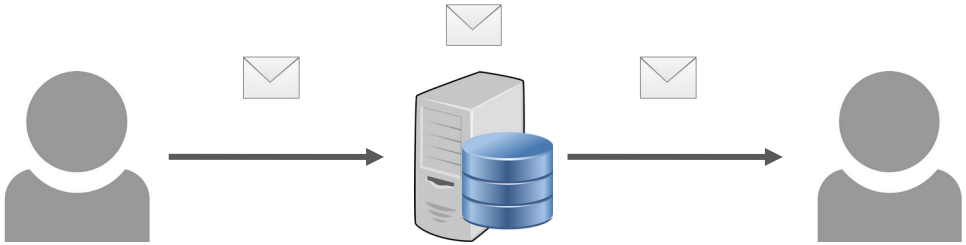
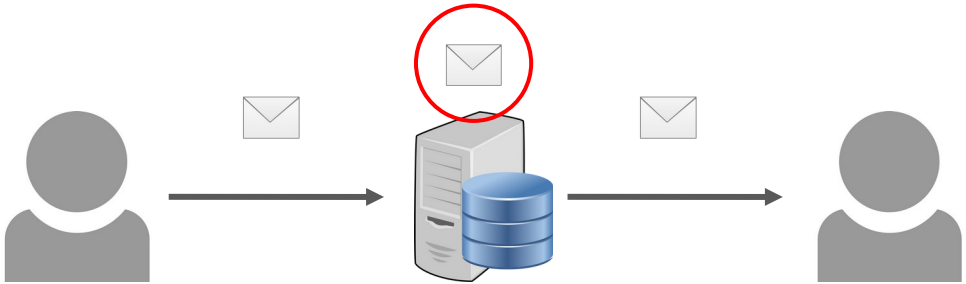
Onion Franking: Abuse Reports for Mix-Based Private Messaging

Matthew Gregoire, Margaret Pierce, Saba Eskandarian

Metadata in E2EE messaging contexts

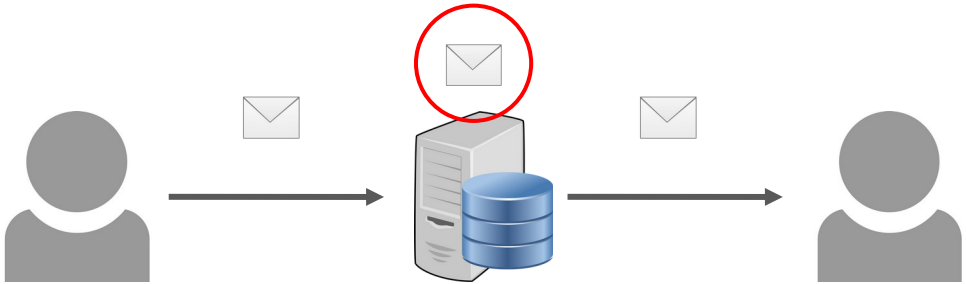


Metadata in E2EE messaging contexts



Metadata in E2EE messaging contexts

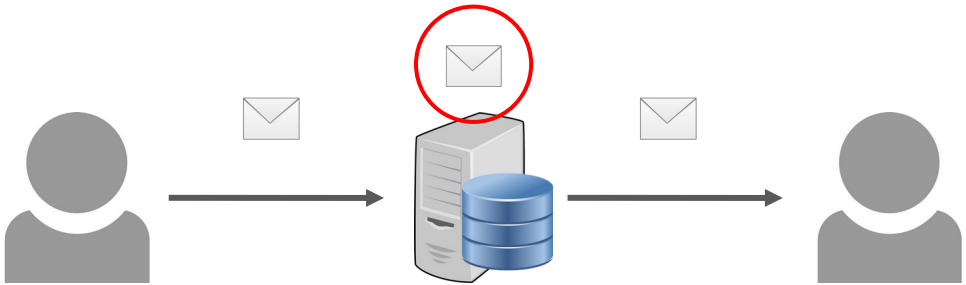
Often just as sensitive as message contents…



Metadata in E2EE messaging contexts

Often just as sensitive as message contents…

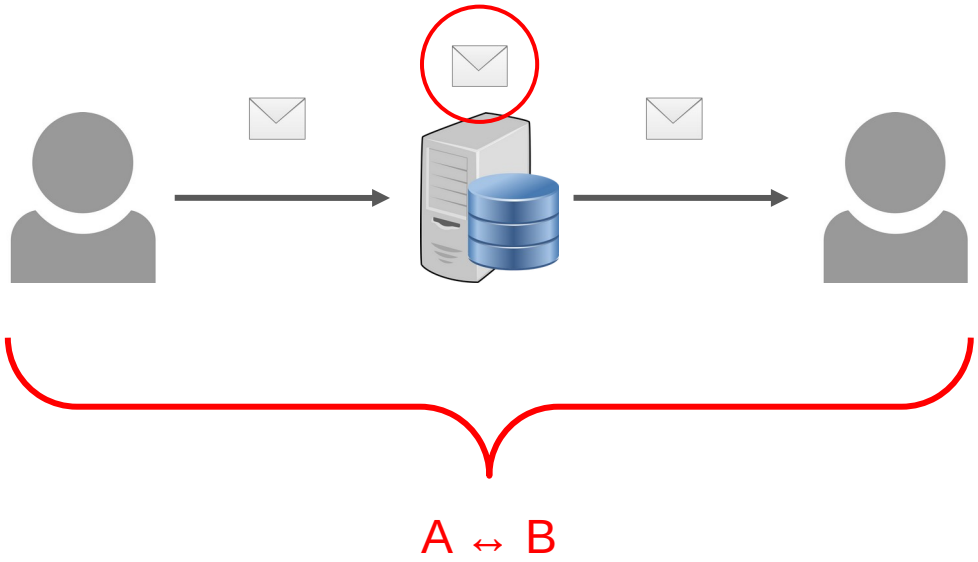
Is who is talking with whom



Metadata in E2EE messaging contexts

Often just as sensitive as message contents…

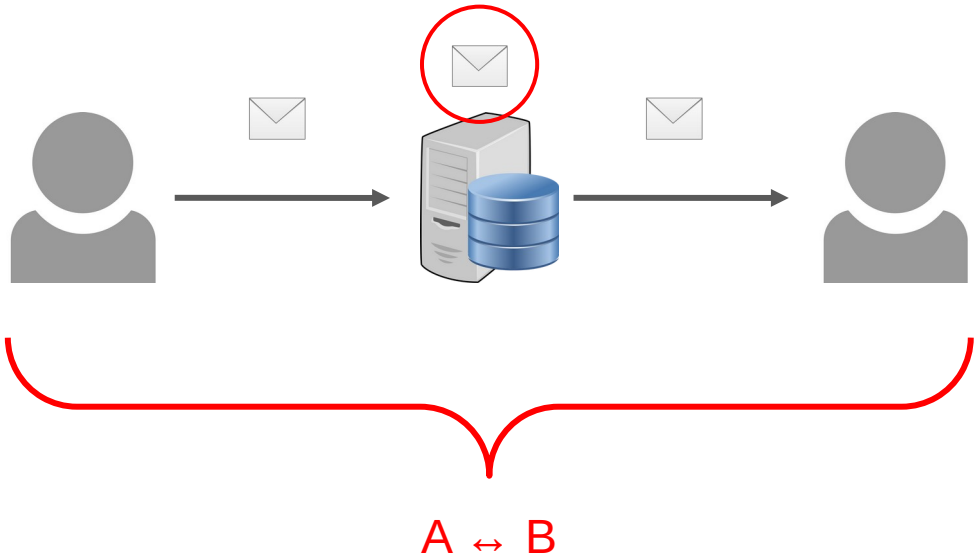
Is who is talking with whom



Metadata in E2EE messaging contexts

Often just as sensitive as message contents…

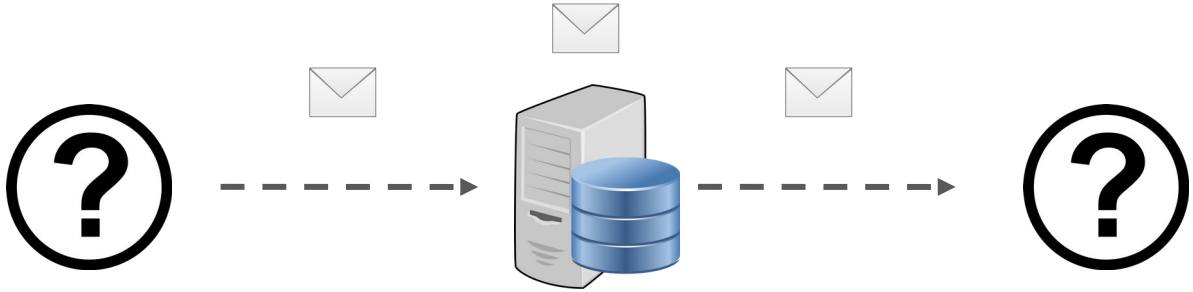
Is who is talking with whom



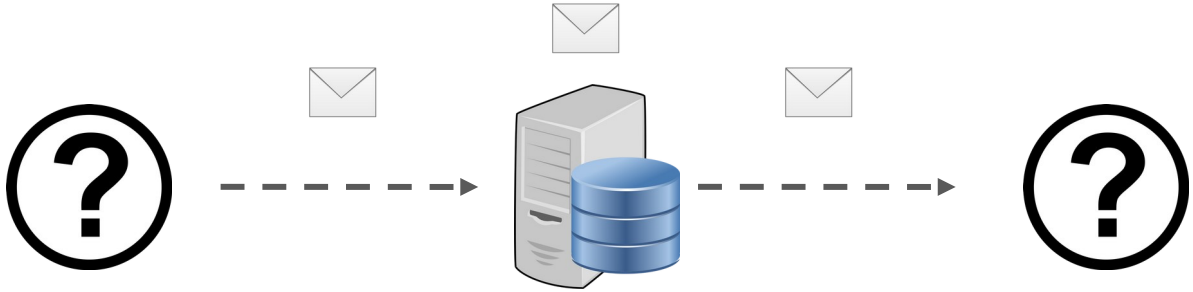
“We kill people based on metadata”

- Michael Hayden, former NSA director

Metadata-Hiding Messaging

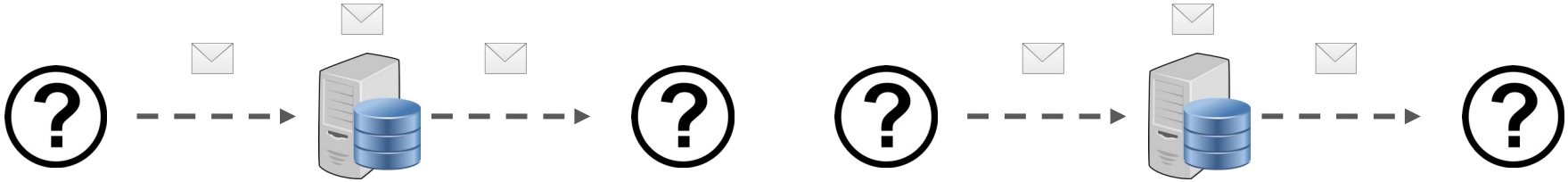


Metadata-Hiding Messaging

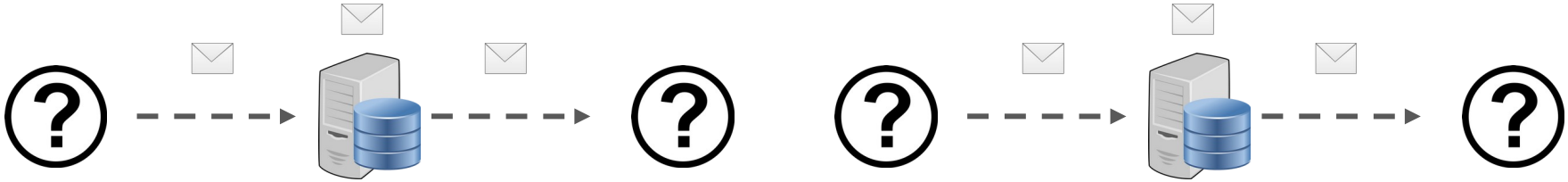


Well-known techniques (e.g. mixnets, DC nets) allow for metadata hiding

But what exactly is our threat model?

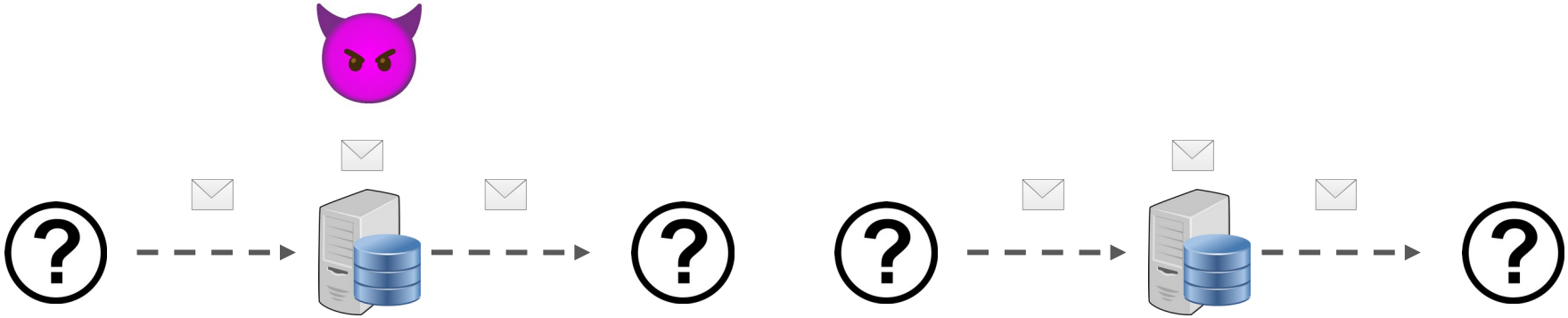


But what exactly is our threat model?



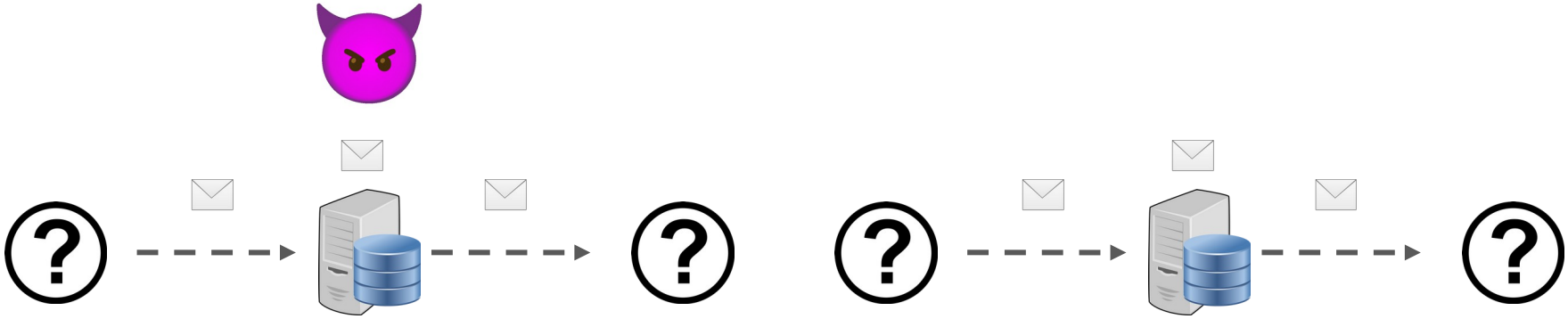
Are we worried about surveillance?

But what exactly is our threat model?



