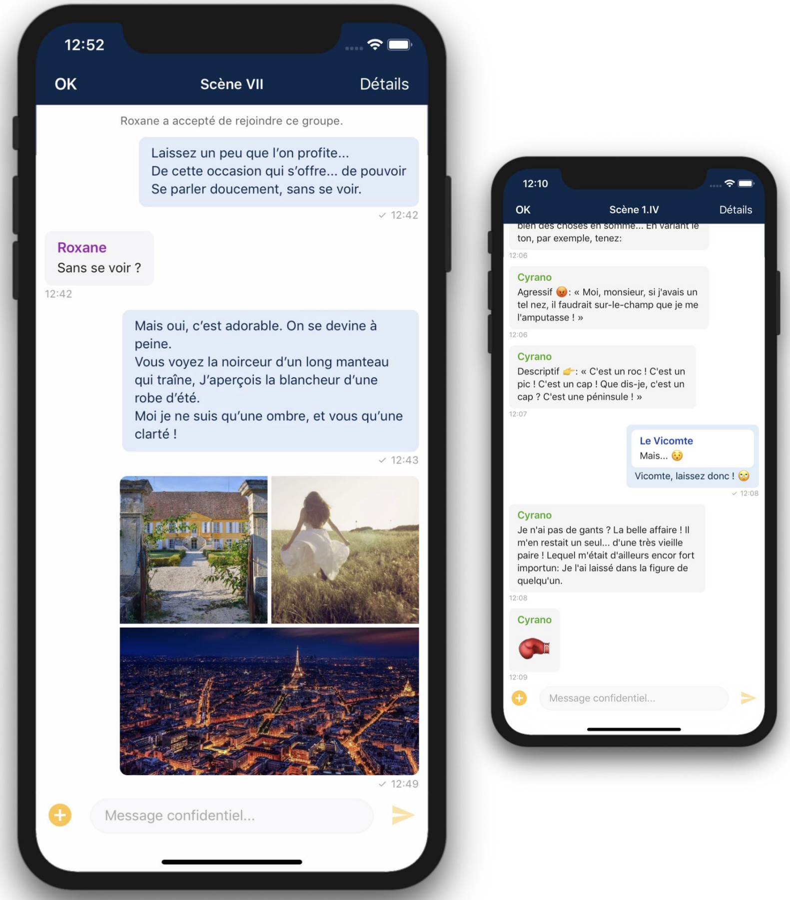
Olvid.

Security Model of Mobile Messaging Apps.



GDR Sécurité Informatique
June 12, 2019

Who are we?

Thomas Baignères

CEO @ Olvid

Cryptography PhD (EPFL)

Matthieu Finiasz

CTO @ Olvid

Cryptography PhD (ENS - INRIA)

Co-founder CybelAngel

True secure messaging

Only cryptography can guarantee the complete security of your communications

1. Security properties
2. Security model
3. Authentication
4. Data encryption
5. Metadata encryption

1. Security properties.
2. Security model
3. Authentication
4. Data encryption
5. Metadata encryption

Which security properties?

The security of a closed-door meeting

In a digital world

Which security properties?

The security of a closed-door meeting

- Everyone knows exactly who he is talking to
- No one else hears what is being said
- The discussion does not leave any trace

In a digital world

Which security properties?

The security of a closed-door meeting

- Everyone knows exactly who he is talking to
- No one else hears what is being said
- The discussion does not leave any trace

In a digital world

- Asynchronous communications
- Attachments of all sorts
- Instantaneity, whatever the distance

3 Pillars

Authentication

Data Encryption

Metadata Encryption

3 Pillars

Authentication	Data Encryption	Metadata Encryption

1 Principle

"Minimal Disclosure"

Always disclose as little information as possible to third parties

3 Pillars

Authentication	Data Encryption	Metadata Encryption

1 Principle

"Minimal Disclosure"

Always disclose as little information as possible to third parties

2 Constraints

Efficiency

Both client-side and server-side, with a minimum number of exchanges

&

Ease of use

As few user constraints as possible

1. Security properties
2. Security model.
3. Authentication
4. Data encryption
5. Metadata encryption

Base security model.

Bellare-Rogaway

The adversary:

- controls **the network**
- controls **intermediate nodes**
- can **start protocols**

Network & nodes control

Read network packets, modify them, insert, delete, reorder, delay, etc.

Discreet adversary

The adversary does not want to be detected
Loose “honest-but-curious” model

The user is an “adversary”.

The user is not an expert

Users do not understand the security implications of their choices. They will make poor security choices.

→ security should never rely on user choices

No password

- Very weak in 50% cases
- Only for “over-securing” something already secure

Security-by-design

If the user has a choice, all alternatives should give a sufficient security level

Security model

The user is his own adversary. He will always pick the worst possible choice.

Forward secrecy.

Devices are considered **“healthy”** (no malware), but device theft can’t be ignored for a mobile application:

- The OS cannot be seen as a sufficient security layer
  → device theft gives access to the **full device content**
- It should not give access to anything else
  → erased contacts and messages should remain **erased forever**

Long term keys should **never be used to encrypt** sensitive data or user content

Forward secrecy.

Devices are considered **“healthy”** (no malware), but device theft can’t be ignored for a mobile application:

- The OS cannot be seen as a sufficient security layer
  → device theft gives access to the **full device content**
- It should not give access to anything else
  → erased contacts and messages should remain **erased forever**

Long term keys should **never be used to encrypt** sensitive data or user content

Long term keys security model

At any point in time, the adversary can steal long term keys.
This should not jeopardize the security of past exchanges.

Multi-user & multi-instance.

Cryptographic models often consider Alice and Bob, isolated from the rest of the world:

- A messaging app can have millions of users
- The adversary does not necessarily target one specific user
  → **"1-in-N"** attack model
- Each user is in contact with dozens of correspondents
  → **multi-instance** attack model
- Behind each device, there is a **human being**, with limited "bandwidth"

Protocols without user interaction

Thousands of instances in parallel
With **thousands** of users

Protocols with user interaction

A few instances in parallel
With **a few** users

The right security model.

Security Model

Like for a "closed-door meeting", the outside world is hostile, but wants to remain unnoticed.

Hypothesis

- *Almost honest* servers
- Users know & trust each other
- User devices are healthy during the conversations

Attack capacity

Adversary controlled servers:

- make copies of messages
- statistical analysis
- modify messages
- try MitM attacks, etc.

Attacker goal

Gather any kind of undisclosed information:

- who speaks to whom?
- how often?
- to say what?

1. Security properties
2. Security model
3. Authentication.
4. Data encryption
5. Metadata encryption

Authentication of a public key.

Setup

- Alice and Bob want to talk
- They share nothing in the digital world
- Both have a long term key pair

Objective

- Exchange their public keys
- Authenticate them
  → tie them to an identification element

Authentication of a public key.

