ClaimChain

A Decentralized Public Key Infrastructure based on Cross-Referenced Hash chains

Marios Isaakidis, George Danezis @ UCL
Bogdan Kulynych, Carmela Troncoso @ IMDEA

December 28, 2016

`bogdankulynych.me/33c3`

Bogdan Kulynych
PhD student, IMDEA Software Institute, Madrid

Twitter: @hiddenmarkov
Email: bogdan.kulynych at imdea.org

NEXTLEAP project
nextleap.eu

Goals

ClaimChain basics

Cross-Referencing

Supporting infrastructure

Privacy and Security

Work in progress

Goals

Modern Key Management needs

- Frequent key updates
- Support for ephemeral keys, OTR, Bitcoin wallets…
- Multi-device support
- Better handling of key compromisation/loss
- Interoperability with legacy agents
- Better Web of Trust
  - Privacy of the social graph
  - Also vouching for the “state” of a PGP key

Modern Key Management needs

- Frequent key updates
- Support for ephemeral keys, OTR, Bitcoin wallets…
- Multi-device support
- Better handling of key compromisation/loss
- Interoperability with legacy agents
- Better Web of Trust
  - Privacy of the social graph
  - Also vouching for the “state” of a PGP key

Modern Key Management needs

- Frequent key updates
- Support for ephemeral keys, OTR, Bitcoin wallets…
- Multi-device support
- Better handling of key compromisation/loss
- Interoperability with legacy agents
- Better Web of Trust
  - Privacy of the social graph
  - Also vouching for the “state” of a PGP key

Modern Key Management needs

- Frequent key updates
- Support for ephemeral keys, OTR, Bitcoin wallets…
- Multi-device support
- Better handling of key compromisation/loss
- Interoperability with legacy agents
- Better Web of Trust
  - Privacy of the social graph
  - Also vouching for the “state” of a PGP key

Modern Key Management needs

- Frequent key updates
- Support for ephemeral keys, OTR, Bitcoin wallets…
- Multi-device support
- Better handling of key compromisation/loss
- Interoperability with legacy agents
- Better Web of Trust
  - Privacy of the social graph
  - Also vouching for the “state” of a PGP key

Modern Key Management needs

- Frequent key updates
- Support for ephemeral keys, OTR, Bitcoin wallets…
- Multi-device support
- Better handling of key compromisation/loss
- Interoperability with legacy agents
- Better Web of Trust
  - Privacy of the social graph
  - Also vouching for the “state” of a PGP key

Modern Key Management needs

- Frequent key updates
- Support for ephemeral keys, OTR, Bitcoin wallets…
- Multi-device support
- Better handling of key compromisation/loss
- Interoperability with legacy agents
- Better Web of Trust
  - Privacy of the social graph
  - Also vouching for the “state” of a PGP key

ClaimChain basics

Claim

- Key material
  - Signature key
  - Recovery key
- Generic things
  - Encryption keys
  - Signal prekeys
  - Identity in social nets / emails
- Revocations
- Cross-references (will get back to this)

Clients maintain per-device append-only logs of claims.

Claim

- Key material
  - Signature key
  - Recovery key
- Generic things
  - Encryption keys
  - Signal prekeys
  - Identity in social nets / emails
- Revocations
- Cross-references (will get back to this)

Clients maintain per-device append-only logs of claims.

Claim

- Key material
  - Signature key
  - Recovery key
- Generic things
  - Encryption keys
  - Signal prekeys
  - Identity in social nets / emails
- Revocations
- Cross-references (will get back to this)