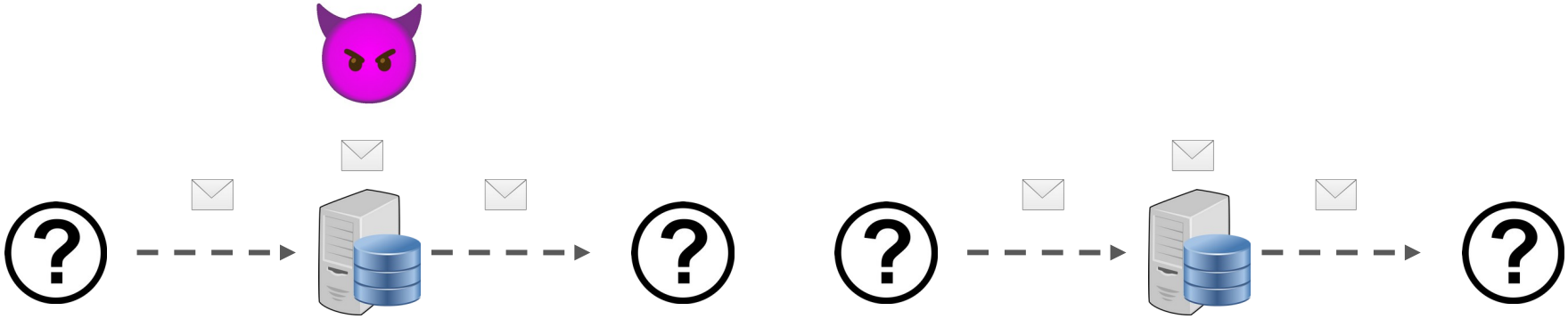
Are we worried about surveillance?

But what exactly is our threat model?



Are we worried about surveillance? ✅

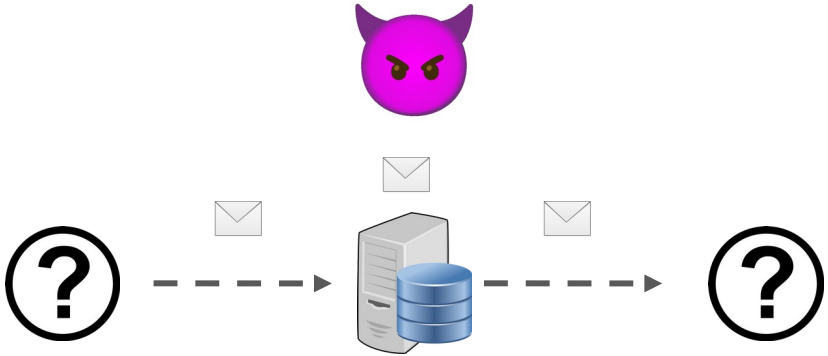
But what exactly is our threat model?

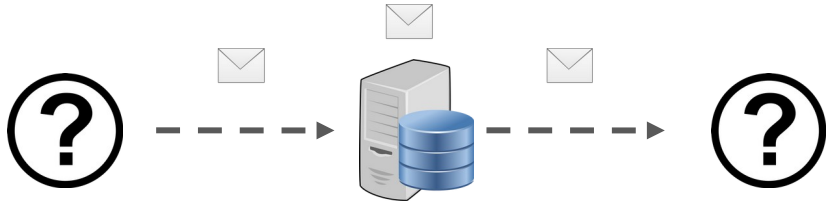


Are we worried about surveillance? ✅

No messages or metadata can be read

But what exactly is our threat model?



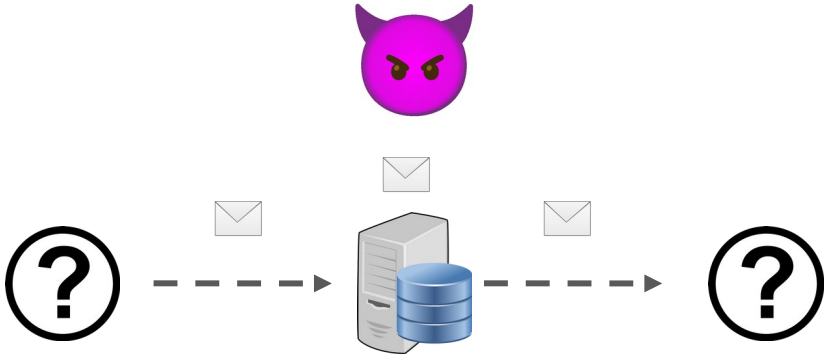


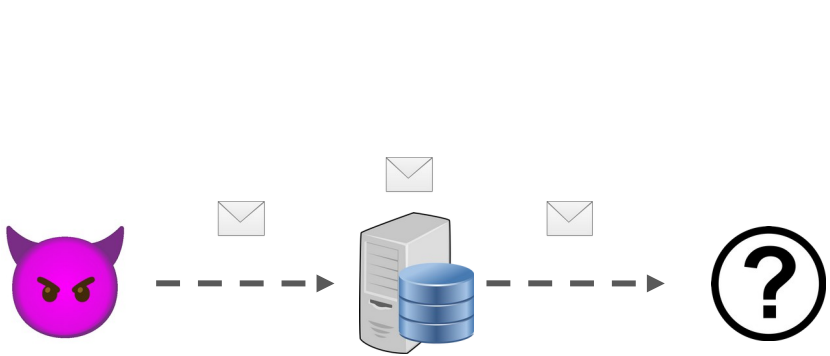
Are we worried about surveillance? ✅

Are we worried about abusive messages?

No messages or metadata can be read

But what exactly is our threat model?





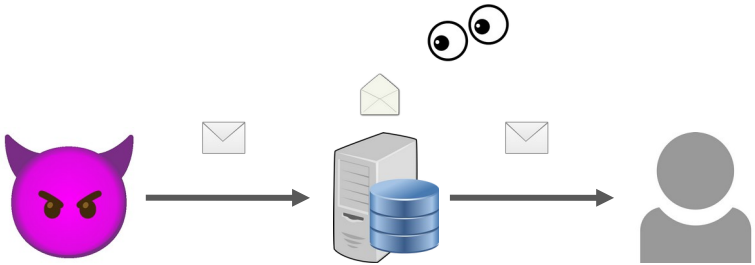
Are we worried about surveillance? ✅

Are we worried about abusive messages?

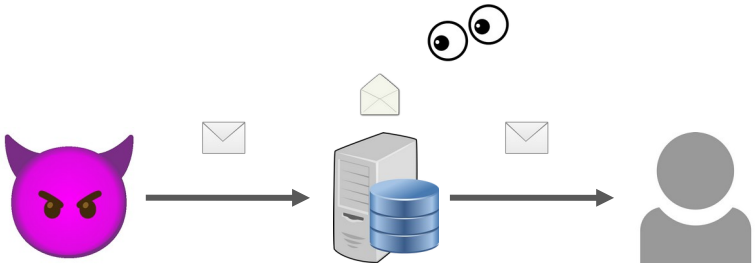
No messages or metadata can be read

Malicious senders

Malicious senders

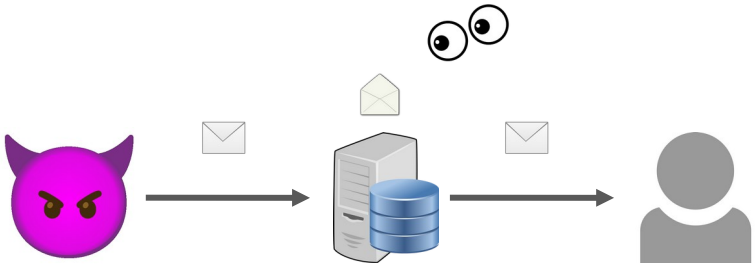


Malicious senders



Without E2EE:

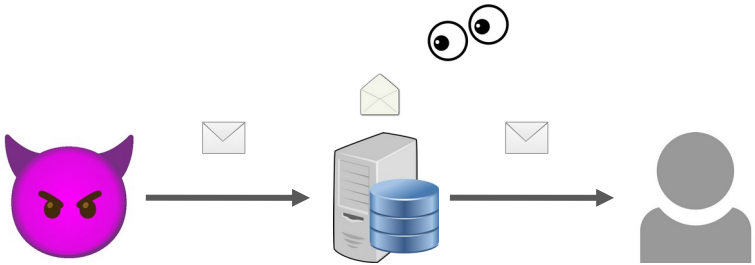
Malicious senders

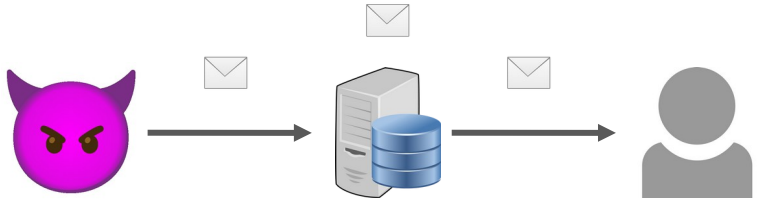


Without E2EE:

The platform can easily verify and moderate reported messages

Malicious senders

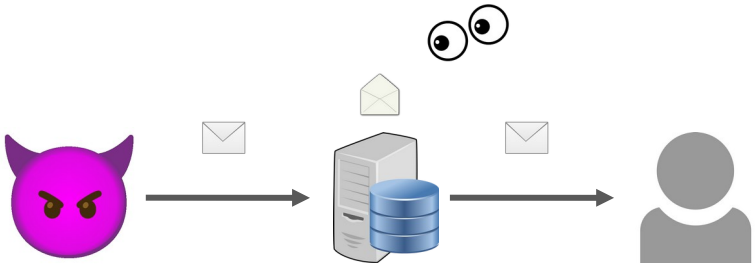


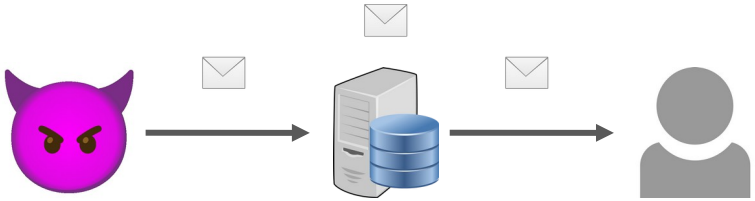


Without E2EE:

The platform can easily verify and moderate reported messages

Malicious senders



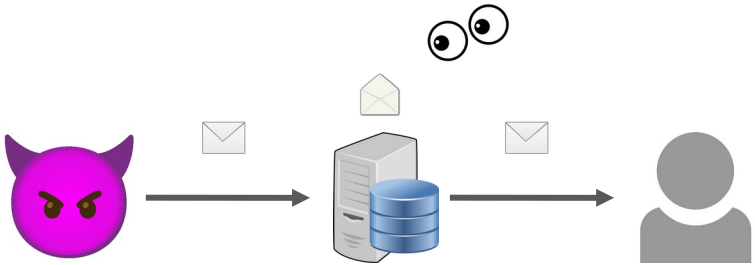


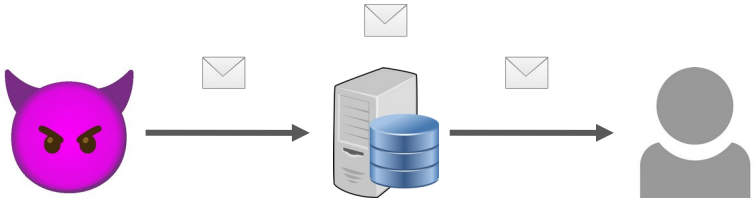
Without E2EE:

The platform can easily verify and moderate reported messages

Recourse no longer seems possible in the E2EE setting ❌

Malicious senders





Without E2EE:

The platform can easily verify and moderate reported messages

Recourse no longer seems possible in the E2EE setting ❌

This is even worse for metadata hiding!

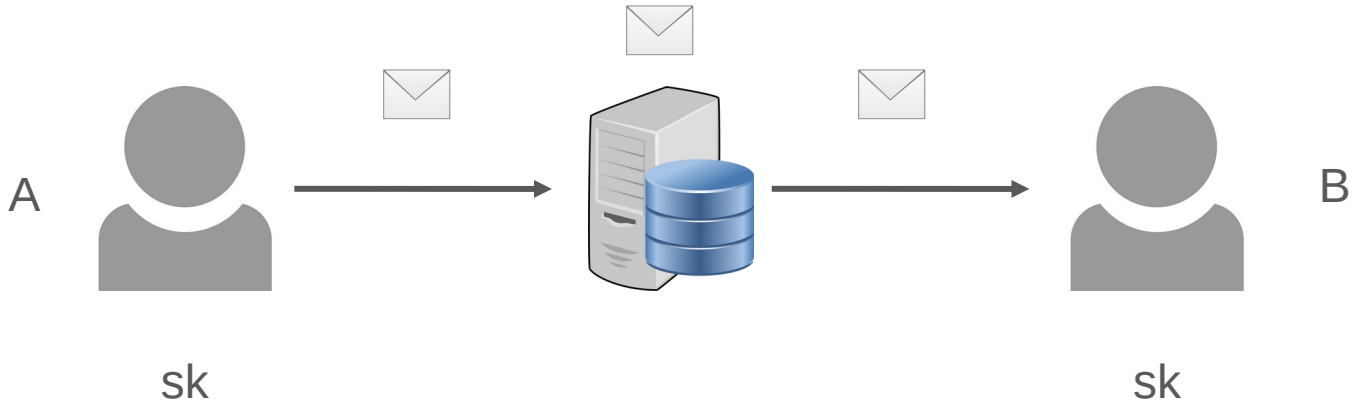
Abuse Reporting: Message Franking

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

Abuse Reporting: Message Franking

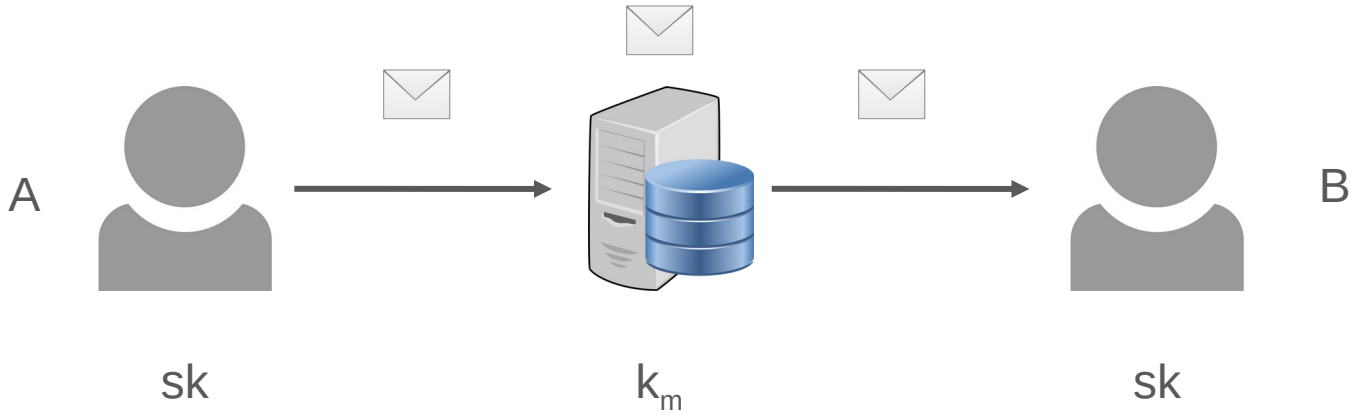
Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

