Security of Push Messaging

Jörg Schwenk

Ruhr University Bochum
Hackmanit GmbH

Push Messaging

Push Messaging

SMS

E-Mail

Instant Messaging (IM)

Push Messaging

SMS

E-Mail

Instant Messaging (IM)

Bad Press

Why Johnny Can't Encrypt:
A Usability Evaluation of PGP 5.0

Alma Whitten
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alma@cs.cmu.edu

J. D. Tygar[1]
EECS and SIMS
University of California
Berkeley, CA 94720
tygar@cs.berkeley.edu

Why Johnny Still Can't Encrypt:
Evaluating the Usability of Email Encryption Software

Steve Sheng
Engineering and Public Policy
Carnegie Mellon University
shengx@cmu.edu

Levi Broderick
Electrical and Computer Engineering
Carnegie Mellon University
lpb@ece.cmu.edu

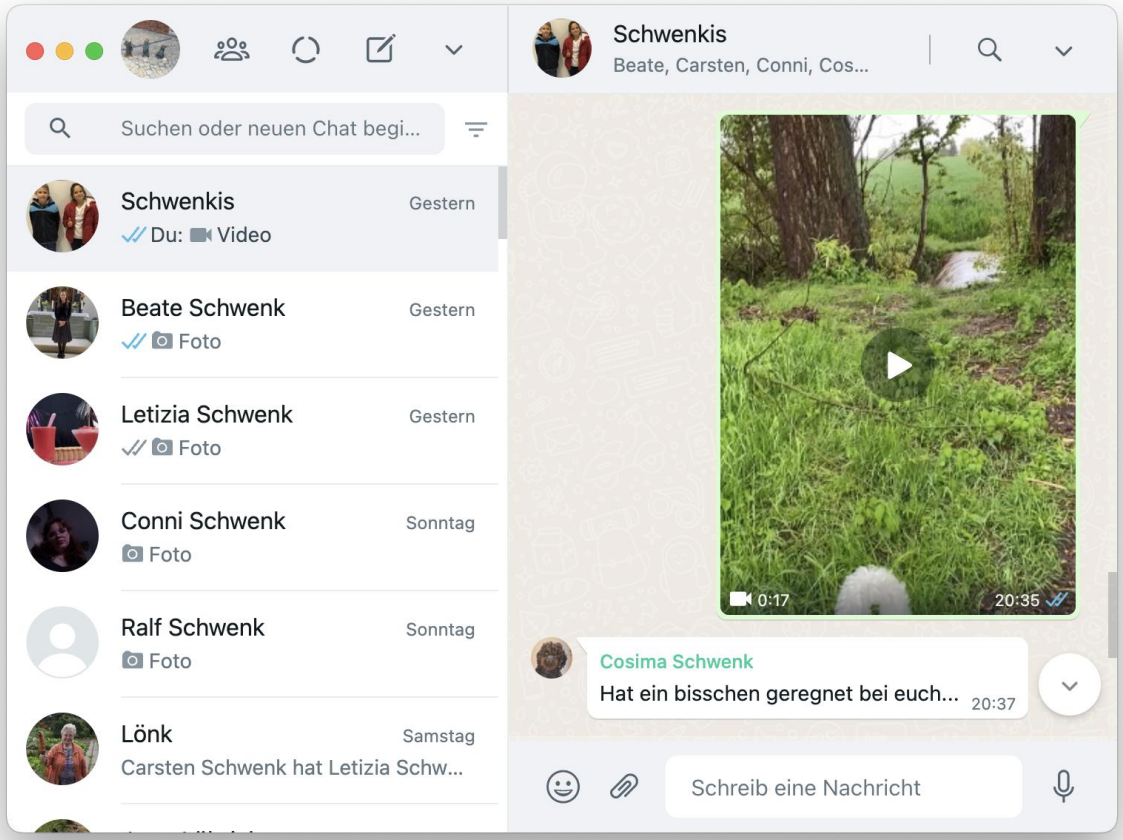
Colleen Alison Koranda
HCI Institute
Carnegie Mellon University
ckoranda@andrew.cmu.edu

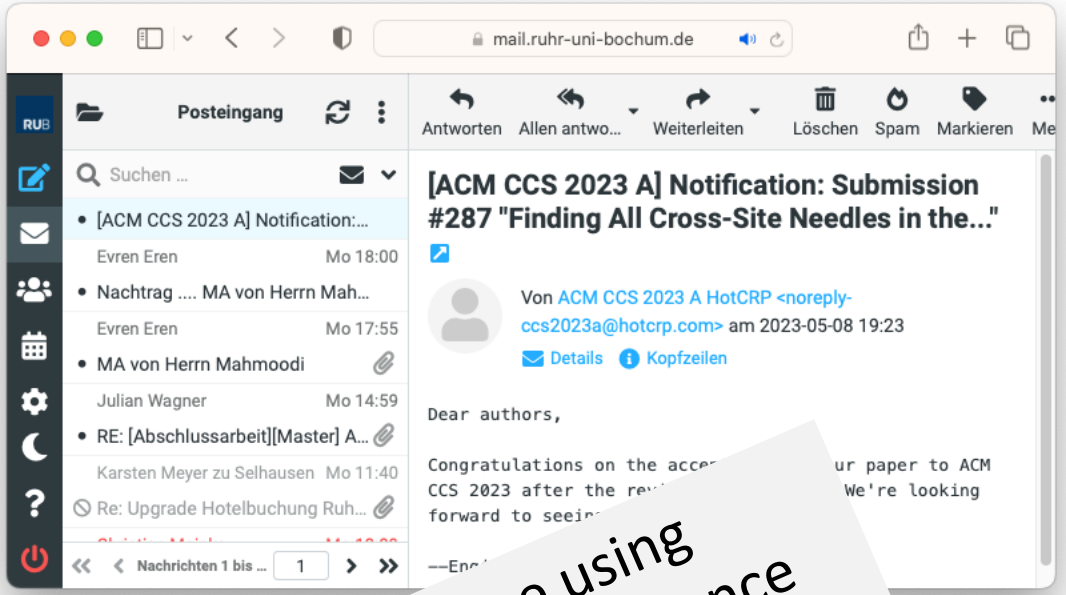
Jeremy J. Hyland
Heinz School of Public Policy and Management
Carnegie Mellon University
jhyland@andrew.cmu.edu

IEEE S&P 2022

https://www.newscientist.com/article/2289747-almost-no-one-encrypts-their-emails-because-it-is-too-much-of-a-hassle/#:~:text=Just%200.06%20per%20cent%20of,January%201994%20and%20July%202021.

Static vs. Dynamic Groups

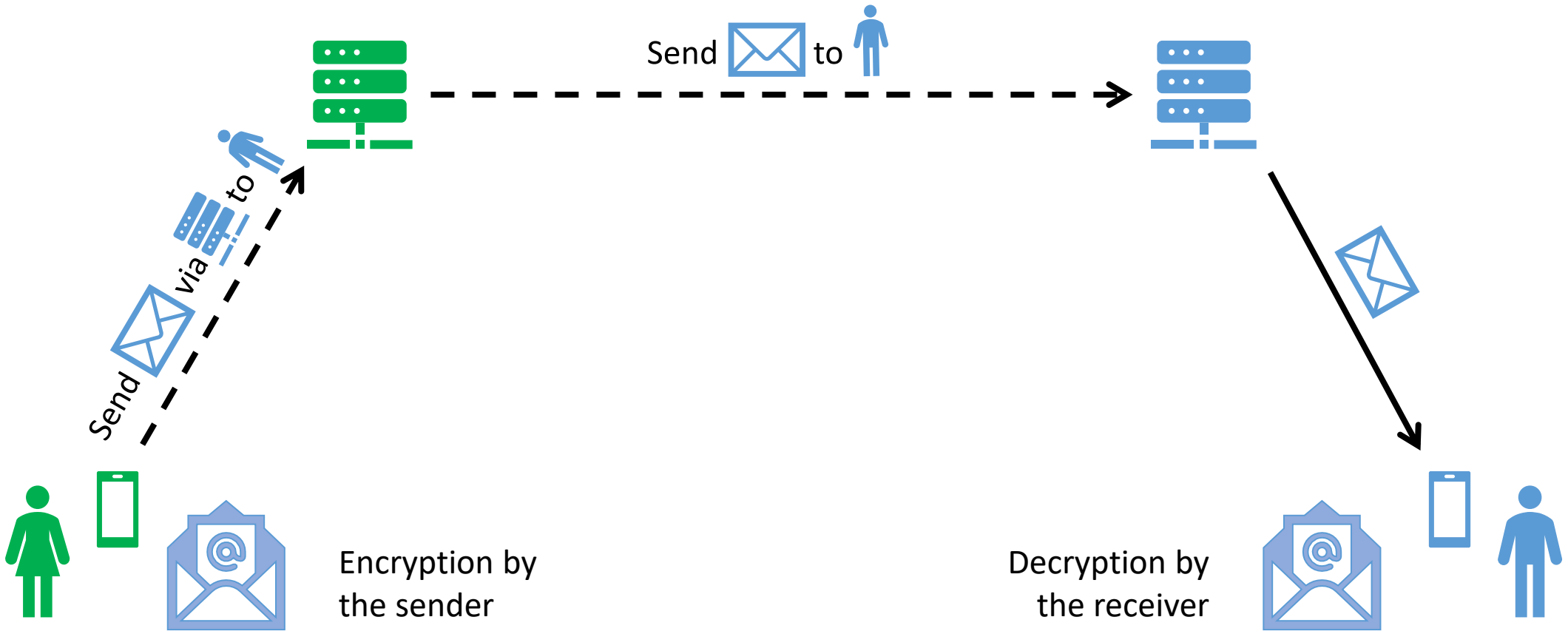




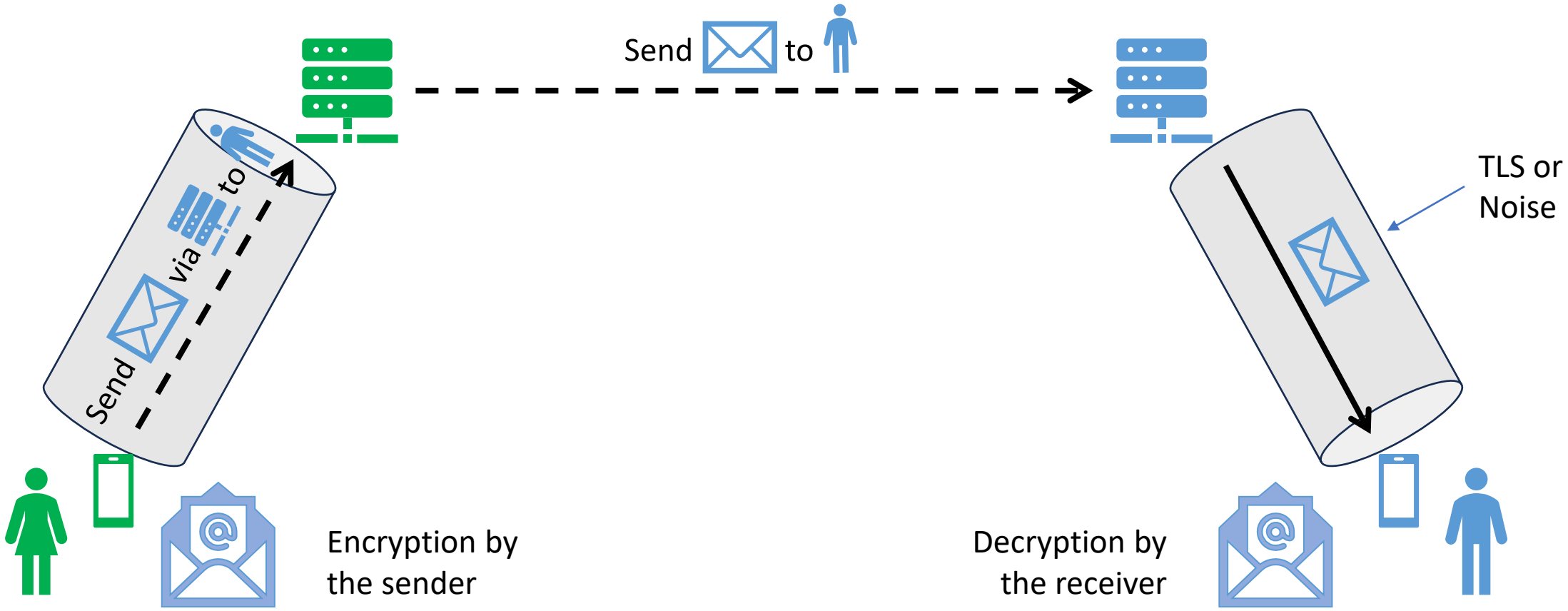
Can you imagine using WhatsApp for conference notifications?

E2EE in Push Messaging

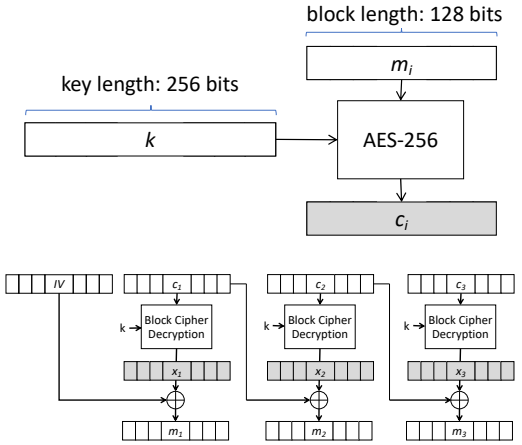
End-to-End-Encryption



End-to-End-Encryption

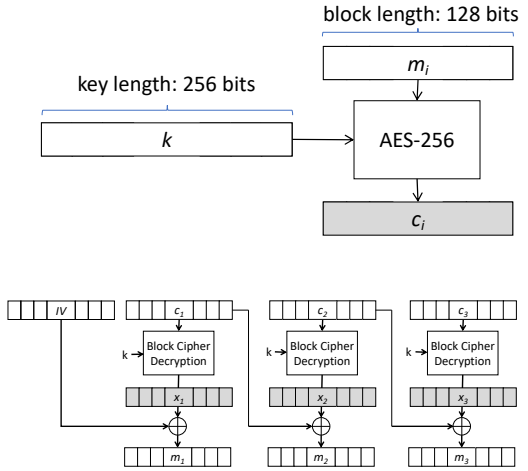


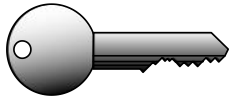
For E2EE we need:



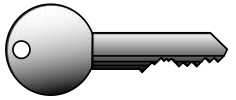
encryption algorithms
(AES, ChaCha20)

encryption modes
(authenticated encryption)





symmetric keys

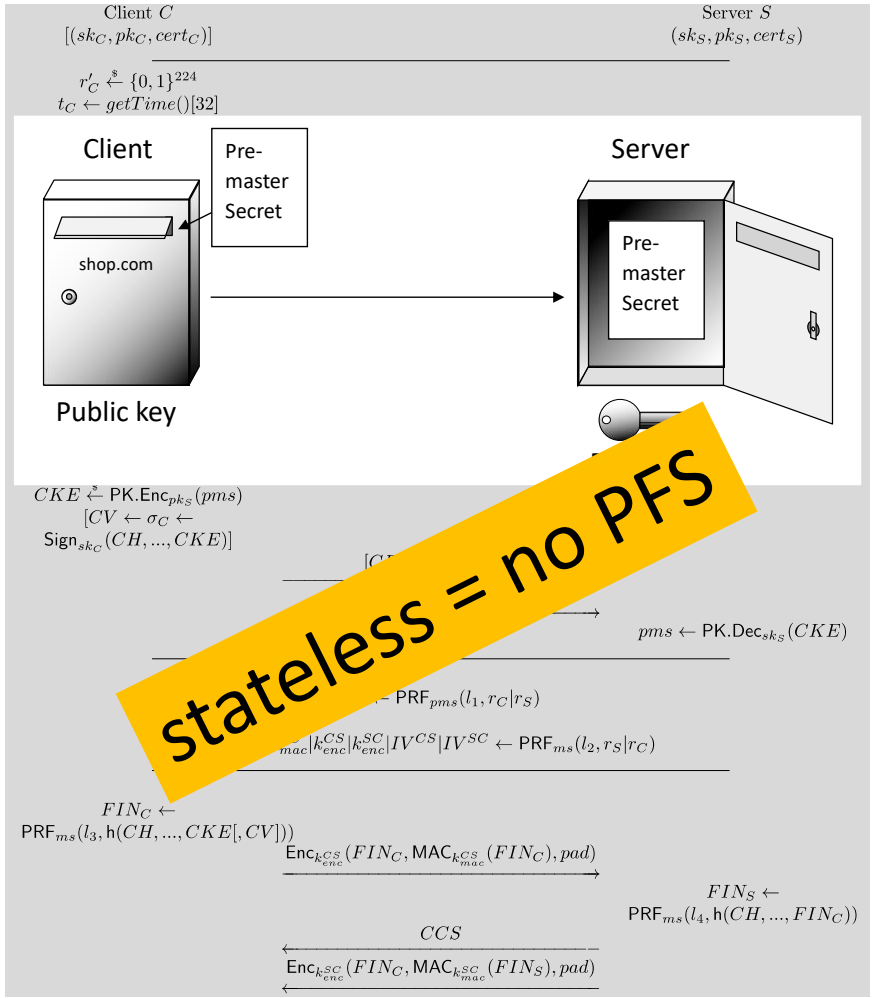


stateless via
Public Key Encryption (PKE)
Key Encapsulation Mechanisms (KEM)

