

# Secure Interoperable Instant Messaging

Technical Report for Bundesnetzagentur

23-05-03

Real-World Cryptography Group  
FAU Erlangen-Nürnberg, Germany

Prof. Dr. Paul Rösler

Chair for Network and Data Security  
Ruhr University Bochum, Germany

Prof. Dr. Jörg Schwenk

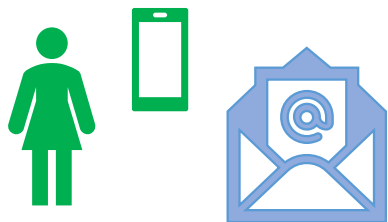
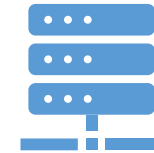
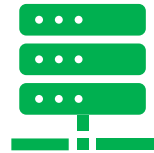
Hackmanit GmbH  
Bochum, Germany

The logo for Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), consisting of the letters 'FAU' in a stylized, blue, outlined font.The logo for Ruhr University Bochum (RUB), consisting of the letters 'RUB' in a bold, white, sans-serif font on a dark blue background.The logo for Hackmanit GmbH, featuring a red 3D cube icon above the word 'HACKMANIT' in a red, sans-serif font.

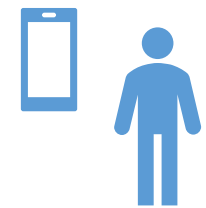
# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

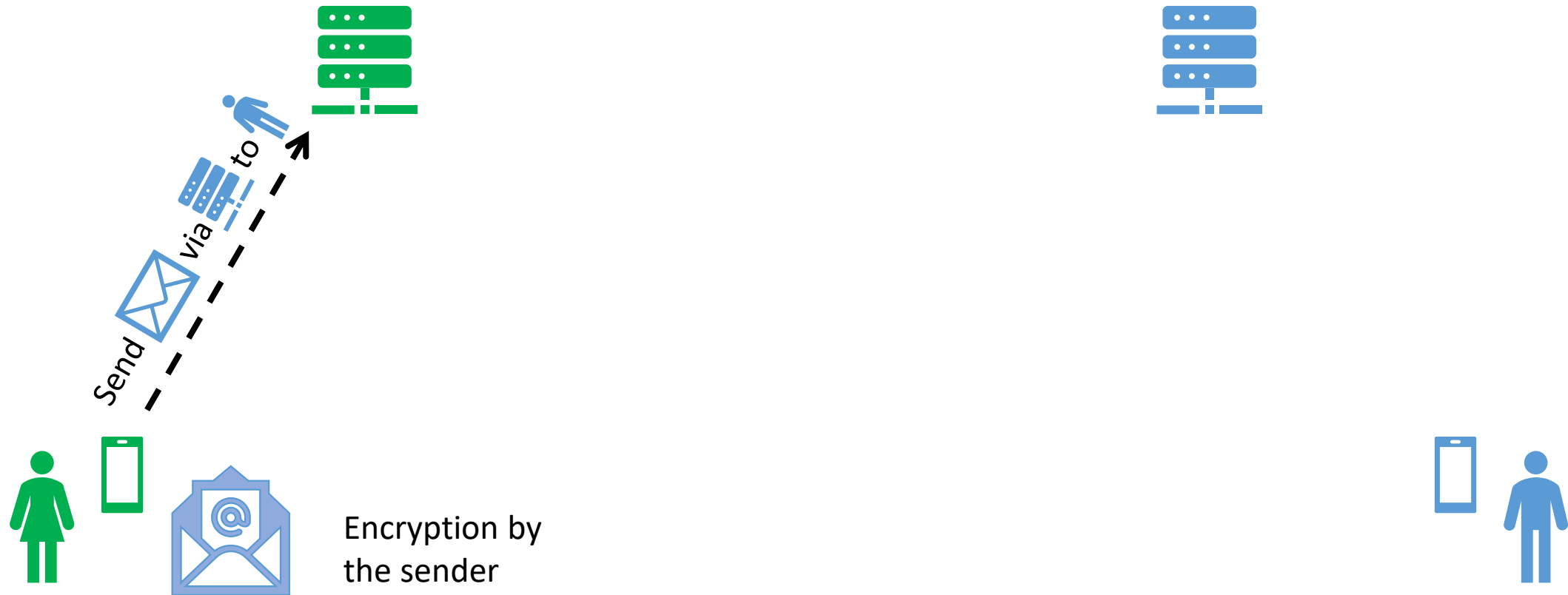
# End-to-End-Encryption



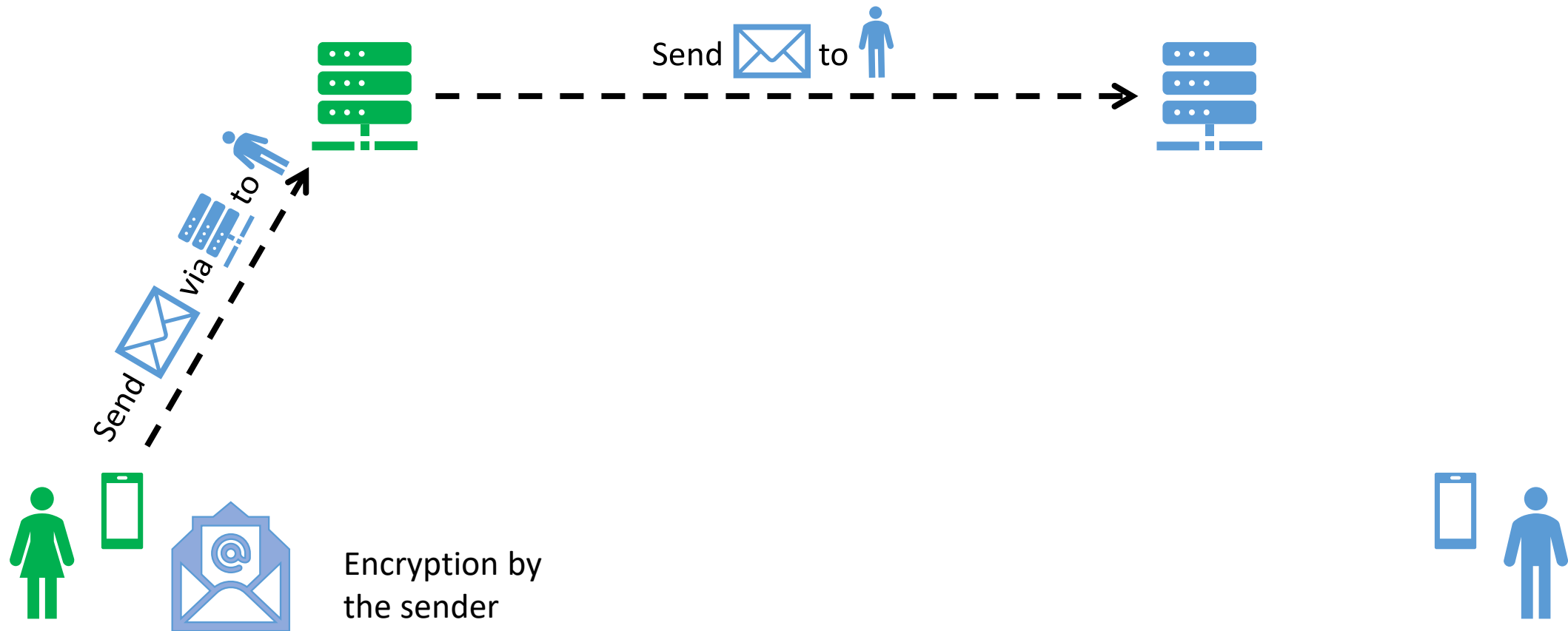
Encryption by  
the sender



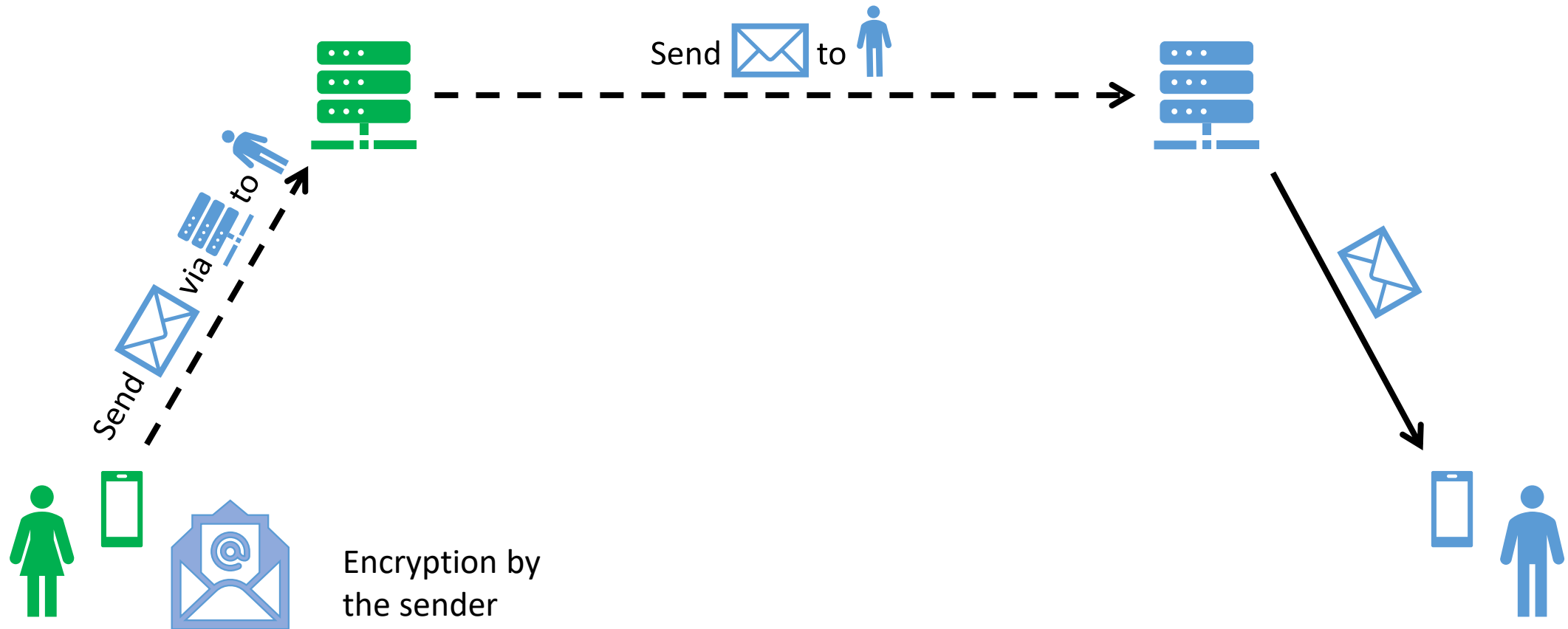
# End-to-End-Encryption



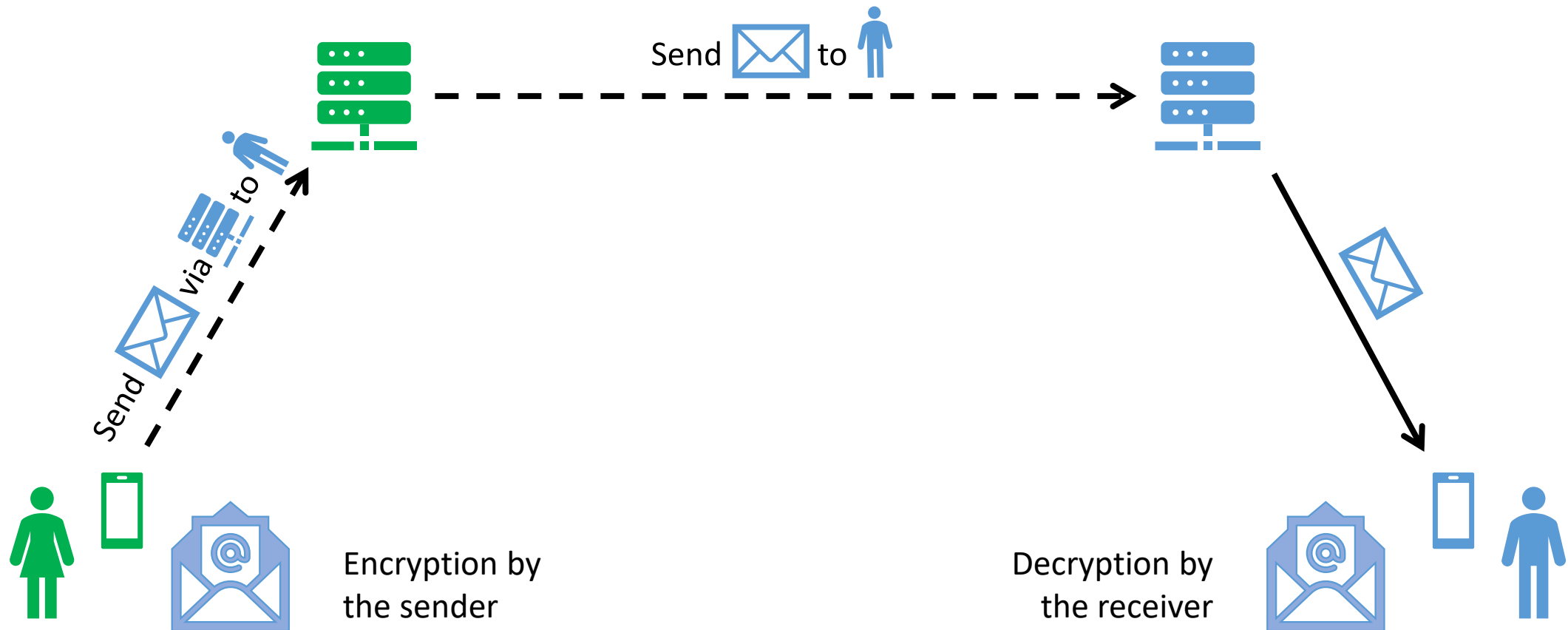
# End-to-End-Encryption



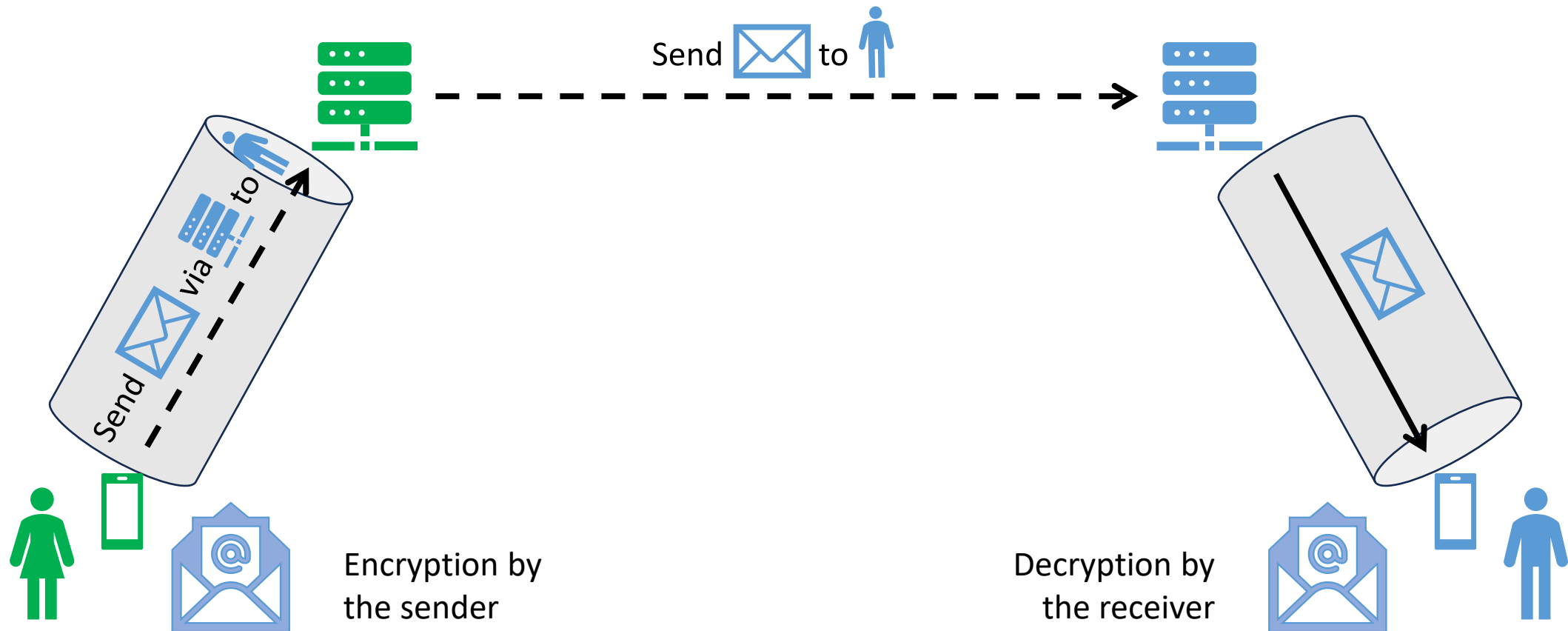
# End-to-End-Encryption



# End-to-End-Encryption

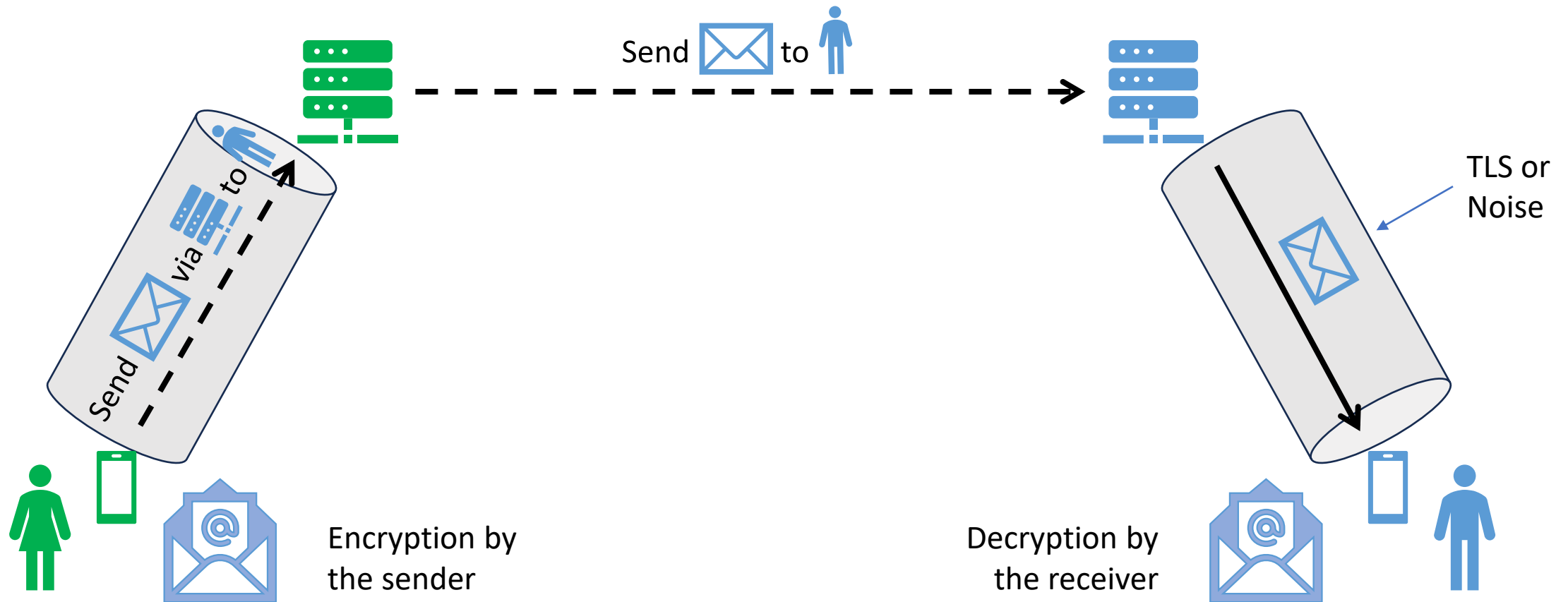


# End-to-End-Encryption





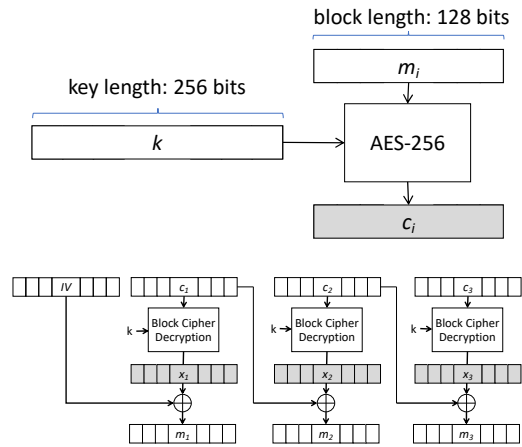
# End-to-End-Encryption



# Agenda

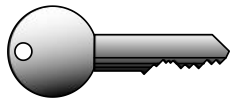
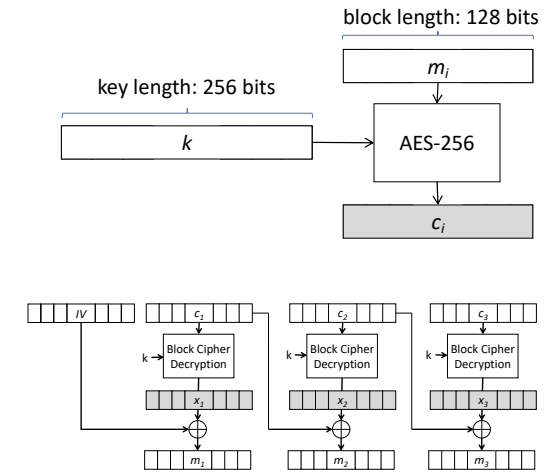
- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# For E2EE we need:



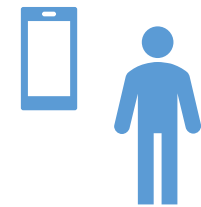
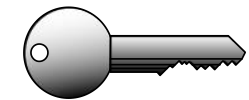
encryption algorithms  
(AES, ChaCha20)

encryption modes  
(authenticated encryption)



symmetric keys

via  
Authenticated Key Agreement  
Ratcheting  
(MLS)



# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# Diffie-Hellman Key Exchange

Public parameters:  $(EC(a, b), P, q)$

*Alice*

*Bob*

$$a \xleftarrow{\$} \mathbb{Z}_q$$
$$\alpha \leftarrow a \cdot P$$

$$\xrightarrow{\alpha}$$

$$b \xleftarrow{\$} \mathbb{Z}_q$$
$$\beta \leftarrow b \cdot P$$

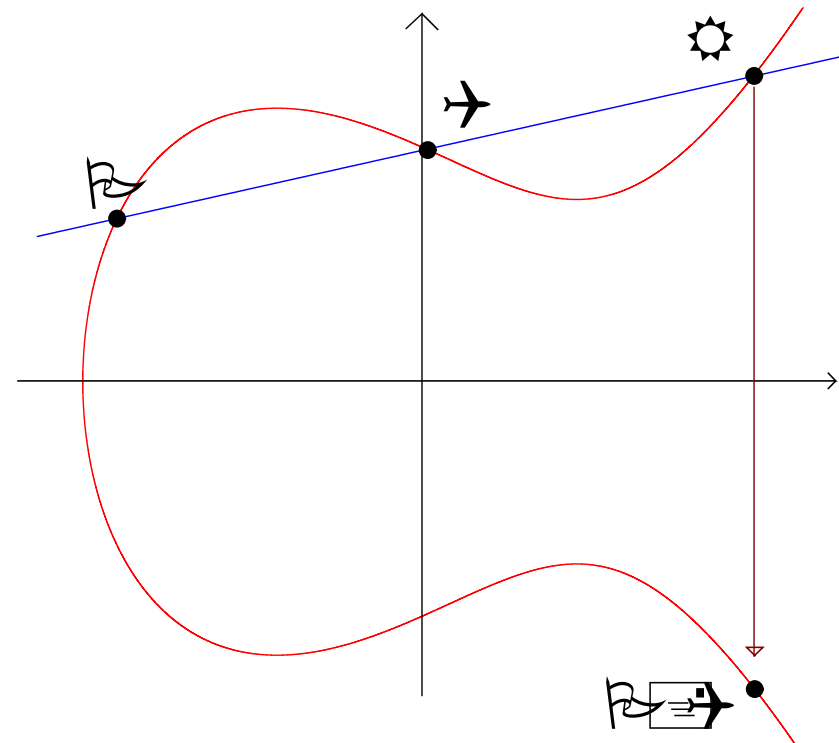
$$\xleftarrow{\beta}$$

$$k \leftarrow b \cdot \alpha = ba \cdot P$$

$$k \leftarrow a \cdot \beta = ab \cdot P$$

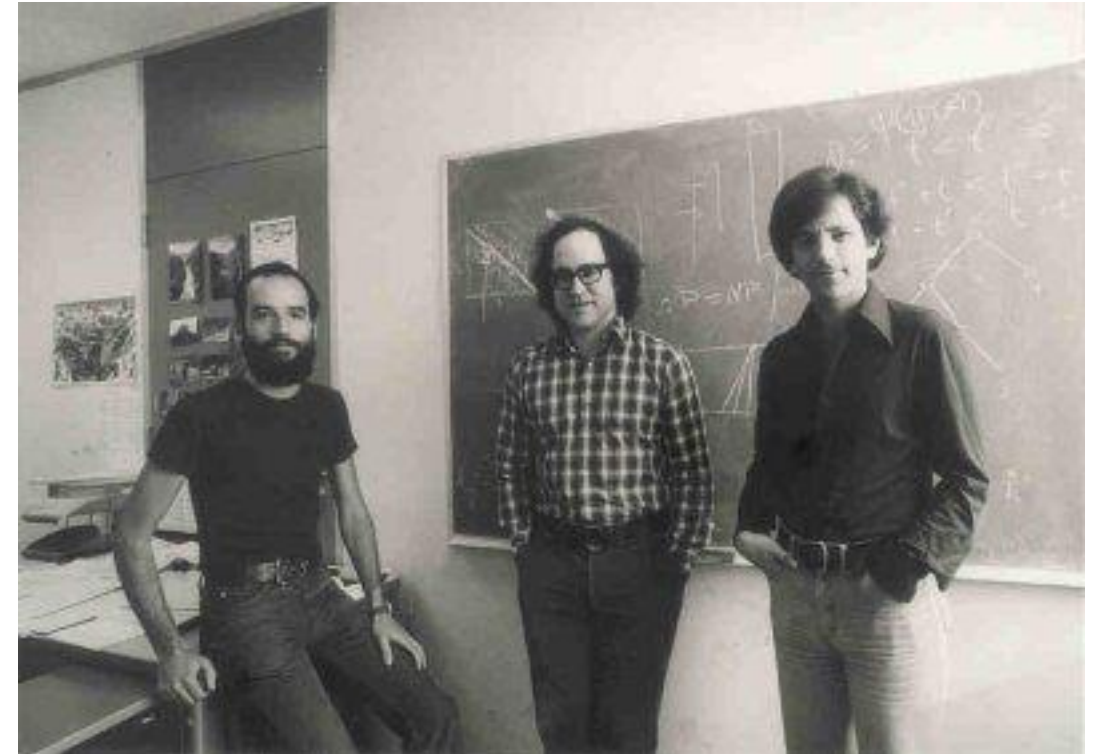
# Diffie-Hellman Key Exchange

- DHKE important building block in modern cryptography
- Mostly used over Elliptic Curves today
- Insecure against active attackers
  - cyber criminals
  - foreign intelligence agencies



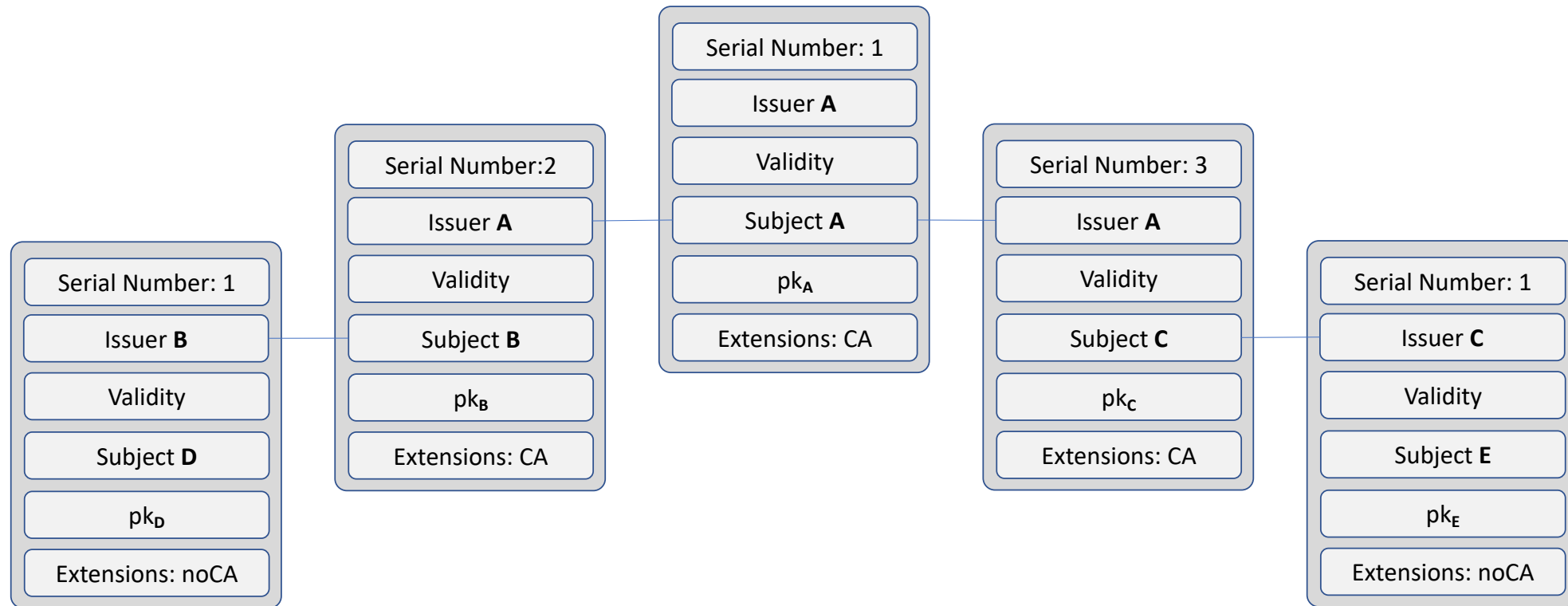
# Digital Signatures

- Rivest-Shamir-Adleman (RSA) Signatures
- Digital Signature Algorithm (DSA) Signatures
  - modulo prime numbers
  - over elliptic curves



Ron Rivest, Adi Shamir, and Leonard Adleman in 1978

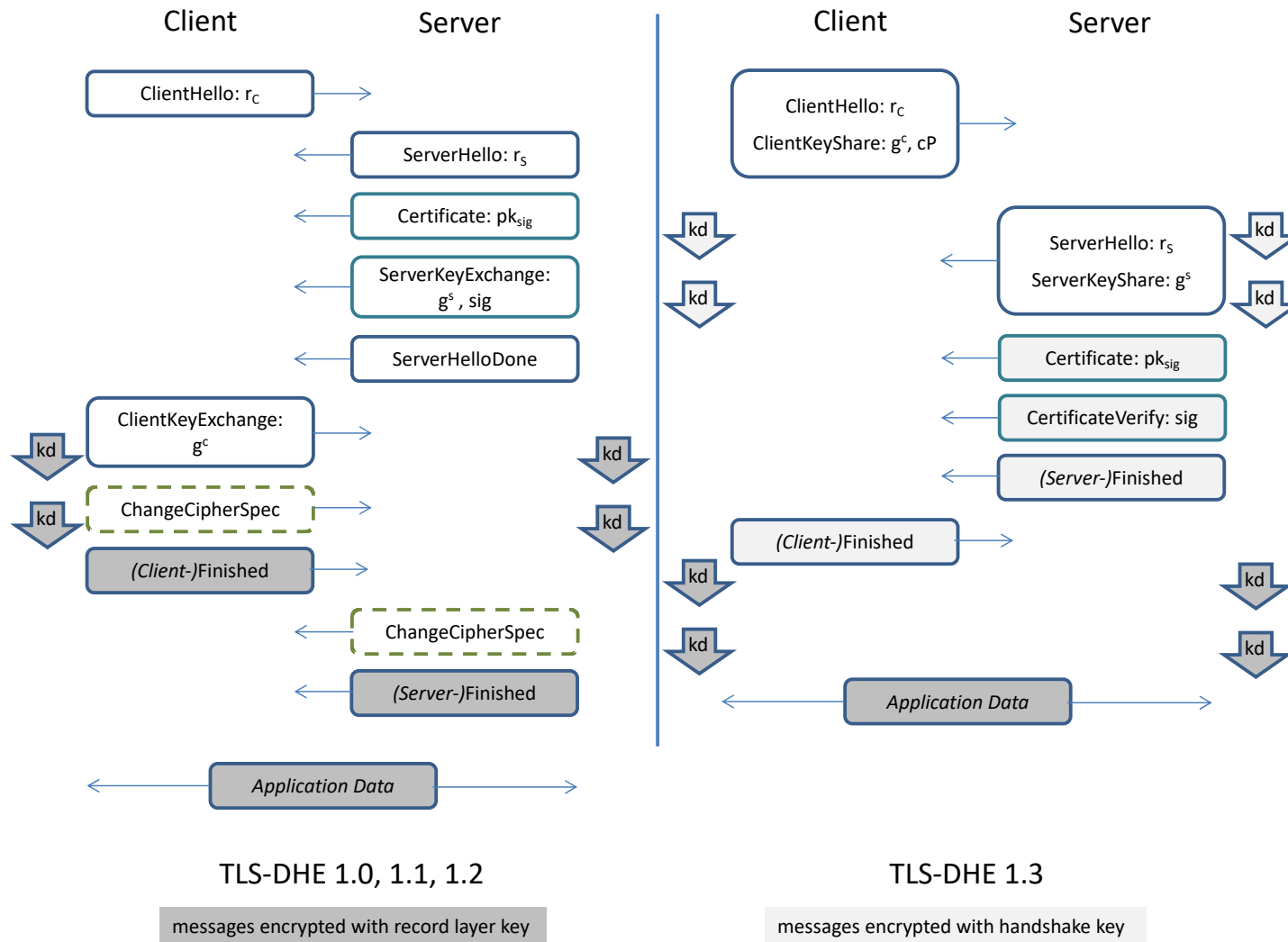
# Public-Key Infrastructures



*Digital signatures allow for inheritance of trust*

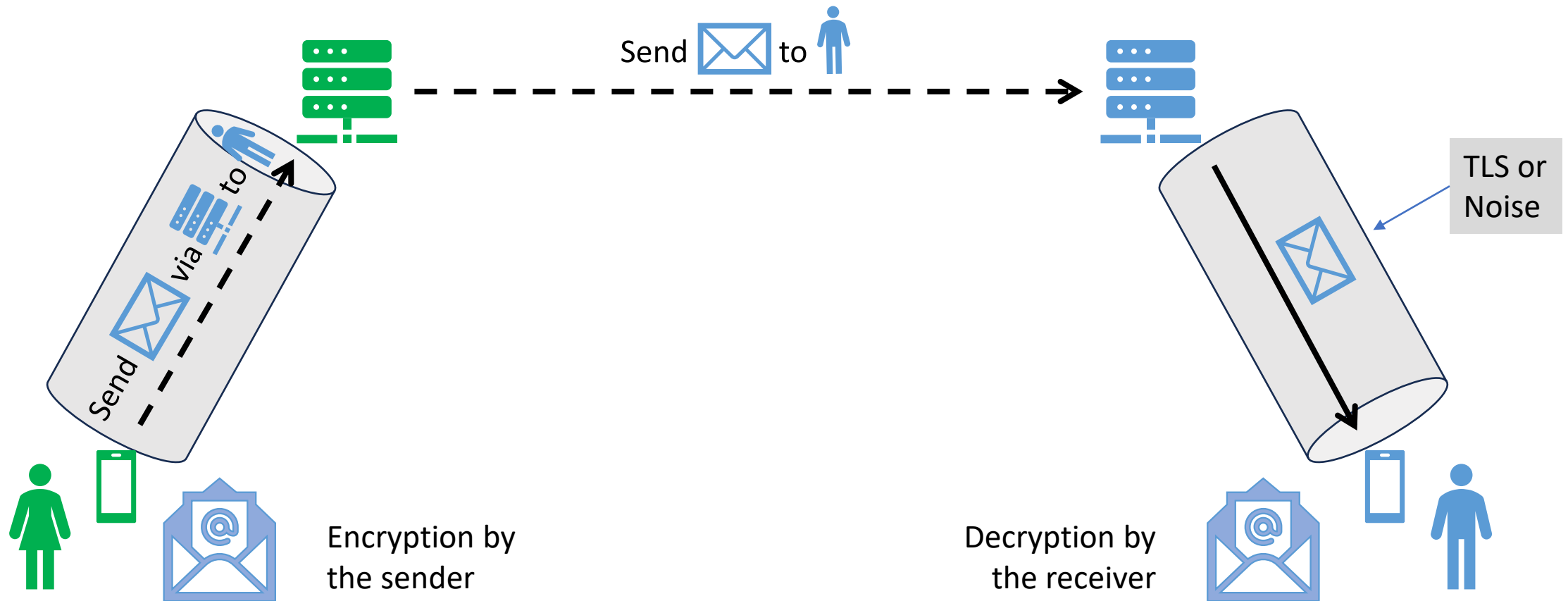


# Transport Layer Security



- TLS-DHE uses DHKE and digital signatures
- Noise and X3DH authenticate differently

# End-to-End-Encryption



# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

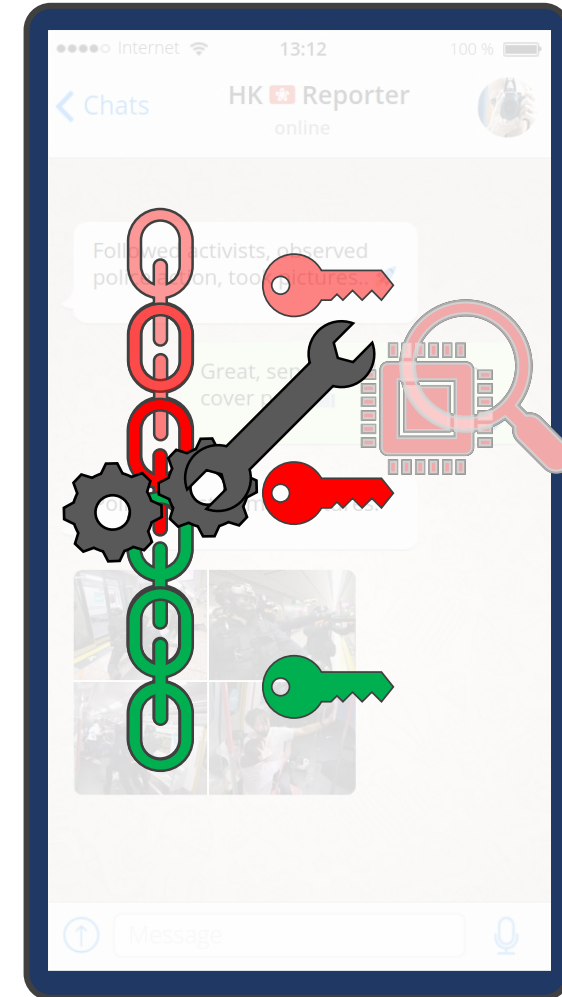
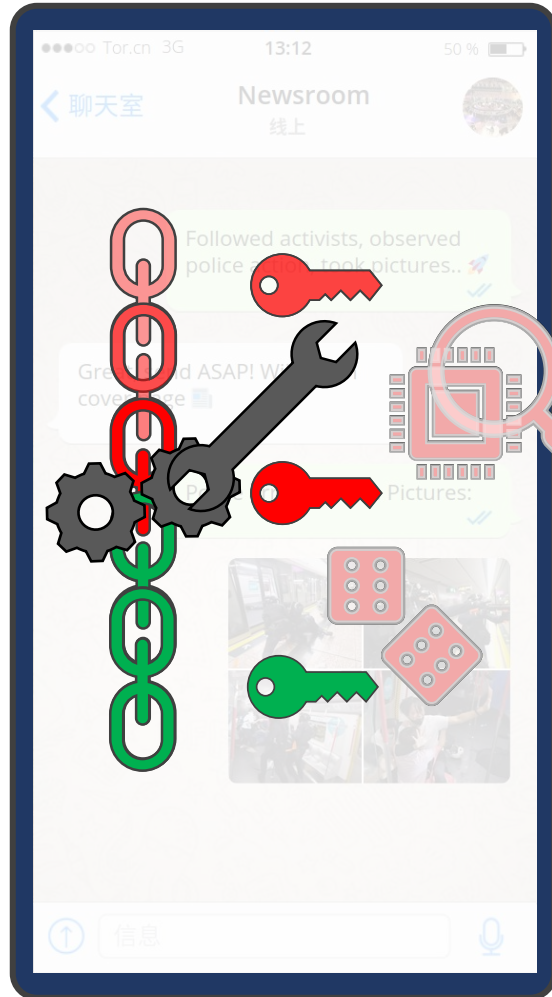
# Overview of Messengers

IM Protocol	Two-Party	Group	Real Time
Signal	Double Ratchet ( <b>DR</b> )	<b>DR</b>	WebRTC
WhatsApp	<b>DR</b>	Sender Key ( <b>SK</b> )	SRTP
Facebook Messenger	<b>DR</b> with Message Franking	<b>SK</b> with Message Franking	Undocumented
Wire	Proteus ( $\approx$ <b>DR</b> ; diff. AE)	Proteus ( $\approx$ <b>DR</b> )	SRTP
Matrix	Olm ( $\approx$ <b>DR</b> ; diff. KDF)	Megolm ( $\approx$ <b>SK</b> )	WebRTC
iMessage	Public-key encryption	Public-key encryption	SRTP
Telegram	MTPROTO	Unencrypted	MTPROTO

