Privacy based decentralized Public Key Infrastructure (PKI) implementation using Smart contract in Blockchain

Sivakumar P [1] and Dr. Kunwar Singh [2]

[1, 2] National Institute of Technology, Trichy, Tamil Nadu 620015
shibv3@gmail.com

Abstract. Public Key Infrastructure (PKI) establishes the relation between user and a respective public key. Existing PKI implementation are majorly centralized in nature. Centralized PKI designs are not optimal and contain security flaws. A decentralized PKI would solve the problem of trusting the authorities in the key technologies behind the Internet. Inherent characteristics of Blockchain and Smart contract could resolve various problems exist with the conventional PKI implementations. Decentralized PKI is designed to build as public ledgers linking identity with public key following Web of Trust model. In decentralized PKI implementation privacy of the users are usually compromised, results in threat to its users. In this paper we discuss the implementation of decentralized PKI with privacy using blockchain and smart contract.

Keywords: PKI, Blockchain, Smart Contract, Privacy, Pretty Good Privacy, Web of Trust, PGP, Ethereum.

1 Introduction

The conventional approach to PKI is centralized in nature and it is implemented through certificate authorities (CAs) and web-of-trust (WoT) model using Pretty Good Privacy (PGP). CAs are trusted entities, who issue a signed certificate (X.509) to users. The public key is required to establish secure transaction between the communicating systems in the internet, on successful authentication of public key certificate of particular entity. Users in the WoT network establish trust in others by verifying that the users are trusted by at least one already trusted entity that certificate is signed by, entity in which the verifier has previously established trust. Due to centralized control conventional PKIs are single points of failure. Centralized PKI system lacks transparency in the issuance of certificates and its relation with user. WoT systems have high barrier to entry due to its difficulty to establish trust relations between all participating parties. We propose a decentralized implementation of PKI in blockchain using smart contracts in ethereum which can easily integrate with the Distributed Apps (DApps) designed in ethereum and have PKI requirement.

2 Background

2.1 Conventional PKI

The PKI certificate provides a record and authentication of the link between a public key and its owner. A PKI system performs registration of users, issuance of certificates, its revocation, storage and maintenance of certificates. A public key certificate in X.509 format contains certain set of attributes to establish the connection between the public key and the entity. The updated certificates of CAs are bundled with all operating system and the browsers existing in the internet domain. Due to centralized control the updated versions of certificates may not be updated in a timely manner and results in high security risks. Dutch based DigiNotar CA was hacked in 2011 and attacker issued rogue certificates for *.google.com, giving the attacker the ability to impersonate Google to any browser that trusted the certificate. The attack on DigiNotar extent quickly over internet due to inherent security flaws of present PKI system and lack of transparency in its operations.

2.2 Decentralized PKI (Web of Trust Model)

An alternative to the centralized chain of CA model of PKI is the Web of Trust. There are no certificate authorities in WoT. Instead any user of the system can sign each other's public key certificate and keys are by design intended to have multiple signatures. If one signer is compromised and the signer's key is revoked, the impact on the trust network is limited. PGP is a WoT model works on trustworthy users in which the trusted users sign each other and maintains private, public key rings.

2.3 PKI using Blockchain

Blockchain is a public distributed permanent ledger to which transaction events are posted and verified by peers in same network before being confirmed in an incentivized system in which members must compete to complete some proof-of-work cryptographic challenge. Bitcoin [1] transactions are mined and build as blocks on existing blockchain in Merkle tree structure. Decentralised nature of the blockchain is already been explored to design PKI to work in peer to peer setup. Certcoin [2] designed on Namecoin is a decentralized PKI which performs almost all functionalities of conventional PKI. Certcoin is immune to single point failures and it maintains identity retention which prevents replication of public key certificates for same users. It also averts generation of rogue certificates due to transparency in its operations. However, Certcoin could not secure privacy of the users involved in PKI transactions. Certcoin has limited role to cater PKI needs of DApps compared to our model.