stateful via
Authenticated Key Agreement
Ratcheting

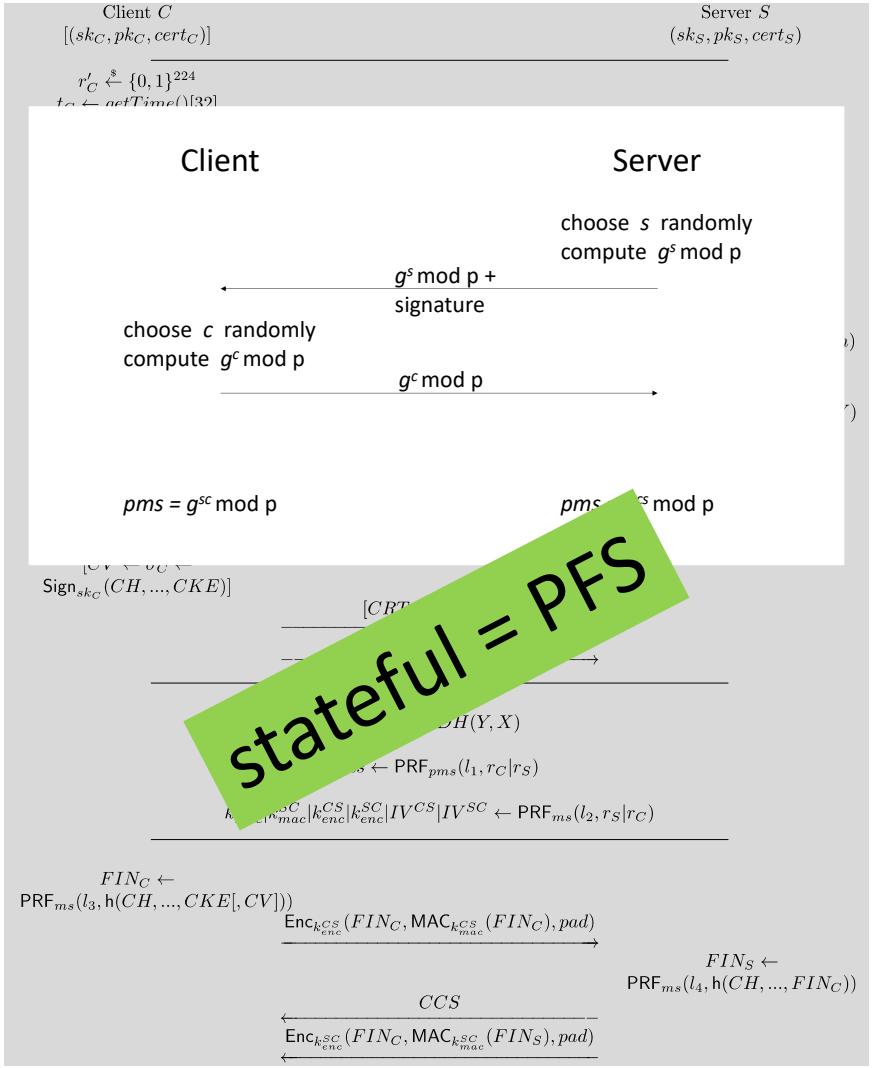
TLS 1.2

TLS-RSA



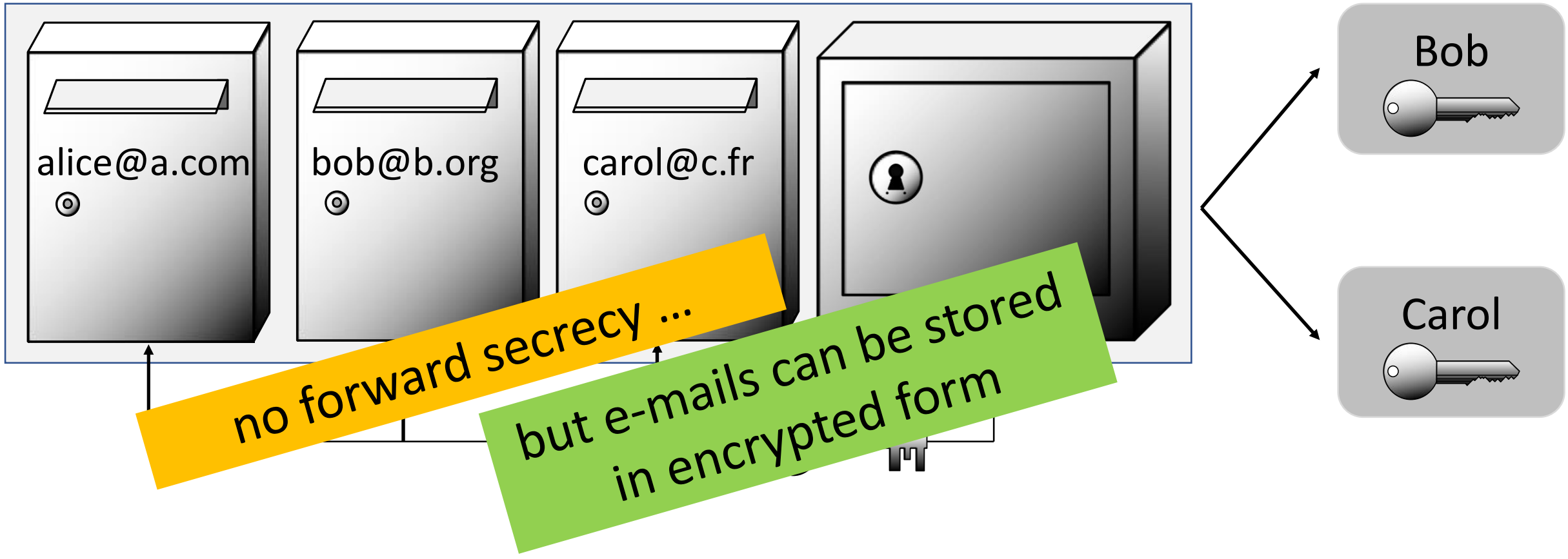
stateless = no PFS

TLS-DHE

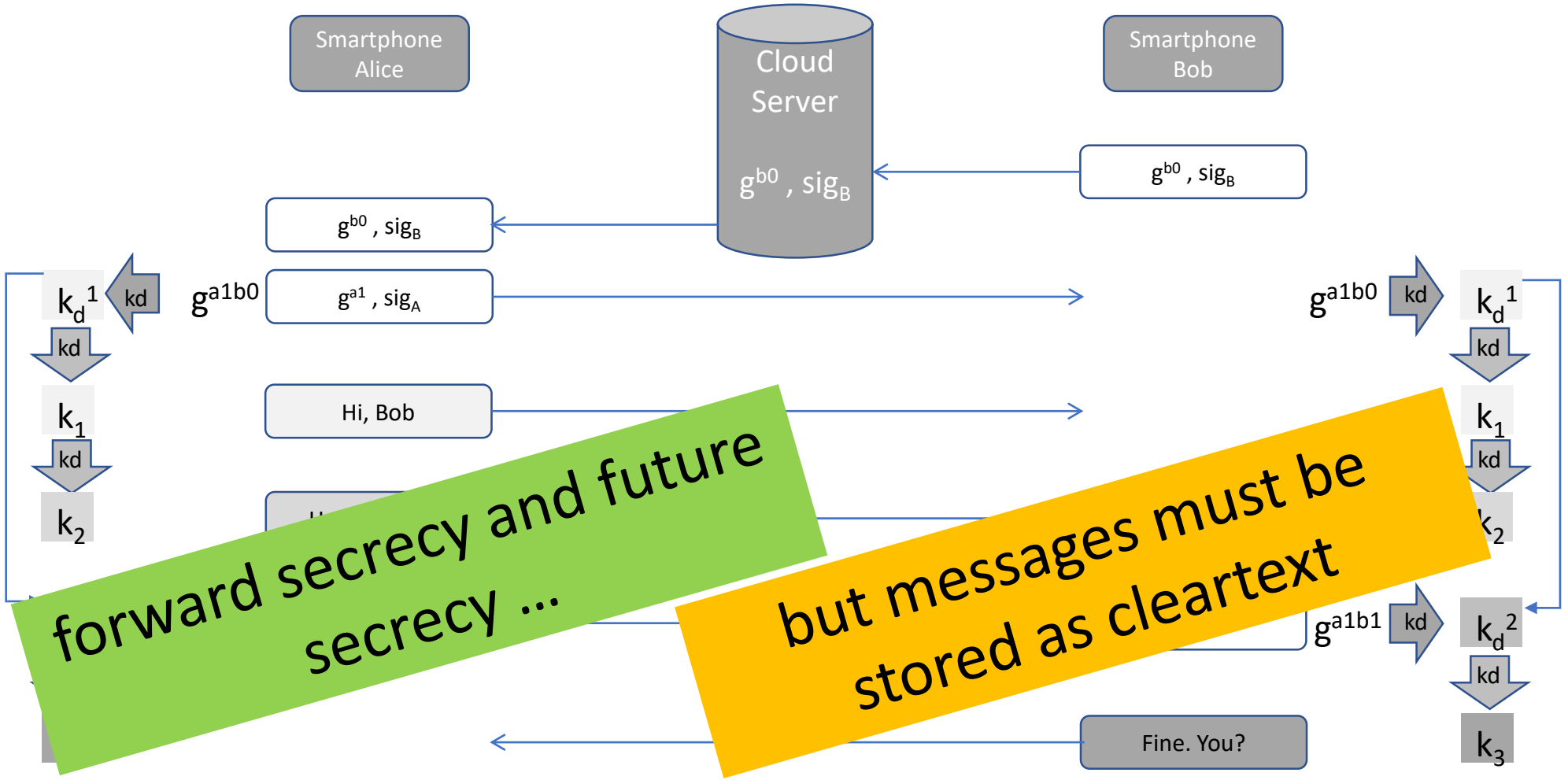


stateful = PFS

E-Mail Encryption is stateless



Double Ratchet – continuous DHKE



Instant Messaging Security

Overview of Messengers

IM Protocol	Two-Party	Group	Real Time
Signal	Double Ratchet (**DR**)	**DR**	WebRTC
WhatsApp	**DR**	Sender Key (**SK**)	SRTP
Facebook Messenger	**DR** with Message Franking	**SK** with Message Franking	Undocumented
Wire	Proteus (≈**DR**; diff. AE)	Proteus (≈**DR**)	SRTP
Matrix	Olm (≈**DR**; diff. KDF)	Megolm (≈**SK**)	WebRTC
Threema	NIKE	NIKE	
iMessage	Public-key encryption	Public-key encryption	SRTP
Telegram	MTProto	Unencrypted	MTProto

- Novel cryptographic mechanism: Double Ratchet (DR)
  - Forward Secrecy
  - Future Secrecy

Overview of Messengers

IM Protocol	Two-Party	Group	Real Time
Signal	Double Ratchet (**DR**)	**DR**	WebRTC
WhatsApp	**DR**	Sender Key (**SK**)	SRTP
Facebook Messenger	**DR** with Message Franking	**SK** with Message Franking	Undocumented
Wire	Proteus (≈**DR**; diff. AE)	Proteus (≈**DR**)	SRTP
Matrix	Olm (≈**DR**; diff. KDF)	Megolm (≈**SK**)	WebRTC
Threema	NIKE	NIKE	
iMessage	Public-key encryption	Public-key encryption	SRTP
Telegram	MTProto	Unencrypted	MTProto

IM Encryption: NIKE

Threema

NIKE: Security properties similar to secure E-Mail

- Private key compromise in NIKE: all messages to and from the client can be decrypted
- Other problems (see paper below, fixed)

Client A		Client B
$(sk, pk) = (a, A = g^a)$		$(sk, pk) = (b, B = g^b)$
$K_{A,B}$ X25519(a, B)		$K_{A,B}$ X25519(b, A)
n $\$\{0,1\}^{128}$		
nonces_A $\text{nonces}_A [\{n\}$		
ctxt $E_{K_{A,B}}(\text{ptxt}; n)$		
	src||dst||...||n||ctxt $\longrightarrow$	
		if $n 2$ nonces_B : discard
		ptxt $D_{K_{A,B}}(\text{ctxt}; n)$
		nonces_B $\text{nonces}_B [\{n\}$

Old Threema Protocol: NIKE

Three Lessons From Threema: Analysis of a Secure Messenger

Kenneth G. Paterson
Applied Cryptography Group, ETH Zurich

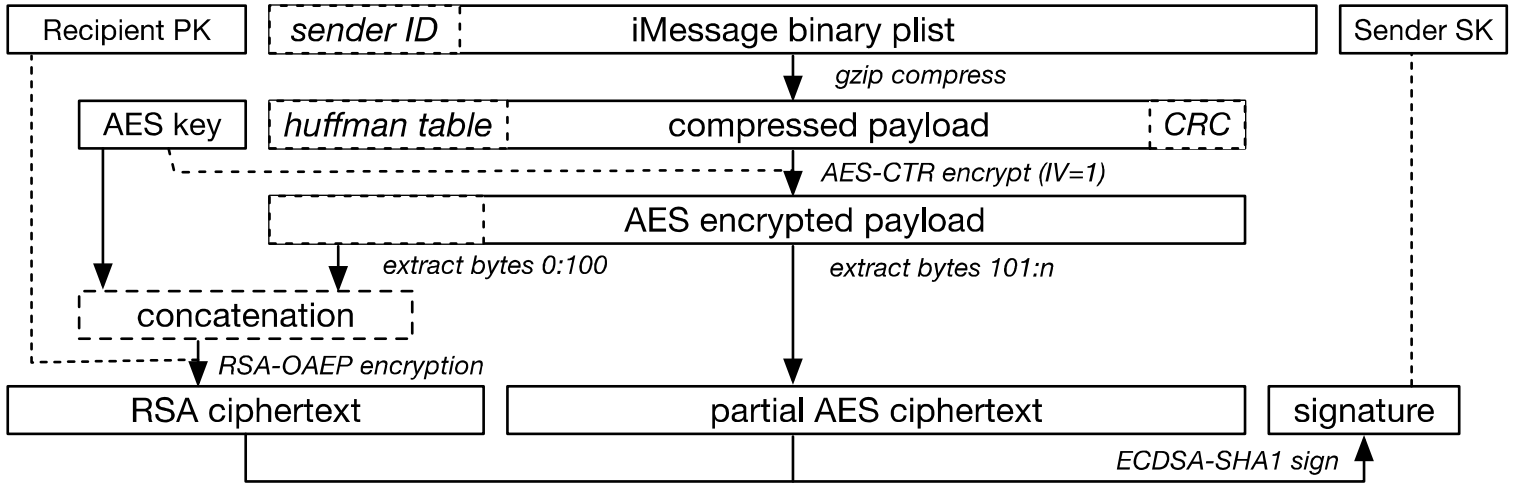
Matteo Scarlata
Applied Cryptography Group, ETH Zurich

Kien Tuong Truong
Applied Cryptography Group, ETH Zurich

USENIX 2022
https://breakingthe3ma.app

IM Encryption: PKE

iMessage



Modified Hybrid Encryption (PKE + AES-CTR) and digital signature

Dancing on the Lip of the Volcano:
Chosen Ciphertext Attacks on Apple iMessage

Christina Garman
Johns Hopkins University
cgarman@cs.jhu.edu

Matthew Green
Johns Hopkins University
mgreen@cs.jhu.edu

Gabriel Kaptchuk
Johns Hopkins University
gkaptchuk@cs.jhu.edu

Ian Miers
Johns Hopkins University
imiers@cs.jhu.edu

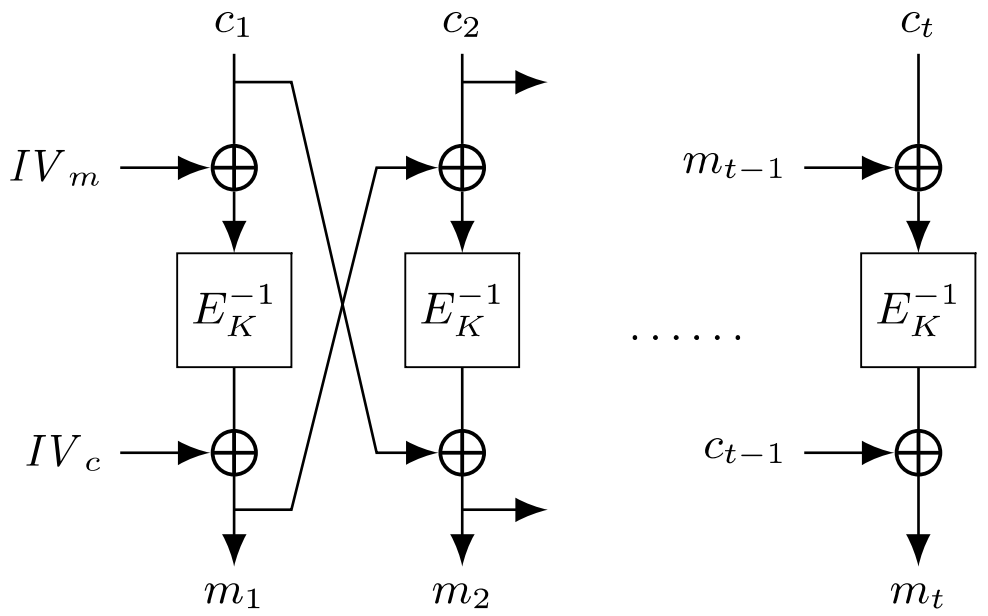
Michael Rushanan
Johns Hopkins University
micharu1@cs.jhu.edu

Usenix 2016

IM Encryption: DHKE?

Telegram

Only the symmetric encryption of MTProto was investigated



Four Attacks and a Proof for Telegram*

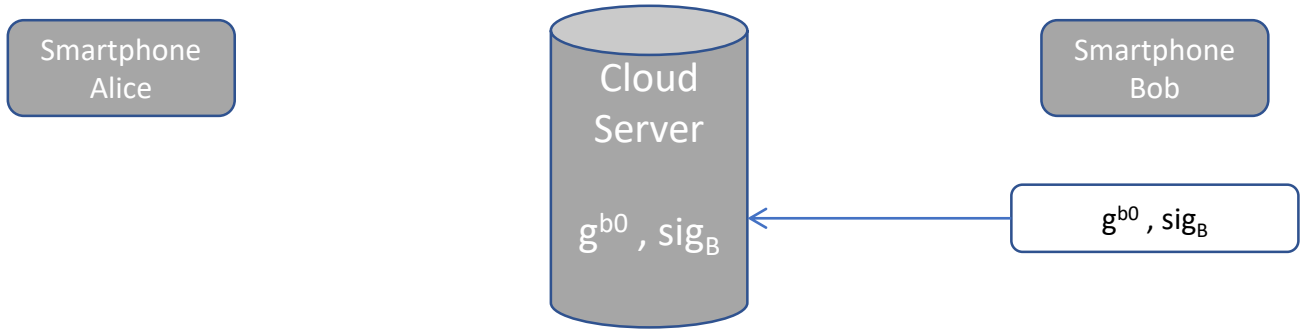
IEEE S&P 2022

Martin R. Albrecht[1], Lenka Mareková[2], Kenneth G. Paterson[3], and Igors Stepanovs[3]

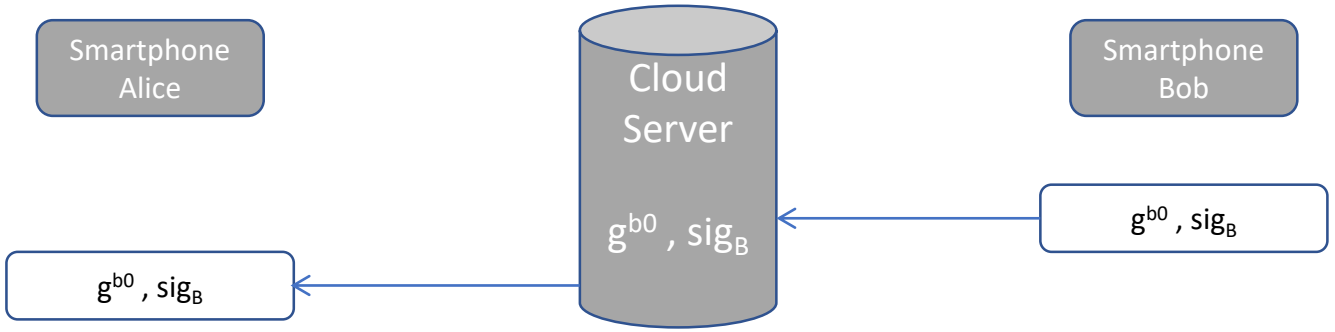
Agenda

- Push Messaging: E-Mail, SMS, Instant Messaging
- E2EE in Push Messaging: Similarities and Differences
  - E2EE vs. Transport Encryption
  - Stateless vs. Stateful E2EE
  - Static vs. Dynamic Groups
- Instant Messaging Security
  - Outliers: Threema, iMessage, Telegram
  - Double Ratchet: Signal, WhatsApp, Facebook Messenger, Wire, Matrix
  - IM Interoperability?
- EFAIL: Why is it still around?
  - Recap: Malleability Gadgets and Direct Exfiltration
  - How to reliably prevent EFAIL

