SNAKE

A privacy-aware online social network
providing anonymity of outsourced data at rest

Alessandro Di Federico

30th Chaos Communication Congress - YBTI

December 28th, 2013

Index

What's SNAKE?

SNAKE is a social network for end-to-end encrypted communications

The starting point: OpenPGP

- Allows to exchange e-mail (and more) securely

The starting point: OpenPGP

- Allows to exchange e-mail (and more) securely
- Provides:
  - confidentiality
  - authenticity
  - integrity

OpenPGP: weak points

- Complicated for the average end-user
- Group communication does not scale
- Exposes the social graph
- Requires out-of-band communication

Index

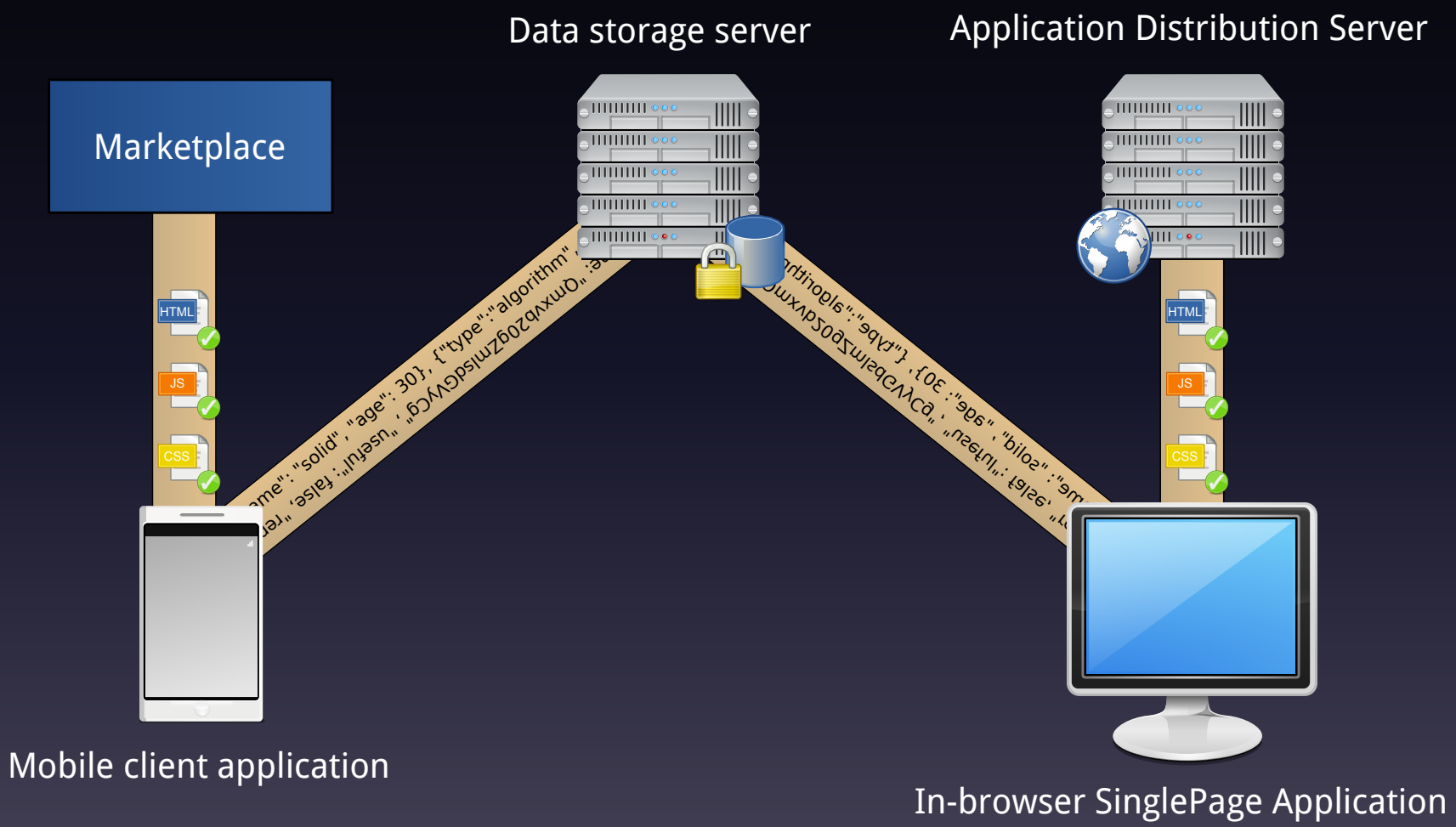
Discarded options

- Overlay

Discarded options

- Overlay
- P2P

Architecture



Code distribution server

- It just distributes the client

Code distribution server

- It just distributes the client
- Can be multiple (GNU/Linux philosophy)