Setup

- Alice and Bob want to talk
- They share nothing in the digital world
- Both have a long term key pair

Objective

- Exchange their public keys
- Authenticate them
  → tie them to an identification element

2 different approaches

Transferable proof

- Using digital signatures by TTP
- Example: Certification Authority

Authentication of a public key.

Setup

- Alice and Bob want to talk
- They share nothing in the digital world
- Both have a long term key pair

Objective

- Exchange their public keys
- Authenticate them
  → tie them to an identification element

2 different approaches

Transferable proof

- Using digital signatures by TTP
- Example: Certification Authority

Interactive proof

- Relying on an authenticated channel
- Examples: PGP, Bluetooth pairing

WhatsApp: Trusted Third Party approach.

Phone number ≠ individual

- Inappropriate identification element
- Might get reattributed to someone else
- Relies on the security of a single SMS 😱

Imposed Trusted Third Party

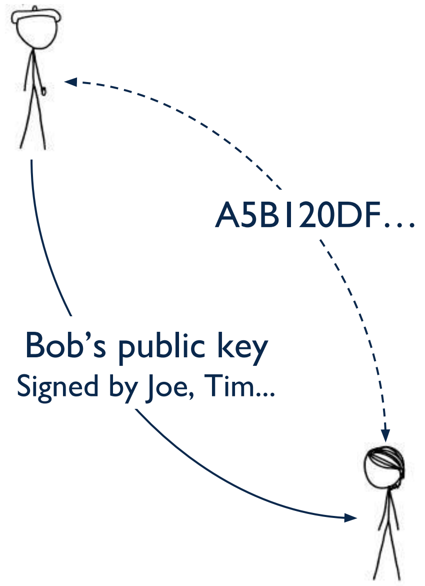
- Foundation of the whole security
- Controlled by WhatsApp
- … or the NSA, or some unnoticed hacker

Users should be able to choose who they trust and how they identify contacts

PGP: hybrid approach.

PGP key authentication relies on a **web of trust**:

- either relying on signatures by trusted PGP users
- or direct authentication through a fingerprint verification
  → face-to-face or phone interaction



Signature validation

- may involve intermediates
- hard to assess trust level
- complex to understand

Fingerprint verification

- tedious
- optional
  → who does that?

Most PGP keys are not authenticated before use

Different situations, different methods…

Fundamental aspects of authentication:

- Never **associate a public key to an identity** without a valid reason to do so
- The user should **choose who he accepts to trusts**
- Propose different methods depending on the user's **"relation" to the contact**

Different situations, different methods…

Fundamental aspects of authentication:

- Never **associate a public key to an identity** without a valid reason to do so
- The user should **choose who he accepts to trusts**
- Propose different methods depending on the user's **"relation" to the contact**

Face-to-face

- Clear authentic channel
- Limited bandwidth
- Fallback method that "always" work

Corporate

- PKI or AD in place
- Already trusted
- Perfect for internal use, does not work outside

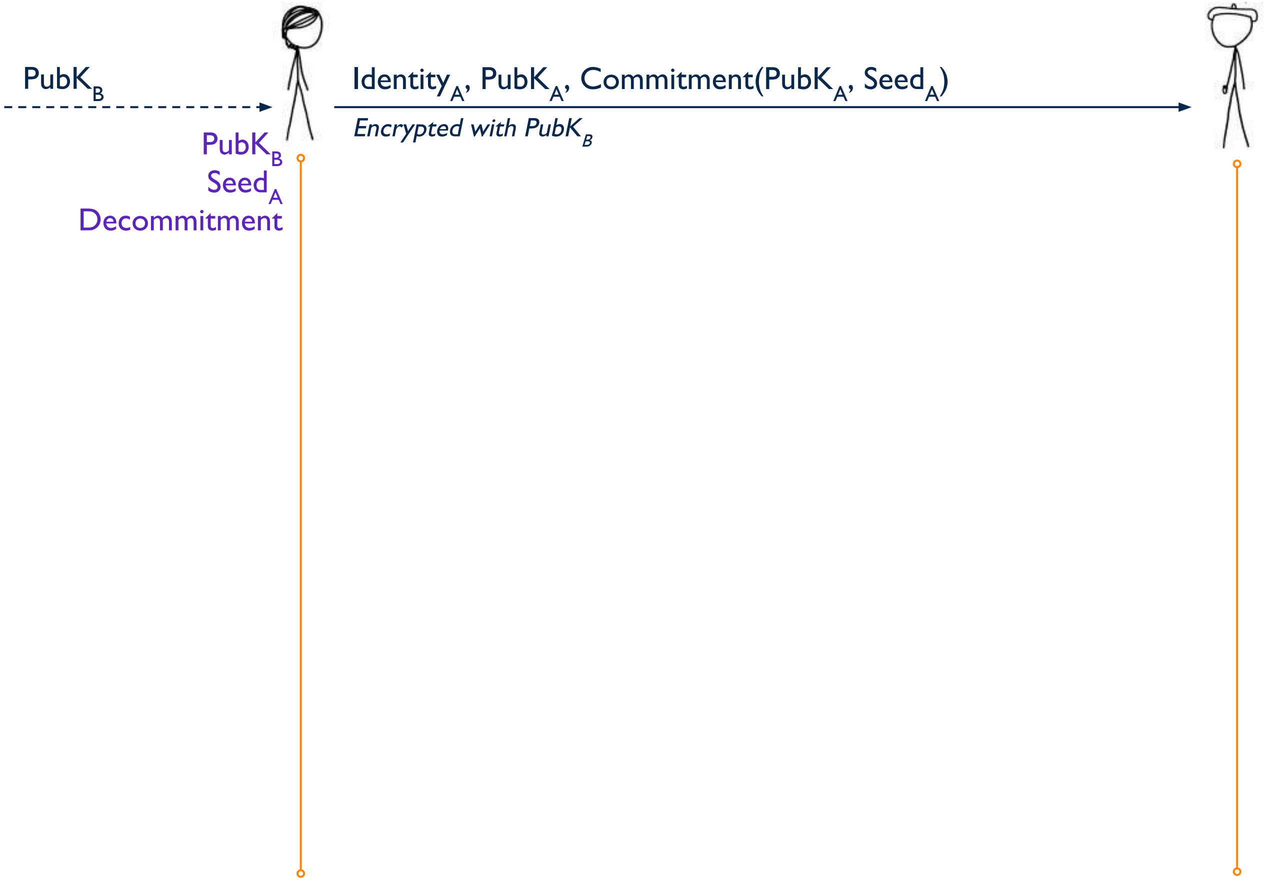
Introduction

- When Alice knows Bob through Charlie
- Charlie is the "relation"
- Charlie must be trusted