# Ratcheting: Continuous Key Exchange

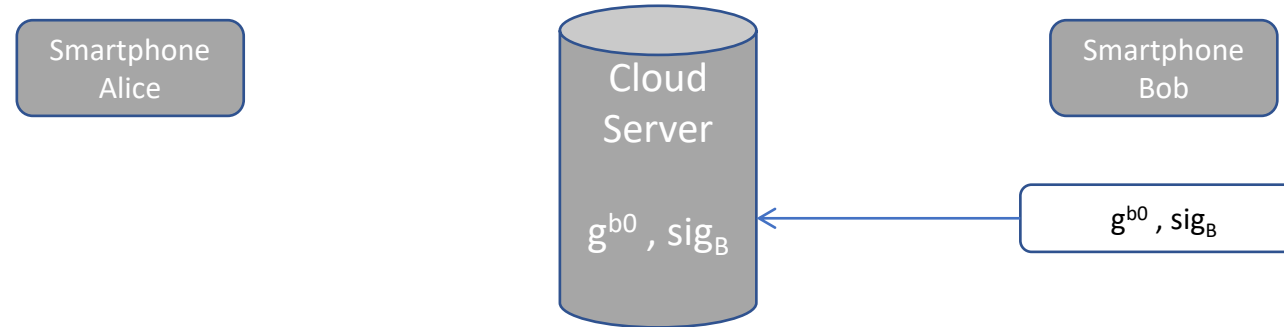
Alice

Bob

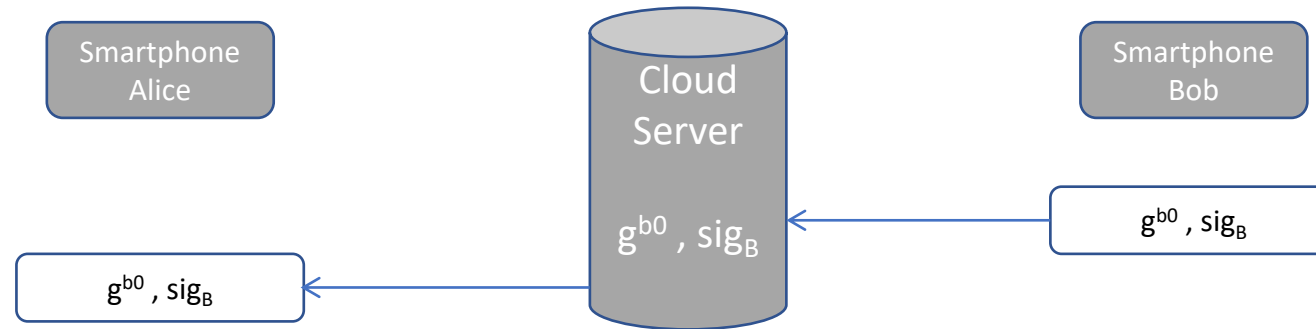


Ratchet:  
Regularly  
replace key  
material in  
local states

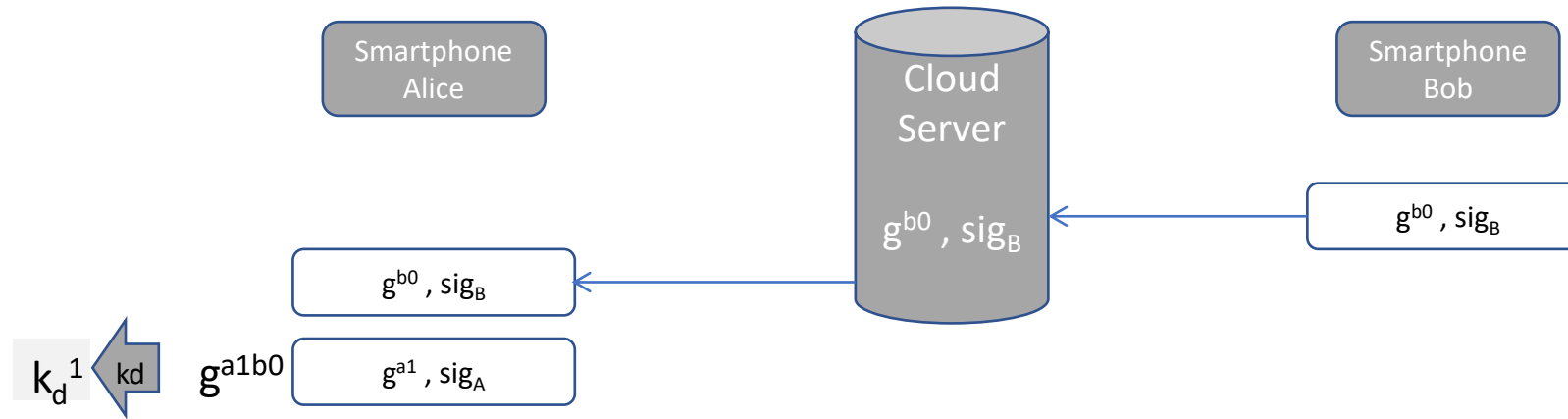
# Double Ratchet – continuous DHKE



# Double Ratchet – continuous DHKE

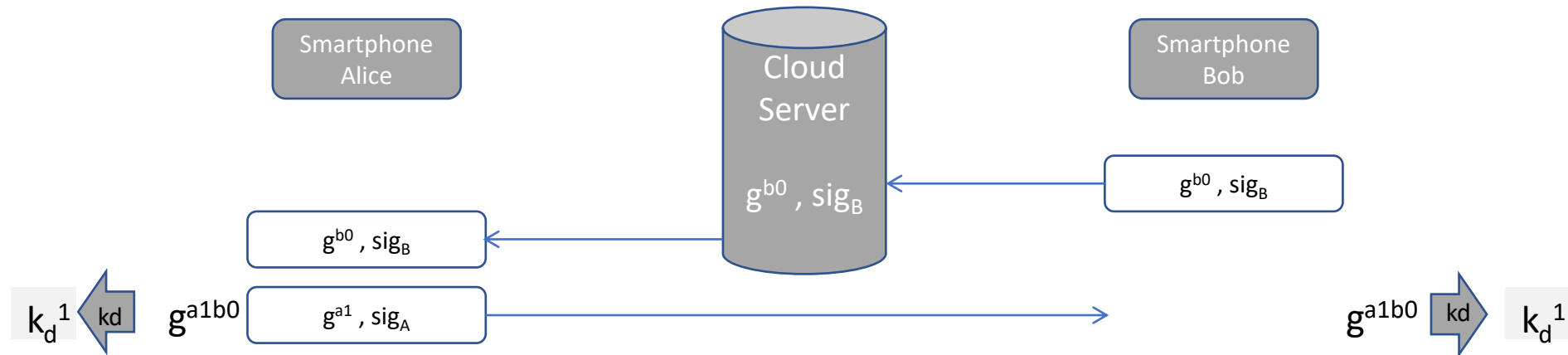


# Double Ratchet – continuous DHKE

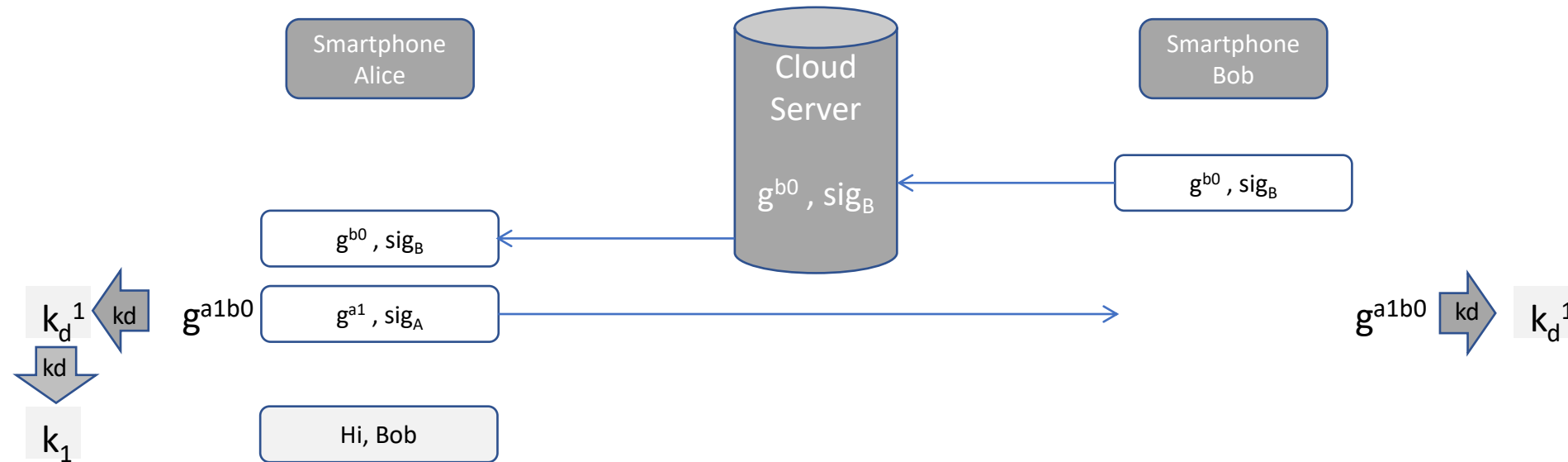




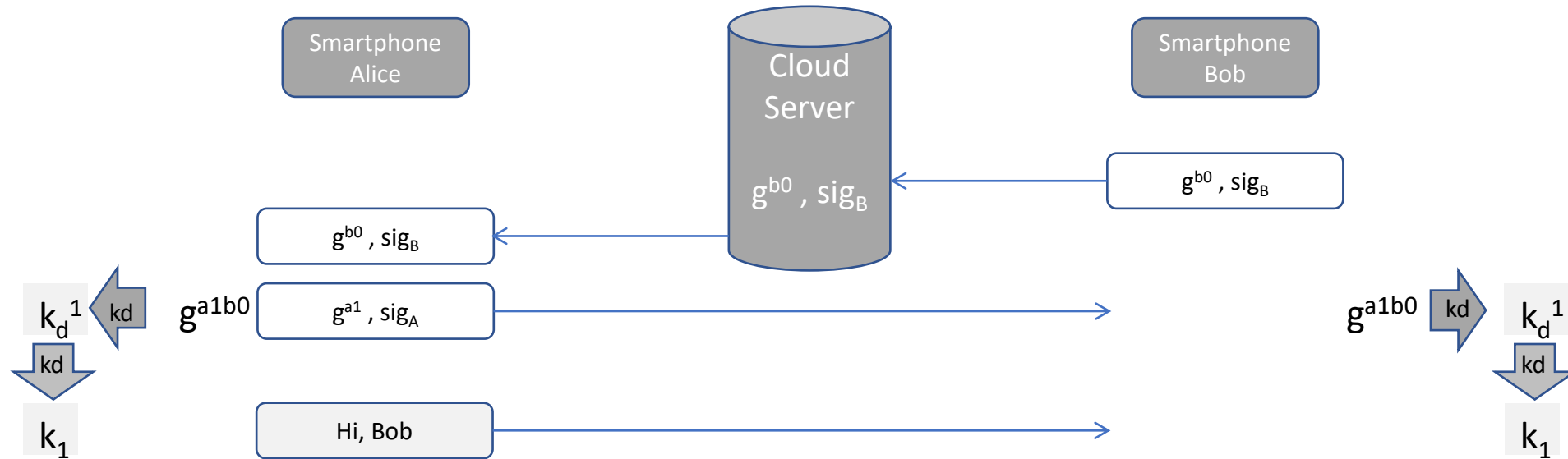
# Double Ratchet – continuous DHKE



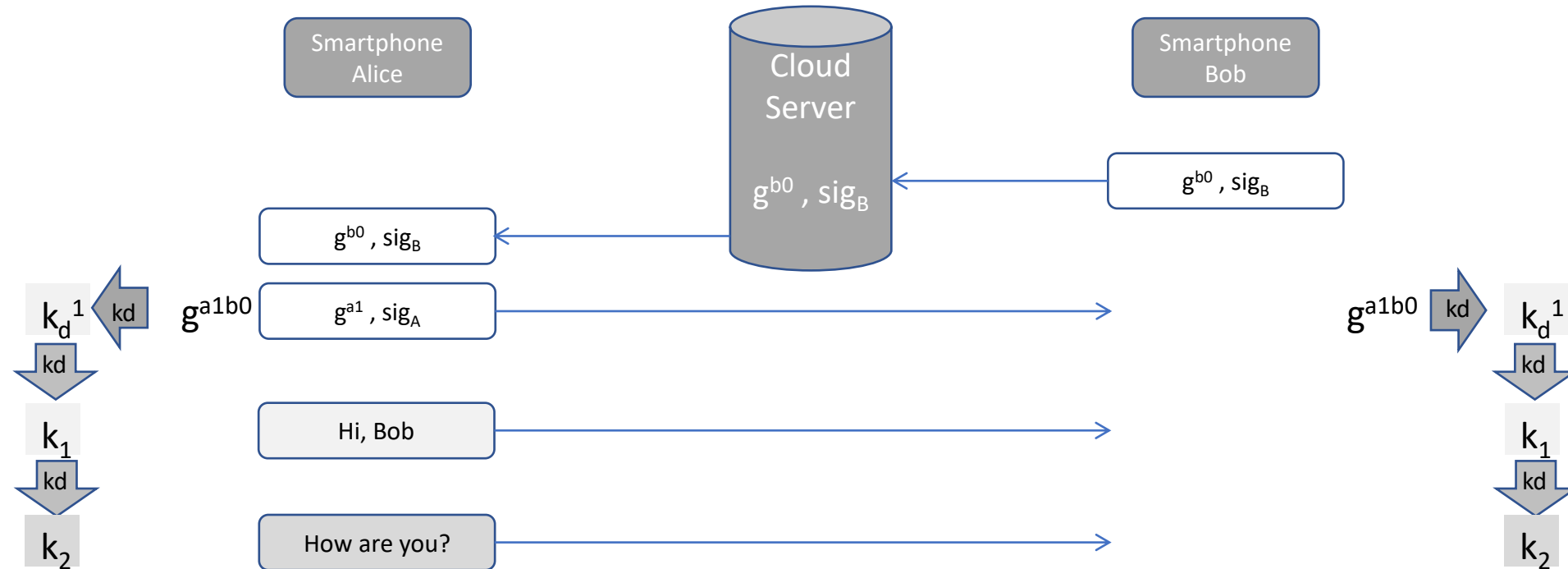
# Double Ratchet – continuous DHKE



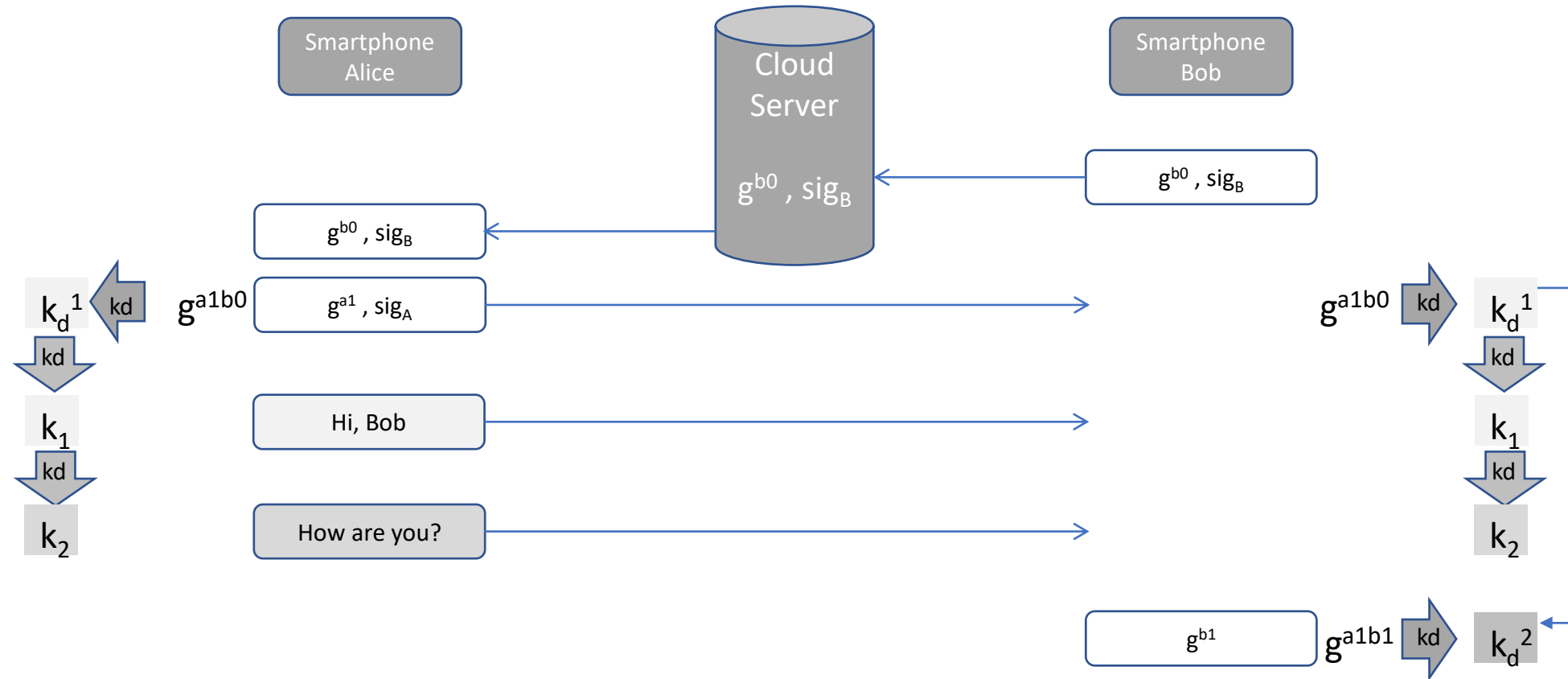
# Double Ratchet – continuous DHKE



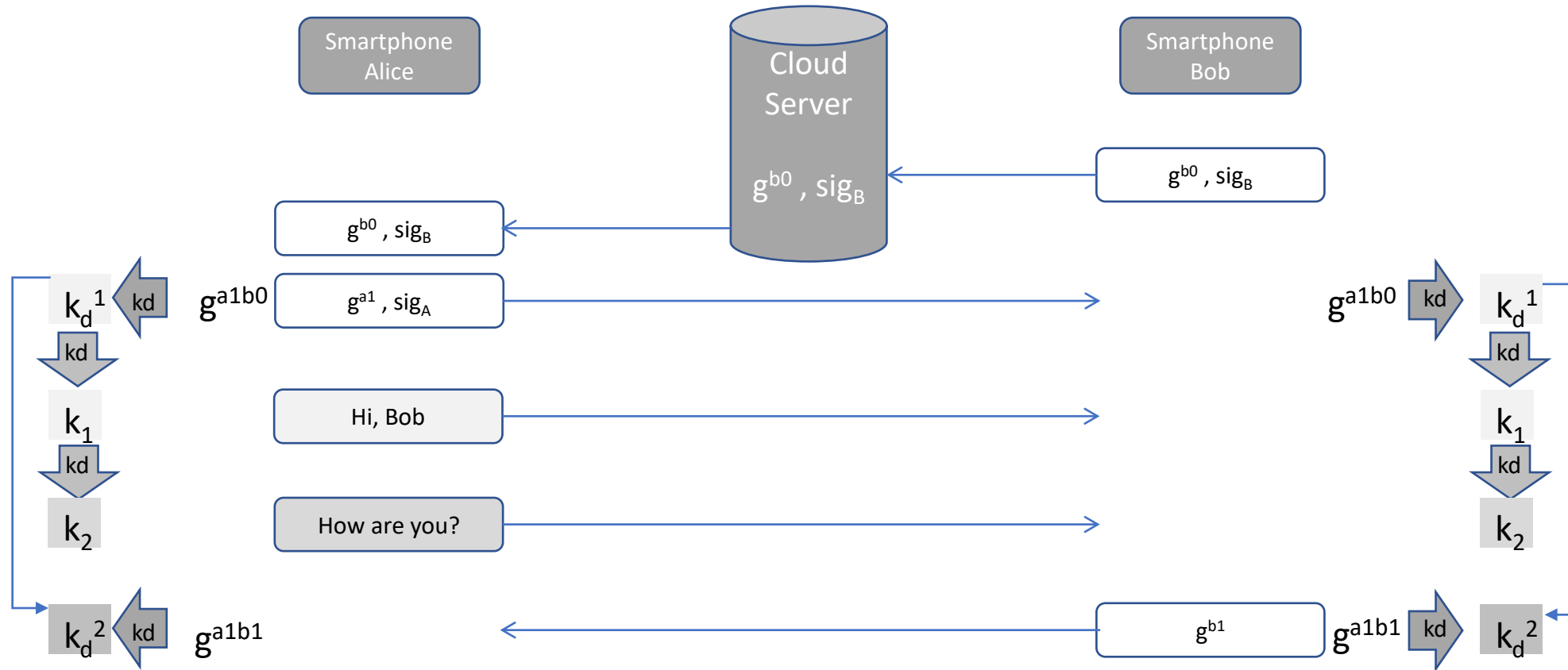
# Double Ratchet – continuous DHKE



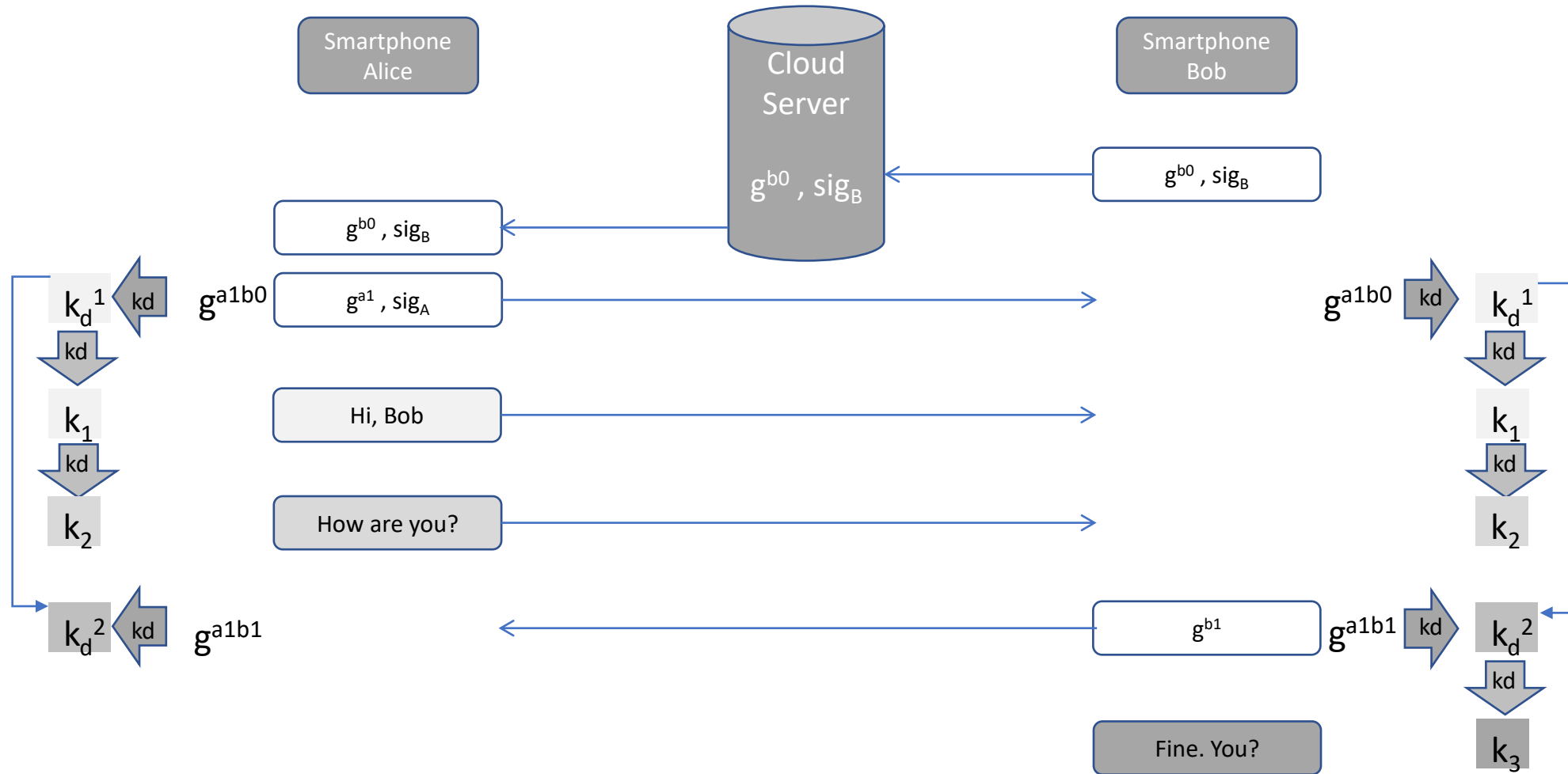
# Double Ratchet – continuous DHKE



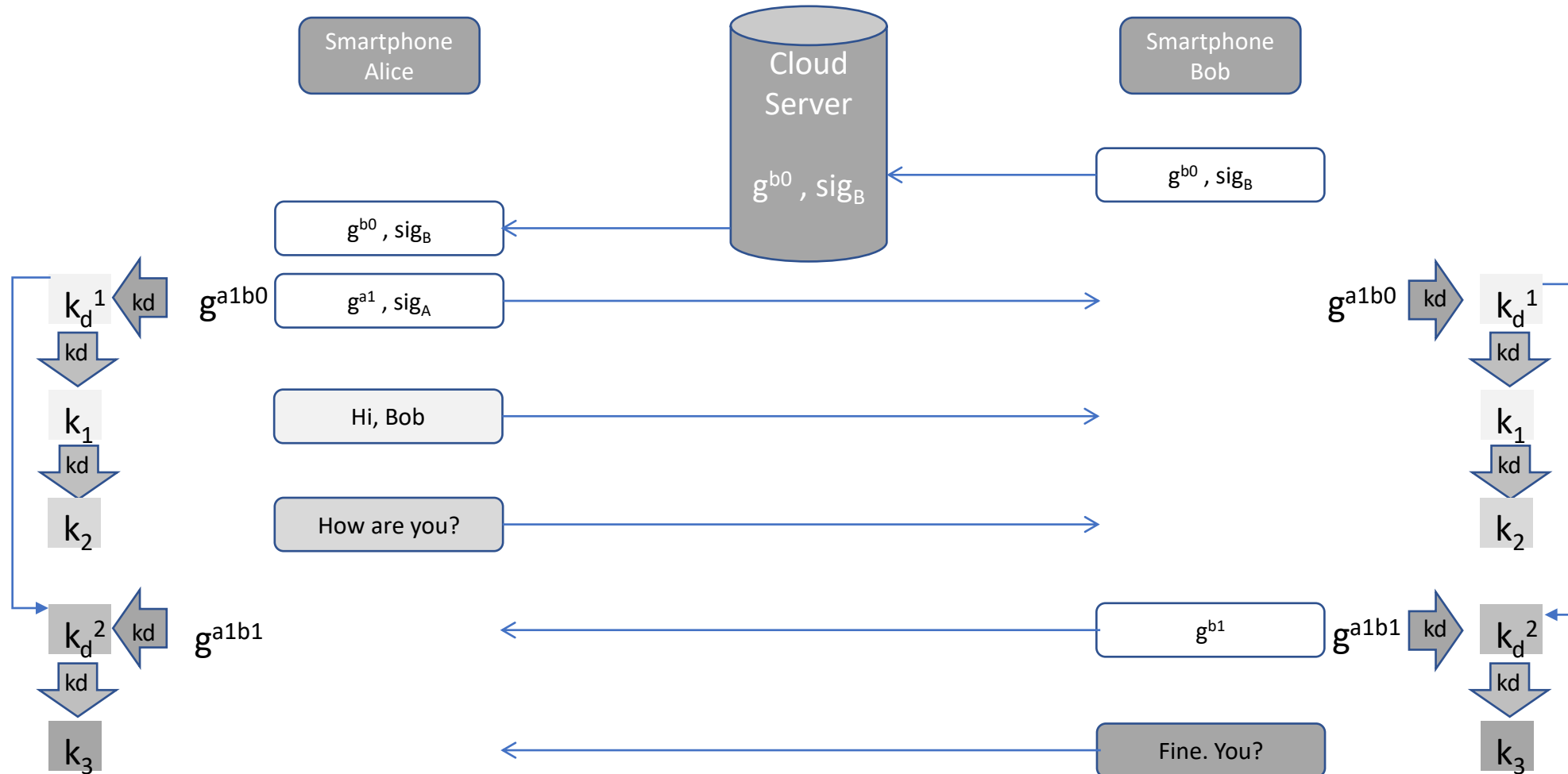