Code distribution server

- It just distributes the client
- Can be multiple (GNU/Linux philosophy)
- It’s optional

Code distribution server

- It just distributes the client
- Can be multiple (GNU/Linux philosophy)
- It’s optional
- If used, it’s a trusted component

Storage server

- Very dumb, just a CRUD interface to a DB

Storage server

- Very dumb, just a CRUD interface to a DB
- Let's consider two scenarios
  - MALICIOUS-SERVER
    - It can try to tamper with user data
    - It can save additional metadata (IP, User agent...)

Storage server

- Very dumb, just a CRUD interface to a DB
- Let's consider two scenarios
  - MALICIOUS-SERVER
    - It can try to tamper with user data
    - It can save additional metadata (IP, User agent...)
  - HONEST-SERVER
    - It doesn't save additional metadata

Client

- It’s an HTML5 in-browser Single Page Application

Client

- It’s an HTML5 in-browser Single Page Application
- The UI recalls those of a classical social network

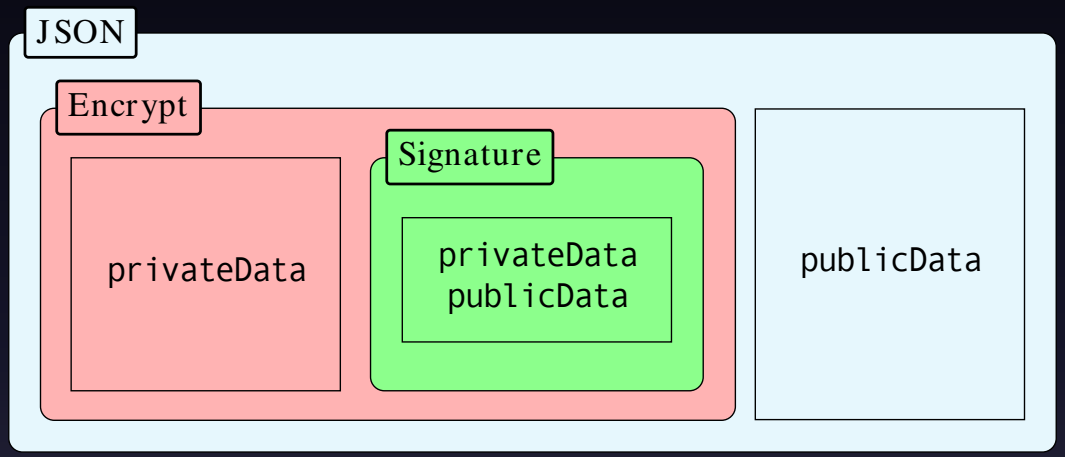
Client

- It’s an HTML5 in-browser Single Page Application
- The UI recalls those of a classical social network
- Encrypts and decrypts all communications on the fly

Client

- It’s an HTML5 in-browser Single Page Application
- The UI recalls those of a classical social network
- Encrypts and decrypts all communications on the fly
- Uses WebCrypto API, a W3C standard
  - PBKDF2 for password-based key derivation
  - AES-GCM per symmetric encryption/decryption
  - SHA-256 as a cryptographic hash function
  - ECDSA for digital signatures

How is an object stored?



Each object is split in two:

`publicData` public keys, unique identifiers and so on

`privateData` all the sensitive data

Index

Index

What’s a friendship?

Establish a secure 1-to-1 communication channel

What's a friendship?

Establish a secure 1-to-1 communication channel

Obtain a common symmetric key

Soooooo...

No, wait, we already talk about this

Soooooo...

