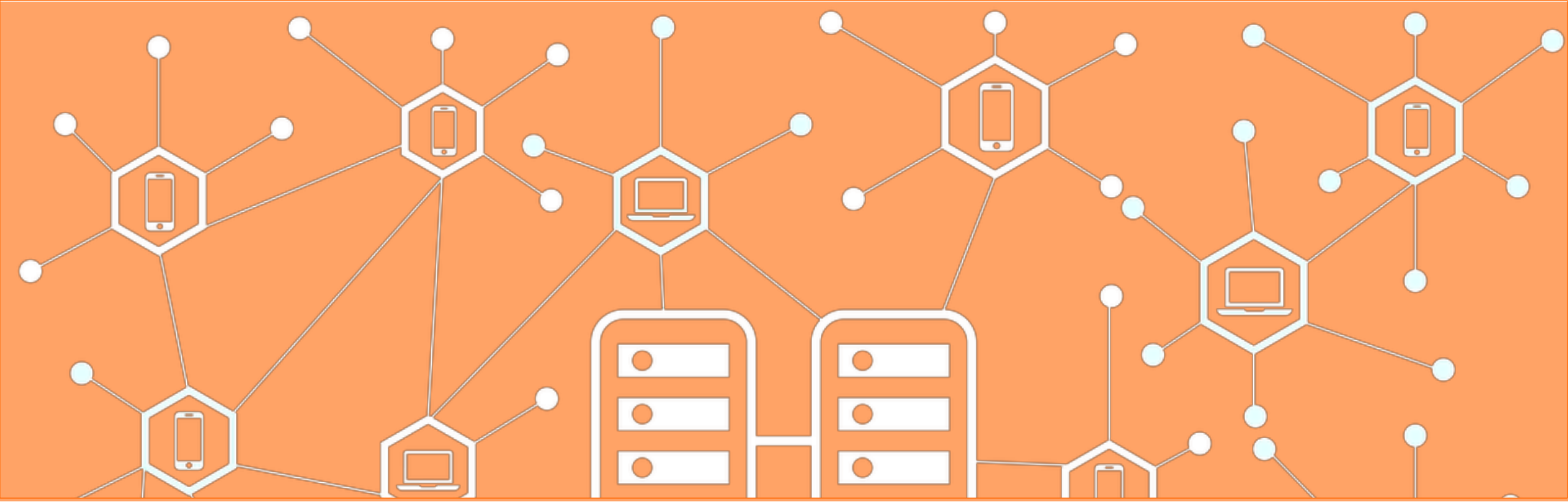


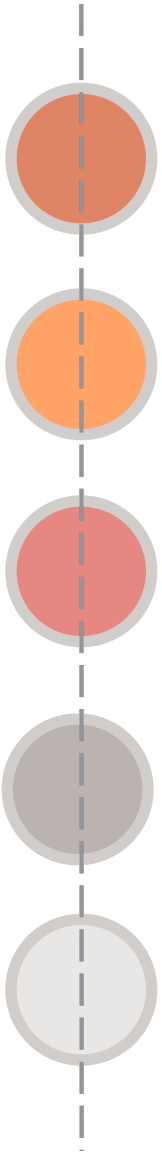
EPFL

Orange Cyberdefense

Security Assessment of Authentication and Authorization Mechanisms in Ethereum, Quorum, Hyperledger Fabric and Corda