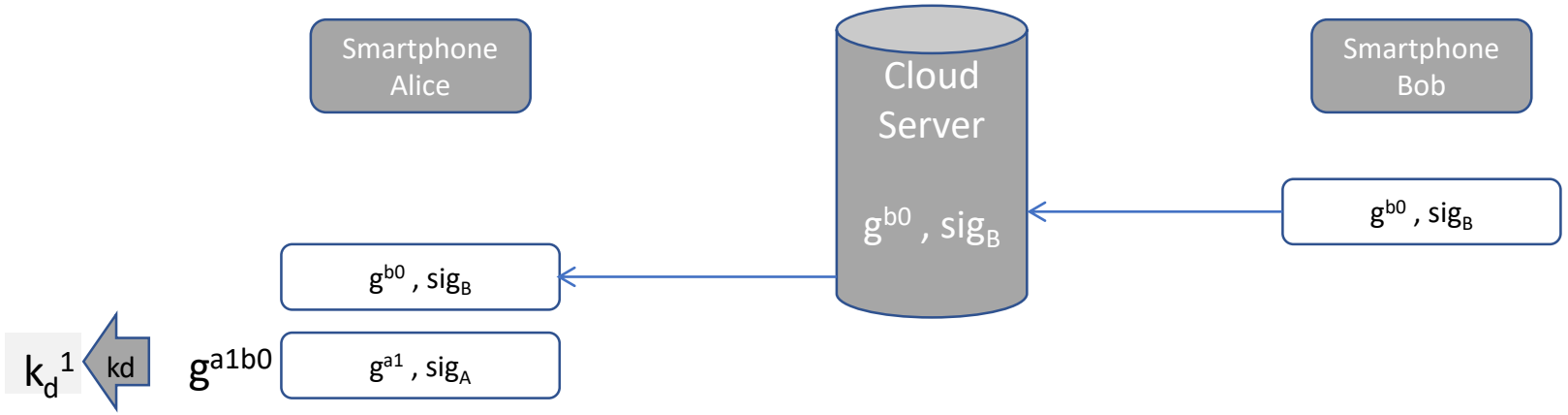
Double Ratchet – continuous DHKE



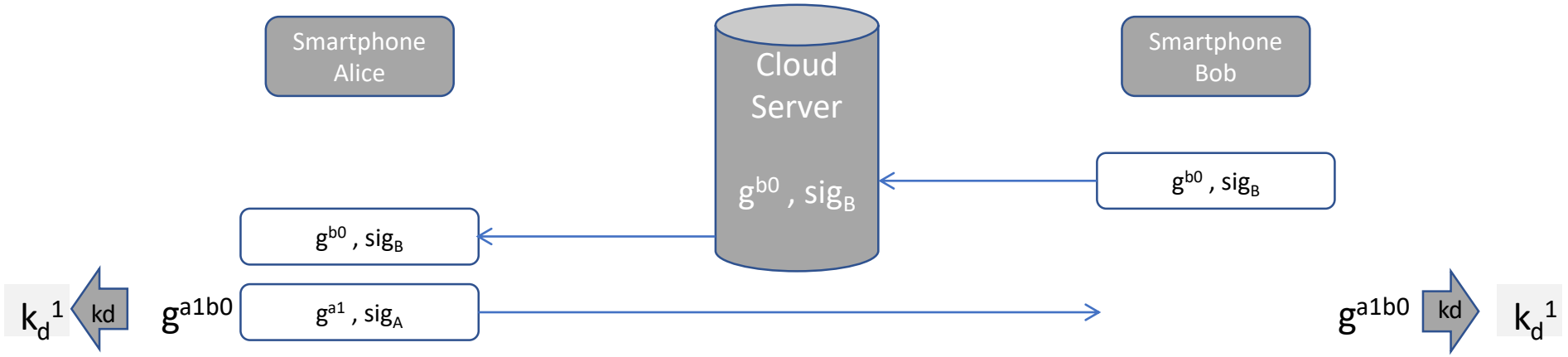
Double Ratchet – continuous DHKE



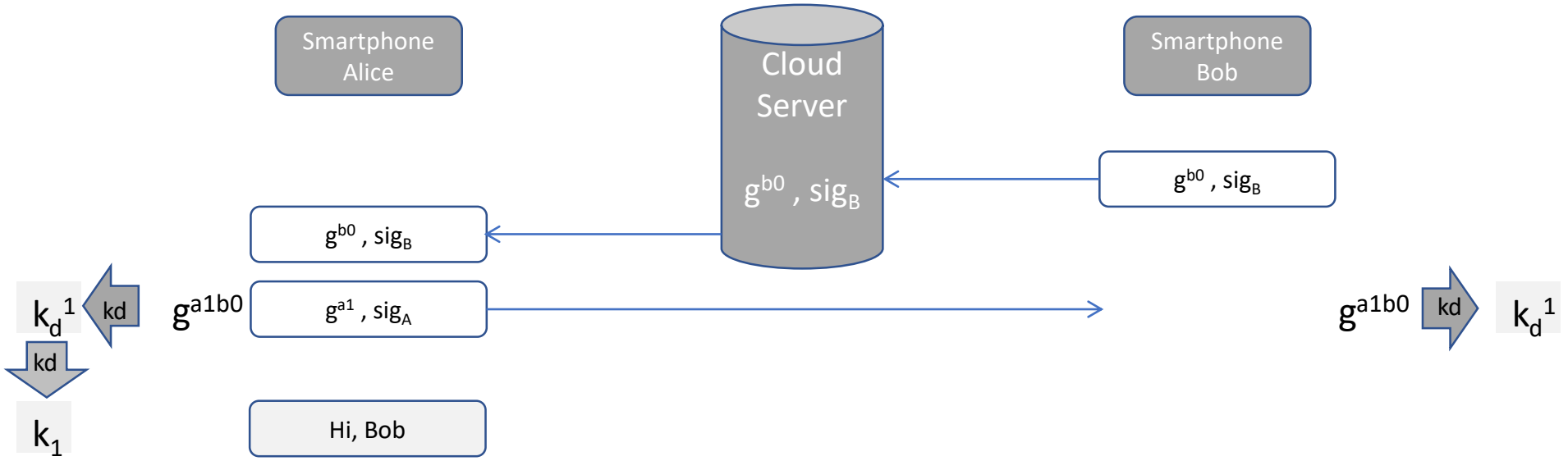
Double Ratchet – continuous DHKE



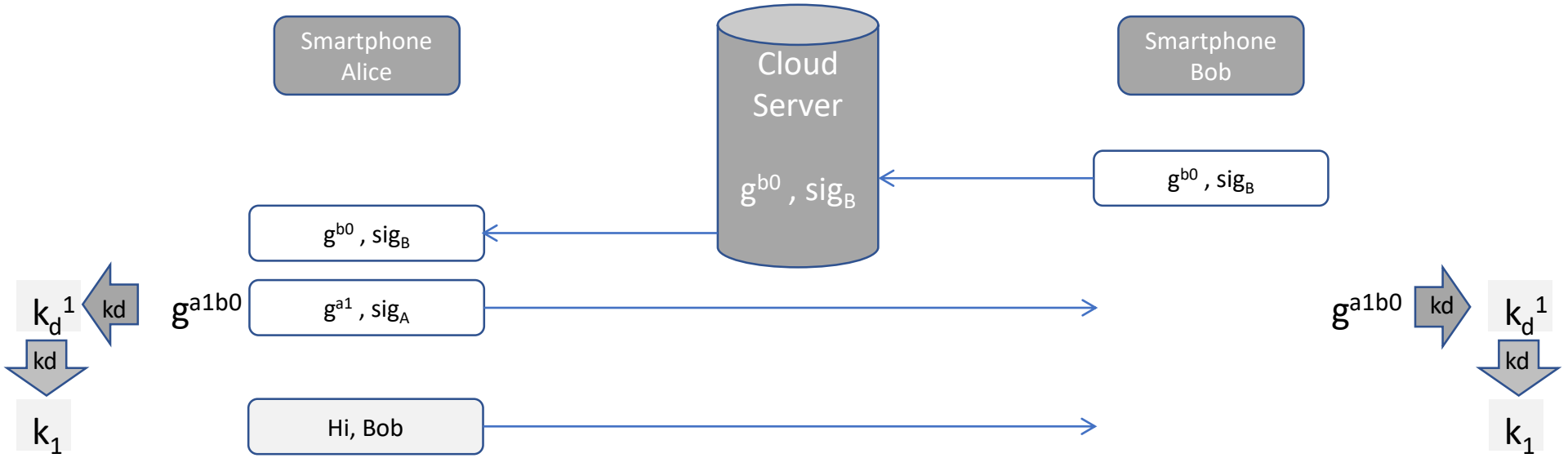
Double Ratchet – continuous DHKE



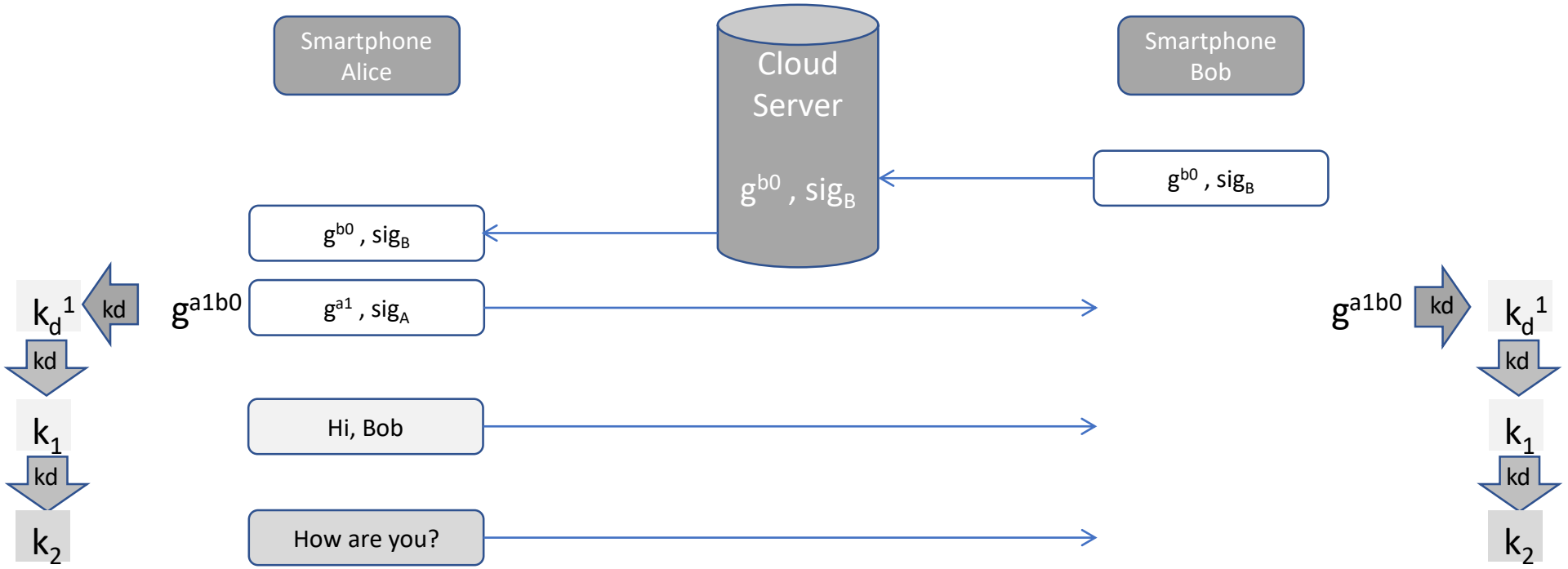
Double Ratchet – continuous DHKE



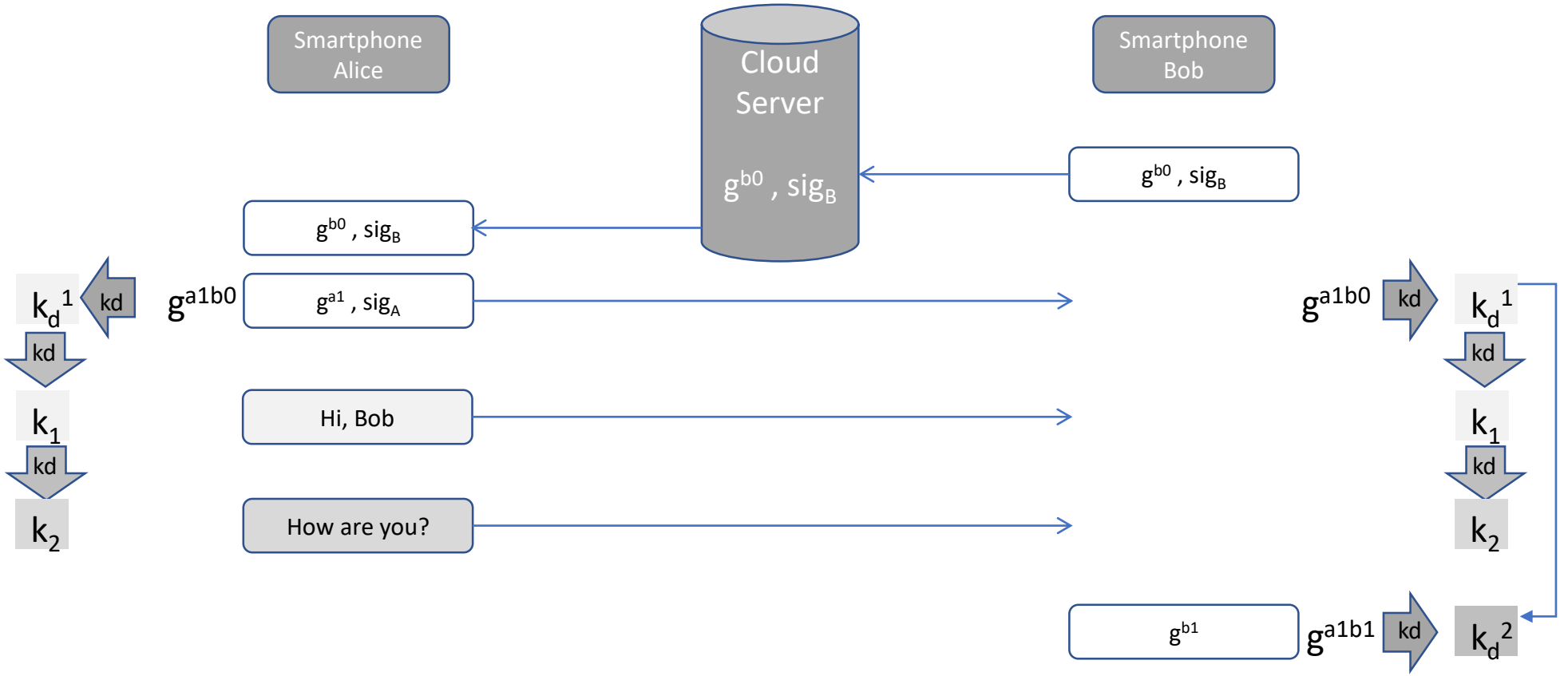
Double Ratchet – continuous DHKE



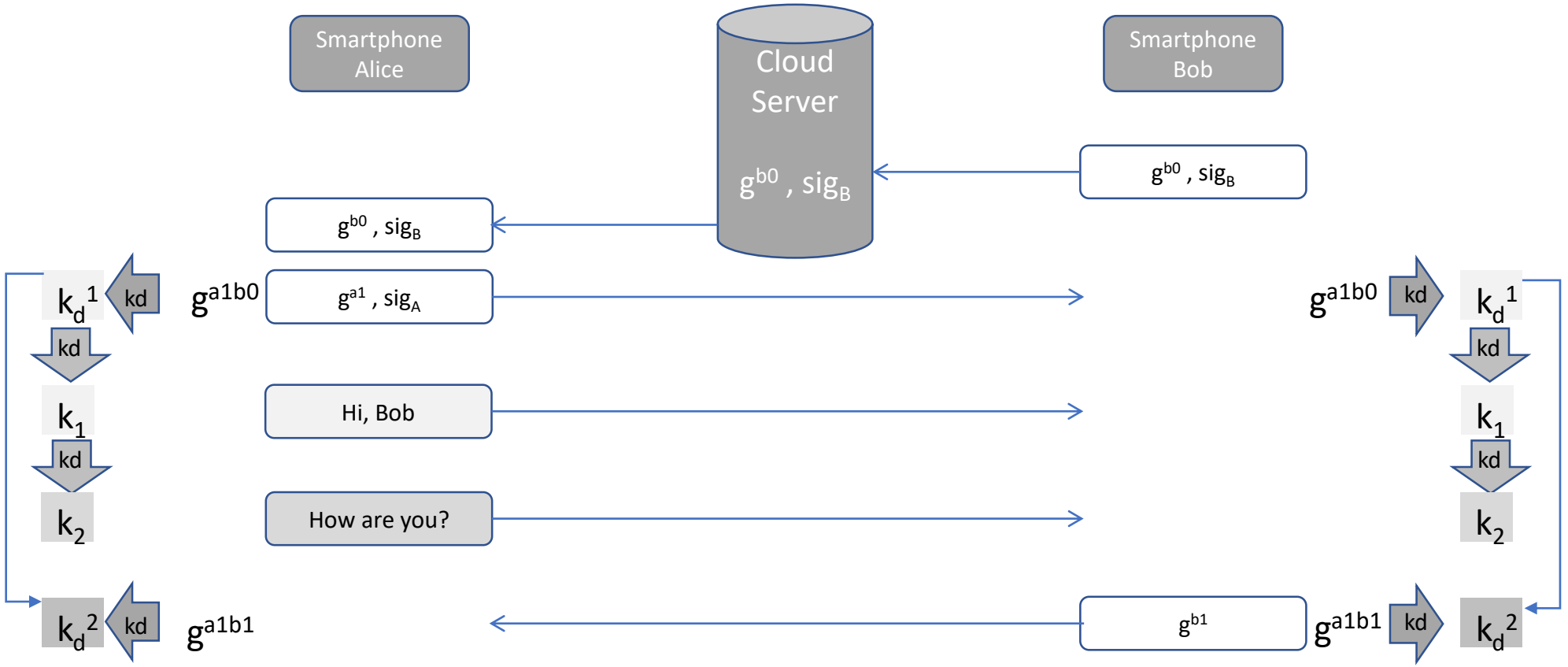
Double Ratchet – continuous DHKE



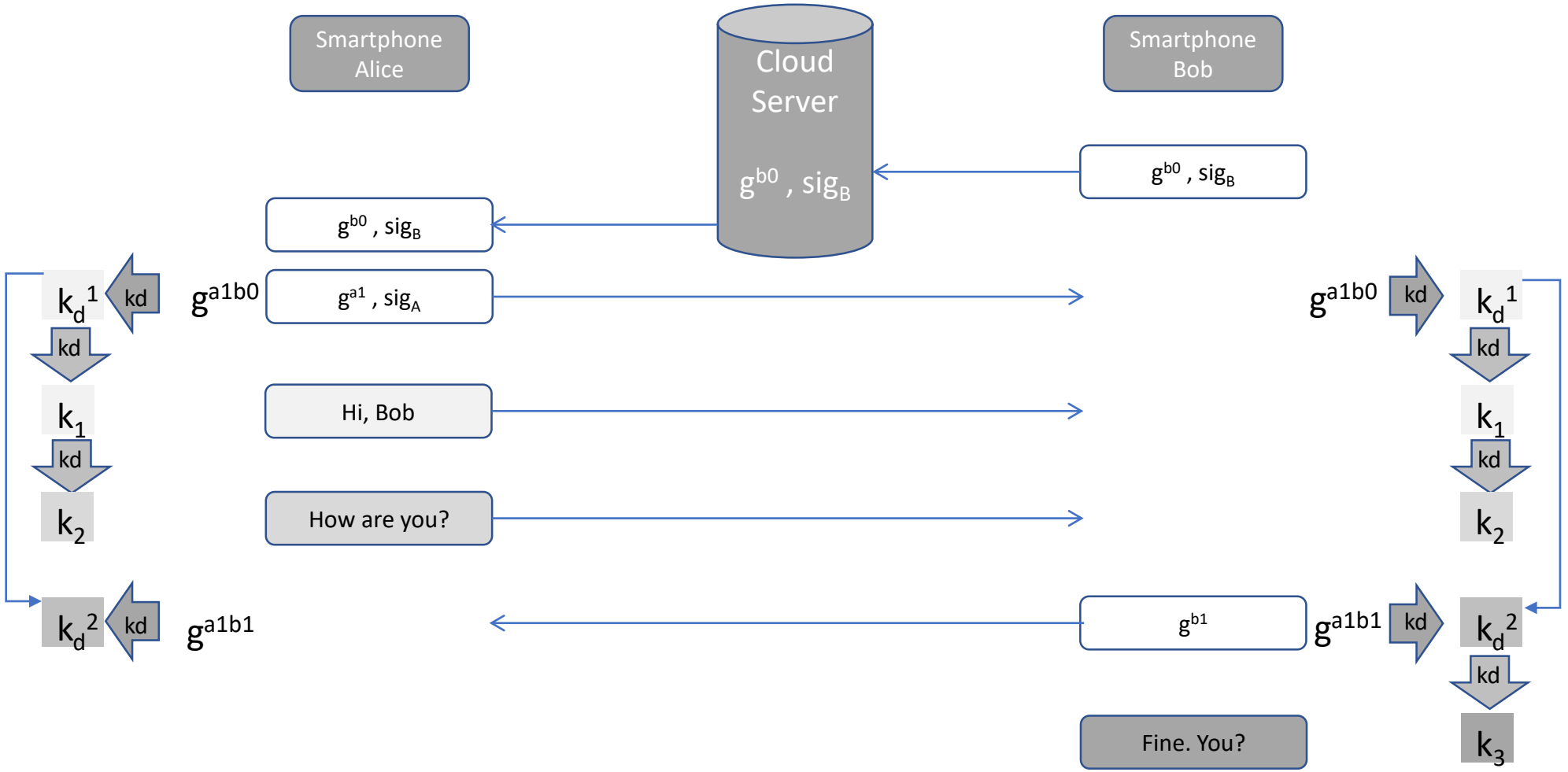
Double Ratchet – continuous DHKE



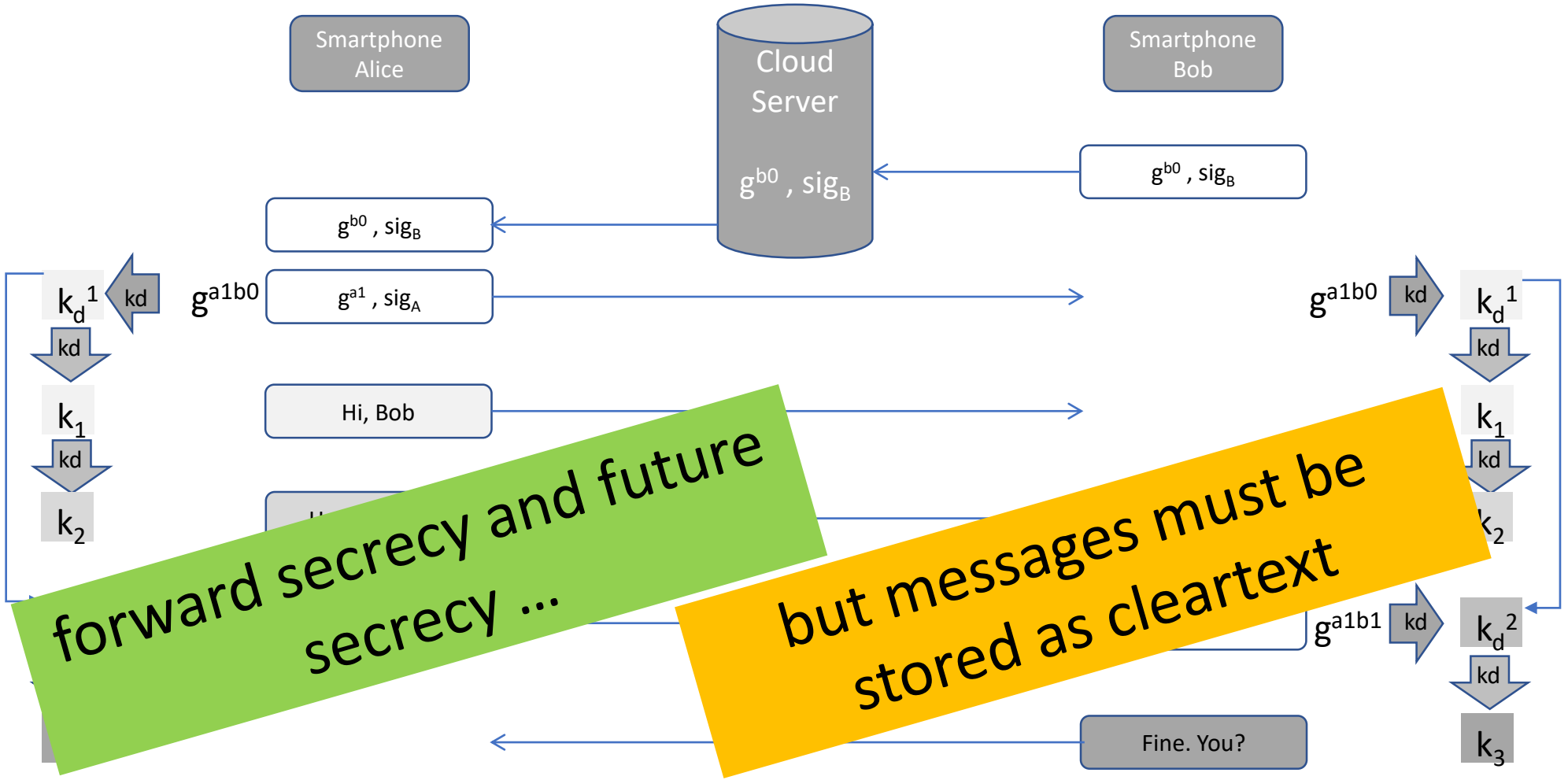
Double Ratchet – continuous DHKE



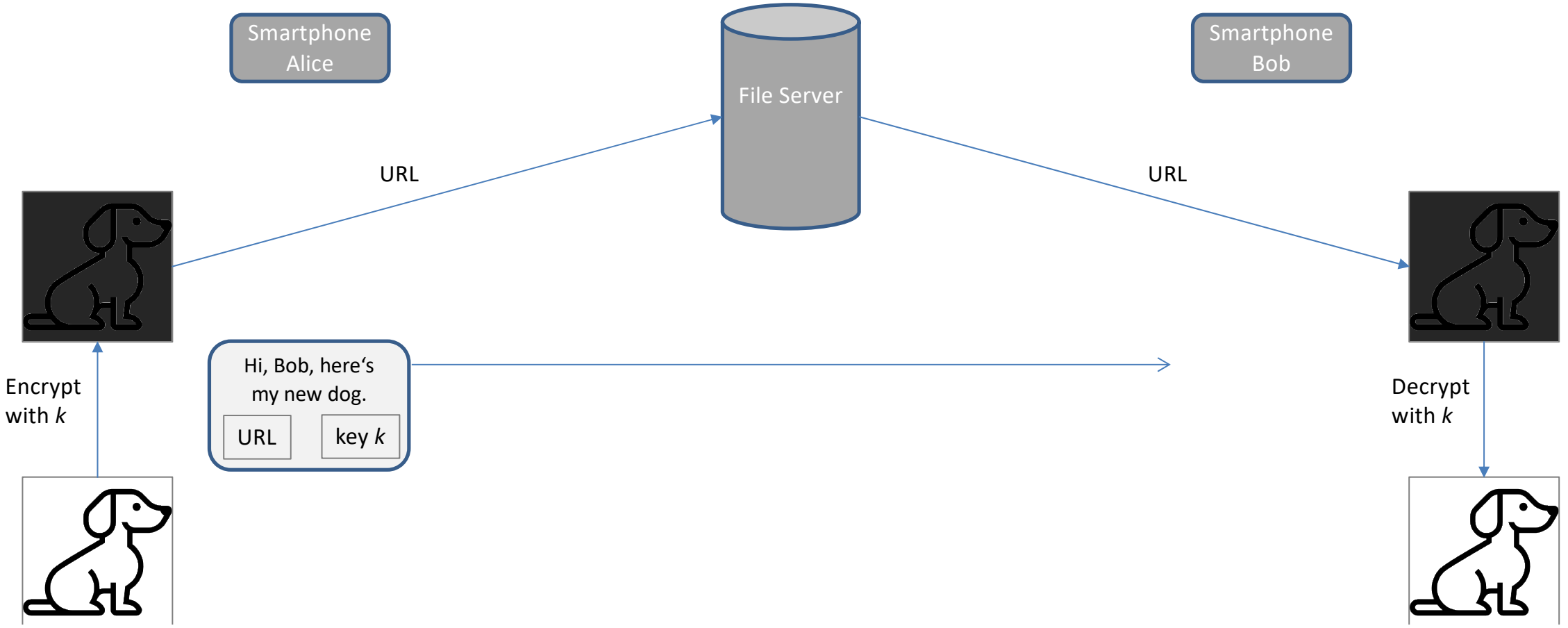
