

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/), except the Matrix and Element logos, which are copyright their owners, and the cartoon characters, which are used without permission.

Source: codeberg.org/andybalaam



How Encryption Works in Matrix

Andy Balaam
artificialworlds.net/blog



Funded by Element



element
creators of Matrix

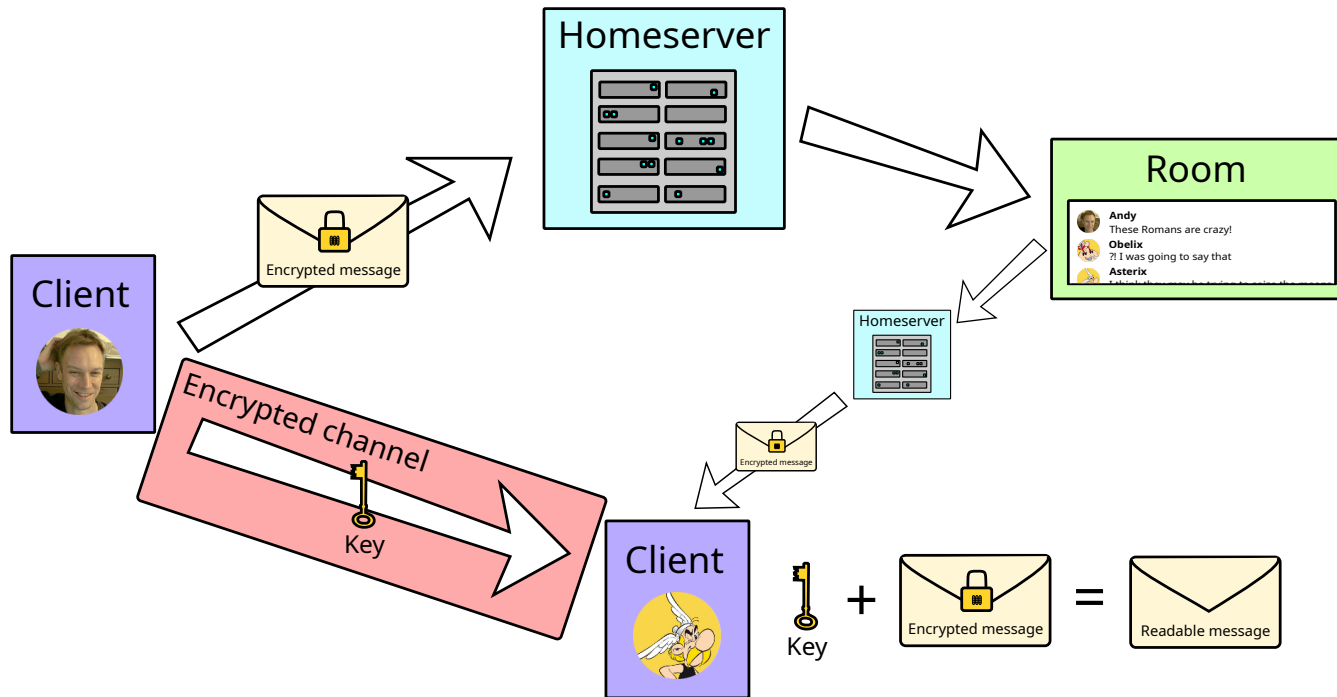
The Crypto team is employed full-time by Element

(to fix your "unable to decrypt" errors)

Contents

- What is encryption?
- What is Matrix?
- Group chat: Megolm
- Device-to-device: Olm

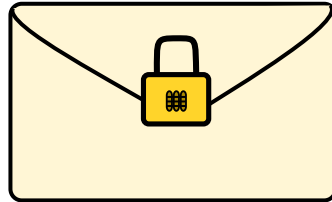
Overview



What is encryption?

Two things:

- Hide message content
- Know who I'm talking to



What is Matrix?

Store data with live updates:

- Drop messages into a room
- Get notified when new messages appear
- Encrypted messages are just like other messages

[matrix]

Matrix encryption is the best

- Server can't read or modify your messages
- Use multiple devices
- Control your own data on the server
- Talk to other people who control their own data
- Recover after losing your phone
- Identity resets are visible
- New members can see history
- (Coming soon) read things sent while logged out

Group chat: Megolm

Room



Andy

These Romans are crazy!



Obelix

?! I was going to say that



Asterix

I think they may be trying to seize the means

Yay, JSON!

Make the room encrypted:

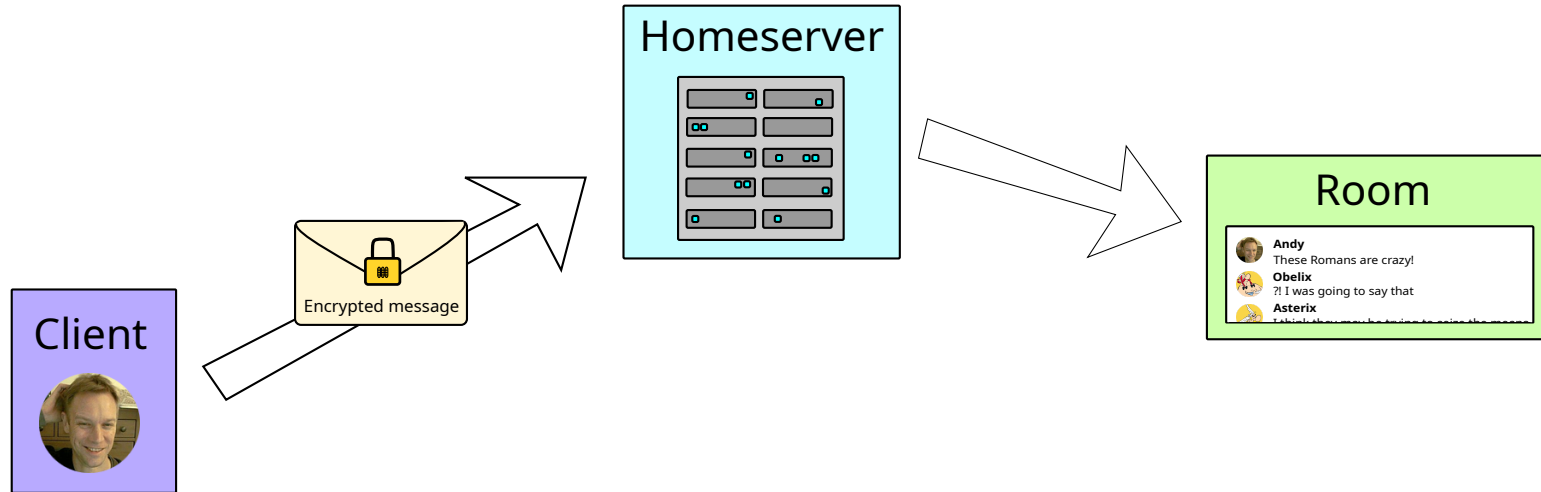
```
{
  "content": {
    "algorithm": "m.megolm.v1.aes-sha2",
    "rotation_period_ms": 604800000,
    "rotation_period_msgs": 100
  },
  "state_key": "",
  "type": "m.room.encryption",
  ...
}
```