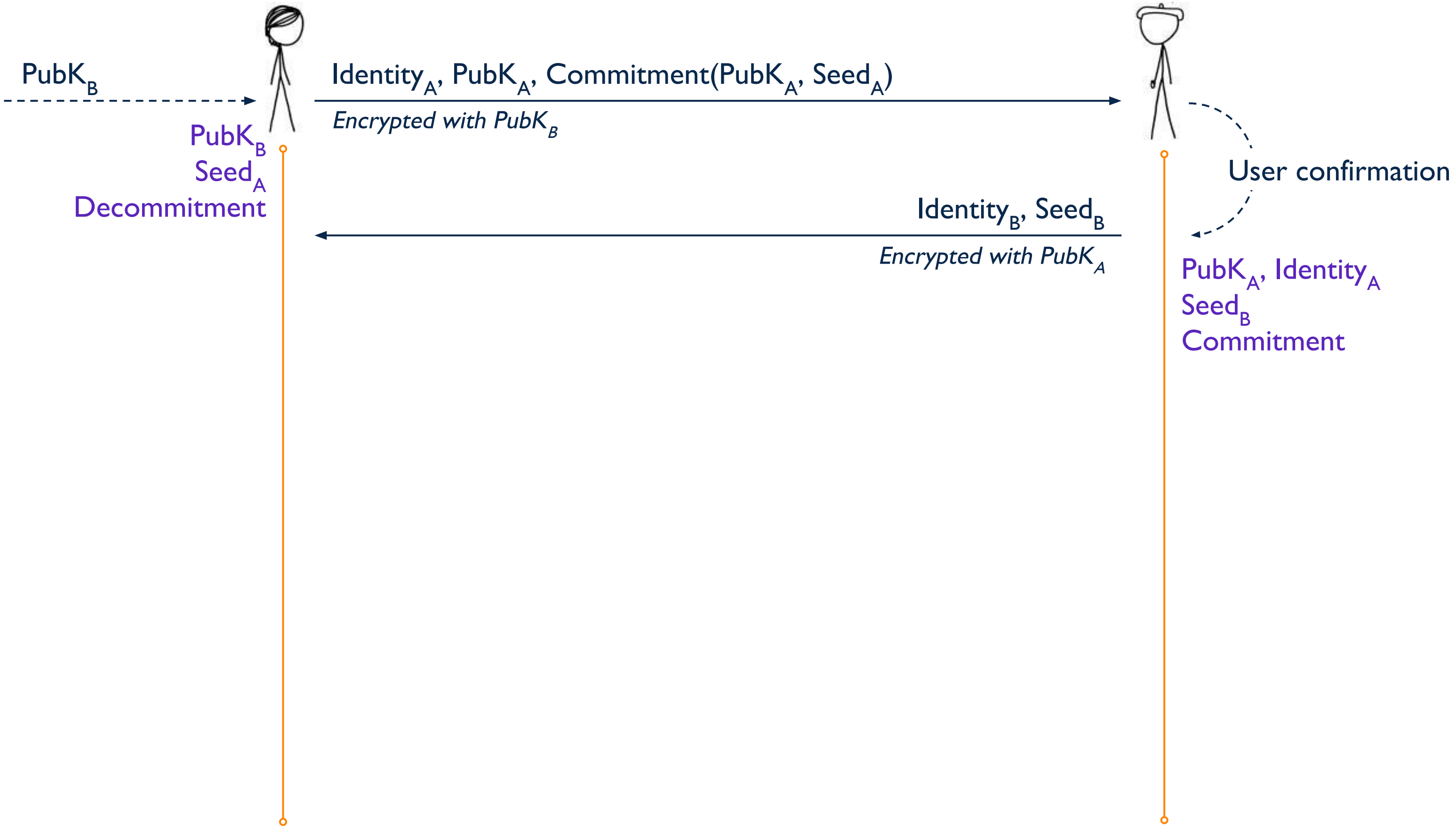
SAML/OAuth

- An email address can be the identification element
- Prove that you own the email address

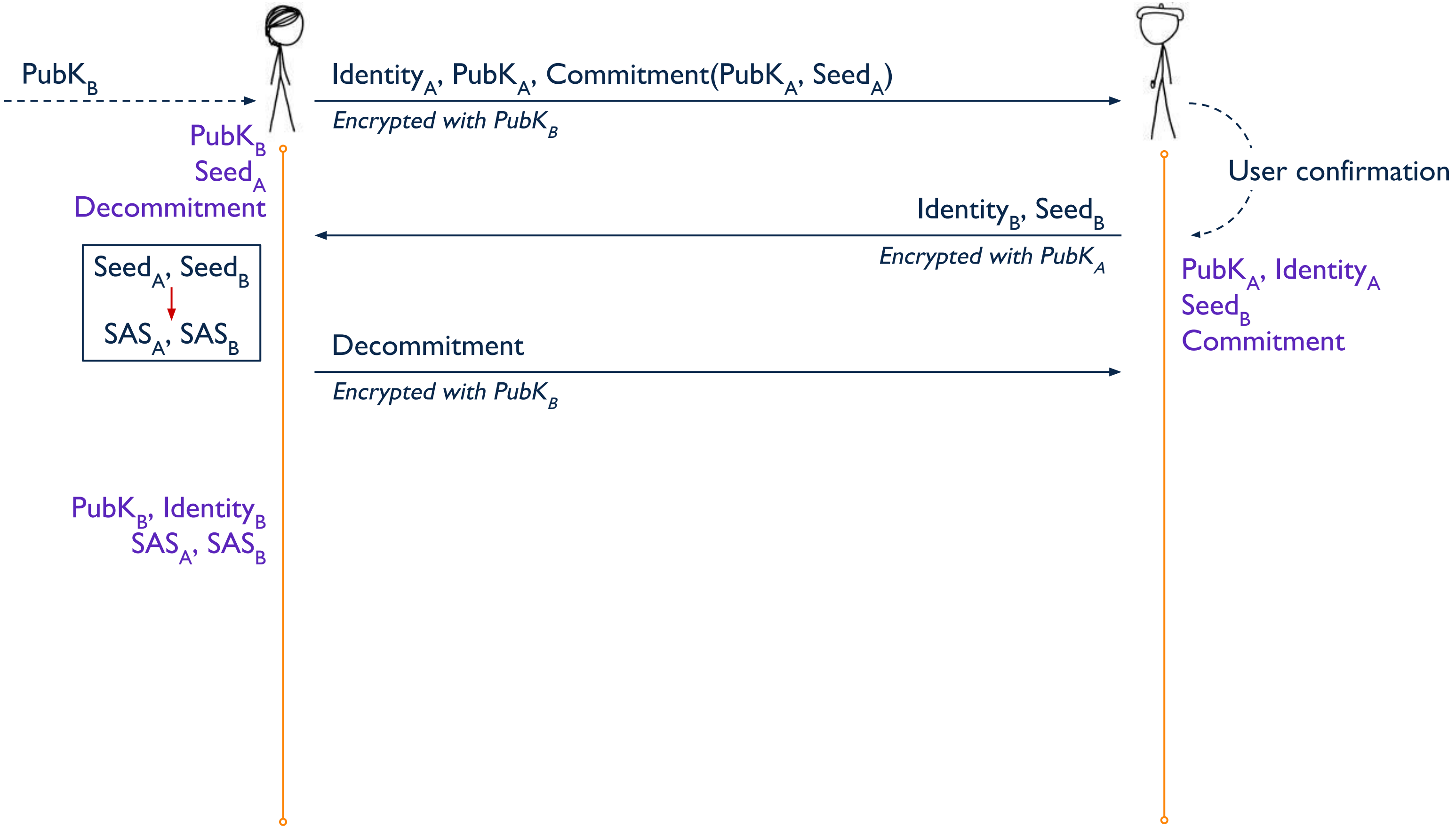
Olvid’s SAS-based key exchange.