# Double Ratchet – continuous DHKE



# Double Ratchet – continuous DHKE



# Double Ratchet – continuous DHKE

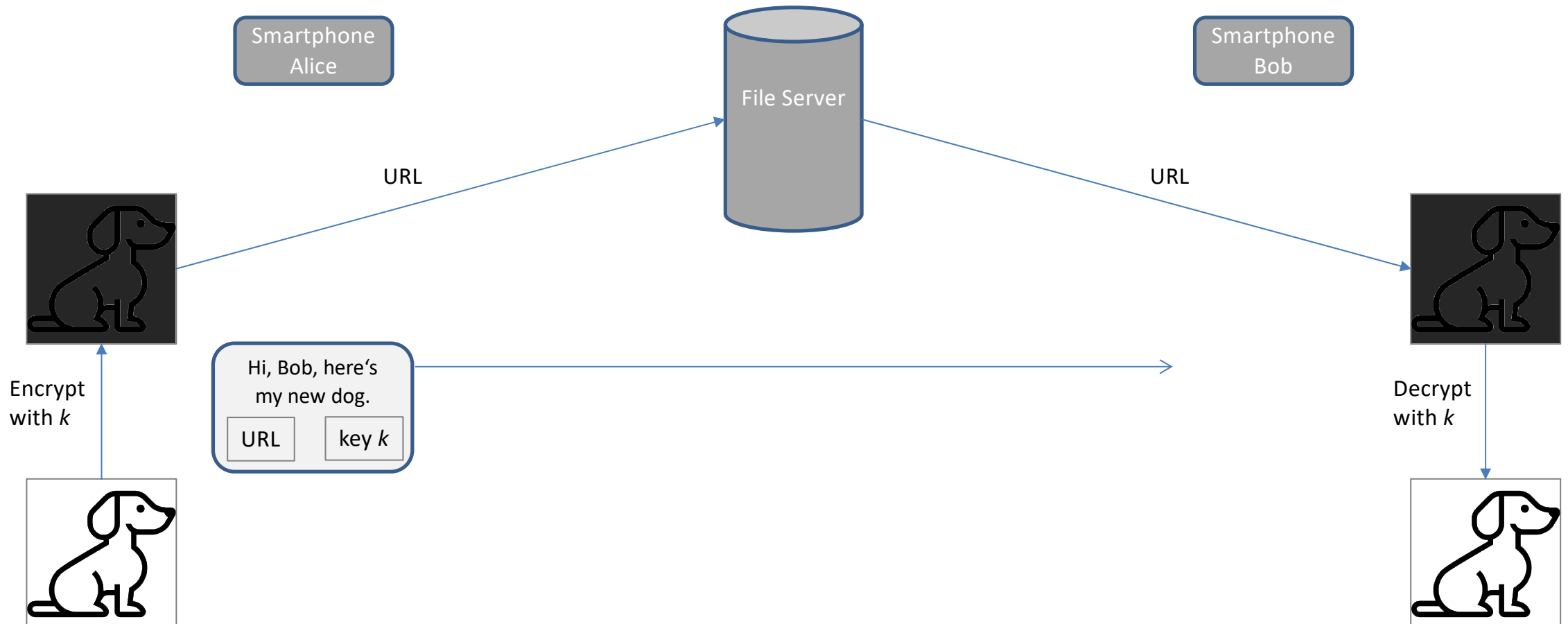




# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# Key and URL via Text Message

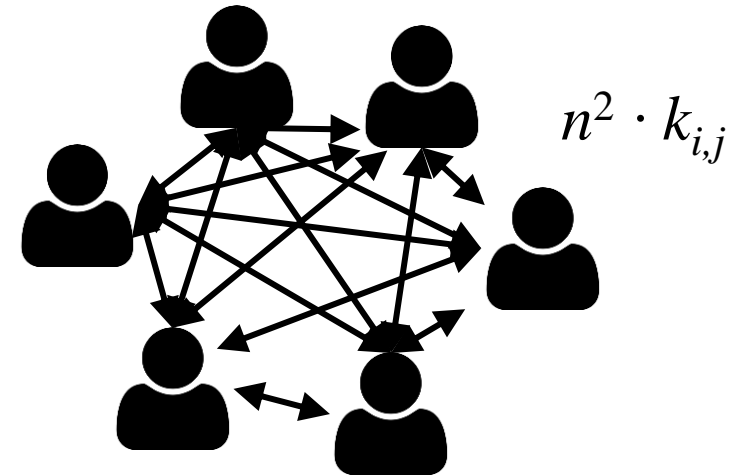


# Agenda

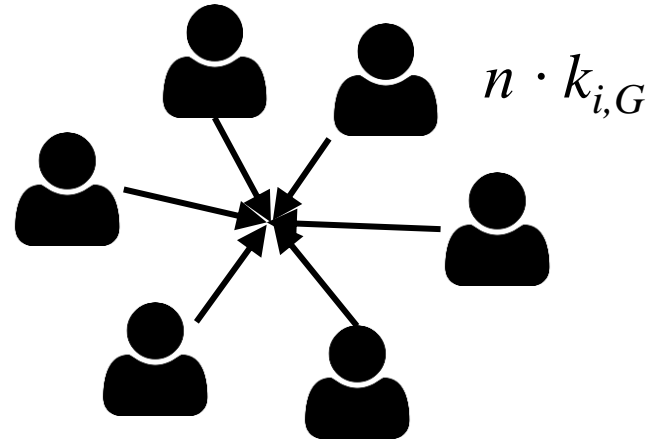
- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# Group Communication

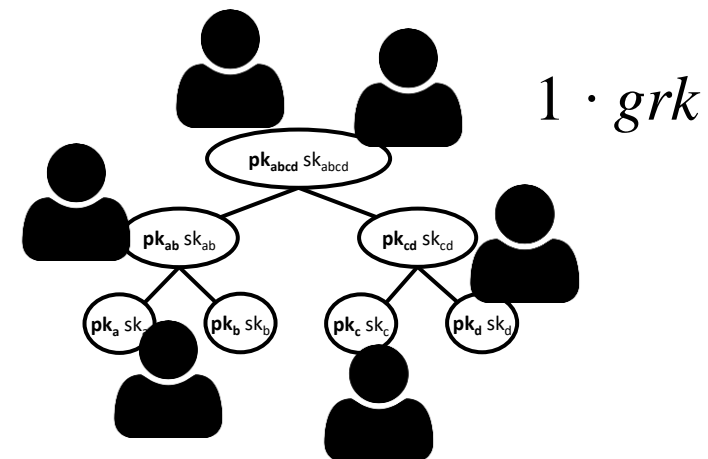
- Pairwise Channels



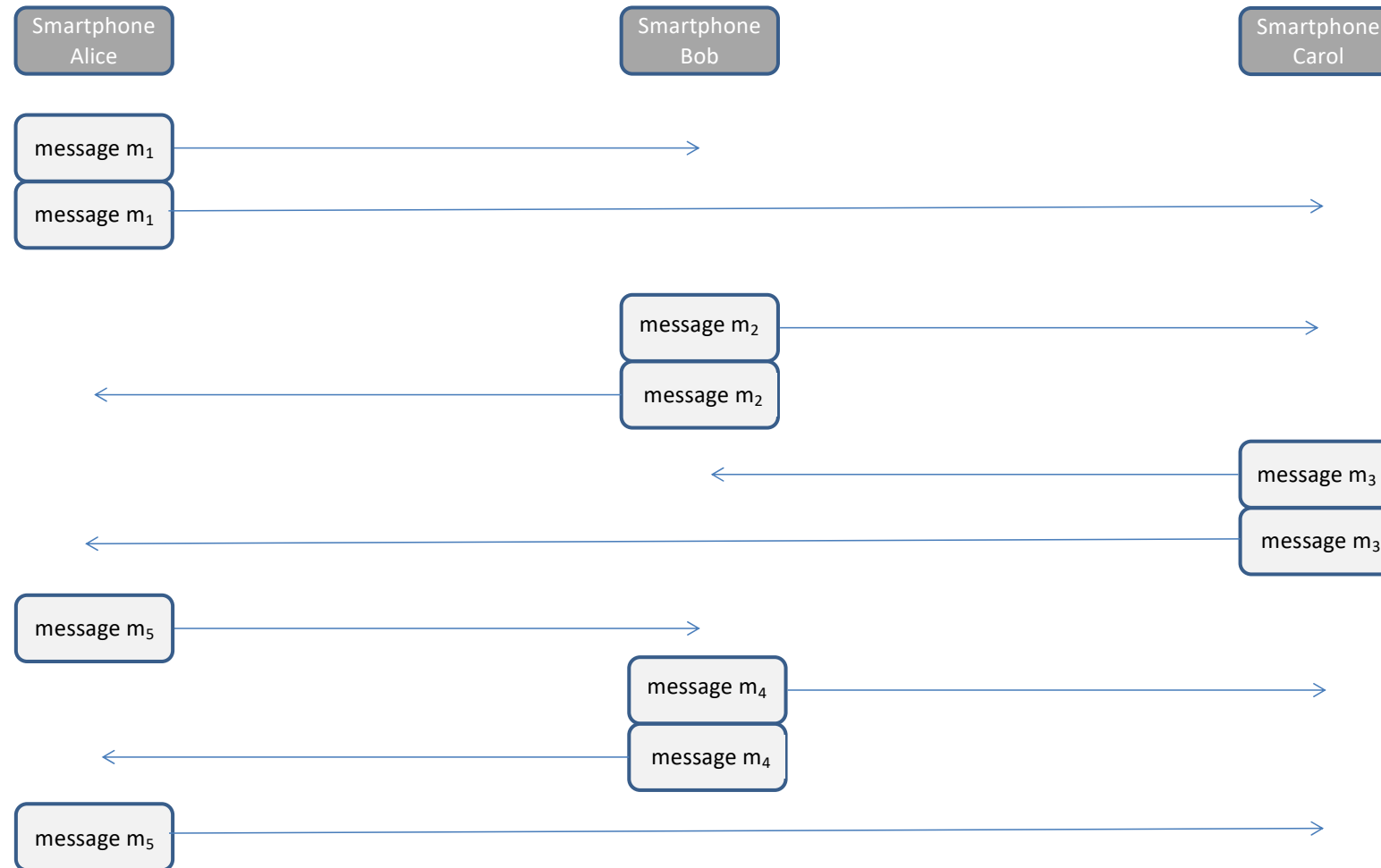
- Sender Key



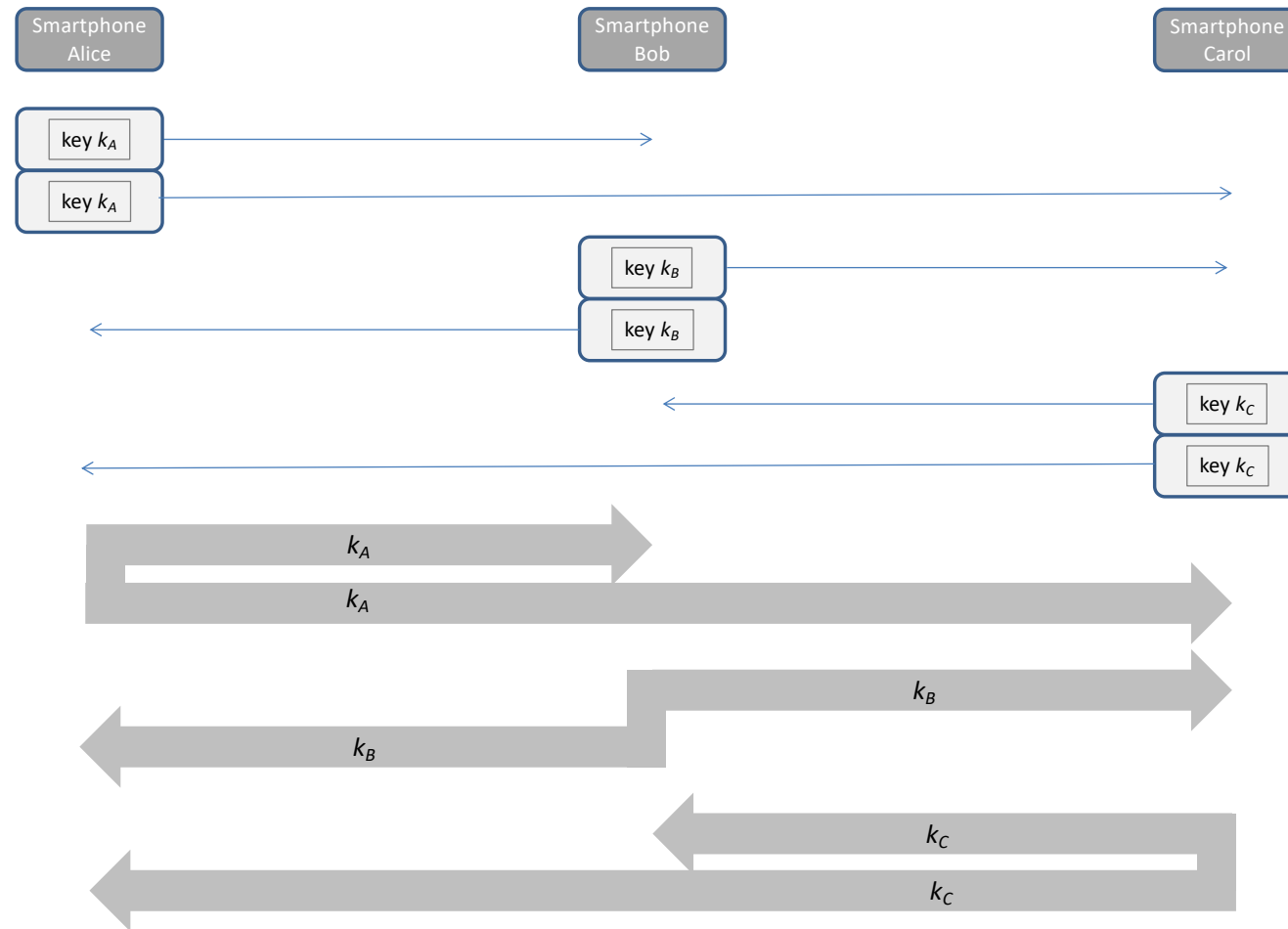
- Group Key



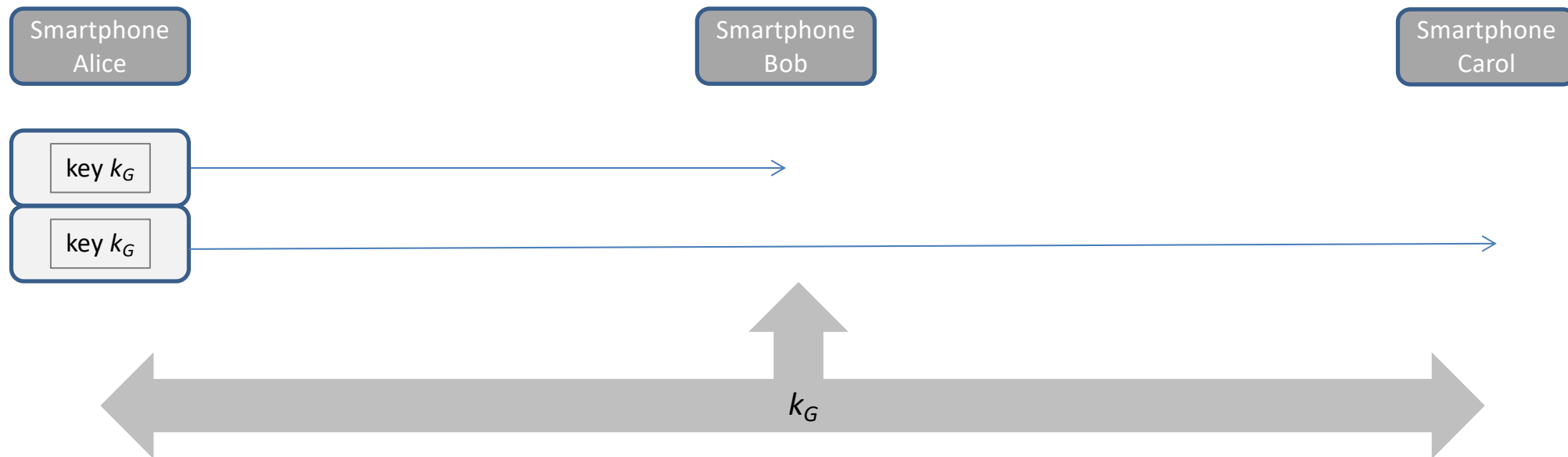
# Group Communication



# Group Communication

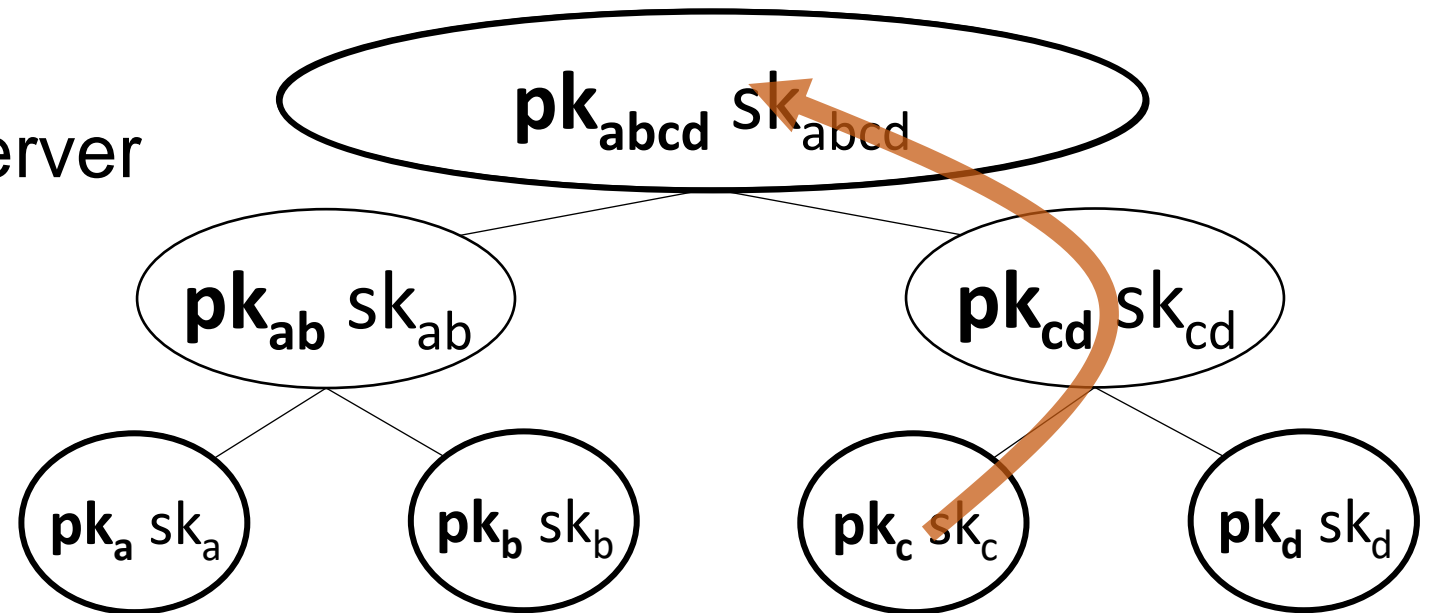


# Group Communication



# IETF MLS

- IETF Standard
- Security similar to Double Ratchet
- More efficient
- Needs synchronization server