Group chat: Megolm

- Sender encrypts each message once
- And sends the key to all recipients

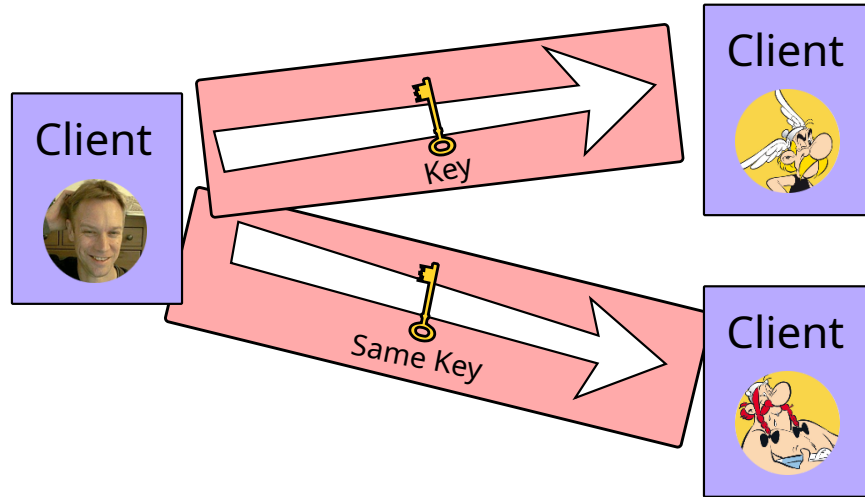
Group chat: Megolm

- Sender encrypts each message once



Group chat: Megolm

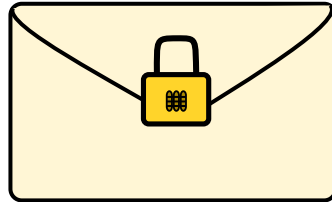
- And sends the key to all recipients



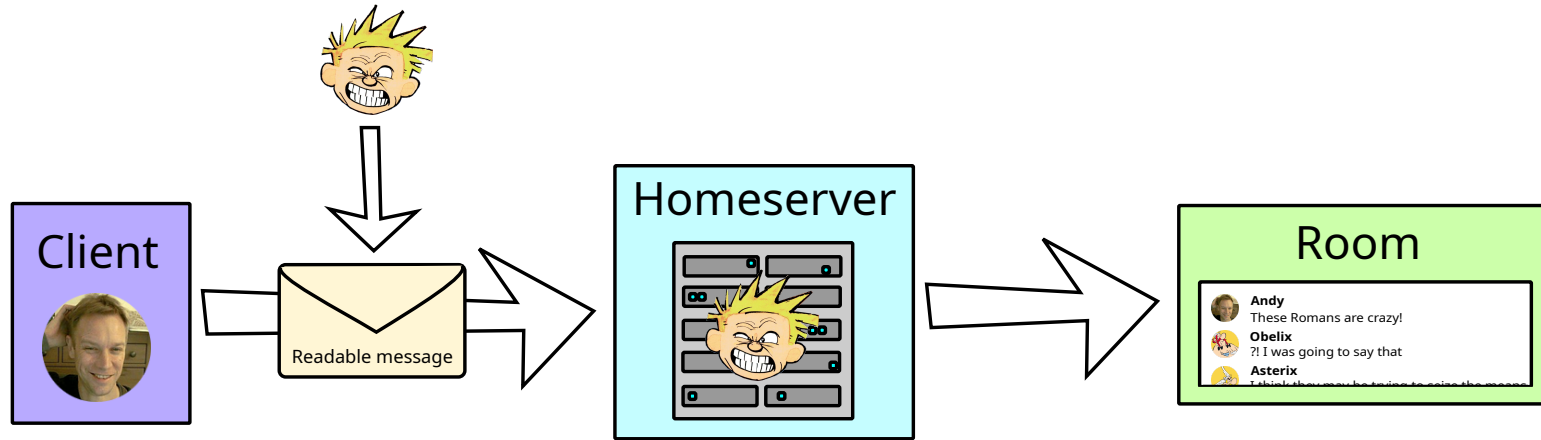
(reminder)

Encryption is:

- Hide message content
- Know who I'm talking to

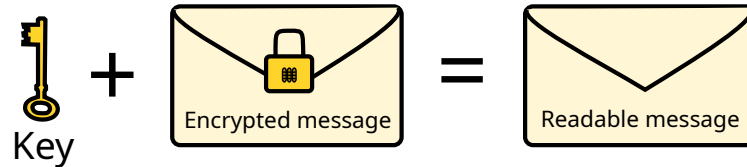
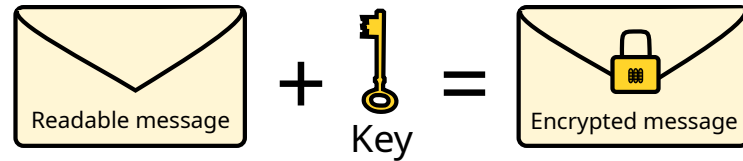


Hide message content



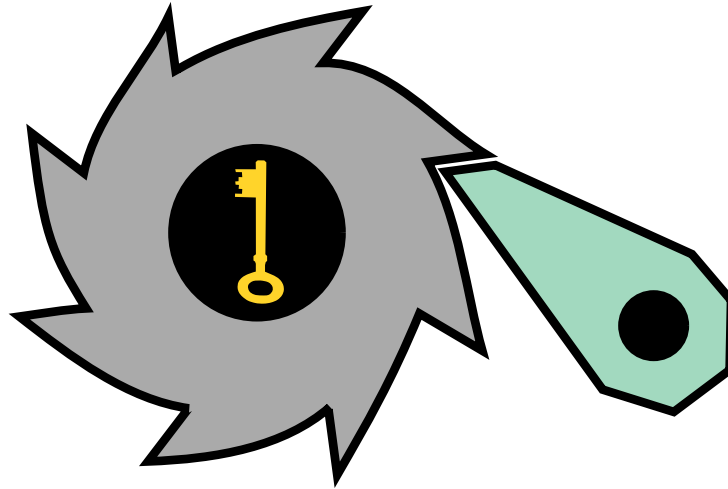
Hide message content

- We use a shared secret
- Encrypting and decrypting use the same key
- All recipients use the same key