Marie-Jeanne Lagarde, Master's thesis 2018-2019

Agenda



Introduction

1. Motivation
2. Scope
3. Research Questions

Methodology

1. Roadmap
2. Analysis framework
3. Hypothesis
4. Experiments
5. Threat model

Platform Security Assessment

1. Ethereum
2. Quorum
3. Hyperledger Fabric
4. Corda

Discussion

Conclusion

Introduction

1. Motivation
2. Scope
3. Research Questions

Methodology

1. Roadmap
2. Analysis framework
3. Hypothesis
4. Experiments
5. Threat model

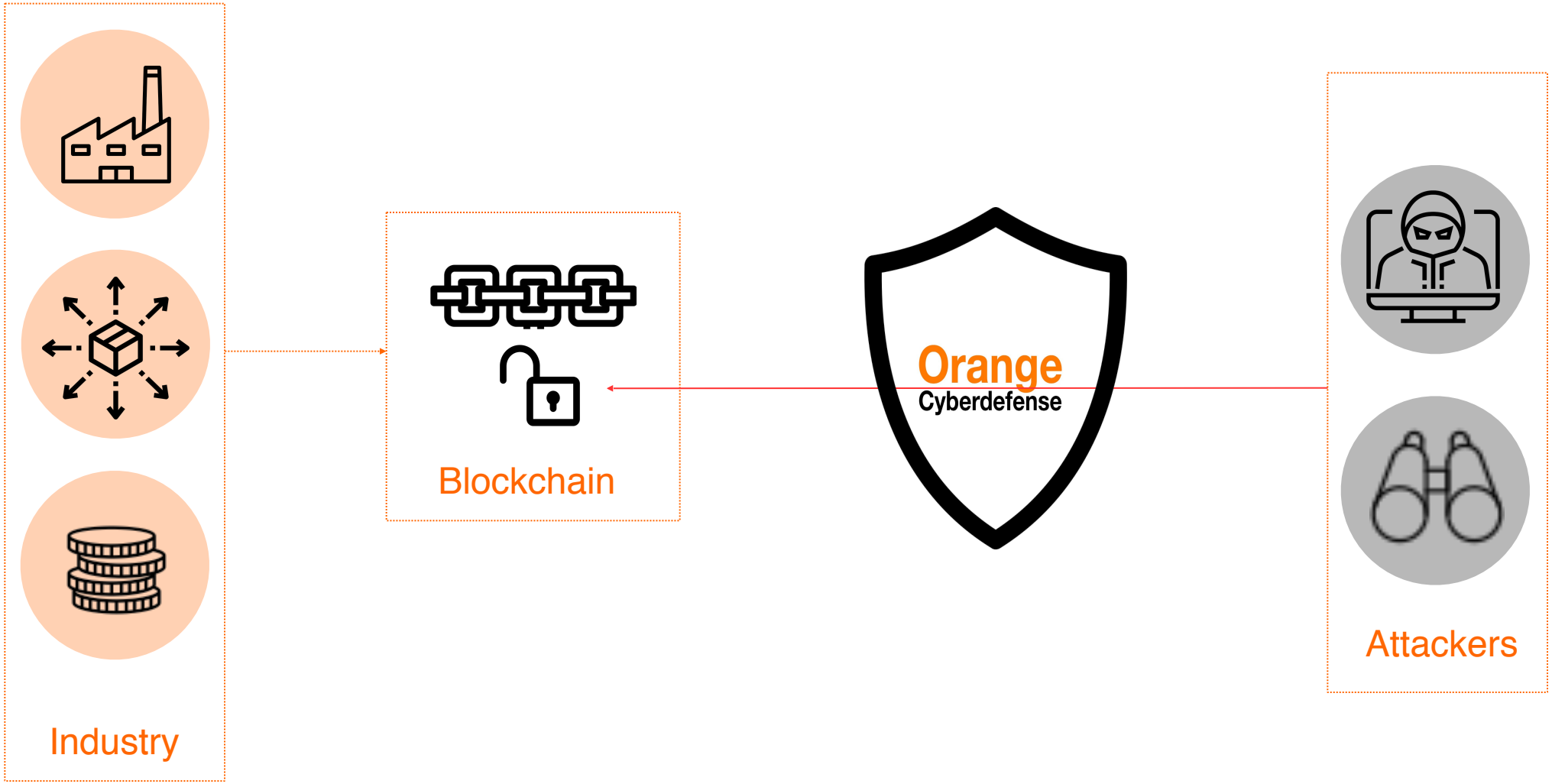
Platform Security Assessment

1. Ethereum
2. Quorum
3. Hyperledger Fabric
4. Corda

Discussion

Conclusion

Motivation



Scope: definition of the subject

"Security of main blockchain technologies"

Ethereum
Quorum
Hyperledger Fabric
Corda

Authentication: verifying the proclaimed identity

Authorization: verifying the access rights

Research Questions

RQ1 How are the mechanisms designed and implemented?

RQ2 What are the vulnerabilities?

RQ3 How can we harden the systems?