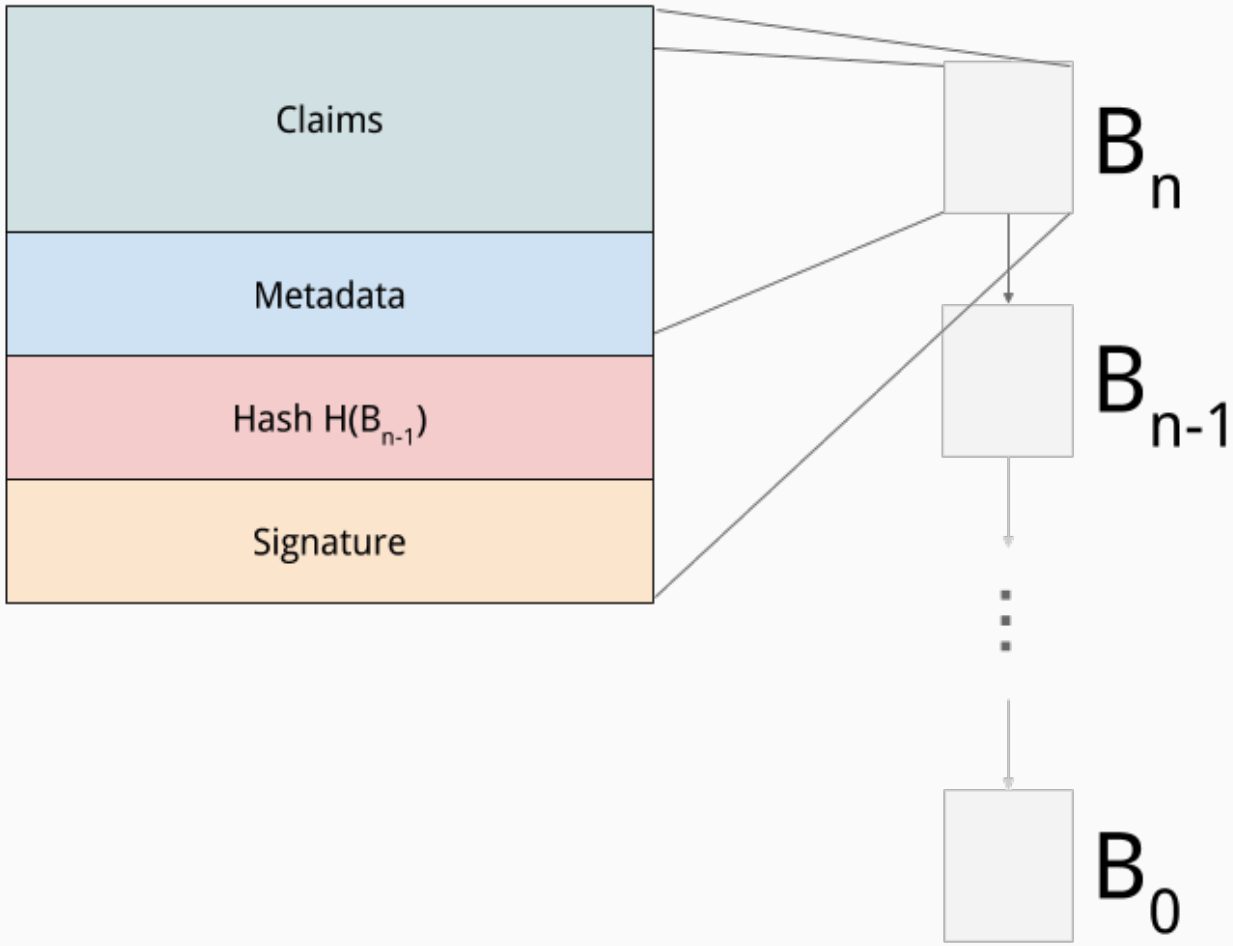
Clients maintain per-device append-only logs of claims.

Claim

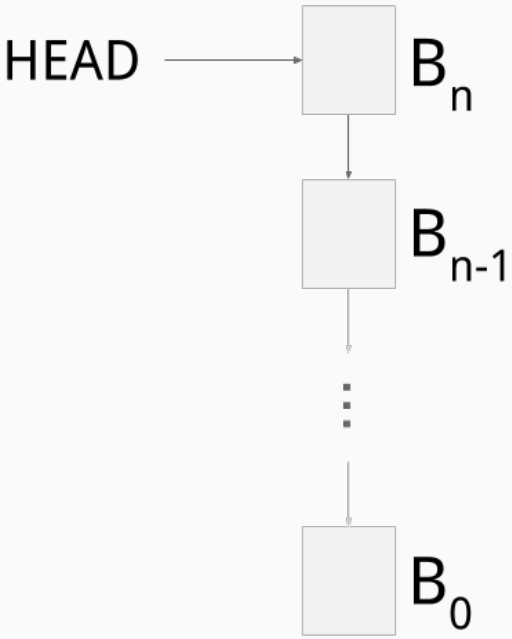
- Key material
  - Signature key
  - Recovery key
- Generic things
  - Encryption keys
  - Signal prekeys
  - Identity in social nets / emails
- Revocations
- Cross-references (will get back to this)

Clients maintain per-device append-only logs of claims.