Double Ratchet – continuous DHKE



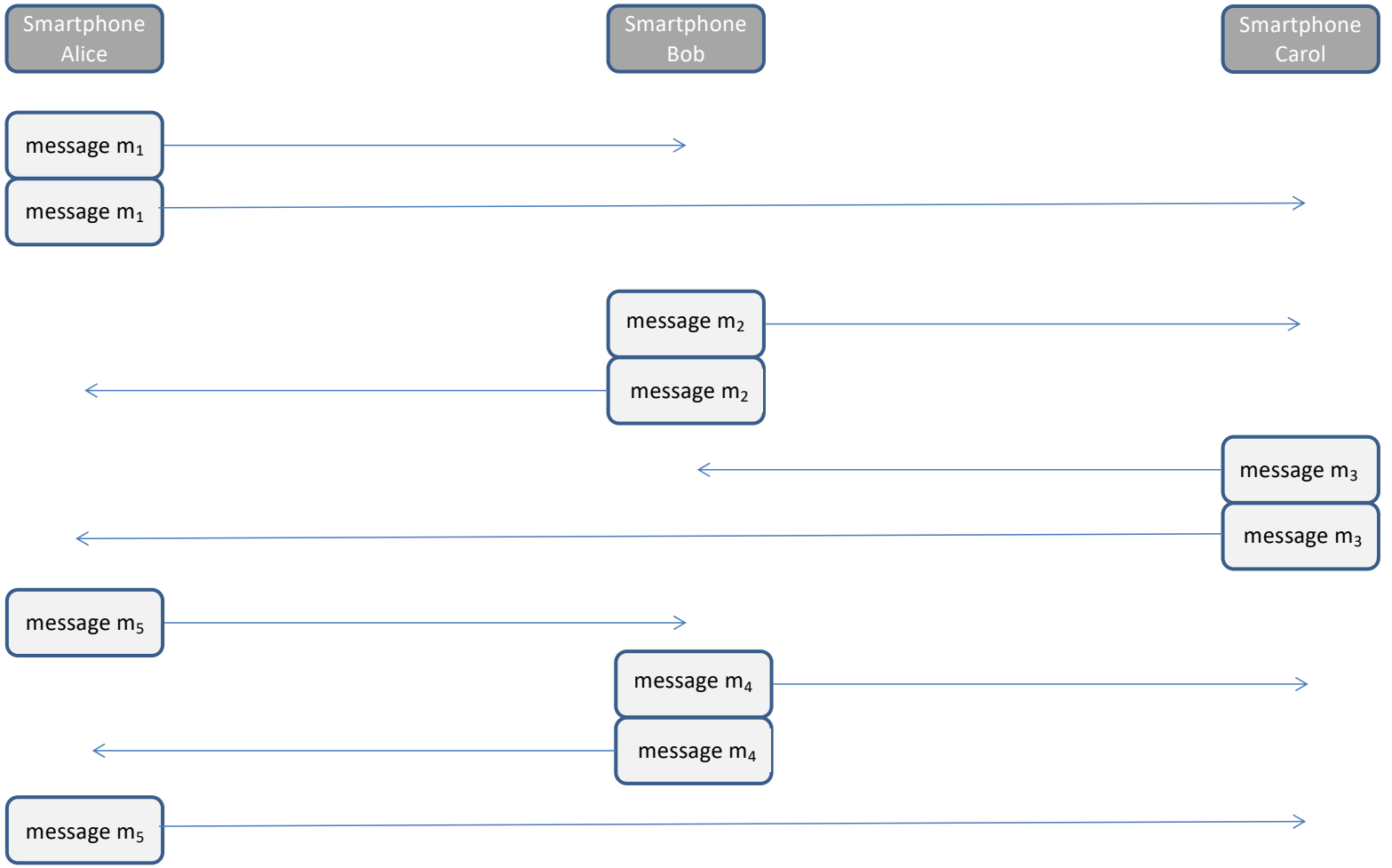
Double Ratchet – continuous DHKE



Key and URL via Text Message

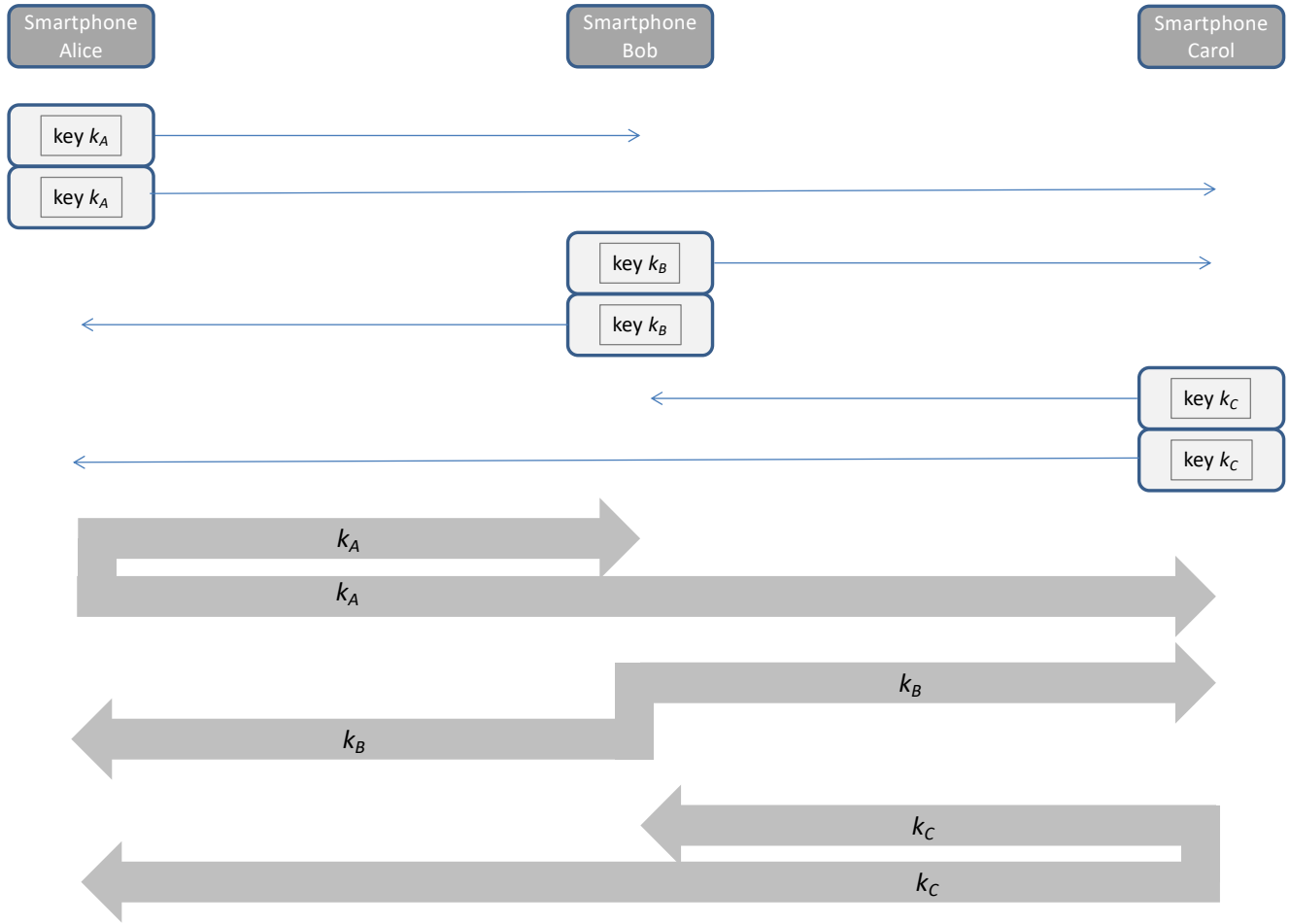


Group Communication: Pairwise Ratcheting

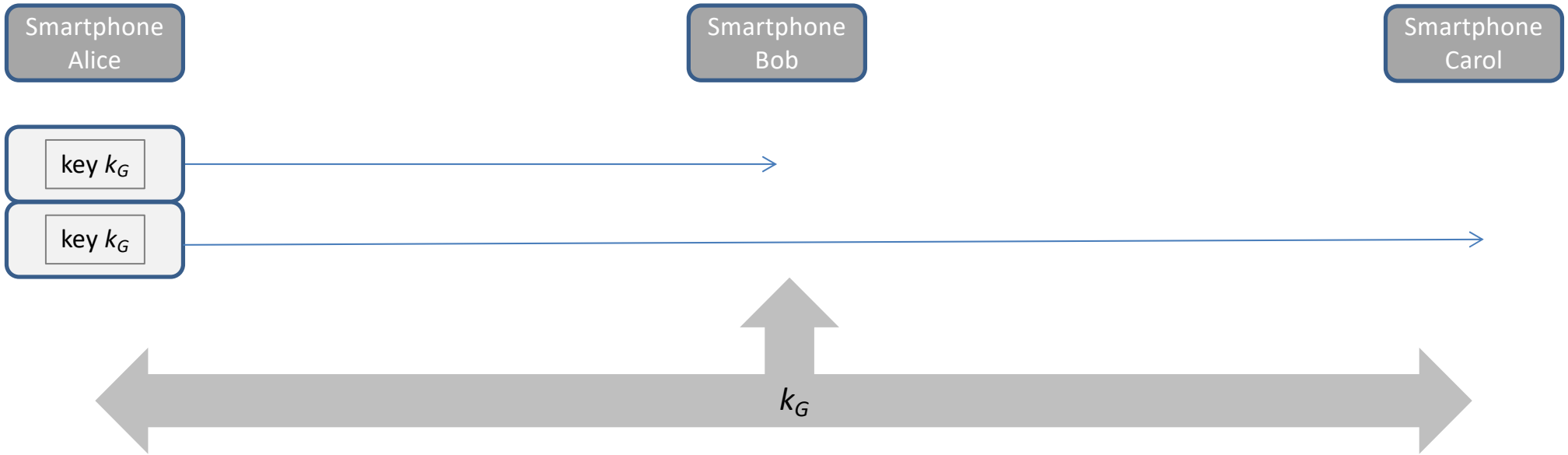


similar to

Group Communication: Sender Key

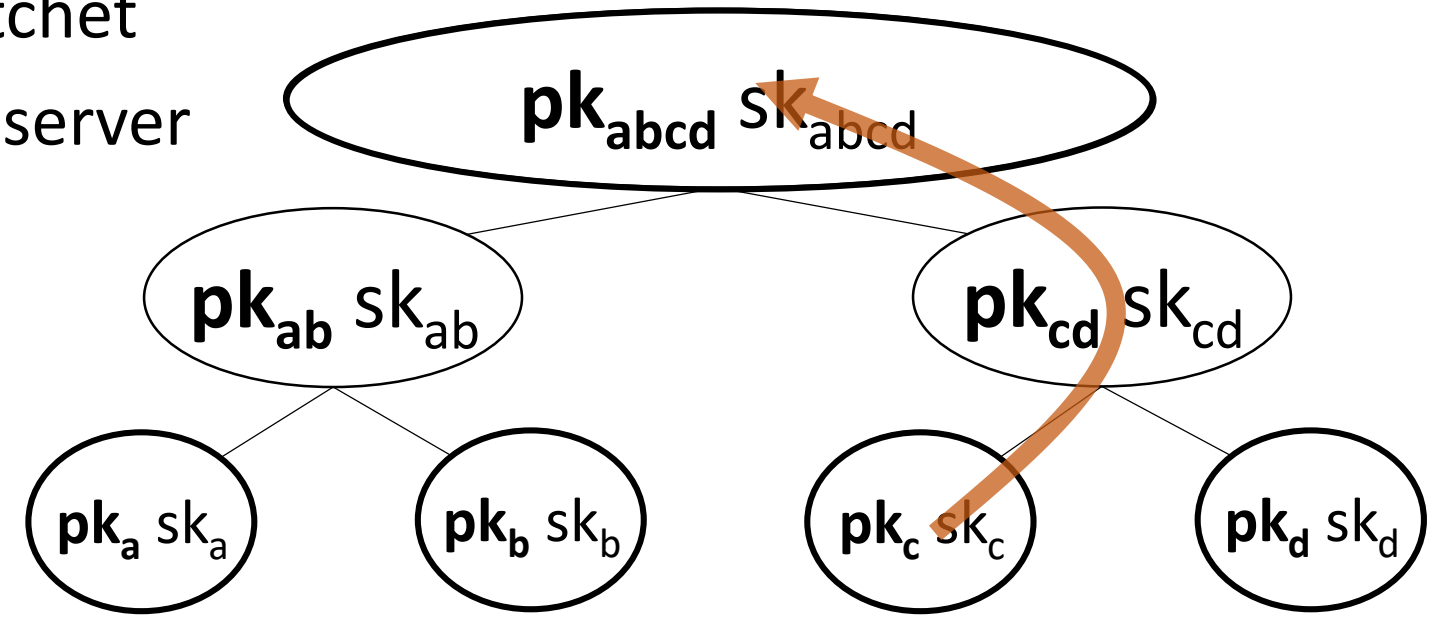


Group Communication: Static Group Key



IETF MLS

- IETF Standard
- Stateful like Double Ratchet
- Needs synchronization server



Instant Messaging Interoperability

Requirements DMA

REGULATION (EU) 2022/1925 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32022R1925. 2022.

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

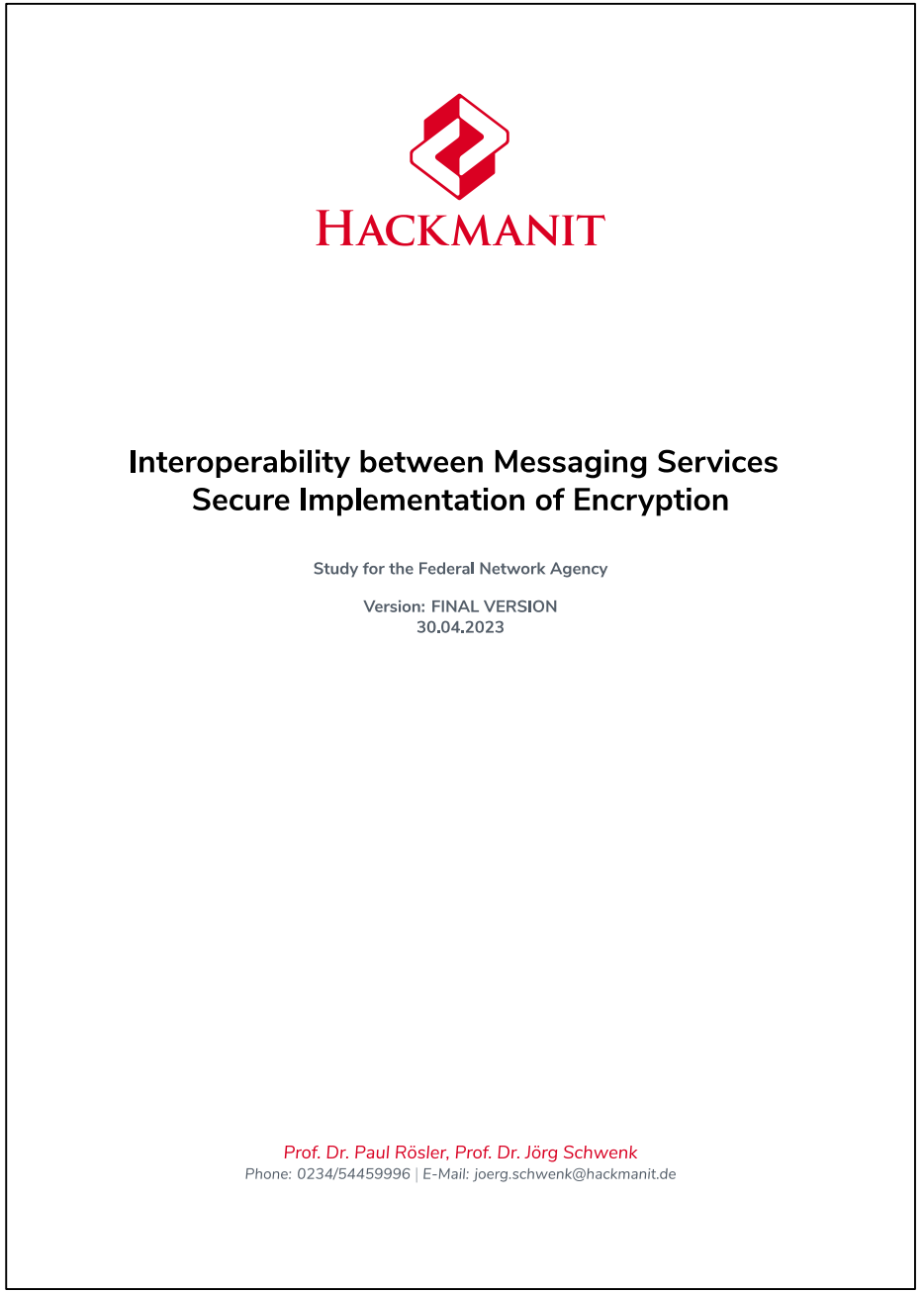