No, wait, we already talk about this

Basically we are able to authenticate public keys:

- without out-of-band communication
- using a Q&A system or...
- a private Web of Trust

Index

Many-to-many communication

- We want to be scalable w.r.t.:
  - Messages exchanged and saved
  - Group management (e.g. user removal)

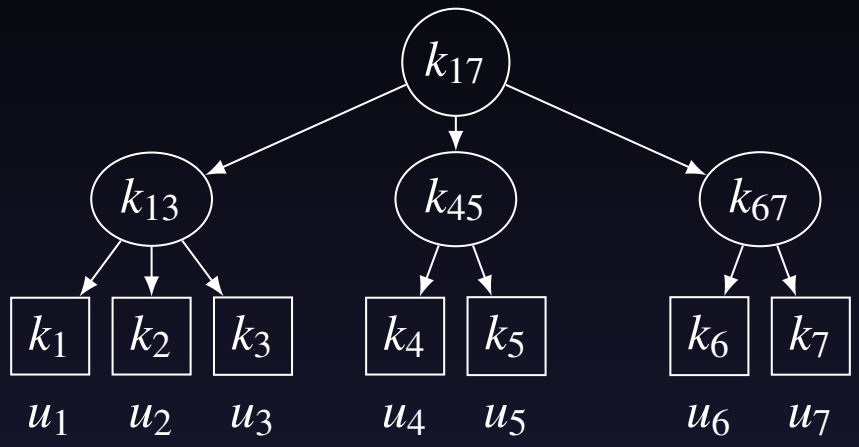
Group communication handling

- Discarded solutions
  - PGP
    - The message key is encrypted multiple times
    - Let in a new member is costly

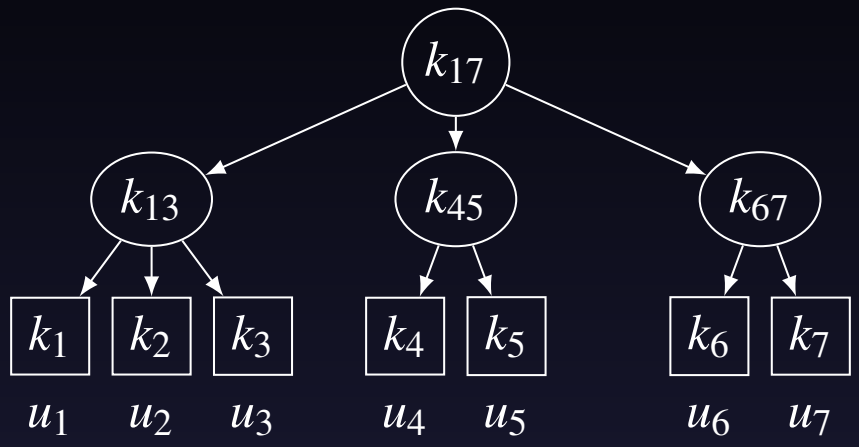
Group communication handling

- Discarded solutions
  - PGP
    - The message key is encrypted multiple times
    - Let in a new member is costly
- Our solution
  - Hierarchic key management
    - The group uses a single key
    - Group can be managed efficiently