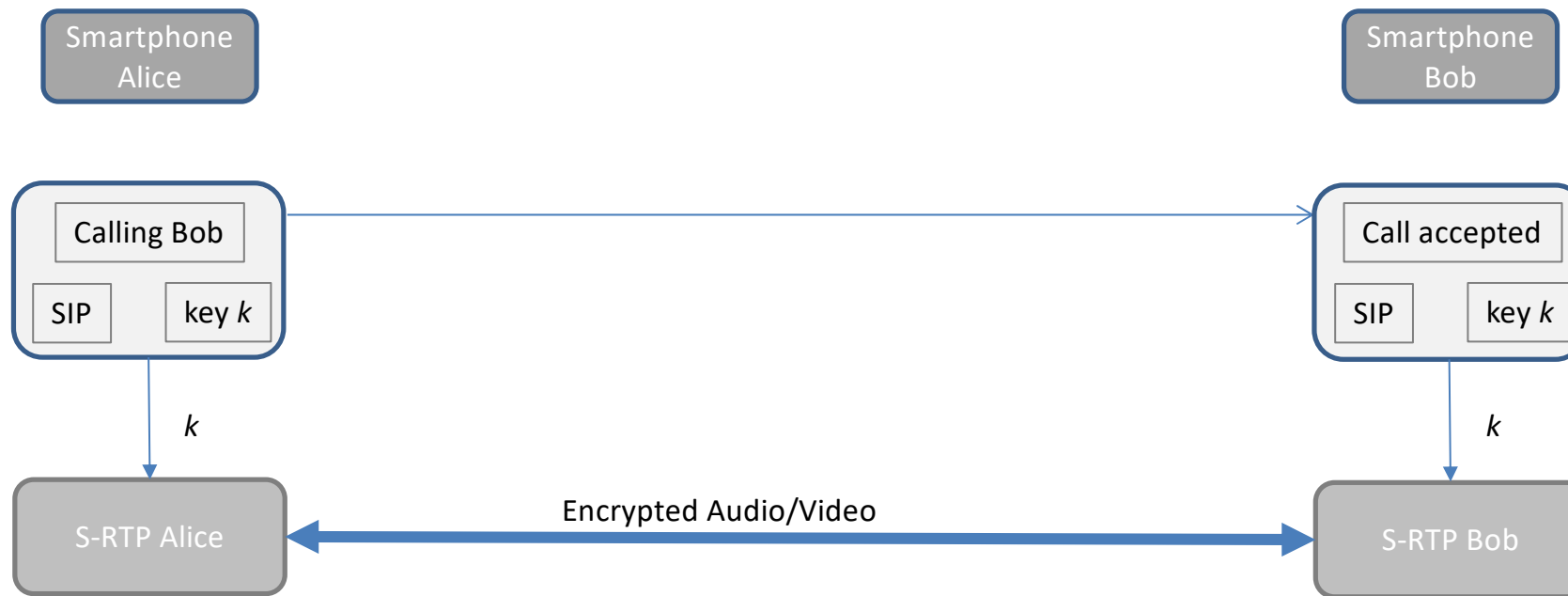




# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# Real-Time Communication



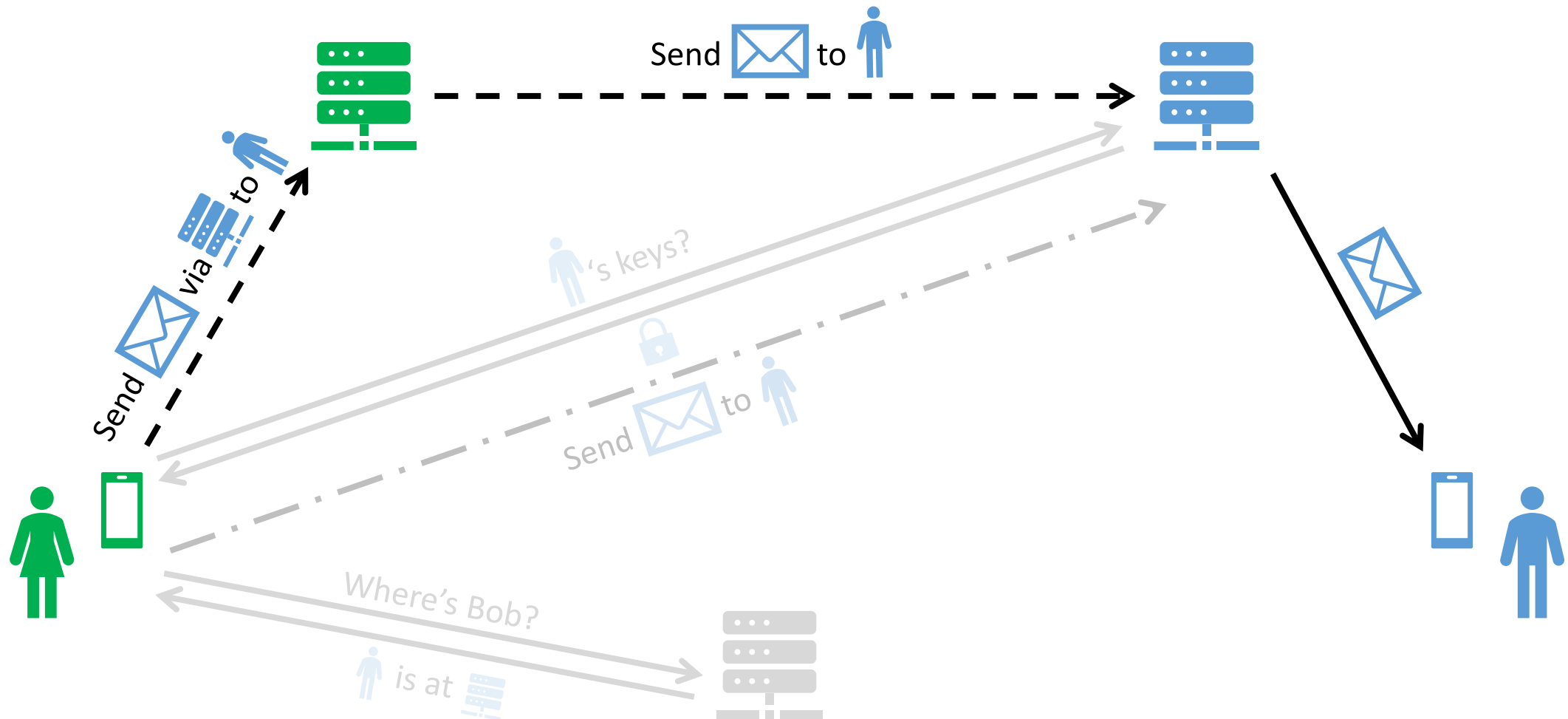
# Overview of Messengers

IM Protocol	Two-Party	Group	Real Time
Signal	Double Ratchet ( <b>DR</b> )	<b>DR</b>	WebRTC
WhatsApp	<b>DR</b>	Sender Key ( <b>SK</b> )	SRTP
Facebook Messenger	<b>DR</b> with Message Franking	<b>SK</b> with Message Franking	Undocumented
Wire	Proteus ( $\approx$ <b>DR</b> ; diff. AE)	Proteus ( $\approx$ <b>DR</b> )	SRTP
Matrix	Olm ( $\approx$ <b>DR</b> ; diff. KDF)	Megolm ( $\approx$ <b>SK</b> )	WebRTC
iMessage	Public-key encryption	Public-key encryption	SRTP
Telegram	MTPROTO	Unencrypted	MTPROTO

# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# Overview of Setting

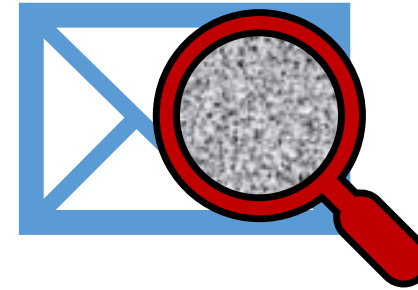


# Requirements

## Functional Interop: Two-Party, Groups, Real Time

### End-to-End Confidentiality:

“*The level of security, including the end-to-end encryption, [...] shall be preserved across the interoperable services.*” §3



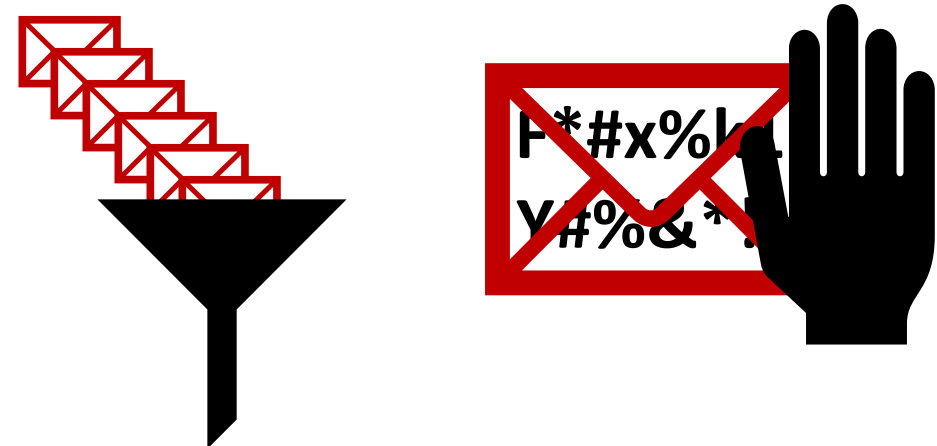
### Metadata Protection:

“*The gatekeeper shall collect and exchange [...] only the personal data of end users that is strictly necessary [...].*” §8

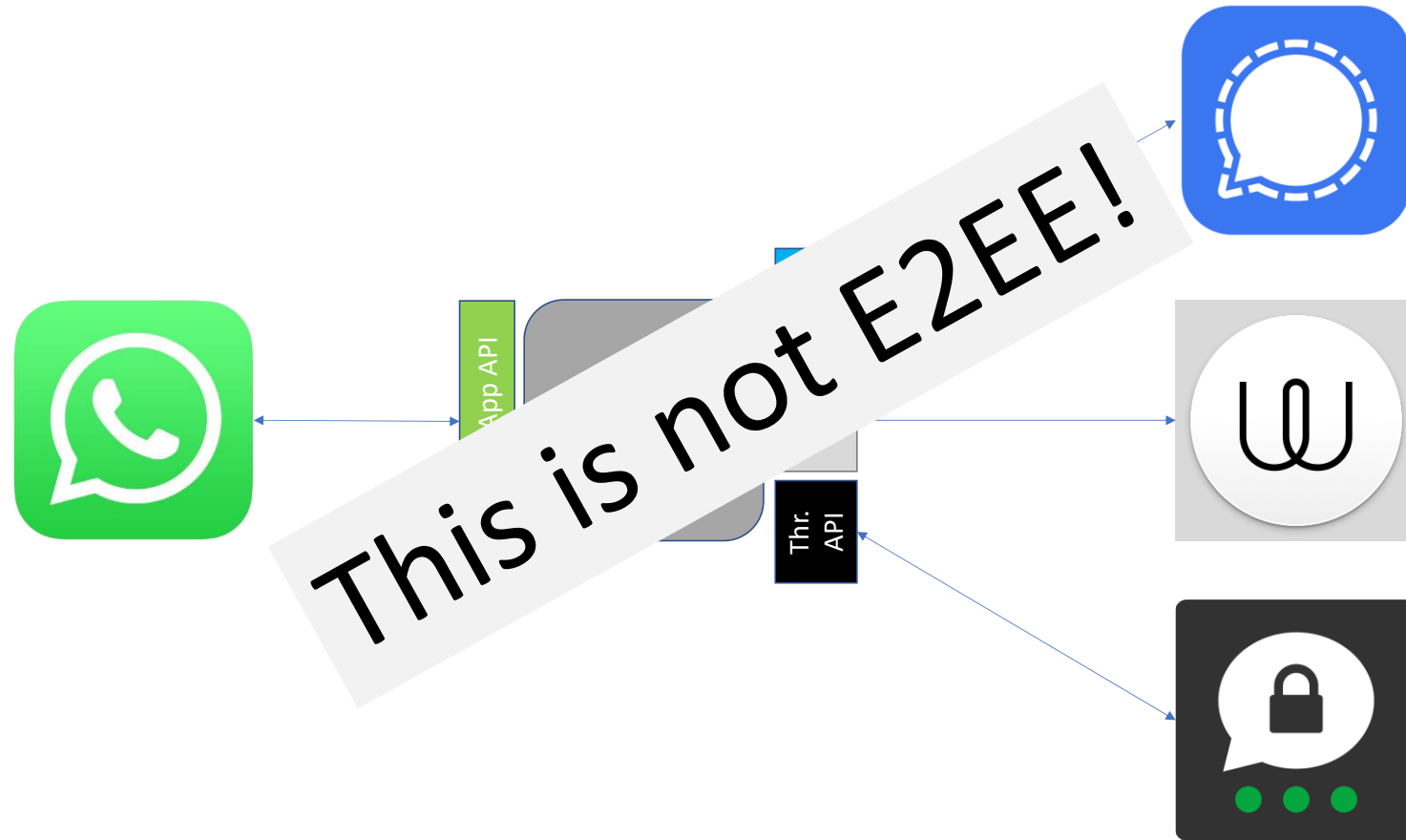


### Abuse Prevention:

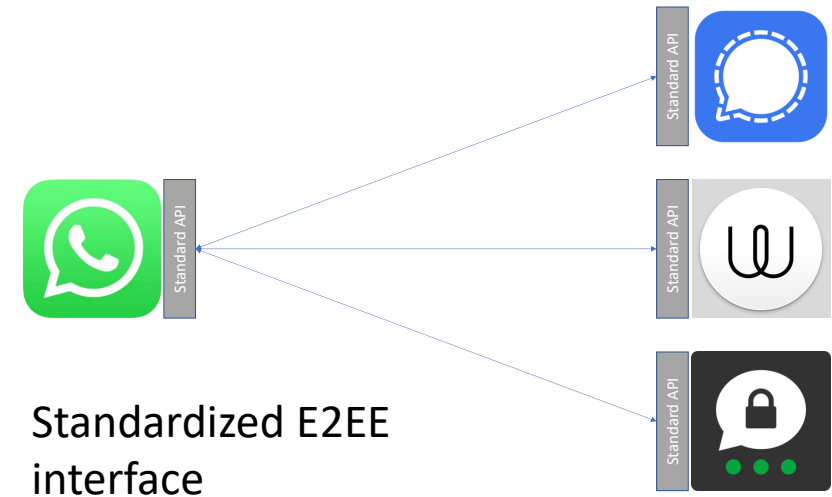
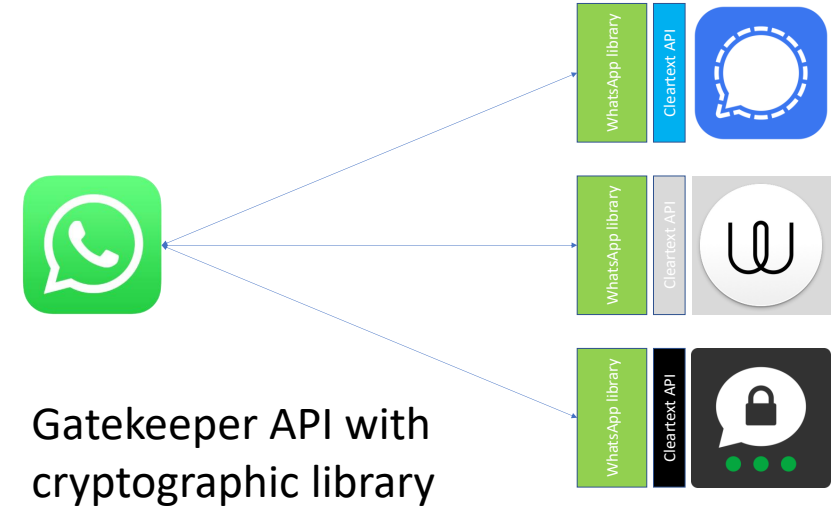
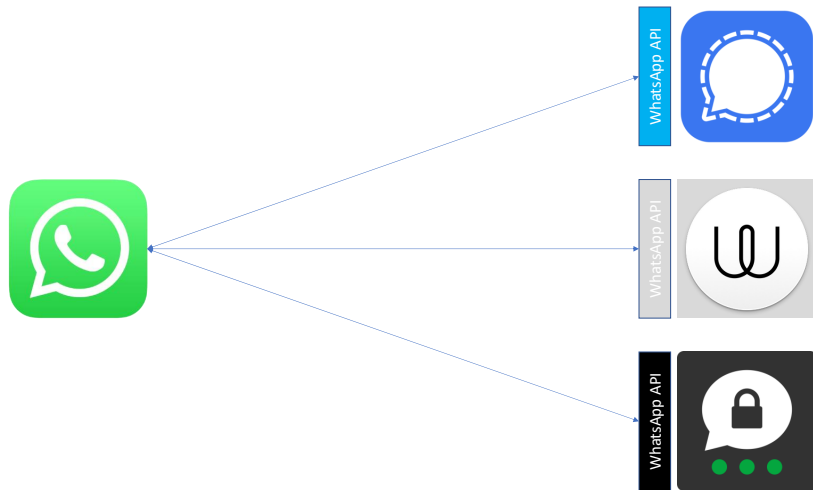
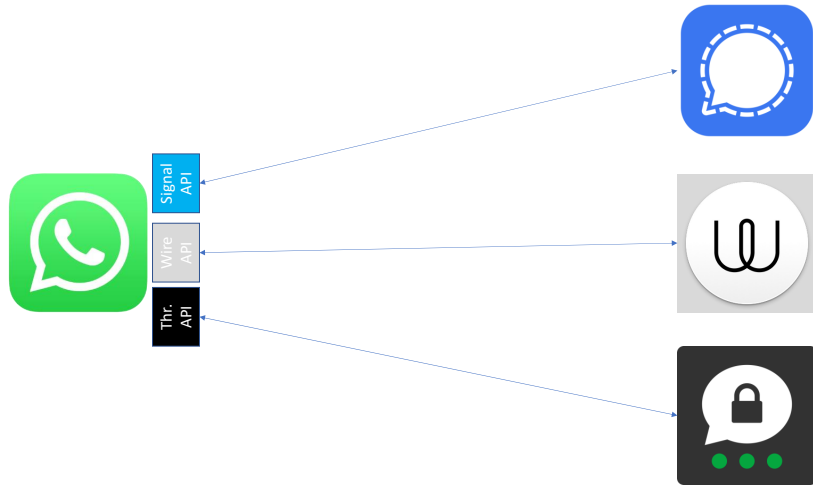
“*The gatekeeper [should be able to take] measures to [...] not endanger the integrity, security and privacy of its services [...].*” §9



# Where to 'translate' E2EE?



# Where to 'translate' E2EE?

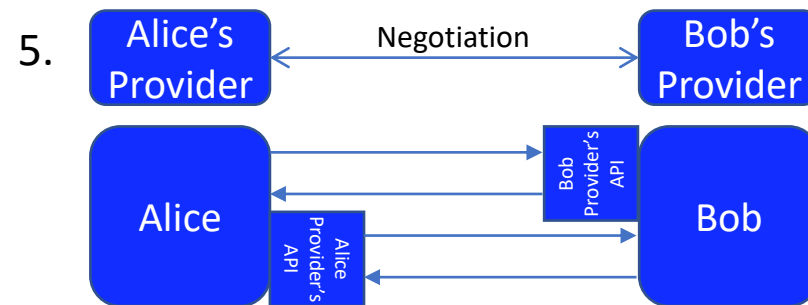
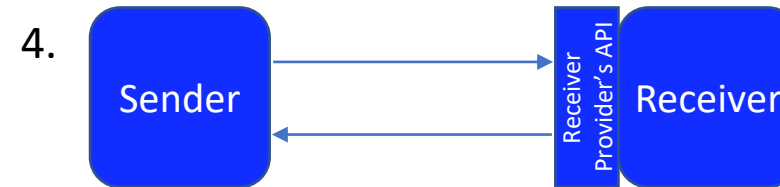
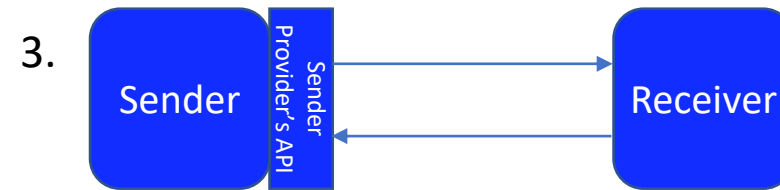




