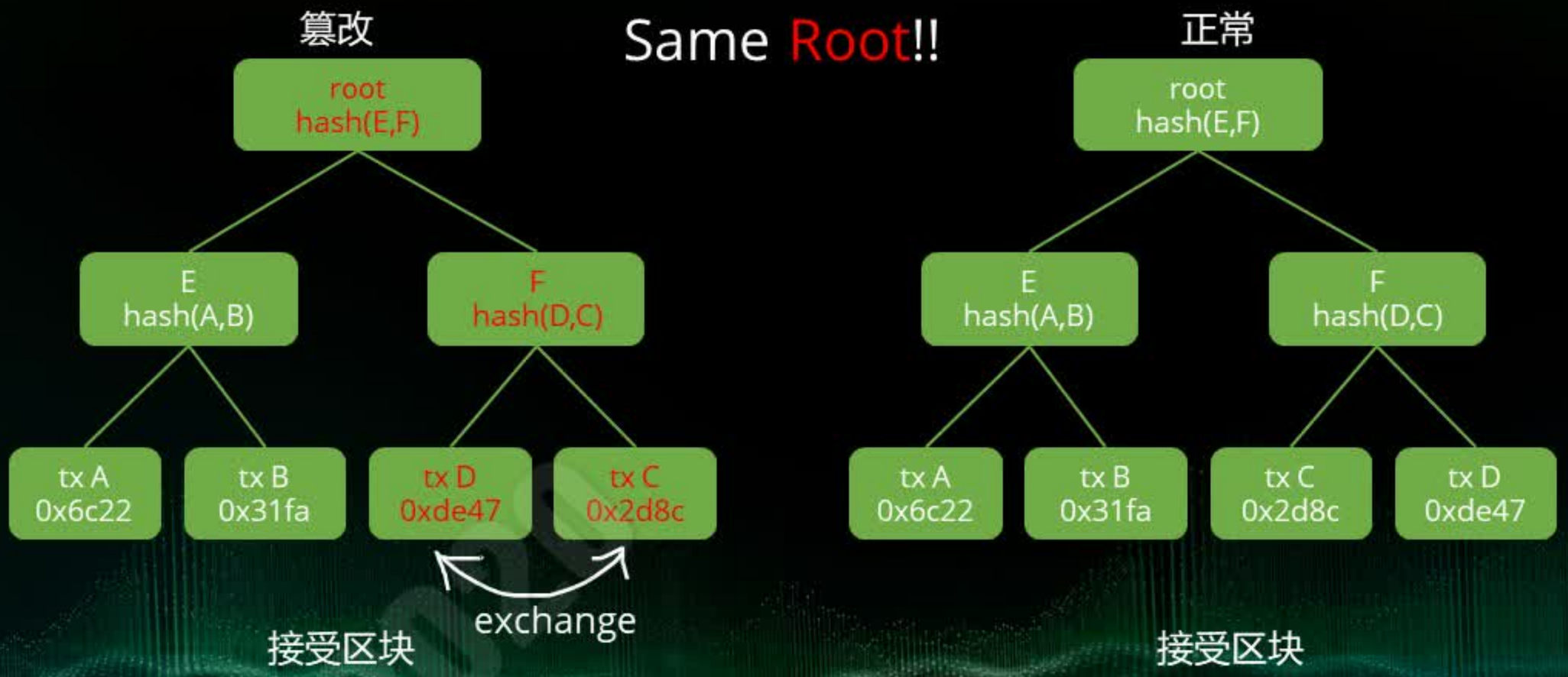
Same Root!!



Merkle tree 实例伪码

```
if hashList[i+1] == None :
    hashList[p] = sha3(hashList[i]+hashList[i])
else :
    if hashList[i] > hashList[i+1] :
        hashList[p] = sha3(hashList[i]+hashList[i+1])
    else:
        hashList[p] = sha3(hashList[i+1]+hashList[i])
```

