Controller of Certifying Authorities
Ministry of Electronics & Information Technology
Government of India

IEEE COMPUTER SOCIETY Bangalore Chapter

IAS IEEE INDUSTRY APPLICATIONS SOCIETY Linking Research to Practice Bangalore Chapter

5th INTERNATIONAL CONFERENCE ON

PUBLIC KEY INFRASTRUCTURE AND ITS APPLICATIONS (PKIA 2024)

SEPTEMBER 5-6th, 2024

Demonstration of Secure Key Management Solution with Use Case in Permissioned Blockchain

Geetha Sivanantham, Gowshalya Shri A M
SETS,Chennai

Dr T R Reshmi
Principal Investigator-UBF
Scientist,
SETS,Chennai

C-DAC

IEEE BANGALORE SECTION

Outline

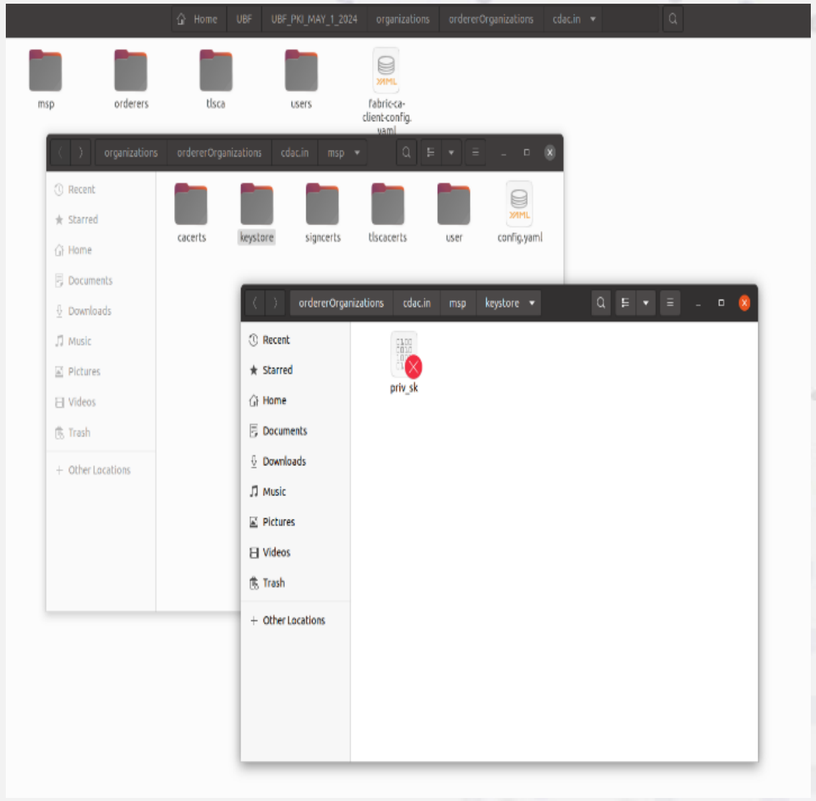
- Introduction
- Background
- Public-Key Cryptographic Standards (PKCS)
- Hyperledger Fabric (HLF) Operations
- Proof of Concept
- Recommendations
- Conclusion
- References

Introduction

- In a PKI infrastructure, private keys grant ownership and control over digital assets; compromising a private key can result in serious, potentially irreversible security breaches.
- Secure storage solutions like HSMs or encrypted vaults, combined with regular key rotation, strong encryption, and stringent access controls, are essential to prevent unauthorized access of private keys.
- The study investigates the security of PKI entities in extensive distributed environments, such as blockchain.
- It features a prototype that integrates an indigenous CA for Hyperledger Fabric (HLF) with Hardware Security Modules (HSMs) — the standard for key protection in cryptography.