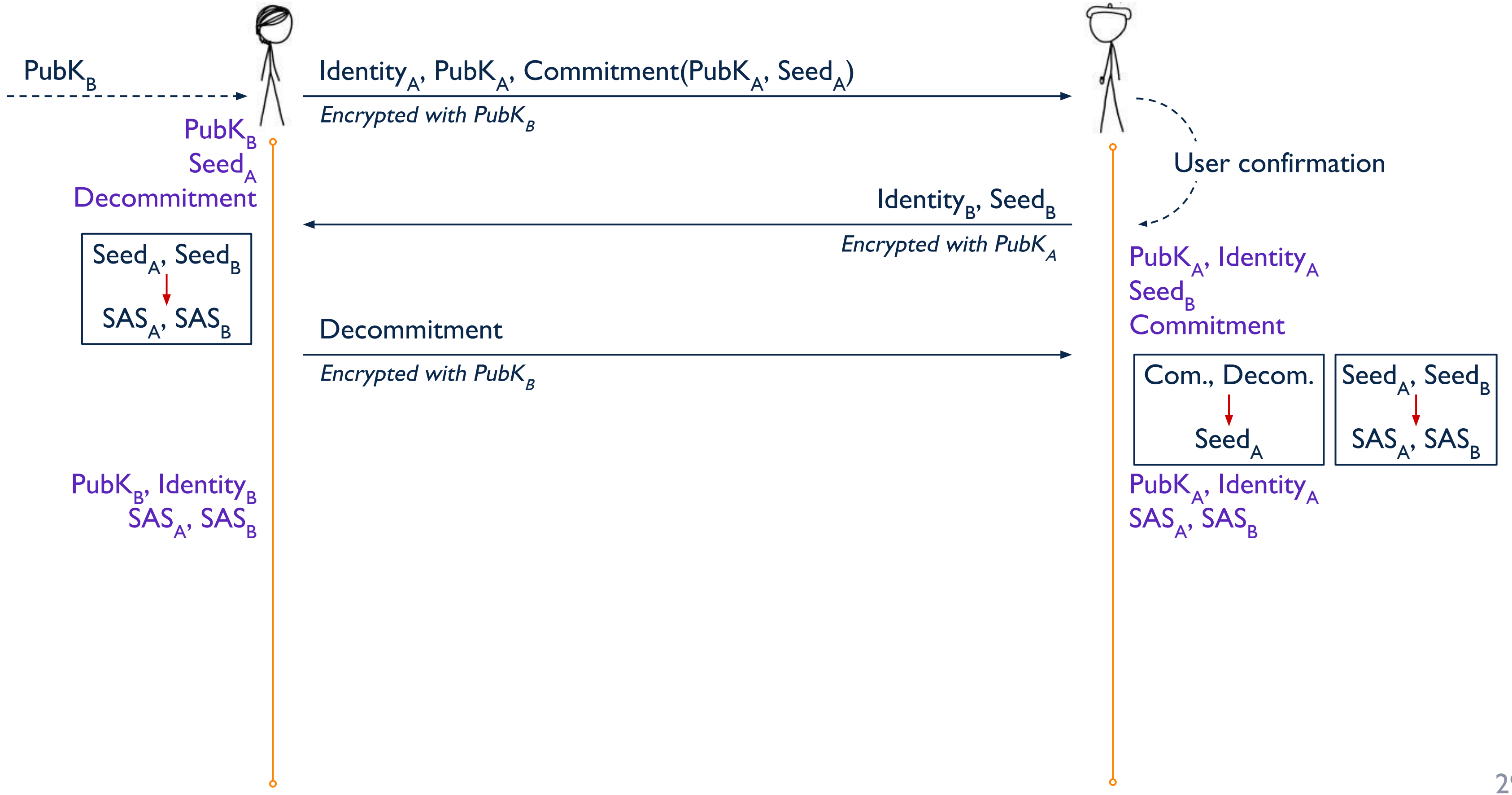
Olvid's SAS-based key exchange.



