Towards a decentralized identity management solution based on blockchain — proof of concept

Fabien Charmet — Télécom SudParis, Institut Mines-Télécom, CNRS Samovar UMR 5157

Maxime Montoya — Univ. Grenoble Alpes, CEA, LETI, DACLE

Mathieu Valois — Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC

Wojciech Wideł — Univ Rennes, INSA Rennes, CNRS, IRISA

26 October 2018

Outline

Background on public key infrastructure (PKI) and blockchains

How blockchains could enhance PKI

Existing approaches

Multichain-based certificate management

Conclusion

Background on public key infrastructure (PKI) and blockchains

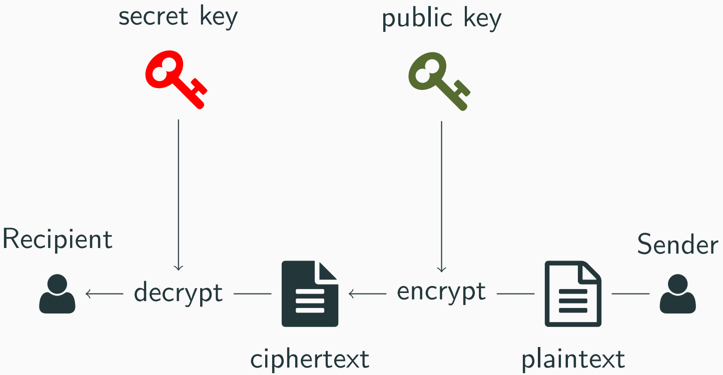
How blockchains could enhance PKI

Existing approaches

Multichain-based certificate management

Conclusion

Public-key encryption

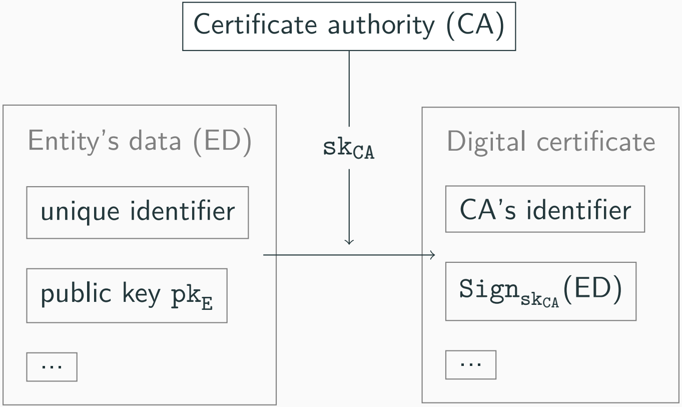


Public key infrastructure

Public key infrastructure (PKI)

- A set of roles and procedures ensuring secure distribution of public keys.
- Based on **digital certificates**.