Abuse Reporting: Message Franking



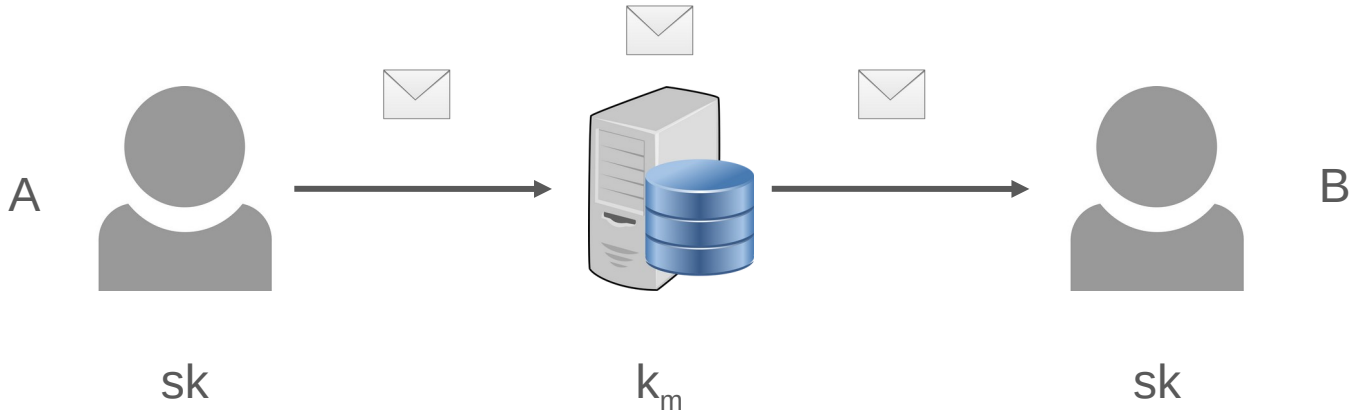
Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

Abuse Reporting: Message Franking



Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

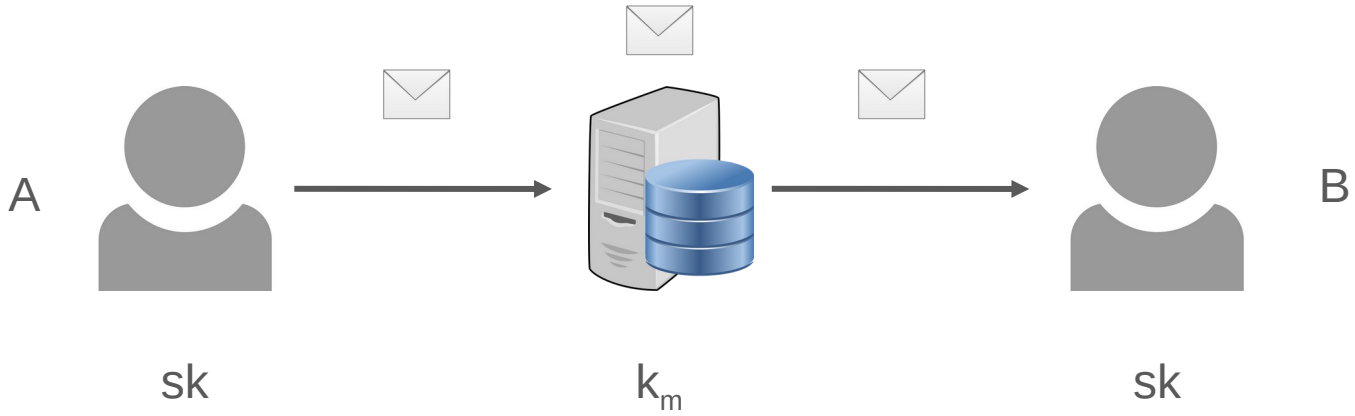
Abuse Reporting: Message Franking



c_1: Message encryption

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

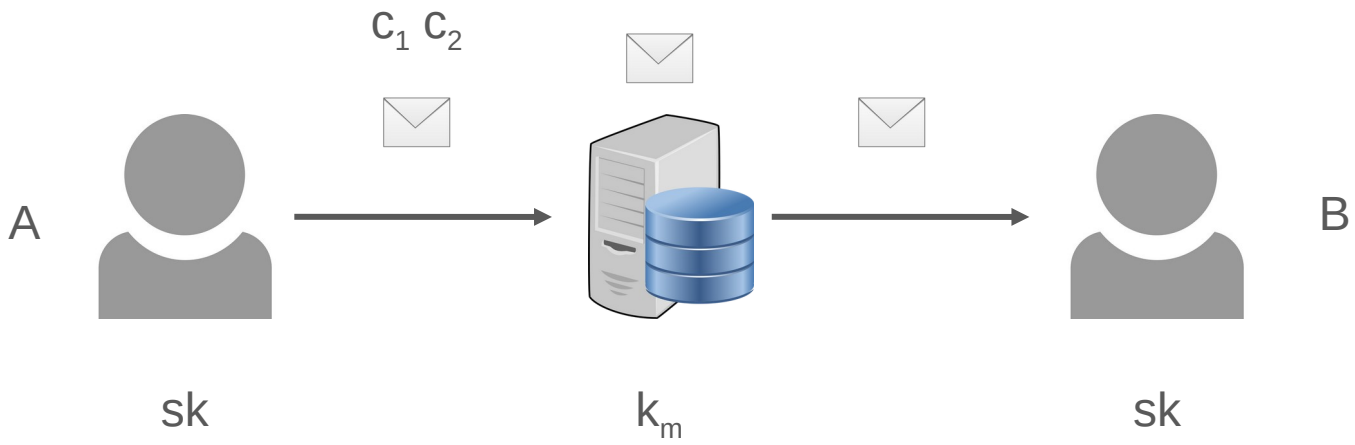
Abuse Reporting: Message Franking



c_1: Message encryption
c_2: Message commitment

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

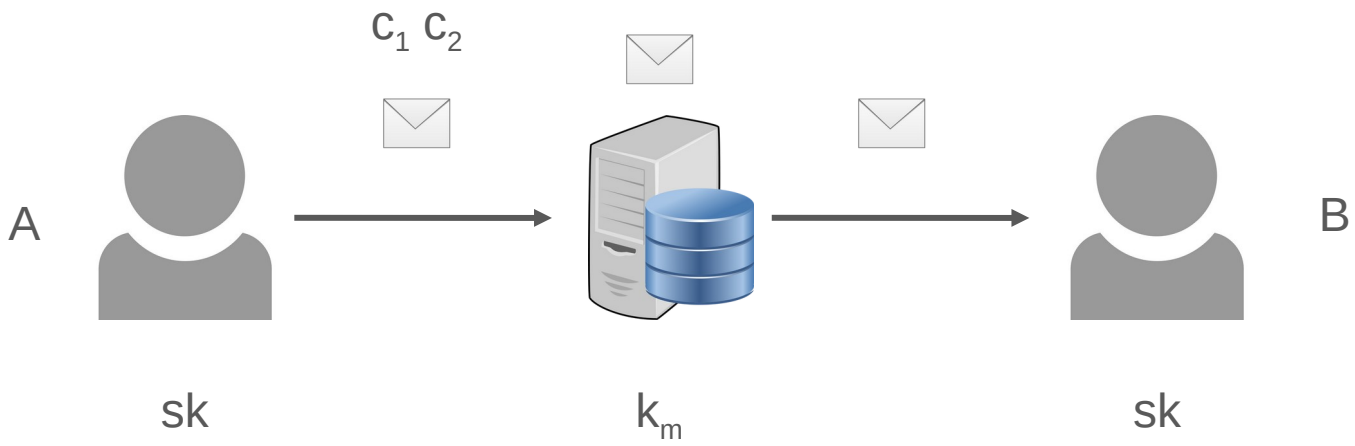
Abuse Reporting: Message Franking



c_1: Message encryption
c_2: Message commitment

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

Abuse Reporting: Message Franking

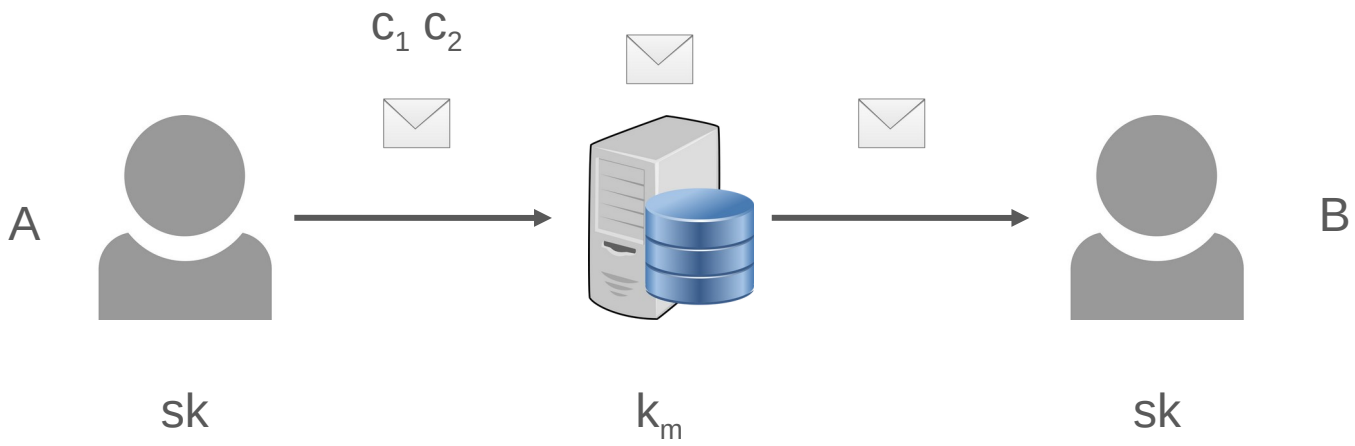


c_1: Message encryption
c_2: Message commitment

ctx: (A, B, timestamp)

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. “Messenger Secret Conversations technical whitepaper.”

Abuse Reporting: Message Franking

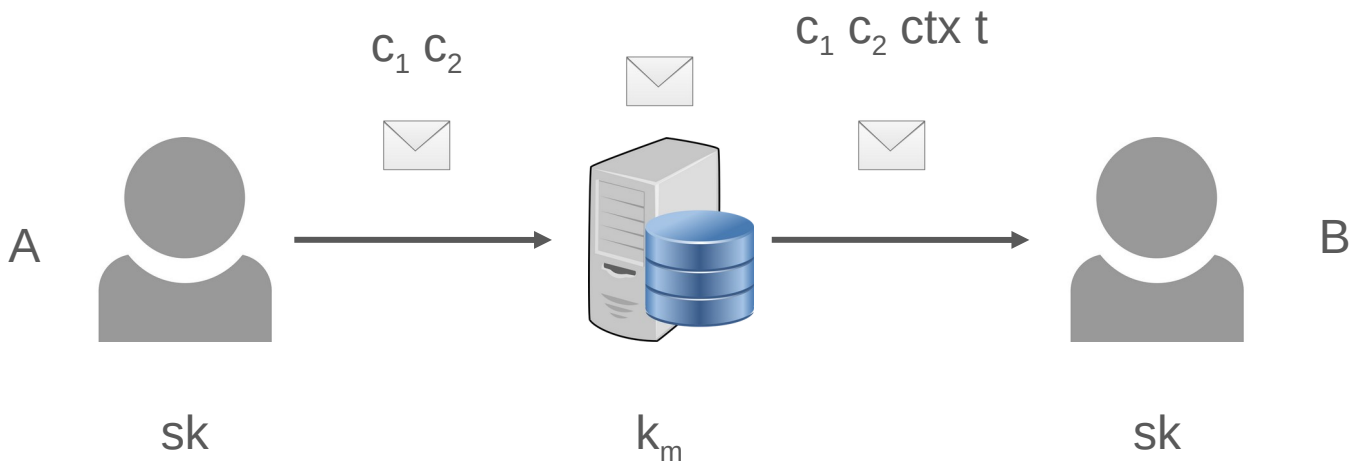


c_1: Message encryption
c_2: Message commitment

ctx: (A, B, timestamp)
t: Platform's tag

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. "Messenger Secret Conversations technical whitepaper."

Abuse Reporting: Message Franking

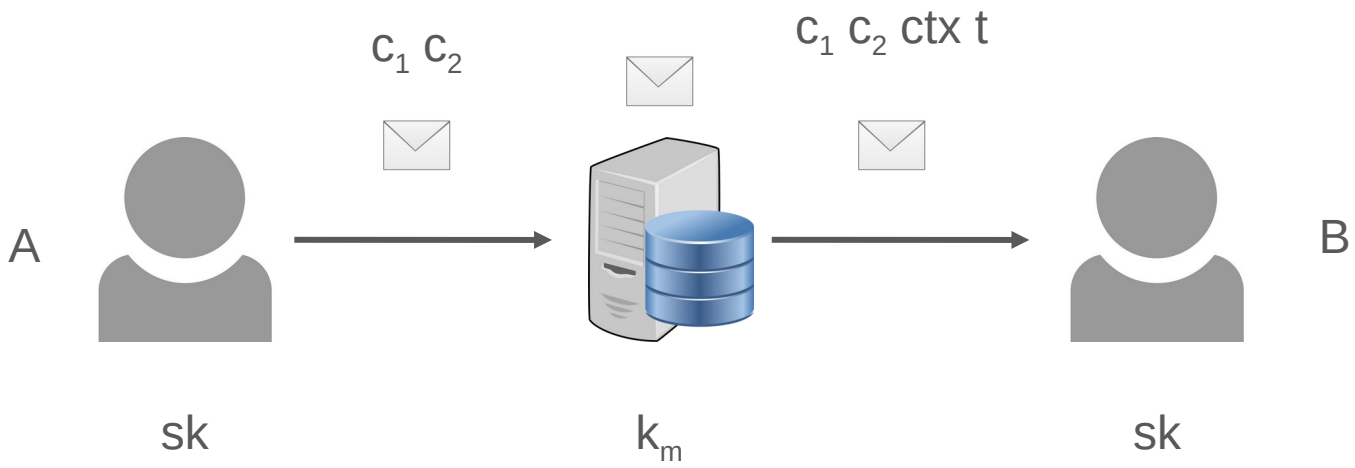


c_1: Message encryption
c_2: Message commitment

ctx: (A, B, timestamp)
t: Platform's tag

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. "Messenger Secret Conversations technical whitepaper."

Abuse Reporting: Message Franking



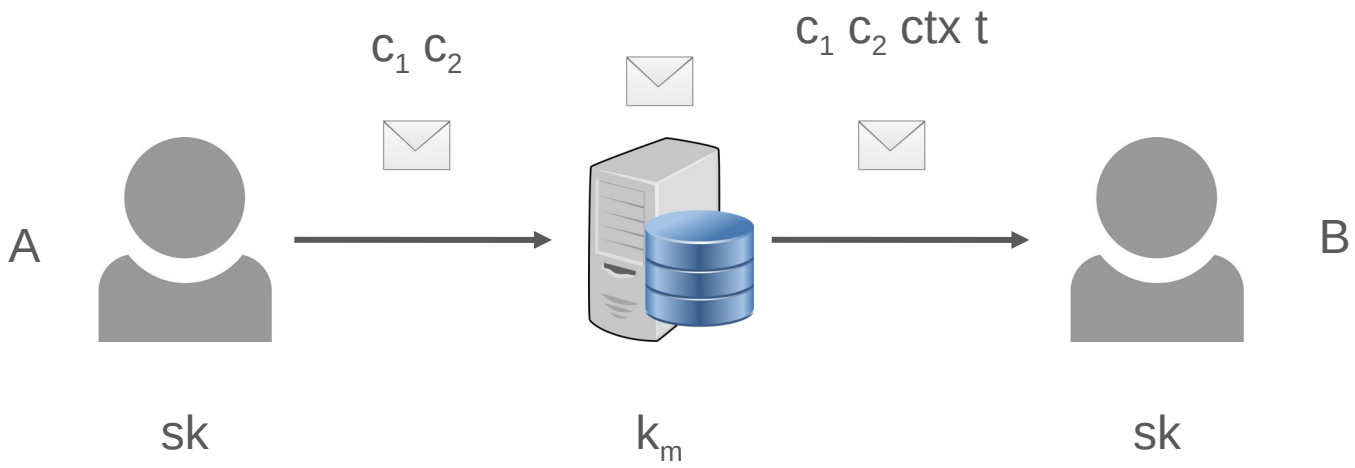
c_1: Message encryption
c_2: Message commitment

ctx: (A, B, timestamp)
t: Platform's tag

If (c_1, c_2) is well formed, display message

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. "Messenger Secret Conversations technical whitepaper."

