Public Key Infrastructure: DLT based Decentralized Public Key Infrastructure System

Section 1 Summary

Use Case Summary			
Use Case ID:	SEM-002	**Use Case Type:**	Horizontal
Use Case Title:	DLT based Decentralized Public Key Infrastructure System	**Is Use Case supporting SDGs**	*Yes*
		Domain:	Security Management
Status of Case	Proof of Concept	**Sub-Domain**	Public Key Infrastructure
Contact information of person submitting/ managing the use-case	*Xinpeng Wei* *Bingyang Liu*	*wexinpeng@huawei.com* *liubingyang@huawei.com*	
Proposing Organization	*Huawei*		
Short Description	PKI, Public Key Infrastructure, acts as the trust foundation in many scenarios, but the current hierarchical PKI system faces the problem of single point of failure. This document describes how to build a decentralized PKI system.		
Long description	A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store & revoke digital certificates and manage public-key encryption. The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email. Currently the PKI system is built in a hierarchical mode, one root CA exist at the top of the system and several intermediate CAs at lower level. The security of the whole system based on the security of root CA, if root CA is corrupted or misbehavior then the whole system fails. By using DLT, a decentralized PKI system can be built without highly centralized root CA, and avoid the single point of failure problem.		
SDG in Focus (when applicable)	Goal 9: Industry, Innovation and Infrastructure 9.3 Develop quality, reliable, sustainable and resilient infrastructure, including regional and transborder infrastructure, to support economic development and human well-being, with a focus on affordable and equitable access for all.		
Value Transfer:	token	**Number of Users:**	Tens of thousands
Types of Users:	ISP, OTT, web user, enterprise, bank, government…		
Stakeholders	certificate authority, anyone needs a certificate		

Data:	1. Token account 2. Digital certificate related information (e.g. Identity, application specific information, cryptographic-related information etc.) 3. Smart contract, including running code for PKI-related operations
Identification:	Both anonymous Identification and identifiable identification should be supported.
Predicted Outcomes:	A decentralized PKI system based on DLT.

Overview of the Business Problem or Opportunity

Currently the PKI system is built in a hierarchical mode, one root CA exist at the top of the system and several intermediate CAs at lower level. The security of the whole system based on the security of root CA, if root CA is corrupted or misbehavior then the whole system fails.

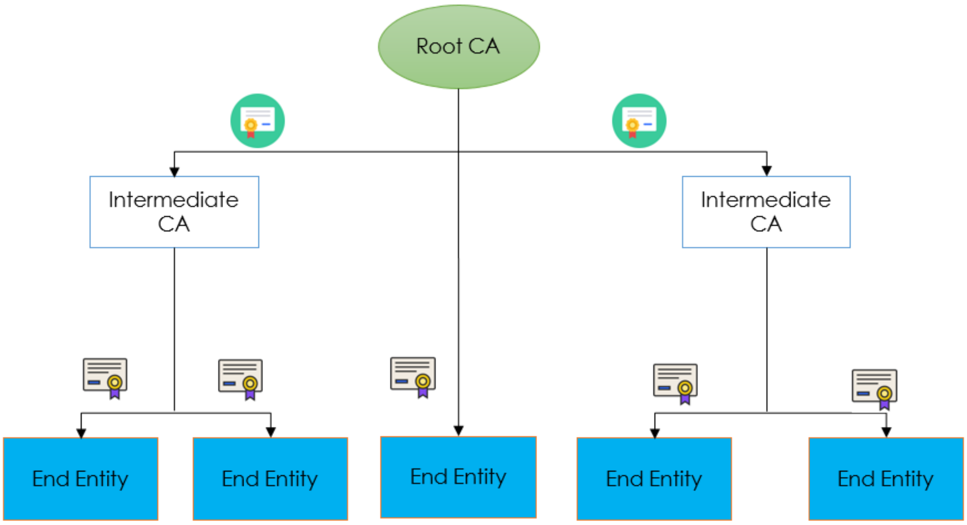


Figure 1: Centralized and Hierarchical Public Key Infrastructure

Why Distributed Ledger Technology?