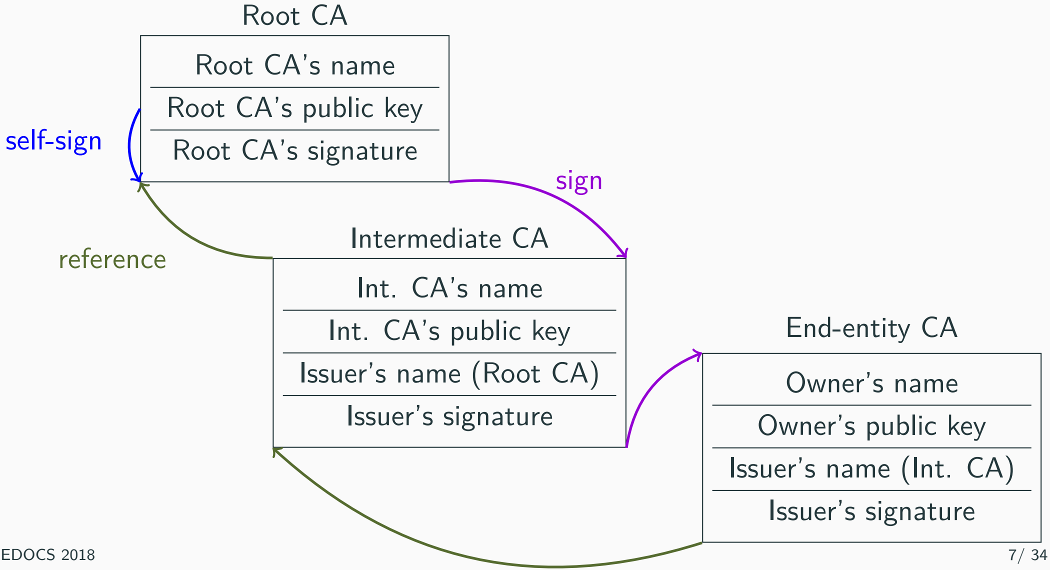
Digital certificate



Certificate authority (CA)
$\mathtt{sk}_{\mathtt{CA}}$
Entity's data (ED)
unique identifier
public key $\mathtt{pk}_{\mathtt{E}}$
$\cdots$
Digital certificate
CA's identifier
$\mathtt{Sign}_{\mathtt{sk}_{\mathtt{CA}}}(\mathrm{ED})$
$\cdots$

CA certifies: $\mathtt{pk}_{\mathtt{E}}$ is indeed the public key of the entity E.

Chain of trust



Public key infrastructure

Revocation of certificates

- Compromised certificates are **revoked** by the issuing CA.
- CA adds revoked certificates to its **certificate revocation list (CRL)**.
- CA publishes updated CRL ~every 24 hours.

Public key infrastructure

Problem: single point of failure

- Corrupt CA $=$ illegitimate certificates.
- Single CA corrupt $=$ PKI's failure.

Public key infrastructure

Problem: single point of failure

- Corrupt CA = illegitimate certificates.
- Single CA corrupt = PKI's failure.

Possible countermeasure

- Store certificates and CRL in an external ledger.
- What kind of ledger?

Blockchain

Definition

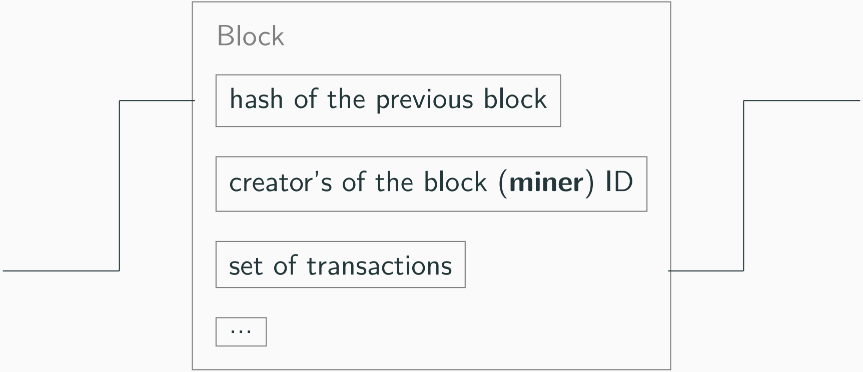
- A public, transparent, append-only ledger.
- Created by members of a peer-to-peer network.
- Immutable and unforgeable records (**blocks**).

Blockchain

Structure

- **Transaction**: atomic event allowed by the blockchain protocol (‘Alice sends Bob 0.1 BTC’, ‘CA issues a certificate’).
- Transactions are **validated** and **broadcasted** throughout the network.
- Validated transactions are stored in **blocks**.
- Blocks are linked together, forming a **chain**.
- **Consensus process**.

Blockchain structure



Block

hash of the previous block

creator's of the block (**miner**) ID

set of transactions

...

Scenario

Current scenario

user:

1. connects to a website
2. browser verifies identity of webserver using PKI

Future scenario

user:

1. connects to a website
2. browser verifies identity of webserver using PKI
3. browser verifies identity if webserver using Blockchain

Background on public key infrastructure (PKI) and blockchains

How blockchains could enhance PKI

Existing approaches

Multichain-based certificate management

Conclusion

Public key infrastructure

Problems

- No way to know if CA is corrupted.
- CA producing certificates for domains they don't own (Iran with Google).
- Some web browsers don't check for certification revocation.