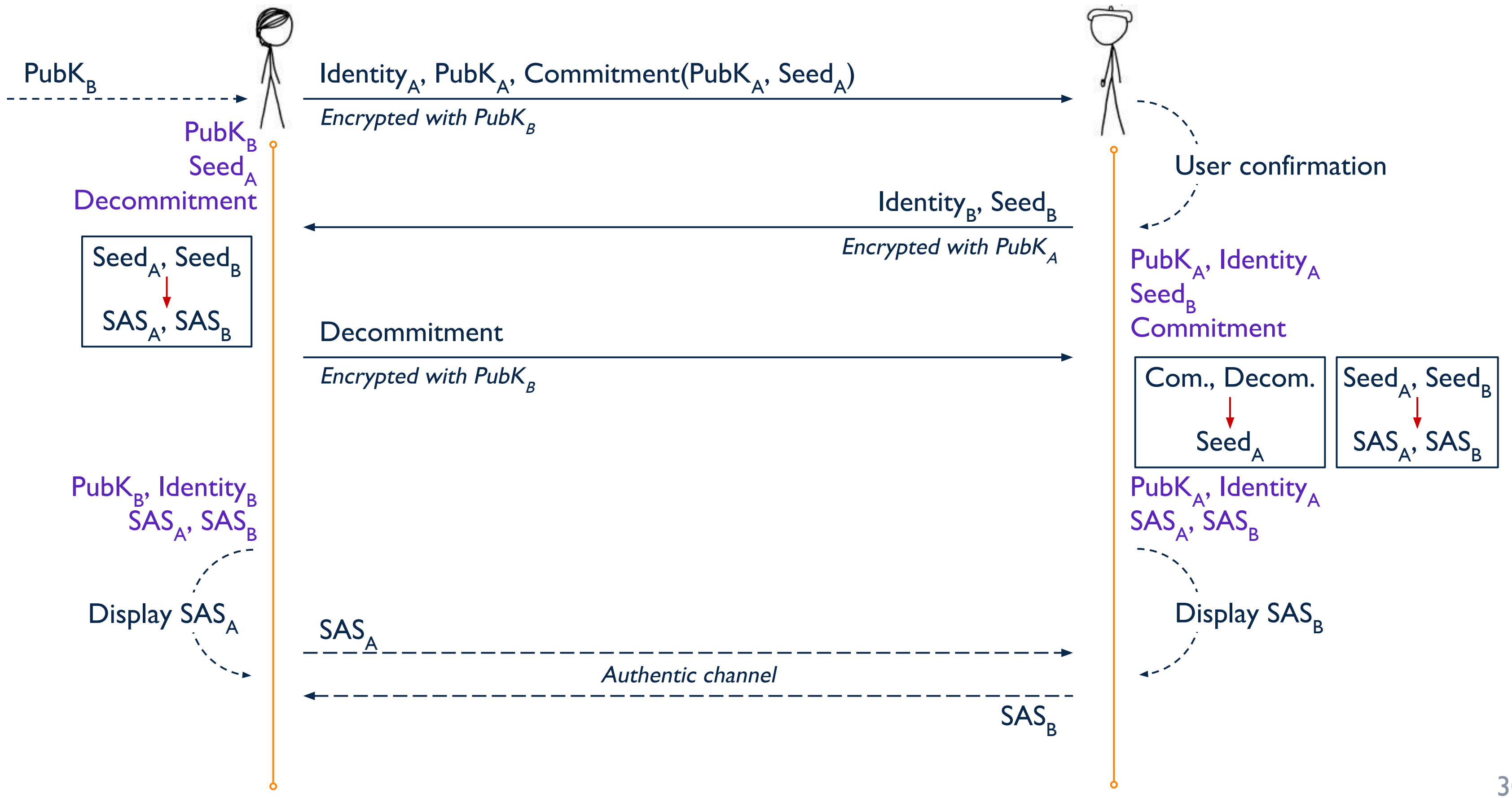
Olvid's SAS-based key exchange.



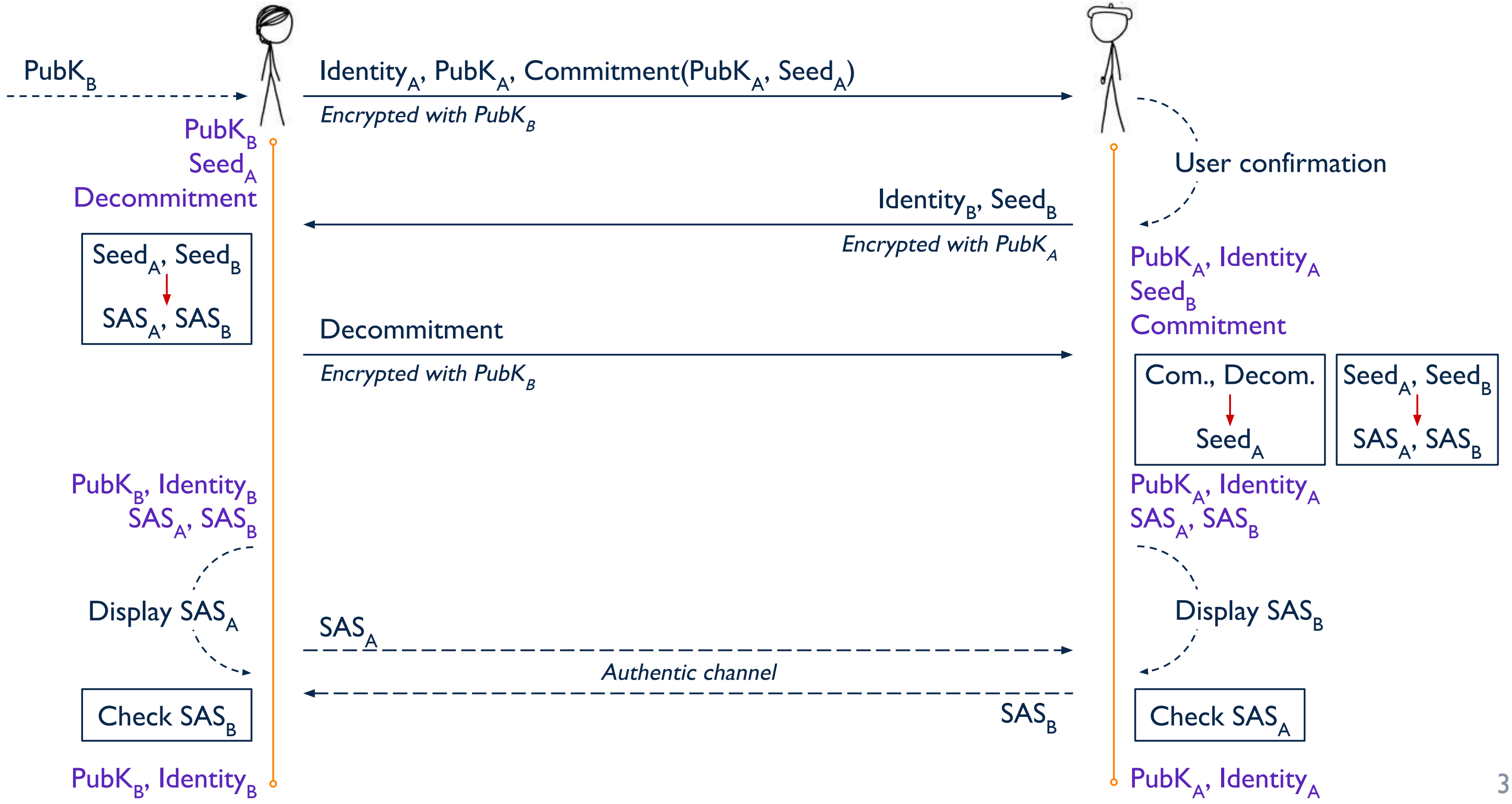
Olvid's SAS-based key exchange.



Olvid's SAS-based key exchange.



Olvid's SAS-based key exchange.



1. Security properties
2. Security model
3. Authentication
4. Data encryption.
5. Metadata encryption

What does data encryption mean?

All **user data** should transit through an **end-to-end secure channel**

Confidentiality	Authenticity	Integrity
Deniability	Forward-secrecy	Backward-secrecy

From authenticated public keys to secure channel.