The distributed and unaltered features of DLT make it easy to build a decentralized system on it, and especially its support of smart contract makes it possible to issue the digital certificate fully automated.

Section 2 Current process

Current Solutions

Certificate Transparency is a solution that can, to a certain extent, mitigate risk caused by mistakenly issued certificates or certificates that have been issued by a certificate authority (CA) that’s been compromised or gone rogue.

Certificate Transparency aims to remedy these certificate-based threats by making the issuance and existence of SSL certificates open to scrutiny by domain owners, CAs, and domain users. Specifically, Certificate Transparency has three main goals[1]:

- Make it impossible (or at least very difficult) for a CA to issue a SSL certificate for a domain without the certificate being visible to the owner of that domain.

- Provide an open auditing and monitoring system that lets any domain owner or CA determine whether certificates have been mistakenly or maliciously issued.

- Protect users (as much as possible) from being duped by certificates that were mistakenly or maliciously issued.

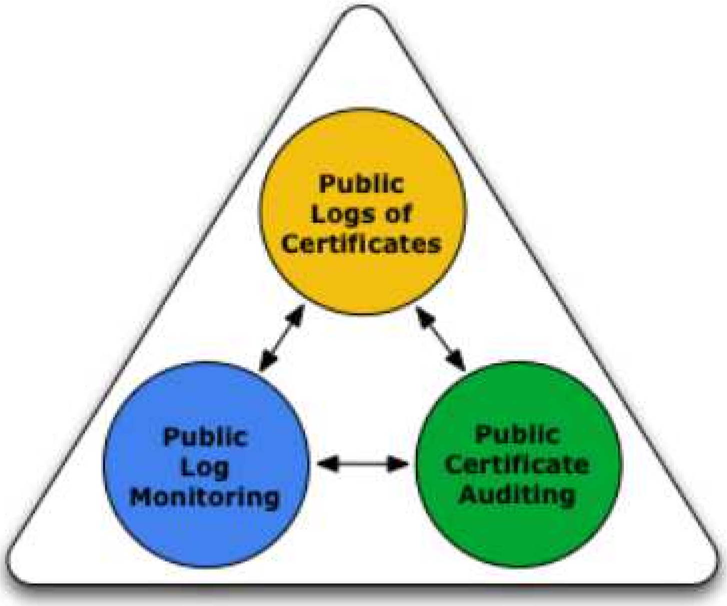
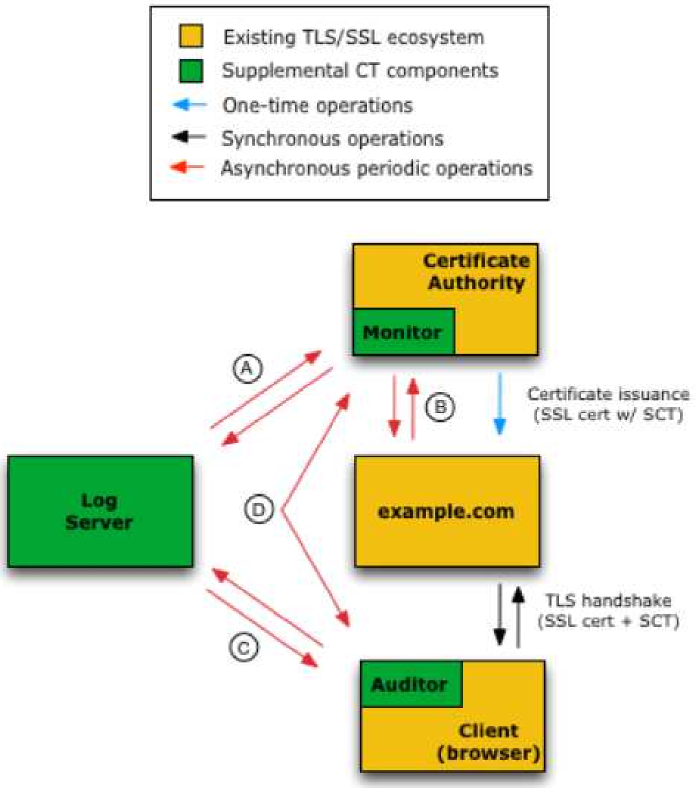