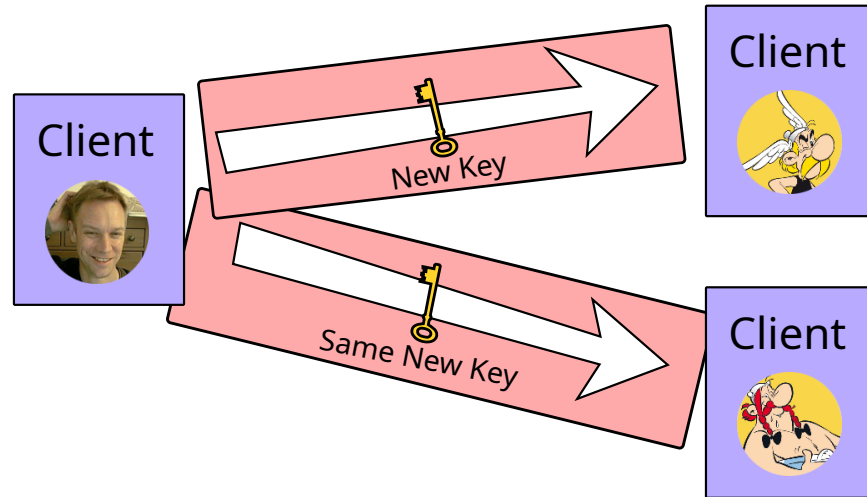
Hide message content

- "Ratchet" the key every message

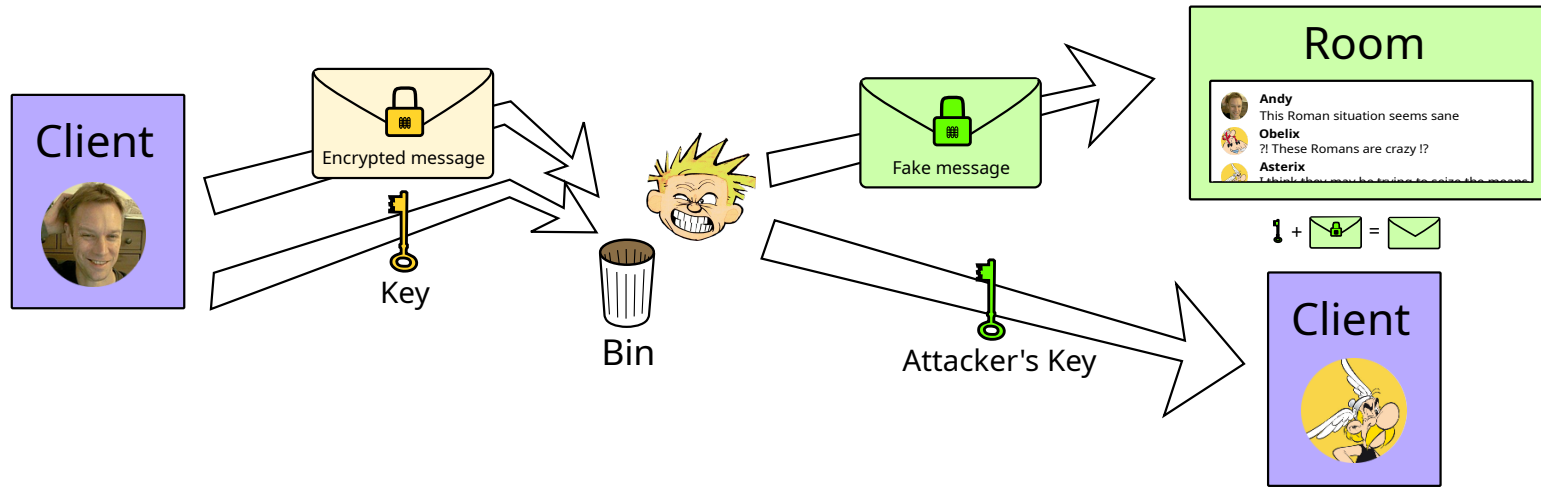


Hide message content

- Replace the key completely every so often



Know who I'm talking to



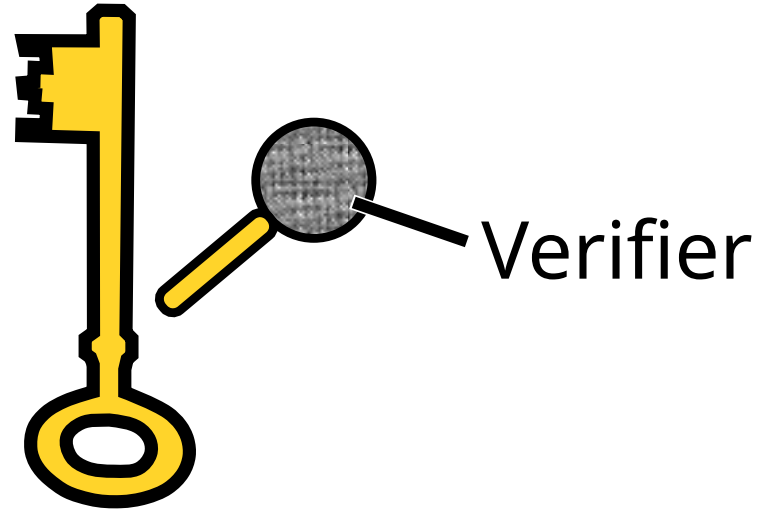
Know who I'm talking to

- We sign messages when we encrypt



Know who I'm talking to

- And include a verifier with each key



Know who I'm talking to

- So we can verify who sent the message



Awesome terminology

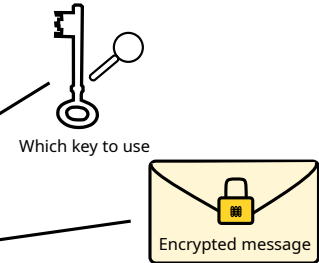
"Message key", also known as:

- Session
- Sender key
- Room key



But you came for the JSON

```
{
  "type": "m.room.encrypted",
  "content": {
    "algorithm": "m.megolm.v1.aes-sha2",
    "sender_key": "(deprecated)",
    "device_id": "(deprecated)",
    "session_id": "X3lUlvLEL...",
    "ciphertext": "AwgAEnACg..."
  },
  "room_id": ...
}
```



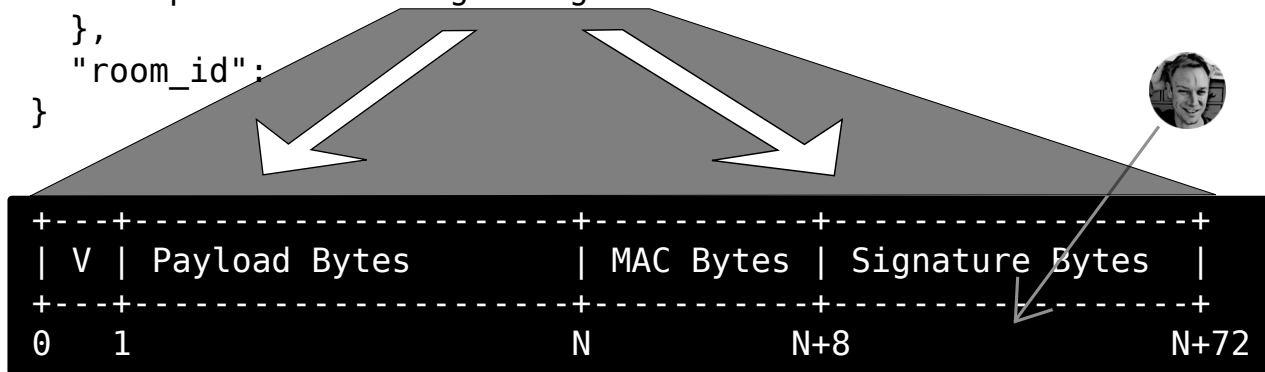
Which key to use

Encrypted message

The diagram illustrates the relationship between the JSON fields and the cryptographic components. A key icon is positioned above the 'session_id' field, with an arrow pointing to it. A yellow envelope icon with a lock is positioned to the right of the 'ciphertext' field, with an arrow pointing to it. The text 'Which key to use' is placed near the key icon, and 'Encrypted message' is placed near the envelope icon.