Abuse Reporting: Message Franking



c_1: Message encryption
c_2: Message commitment

ctx: (A, B, timestamp)
t: Platform's tag

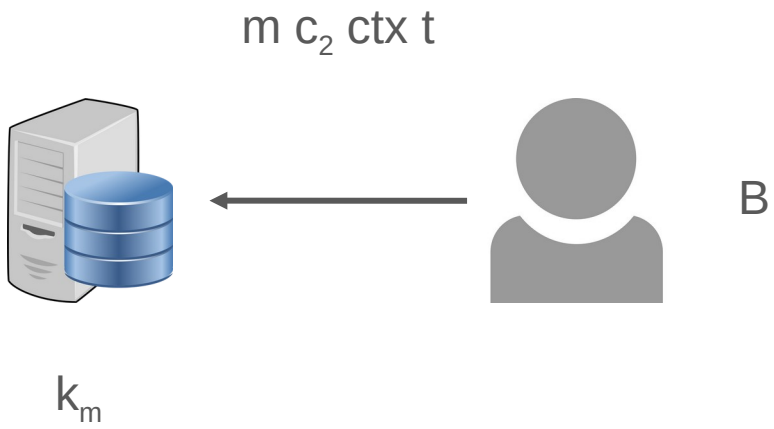
If (c_1, c_2) is well formed, display message

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message franking via committing authenticated encryption." *Advances in Cryptology–CRYPTO 2017*
Facebook. "Messenger Secret Conversations technical whitepaper."

Message Franking (Reporting)

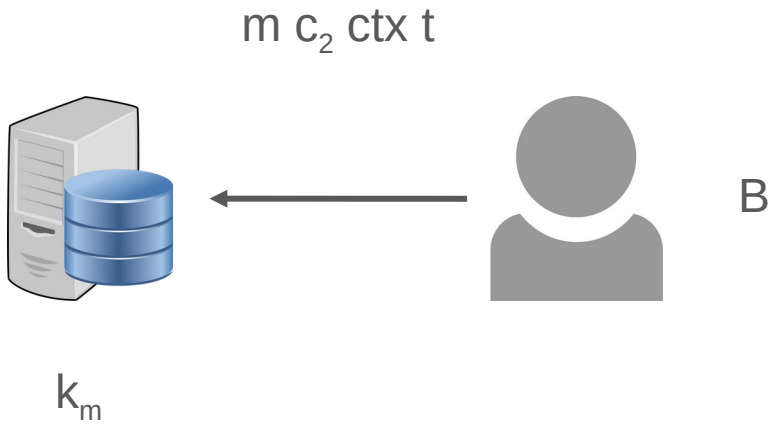
k_m

Message Franking (Reporting)



Message Franking (Reporting)

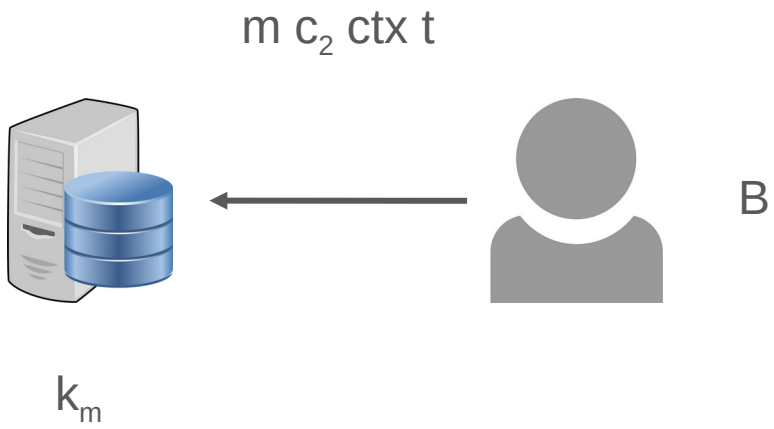
The receiver sends franking data back to the platform



Message Franking (Reporting)

The receiver sends franking data back to the platform

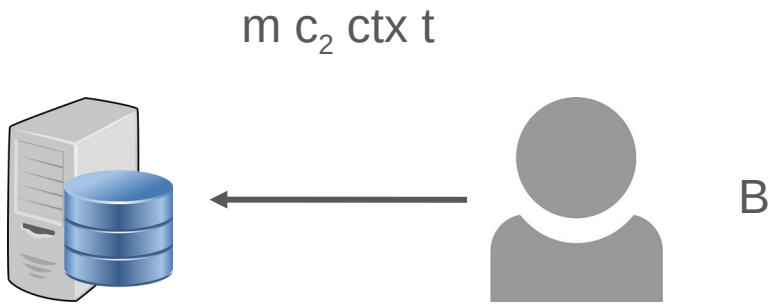
If valid: report accepted ✅



Message Franking (Reporting)

The receiver sends franking data back to the platform

If valid: report accepted ✅

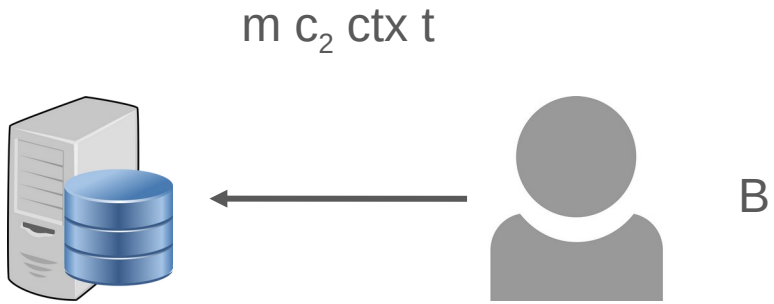


Summary

Message Franking (Reporting)

The receiver sends franking data back to the platform

If valid: report accepted ✅



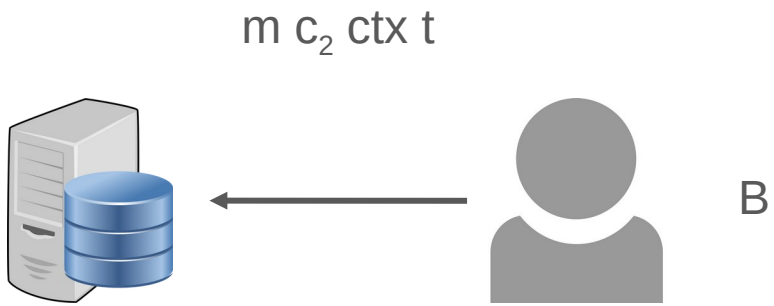
Summary

Just a little bookkeeping enables abuse reporting on E2EE messages!

Message Franking (Reporting)

The receiver sends franking data back to the platform

If valid: report accepted ✅

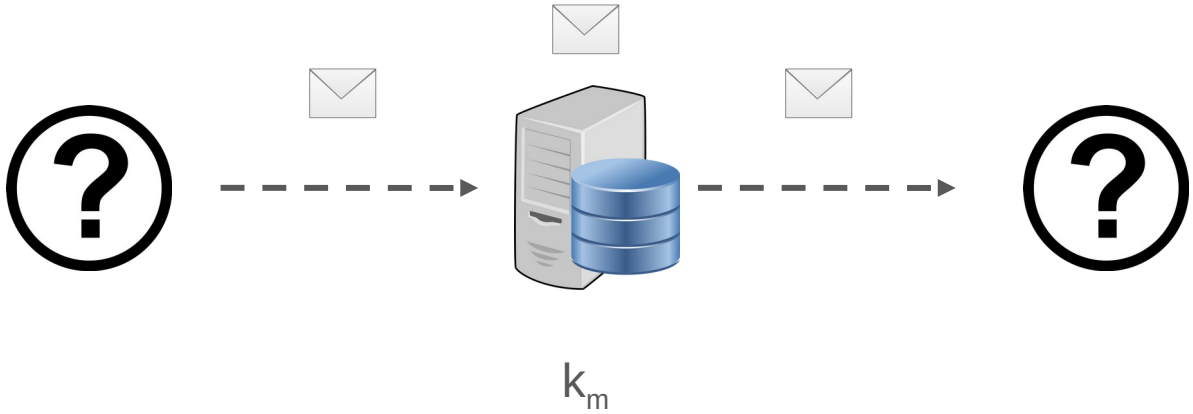


Summary

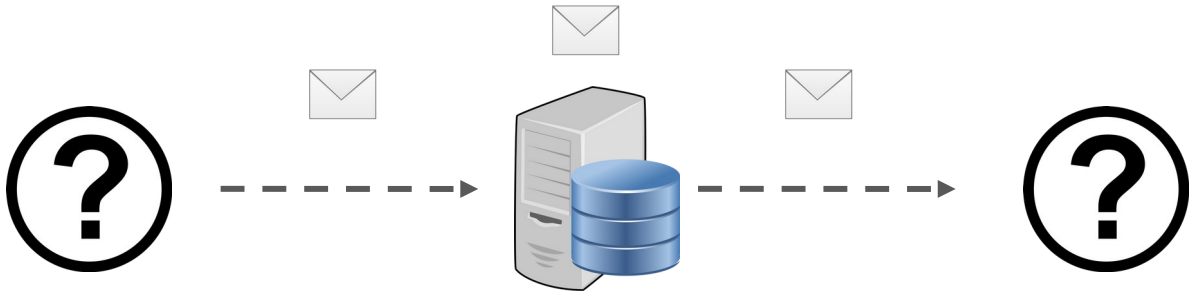
Just a little bookkeeping enables abuse reporting on E2EE messages!

No party is able to exploit this reporting mechanism as a new vector for abuse.

Metadata-Hiding Message Franking (first try)



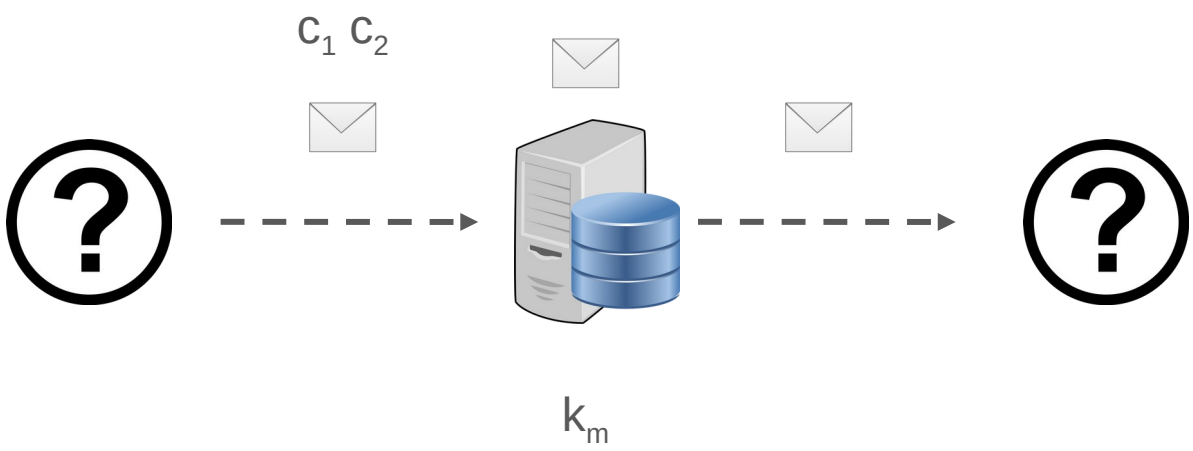
Metadata-Hiding Message Franking (first try)



$c_1 \leftarrow \text{Enc}(sk, (m, r))$

$c_2 \leftarrow \text{Commit}(m, r)$

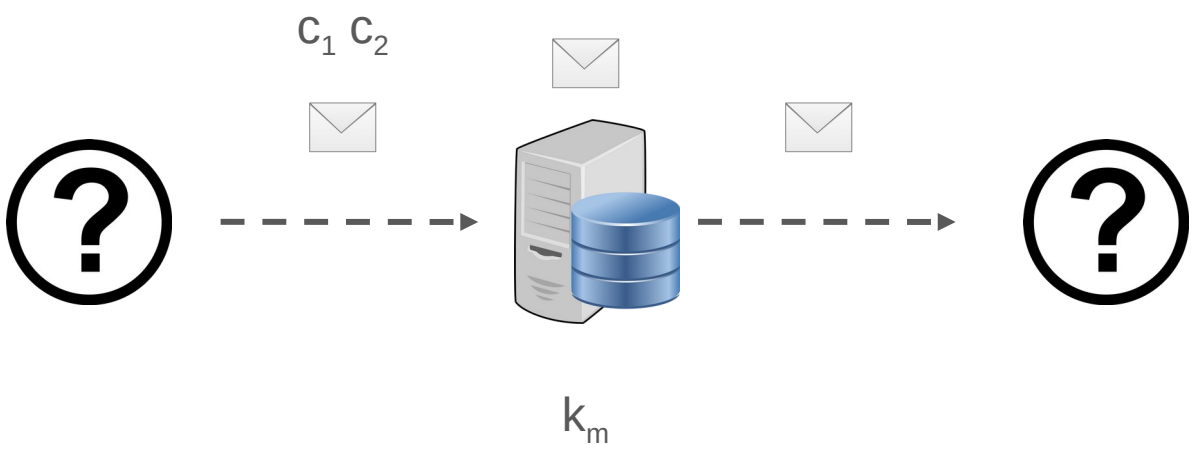
Metadata-Hiding Message Franking (first try)



$c_1 \leftarrow \text{Enc}(sk, (m, r))$

$c_2 \leftarrow \text{Commit}(m, r)$

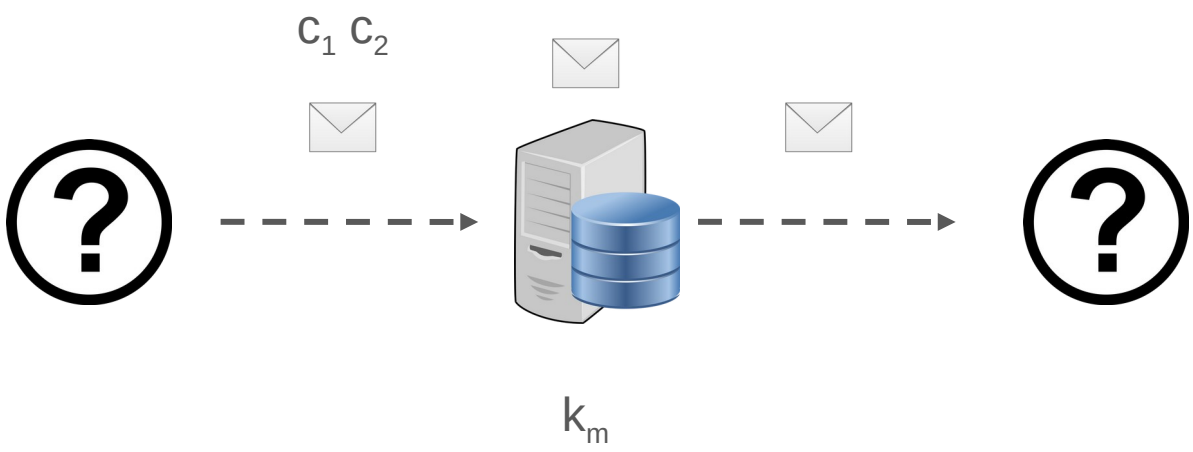
Metadata-Hiding Message Franking (first try)



$c_1 \leftarrow \text{Enc}(sk, (m, r))$
$c_2 \leftarrow \text{Commit}(m, r)$

$ctx \leftarrow (A, B, \text{timestamp})$ ❌
$t \leftarrow \text{MAC}(k_m, (c_2, ctx))$

Metadata-Hiding Message Franking (first try)

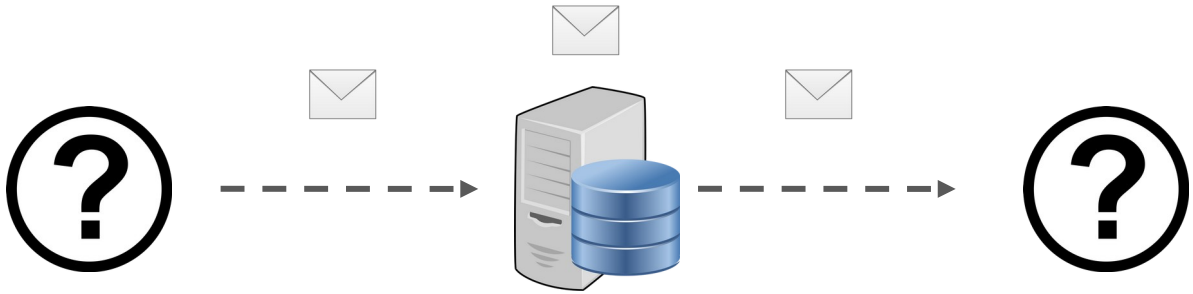


$c_1 \leftarrow \text{Enc}(sk, (m, r))$
$c_2 \leftarrow \text{Commit}(m, r)$

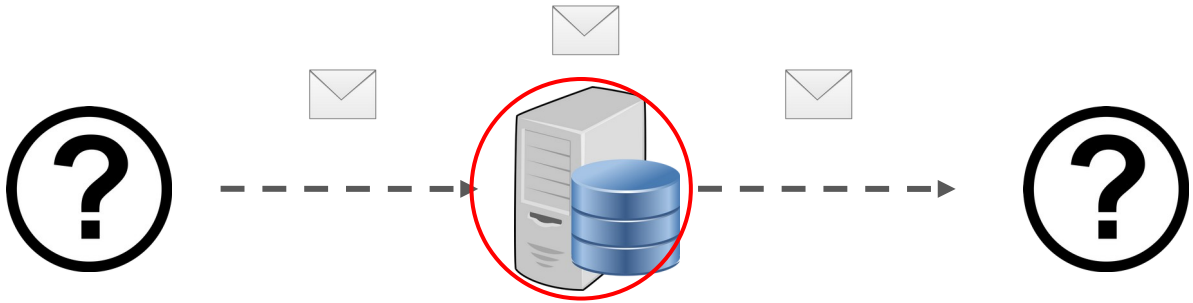
$ctx \leftarrow (A, B, \text{timestamp})$ ❌
$t \leftarrow \text{MAC}(k_m, (c_2, ctx))$

The platform can't later be convinced that a specific user sent a reported message!