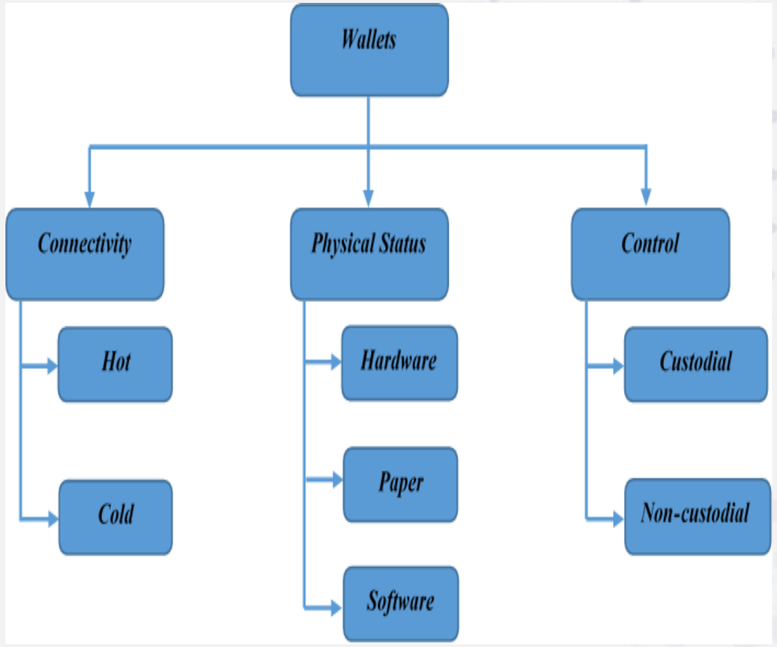
Background -HLF

- Hyperledger Fabric (HLF) is widely used in private (permissioned) blockchain frameworks for its flexibility and application in fields like supply chain management, e-healthcare, and IoT. In HLF, access to ledger services requires valid cryptographic credentials, with node identities verified through digital signatures.
- The Identity and Access Management (IAM) system in HLF is handled by the Fabric CA, which provide PKI based crypto credentials and key pairs to nodes.
- As a part of national blockchain service Fabric CA in HLF has been replaced with an external PKI-based CA. A secure key management system for HLF using an indigenous CA is our point of study.



Background -Wallets

- Crypto wallets are digital tools that allow us to store the private key and do transactions. It is a combination of public address and private key.
- The wallets are categorized based on three main criteria like connectivity, nature of physical existence and control over wallets.



Public-Key Cryptographic Standards (PKCS)

- The PKCS11 also known as cryptographic token interface standard (cryptoki) provides an API for devices that hold cryptographic information.
- The PKCS11 standard is implemented in default inside HSM by the vendors.
- The default fabric-CA component in HLF comes with an option to enable PKCS11 binaries by which the file system storage can be changed to HSM.
- In the proposed work, the HLF uses an external CA which has its own implementation of ECDSA. Hence to enable PKCS11 interface in this CA, PKCS 11 libraries have been modified in order to use the custom ECDSA implementation.

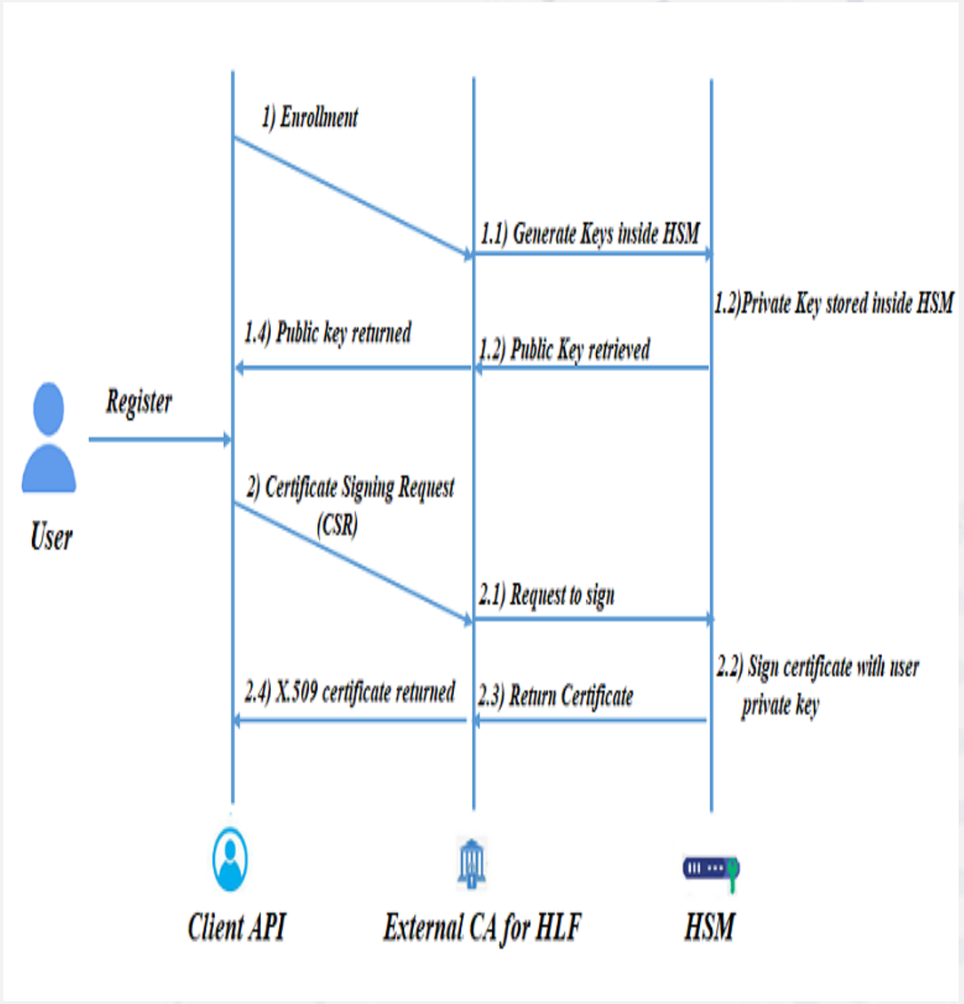
Step1:Initialize the cryptographic library	•C_Initialize(NULL_PTR);
Step 2: Get the number of available slots	•CK_ULONG slotCount; •C_GetSlotList(CK_TRUE, NULL_PTR, &slotCount);
Step 3: Get the list of available slots	•CK_SLOT_ID slots[slotCount]; •C_GetSlotList(CK_TRUE, slots, &slotCount);
Step 4: Log in to the token (user authentication)	•C_Login(session, CKU_USER, userPin, userPinLength);