Introduction

1. Motivation
2. Scope
3. Research Questions

Methodology

1. Roadmap
2. Analysis framework
3. Hypothesis
4. Experiments
5. Threat model

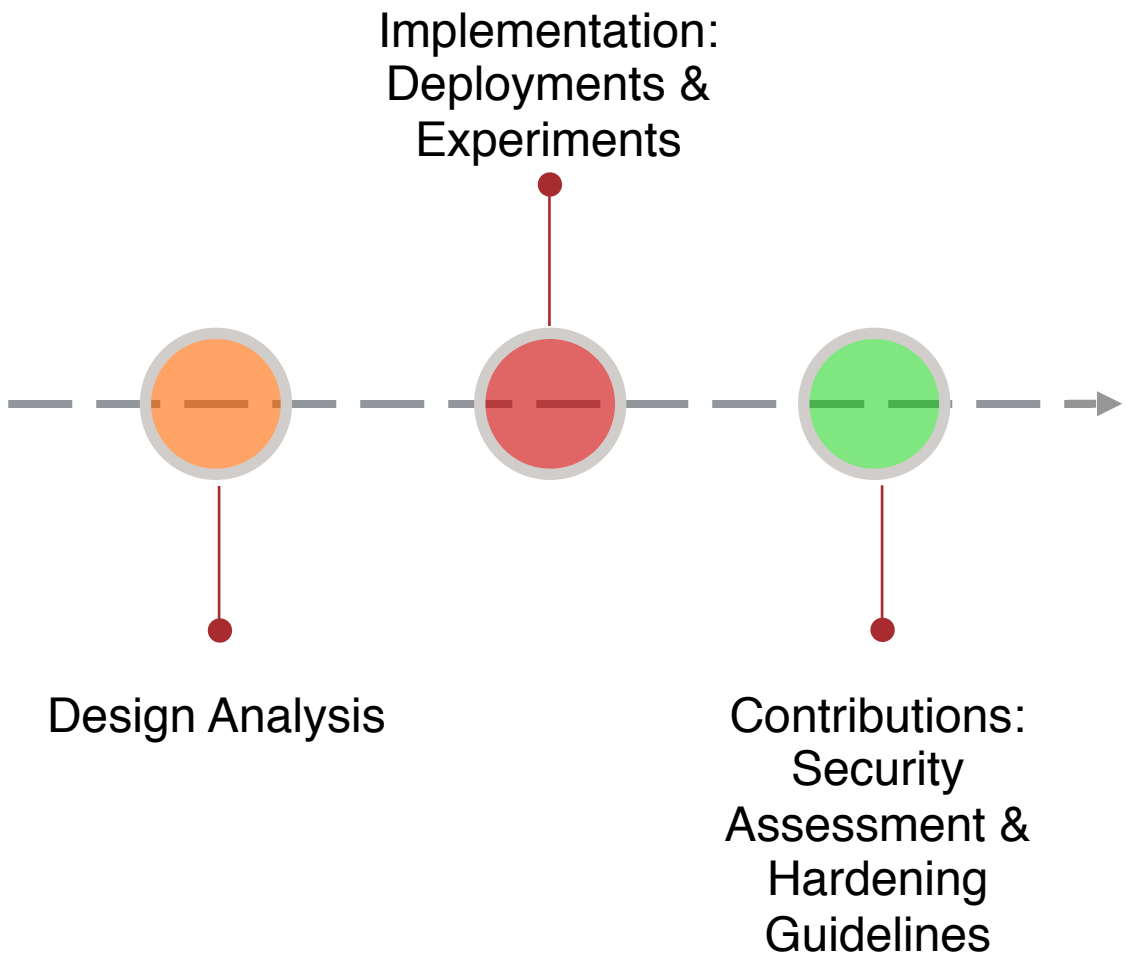
Platform Security Assessment

1. Ethereum
2. Quorum
3. Hyperledger Fabric
4. Corda

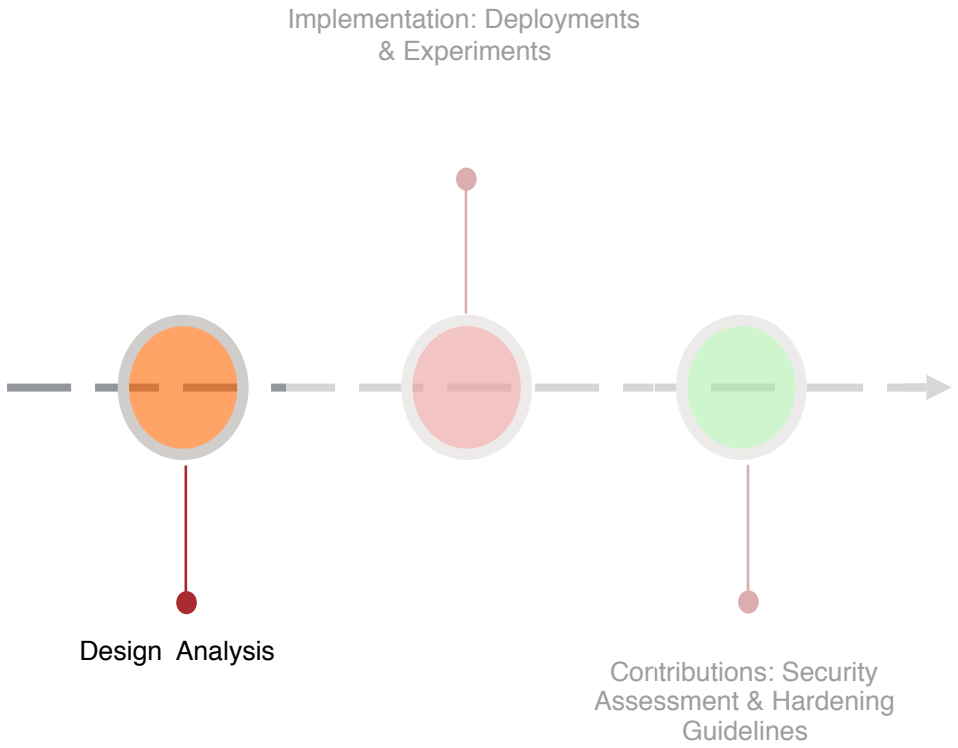
Discussion

Conclusion

Roadmap


Implementation:
Deployments &
Experiments
Design Analysis
Contributions:
Security
Assessment &
Hardening
Guidelines

Analysis Framework



