Combining Multiple ID's, Attributes, and Policies to Provide Secure Access Control within Hyperledger Fabric

Author: Daan Gordijn, Supervisor: Roland Kromes, Responsible Professor: Kaitai Liang
Cyber Security Group, Delft University of Technology

1. Background

Hyperledger Fabric
- A "permissioned distributed ledger technology platform" [2]
- Solves many enterprise-level issues of traditional, permissionless blockchain technologies such as Bitcoin or Ethereum

Secure access control in Fabric
- Currently implemented by X.509 user certificates
  - Must be issued by Certificate Authorities (CA's)
  - Must be validated by Membership Service Providers (MSP's)

Research into attribute-based access control
- Has been performed in many studies such as [3], [4] and [5]
- Has **not** been much applied specifically in the context of Fabric
- The concept of combining multiple ID's, attributes, and policies has **not** been studied in-depth

Problems
- Lack of **decision variables** for access control decisions
- Potential of **role explosion** in large-scale organizations
- Inability to effectively implement the **principle of least privilege**
  - Potential to expose sensitive business information to unintended users
  - Potential to be too restrictive and exclude users from required information

2. Research Question

Q: "How can secure access control in Hyperledger Fabric be guaranteed by combining multiple ID's, attributes, and policies with the components that regulate access control?"

- **What** is Hyperledger Fabric?
- **What** is secure access control in the context of Fabric?
- **What** are the components that regulate access control in Fabric?
- **How** can multiple ID's, attributes, and policies be combined in Fabric?
- **How** are the components for access control currently interacting in Fabric?
- **What** is the performance impact of ID-, attribute-, and policy-based access control in Fabric?

3. Methodology

Combination of **literature research** and **implementation** of custom smart contracts which provide access control using ID's, attributes, and policies.

Milestone 1
- Study Hyperledger Fabric Documentation
- Setup Local Test-Network

Milestone 2
- Study Literature
- Implement X.509 Generation CLI (Using Fabric CA)
- Implement Smart Contract 1 ("Access Controller")

Milestone 3
- Study Literature
- Implement Smart Contract 2 (Live Demo)
- Implement Client Application (Live Demo)

Milestone 4
- Study Literature
- Analyze Performance

Milestone 5
- Formulate Conclusions
- Formulate Future Work

4. Design

Current Implementation
- **Certificate Authorities** and **Membership Service Providers** provide the first layer of security by checking if certificates are issued by trusted parties

Proposed Implementation
- **Multiple attributes** can be combined by combining one or multiple attribute checks (EQUALS, INCLUDES) using Boolean operators (AND, OR, NOT)
- **Multiple policies** can be combined by having a smart contract maintain a different access policy for each operation that can be performed with it
- **Multiple ID's** can be combined by hashing and signing a "parent certificate", and storing this hash and signature in the "child certificate"
- By maintaining all parent certificates in a **hashmap** on the shared ledger, smart contracts on the blockchain can retrieve and verify these certificates

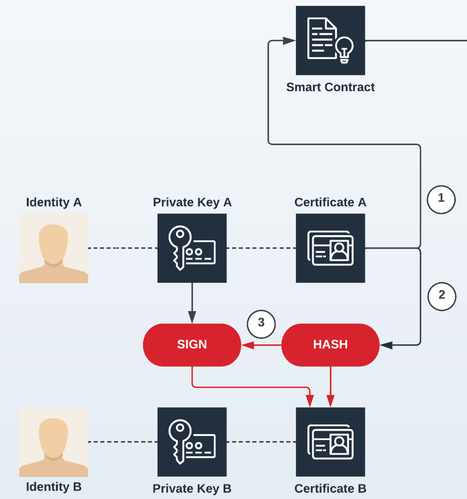


Figure 2. Combining multiple identities

- A **security smart contract** extracts a client's certificate and determines whether access should be granted or denied, based on the provided policy
- Any **other smart contract** can invoke this "special" smart contract to determine whether it should handle the submitted request or deny access