Hierarchic key distribution

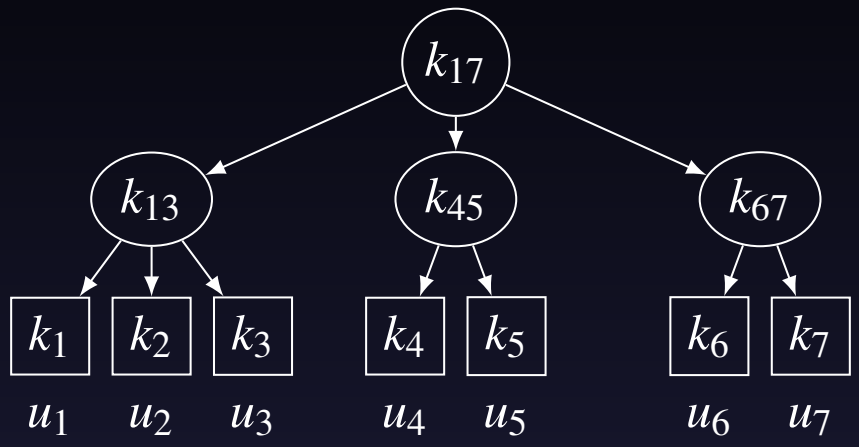


Hierarchic key distribution



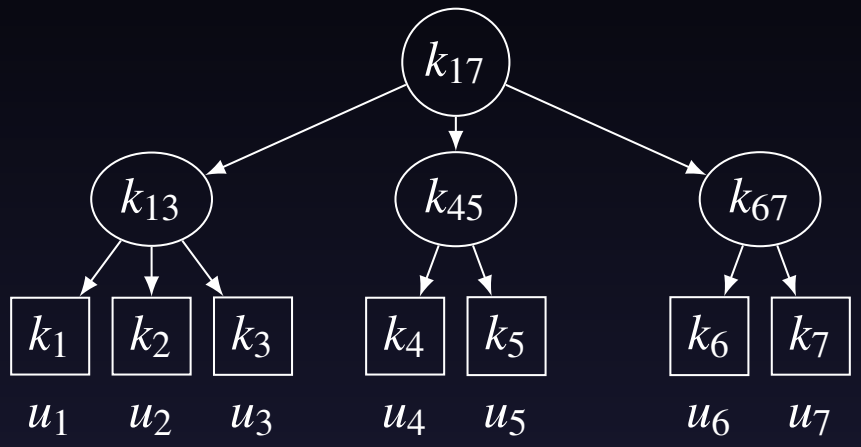
- Nodes contain cryptographic keys
  - The root node contains the group key

Hierarchic key distribution



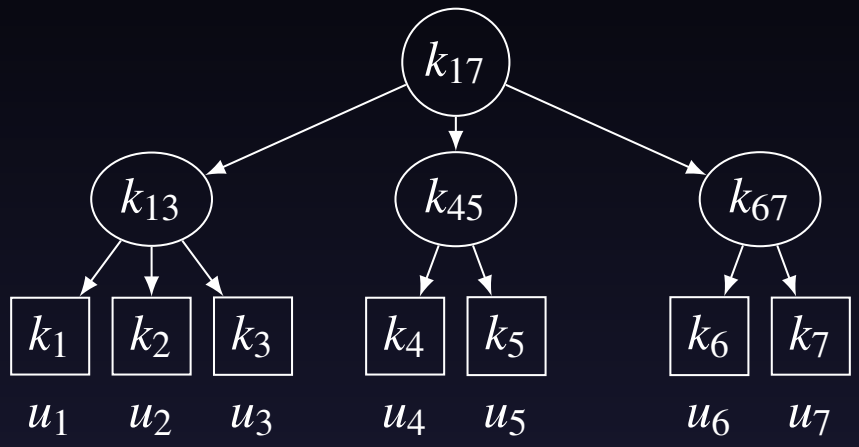
- Nodes contain cryptographic keys
  - The root node contains the group key
  - Leaf nodes represent users

Hierarchic key distribution



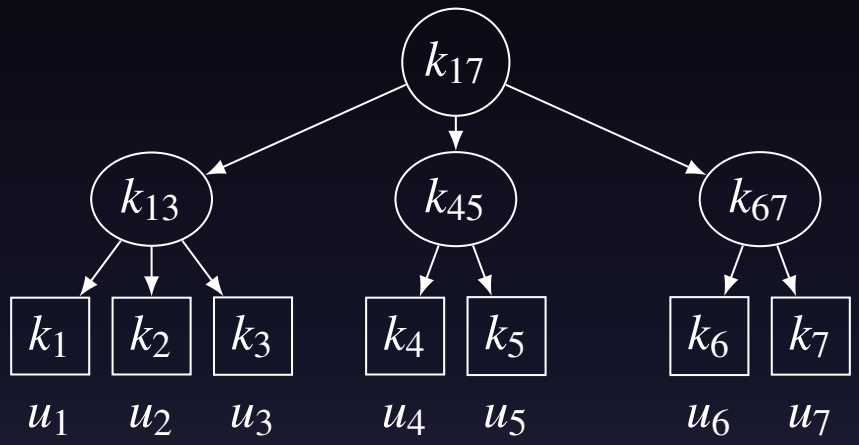
- Nodes contain cryptographic keys
  - The root node contains the group key
  - Leaf nodes represent users
  - A user knows keys from his leaf up to the root