1) Network permissioning

2) Transaction

3) Remote user (off-site location access)

Hypothesis for deployment

H1

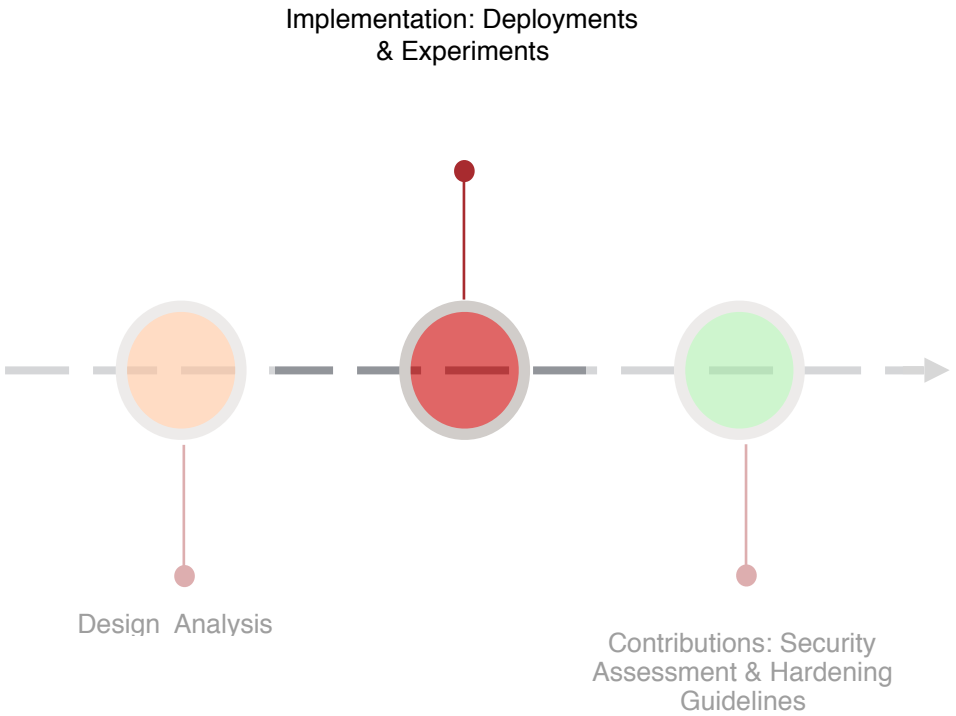
Most deployed platform versions are the most likely to be targeted by attackers

H2

Users tend to adapt their systems from existing official sample scripts

Ethereum	Quorum	Hyperledger Fabric	Corda
Geth Client v1.8.23	Quorum 2.2.1 using 7nodes demo with Tessera	Hyperledger Fabric v1.3.0 using Deploy your first network tutorial	Corda Example Dapp v3.3