2.4 Smart Contracts in Ethereum

Ethereum [3] introduce smart contracts [4] and a way to perform actions by the rules defined in the contract. Smart contracts in Ethereum are written in a low-level stack-based bytecode language executed on top of Ethereum and integrated as Ethereum Virtual Machine (EVM) code. Smart contracts can be written in Solidity or Serpent and compiled to deploy into Ethereum with minimal transaction cost paid in ether. It is a set of self-executing code and invoked by trigger of events inside the functions defined in a contract. Contracts are assigned with an address, the sender and receiver nodes use this address to communicate between wallet and other contracts. Contracts maintain its state, data can be stored by invoking the events and functions when certain set of rules defined are met and all parties agree it. Triggered contractual events are treated as transactions and mined for its inclusion in blockchain.

3 Proposal : Decentralized PKI with privacy in Blockchain

3.1 High-level description and diagram

Ethereum is a programmable blockchain. PKI operations are implemented as function in smart contract in ethereum. In this system the entities are represented using public key and ethereum address. Each entity can have multiple attributes to identify the owner of entity. Each transaction is represented using public key, then respective entity id and the PKI action. Smart contract is used to define the events and functions pertaining to various operations of PKI. Smart contract can act as a plug and play module inside the blockchain, which designed to control set of conditions to invoke specific PKI operations. The PKI functions and events are written in solidity and deployed in EVM which enables easy management of PKI operations. Following set of PKI operations are made available in the smart contract designed for decentralized PKI:-

Registration of Entity. Users are added to this PKI system by invoking the registration event from smart contract. Entity can be characterized by set of attributes such as Ethereum Address, Public Key, Attribute Id, data and data hash. This event collects the details pertaining to the entity and passed to smart contract and it further forward it as transaction to ethereum. The pending transactions are mined and blocks for new entity are added to blockchain after successful mining. Each entity is mapped with an auto generated id issued by smart contract.

Signing of Attributes. Entities can be identified using the attributes comprised with registration event. Each attribute of the entity can be signed by the system through underlying contract and issues as transaction. The signed entities can be made available as trusted by other entities or users. Signing and trusting of attributes by other entities facilitate WoT model in our decentralized PKI. Provision to include the expiry date for the signature is enforced through smart contract, which sets a life period for valid signatures.

Retrieval of Attributes. Attributes of the entities can be searched by filtering the blockchain using respective IDs of events raised by the smart contract.

Revoke Signature. It is one of the major functions required by any PKI to revoke the signatures on attributes or entities signed by other entities. Revocation is required when a user lost his key or it is compromised. Smart contract invoke the revocation event and revoke the signature on specified entity and post it as a transaction.

PGP for WoT. OpenPGP can implement in local systems as GNU Privacy Guard (GPG). In our decentralized PKI model, PGP public key can bind to ethereum address which enables the users to trust the PGP key and trust entity registered with blockchain.

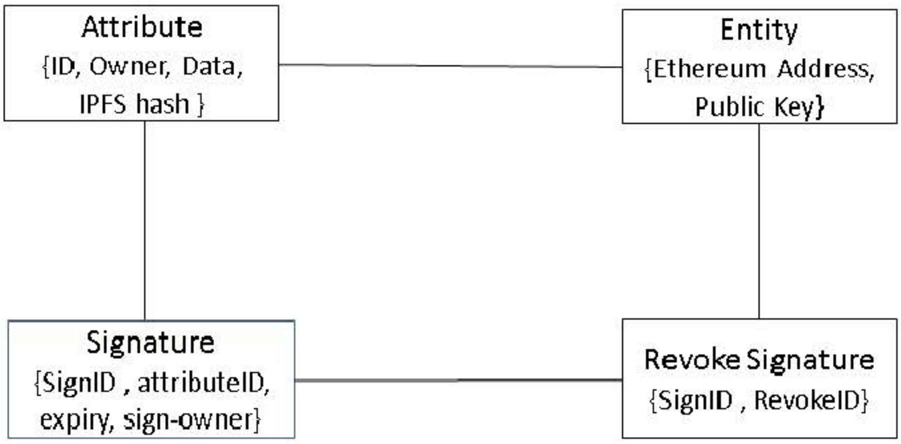
Attribute
{ID, Owner, Data, IPFS hash }
Entity
{Ethereum Address, Public Key}
Signature
{SignID , attributeID, expiry, sign-owner}
Revoke Signature
{SignID , RevokeID}

Fig. 1. Decentralized PKI smart contract functions with attributes

3.2 Privacy in Proposed PKI

Privacy in proposed PKI is enforced through smart contract obfuscation and public key address translations. Address translation of public key results in generation of different public key for each transaction from a particular ethereum node and it results in anonymity and enforces privacy. Entity data, balance, attributes are stored in blockchain can be encrypted and decrypted using private and public key of ethereum node using smart contract events and it appears, like execution happens inside a black box which prevents attackers to extract any data.