But you came for the bytes

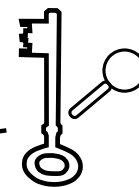
```
{  
  "type": "m.room.encrypted",  
  "content": {  
    "algorithm": "m.megolm.v1.aes-sha2",  
    "sender_key": "(deprecated)",  
    "device_id": "(deprecated)",  
    "session_id": "X3lUlvLEL...",  
    "ciphertext": "AwgAEnACg..."  
  },  
  "room_id":  
}
```



But you came for the JSON

Sending a key:

```
{
  "content": {
    "algorithm": "m.megolm.v1.aes-sha2",
    "room_id": "!roomid:s.co",
    "session_id": "X3lUlVLEL...",
    "session_key": "AgAAADxK..."
  },
  "type": "m.room_key"
}
```



Which key this is...



...but also this **is** the verifier



Key

Group chat: Megolm

Room



Andy

These Romans are crazy!



Obelix

?! I was going to say that



Asterix

I think they may be trying to seize the means

****Pause****

Why are we doing this?

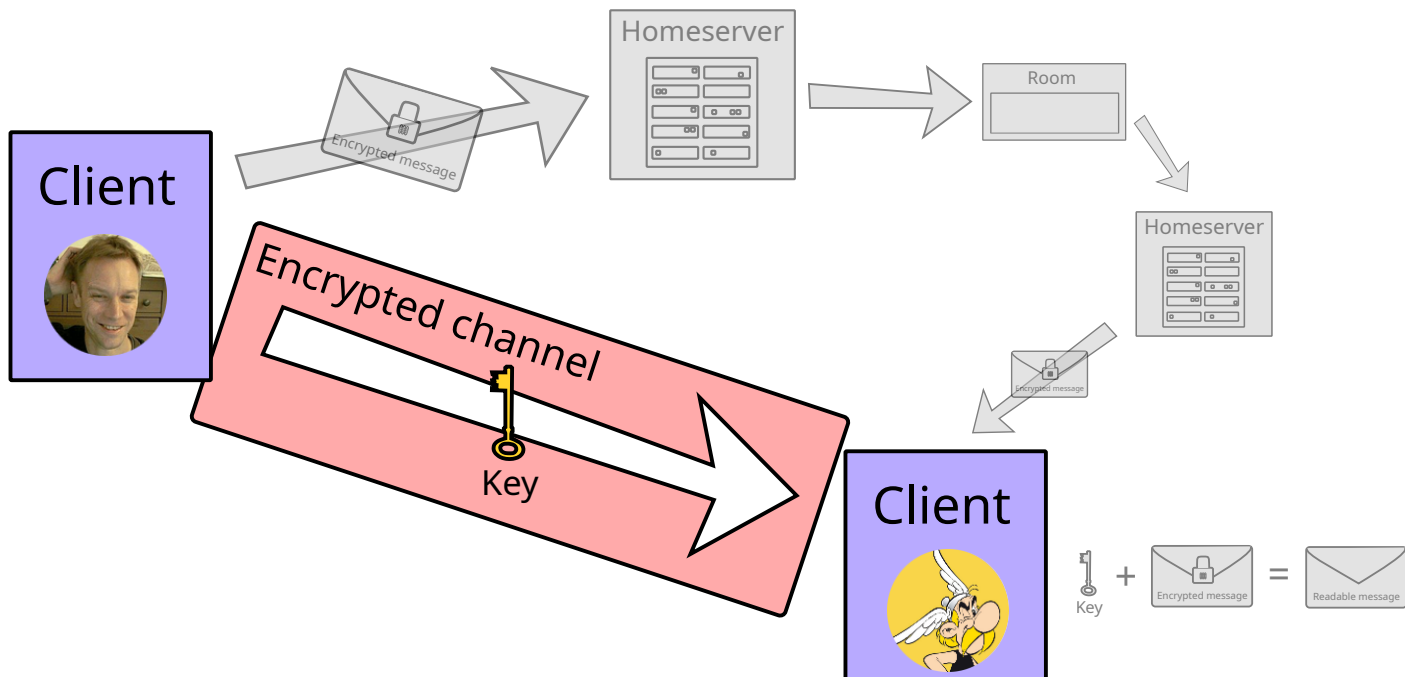
****Pause****

Why are we doing this?

- To resist surveillance capitalism
- To fight corrupt officialdom
- To undermine global spheres of influence

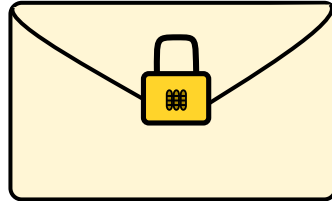
Device-to-device: Olm

But what is this encrypted channel?