Metadata-Hiding Message Franking

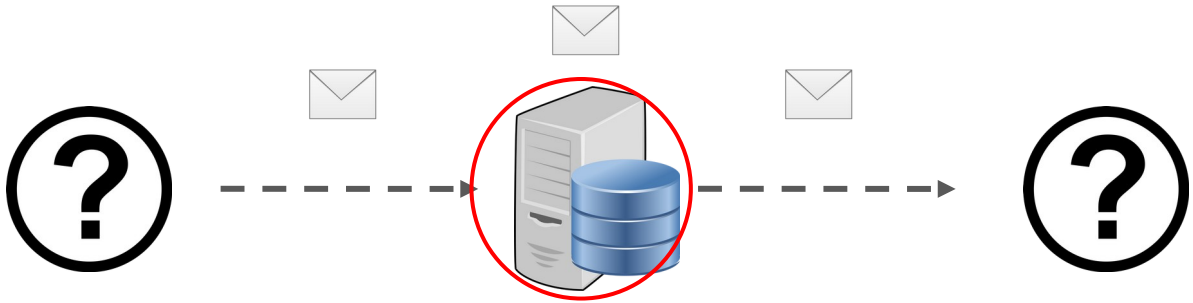


Metadata-Hiding Message Franking



Can we use the structure of the messaging platform to our advantage?

Metadata-Hiding Message Franking



Can we use the structure of the messaging platform to our advantage?

This is (in general) tricky to solve efficiently! We'll take advantage of the structure of onion encryption

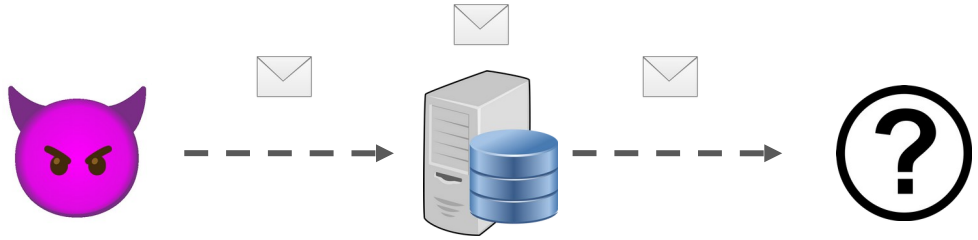
Our contribution: **Onion Franking**

We provide abuse reporting

with strong performance

for metadata-hiding messaging platforms

based on onion encryption.



Our contribution: **Onion Franking**

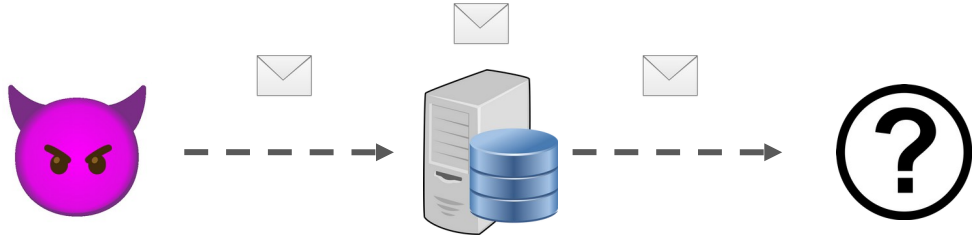
We provide abuse reporting

with strong performance

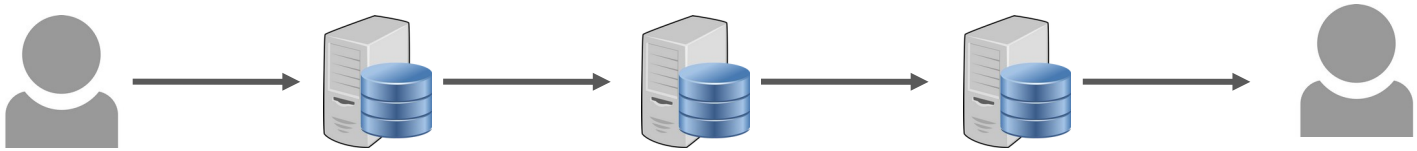
for metadata-hiding messaging platforms

based on onion encryption.

Along the way, we need to break previous franking abstractions to achieve stronger security.

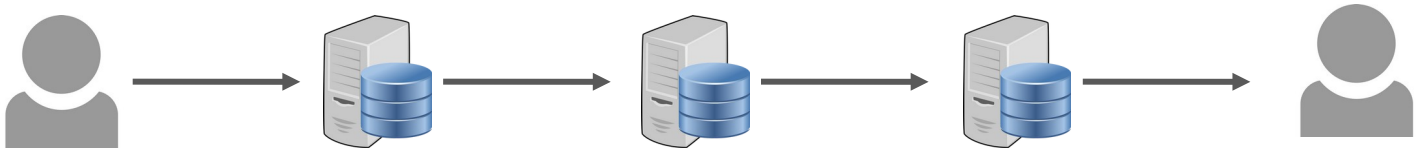


Onion Encryption



Onion Encryption

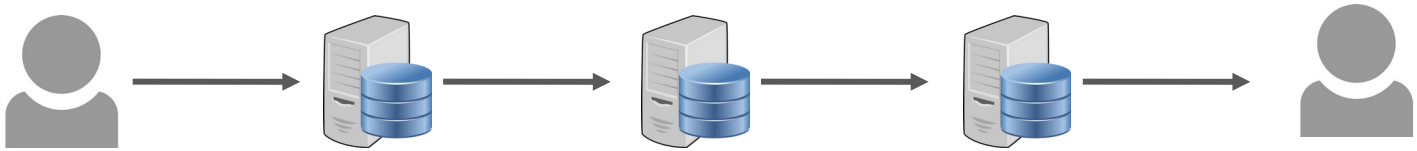
Used in:



Onion Encryption

Used in:

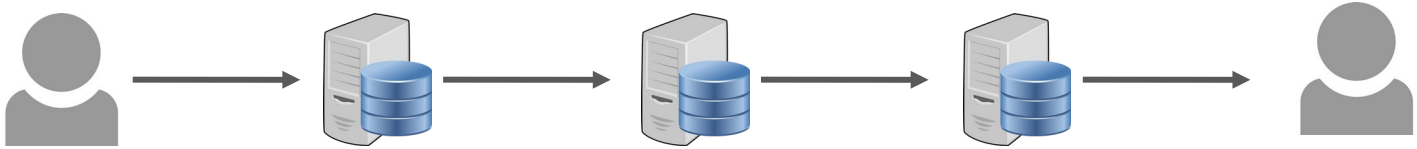
- Mixnets



Onion Encryption

Used in:

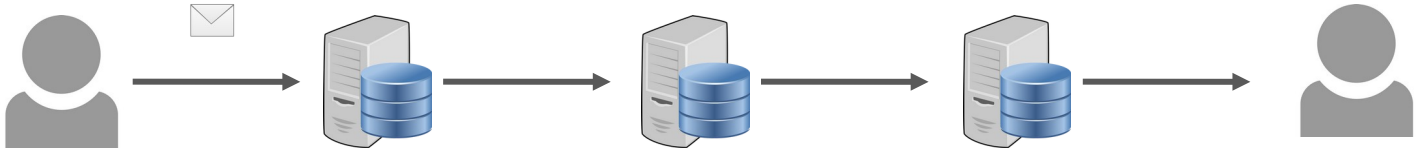
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

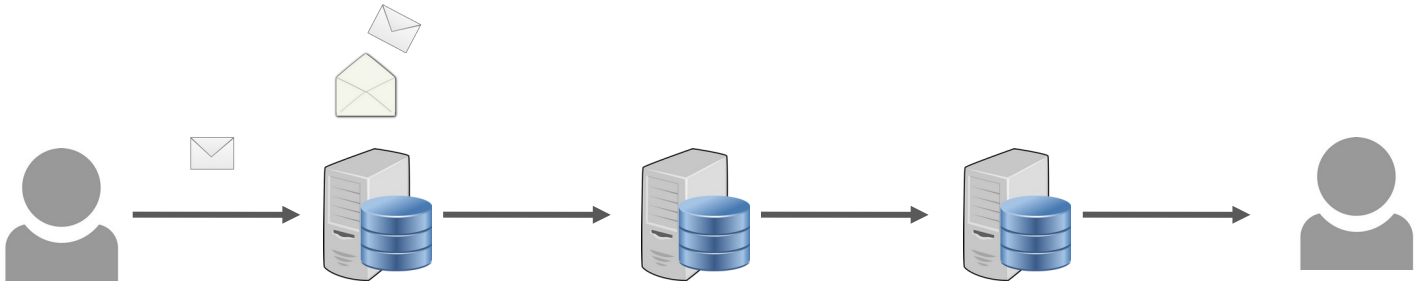
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

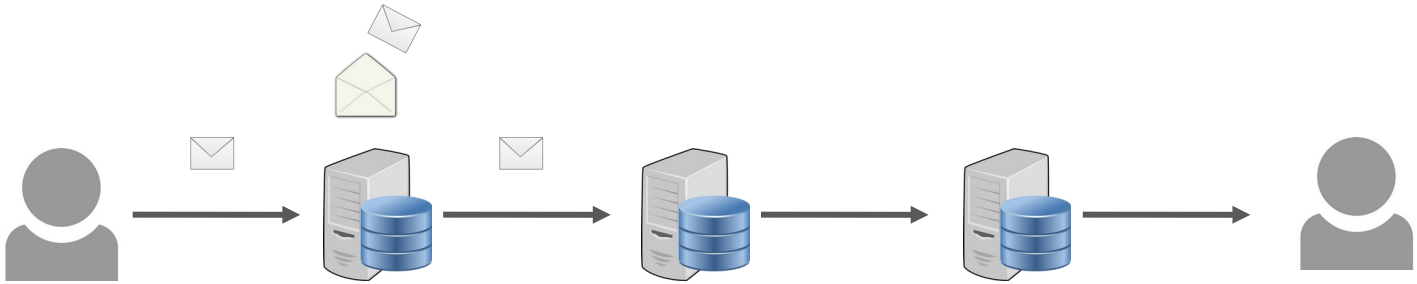
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

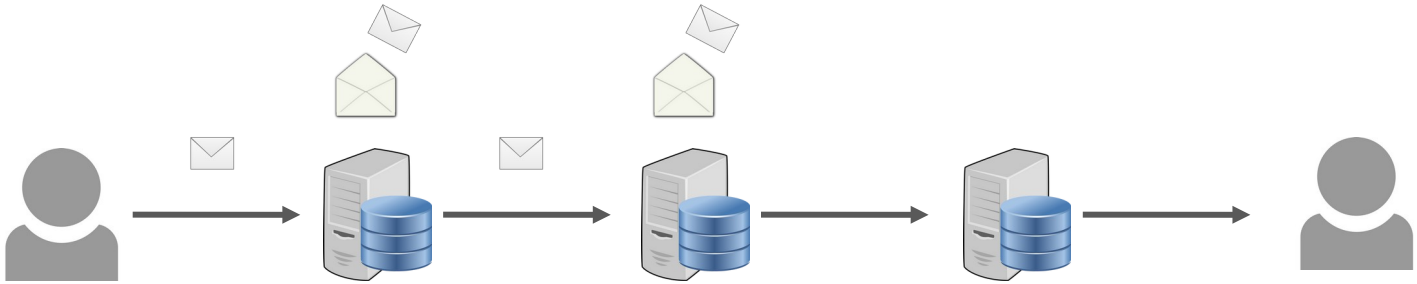
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

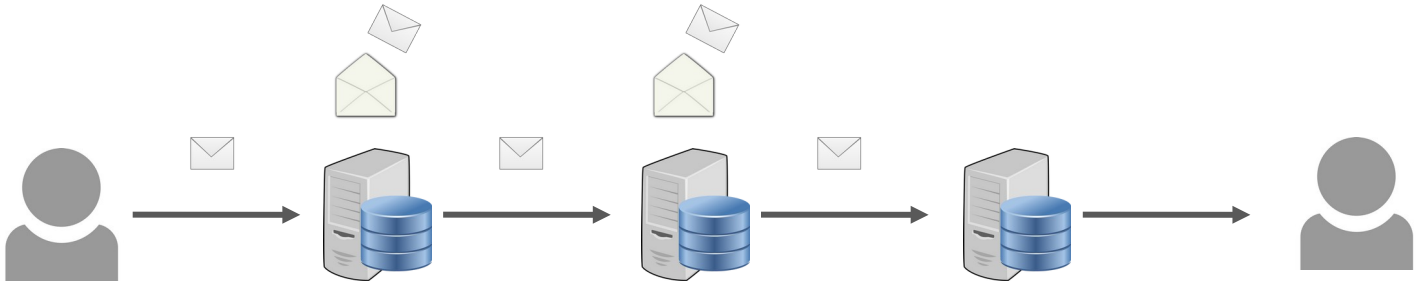
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

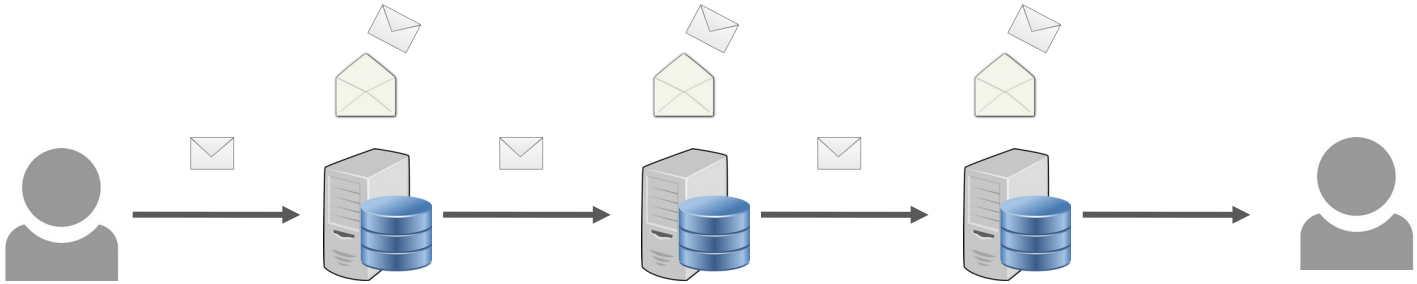
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