Step 5: Define the key generation mechanism and template	•CK_MECHANISM mechanism = { CKM_CUSTOMEC_KEY_GEN, NULL_PTR, 0 };
	•CK_ATTRIBUTE keyTemplate[] = {
	{CKA_CLASS, &class, sizeof(class) },
	{ CKA_KEY_TYPE, &keyType, sizeof(keyType) },
	{ CKA_VALUE_LEN, &keyLength, sizeof(keyLength) },
	{ CKA_SIGN, &true, sizeof(true) },
	{ CKA_VERIFY, &true, sizeof(true) }
	};

Step 6: Generate the key	•CK_OBJECT_HANDLE key; •C_GenerateKey(session, &mechanism, keyTemplate, sizeof(keyTemplate), sizeof(CK_ATTRIBUTE), &key);
Step 7: Define the signing mechanism	•CK_MECHANISM signMechanism = { CKM_SHA256_CUSTOMEC_PKCS, NULL_PTR, 0 };
Step 8: Sign data	•C_SignInit(session, &signMechanism, privateKey); •C_Sign(session, data, dataLength, signature, &signatureLength);
Step 9: Verify signature	•C_VerifyInit(session, &signMechanism, publicKey); •C_Verify(session, data, dataLength, signature, signatureLength);
Step 10: Log out from the token	•C_Logout(session);
Step 11: Close the session	•C_CloseSession(session);