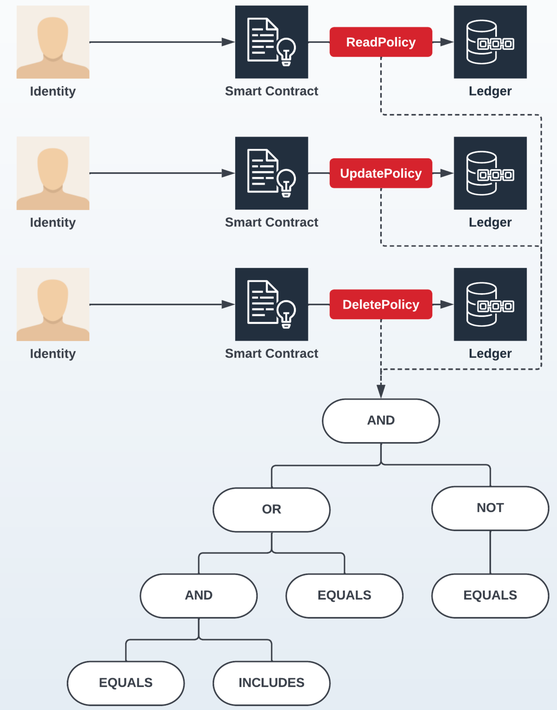


Figure 1. Combining multiple attributes and policies

5. Results

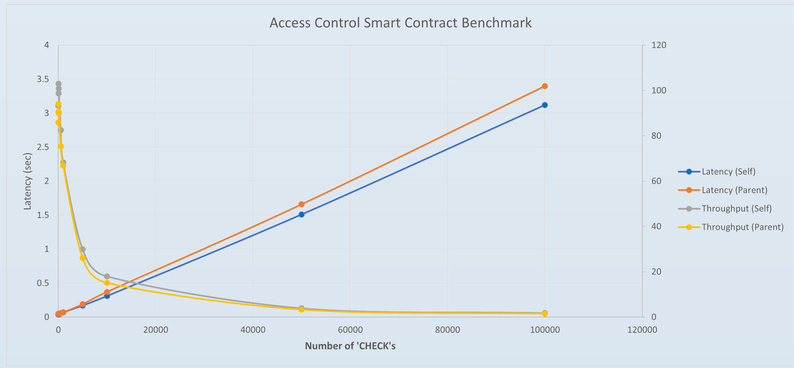
- The latency of this aforementioned "**security smart contract**" is almost linear with respect to the number of attributes that it needs to check
- For real-world cases, the increase in latency with respect to the number of attribute checks is *minor*

Latencies (Access with "own attributes")

1 attribute	0.04 seconds
1000 attributes	0.07 seconds
10,000 attributes	0.31 seconds
100,000 attributes	3.12 seconds

Latencies (Access with "parent attributes")

1 attribute	0.05 seconds
1000 attributes	0.07 seconds
10,000 attributes	0.37 seconds
100,000 attributes	3.40 seconds



6. Conclusions

- In this study, a **new implementation** for secure access control within Hyperledger Fabric blockchain technology has been proposed
- **Multiple attributes** have been combined using a simple scheme that build access policies by combining attribute checks with Boolean operators
- **Multiple policies** have been combined by storing multiple access policies on the blockchain ledger, and dynamically selecting the suitable one
- **Multiple ID's** have been combined by setting the hash and signature of a parent certificate as attributes while storing these parents on the blockchain
- The security smart contract, certificate generation tool ("certgen"), and demo application have been implemented and are publicly **available via GitHub**[1]
- The **runtime overhead** caused by the invocation of the special smart contract was analyzed, and has shown to be *minor* in comparison with the base case

7. Future Work

- Research if it is possible to improve the runtime of the current access control smart contract to reduce the latency and improve the throughput
- Research if it is possible to allow users to set multiple parent certificates, either by allowing array-typed attributes or by performing recursive lookups
- Research if it is possible to allow more extensive policy definitions, for example by providing clients with more check or operator types
- Research if it is possible to store the private keys of clients in **Hardware Security Modules (HSM)** to improve the security of the private keys

8. Definition of Terms

CA: Certificate Authority
IoT: Internet of Things
HSM: Hardware Security Module
IPFS: InterPlanetary File System [1]
MSP: Membership Service Provider

ORG#: Organization "#", which can transact on the channel
X.509: Standard defining the format of public key certificates
Chaincode: Deployed package of one or more smart contracts

9. References

[1] IPFS Community. "IPFS Documentation". [Online]. Available: https://docs.ipfs.io/. Accessed: May 12, 2022.
[2] Hyperledger Fabric Community. "Hyperledger Fabric Documentation". [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/. Accessed: May 12, 2022.
[3] L. Song, M. Li, Z. Zhu, P. Yuan, and Y. He, "Attribute-Based Access Control Using Smart Contracts for the Internet of Things", in *Procedia Computer Science*, 2020, pp. 231-242, doi: 10.1016/j.procs.2020.06.079.
[4] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT", in *IEEE Access*, 2019, pp. (99):1-1, doi: 10.1109/ACCESS.2019.2905846.
[5] X. Zhao, S. Wang, Y. Zhang, and Y. Wang, "Attribute-Based Access Control Scheme for Data Sharing on Hyperledger Fabric", in *Journal of Information Security and Applications*, 2022, pp. 103182, doi: 10.1016/j.jisa.2022.103182

[1] Available via https://github.com/daangordijn/Fabric-Access-Control

TU Delft