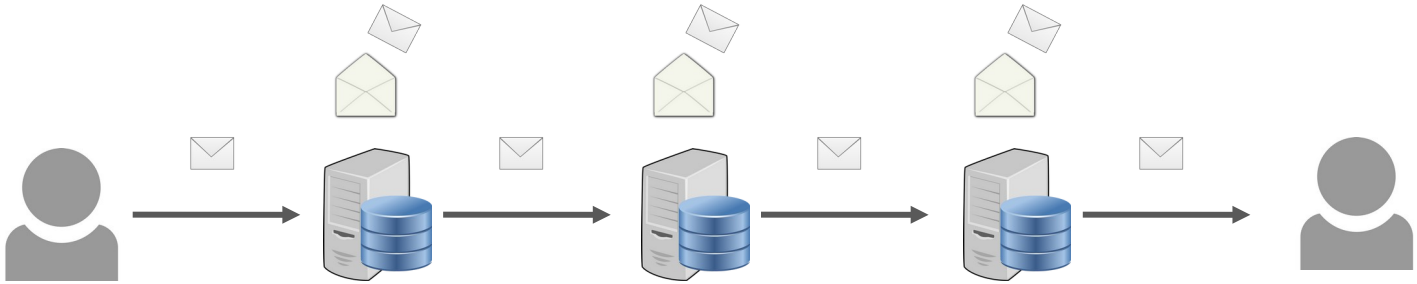
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

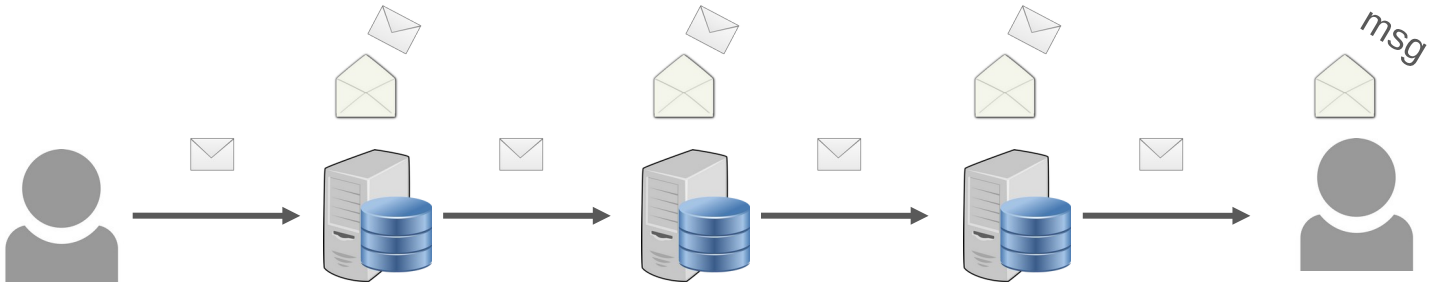
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

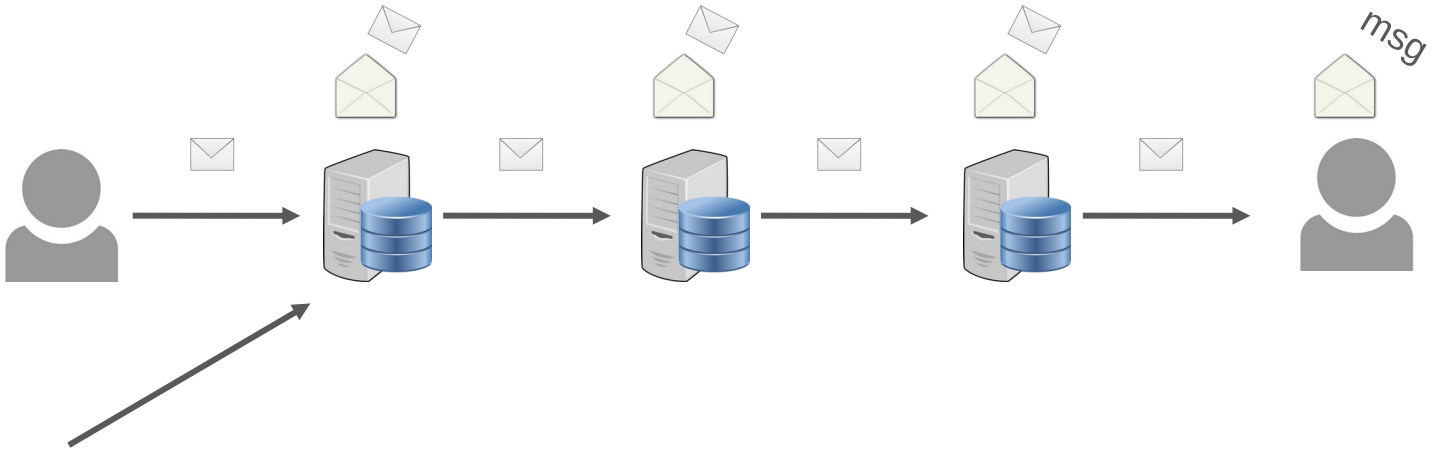
- Mixnets
- Onion routing (e.g. Tor)



Onion Encryption

Used in:

- Mixnets
- Onion routing (e.g. Tor)

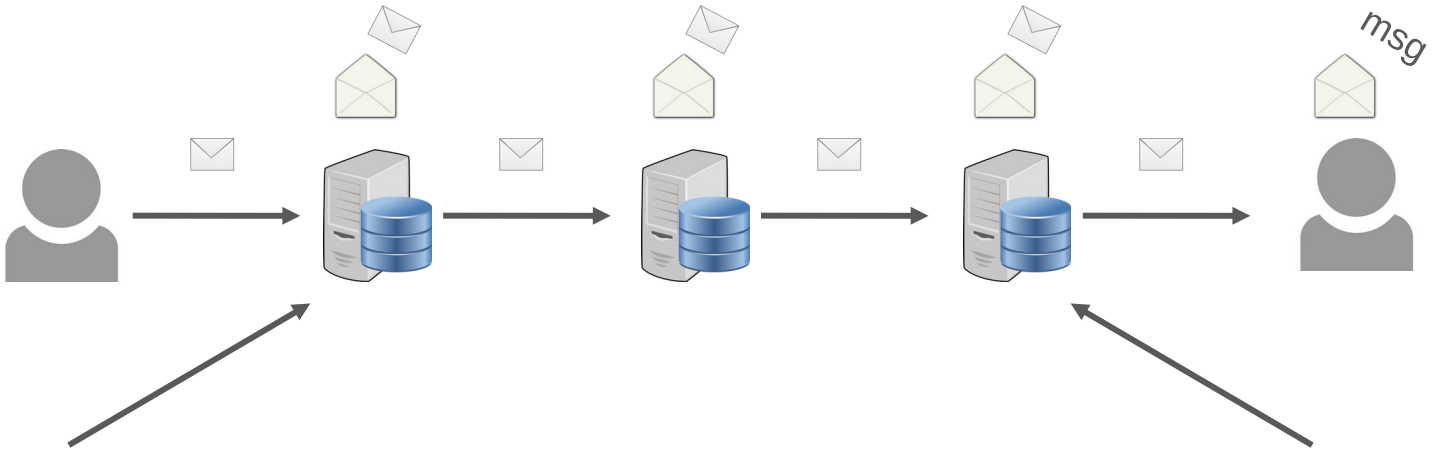


S_1 only sees sender

Onion Encryption

Used in:

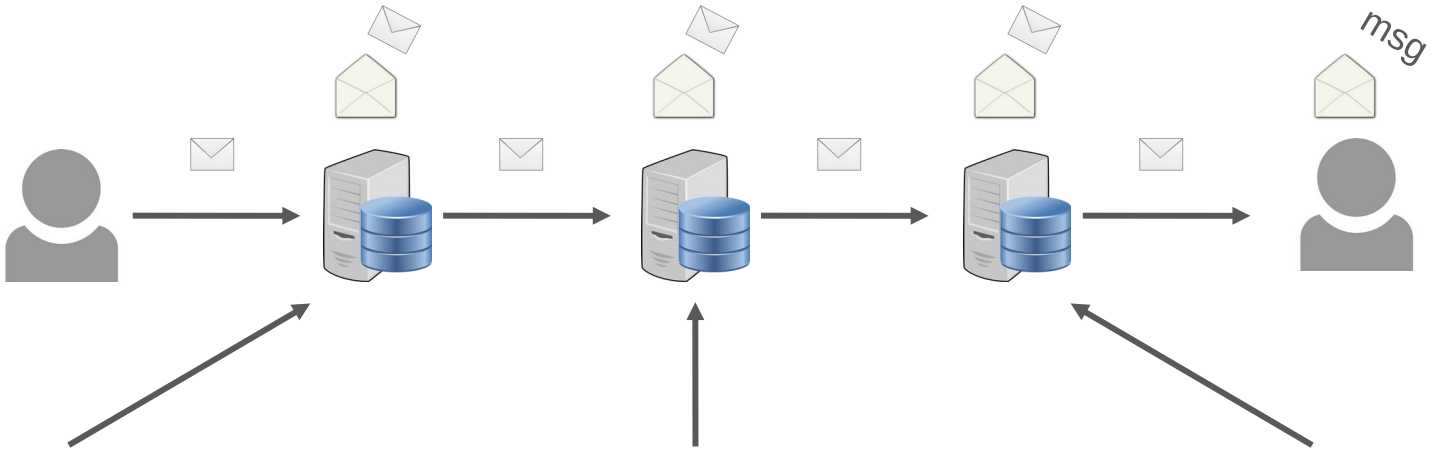
- Mixnets
- Onion routing (e.g. Tor)



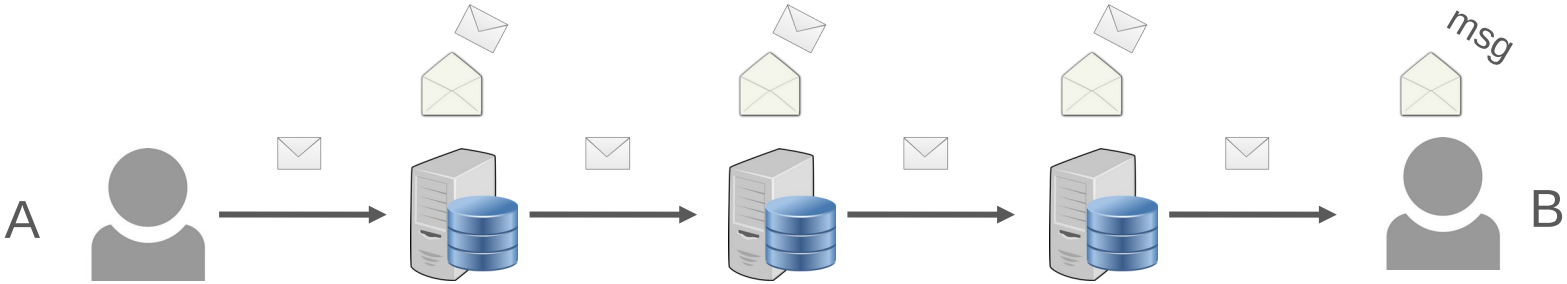
Onion Encryption

Used in:

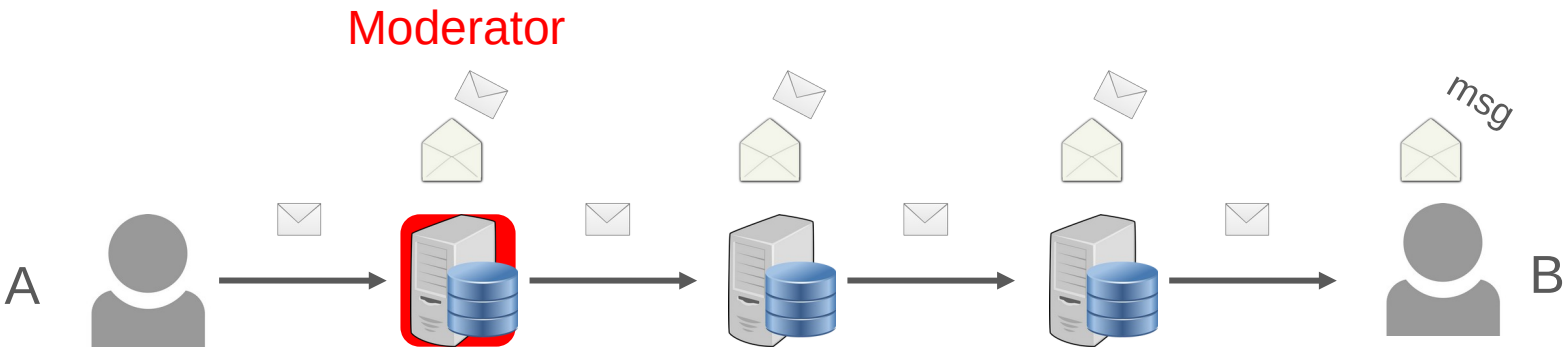
- Mixnets
- Onion routing (e.g. Tor)



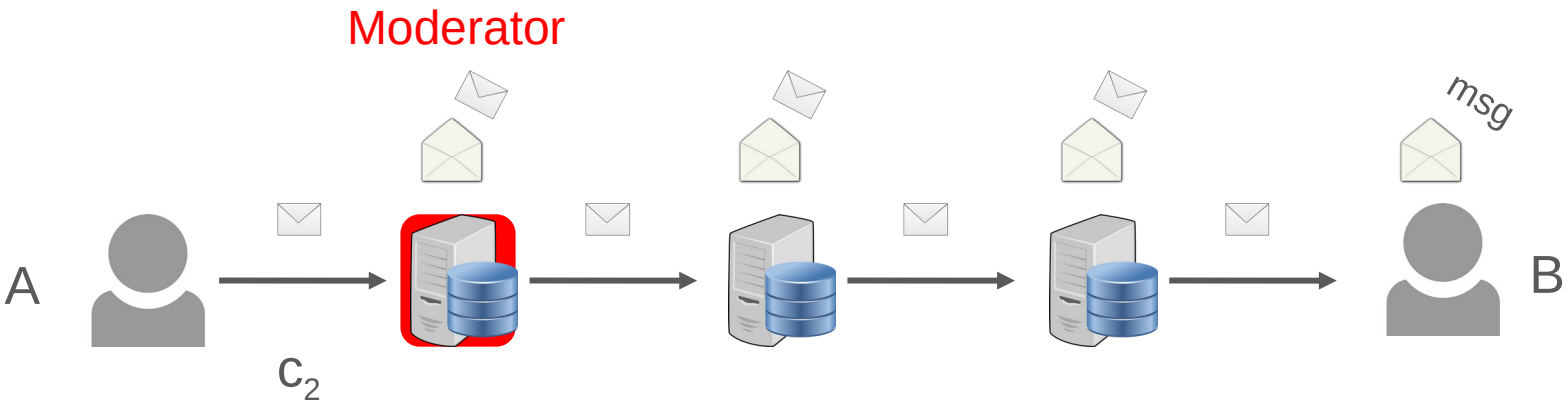
Onion Franking (first try)