Hierarchic key distribution



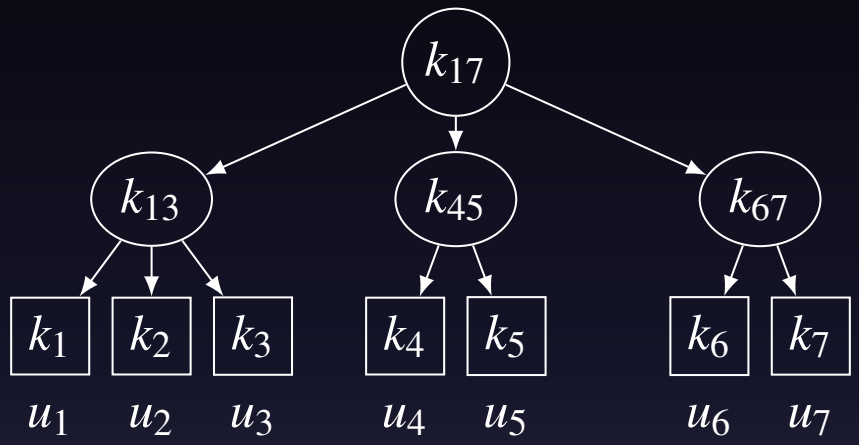
- Nodes contain cryptographic keys
  - The root node contains the group key
  - Leaf nodes represent users
  - A user knows keys from his leaf up to the root
- Intermediate keys allow communication with subgroups

Tree characteristics



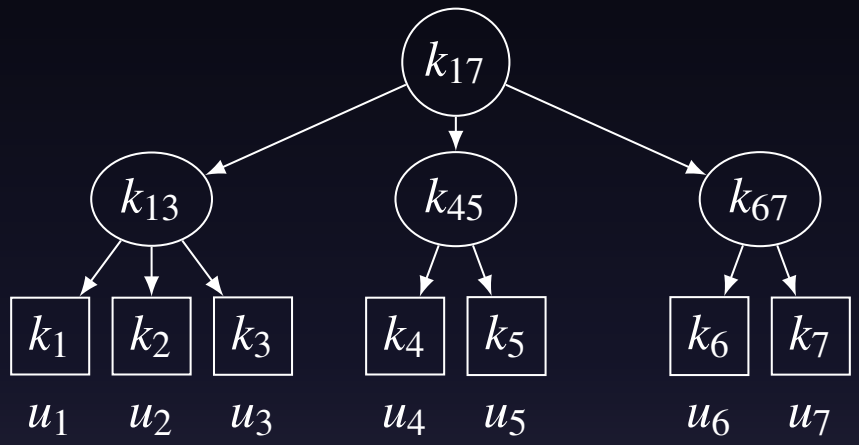
- It's called NSBHO

Tree characteristics



- It's called NSBHO
- It's **not** a search tree

Tree characteristics



- It's called NSBHO
- It's **not** a search tree
- Reduces management costs

Implementations choices

- We use a symmetric key for group communications
- The key is changed only upon user removal
- Inserting a new member has zero cost
- Removing a member has logarithmic cost
- We optimized the grade of the tree for 5% removal

Index

Not just confidentiality of contents

SNAKE also offers anonymity of data
through suppression of public metadata

Metadata

Looking at a dump of the database you **cannot**:

Metadata

Looking at a dump of the database you **cannot**:

- Understand who is the sender of a message

Metadata

Looking at a dump of the database you **cannot**:

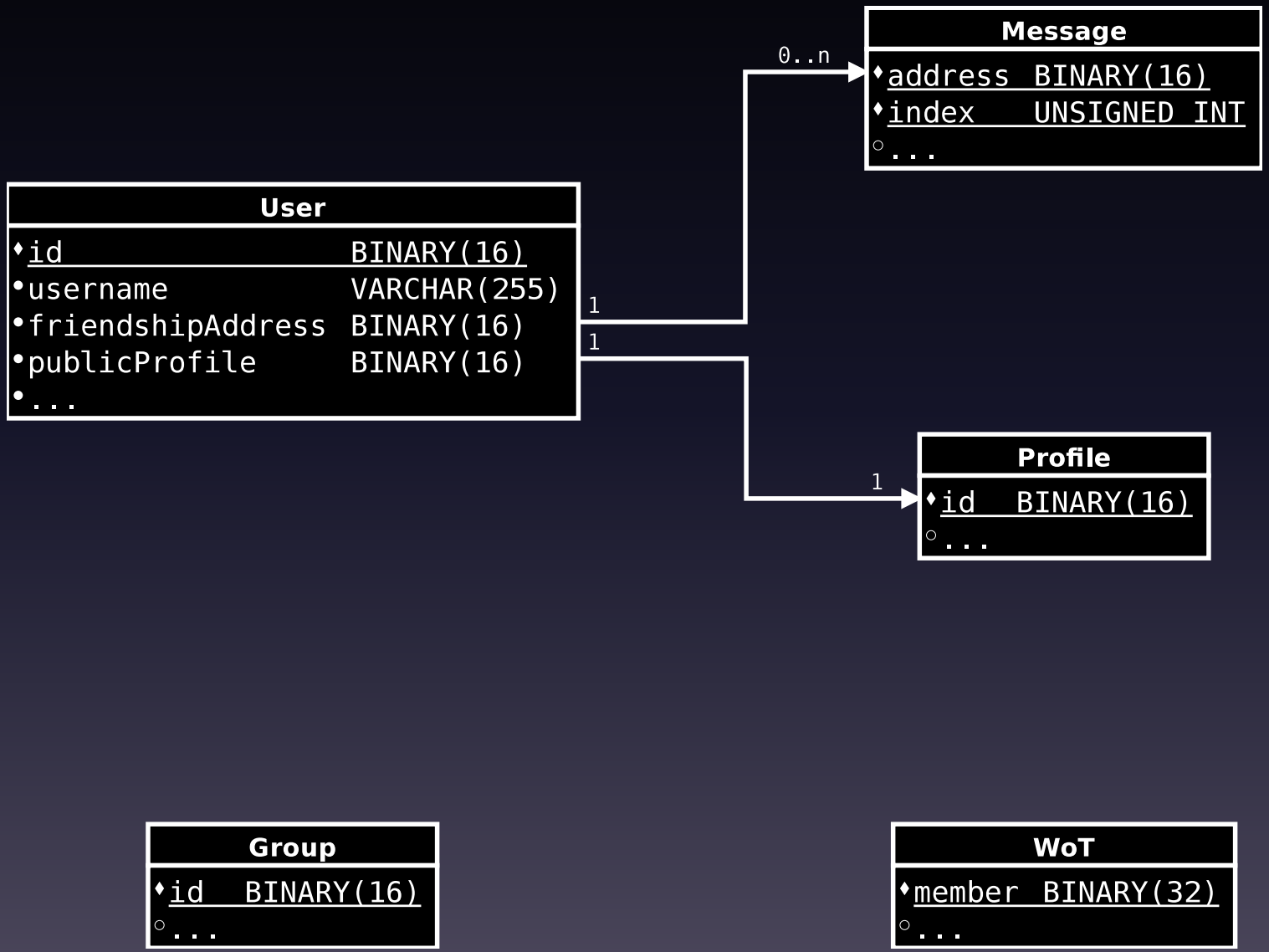
- Understand who is the sender of a message
- Understand who is the recipient

Metadata

Looking at a dump of the database you **cannot**:

- Understand who is the sender of a message
- Understand who is the recipient
- Understant if two users are friends or not