Solution: blockchain

- Another channel for verifying certificate's validity.
- **Transparency** and **traceability**.
- Secure distributed log that cannot be altered.
- The whole chain of trust is stored.
- Revocation lists are stored.

Applications

Web browsing

- Privacy and confidentiality issue: are visited websites what they pretend to be?
- Millions of certificates, with variable lifetime

Connected cars

- Safety issue: connected or even autonomous cars might need to check that the surrounding cars are legitimate
- Thousands of certificates, with a one-week lifetime

Background on public key infrastructure (PKI) and blockchains

How blockchains could enhance PKI

Existing approaches

Multichain-based certificate management

Conclusion

Blockchain and smart contracts

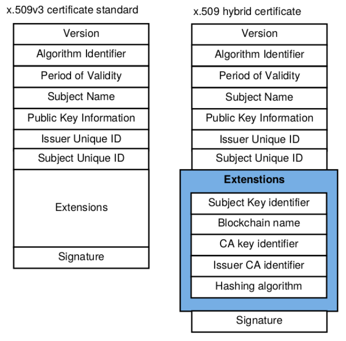
Smart contracts in Ethereum

- Ethereum is a blockchain that supports **smart contracts**
- Smart contracts are special entities, written in the blockchain
  - Execution conditions predefined and agreed on
  - Execute when these conditions are met
  - Each transaction with a smart contract is a transaction in the blockchain

Existing approaches[1]

Ethereum smart contracts

- Each certification authority has **smart contracts** that store a list of issued certificates and a revocation list
- Specific format for certificates: **hybrid certificates**



x.509v3 certificate standard

Version
Algorithm Identifier
Period of Validity
Subject Name
Public Key Information
Issuer Unique ID
Subject Unique ID
Extensions
Signature

x.509 hybrid certificate

Version
Algorithm Identifier
Period of Validity
Subject Name
Public Key Information
Issuer Unique ID
Subject Unique ID
Extenstions
Subject Key identifier
Blockchain name
CA key identifier
Issuer CA identifier
Hashing algorithm
Signature

[1] A. Yakubov et al., "A blockchain-based PKI management framework," NOMS 2018 - IEEE/IFIP Network Operations and Management Symposium, Taipei, 2018, pp. 1-6.

Existing approaches

Data fields in Bitcoin-based blockchains

- Special **OP_RETURN** field can contain arbitrary data
    - Many applications, such as Intellectual Property
- Bitcoins: maximum size of 80 bytes
- Several blockchains could be used, such as Bitcoin or Namecoin

Background on public key infrastructure (PKI) and blockchains

How blockchains could enhance PKI

Existing approaches

Multichain-based certificate management

Conclusion

Multichain-based certificate management

Multichain

- fork of the Bitcoin source code
- hugely simplifies private Blockchains creation and management
- lot of settings available
- node permission control
- arbitrary-sized data field in transactions
- very well documented

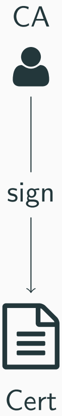
Comparison

	Smart contracts	OP_RETURN	Multichain
Usability - customization	-	-	+
Cost	-	-	+
Compatibility with existing PKIs	-	+	+
Permissions	-	-	+
Size of certificates	+	-	+
Scalability	+	-	-