Metadata Protection:

"*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

Technical Report about interoperability of IM



HACKMANIT

Interoperability between Messaging Services
Secure Implementation of Encryption

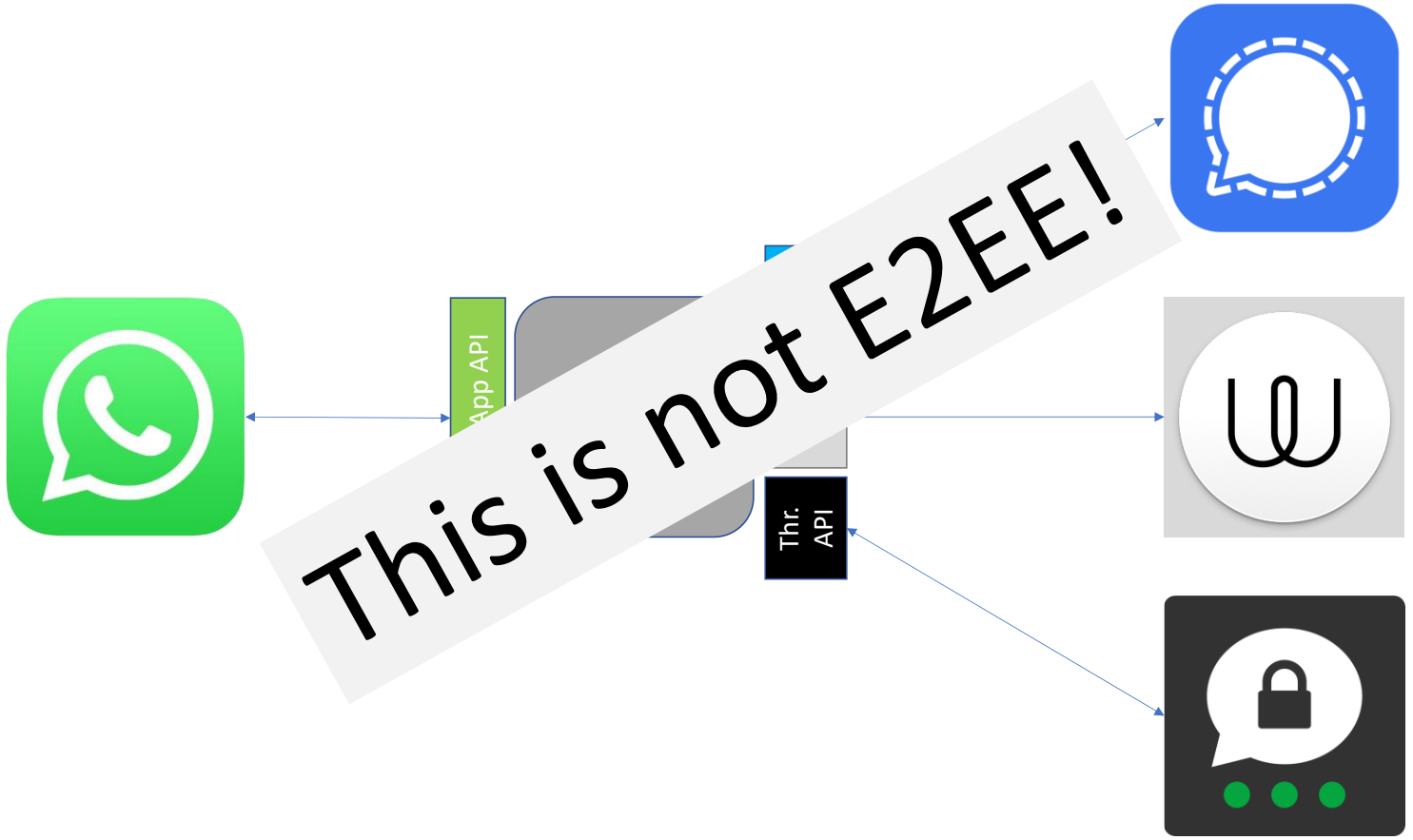
Study for the Federal Network Agency

Version: FINAL VERSION
30.04.2023

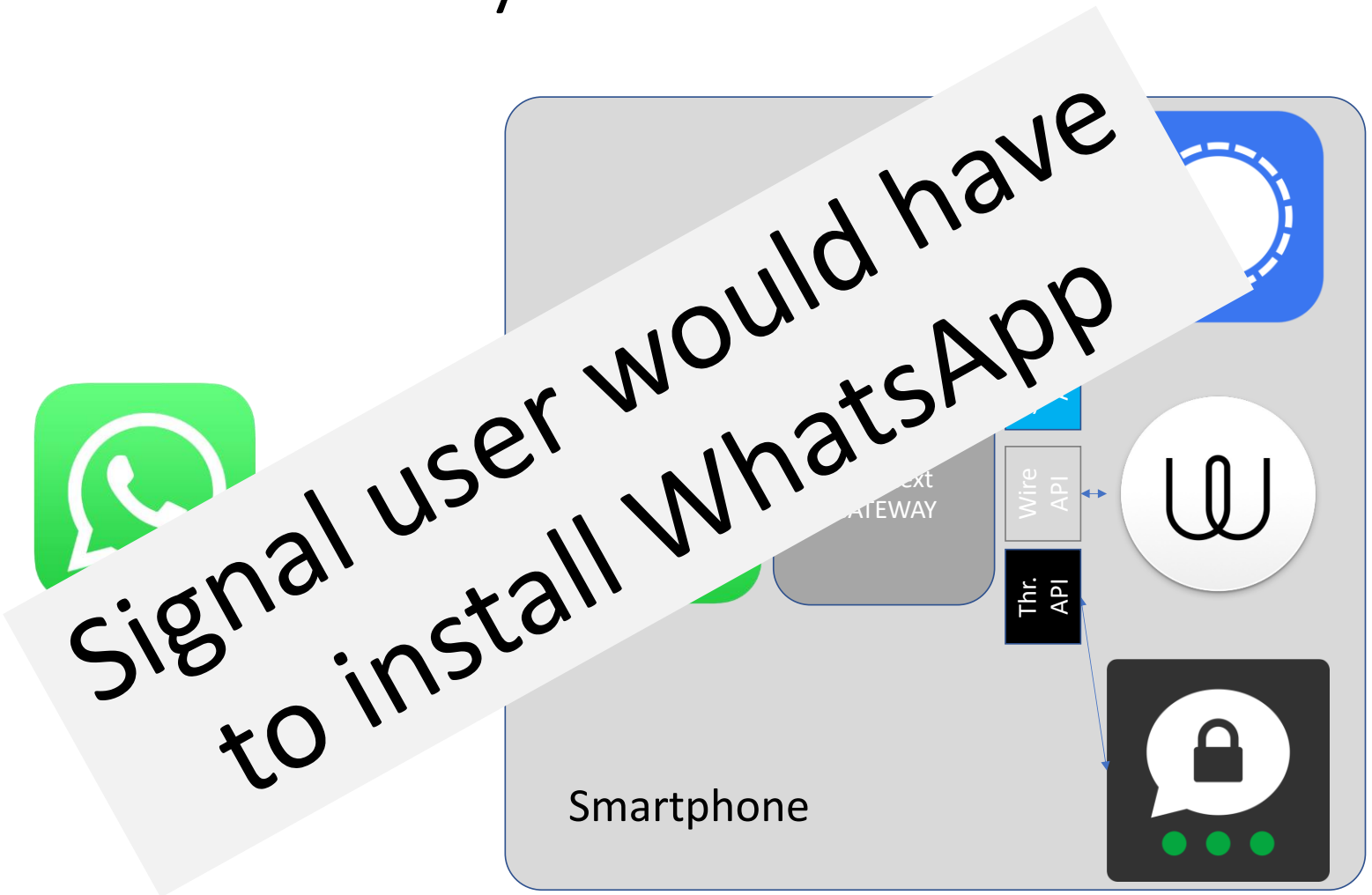
Prof. Dr. Paul Rösler, Prof. Dr. Jörg Schwenk
Phone: 0234/54459996 | E-Mail: joerg.schwenk@hackmanit.de

www.bundesnetzagentur.de/online-kommunikation#IOPStudy

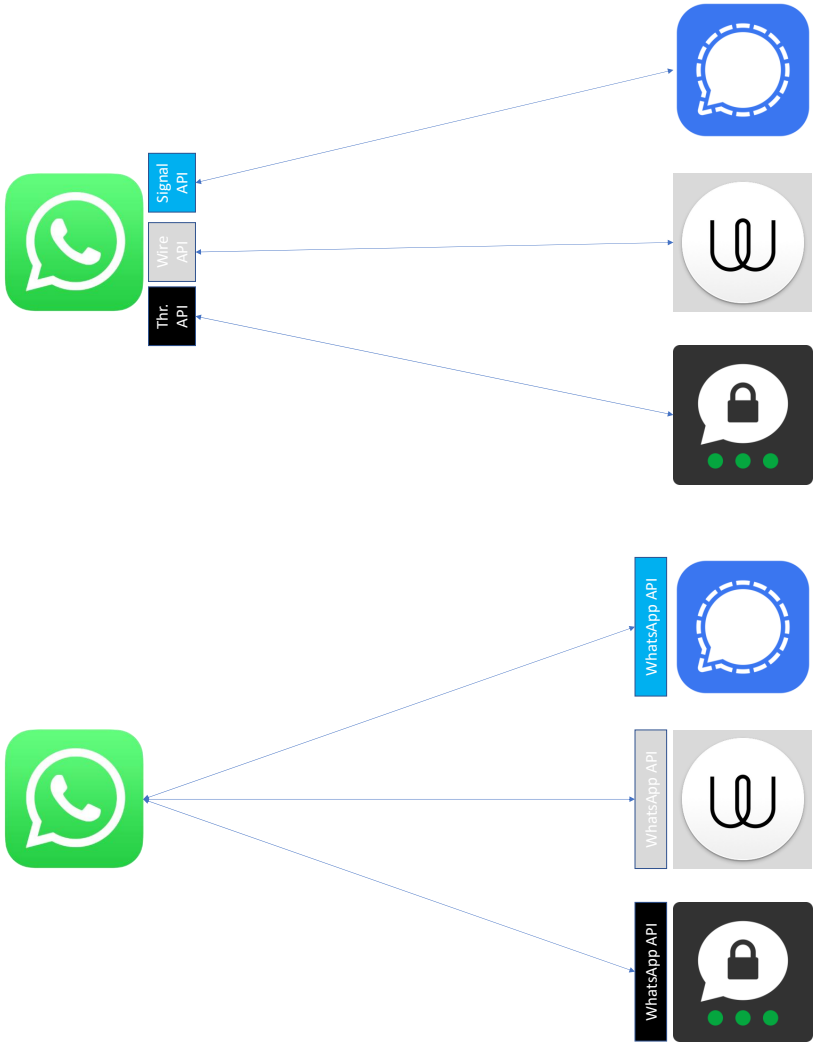
Server-Side Gateway

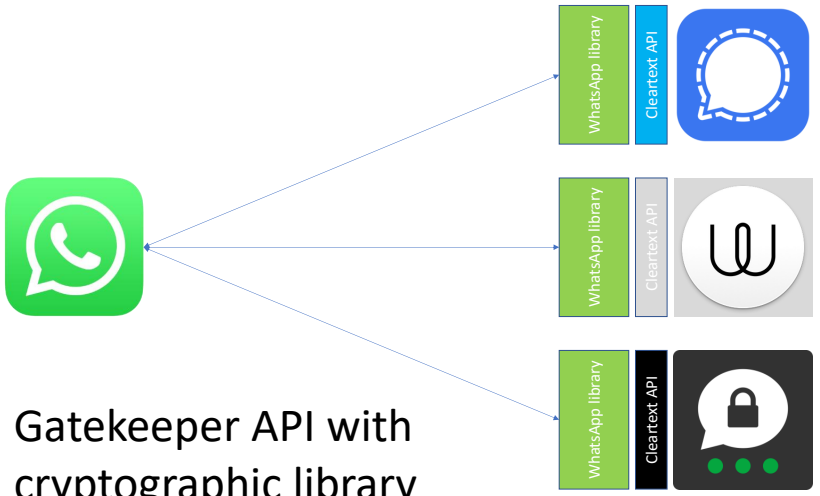


Cient-Side Gateway

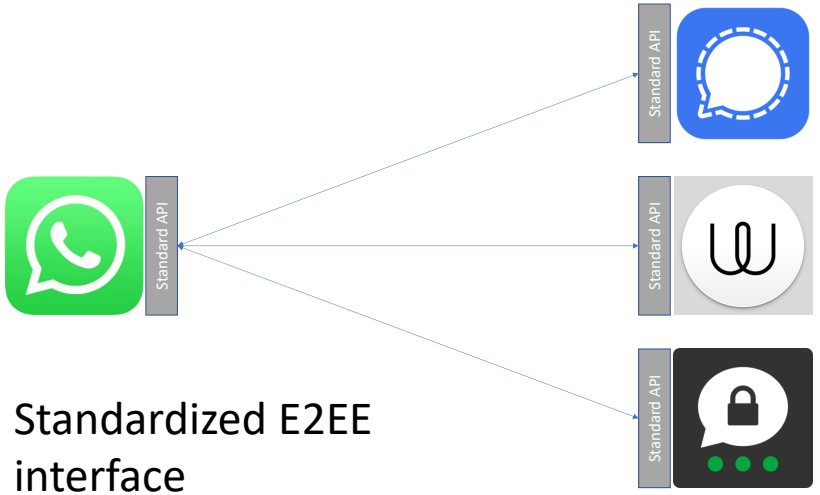


Where to ‘translate’ E2EE?