Figure 2: Basic Components of Certificate Transparency[1]

Existing Flow (as-is)		
Step	**User Actions**	**System Actions**
1.	Certificate authority submit certificate to Log Server.	N/A
2.	Log Server provides a response to certificate authority to acknowledge the submission.	N/A
3.	Monitors watch logs for suspicious certificates and verify that all logged certificates are visible.	N/A
4.	Certificate owners query Monitors to verify that nobody has logged illegitimate certificate for their domain.	N/A

Existing Flow (as-is)		
Step	**User Actions**	**System Actions**
5.	Auditors verify that logs are behaving properly; they can also verify that a particular certificate has been logged.	N/A
6.	Monitors and Auditors exchange information about logs to help detect forked or branched logs.	N/A

Process scheme (as-is)


🟨 Existing TLS/SSL ecosystem
🟩 Supplemental CT components
← One-time operations
← Synchronous operations
← Asynchronous periodic operations
Certificate Authority
Monitor
Log Server
example.com
Auditor
Client (browser)
A
B
C
D
Certificate issuance (SSL cert w/ SCT)
TLS handshake (SSL cert + SCT)

Ⓐ Monitors watch logs for suspicious certs and verify that all logged certs are visible.

Ⓑ Certificate owners query monitors to verify that nobody has logged illegitimate certs for their domain.

Ⓒ Auditors verify that logs are behaving properly; they can also verify that a particular cert has been logged.

Ⓓ Monitors and auditors exchange information about logs to help detect forked or branched logs.

Figure 3: Typical Configuration of Certificate Transparency System [1]

Data and information (as-is)		
Data	**Type**	**Description**
1	Certificate	Certificates are stored in certificate Log Server.

Data and information (as-is)		
Data	**Type**	**Description**

Participants and their roles (as-is)		
Actor	**Type/Role**	**Description**
1	Log Server	Log Server is simple network services that maintain cryptographically assured, publicly auditable, append-only records of certificates
2	Monitors	Monitors are publicly run servers that periodically contact all of the log servers and watch for suspicious certificates.
3	Auditors	Auditors are lightweight software components that typically perform two functions.

Other Notes
N/A

Section 3 Expected process

Expected Flow (to-be)		
Step	**User Actions**	**System Actions**
1.	End Entity apply for certificate from distributed ledger by sending transactions to specific smart contract.	The distributed ledger checks if the application from End Entity is acceptable, if true records the application request.
2.	Client verifies certificate base on the ledger.	The distribute ledger provides certificate-related information to the Client.

Process scheme (to-be)

There are three kinds of certificate:

DV certificate: Domain validated, the most common type of SSL certificate. They are verified using only the domain name.

OV Certificate: Organization validated, requiring more validation than DV certificates, but provide more trust. The organization's name is also listed in the certificate, giving added trust that both the website and the company are reputable. OVs are usually used by corporations, governments and other entities that want to provide an extra layer of confidence to their visitors.

EV Certificate: Extended validation, providing the maximum amount of trust to visitors, and also require the most effort by the CA to validate. As in the OV, the EV lists the company name in the certificate itself, However, a fully validated EV certificate will also show the name of the company or organization in the address bar itself, and the address bar is displayed in green.

The application of OV Certificate and EV Certificate needs endorsement from specific Endorser, but the application of DV certificate doesn't need endorsement. In order to cope with single point of failure problem for Endorser, the endorse procedure could be required endorsement from multiple Endorsers. The EV Certificate could always be used for domain validation purpose even in case the endorsement is corrupted.