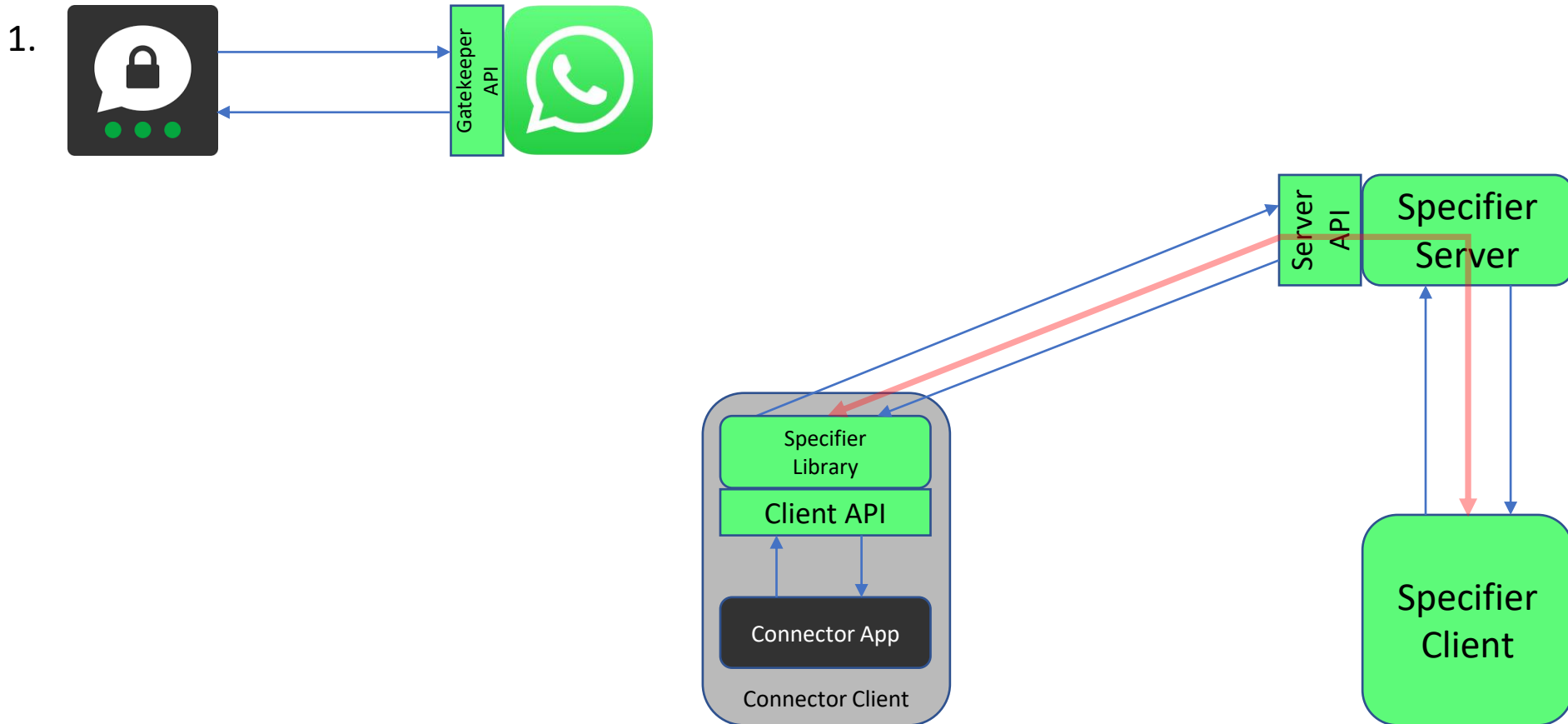
# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# Location of API



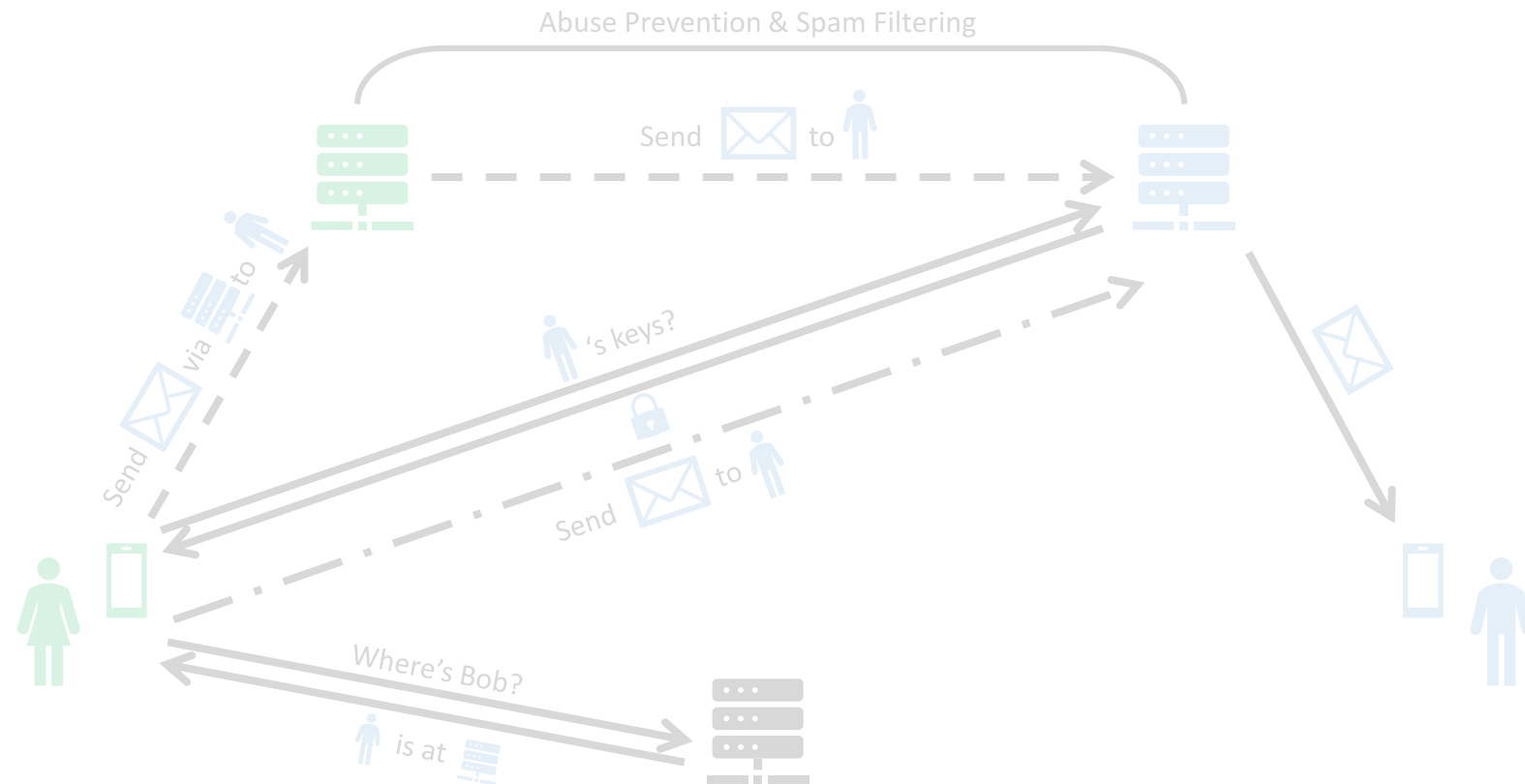
# Location of API: Client & Server



# Communication Protocol

## 1. Gatekeeper's Core Protocol + X

## 2. Standardization of Protocol

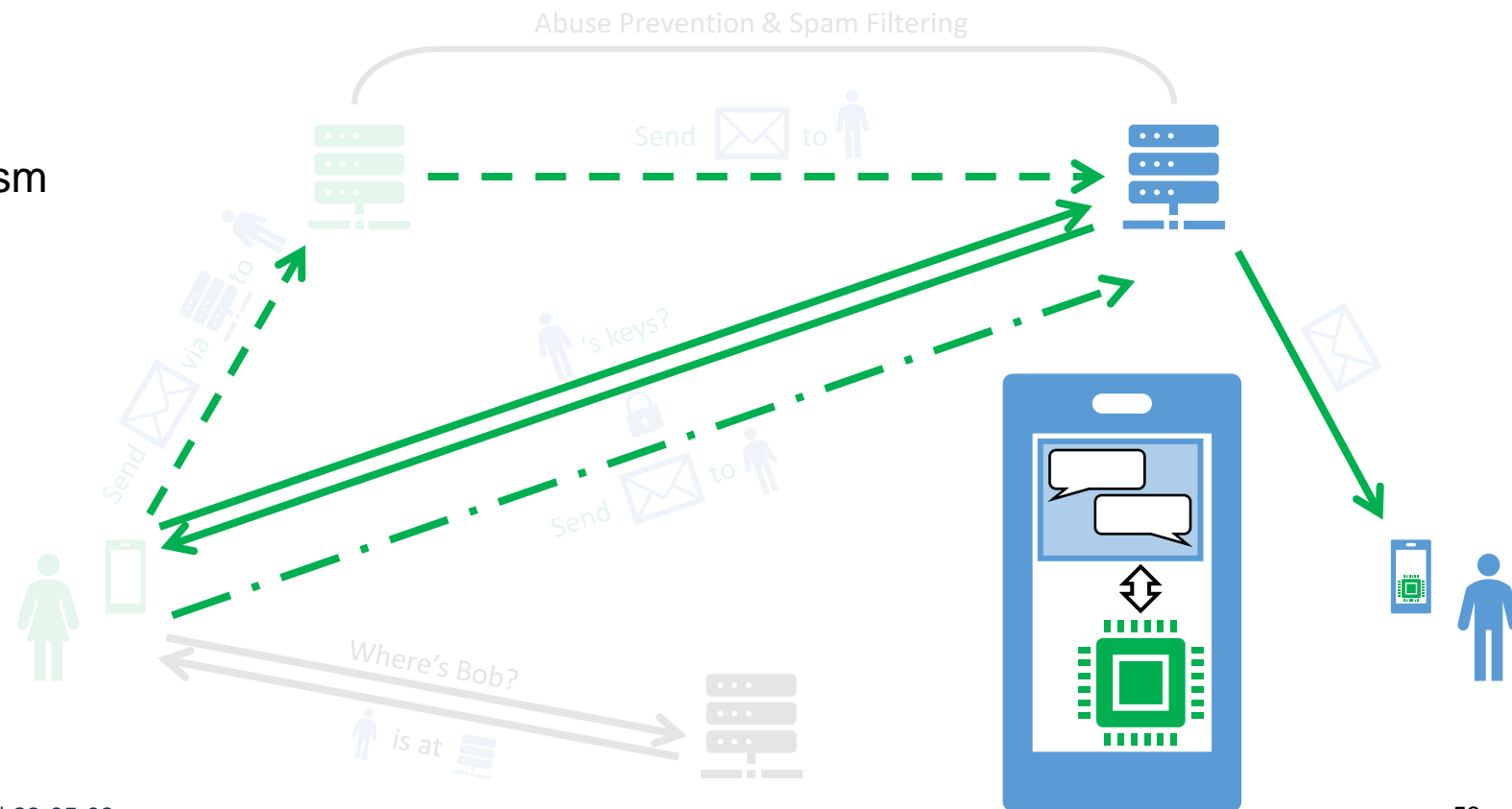


# Communication Protocol

## 1. Gatekeeper's Core Protocol + X

- API + Documentation
- Client: Library?
- Server:
  - Replicate key distribution
  - Forwarding service
  - Abuse prevention (interactively)
  - Reporting and blocking mechanism

## 2. Standardization of Protocol

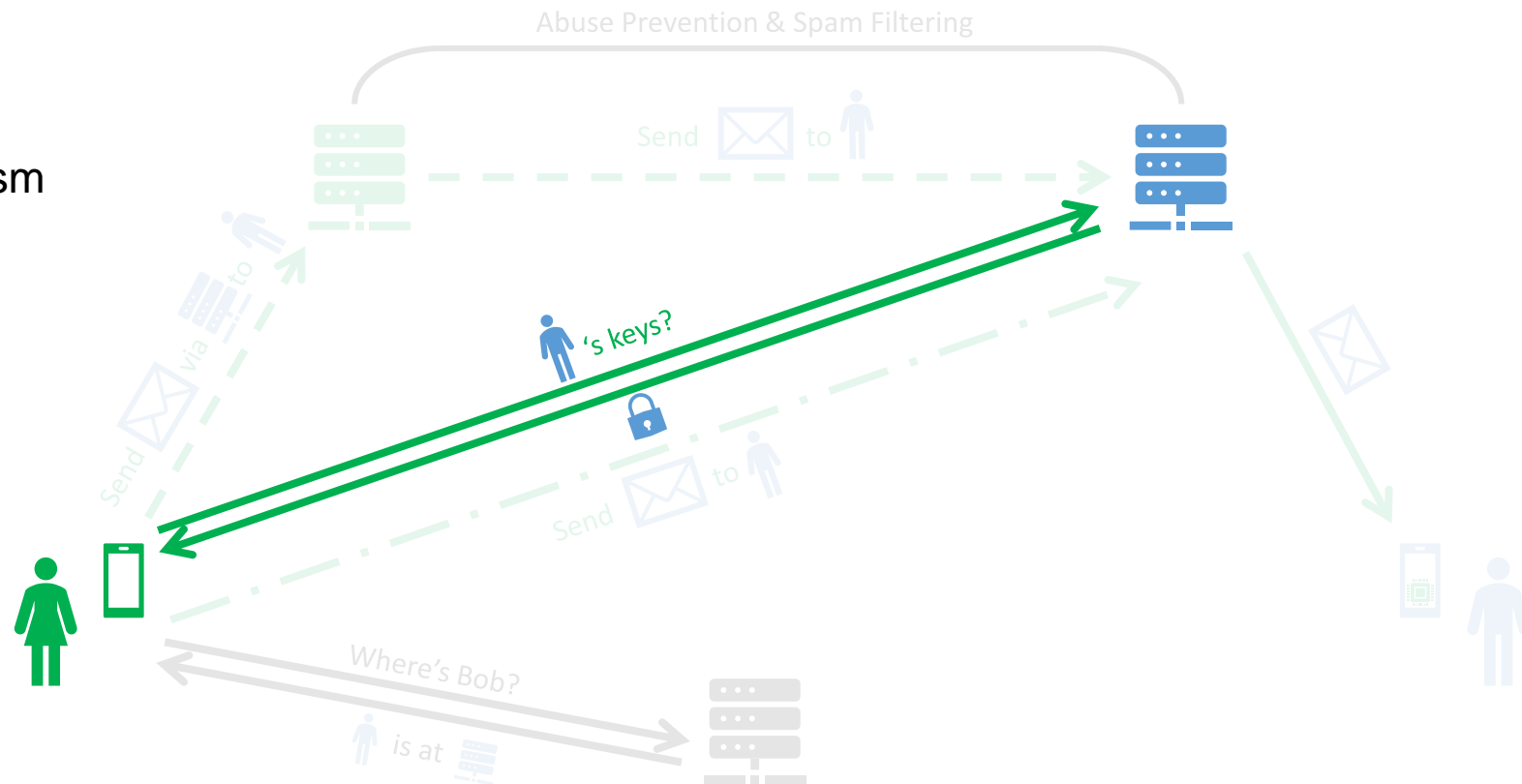


# Communication Protocol

## 1. Gatekeeper's Core Protocol + X

- API + Documentation
- Client: Library?
- Server:
  - Replicate key distribution
  - Forwarding service
  - Abuse prevention (interactively)
  - Reporting and blocking mechanism

## 2. Standardization of Protocol

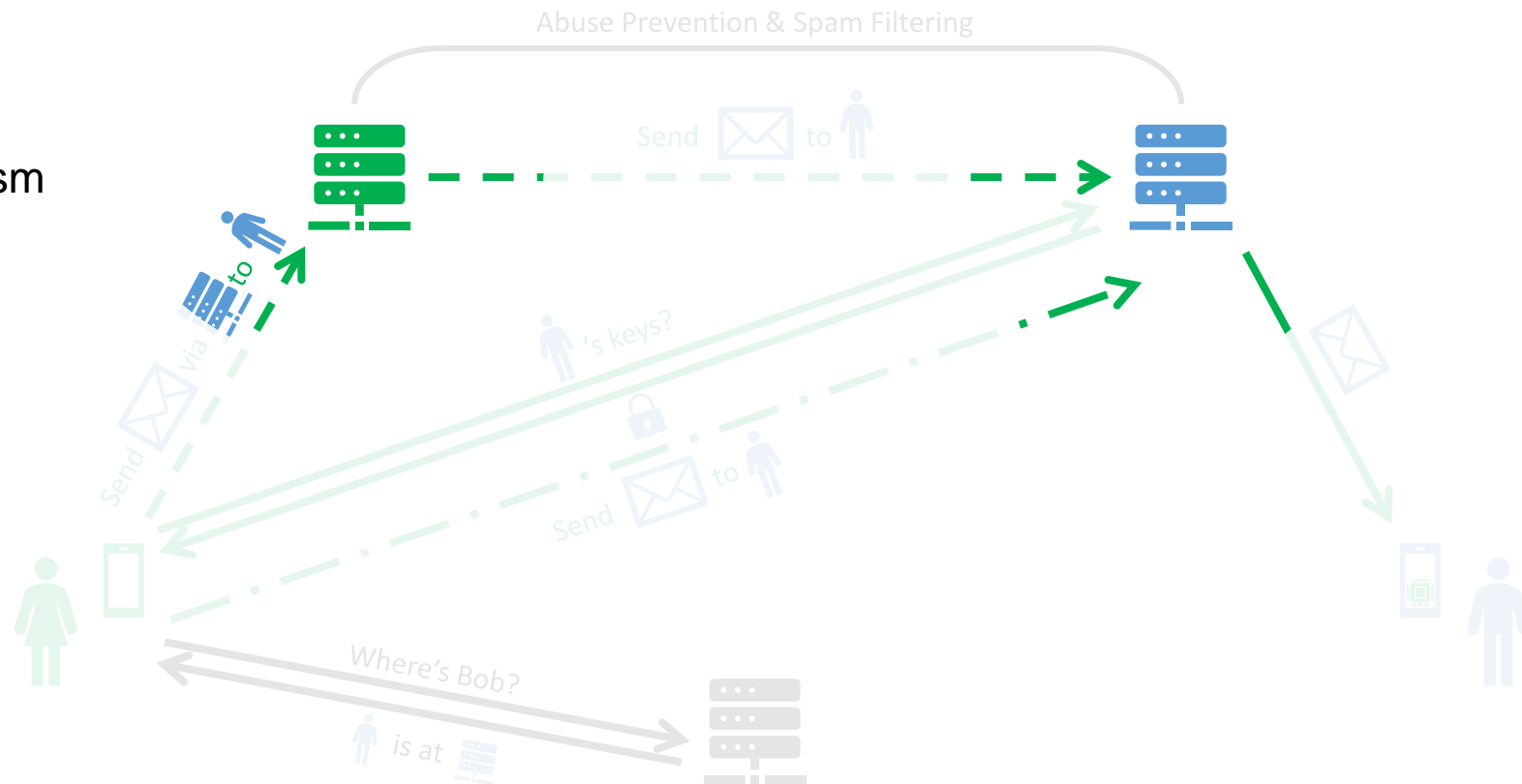


# Communication Protocol

## 1. Gatekeeper's Core Protocol + X

- API + Documentation
- Client: Library?
- Server:
  - Replicate key distribution
  - Forwarding service
  - Abuse prevention (interactively)
  - Reporting and blocking mechanism

## 2. Standardization of Protocol

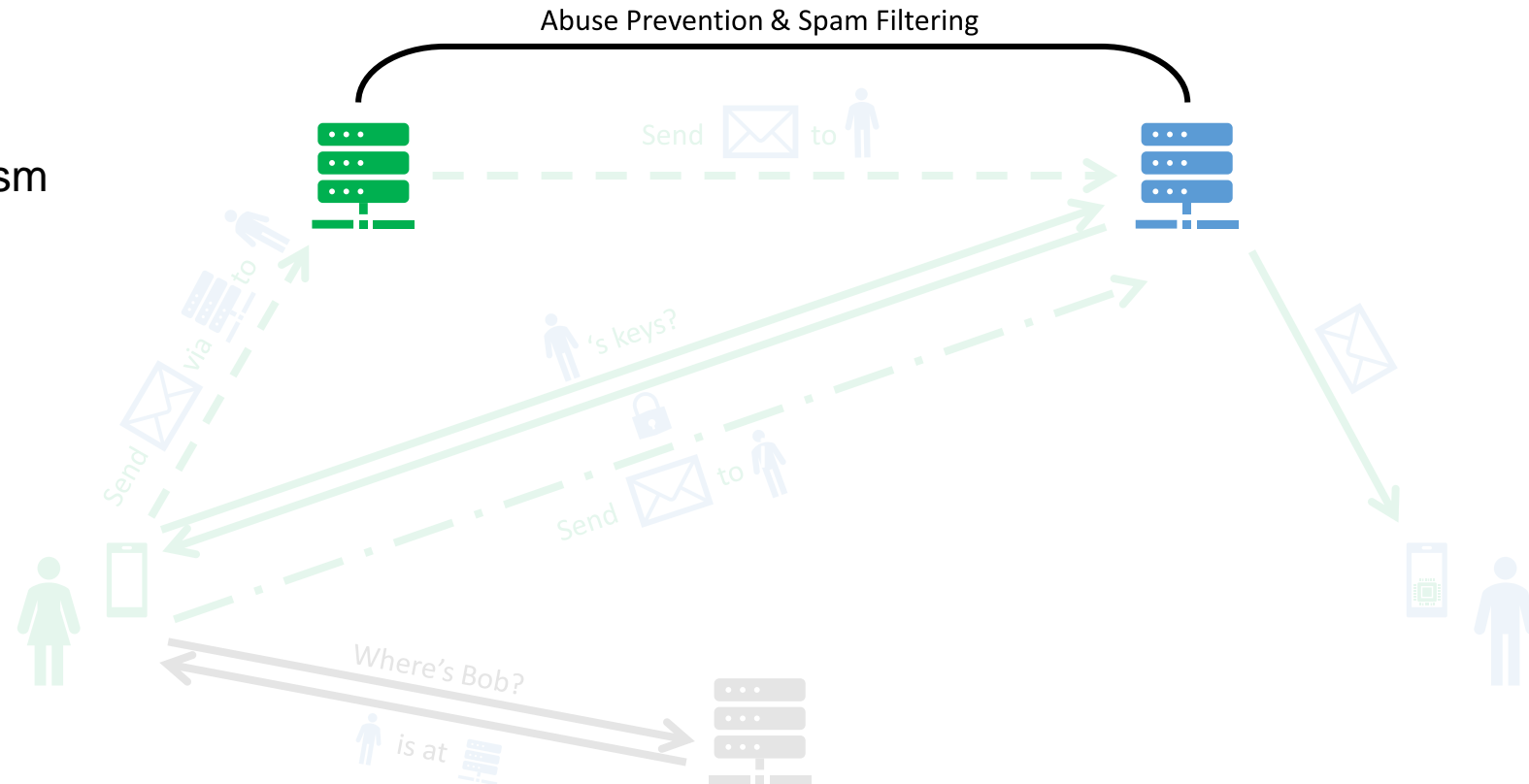


# Communication Protocol

## 1. Gatekeeper's Core Protocol + X

- API + Documentation
- Client: Library?
- Server:
  - Replicate key distribution
  - Forwarding service
  - Abuse prevention (interactively)
  - Reporting and blocking mechanism

