
Winfried
Tilanus
privacy
e-health
information
patient
identity
data
analysis
system
practitioner
healthcare
treatment
relation
relationship
definition
research
interventions
control
might
example
people
patients
automated
directive
panopticon
bubble
relations
processes
patterns
time
also
much
one
clear
legal
becomes
possible
lot
potential
process
expect
used
level
done
makes
always
context
expectations
health
records
certain
results
forced
autonomy
positioning
many
often
like
MEDDEV
WGBO
surveillance
ethical
social
society
environment
part
must
care
tailored
inside
within
safe
medical
systems
public
relationships
become
third
new
two
tailoring
approach
user
digitalization
ability
Also
demands
limited
way
edge
problem
party
individual
panopticons
using
take
concept
different
large
software
case
behaviour
behavior
influence
something
power
kind
quite
understand
question
normative
without
informed
better
evidence
problems
customers
help
want
digital
limits
created
consent
first
less
even
big
organizations
events
try
see
make
implications
scale
may
look
active
able
amount
online
techniques
paper
quality
somebody
intervention
need
based
second
use

OSCAL'19

The hairy issue of end-to-end encrypted instant messaging

Winfried Tilanus
xmpp:winfried@tilanus.com
mailto:winfried@tilanus.com
OSCAL'19 2019-05-18

About me

- 20y experience with messaging in care
- Member of XMPP Standards Foundation
  - Seen 4 standards for end-to-end-encryption
- This is my personal reflection, not an opinion of the XSF

This talk:

- Threat model issues with E2EE
- Technical issues with E2EE

Encrypting Messaging

- Connection encryption:
  - Decrypted and re-encrypted at servers
  - Servers process messages in plaintext
  - Servers need routing information
- E2E encryption:
  - Decrypted at endpoints
  - Servers still need routing information

Added value of E2EE?

- Not decrypted at hops
- Useful when you don’t trust your servers
- But you still have to trust your servers with metadata for routing

Attack scenarios

- Secret service performing large scale monitoring
- Big tech company analysing messages for advertisement

Secret service attack

- Someone comes to attention of secret service
- Secret service uses metadata to analyse network and pinpoints persons who need closer observation
- Option 1: eavesdrop communication of that person
- Option 2: hack devices of that person

Secret service attack (2)

- E2EE does not protect against reconnaissance by secret service
- E2EE only slightly raises the bar for a full attack by a secret service, they will be hacking sooner
- Hacking is attractive anyway: easy and much more information

Big company attack

- Use messaging metadata to chart social circles people move in
- Use statistics to assume properties of individuals
- Sell advertisement space to other companies based on the assumed properties

Big company attack (2)

- Done solely with metadata
- E2EE does not protect at all

Attack model issue

E2EE does not protect against the attacks that are commonly named to justify the need for it.

But E2EE can be effective

- When a server operator wants to protect itself to data requests from law enforcement: you can’t hand over data you don’t have.
- When a server operator *partly* uses (cloud)infrastructure the operator does not trust itself. It can shield content from those operators.

What is E2EE for?

E2EE without metadata protection does not protect end users. It protects server operators

Part 2: technical issues

- Message storage and forward
- Audit trails & Archiving
- Group chats & Multiple devices
- Key verification

Not solved well right now, but might be solvable

Store and forward

- Perfect forward secrecy depends on fast rotation of keys and disposal of them
- Store and forward depends on stability of keys
- This is always a trade-off

Audit trails and archiving

- In some settings, like human rights activism, security means “no traces at all”
  - Solvable with perfect forward secrecy and in-memory processing
- In some settings, like healthcare and legal, security means “audit trail”
  - Audit trails and E2EE don’t mix very well, it is more or less creating a wiretap.

Creating an archive

- At endpoints
  - Re-encryption for archiving or storable E2EE?
  - Needs message signing to avoid tampering
- At server
  - Only makes sense when messages stay encrypted at the server and the encrypted messages are also send to a trusted archive

One-to-many messages

- Examples: Chat groups and keeping multiple devices (phone, desktop) in sync
- Sharing one secret key over many endpoints is a security risk
- Encrypting and sending messages from one endpoint to each endpoint is not scalable

One-to-many messages

- Can we distribute to multiple keys at the server without decrypting at the server?
- Yes we can!
  - Diffie-Hellman allows to create a ‘group-key’
  - Possible to add or remove keys
- No reviewed & operational standard yet
  - IETF has draft for “Message Level Security”

Key verification

- Leap of faith?
  - Not as bad as it sounds, but still bad
- Web of trust?
  - Superficial checks
- Trusted third party?
  - Didn’t we want get arid of trusted third parties with E2EE?
- Verification in person?
  - Limited use cases and not scalable

Key verification (2)

- How to handle changes in keys?
- How to handle multiple devices?

Fail at key verification and your E2EE is useless

Conclusion

Bad encryption is better then no encryption

or

Bad encryption is a false sense of security

Credits

- Thanks to David Cridland for bringing in new developments in crypto
  - Read his blog “crypto show and tell” at: https://dev.to/dwd/
- Message Level Security:
  - https://datatracker.ietf.org/wg/mls/documents/