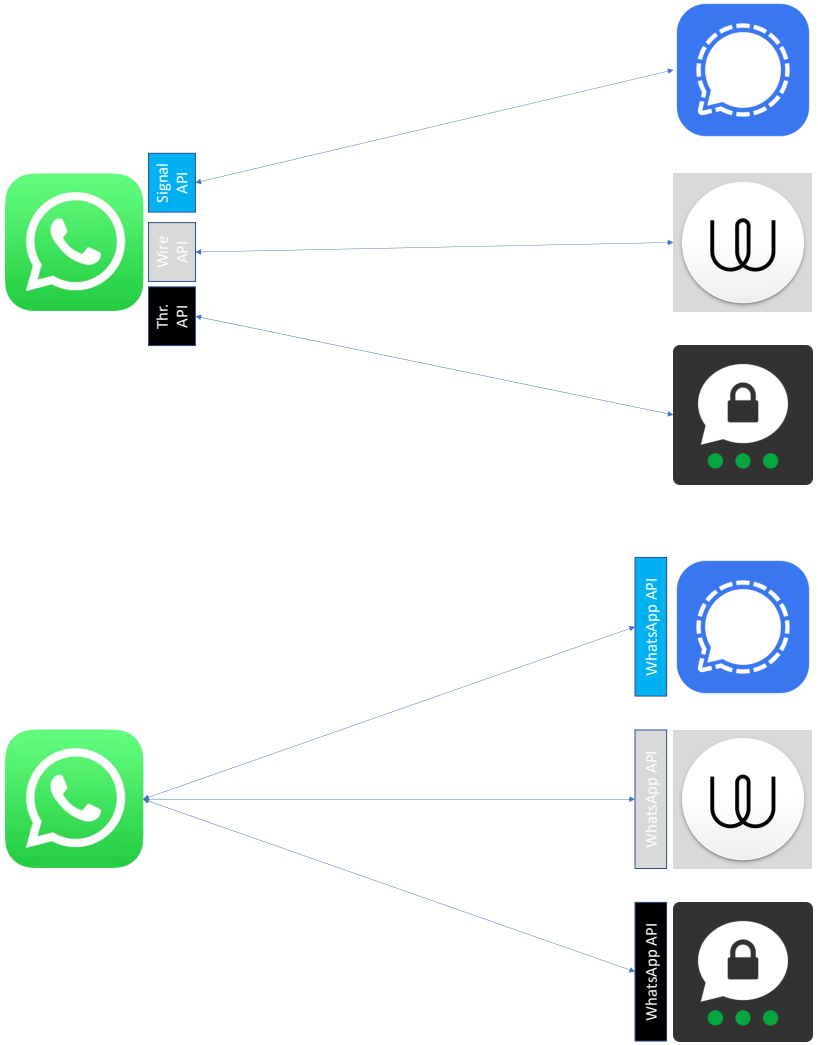


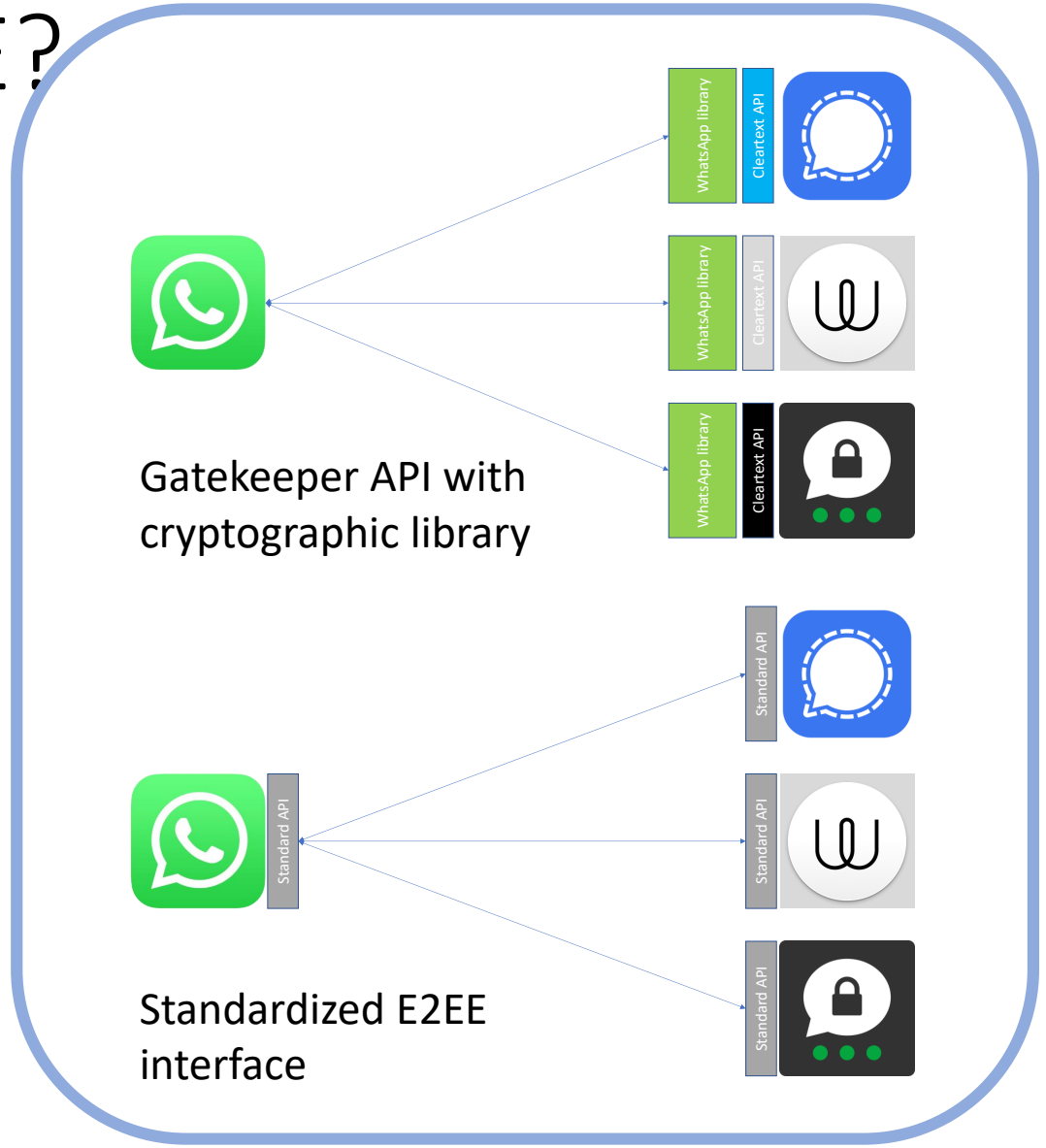
Gatekeeper API with cryptographic library



Standardized E2EE interface

Where to ‘translate’ E2EE?

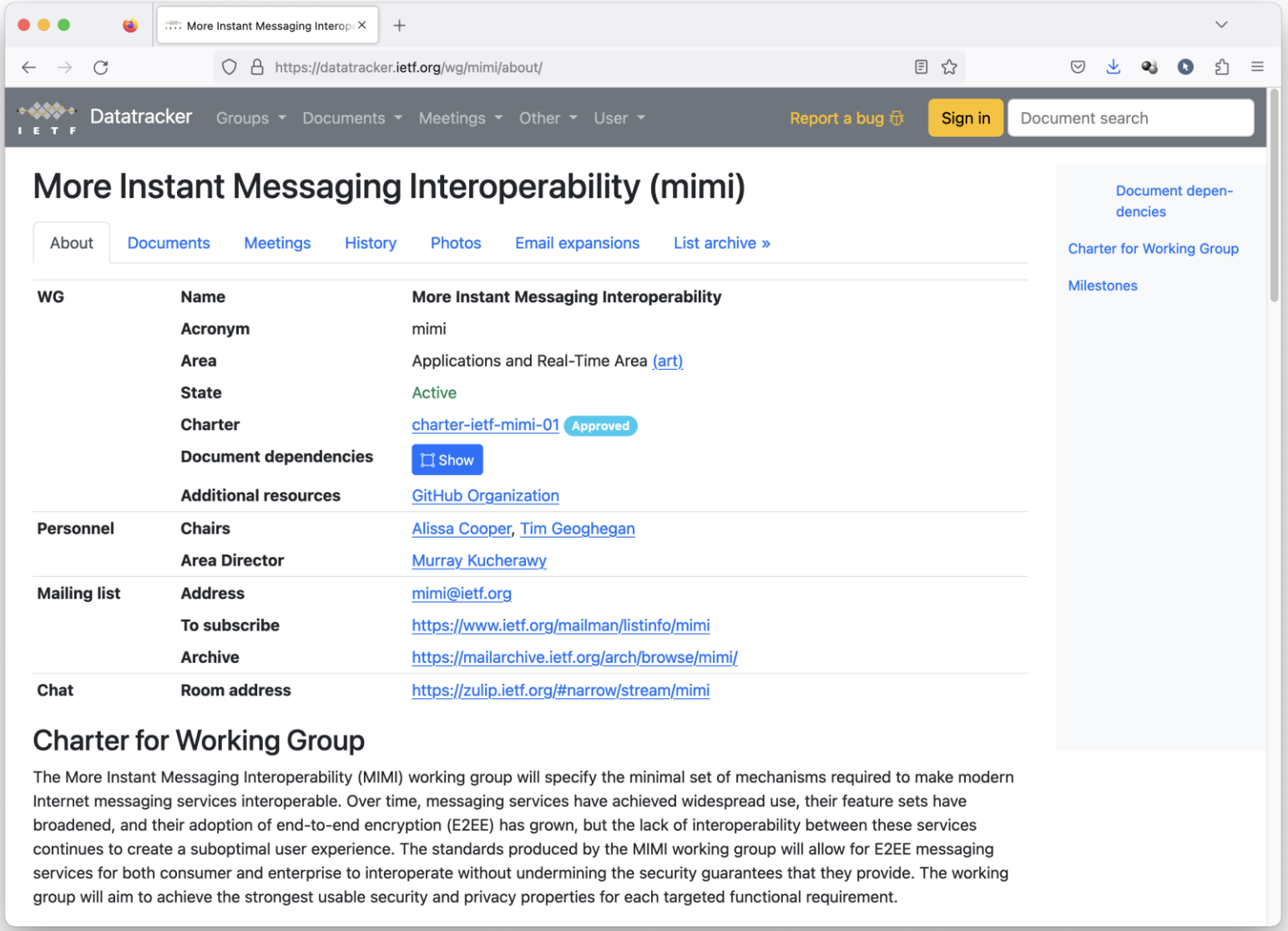




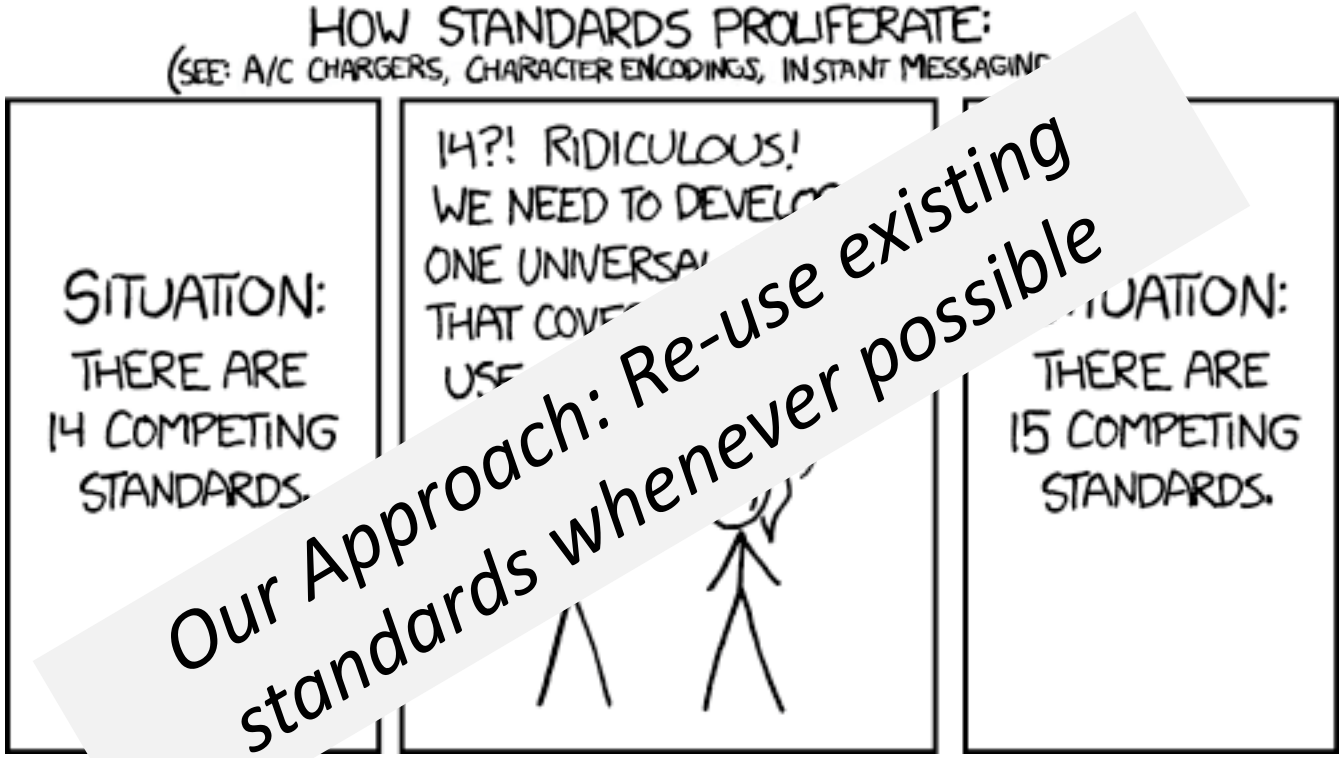
IETF MIMI

„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.“



Standardization Goal



Our Approach: Re-use existing standards whenever possible

Identities, Key Distribution, Trust

- Standardize naming scheme similar to e-mail:

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 (**DR**)	**DR**	WebRTC
WhatsApp	**DR**	Sender Key (**SK**)	SRTP
Facebook Messenger	**DR** with Message Franking	**SK** with Message Franking	Undocumented
Wire	Proteus (≈**DR**; diff. AE)	Proteus (≈**DR**)	SRTP
Matrix	Olm (≈**DR**; diff. KDF)	Megolm (≈**SK**)	WebRTC
Threema	NIKE	NIKE	
iMessage	Public-key encryption	Public-key encryption	SRTP
Telegram	MTProto	Unencrypted	MTProto

- 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

Summary IM Interoperability

- 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

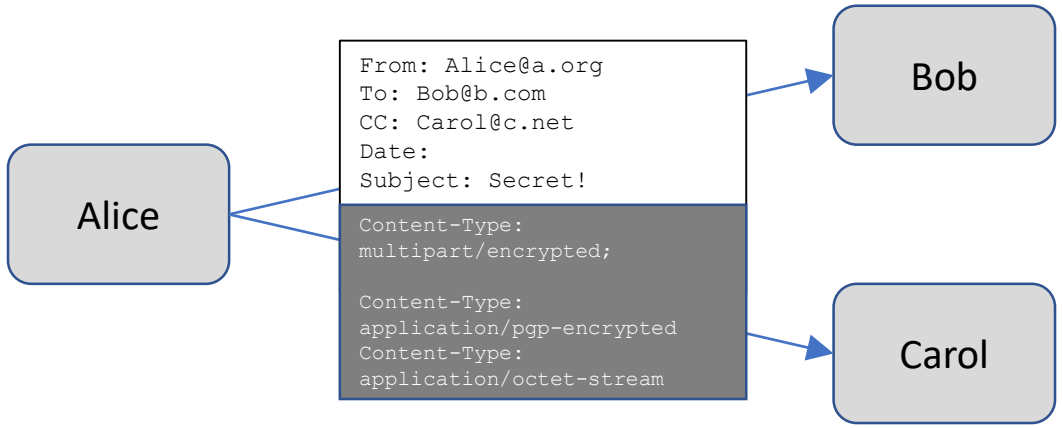
Can we get rid of  ?