## 2. Standardization of Protocol





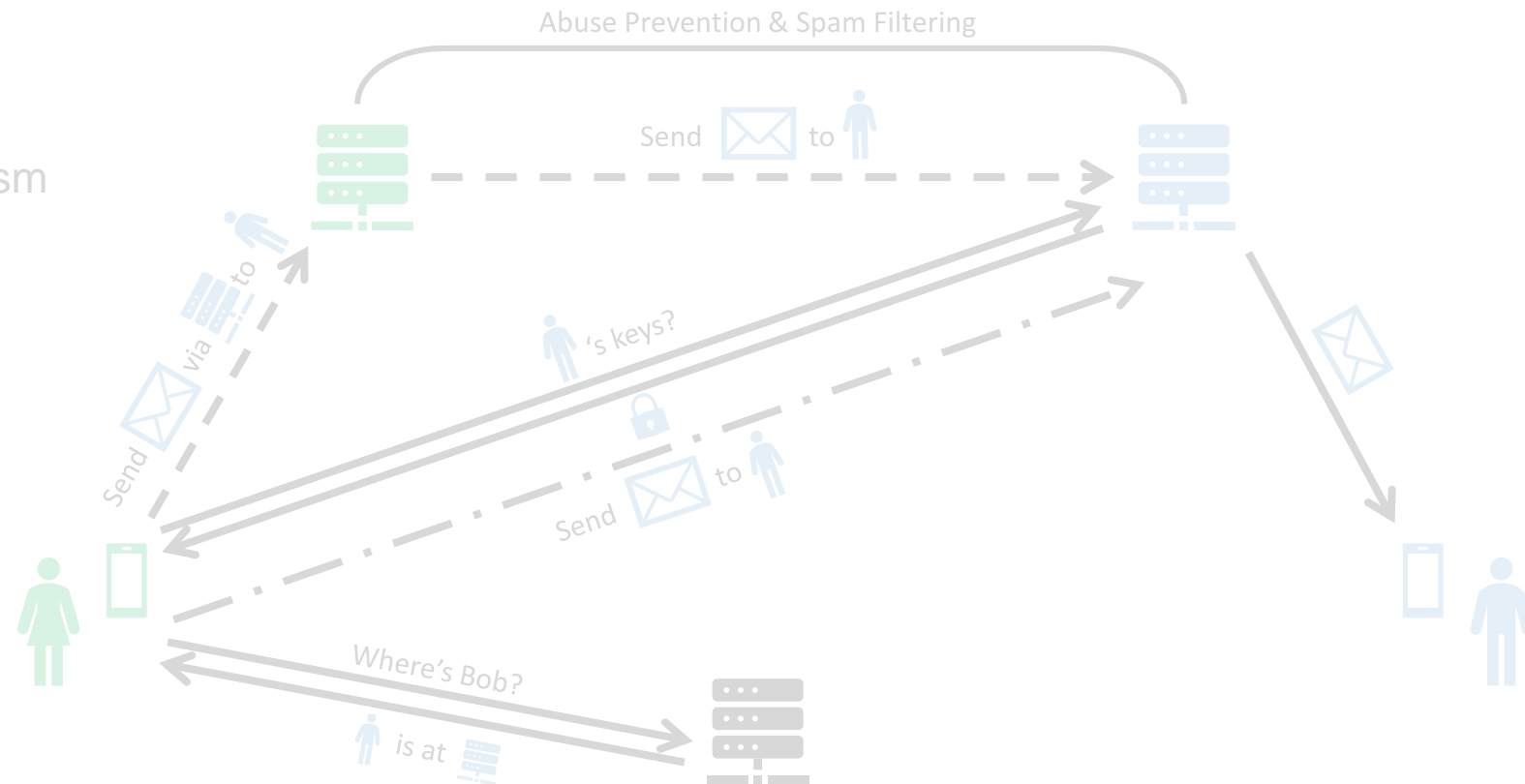
# Communication Protocol

## 1. Gatekeeper's Core Protocol + X

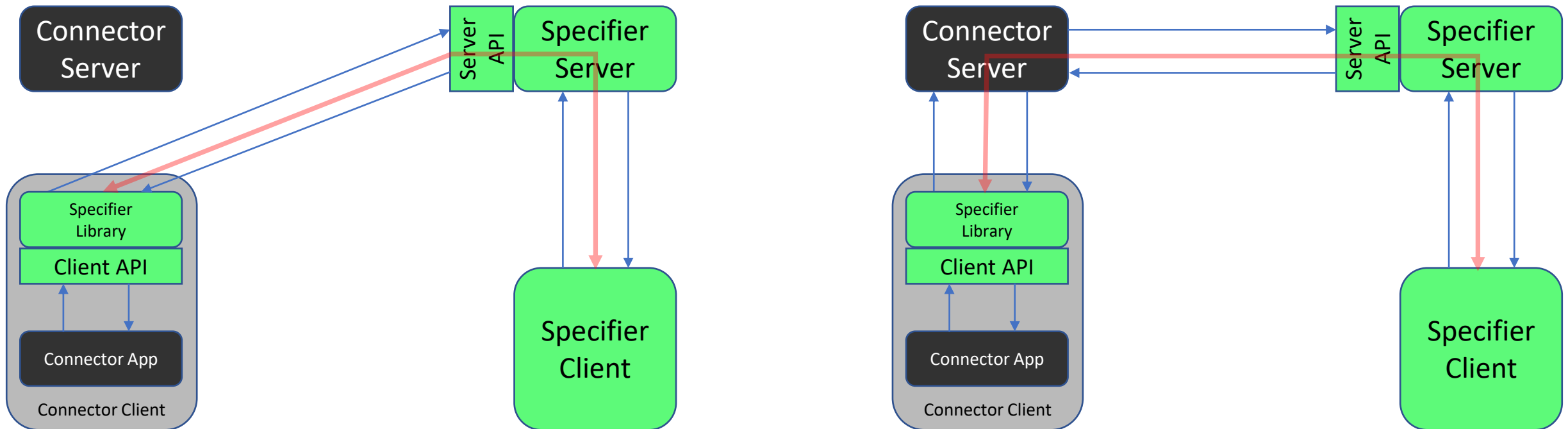
- API + Documentation
- Client: Library?
- Server:
  - Replicate key distribution
  - Forwarding service
  - Abuse prevention (interactively)
  - Reporting and blocking mechanism

## 2. Standardization of Protocol

- New protocol
  - $\geq$  Best gatekeeper
  - Simple standard update
  - Individual abuse prevention
- Don't start with solution

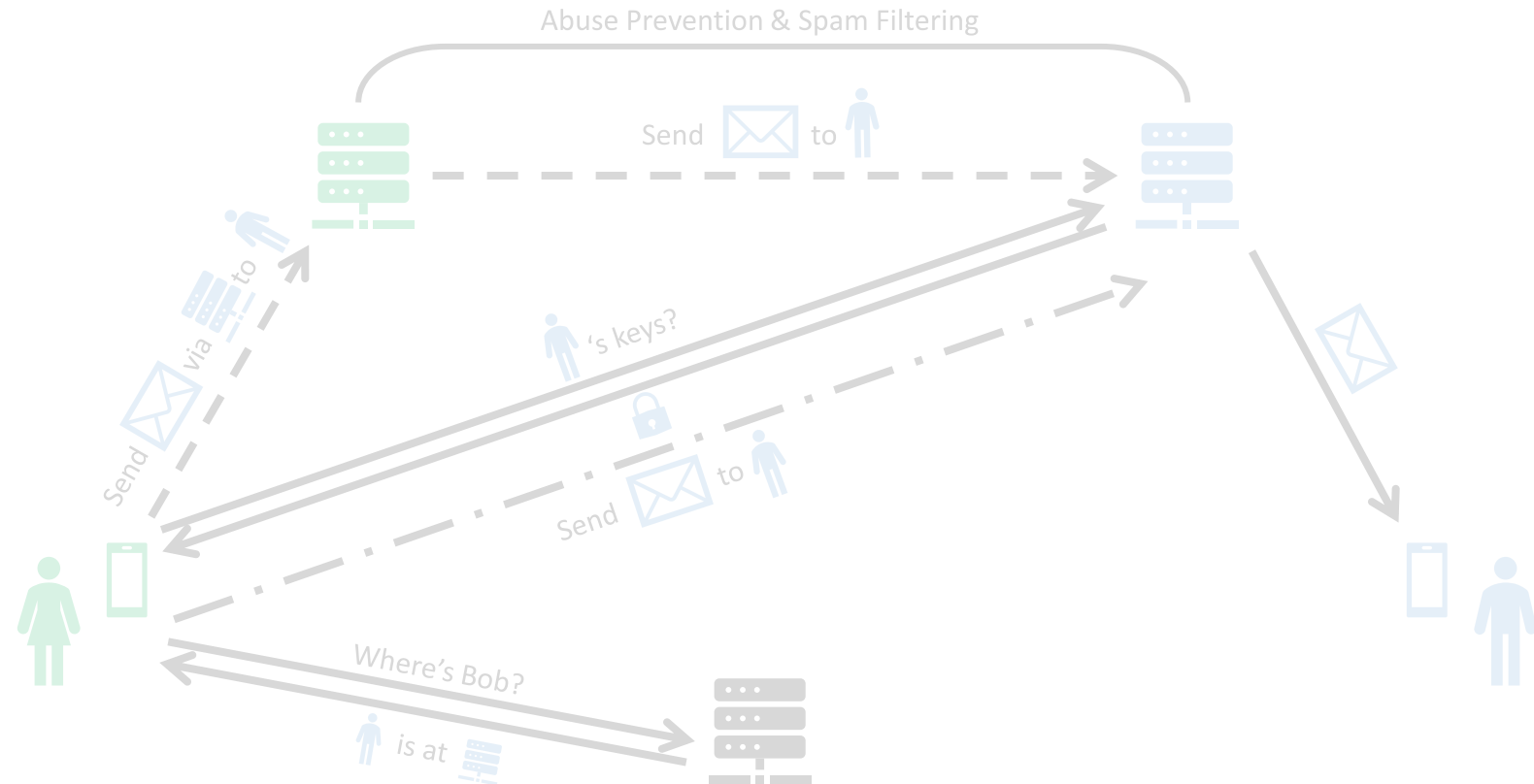


# Transmission Path



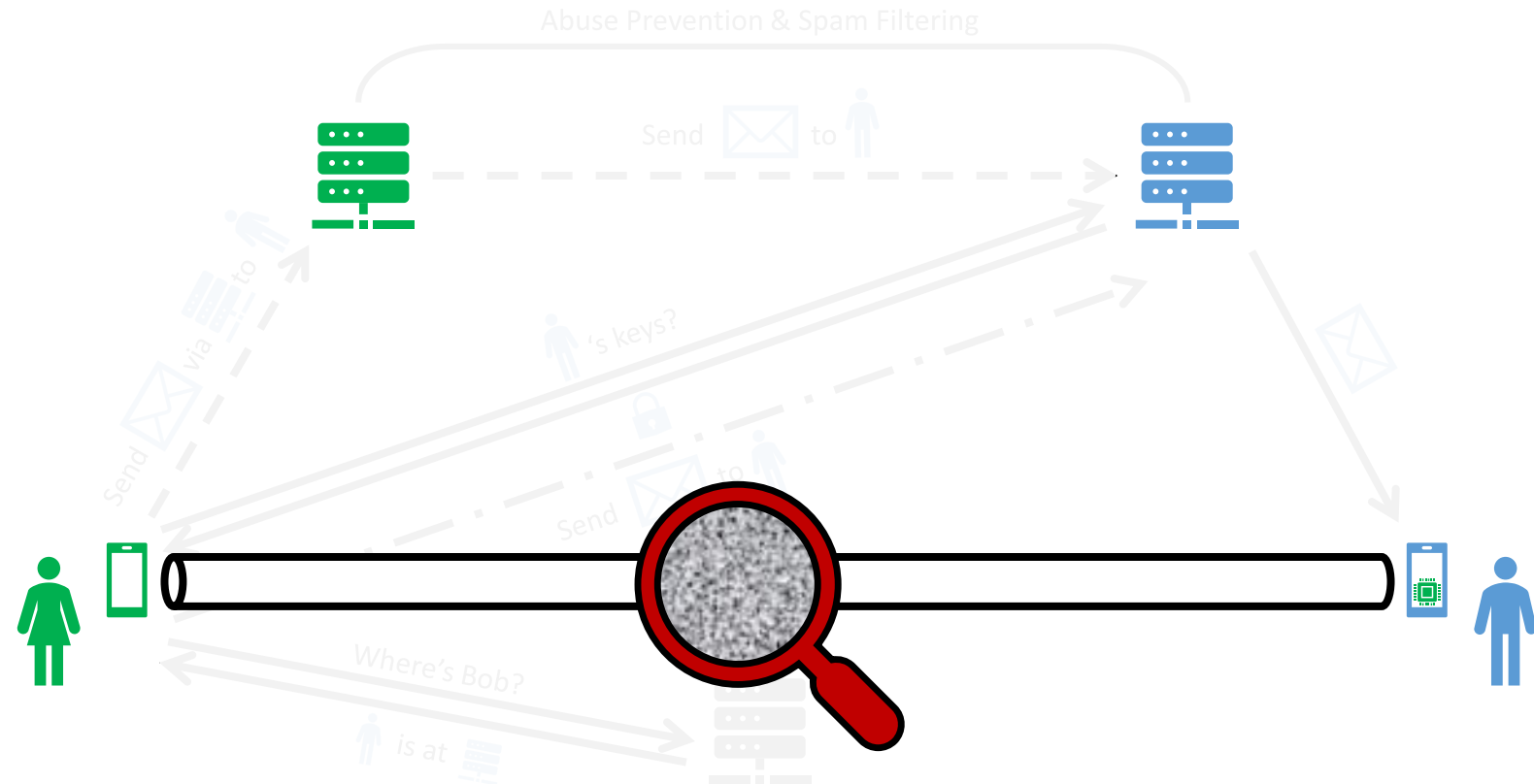
# Confidentiality, Privacy & Abuse Prevention

Aka. “+ X”



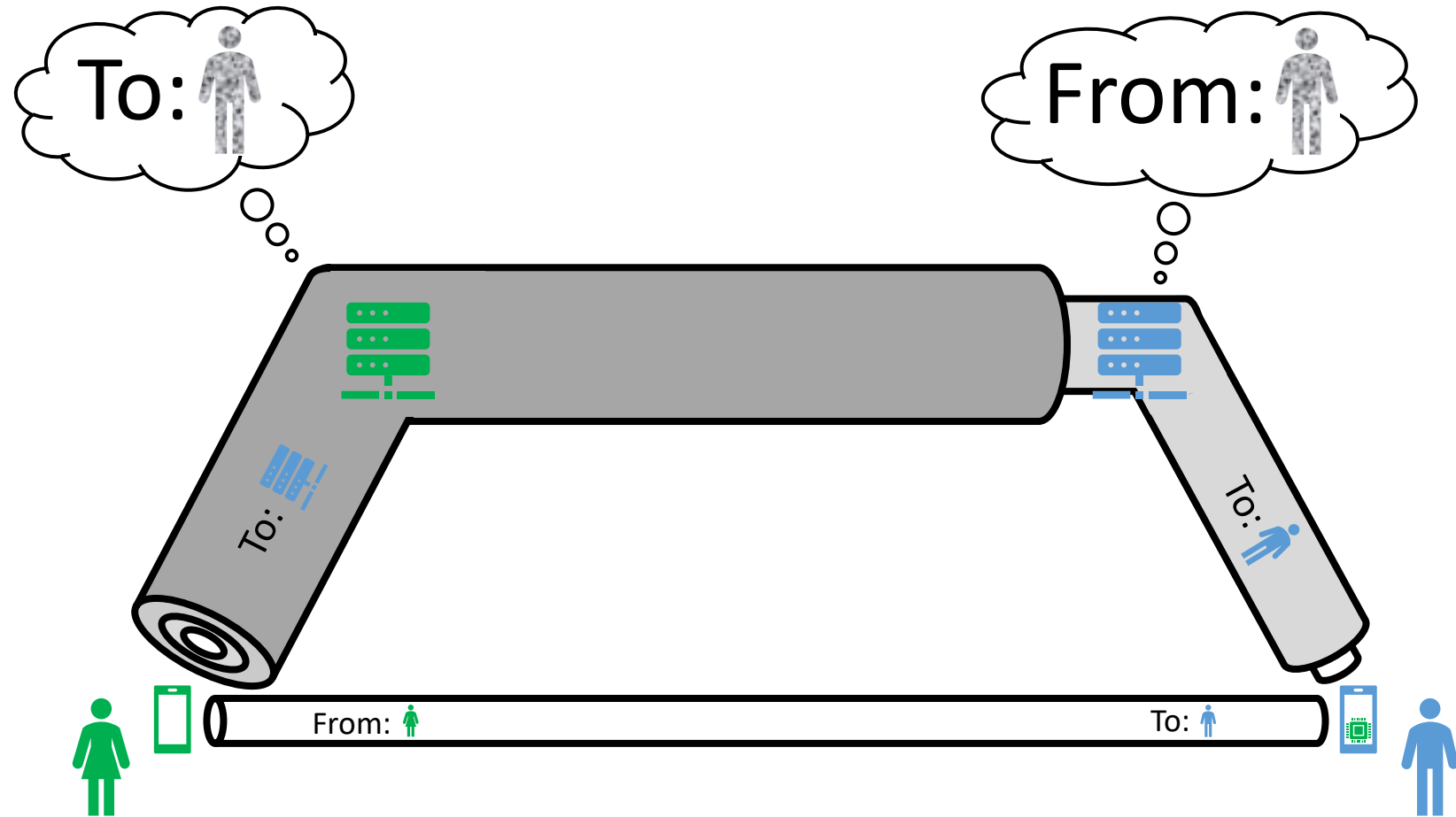
# Confidentiality, Privacy & Abuse Prevention

- Message Delivery
  - Confidential + Y



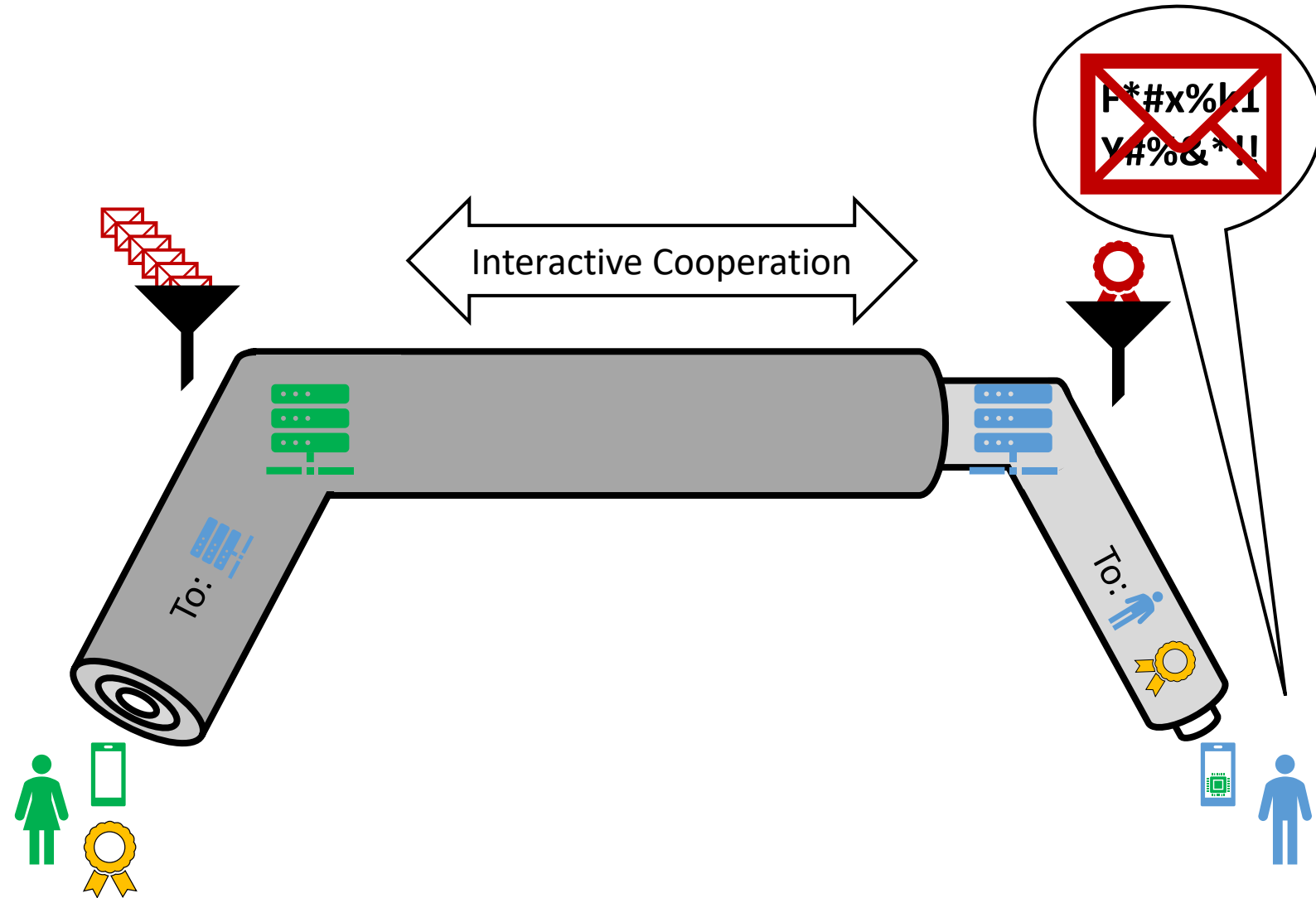
# Confidentiality, Privacy & Abuse Prevention