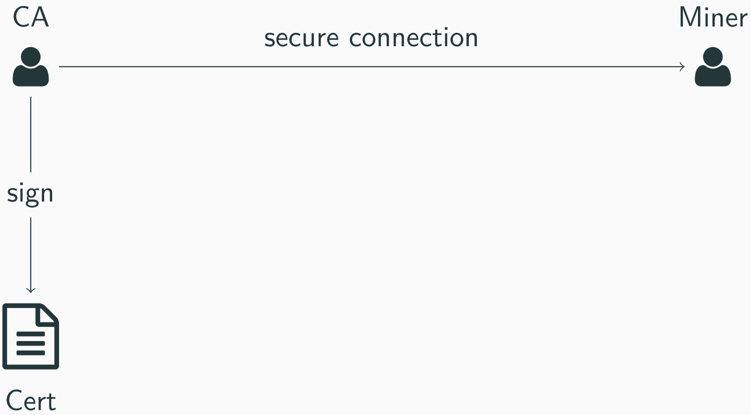
Design

CA

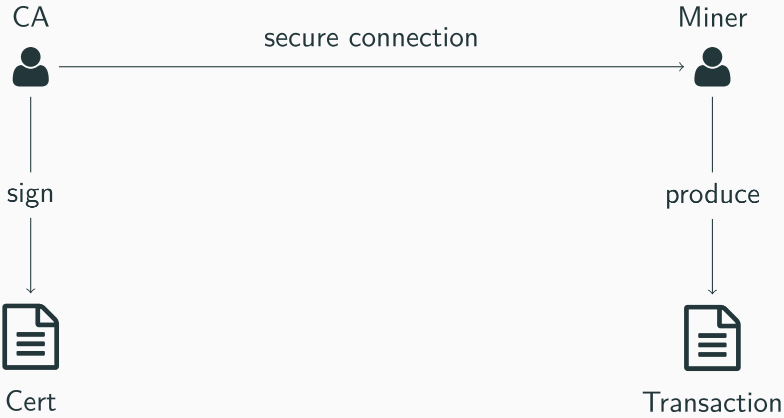
Design



Design



Design



CA

secure connection

Miner

sign

produce

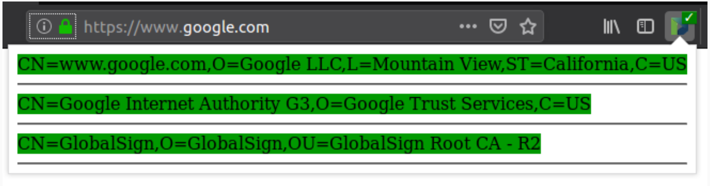
Cert

Transaction

Scenario

1. final user visits a website with web browser
2. classical identity verification is used (PKI)
3. browser plug-in installed on the user browser
4. local daemon is running, waiting for queries
5. plugin-in retrieves certificates, asking to daemon if such a certificate is valid
6. displays whether certificates should be trusted or not

Demo

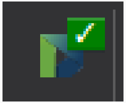


https://www.google.com

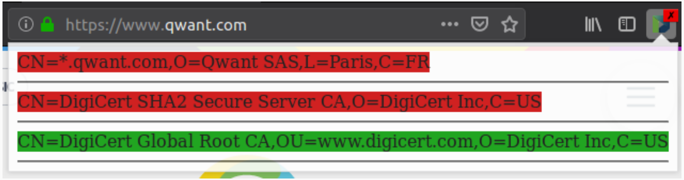
CN=www.google.com,O=Google LLC,L=Mountain View,ST=California,C=US

CN=Google Internet Authority G3,O=Google Trust Services,C=US

CN=GlobalSign,O=GlobalSign,OU=GlobalSign Root CA - R2



Demo

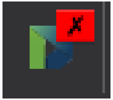


https://www.qwant.com

CN=*.qwant.com,O=Qwant SAS,L=Paris,C=FR

CN=DigiCert SHA2 Secure Server CA,O=DigiCert Inc,C=US

CN=DigiCert Global Root CA,OU=www.digicert.com,O=DigiCert Inc,C=US



Cost and scalability

Use case: Let's Encrypt

- Certification authority
- Delivered 100M certificates over 20 months
  - More than 160K per day

Application to multichain-based certificates management

- Around 280 Go of memory for 100M certificates
  - Bitcoin: around 90 Go over 20 months
- The whole blockchain has to be read when searching for a specific certificate
  - Ideally, only the delivery day would have to be checked in the blockchain

Background on public key infrastructure (PKI) and blockchains

How blockchains could enhance PKI

Existing approaches

Multichain-based certificate management

Conclusion

Conclusion

Problem
How to detect a malicious CA?

Solution
Add an extra channel to verify certificates using the blockchain

Future Work

- Implement PKI functions using the blockchain
- Explore the use of smart contracts
- Elaborate a business model

Feedback

- Interesting topic with no previous knowledge
- Working PoC with exciting perspectives
- Pleasant teamwork and environment

Organization

Questions

Thank you for your attention. Questions?