Setup

- Alice and Bob want to talk
- They trust each other's long term public key

Objective

- Agree on a shared secret
- Use it to bootstrap a secure channel

Public keys $\longrightarrow$ Shared secret $\longrightarrow$ Secure channel

From authenticated public keys to secure channel.

Setup

- Alice and Bob want to talk
- They trust each other's long term public key

Objective

- Agree on a shared secret
- Use it to bootstrap a secure channel

Public keys → Shared secret → Secure channel

- Many approaches (DH, KEM, etc.)
- One principle:
  - Ephemeral keys
  - Authenticated using long term keys

From authenticated public keys to secure channel.

Setup

- Alice and Bob want to talk
- They trust each other's long term public key

Objective

- Agree on a shared secret
- Use it to bootstrap a secure channel

Public keys → Shared secret → Secure channel

- Many approaches (DH, KEM, etc.)
- One principle:
  - Ephemeral keys
  - Authenticated using long term keys

- Self ratcheting to derive:
  - one-time keys
  - "random" messages ids
- Used for authenticated encryption

Olvid's two kinds of encryption.

Asymmetric
(long term key)

- Used during the creation of the secure channel
- And nowhere else!

Symmetric
(Secure channel)

- One-time keys → with double ratcheting
- Authenticated encryption
- Message id allows to efficiently determine which secret key to use for decryption

Encrypted data format

`<recipient public key> + <noise>`

- **Asymmetric case:** `<noise> = <encrypted data>`
- **Symmetric case:** `<noise> = <message id> + <encrypted data>`

Olvid's military grade encryption ¯_(ツ)_/¯.

Asymmetric
(long term key)

- KEM ⟶ ECIES (Curve25519)
- KDF ⟶ secure PRNG (HMAC with SHA256)

Symmetric
(secure channel)

- Encrypt then MAC
- Encryption: AES256 in CTR mode
- Authentication: HMAC with SHA256

1. Security properties
2. Security model
3. Authentication
4. Data encryption
5. Metadata encryption.

Metadata in encrypted mail.

```
Return-Path: <alice@wanadoo.fr>
Received: from [10.0.101.17] (tui75-2-82-66-245-153.wanadoo.fr. [76.66.245.153])
        by smtp.cegetel.net with ESMTPSA id w125sm2216593wmw.18.2019.05.09.03.26.14
        for <bob@cegetel.net>
        (version=TLSv1/SSLv3 cipher=OTHER);
        Fri, 05 Apr 2019 03:26:15 -0700 (PDT)
Subject: Document confidentiel
References: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
To: Bob <bob@cegetel.net>
From: Alice <alice@wanadoo.fr>
X-Forwarded-Message-Id: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Message-ID: <56F26F45.2080208@wanadoo.fr>
Date: Fri, 05 Apr 2019 11:26:13 +0200
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Thunderbird/60.6.1
MIME-Version: 1.0
In-Reply-To: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Content-Type: multipart/mixed; boundary="------------030309080003040107080504"

This is a multi-part message in MIME format.
--------------030309080003040107080504
Content-Type: text/plain; charset=windows-1252
Content-Transfer-Encoding: 8bit

-----BEGIN PGP MESSAGE-----
Charset: windows-1252
Version: GnuPG v2

hQEMA/zpMwW7l2uOAQf/UBMBBMN0PDgs9bSEpXshUBKVXULpBsbg/M8LLnomdgTm
cs0+0HsINcY6+d5wLOTdPIVbK9iYoUzAhkfmjFya8/2Ntj1dd5C7F9tsREcQjJXT
dWtCoG1QPBwp7gBRmcU1nYK0zWga9VMB782XsDJLPFclKMUNS3CmAKy0aZby7sCS
nKGb8P22wk6odCS5NTIxazvLbnLz24MCUgVbaTkksUYuhvlH0PNu+nVvg4nEdoWe
VGG9LX+RknNqSHjrI7bys73w8N/VWuxKBrSgbTmmYIyjoJwA420b5/O7gIujZiI0
WdHjLNWH77OHAp2dtF4ggoZCwBy4WTVcU+1SdwNqBTXI8j1whZk1nf+/SO8b7Sg2
HPgrsMTxnaUf
=isg2
-----END PGP MESSAGE-----

--------------030309080003040107080504
Content-Type: application/octet-stream;
 name="brevet end2end encryption.docx.pgp"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
 filename="brevet end2end encryption.docx.pgp"

hQEMA/zpMwW7l2uOAQf/V2zalW4esYvN2STnekSx7HSREWs8ZC752QLMIJ/6hSTEVcdaMycp
guP4bC8vBeFq5aelofgxjf+ki3Xm1HY4dEPfiWMPpuaZuLcOw9cdZftsb4S6khe99z91aNS7
NyNZNPraqEy3pkzjaROvwsDXoiCm4ZtGaV5TSErCknd8X3IfcH1icMxdFoOBBOhLv/WckxC9
11cWGAXhRDEMC/hvIsknnH5RhEtYJDaEfK56CVmxl3BQT9c7/PRzda8EFeEn6z/i3JUquir3
TEGvXaiOPwt0W+l/w1a7g8lPf6SdEM+DY8xWbEAlpvNfofG4VaPr5Py1I+QVmiOHo/FxZJnO
```