OpenPGP and S/MIME Usability

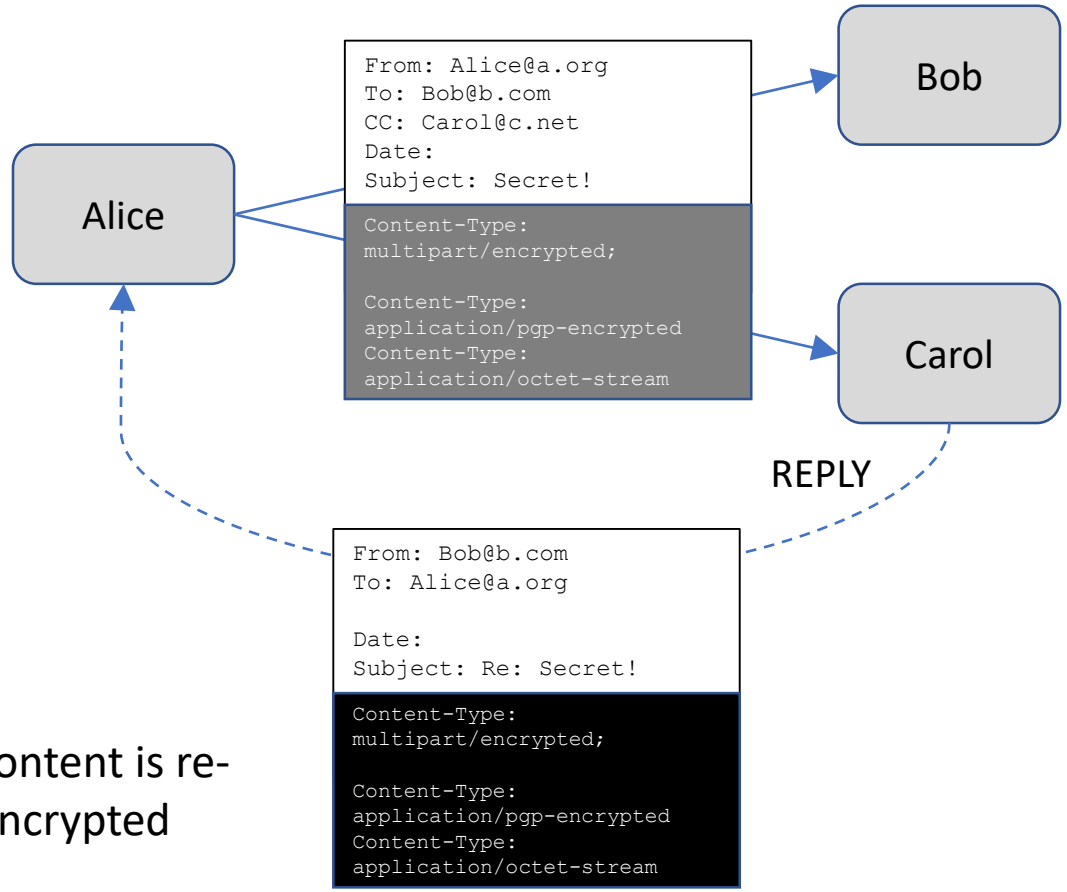
- Learn from IM!
- Allow expert users to configure manually
- Help non-expert users by making decisions for them …
  - … but keep at least TOFU (OpenPGP Autocrypt)

Are EFAIL and REPLY Attacks Still a Problem?

Use Cases for E-Mail E2E Encryption

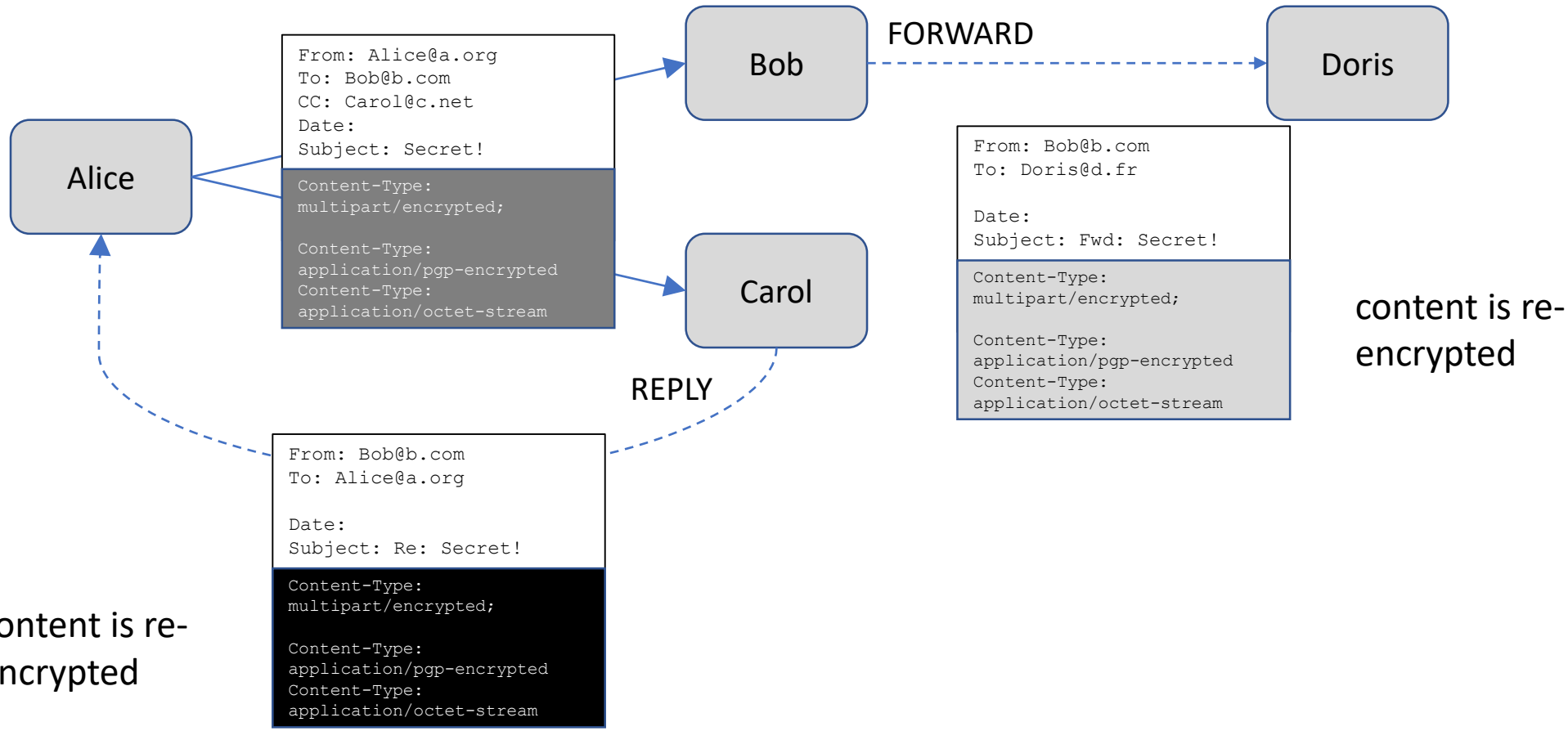


Use Cases for E-Mail E2E Encryption

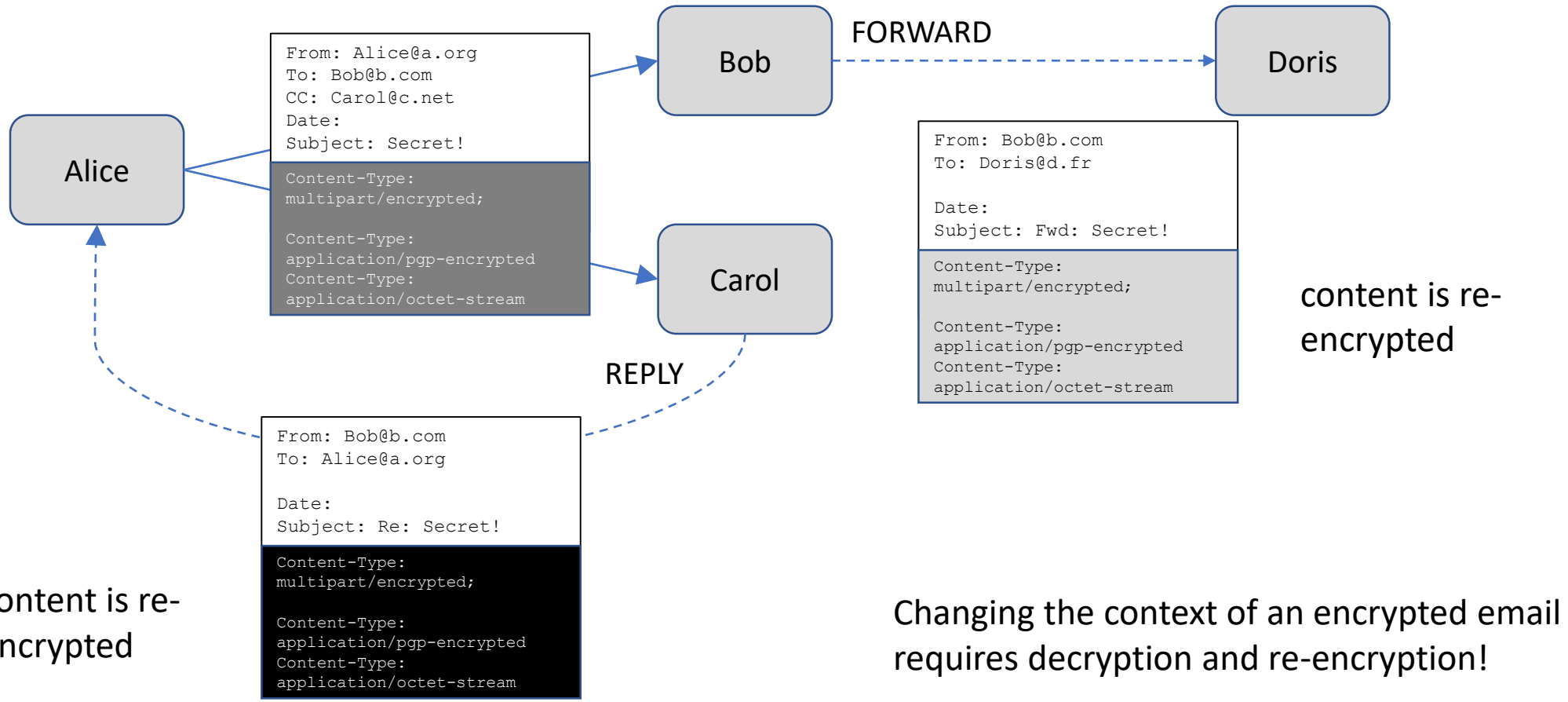


content is re-encrypted

Use Cases for E-Mail E2E Encryption

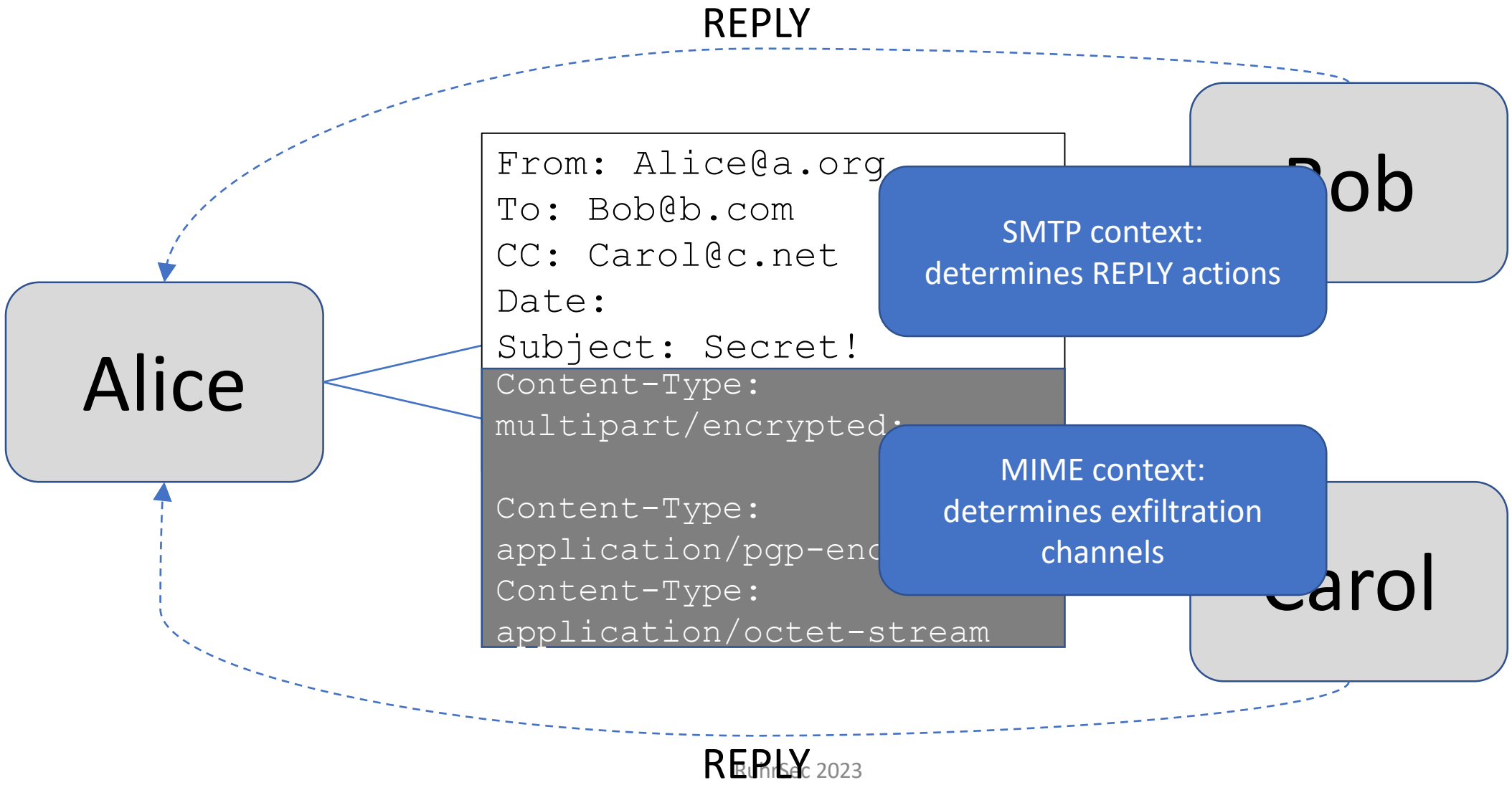


Use Cases for E-Mail E2E Encryption



Changing the context of an encrypted email requires decryption and re-encryption!

Decryption Context

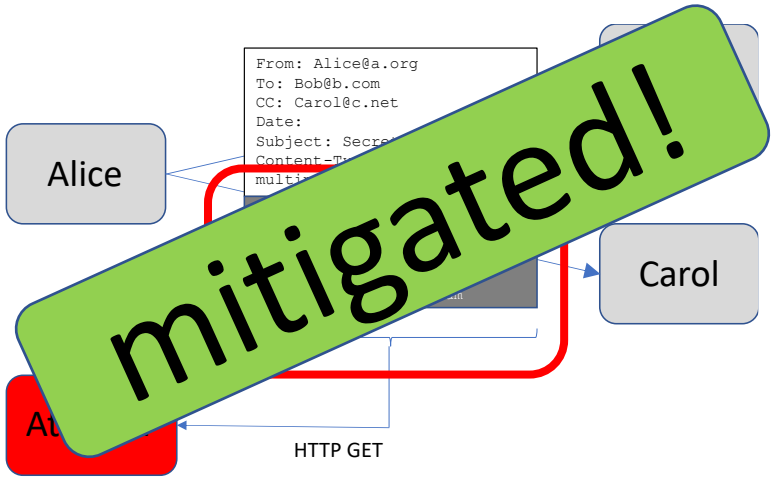


Decryption Context vs. Ciphertext

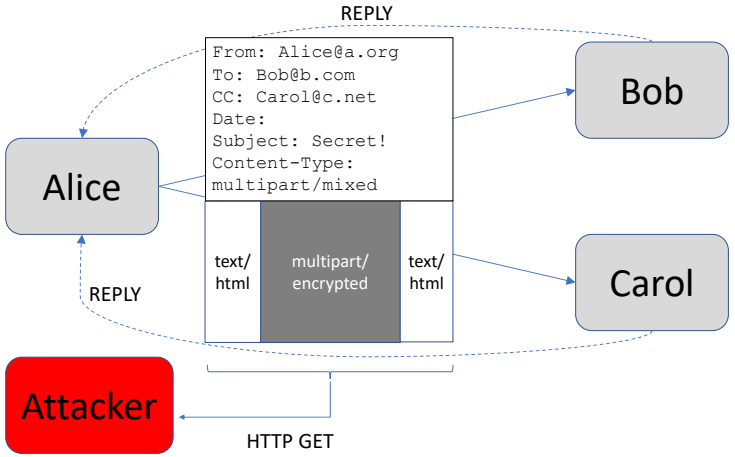
	REPLY	FORWARD			
Decryption Context	changes	changes			
Ciphertext	changes	changes			

Attack Classes

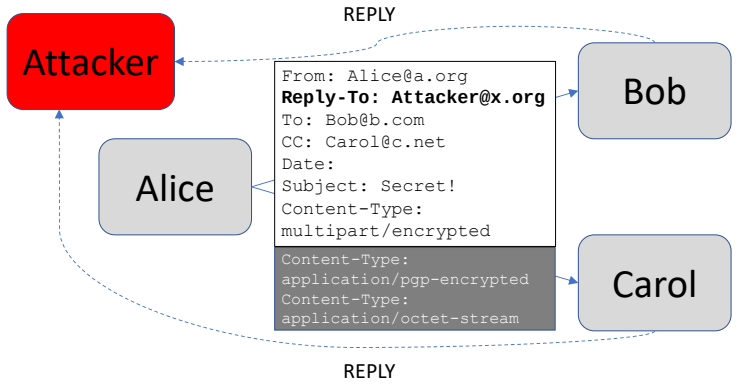
EFAIL Malleability Gadgets



EFAIL Direct Exfiltration



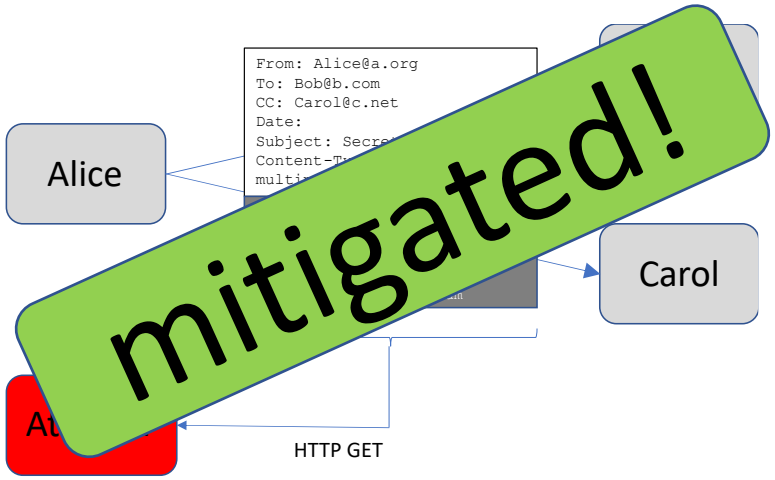
REPLY attacks



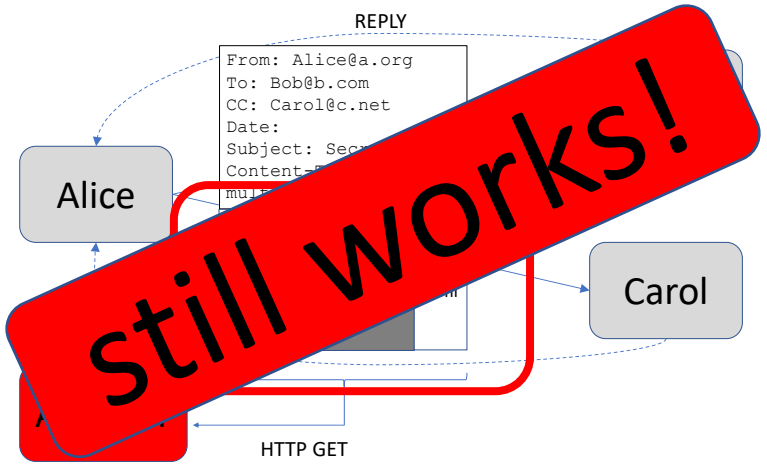
- ciphertext is modified
- mitigated through AEAD
- SENDER-enforced