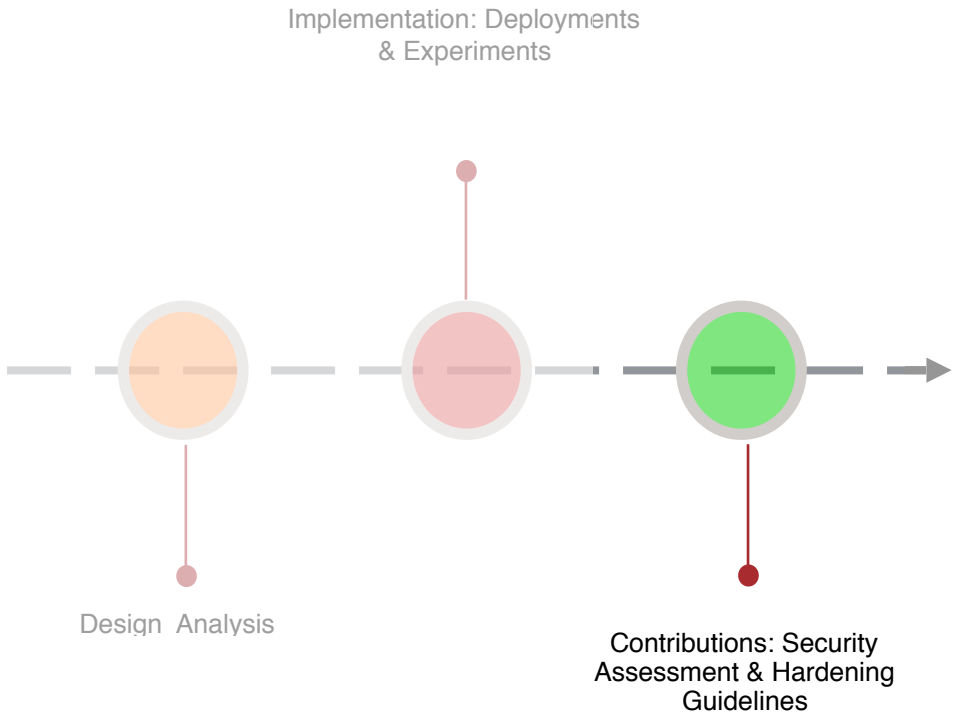
Experiments: a Total of 18 Experiments Conducted



Role of the experiments:

- Assess behaviour
- Test uncertain behaviours
- Assess the popularity of known attacks
- Demonstrate possible vulnerabilities

Threat model definition



1) Authentication threats:

- Brute force / dictionary attack
- Password sniffing attack
- Key compromise attack
- Replay attack
- MITM / Session hijacking
- Source non-repudiation
- DDoS and DoS

2) Authorization threats:

- Elevation of privileges
- Exploitation of access granting vulnerabilities

3) Security single points of failure

4) Default parameters vulnerabilities

Introduction

1. Motivation
2. Scope
3. Research Questions

Methodology

1. Roadmap
2. Analysis framework
3. Hypothesis
4. Experiments
5. Threat model

Platform Security Assessment

1. Ethereum
2. Quorum
3. Hyperledger Fabric
4. Corda

Discussion

Conclusion

Ethereum: Authentication and Authorization Mechanisms

Authenticated channel for node communication	**key-based**
Transaction sender authentication	**key-based**
General remote user authentication	**non-existing**
Account owner remote authentication	**passphrase-based**
Remote user authorization	**Depends on which modules are enabled**