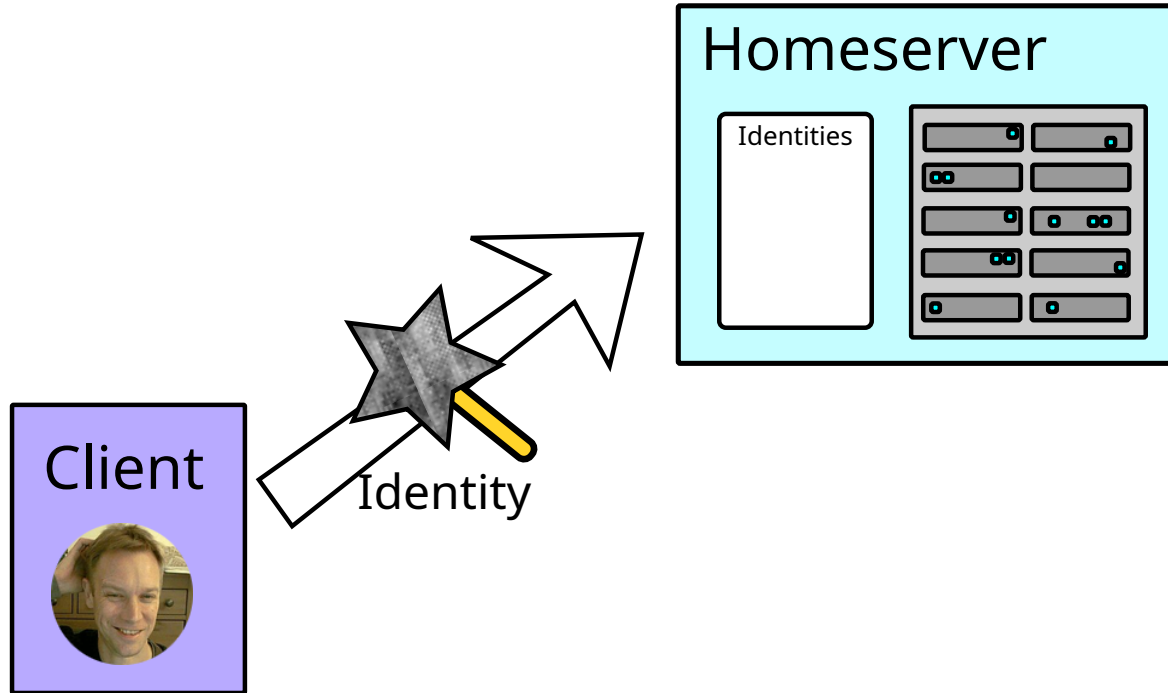


Encryption is...

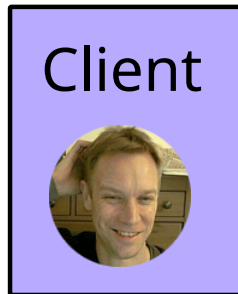
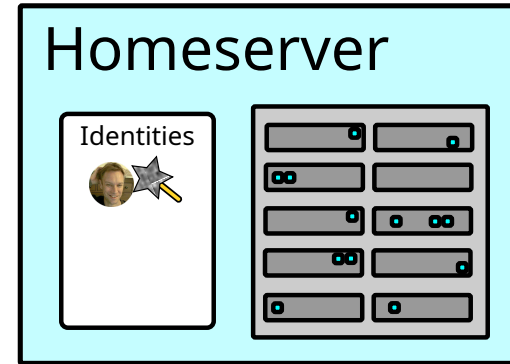
- Hide message content
- Know who I'm talking to



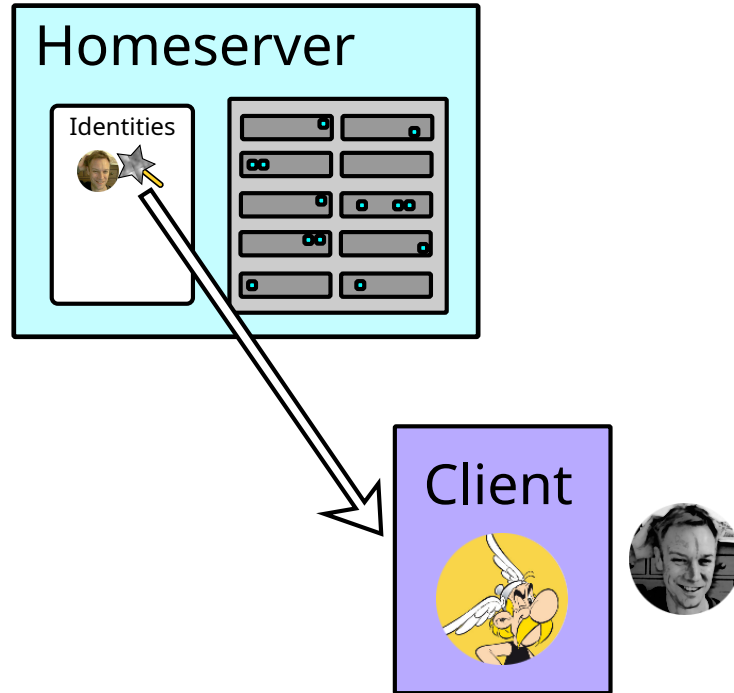
Know who I'm talking to



Know who I'm talking to



Know who I'm talking to



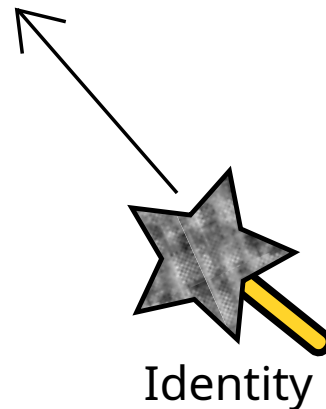
Know who I'm talking to



JSON or it didn't happen

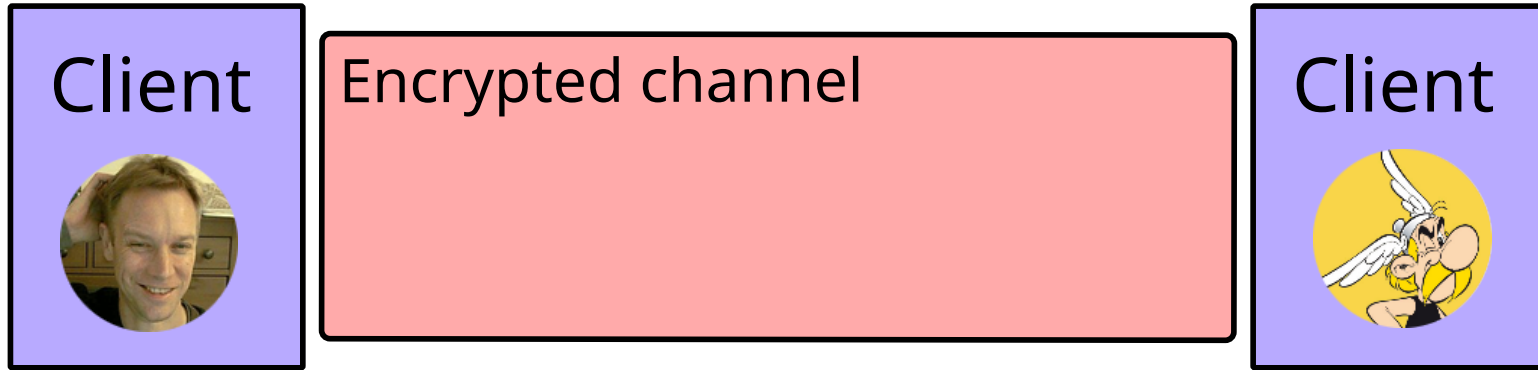
POST `/_matrix/client/v3/keys/device_signing/upload`

```
{
  "master_key": {
    "keys": {
      "ed25519:base64+master+public+key": "AzB80ia..."
    },
    "usage": [
      "master"
    ],
    "user_id": "@andy:example.com"
  },
  ...
}
```