Hash chains of claims



Claim chain imprint

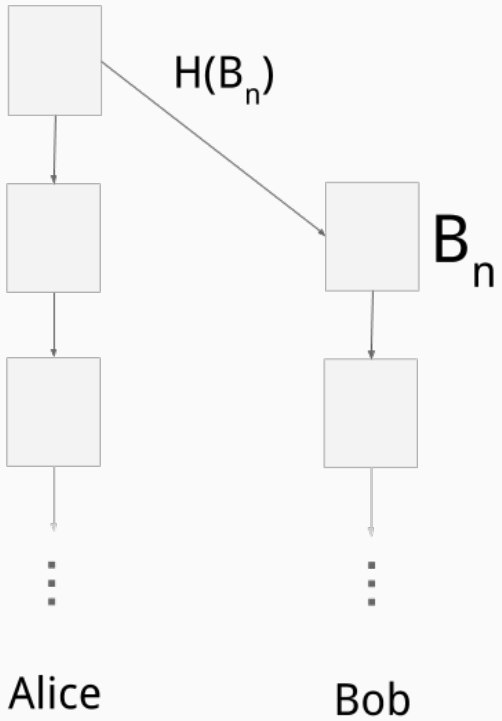


Imprint is a hash of the chain head: $H(B_n)$

- Compact representation of the chain state
- Can verify the integrity of the chain top to bottom
- Signatures allow to verify new blocks

Cross-Referencing

Cross-referencing



- Alice commits to an imprint of Bob's chain
- Resulting in WoT which also tracks the updates of chains

Social evidence processing policy

Validating someone's claim chain need to involve social verification to detect forks (compromise) or fake imprint.

- A client decides a set of other nodes they choose to trust
- Defines client's the trust model

Social evidence processing policy

Validating someone's claim chain need to involve social verification to detect forks (compromise) or fake imprint.

- A client decides a set of other nodes they choose to trust
- Defines client's the trust model

Social evidence processing policy

Validating someone's claim chain need to involve social verification to detect forks (compromise) or fake imprint.

- A client decides a set of other nodes they choose to trust
- Defines client's the trust model

Supporting infrastructure

Storage infrastructure

Options to distribute the claim chains:

- Peer-to-peer / In-band
  - Not efficient
- Centralized storage / the Cloud
  - Can be highly available
  - Easy to deploy
  - No need to trust for integrity!
  - Privacy problems
  - Other security problems
- DHT, etc.

Chains can be stored in KV stores with $K = H(B_i), V = B_i$.

Storage infrastructure

Options to distribute the claim chains:

- Peer-to-peer / In-band
  - Not efficient
- Centralized storage / the Cloud
  - Can be highly available
  - Easy to deploy
  - No need to trust for integrity!
  - Privacy problems
  - Other security problems
- DHT, etc.

Chains can be stored in KV stores with $K = H(B_i), V = B_i$.

Storage infrastructure

Options to distribute the claim chains:

- Peer-to-peer / In-band
  - Not efficient
- Centralized storage / the Cloud
  - Can be highly available
  - Easy to deploy
  - No need to trust for integrity!
  - Privacy problems
  - Other security problems
- DHT, etc.

Chains can be stored in KV stores with $K = H(B_i), V = B_i$.

State tracking mechanism

Need a kind of ”DNS” to resolve names to latest head imprints

- In-band
  - Opportunistic encryption-like
  - Easy to deploy
  - No availability
- Centralized
  - Privacy problems
  - Can be highly available
- Gossiping, DHT, The Blockchain, etc.

State tracking mechanism

Need a kind of ”DNS” to resolve names to latest head imprints

- In-band
  - Opportunistic encryption-like
  - Easy to deploy
  - No availability
- Centralized
  - Privacy problems
  - Can be highly available
- Gossiping, DHT, The Blockchain, etc.

State tracking mechanism

Need a kind of ”DNS” to resolve names to latest head imprints

- In-band
  - Opportunistic encryption-like
  - Easy to deploy
  - No availability
- Centralized
  - Privacy problems
  - Can be highly available
- Gossiping, DHT, The Blockchain, etc.

State tracking mechanism

Need a kind of ”DNS” to resolve names to latest head imprints

- In-band
  - Opportunistic encryption-like
  - Easy to deploy
  - No availability
- Centralized
  - Privacy problems
  - Can be highly available
- Gossiping, DHT, The Blockchain, etc.

Privacy and Security

Access control

- Clients can encrypt blocks so that only chosen groups can read them
- Naive way — encrypt blocks with a session key, encrypt session key with other people public keys
- Attribute-based or predicate-based encryption

Query privacy

Centralized storage infrastructure or state tracking mechanism can learn the social graph

- Privacy through anonymity
- Dummy queries
- Private information retrieval
  - Not practical
  - Relaxed PIR hard to deploy

Summary

ClaimChain:

- Put claims of any nature, mainly cryptographic material, in high-integrity stores
- Clients commit to states of other chains
- Each client defines their source of authority about states
- Complementary to opportunistic encryption efforts
- Allow to be stored on untrusted storage
- Other than setting social policy, can be made automatic

Thank you!

Bogdan Kulynych
PhD student, IMDEA Software Institute, Madrid

Twitter: @hiddenmarkov
Email: `bogdan.kulynych` at `imdea.org`

NEXTLEAP project
`nextleap.eu`