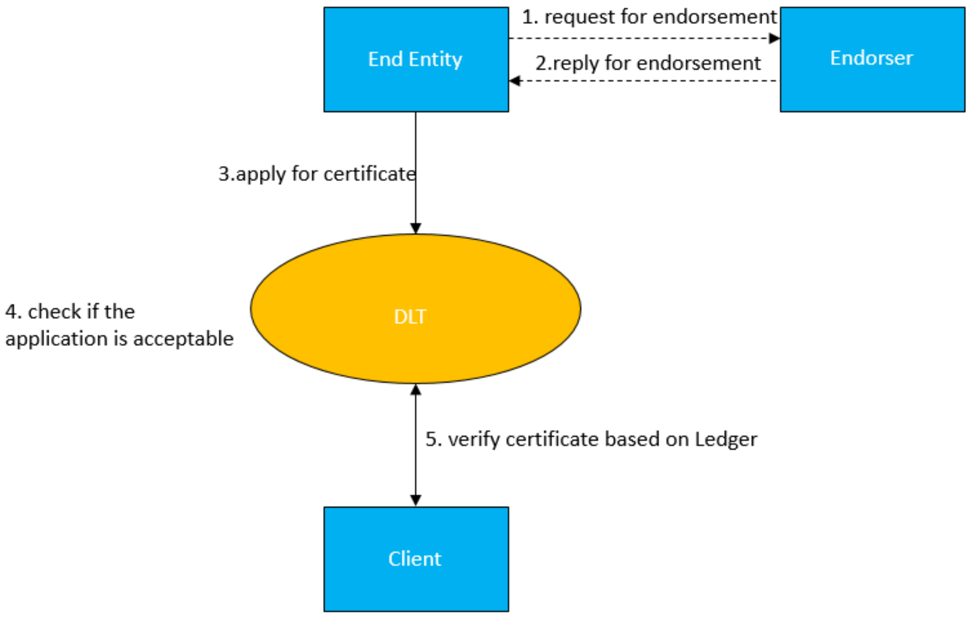

End Entity
1. request for endorsement
2.reply for endorsement
Endorser
3.apply for certificate
4. check if the application is acceptable
DLT
5. verify certificate based on Ledger
Client

Figure 4: Procedure of DLT-based Public Key Infrastructure

Participants and their roles		
Actor	**Type/Role**	**Description**
1	End Entity	The entity that apply certificate from decentralized public key infrastructure.
2	Endorser	Providing endorsement for End Entity. The Endorser could be implemented as a smart contract in the ledger.

Participants and their roles		
Actor	**Type/Role**	**Description**
3	Client	The entity that verifies certificate in specific application scenario, e.g. web browser.

Data and information		
Data	**Type**	**Description**
1	Certificate-related information	Digital certificate related information (e.g. Identity, application specific information, cryptographic-related information etc.)
2	Certificate application transaction	End Entities use transactions to interact with the ledger.
3	Smart contract	Including running code for PKI-related operations.
4	Token account	Each End Entity has a token account in the ledger.

Security and privacy
1. The security of distributed ledger itself is very critical.

Main Success Scenario
1. All information exchange and payments occur in Distributed Ledger in automatic mode. 2. Payment and service are exchanged without human intervention.

Conditions (pre- or post-)
1. The token must be created in some way. 2. All parties are connected to DLT system.

Performance needs
1. Transactions processing near real time; 2. 24/7/365 availability; 3. Volume of transactions > 1000 TPS.

Legal considerations
N/A

Risks
1. DLT-related security risk.

Special Requirements

N/A

External References and Miscellaneous

[1] http://www.certificate-transparency.org/what-is-ct

Other Notes

N/A

____

Appendix 1

Domains and subdomains for use cases categorization

Vertical:

1. Finance
    a. Financial management & accounting
    b. International & interbank payments
    c. Clearing and settlement
    d. Reduction of Fraud
    e. Financial messaging
    f. Asset lifecycles and history
    g. Trade finance
    h. Regulatory compliance & audit
    i. AML/KYC
    j. Insurance
    k. Peer-to-peer transactions
2. Healthcare
    a. Pharma
    b. Biotechnology
    c. Medicine
3. Industries
    a. Manufacturing
    b. Energy
    c. Chemical
    d. Retail
    e. Real estate
    f. IT and telco
    g. Supply chain management
    h. Transportation
    i. Agriculture
4. Government and public sector
    a. Taxes
    b. Government and non-profit transparency
    c. Legislation, compliance & regulatory oversight
    d. Voting
    e. Taxation and customs
    f. Intellectual property management
    g. Land Registries

Horizontal:

1. Identity Management
2. Security Management
    a. Public Key Infrastructure

3. Internet of Things
4. Data processing, storage and management
    a. Data Validation (includes provenance)