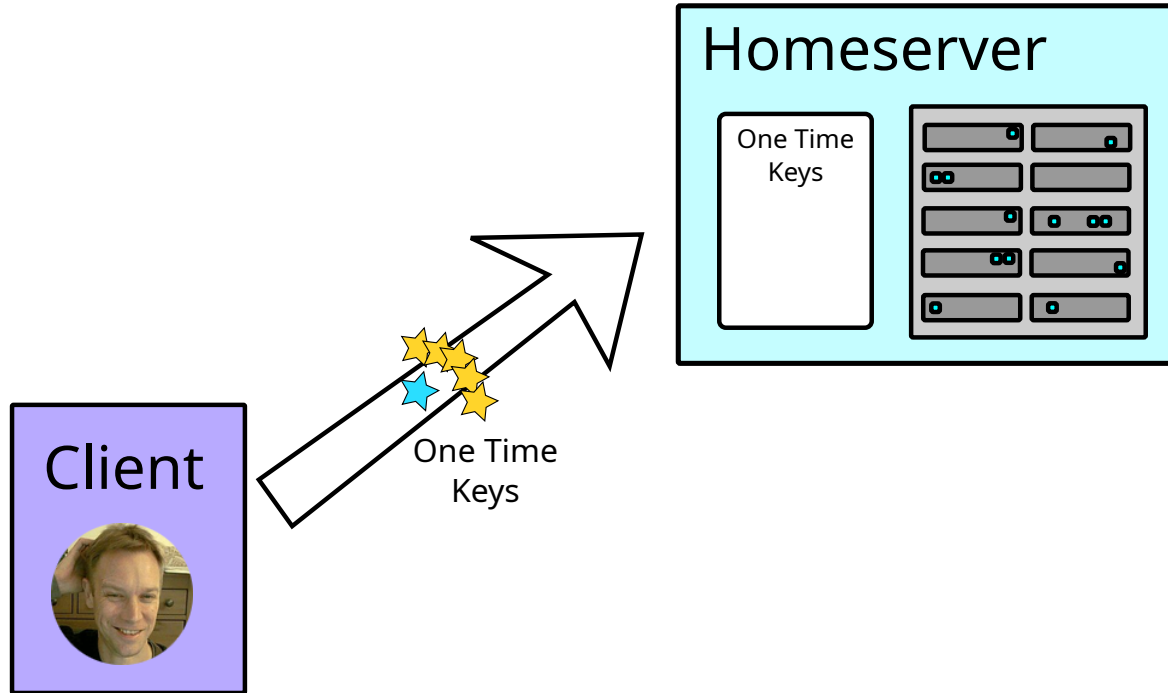


Identity

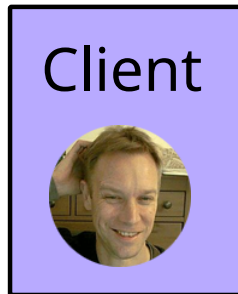
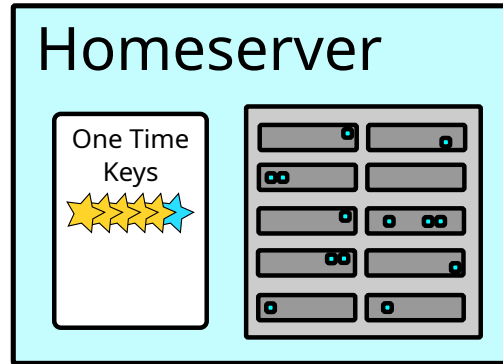
Hide message content



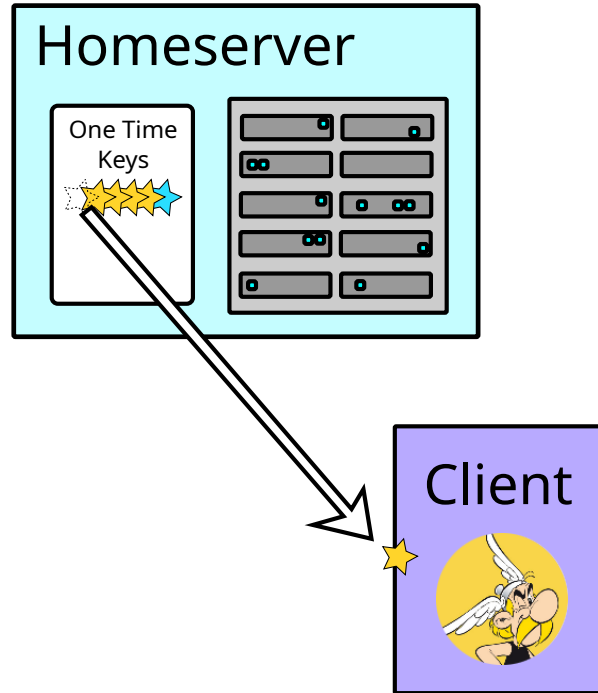
One-time keys



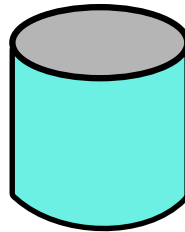
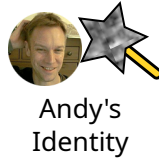
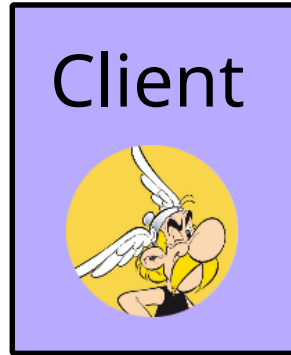
One-time keys