Merkle tree 实例伪码



影响

Init state:
A:5, B:0, C:0

A ------5------> B ✓
B ------5------> C ✓

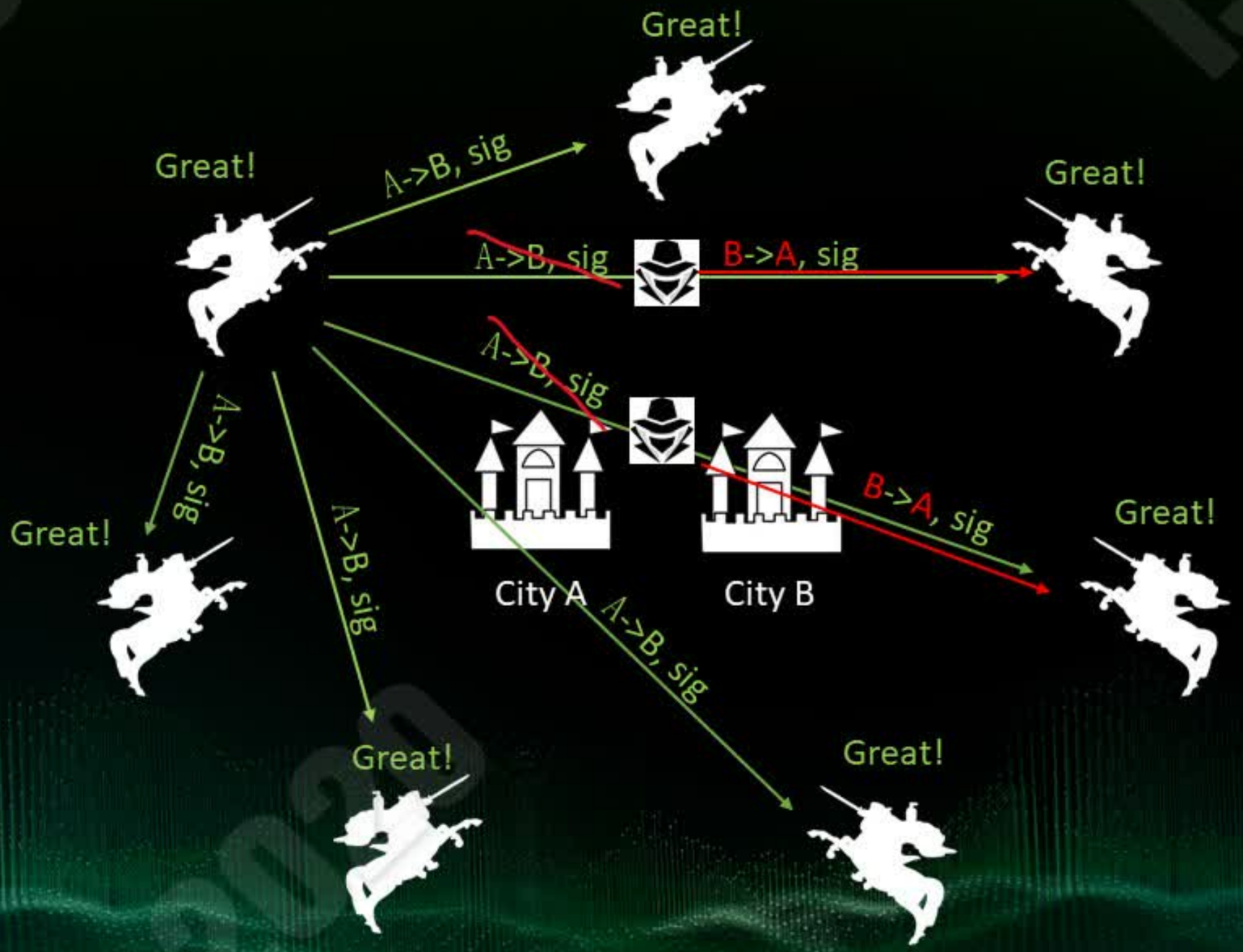
End state:
A:0, B:0, C:5

Init state:
A:5, B:0, C:0

B ------5------> C ✗
A ------5------> B ✓

End state:
A:0, B:5, C:0

双生树漏洞


Great!
Great!
A->B, sig
Great!
A->B, sig
B->A, sig
A->B, sig
A->B, sig
B->A, sig
Great!
Great!
Great!
Great!
Great!
City A
City B

双生树漏洞

难以察觉

且不可逆

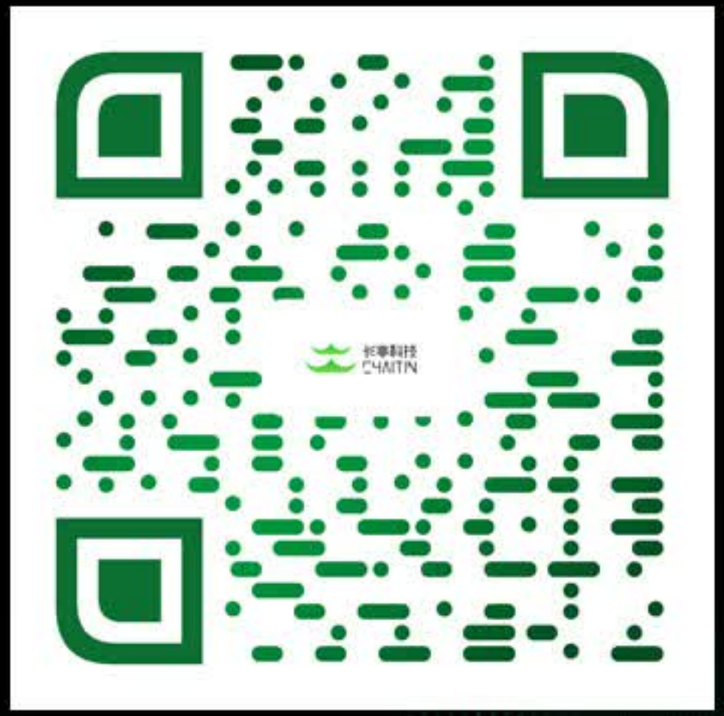
区块链漏洞定级标准

危害程度

利用难度

	严重危害	高危害	中危害	低危害
低难度	严重	高危	中危	低危
中难度	严重	中危	中危	低危
高难度	高危	低危	低危	低危
极高难度	低危	N/A	N/A	N/A

区块链安全检查项



THANKS

ISC 2020
第八届互联网安全大会
INTERNET SECURITY CONFERENCE 2020
数字孪生时代下的新安全
New Security in the Digital Twin Era