Metadata in encrypted mail.

```
Return-Path: <alice@wanadoo.fr>
Received: from [10.0.101.17] (tui75-2-82-66-245-153.wanadoo.fr. [76.66.245.153])
        by smtp.cegetel.net with ESMTPSA id w125sm2216593wmw.18.2019.05.09.03.26.14
        for <bob@cegetel.net>
        (version=TLSv1/SSLv3 cipher=OTHER);
        Fri, 05 Apr 2019 03:26:15 -0700 (PDT)
Subject: Document confidentiel
References: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
To: Bob <bob@cegetel.net>
From: Alice <alice@wanadoo.fr>
X-Forwarded-Message-Id: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Message-ID: <56F26F45.2080208@wanadoo.fr>
Date: Fri, 05 Apr 2019 11:26:13 +0200
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Thunderbird/60.6.1
MIME-Version: 1.0
In-Reply-To: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Content-Type: multipart/mixed; boundary="------------030309080003040107080504"

This is a multi-part message in MIME format.
--------------030309080003040107080504
Content-Type: text/plain; charset=windows-1252
Content-Transfer-Encoding: 8bit

-----BEGIN PGP MESSAGE-----
Charset: windows-1252
Version: GnuPG v2

hQEMA/zpMwW7l2uOAQf/UBMBBMN0PDgs9bSEpXshUBKVXULpBsbg/M8LLnomdgTm
cs0+0HsINcY6+d5wLOTdPIVbK9iYoUzAhkfmjFya8/2Ntj1dd5C7F9tsREcQjJXT
dWtCoG1QPBwp7gBRmcU1nYK0zWga9VMB782XsDJLPFclKMUNS3CmAKy0aZby7sCS
nKGb8P22wk6odCS5NTIxazvLbnLz24MCUgVbaTkksUYuhvlH0PNu+nVvg4nEdoWe
VGG9LX+RknNqSHjrI7bys73w8N/VWuxKBrSgbTmmYIyjoJwA420b5/O7gIujZiI0
WdHjLNWH77OHAp2dtF4ggoZCwBy4WTVcU+1SdwNqBTXI8j1whZk1nf+/SO8b7Sg2
HPgrsMTxnaUf
=isg2
-----END PGP MESSAGE-----

--------------030309080003040107080504
Content-Type: application/octet-stream;
 name="brevet end2end encryption.docx.pgp"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
 filename="brevet end2end encryption.docx.pgp"

hQEMA/zpMwW7l2uOAQf/V2za1W4cSYvN2SfhekSx7HSREWs8ZC752QLMIJ/6hSTEVcdaMycp
guP4bC8vBeFq5aelofgxjf+ki3Xm1HY4dEPfiWMPpuaZuLcOw9cdZftsb4S6khe99z91aNS7
NyNZNPraqEy3pkzjaROvwsDXoiCm4ZtGaV5TSErCknd8X3IfcH1icMxdFoOBBOhLv/WckxC9
11cWGAXhRDEMC/hvIsknnH5RhEtYJDaEfK56CVmxl3BQT9c7/PRzda8EFeEn6z/i3JUquir3
TEGvXaiOPwt0W+l/w1a7g8lPf6SdEM+DY8xWbEAlpvNfofG4VaPr5Py1I+QVmiOHo/FxZJnO
```

Subject: Document confidentiel
To: Bob <bob@cegetel.net>
From: Alice <alice@wanadoo.fr>

Date: Fri, 05 Apr 2019 11:26:13 +0200

filename="brevet end2end encryption.docx.pgp"

Metadata encryption?

Objective

Encrypt everything except the recipient
No unencrypted metadata

Reasons

- Minimal disclosure
- Leave no trace
- Anonymity with respect to third parties

Challenges

- Encrypt everything
  → identification of the decryption key
- Anonymity
  → pseudonymity & unlinkability

Anonymity: pseudonymity is easy.

Pseudonymity

- Never disclose more than a “pseudonym” to third parties (i.e. the server)
- Typically a public key

Why?

- The server does not need identification elements
- Only contacts/users do

But…

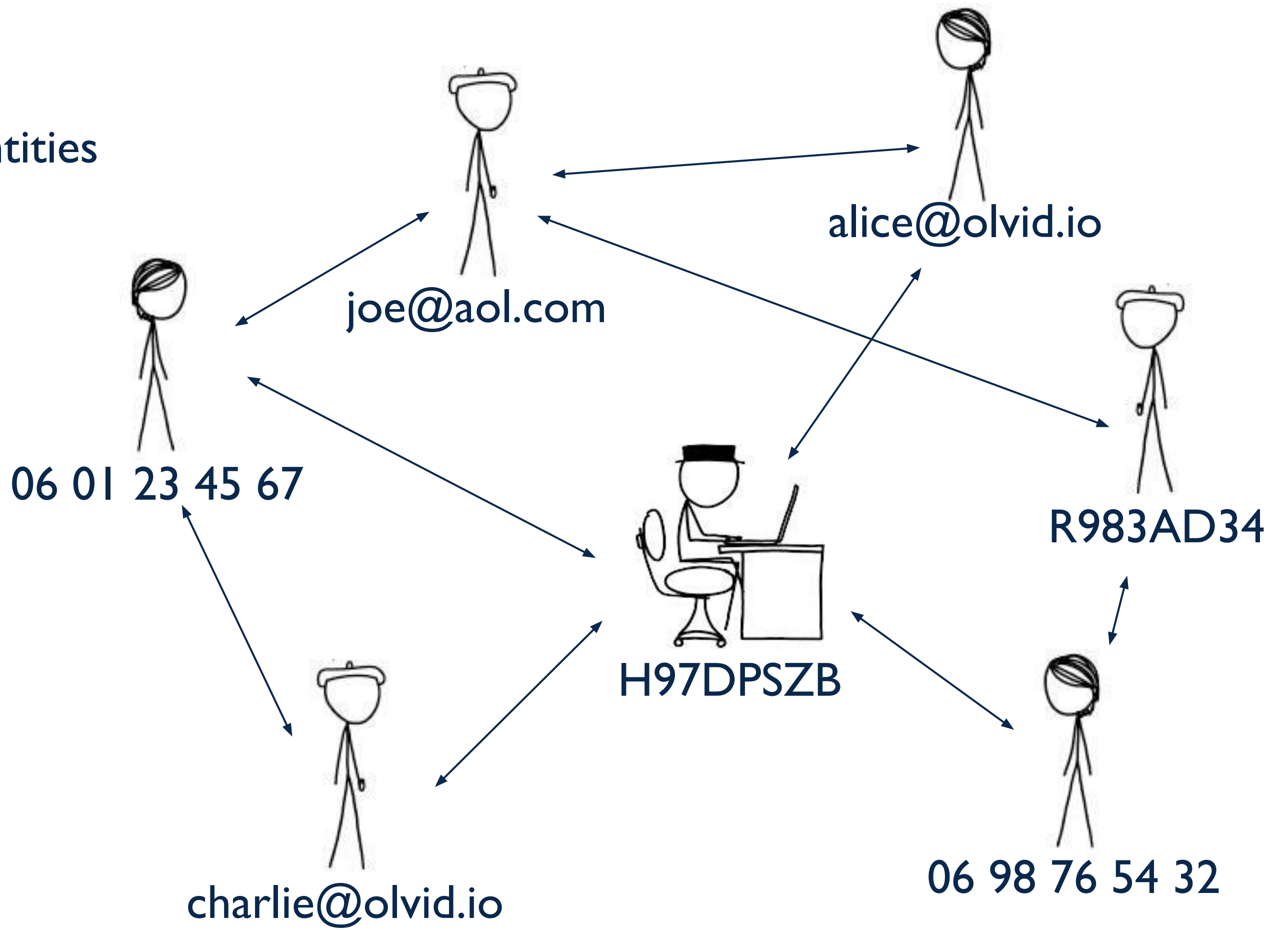
- Centralized key distribution requires an identification element

Example of Threema:

- Each key is associated to a **Threema Id** like H97DPSZB
- Attaching identification elements to it is optional, but **possible/encouraged**
- Most Threema users disclose identification elements so their friends can find them
  → possible to build a social graph and **identify remaining pseudonyms**