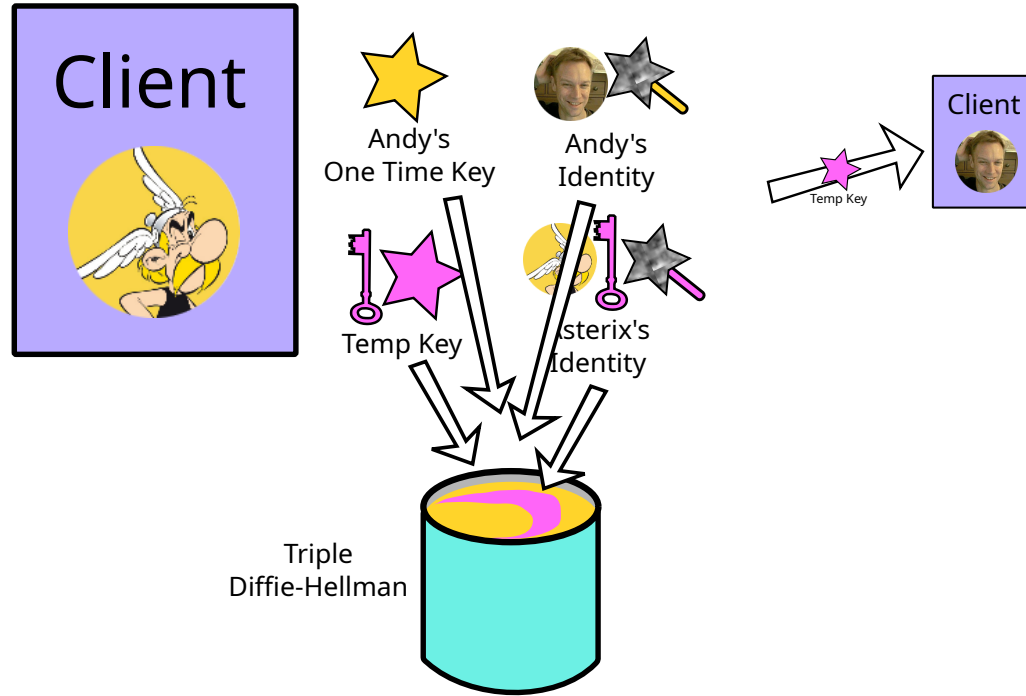
Establish an Olm session



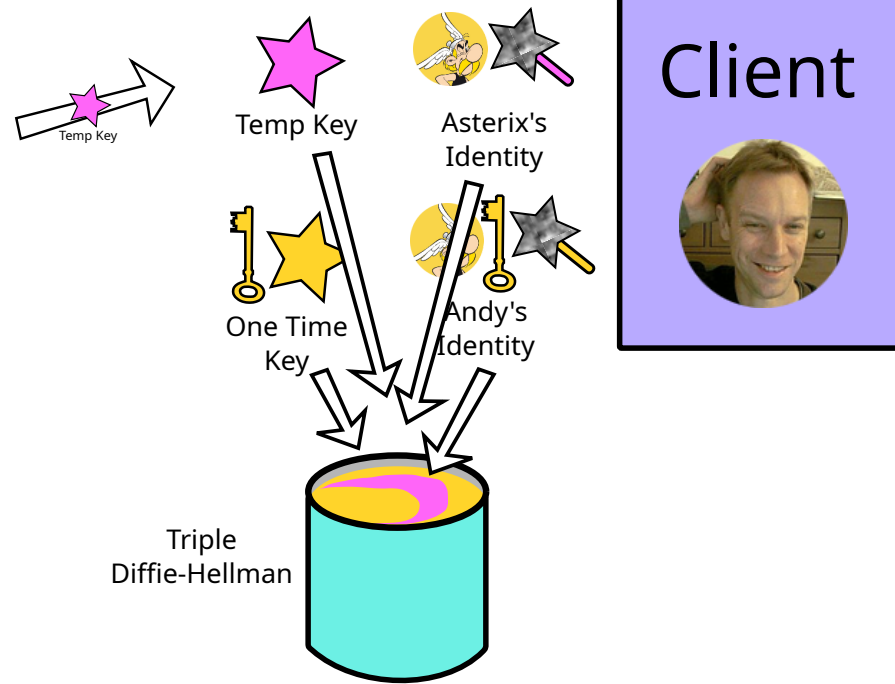
Establish an Olm session



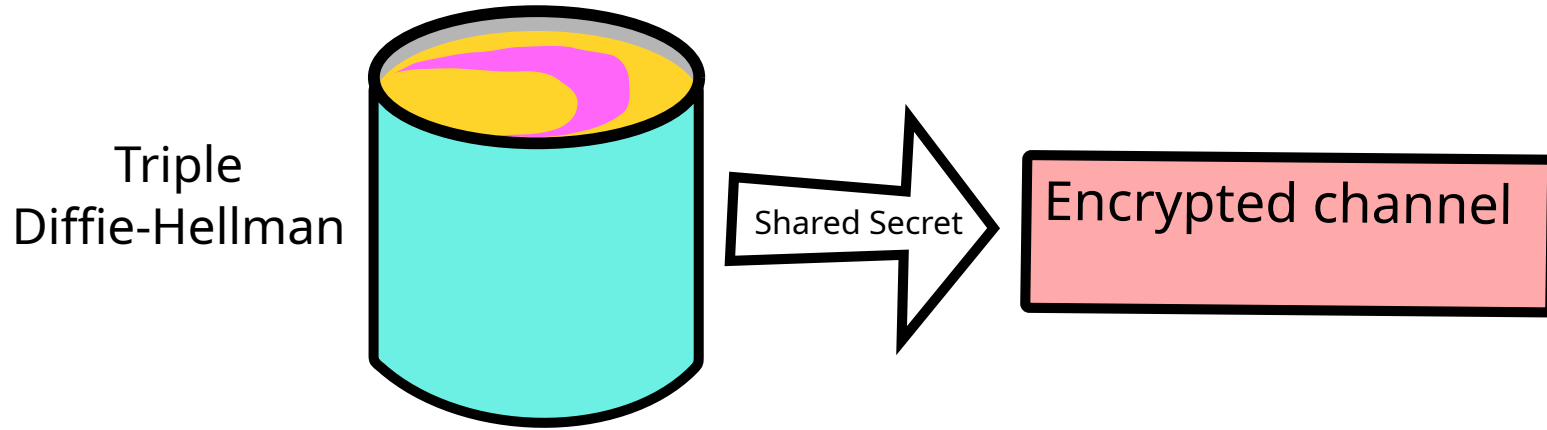
Establish an Olm session



Establish an Olm session



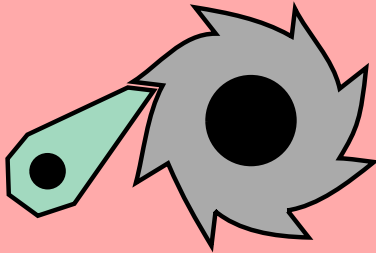
Establish an Olm session



And we do the double ratchet

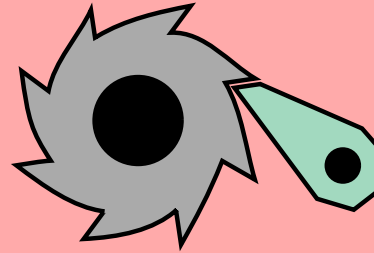
Encrypted channel

Key Ratchet



Forward secrecy:
new keys don't
unlock old messages

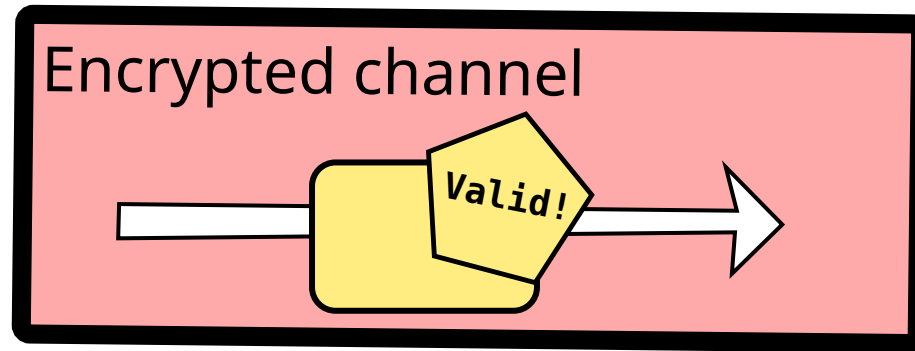
More Diffie-Hellmans



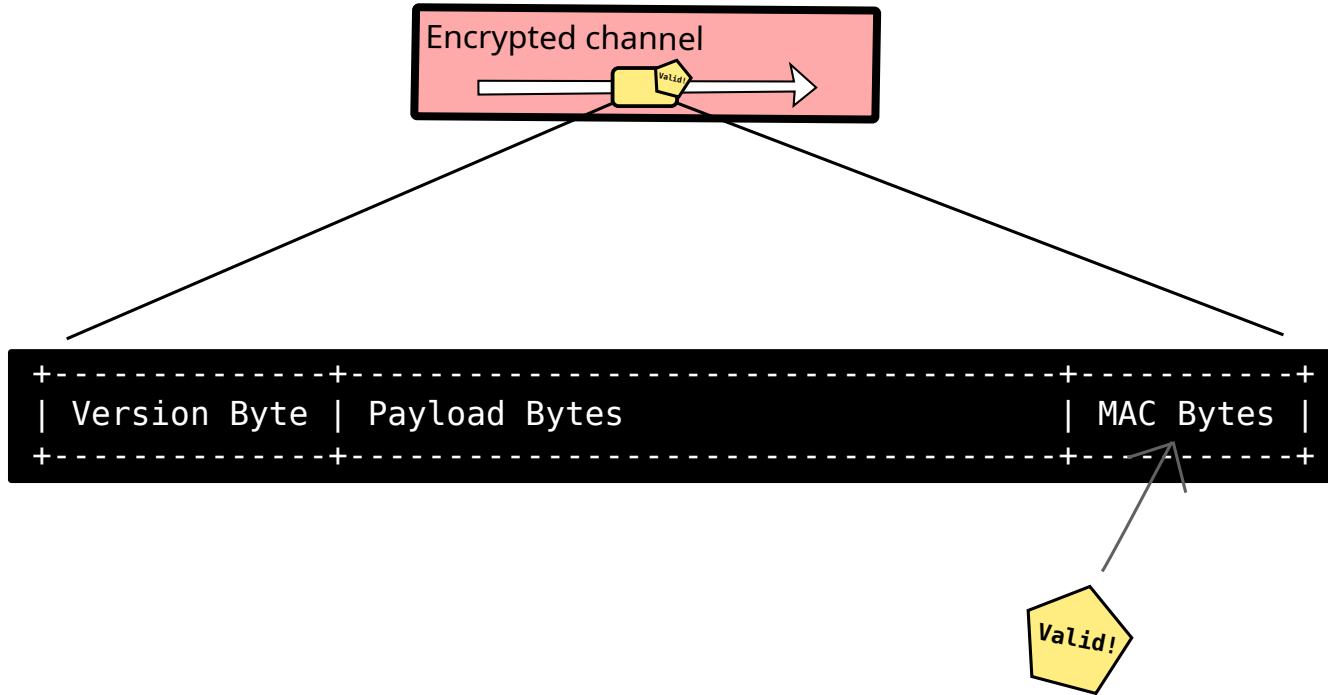
Post-compromise security:
old keys don't
unlock new messages

Olm messages are deniable

- I know you sent it
- No-one else knows I didn't fake it



Olm messages are deniable



Encryption is...

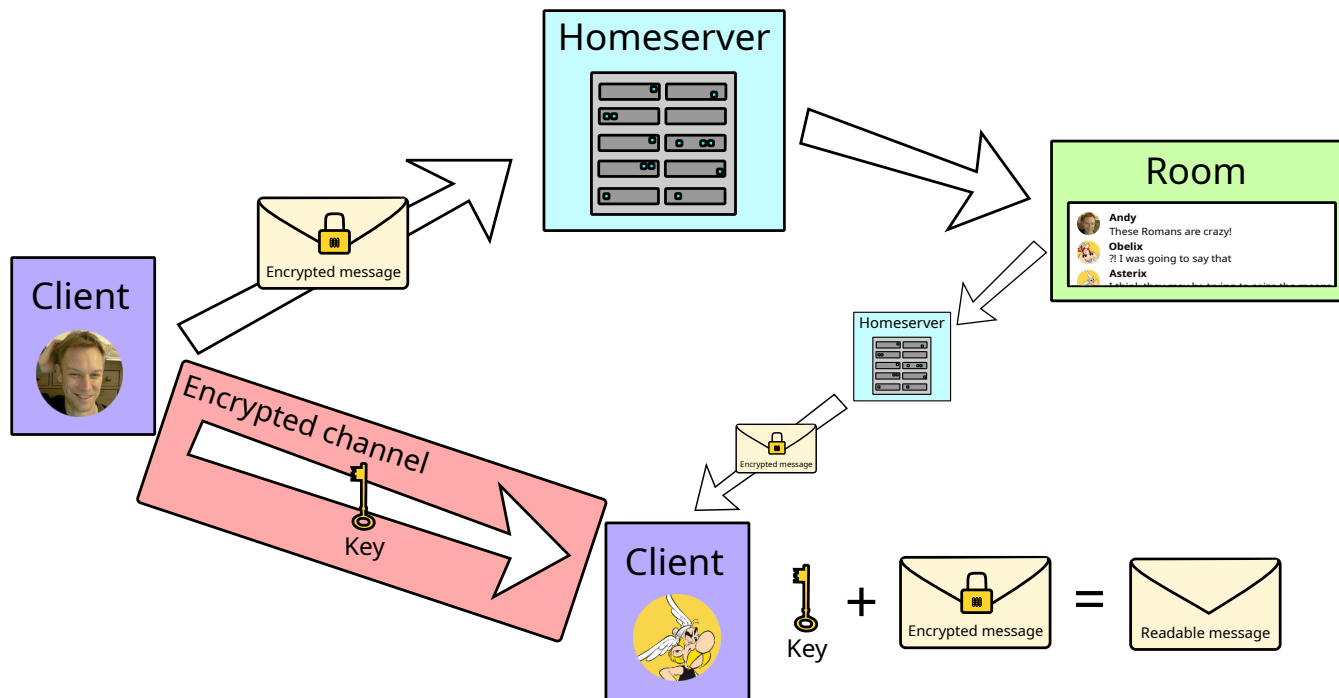
- Hide message content
- Know who I'm talking to

• DENY I SAID IT

Summary

- Olm - establish a secure channel:
 - using public identity and a one-time-key
 - The double ratchet keeps it secret
- Megolm - messages in the room:
 - Send the message key via the secure channel
 - Encrypt my message once and send to the room

Summary



More info

Spec:

spec.matrix.org/v1.19/client-server-api/#end-to-end-encryption
spec.matrix.org/v1.19/olm-megolm/megolm/
spec.matrix.org/v1.19/olm-megolm/olm/

More from me:

artificialworlds.net/blog

This work is funded by Element:



element
creators of Matrix

No AI was used in this presentation.
Some cartoon characters were shamelessly reproduced without permission: contact me to request their removal.

Things we didn't talk about

- Who is in the room?
- Changing identity
- Chains of cross-signing
- What the ratchets actually are
- What Triple Diffie-Helman is
- Key storage, history sharing
- Recovery key
- Dehydrated devices