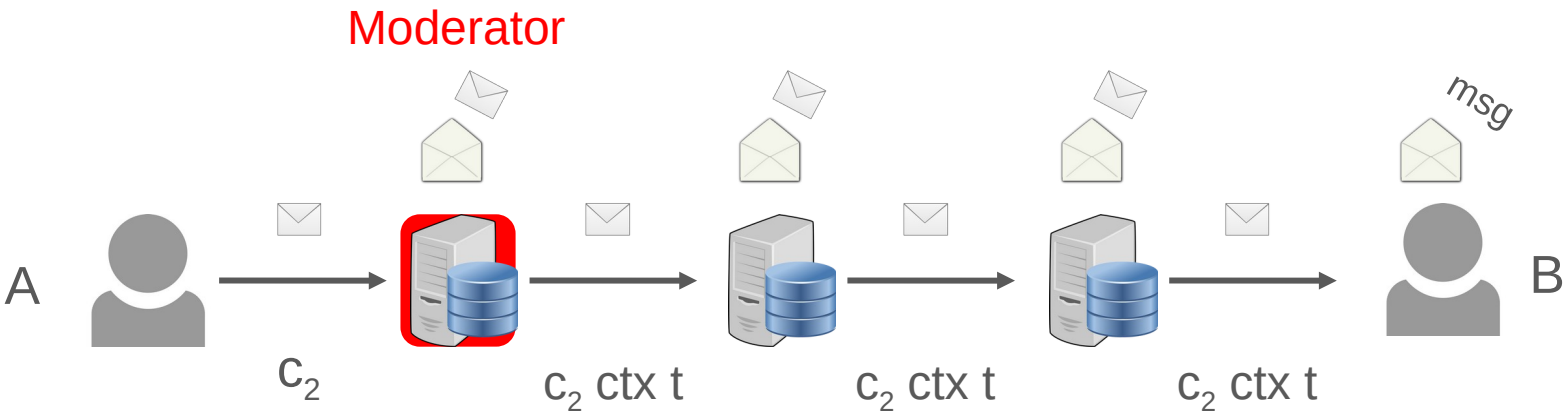
Onion Franking (first try)



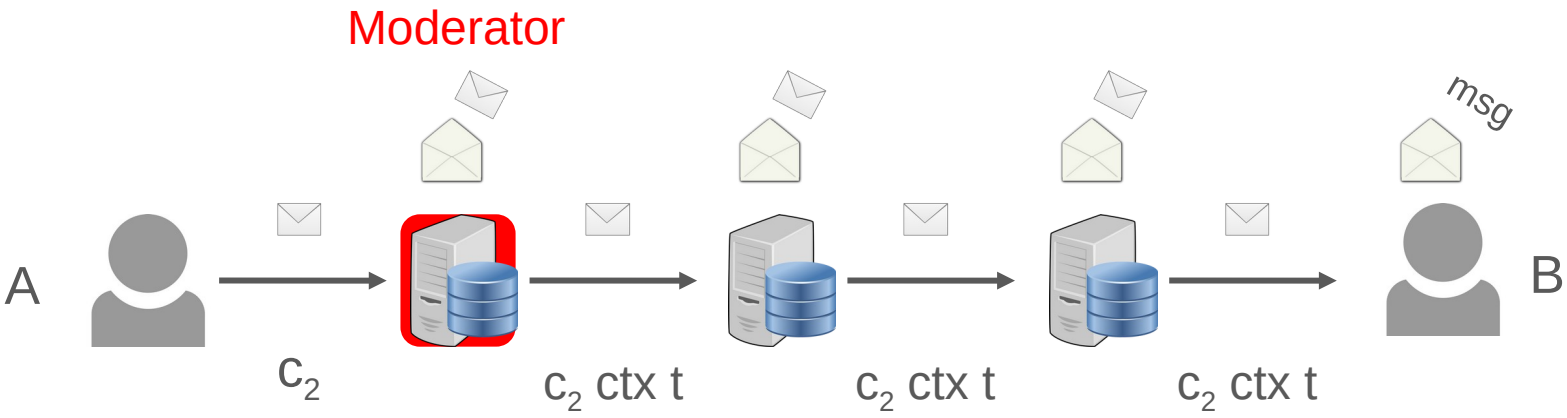
Onion Franking (first try)



Onion Franking (first try)

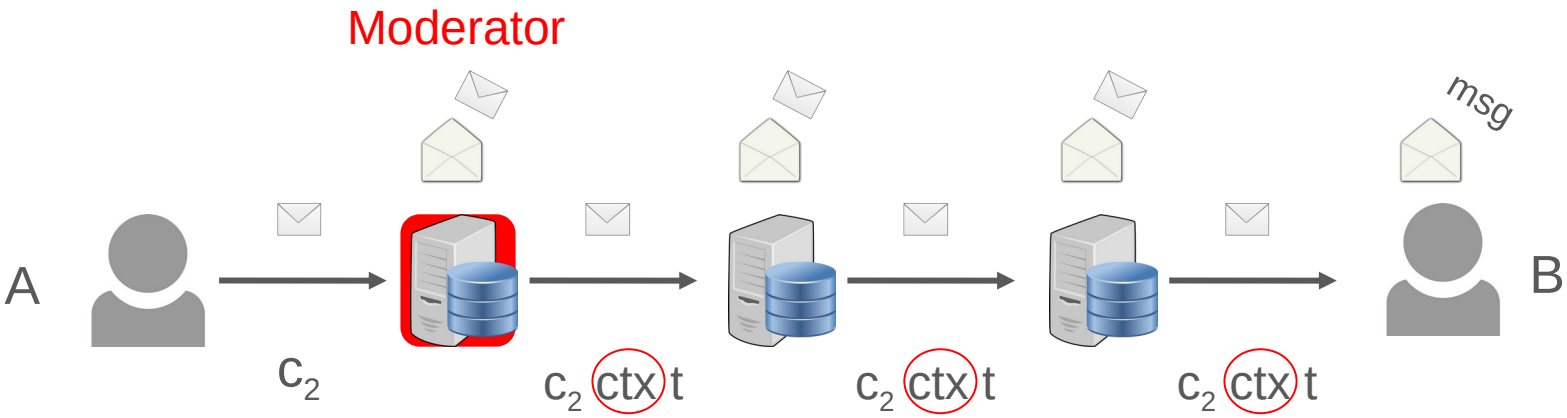


Onion Franking (first try)



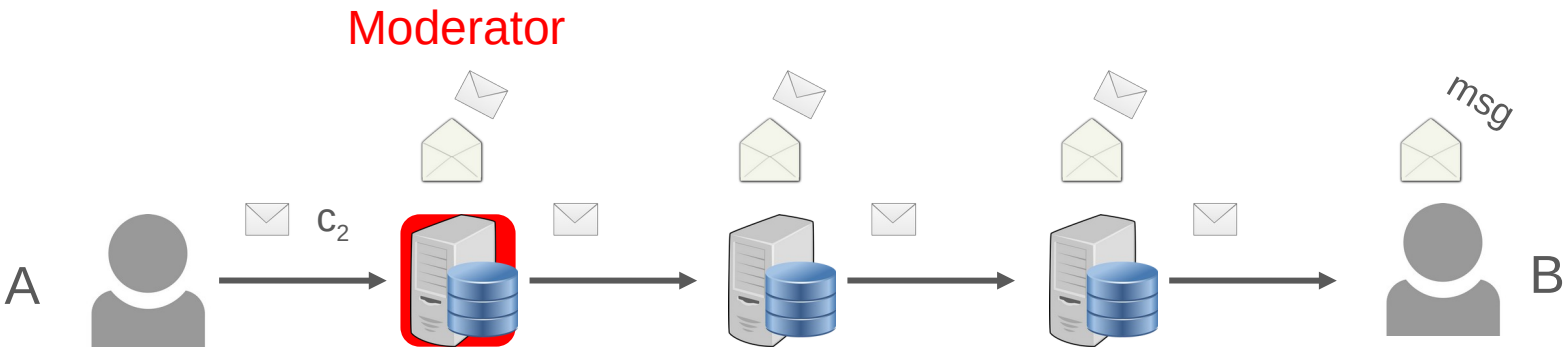
- We’ve completely broken metadata hiding.

Onion Franking (first try)

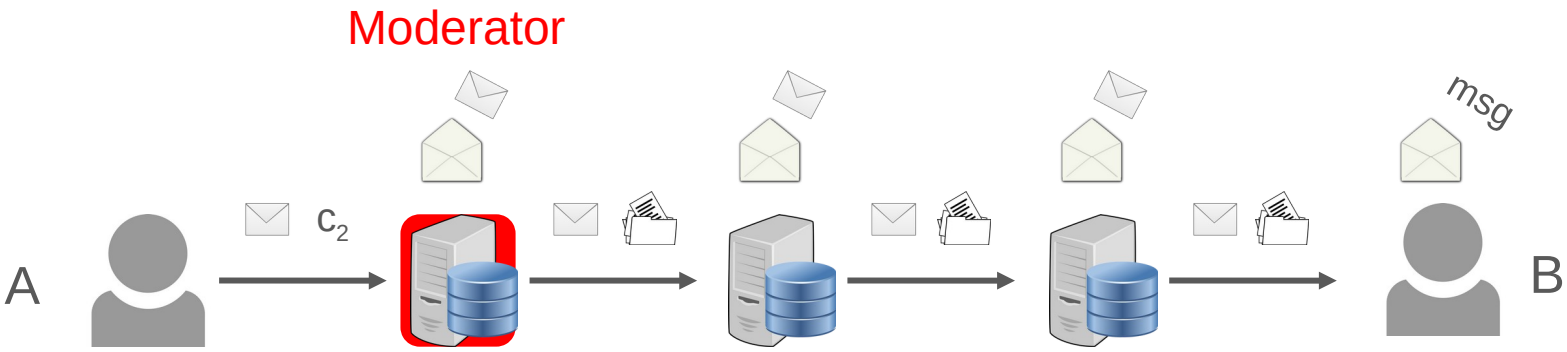


- We’ve completely broken metadata hiding.

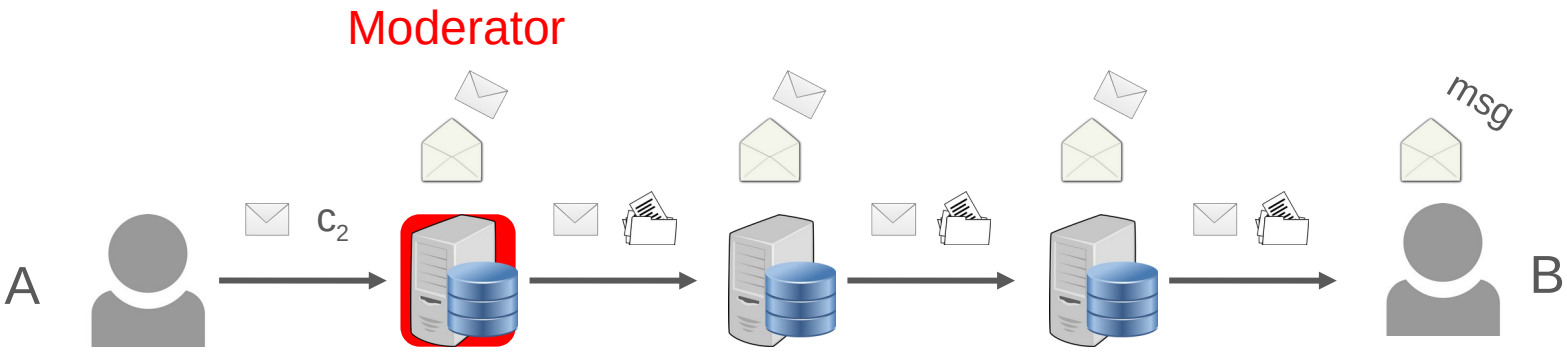
Onion Franking



Onion Franking

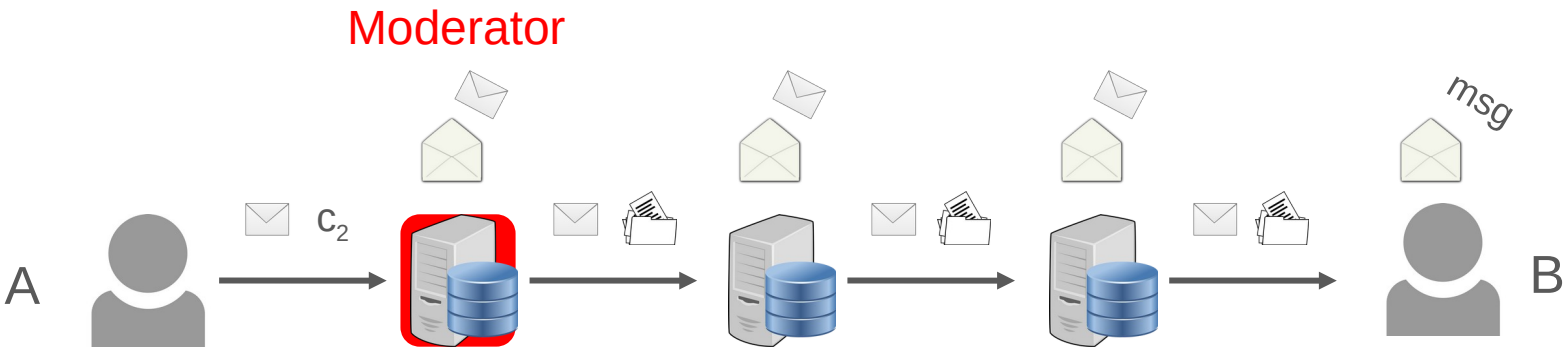


Onion Franking



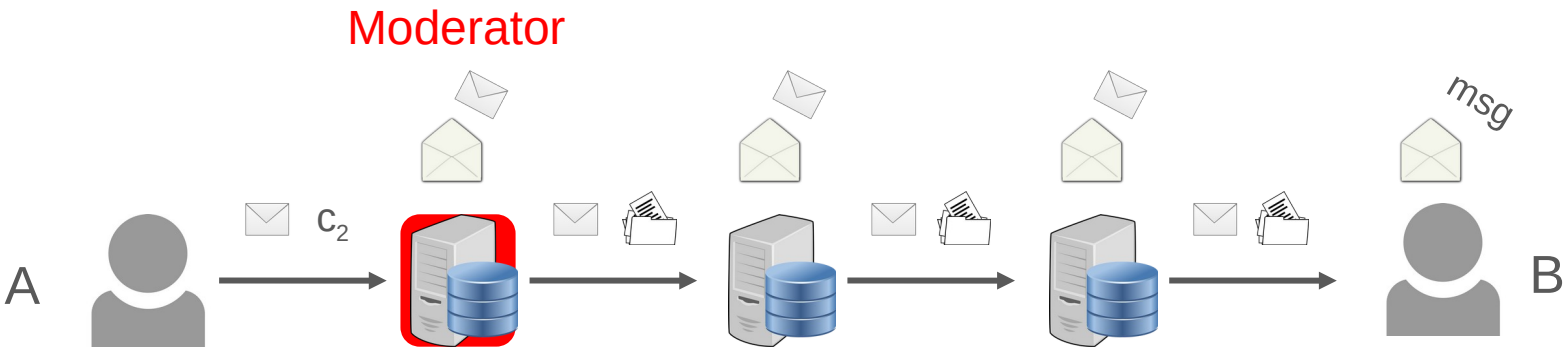
- Our scheme masks data in transit

Onion Franking



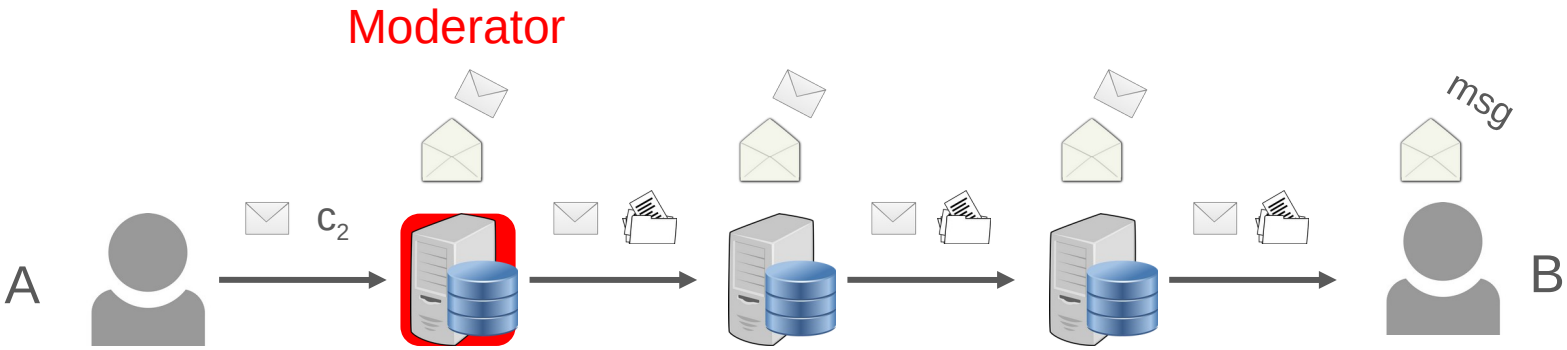
- Our scheme masks data in transit
- The receiver can still recover the franking data