The servers knows:

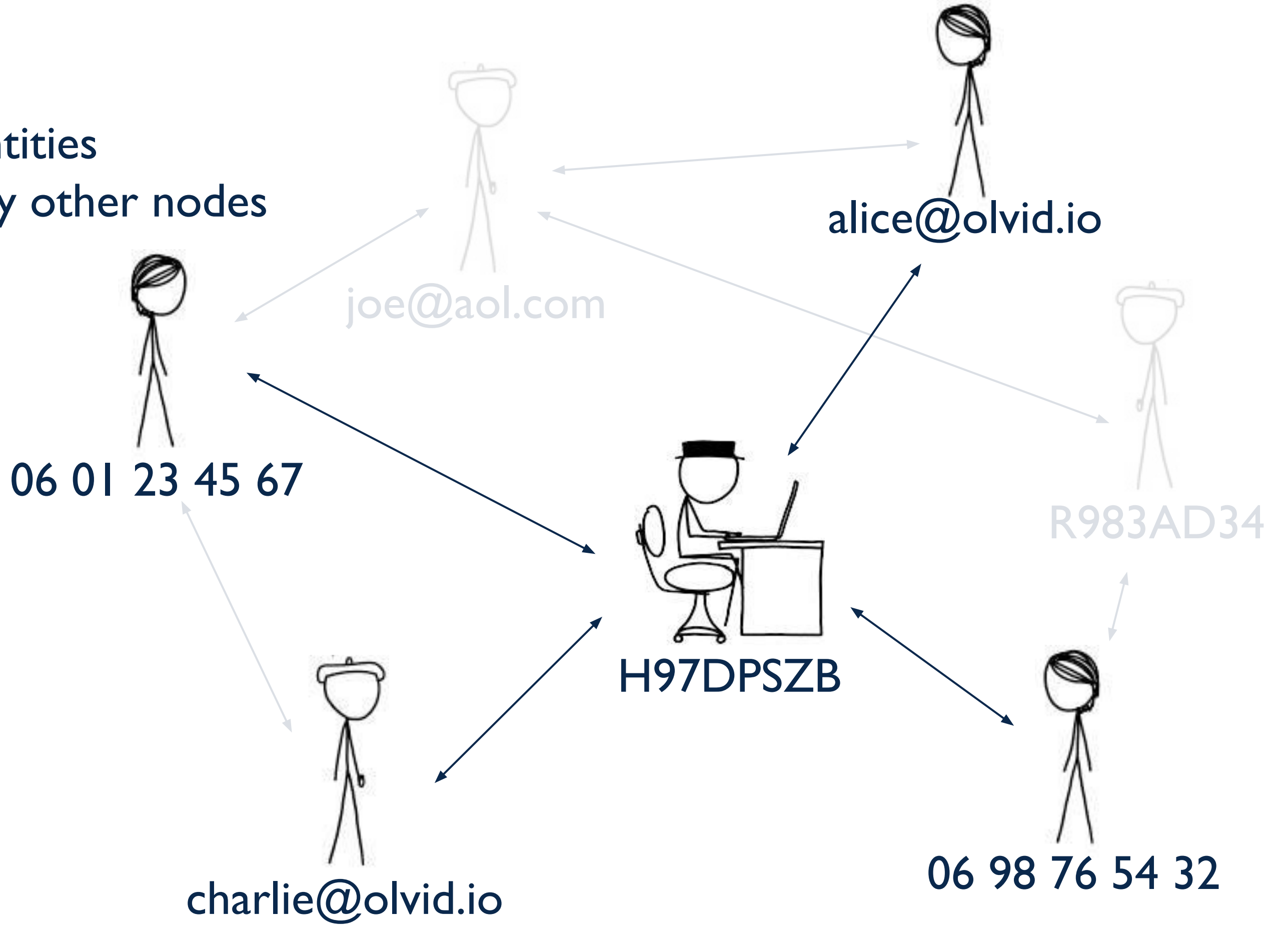
- user links
- a majority of identities



The servers knows:

- user links
- a majority of identities

  → easy to identify other nodes



Anonymity: pseudonymity is easy.

Pseudonymity

- Never disclose more than a "pseudonym" to third parties (i.e. the server)
- Typically a public key

Why?

- The server does not need identification elements
- Only contacts/users do

But…

- Centralized key distribution requires an identification element

Example of Threema:

- Each key is ass
- Attaching ident **ncouraged**
- Most Threema nds can find them
  → possible to build a social graph and **identify remaining pseudonyms**

Pseudonymity cannot be optional

Everyone is pseudonymous, or no one is

Anonymity: unlinkability is hard.

Unlinkability

Impossibility to:

- Link two pseudonyms
- Determine pseudonyms that are “related”

Why?

- Best possible anonymity
- Impossible to determine number of contacts, etc.

But…

- Many elements can establish a link: IP address, push notifications, timings, etc.

Unlinkability of:

- Pseudonyms in a discussion group → **impossible** with statistical analysis of timings
- Two pseudonyms on the same device → **impossible** with push notifications
- Two pseudonyms exchanging messages → requires fully anonymous sending

Anonymity: unlinkability is hard.

Unlinkability

Impossibility to:

- Link two pseudonyms
- Determine pseudonyms that are "related"

Why?

- Best possible anonymity
- Impossible to determine number of contacts, etc.

But…

- Many elements can establish a link: IP address, push notifications, timings, etc.

Unlinkability of:

- Pseudonyms in a
- Two pseudonym
- Two pseudonym

al analysis of timings

notifications

ous sending

Unlinkability requires

- Proxy or Tor network
- Avoiding any group
- Having mostly one way discussions

Push notifications.

Required for **instantaneity** and **user experience**

- Challenging to implement: iOS and Android expect cleartext content
- Security risk: one more server/adversary to consider

What information do Apple & Google need?

Push notifications.

Required for **instantaneity** and **user experience**

- Challenging to implement: iOS and Android expect cleartext content
- Security risk: one more server/adversary to consider

What information do Apple & Google need?

Almost nothing

- A push notification **token** given by the OS
  → allows Apple/Google to identify a user
- But a **single token** per App per device

But also…

- A **random identifier** to handle multiple pseudonyms on the same device
- Apple/Google and the server can link them

Apple/Google should not be able to link a pseudonym to an identity
→ they must never learn the user's pseudonym/public key

Key takeaways.

Key takeaways.

- Having the **security of a closed-door meeting** in the digital world is not straightforward
- There are **many aspects to consider** when discussing messaging security
- **Key distribution** remains the main **security risk** as no “one-size fits all” method exists
- **Data encryption**, though tricky, is something we know how to do
- **Anonymity** is a difficult topic but **true pseudonymity** would already be a real progress

Merci.