4 Implementation

Proposed system with basic PKI functions has been implemented in Ethereum EVM using smart contract by organizing the functions and events in a structured way. The privacy feature of decentralized PKI is designed to implement as below.

4.1 Key management in Ethereum

Ethereum has a private key and public key. Private key is randomly generated of size 32 bytes [5] using ethereum standard secp256k1 curve . Public key is of 64 bytes is derived from private key using Elliptic Curve Digital Signature Algorithm (ECDSA). Ethereum address of the node is generated using Keccak256 hash of the public key and the last 20 bytes of that resulted hash is the address of the node.

4.2 Public Key Translation

Privacy of the PKI can be implemented using a key translation method through which a set of public keys are generated from same private key and ensures each PKI transaction has a new public key related to private key of the ethereum node. The key translation process brings anonymity to blockchain nodes and it is difficult to track the transactions meant for any address.

Consider the Receiver Node A has a public key '**A'** and private key '**a**' such that

$$A = a.G \text{ , where G is the generator of an elliptic curve} \tag{1}$$

Node B is a sender and sends a transaction to Node A with a public key valid only for this single transaction to ensure privacy. Node B generates an ephemeral key pair with '**B**' an elliptic curve point and '**b**' is a 256 bit integer. Node B sends 'B' to Node A and now both can calculate shared secret

$$B = b.G \tag{2}$$

$$b.A = B.a$$

$$b.a.G = b.G.a \text{ , from (1) and (2)}$$

Now Node B can generate public key using this shared secret

$$\mathrm{Keccak256}[A + H(b.A).G]$$

Node A can spend the money using private key $[a + H(a.B)]$

4.3 Obfuscated Smart Contract

Data privacy in decentralized PKI in blockchain can implement using obfuscated smart contract. Methods and events in smart contract are obfuscated and data writing to storage is encrypted using public key. Decryption of secure data is done using private key. Smart contract methods and events internally encrypt and decrypt data using the private/public keys. The node addresses not in access control list are not permitted to access the encrypted data from obfuscated smart contract and data privacy is ensured.

5 Transaction Costs

Smart contracts for decentralized PKI are used to send multiple payments in a single transaction which optimizes cost. Gas cost changes with the size of data to process in each transaction. It is not cost effective to store large amount of data with blockchain, it increase the gas cost on transactions. Inter Planetary File System (IPFS) [6] can be used to store large amount of data pertaining to any entity in our proposed system. IPFS returns a link to data and hash of the that IPFS link is stored in blockchain hence the gas cost will remain constant for IPFS links.

6 Conclusion and Future work

Decentralized PKI can be implemented using ethereum blockchain and smart contract. Transparency in decentralized PKI operations are enforced through proposed system. Proposed system is not prone to single point failures. Large number of decentralized apps (DApps) is designed on Ethereum platform; our proposed PKI system can be easily integrated along with these applications to cater their PKI requirements. As further development, efficiency of the system needs to be improved. Proposed system has to be scaled up for peer to peer network with heavy. Privacy in proposed PKI can also be implemented using various other key exchange schemes viz Ring Signatures, Zero Knowledge proofs..etc.

7 References

1. Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system, 2008." (2012): 1-9.
2. Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. Certcoin: A namecoin based decentralized authentication system,. 2014.
3. G. Wood. Ethereum: A secure decentralised generalised transaction ledger. http://gavwood.com/paper.pdf.
4. A Next-Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-White-Paper
5. Andreas M Antonopoulos. Mastering Bitcoin: Programming the open blockchain. O'Reilly Media, 2017.
6. IPFS reference. https://ipfs.io/docs/commands/