Experiment: RPC honeypot

Gather information about the motives and tactics of attackers

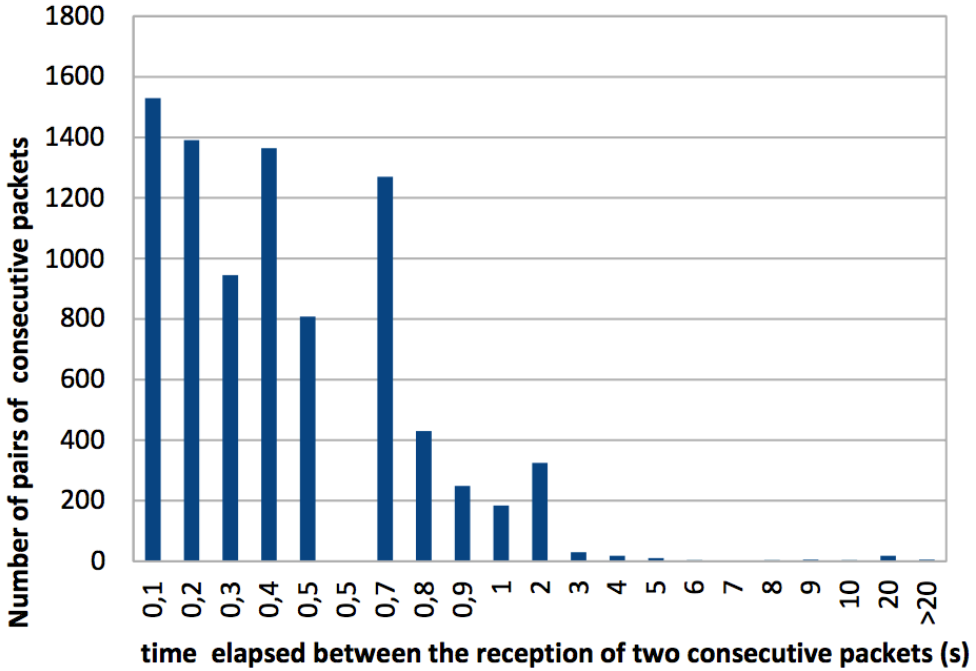
Goals

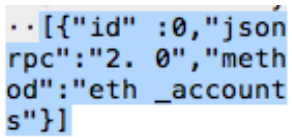
- Default account unlock duration 300s
- Measure **the likelihood of an attack occuring within this lapse of time**

Setup

- Deploy a node with **all RPC modules enable** listening **to all incoming port**.
- Capture the traffic during **one and a half hour** using Wireshark.

Experiments: RPC honeypot results





Packet captured with Wireshark

- **10849 packets** observed from 13 different attackers
- The largest interval between two attempts is **20 seconds**
- **Dictionary attack**
- Main purpose is **financial benefits**

Orange Cyberdefense EPFL

Authentication Vulnerabilities

Node authenticated Communication	Remote account owner authentication
Isolate a node from the network	Funds stealing

Key compromise: **Elliptical curve secp256k1 vulnerable to Pollards rho speed up attacks *(Hartwig Mayer research*** [1]*, only successful on 109 bit long keys)*

Quorum : Authentication and Authorization Mechanisms of Ethereum Permissioned Platform

Node authentication	**Key-based and optionally certificate-based if TLS CA**
Transaction sender authentication	**key-based**
Transaction receiver authorization	**ACL**
General remote user authentication	**non-existing**
Account owner remote authentication	**passphrase-based**
Remote user authorization	**Depends on which modules are enabled**

TLS can be enabled in the modes: CA, Trust on First use (TOFU), whitelist

Experiments : permissioning & honeypot

1) Different permissioning files → Triggers inconsistent behaviour

2) Dynamicity of addition/revocation →

TLS Mode	None	CA	TOFU & CA	TOFU	Whitelist
Dynamic addition of a node	Yes	Yes	Yes	Yes	No
Dynamic revocation of a node	Yes	Yes via CRL	Yes via CRL	No	No

3) Honeypot for espionage → No attacker is detected

Authentication Vulnerabilities: many similarities with Ethereum

Node authentication	Remote account owner authentication	Block communication via HTTP
Espionage/Sabotage	Espionage/Sabotage	Espionage/Sabotage

Key compromise: **Elliptical curve secp256k1 vulnerable to Pollards rho speed up attacks *(Hartwig Mayer research*** [1], *only successful on 109 bit long keys)*

Vulnerabilities

Single points of failure	Elevation of Privileges	Exploitation of Access Granting and Revoking Vulnerabilities
• Root CA is used for TLS and Identity	• Module-enabling attacks • Transaction access using RPC	• TLS TOFU and Whitelist modes do support node revocation • Different permissioning files