Survived join paths



Anonymity

We provide anonymity of data **at rest**

The scenario

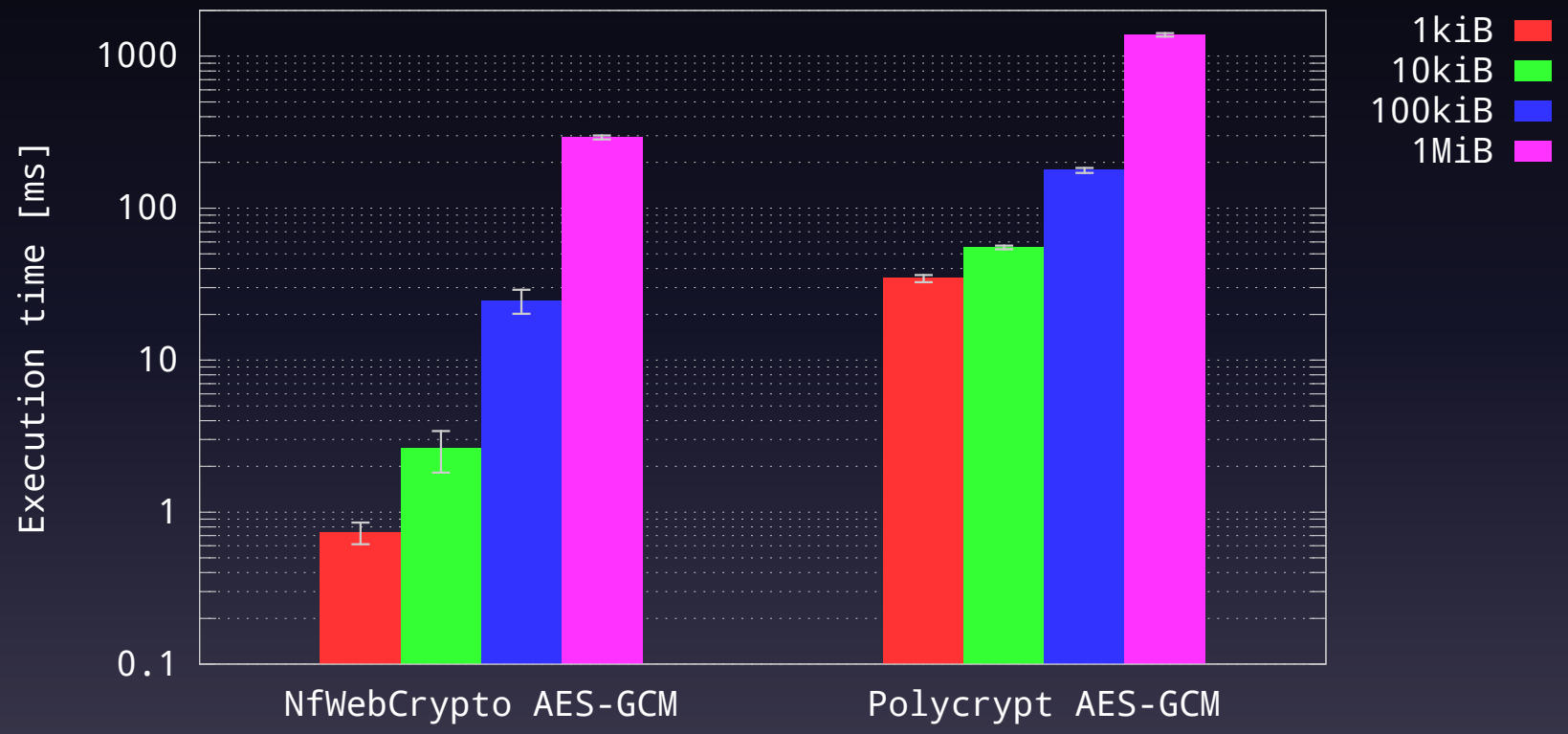
We can provide it if the storage server is honest

Index

WebCrypto API

- We benchmarked two implementations
  - PolyCrypt (JavaScript)
  - NfWebCrypto (Google Chrome/Chromium plugin using OpenSSL)

Encryption performance



Overhead for the end user

- We measured the introduced overhead for:
  - login
  - registration
  - message exchange
  - …

Overhead for the end user

- We measured the introduced overhead for:
  - login
  - registration
  - message exchange
  - …
- We used:
  - the NfWebCrypto implementation
  - a dummy implementation

Network

- We simulated three networks configuration

Network

- We simulated three networks configuration

  NET-LOOPBACK client and server communicate over loopback

Network

- We simulated three networks configuration

  NET-LOOPBACK client and server communicate over loopback

  NET-FAST client and server communicate through a network with a RTT of 40 ms and 10 Mbit of bandwidth