- Message Delivery
  - Confidential + Y
  - Private



# Confidentiality, Privacy & Abuse Prevention

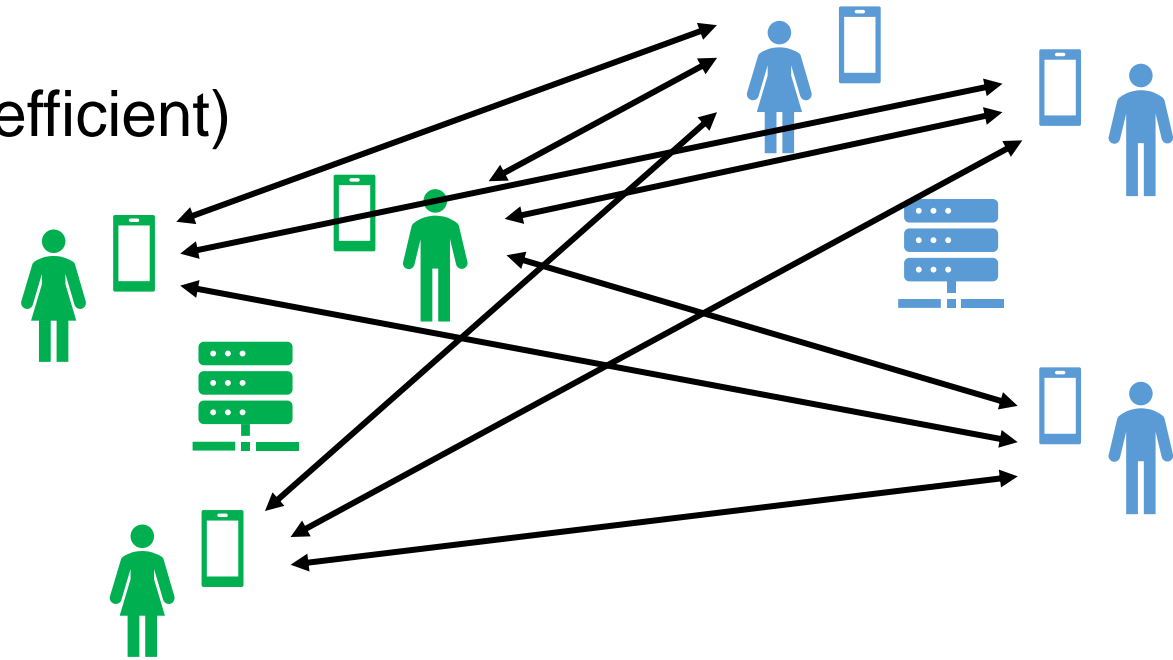
- Message Delivery
  - Confidential + Y
  - Private
- Abuse Reporting
- Effective User Blocking
  - For Individuals
  - From Platform
- Spam Filtering



# Group Messaging

## Alternatives:

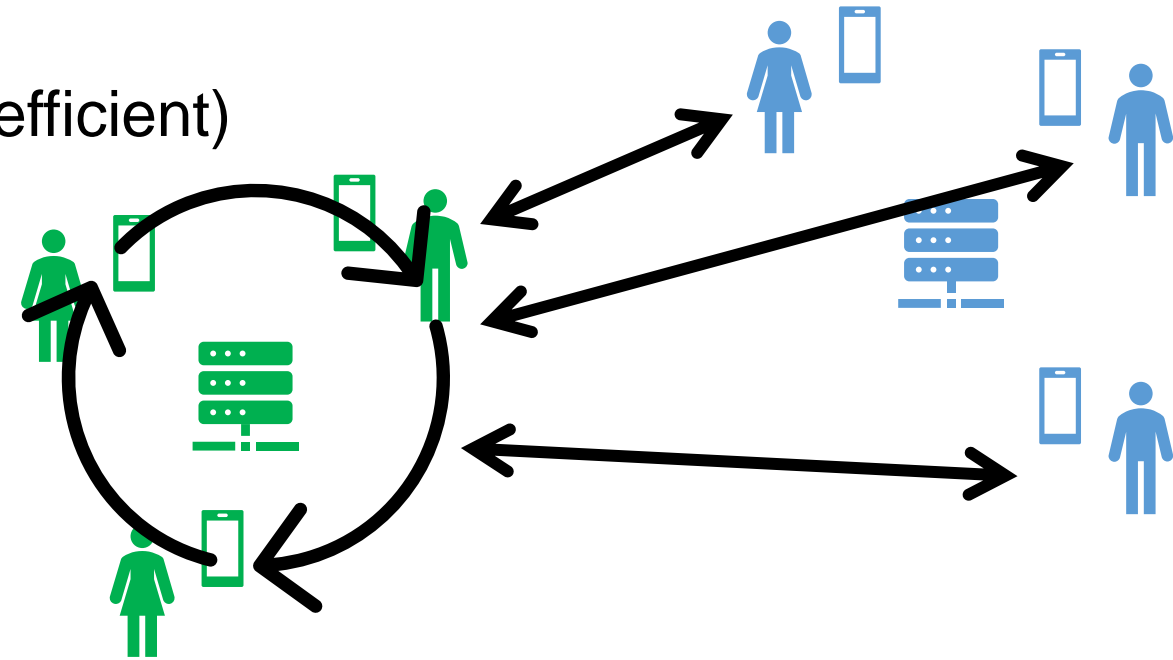
1. Via Pair-Wise Channels (Simple but Inefficient)
2. Gatekeeper's Core Protocol
3. Connect Providers' Subgroups
4. Standardize Group Protocol



# Group Messaging

## Alternatives:

1. Via Pair-Wise Channels (Simple but Inefficient)
2. Gatekeeper's Core Protocol
3. Connect Providers' Subgroups
4. Standardize Group Protocol

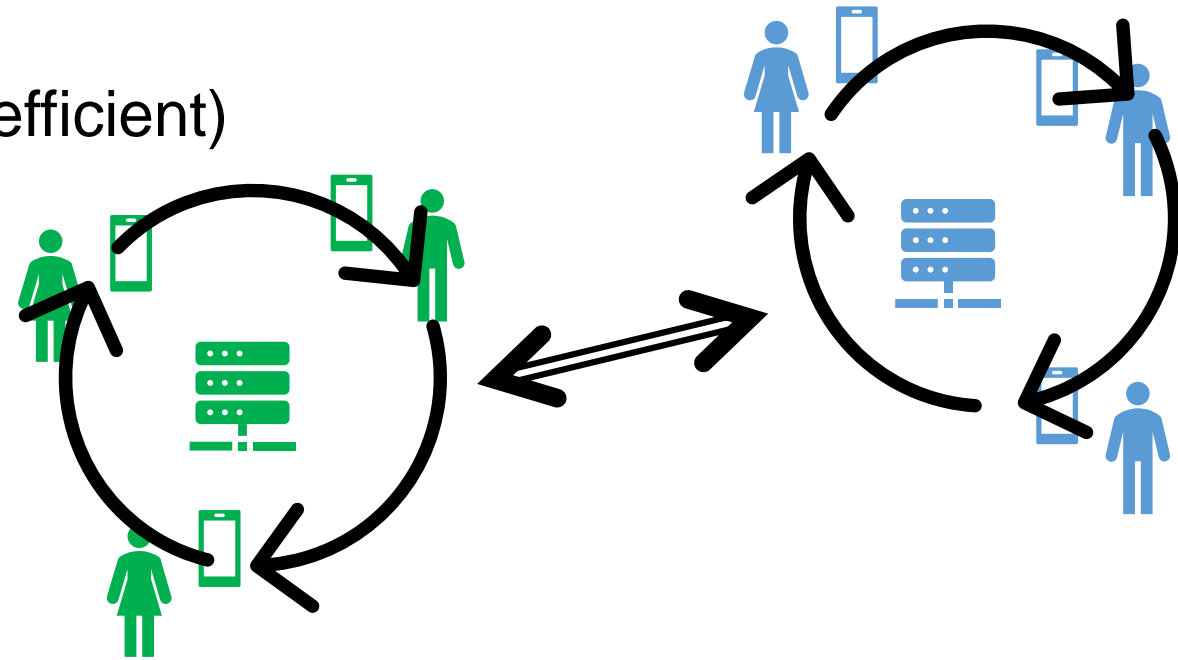




# Group Messaging

## Alternatives:

1. Via Pair-Wise Channels (Simple but Inefficient)
2. Gatekeeper's Core Protocol
3. Connect Providers' Subgroups
4. Standardize Group Protocol



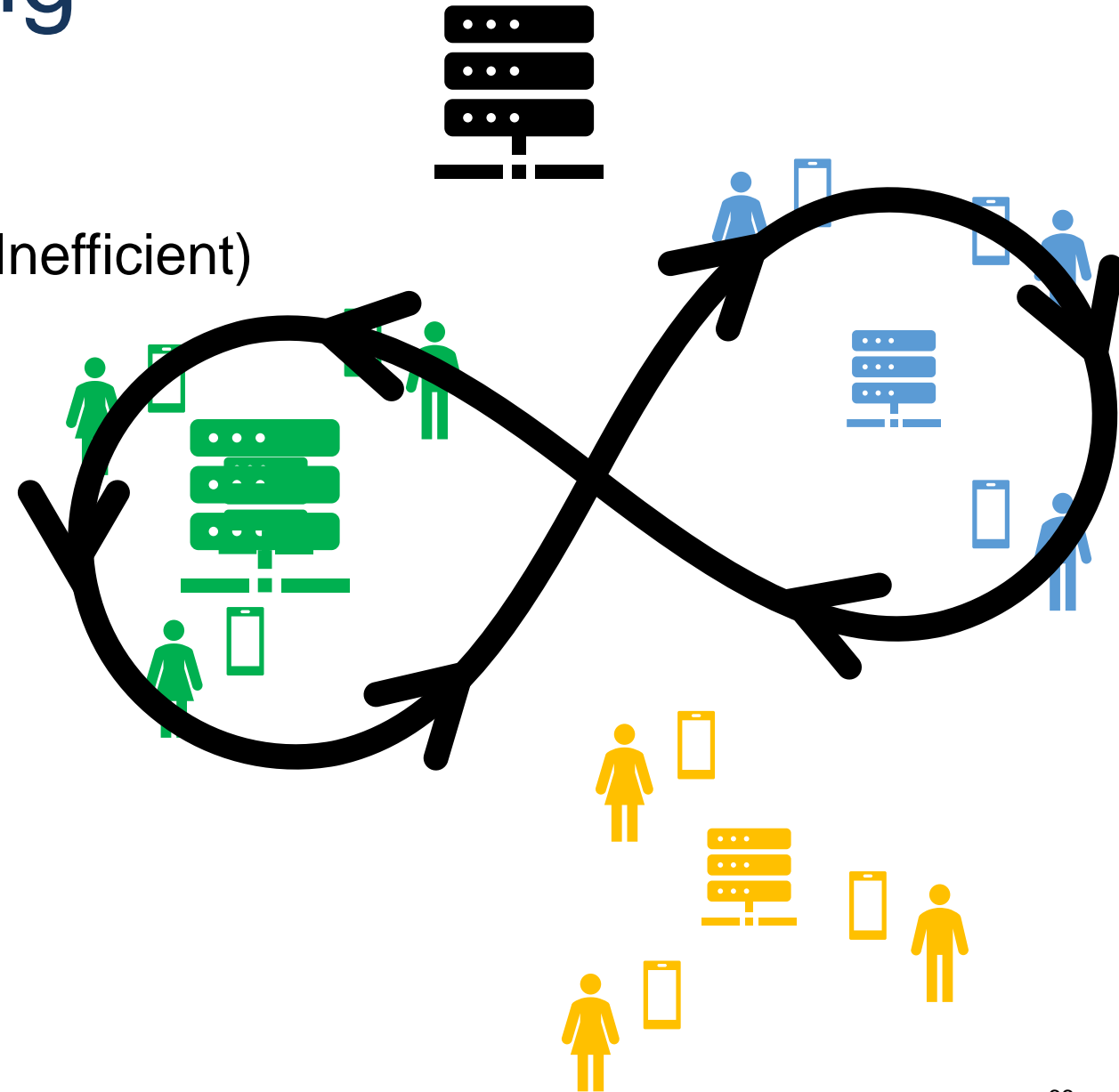
# Group Messaging

## Alternatives:

1. Via Pair-Wise Channels (Simple but Inefficient)
2. Gatekeeper's Core Protocol
3. Connect Providers' Subgroups
4. Standardize Group Protocol

## Problems and Solutions:

- Consistent Group Management:
  - Centralized?
  - Where?
- Multiple Gatekeepers & Non-Gatekeepers



# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# IETF MIMI

*„The More Instant Messaging Interoperability (MIMI) working group will specify the minimal set of mechanisms required to make modern Internet messaging services interoperable.“*

*„The working group will aim to achieve the strongest usable security and privacy properties for each targeted functional requirement.“*

The screenshot shows the IETF Datatracker website for the More Instant Messaging Interoperability (mimi) working group. The page includes a navigation bar with 'Datatracker', 'Groups', 'Documents', 'Meetings', 'Other', and 'User' menus. A search bar and 'Sign in' button are also present. The main content area features a title 'More Instant Messaging Interoperability (mimi)' and a sub-navigation menu with 'About', 'Documents', 'Meetings', 'History', 'Photos', 'Email expansions', and 'List archive'. Below this is a table with the following data:

WG	Name	More Instant Messaging Interoperability
	Acronym	mimi
	Area	Applications and Real-Time Area ( <a href="#">art</a> )
	State	Active
	Charter	<a href="#">charter-ietf-mimi-01</a> <span>Approved</span>
	Document dependencies	<a href="#">Show</a>
	Additional resources	<a href="#">GitHub Organization</a>
Personnel	Chairs	<a href="#">Alissa Cooper</a> , <a href="#">Tim Geoghegan</a>
	Area Director	<a href="#">Murray Kucherawy</a>
Mailing list	Address	<a href="mailto:mimi@ietf.org">mimi@ietf.org</a>
	To subscribe	<a href="https://www.ietf.org/mailman/listinfo/mimi">https://www.ietf.org/mailman/listinfo/mimi</a>
	Archive	<a href="https://mailarchive.ietf.org/arch/browse/mimi/">https://mailarchive.ietf.org/arch/browse/mimi/</a>
Chat	Room address	<a href="https://zulip.ietf.org/#narrow/stream/mimi">https://zulip.ietf.org/#narrow/stream/mimi</a>

Below the table, there is a section titled 'Charter for Working Group' with the following text:

The More Instant Messaging Interoperability (MIMI) working group will specify the minimal set of mechanisms required to make modern Internet messaging services interoperable. Over time, messaging services have achieved widespread use, their feature sets have broadened, and their adoption of end-to-end encryption (E2EE) has grown, but the lack of interoperability between these services continues to create a suboptimal user experience. The standards produced by the MIMI working group will allow for E2EE messaging services for both consumer and enterprise to interoperate without undermining the security guarantees that they provide. The working group will aim to achieve the strongest usable security and privacy properties for each targeted functional requirement.

# Standardization Goal

*Re-use existing standards whenever possible*

# Identities, Key Distribution, Trust

- Standardize naming scheme similar to e-mail:

localname@interop.whatsapp.com



Local username  
in WhatsApp

Standardized name of IM provider  
Most flexible: DNS domain

# Identities, Key Distribution, Trust

- Standardize initial authentication:
  - IM providers use long-lived signature or X3DH keys to authenticate users
  - Signature key can be adapted to any authentication scheme
    - If signatures are used to authenticate users, use them directly
    - If X3DH is used, sign a long-lived X3DH key of the user
- Standardize signature schemes for key distribution and Trust establishment

# Text messaging

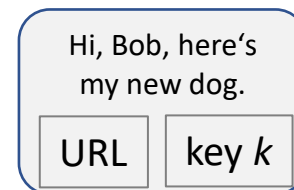
IM Protocol	Two-Party	Group	Real Time
Signal	Double Ratchet ( <b>DR</b> )	<b>DR</b>	WebRTC
WhatsApp	<b>DR</b>	Sender Key ( <b>SK</b> )	SRTP
Facebook Messenger	<b>DR</b> with Message Franking	<b>SK</b> with Message Franking	Undocumented
Wire	Proteus ( $\approx$ <b>DR</b> ; diff. AE)	Proteus ( $\approx$ <b>DR</b> )	SRTP
Matrix	Olm ( $\approx$ <b>DR</b> ; diff. KDF)	Megolm ( $\approx$ <b>SK</b> )	WebRTC
iMessage	Public-key encryption	Public-key encryption	SRTP
Telegram	MTPProto	Unencrypted	MTPProto

- Double Ratchet de facto standard for key management
  - many details differ and need to be standardized
- MLS may become an alternative once it is deployed



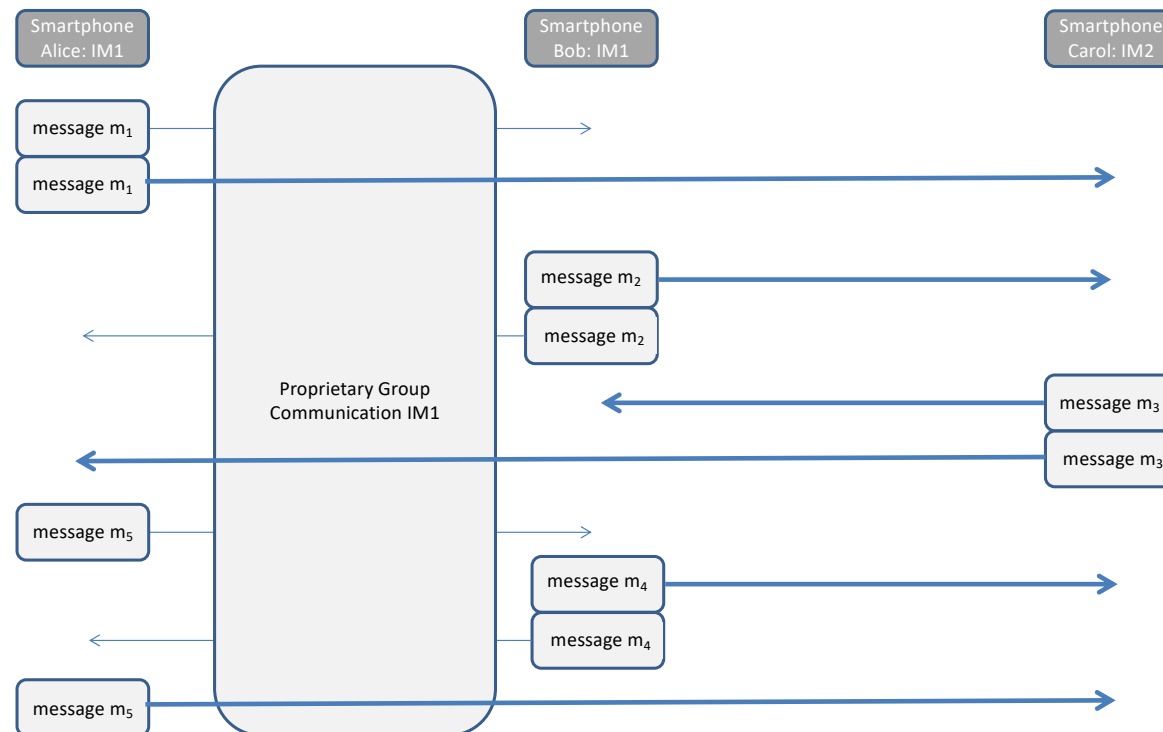
# File transfer

- General mechanism similar across IM applications
- Need for standardization:
  - Data format of encrypted files, e.g., JSON Web Encryption
  - Data format for download URL and decryption key in text message



# Group Messaging

- MLS, once it is successfully deployed
- Individual text message channels for competitor members in gatekeeper groups



# Real-Time Communication

- IETF WebRTC to enable browser-to-browser audio and video calls
  - Signaling channel over Web Application
  - Key exchange via DTLS
  - Encrypted communication via SRTP
- Adaption to IM (already done by IM providers)
  - Signaling and key exchange can be done via text chats
  - SRTP seems a good choice for encrypted communication

# Summary of Standardization

- Recommended standardization (short term)
  - Naming scheme for IM clients
  - Initial authentication via digital signatures
- Optional standardization (long term)
  - E2EE for text messages (double ratchet or MLS)
  - File transfer (data formats)
  - Group messaging (double ratchet or MLS)
  - Real-time communications (SRTP plus standardization of signaling and key exchange)

# Agenda

- Instant Messaging: Shared Cryptographic Mechanisms
  - Authenticated Encryption
  - Key Agreement
- Instant Messaging: Novel Cryptographic Mechanisms
  - Text messaging
  - File transfer
  - Group communication
  - Real-time communication
- Interoperable Instant Messaging
  - API Approach
  - Standardization Approach
- Summary

# Summary

- E2EE Confidentiality & Privacy:
  - Required by DMA!
  - Practically achievable through APIs and/or standardization ✓
- Gatekeeper API vs. IM Standard
  - Standardization:
    - Slow
    - Equal overhead for all parties
  - Gatekeeper API:
    - Fast, agile
    - Cryptographic library provided by the gatekeeper
  - Approach
    - Start with API, standardize only basic functionality

# Summary: Approach

- Minimal initial standardization: Plaintext formats, ID scheme
- Gatekeeper libraries
- Clear documentation
- Gradual additional standardization
- Agile, flexible standard & space for individual protocols

## Missing aspects:

- Data migration
- Update mechanisms
- Adversarial providers (e.g., complicate interop)