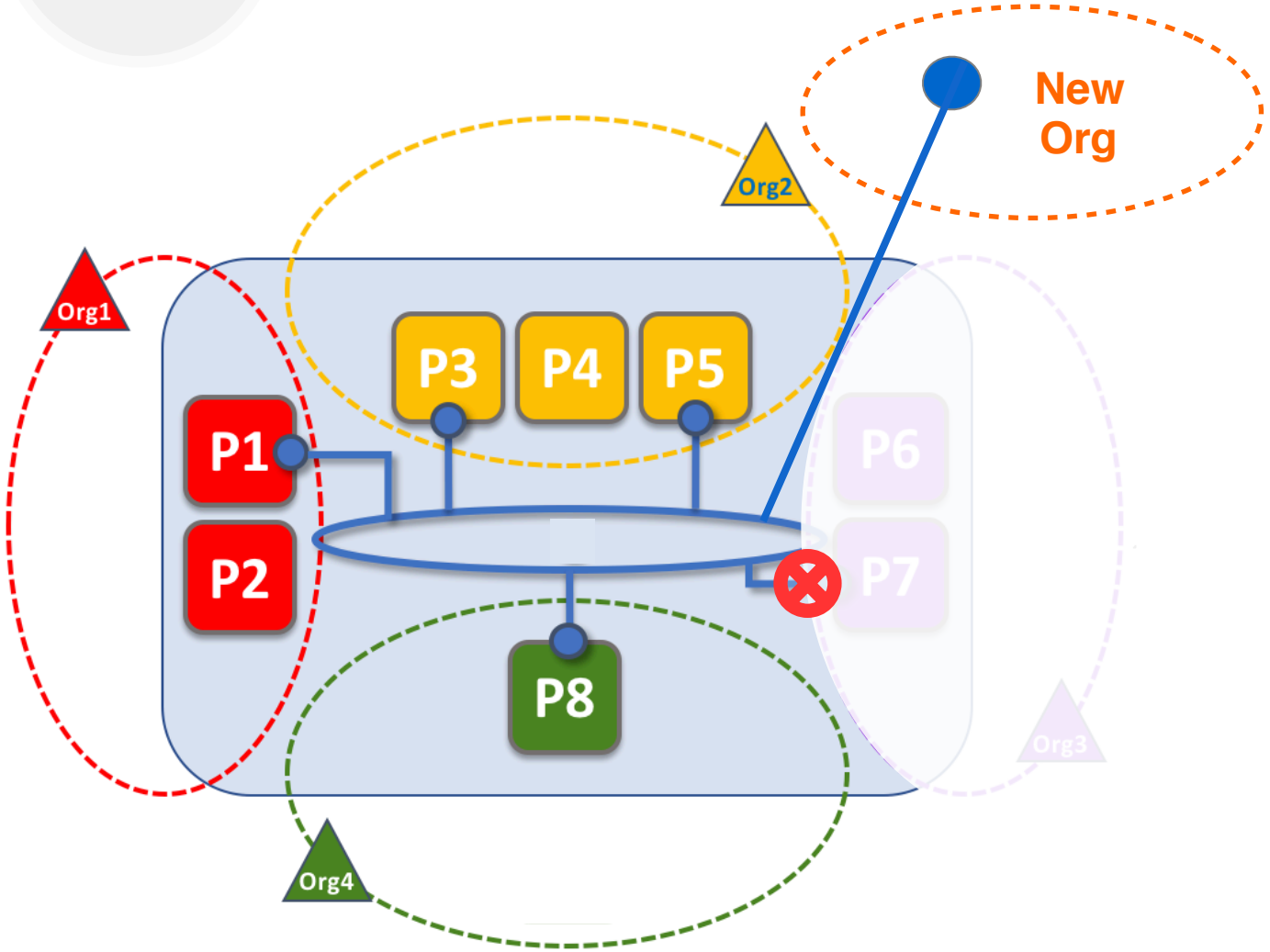
Additional vulnerability: TOFU mode prevents a node from changing a compromised key pair.

Hyperledger Fabric : Authentication and Authorization Mechanisms

Message sender authentication	**Certificate-based**
Node role granting	**ABAC**
Transaction sender authorization	**ACL**

Hyperledger Fabric offers a module called Fabric CA which handles certificate issuing.

Experiment



A malicious majority can prevent an organization from being noticed of a change in the configuration

1) **Majority vote Org3 removal (but Org3 is not noticed that it has been removed)**

2) **Majority vote New Org Addition**

3) **Majority vote Org3 Addition (but Org3 is not noticed that it has been re-added)**

Authentication Vulnerabilities

Message sender authentication	Message sender authentication in Fabric CA
Sabotage (inoperative network)	Malicious registrations
	
By default, mutual TLS disabled	**By default, TLS disabled**

Vulnerabilities

Single points of failure	Elevation of Privileges	Exploitation of Access Granting and Revoking Vulnerabilities
• Single node orderer • Single root CA: if same root CA is used for TLS and MSP identities	Lack of smart contract sandboxing causing possible elevation of privileges (Nettitude, *Security Assessment report [2]*)	• No support to revoke TLS certificates • No expiration of identity certificate