Network

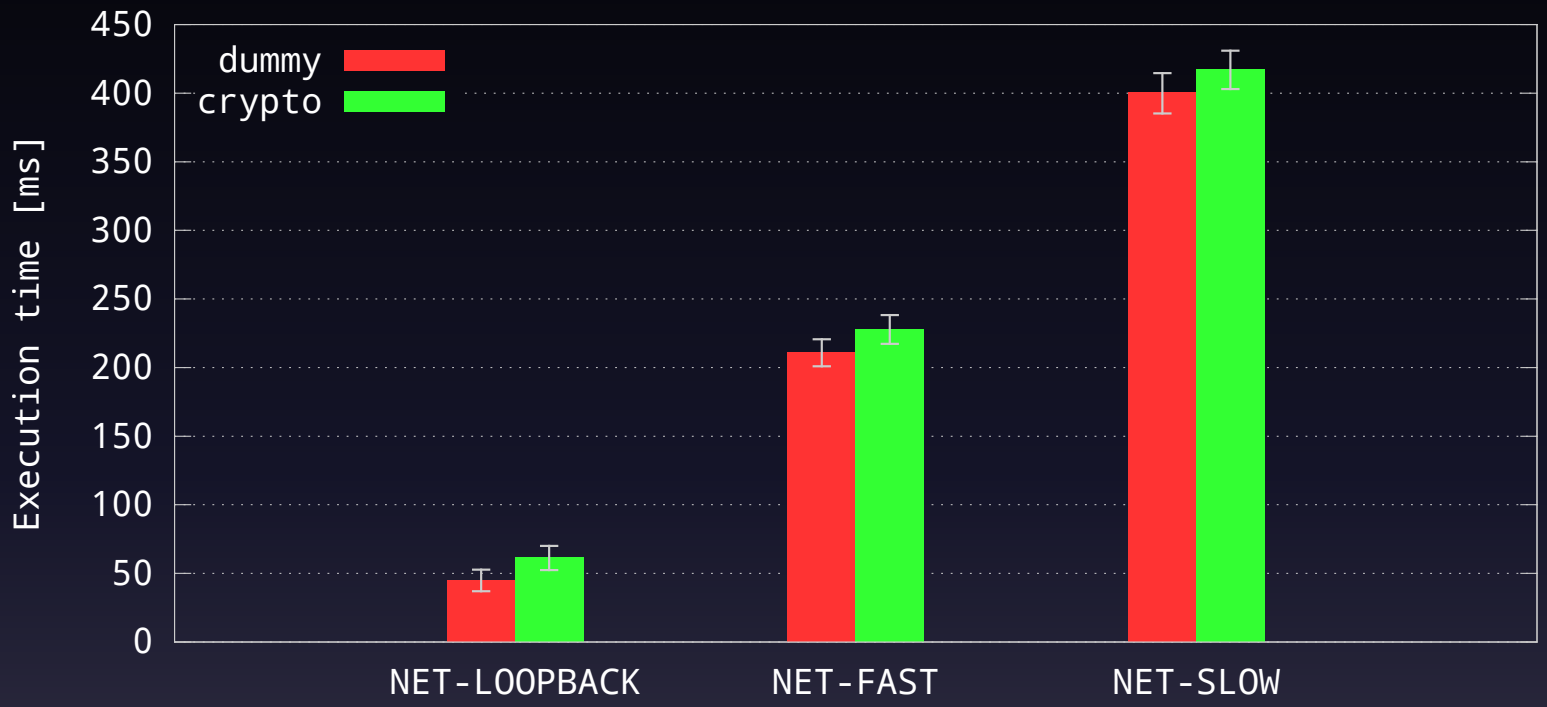
- We simulated three networks configuration

NET-LOOPBACK
: client and server communicate over loopback

NET-FAST
: client and server communicate through a network with a RTT of 40 ms and 10 Mbit of bandwidth

NET-SLOW
: client and server communicate through a network with a RTT of 100 ms and 1 Mbit of bandwidth

Overhead for login



Rete	Overhead
LOOPBACK	26.72%
FAST	7.5%
SLOW	4.09%

Index

What we've done

- A system architecture dividing responsibilities
- Privacy-aware and in-band public key authentication
- Efficient and scalable group communication
- Anonymity of data at rest

Future developments

- Real time chat

Future developments

- Real time chat
- Simple secure file sharing

Future developments

- Real time chat
- Simple secure file sharing
- Collaborative online office suite

Coordinates

Questions?
http://snake.li/

License