Attack Classes

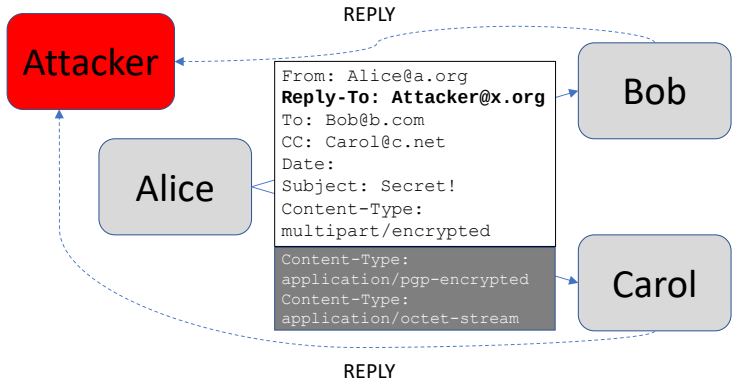
EFAIL Malleability Gadgets



EFAIL Direct Exfiltration



REPLY attacks

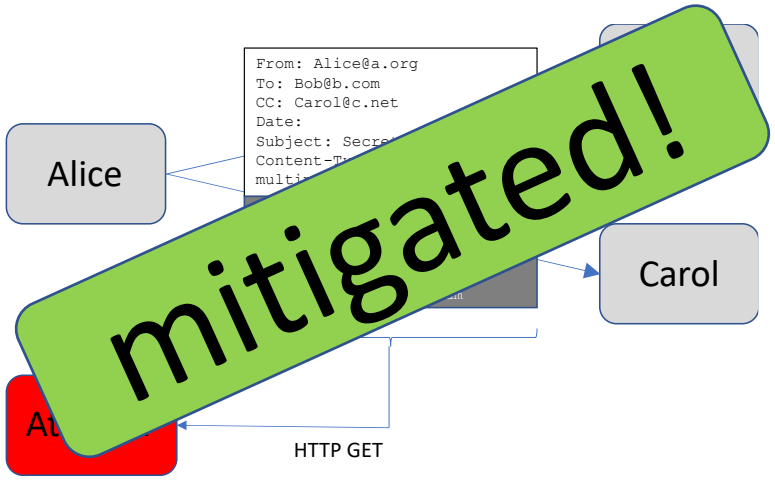


- <u>ciphertext is modified</u>
- mitigated through AEAD
- SENDER-enforced

- <u>MIME context is modified</u>
- different partial mitigations
- RECIPIENT-enforced
- novel attack variants

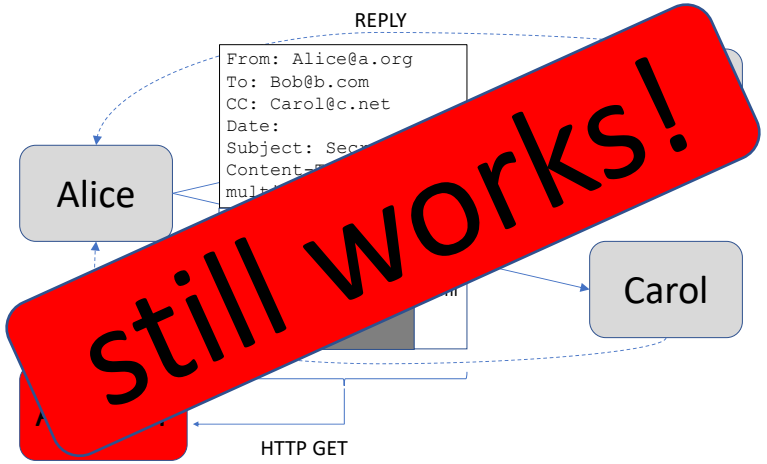
Attack Classes

EFAIL Malleability Gadgets



- ciphertext is modified
- mitigated through AEAD
- SENDER-enforced

EFAIL Direct Exfiltration



- MIME context is modified
- different partial mitigations
- RECIPIENT-enforced
- novel attack variants

REPLY attacks

- SMTP context is modified
- no mitigation

Decryption Context vs. Ciphertext

	REPLY	FORWARD	EFAIL-MG	EFAIL-DE	REPLY-Att.
Decryption Context	changes	changes	same	modified	modified
Ciphertext	changes	changes	modified	same	same

Decryption Context vs. Ciphertext

	REPLY	FORWARD	EFAIL-MG	EFAIL-DE	REPLY-Att.
Decryption Context	changes	changes	same	modified	modified
Ciphertext	changes	changes	modified	same	same
Mitigation	-	-	AE(AD)		

Decryption Context vs. Ciphertext

	REPLY	FORWARD	EFAIL-MG	EFAIL-DE	REPLY-Att.
Decryption Context	changes	changes	same	modified	modified
Ciphertext	changes	changes	modified	same	same
Mitigation	-	-	AEAD	AEAD with DC as AD	AEAD with DC as AD

Decryption Context: Example

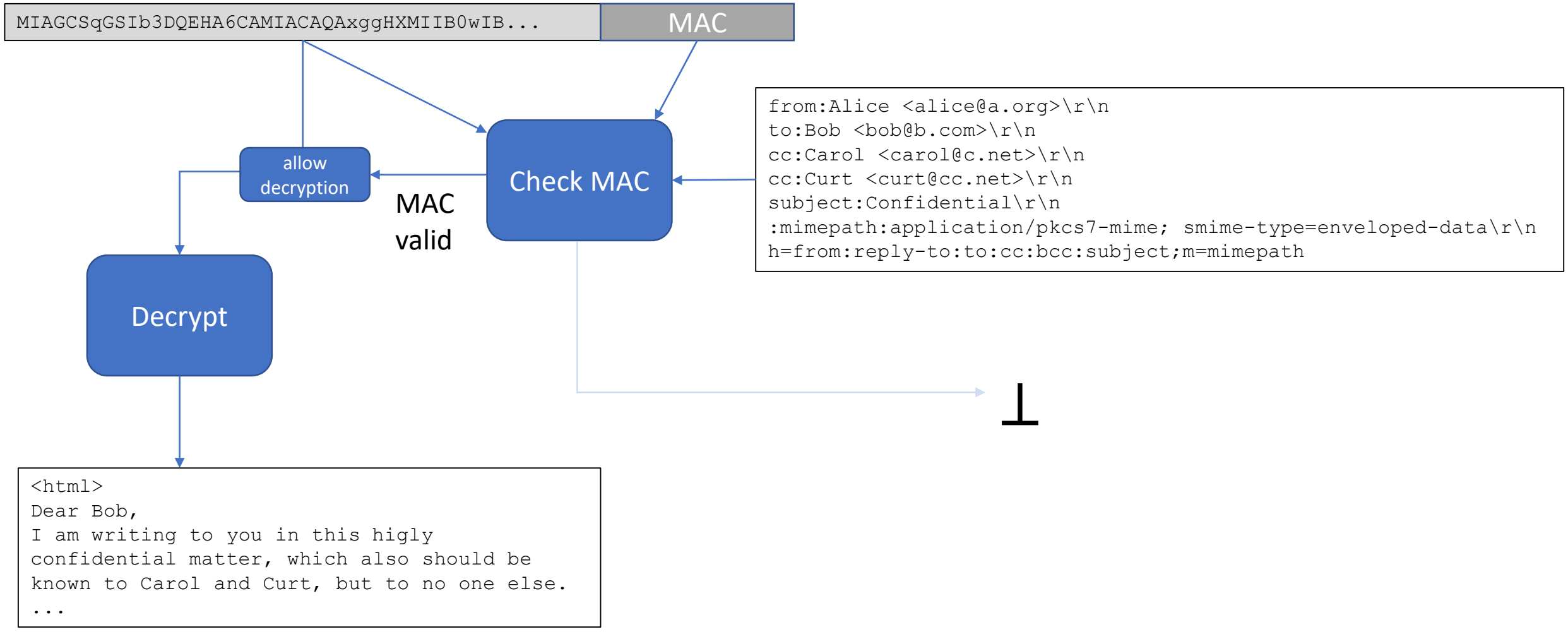
```
From: Alice <alice@a.org>
To: Bob <bob@b.com>
CC: Carol <carol@c.net>
CC: Curt <curt@cc.net>
Subject: Confidential
Decryption-Context: h=from:reply-to:to:cc:bcc:subject;m=mimepath
Content-Type: application/pkcs7-mime; smime-type=enveloped-data
Content-Transfer-Encoding: base64

MIAGCSqGSIb3DQEHA6CAMIACAQAxggHXMIIB0wIB...
```

E-mail source code

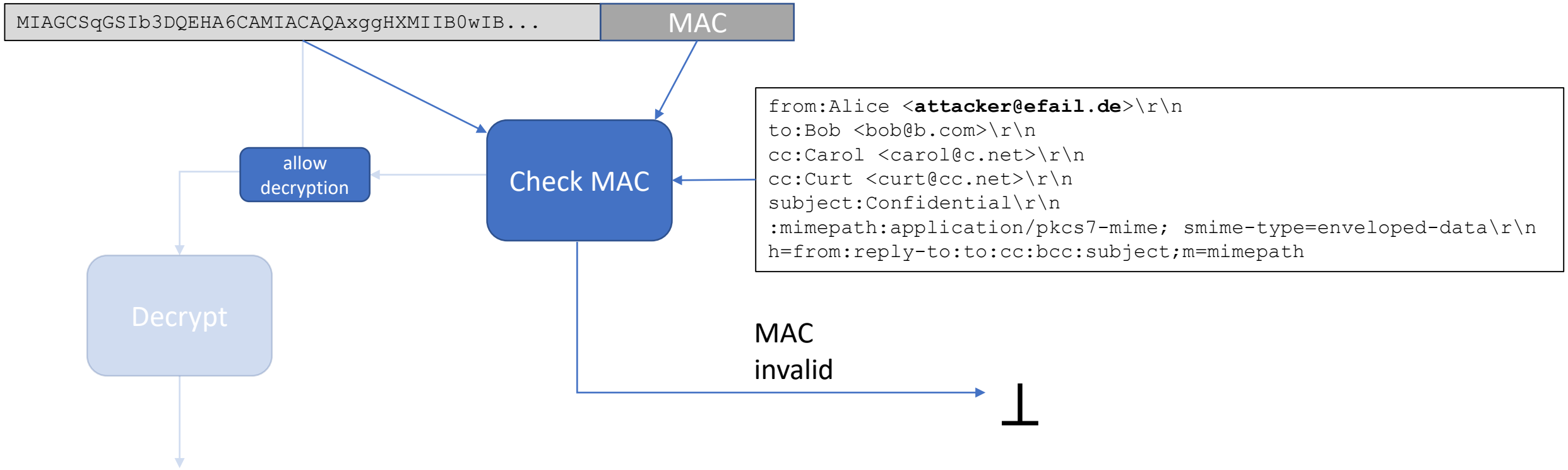
DC string

```
from:Alice <alice@a.org>\r\n
to:Bob <bob@b.com>\r\n
cc:Carol <carol@c.net>\r\n
cc:Curt <curt@cc.net>\r\n
subject:Confidential\r\n
:mimepath:application/pkcs7-mime; smime-type=enveloped-data\r\n
h=from:reply-to:to:cc:bcc:subject;m=mimepath
```

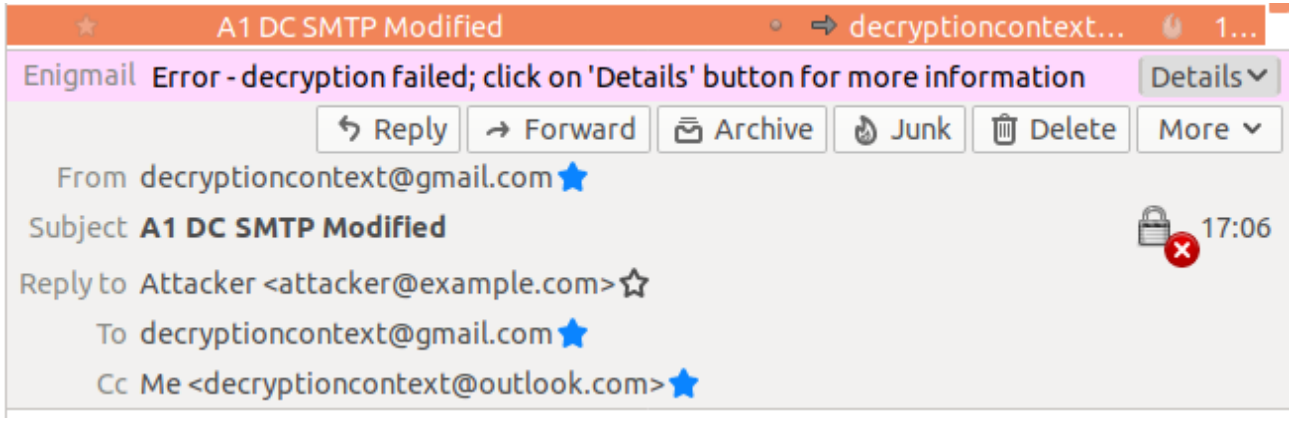
AEAD: Authenticated Encryption with Associated Data - Decryption



AEAD: Authenticated Encryption with Associated Data - Decryption



Evaluation: Implementation



Based on

- Enigmail OpenPGP plugin for Thunderbird
- Experimental version of GnuPG with AEAD support (RFC 4880-bis-08)
- 1 week, 250 LoC

Evaluation: False Positives

	Inbound	Outbound	Internal
AOL Mail	-	-	-
FastMail	-	-	-
Gmail	-	-	-
GMX Mail	-	-	-
Hushmail	○ M_1	○ B_1, M_1	-
iCloud	○ H_1, H_2	-	○ H_1, H_2
Mail.ru	-	-	-
Outlook.com	○ B_2	○ H_3, H_4 ↯ H_5, M_3	○ H_3, H_4 ↯ H_5, M_3
Runbox	-	-	-
Yahoo! Mail	-	-	-
Zoho Mail	-	-	-

-	**No changes to original headers or body.**
○	**Modifications not changing the DC string.**
B_1	Addition of \r\n at the end of the body.
H_1	Modification of letter case in some header fields.
H_2	Removal of quotes around boundary parameter in `content-type`.
H_3	Removal of `user-agent`.
H_4	Rewrite of `date` as Greenwich Mean Time.
M_1	Addition of `content-transfer-encoding` in each MIME part.
B_2	Removal of any text before first MIME part.
↯	**Modifications changing the DC string.**
H_5	Rewrite/Merging of (multiple) `from` and `to` headers.
M_3	Insertion of a new MIME part and modification of existing ones.

- Only Outlook.com (Exchange), well known issue

Published and … briefly discussed

Mitigati

Jörg Schv
Ruhr University
joerg.schwenk

Jens Mü
Ruhr University
jens.a.mueller

We held two polls, one for context for encryption and one for context for signing, asking whether to include the context parameter in the draft without further specification of what its value should be. Both polls had 9 people declare opinions, and they came out with the same results for signing and encryption. Of the 9 expressing an opinion, 2 supported including context parameters in this draft without further specification, and 7 opposed.

A followup poll, asking who thought this should be tackled by the WG after we complete the crypto refresh also had 9 participants, all of them in favor.

My conclusion from the interim is that the consensus here is rougher than we'd all like, but we will probably not include an explicit context parameter in the crypto refresh.

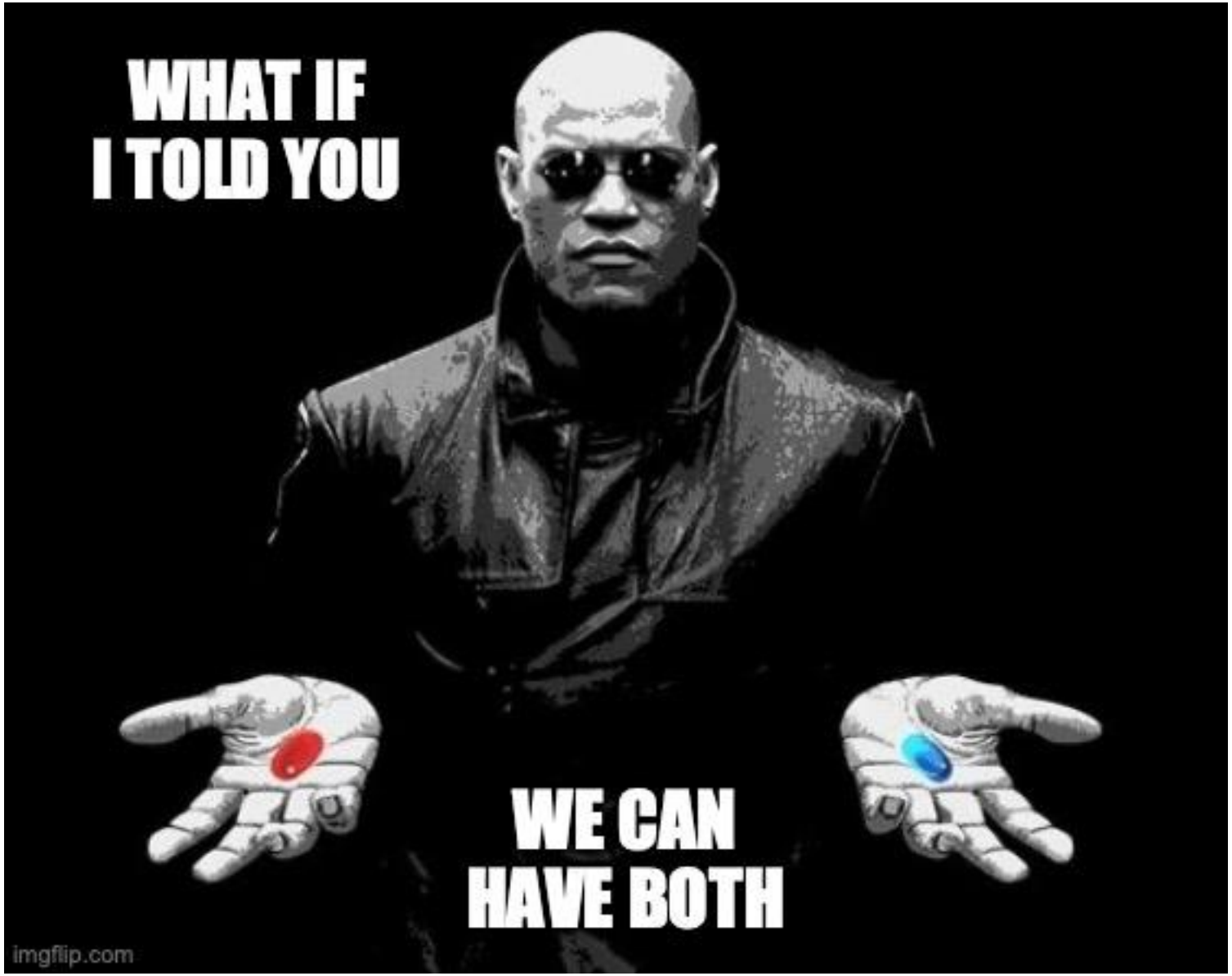
I would expect if the WG survives a rechartering, this would be one of the top priority items.

--dkg

| Show header

Summary: Instant Messaging vs. Secure E-Mail

	S/MIME & OpenPGP	Ratcheting IM
E2EE	message level	transport level
Security properties	Hybrid PKE, stateless	Forward Secrecy, Future Secrecy, stateful
Message storage	Hybrid PKE, stateless	Cleartext, local encryption
Attacks	EFAIL direct exfiltration, REPLY attacks	Threema, Telegram, ???


WHAT IF
I TOLD YOU
WE CAN
HAVE BOTH
imgflip.com

Thanks for your attention!