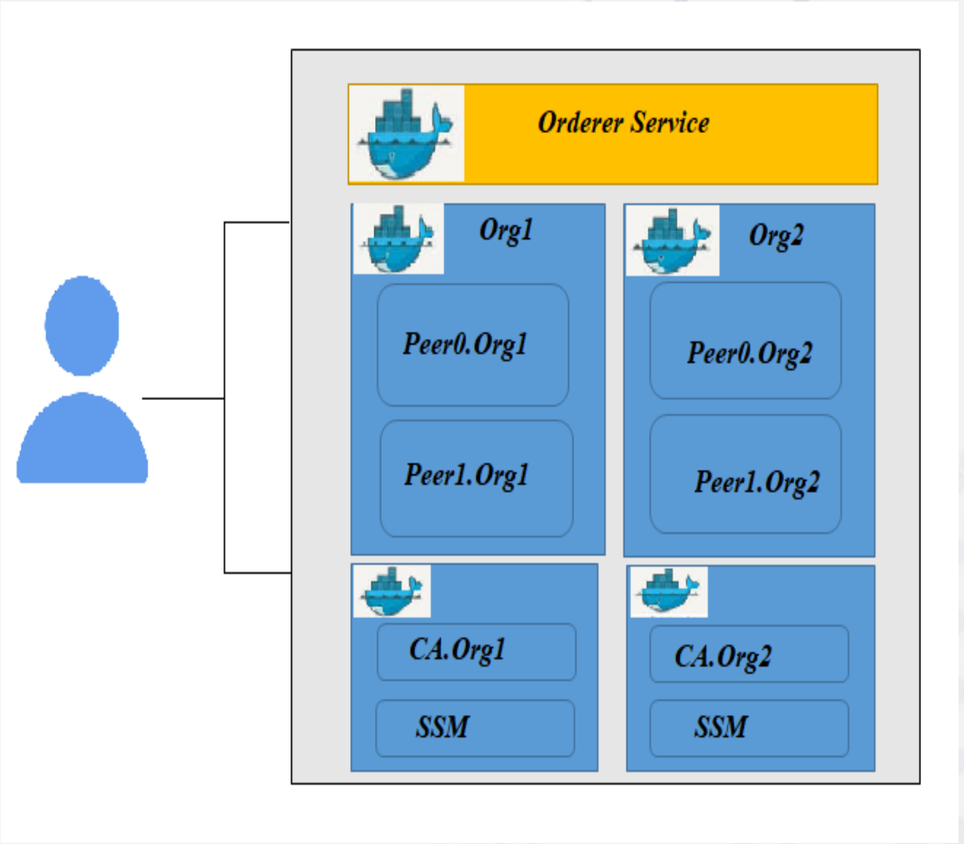
HLF- Operations

- The two major operations of HLF that includes the private key will be the enrollment process and the transaction proposal and signing.
- The enrollment process needs the CA for generation of credentials like key pairs and certificates. The private key is stored inside the HSM permanently while the public key is retrieved and returned to the client.
- In the same way as enrollment, client application generates a transaction proposal, which includes details about the transaction such as the chaincode(smart contract).The client application signs the transaction proposal with its private key.



Proof of Concept

- The default network configuration of HLF with two organizations, one ordering service node, two peers per organizations and one CA per organization is used.
- The softHSM is integrated with external CA and is run as separate docker container.
- The HLF will be loaded as another docker container.
- A client who want to enroll may be attached to any one of the organization.



Recommendations

- The Hardware Security Module (HSM) is actively involved in both the enrollment and signing processes. Performance may vary depending on the firmware or HSM version integrated.
- When integrating an external Certificate Authority (CA) with Hyperledger Fabric (HLF), differences in cryptographic standards may occur. We recommend modifying the PKCS#11 libraries to implement custom digital signature algorithm. This modification may introduce some overhead in the key generation process.

Conclusion and Future Work

- The feasibility of integrating HSM into permissioned blockchain framework such as HLF with custom developed CA has been attempted.
- The POC is presented with software emulation of HSM (softHSM) hosted as a docker service with custom CA. The generated private keys are stored inside SSM and the CA based crypto operations using private key are validated through HSM.
- The potential limitation in terms of scalability of slots in HSM has to be validated when it is to be deployed in blockchain networks.
- The complete performance benchmarking and optimization of crypto operations in custom CA will be accomplished in future.

References

- O. Boireau, “Securing the blockchain against hackers,” Network Security, vol. 2018, no. 1, pp. 8–11, Jan. 2018, doi: https://doi.org/10.1016/s1353-4858(18)30006-0.
- W. M. Shbair, E. Gavrilov, and R. State, “HSM-based Key Management Solution for Ethereum Blockchain,” 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), May 2021, doi: https://doi.org/10.1109/icbc51069.2021.9461136.
- S. Wang et al., “On Private Data Collection of Hyperledger Fabric,” Jul. 2021, doi: https://doi.org/10.1109/icdcs51616.2021.00083.
- E. Androulaki et al., “Hyperledger fabric: A Distributed Operating System for Permissioned Blockchains,” Proceedings of the Thirteenth EuroSys Conference on - EuroSys ’18, 2018, doi: https://doi.org/10.1145/3190508.3190538.
- G. Santhosh and T. R. Reshmi, “Enhancing PKI Security in Hyperledger Fabric with an Indigenous Certificate Authority,” Sep. 2023, doi: https://doi.org/10.1109/pkia58446.2023.10262412.
- G. Hileman and M. Rauchs, “2017 Global Cryptocurrency Benchmarking Study,” SSRN Electronic Journal , 2017, doi: https://doi.org/10.2139/ssrn.2965436.
- Riccardo Focardi and F. L. Luccio, “Secure Upgrade of Hardware Security Modules in Bank Networks,” Lecture notes in computer science, pp. 95–110, Jan. 2010, doi: https://doi.org/10.1007/978-3-642-16074-5_7.
- J. Ivarsson and A. Nilsson, A review of hardware security modules: Fall 2010, [online] Available: https://www.opendnssec.org/wp-content/uploads/2011/01/A-Review-of-Hardware-Security-Modules-Fall-2010.pdf.
- “OpenDNSSEC» SoftHSM.” https://www.opendnssec.org/softhsm/
- PKCS #11 Cryptographic Token Interface Base Specification Version 2.40, April 2015.
- C. Cachin, "Architecture of the hyperledger blockchain fabric", Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers, vol. 310, pp. 4, 2016.
- L. E. Hughes, Pro Active Directory Certificate Services. Apress, 2022.

THANK YOU