Corda: Authentication and Authorization Mechanisms

Message sender authentication	**Certificate-based**
Node role granting	**ABAC**
Transaction sender authorization	**Depends on notaries nodes**
General remote user authentication	**Password-based**
Remote user authorization	**Capability list**

Corda: two flavours network

Corda Business Network

- Publicly available
- Identity registration managed by R3
- Possibility to build a restricted business network for nodes using the same smart contract
- Cost 2500$/year

Corda Independently Managed Network

- Root CA, Network Map and Intermediate CA must be implemented from scratch (build HTTP servers)
- Free

Deployment out of this project scope (price/complexity)
Experiment: Deployment of Corda official demo, assessment of default parameters

Authentication Vulnerabilities

Message sender authentication	Remote user authentication
Sabotage	User Impersonation Espionage Sabotage
	
By default, mutual TLS enabled	**By default, RPC TLS disabled**

Vulnerabilities

Single points of failure	Elevation of Privileges	Exploitation of Access Granting and Revoking Vulnerabilities
• Single Root CA • Single Doorman • Single Network Map • Single notary configuration	Elevation of privileges attack using a smart contract (Corda does not implement specific security controls to prevent privileges escalation)	For Corda Business Network, obvious threat that privilege granting depends on an untrusted third party (R3)

Additional vulnerability: User prevented from changing a compromised key pair (NetworkMapClient throws an exception when trying to publish a NodeInfo corresponding to a name that has been registered before)

Introduction

1. Motivation
2. Scope
3. Research Questions
4. Hypothesis

Methodology

1. Roadmap
2. Analysis framework
3. Hypothesis
4. Experiments
5. Threat model

Platform Security Assessment

1. Ethereum
2. Quorum
3. Hyperledger Fabric
4. Corda

Discussion

Conclusion

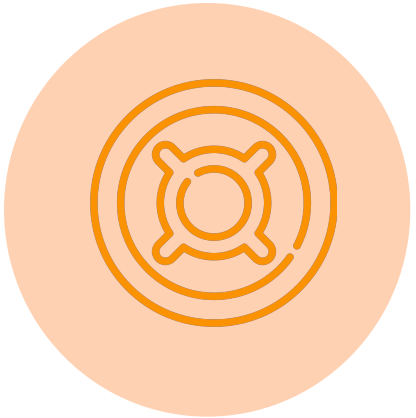
Discussion

Hypothesis 2 Validation

Illegal to conduct a large scan of running nodes.

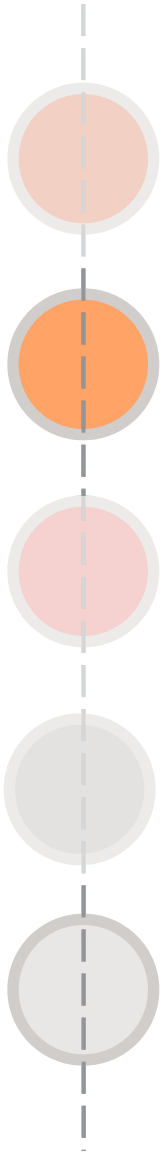
Platforms Obsolescence

Platform design and implementation are often subject to change.



Threat Model Definition

Platforms have a singular architecture, thus the model might not properly cover potential threats of other platforms.



Introduction

1. Motivation
2. Scope
3. Research Questions

Methodology

1. Roadmap
2. Analysis framework
3. Hypothesis
4. Experiments
5. Threat model

Platform Security Assessment

1. Ethereum
2. Quorum
3. Hyperledger Fabric
4. Corda

Discussion

Conclusion

Conclusion

- Recurrent vulnerabilities among platforms:
  - Weak or non-existing remote user authentication schemes
  - Absence of password policy
  - Lack of sandboxing of smart contract execution
- TLS optional in permissioned blockchain → wide exposure
- Weak default parameters are frequently preferred to ease software adoption and functionality demonstrations irrespective of the consequences on security

References

[1] Hartwig Mayer, CoinFabric. *ECDSA Security in Bitcoin and Ethereum: a Research Survey*. [Online; accessed 14-February-2019]. 2016. URL: https://pdfs.semanticscholar.org/1785/6bad4335c8ca7419aab2c715ea25ce5e0621.pdf.

[2] Graham Shaw. *Security Assessment Management Report*. [Online; accessed 14-February-2019]. 2017. URL: hhttps://wiki.hyperledger.org/display/HYP/Project+Audits?preview=%2F2393550%2F2393585%2Fmanagement_report_linux_foundation_fabric_august_2017_v1.1.pdf