Onion Franking

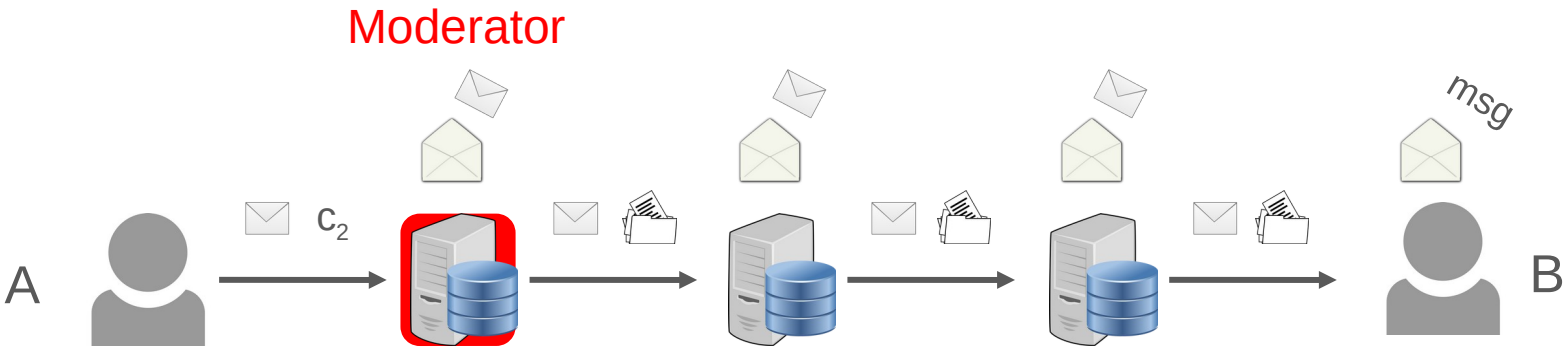


- Our scheme masks data in transit
- The receiver can still recover the franking data (and verify that it hasn't been tampered with)

Onion Franking



Onion Franking



By passing extra data through the mixnet
(carefully masking and unmasking)
We can enable abuse reporting without breaking metadata hiding!

Results

	Send	ModProcess	Process	Read	Moderate
AMF	233.1µs	N/A	N/A	225.1µs	225.2µs
Hecate	16.2µs	N/A	15.4µs	100.1µs	101.8µs
Onion Franking (optimized)					
E2EE Franking	0.9µs	0.2µs	N/A	0.8µs	0.4µs

Issa, Rawane, Nicolas Alhaddad, and Mayank Varia. "Hecate: Abuse reporting in secure messengers with sealed sender." *31st USENIX Security Symposium (USENIX Security 22)*. 2022.
Tyagi, Nirvan, et al. "Asymmetric message franking: Content moderation for metadata-private end-to-end encryption." *CRYPTO 2019*.

Results

	Send	ModProcess	Process	Read	Moderate
AMF	233.1µs	N/A	N/A	225.1µs	225.2µs
Hecate	16.2µs	N/A	15.4µs	100.1µs	101.8µs
Onion Franking (optimized)	**1.6µs**	**0.6µs**	**1.6µs**	**1.5µs**	**0.5µs**
E2EE Franking	0.9µs	0.2µs	N/A	0.8µs	0.4µs

Results

	Send	ModProcess	Process	Read	Moderate
AMF	233.1µs	N/A	N/A	225.1µs	225.2µs
Hecate	16.2µs	N/A	15.4µs	100.1µs	101.8µs
Onion Franking (optimized)	**1.6µs**	**0.6µs**	**1.6µs**	**1.5µs**	**0.5µs**
E2EE Franking	0.9µs	0.2µs	N/A	0.8µs	0.4µs

We achieve performance **on par with E2EE Franking**, while still allowing metadata hiding!

Stronger Accountability

Stronger Accountability

This scheme achieves **accountability**

Stronger Accountability

This scheme achieves **accountability**

A malicious sender can’t send a message where

Stronger Accountability

This scheme achieves **accountability**

A malicious sender can’t send a message where

1. The receiver accepts, but

Stronger Accountability

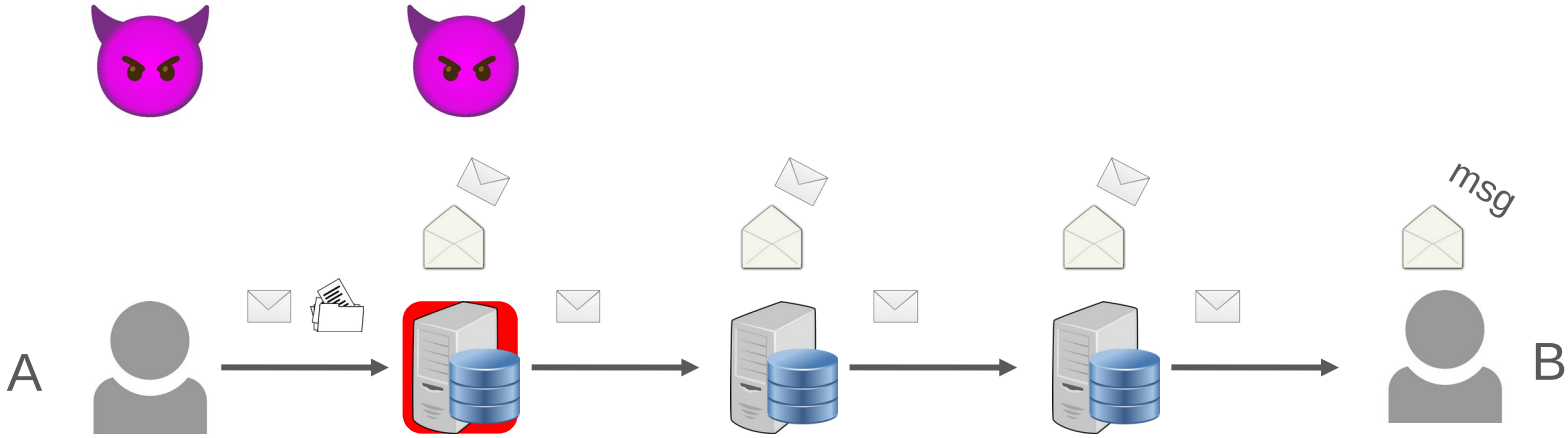
This scheme achieves **accountability**

A malicious sender can’t send a message where

1. The receiver accepts, but
2. The moderator rejects the report

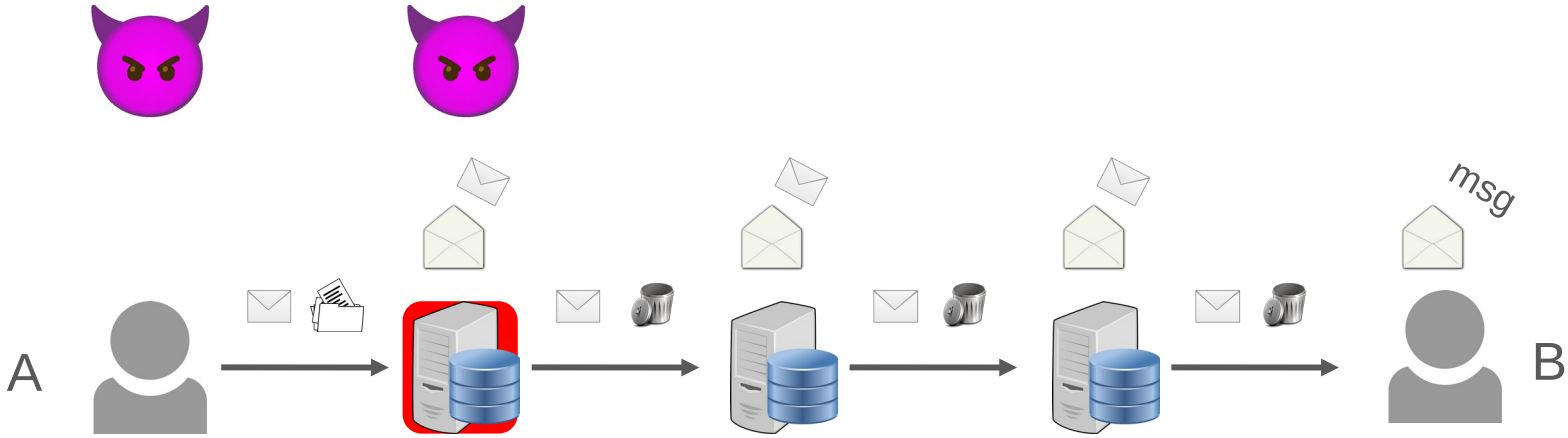
Stronger Accountability

What if the moderator colludes with a user to render their messages unreportable?

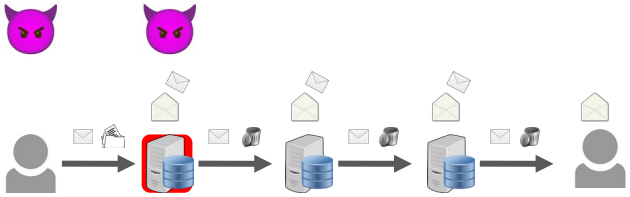


Stronger Accountability

What if the moderator colludes with a user to render their messages unreportable?

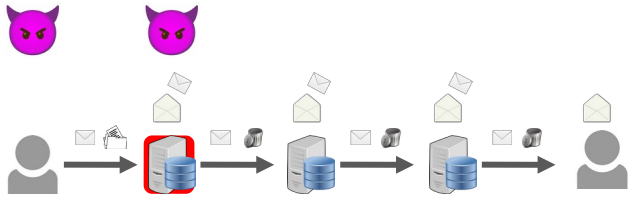


Stronger Accountability



We can enforce honest moderation at message sending time!

Stronger Accountability

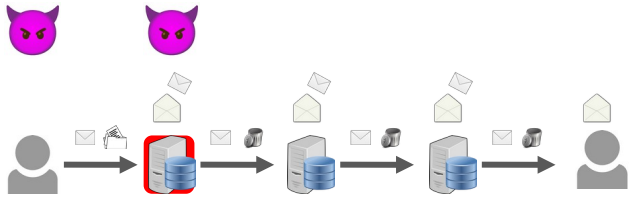


We can enforce honest moderation at message sending time!

Enforce honest moderator behavior with zero-knowledge proofs

(And do this efficiently)

Stronger Accountability



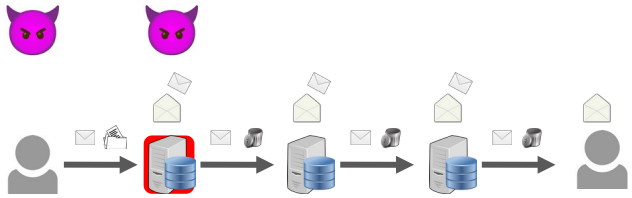
We can enforce honest moderation at message sending time!

Enforce honest moderator behavior with zero-knowledge proofs

(And do this efficiently)

Achieve probabilistic guarantees by sending “trap messages” which are meant to be reported

Stronger Accountability



We can enforce honest moderation at message sending time!

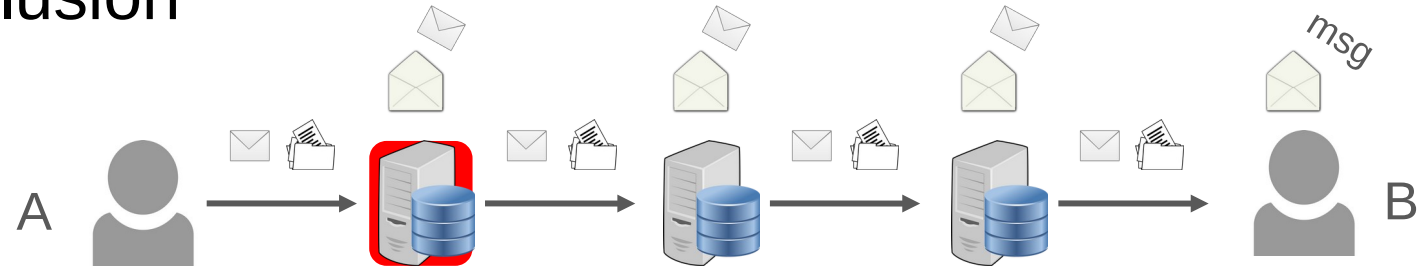
Enforce honest moderator behavior with zero-knowledge proofs

(And do this efficiently)

Achieve probabilistic guarantees by sending “trap messages” which are meant to be reported

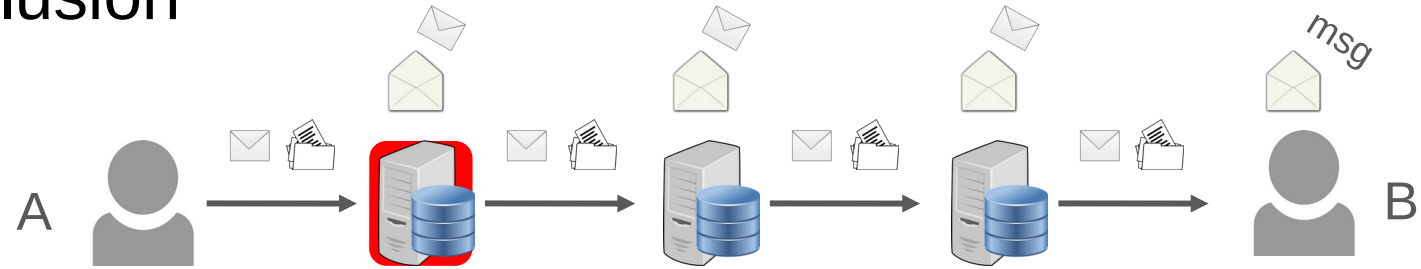
These techniques are applicable to message franking in general!

Conclusion



Code:

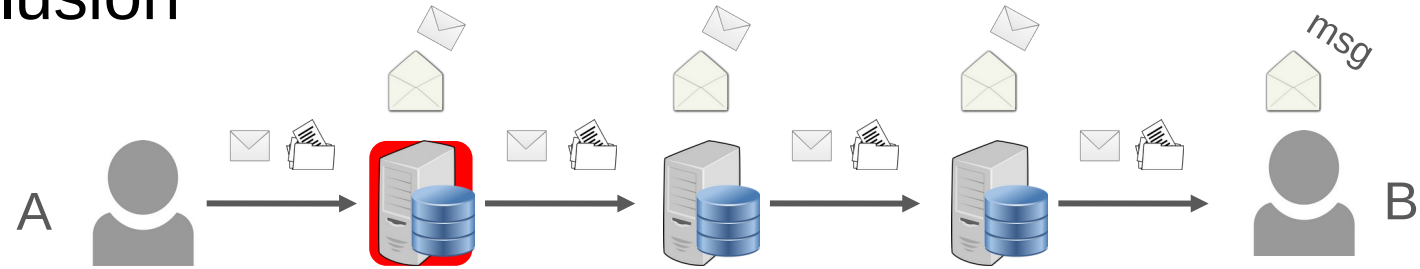
Conclusion



Onion Franking provides abuse reporting with strong performance for metadata-hiding messaging platforms based on onion encryption.

Code:

Conclusion



Onion Franking provides abuse reporting with strong performance for metadata-hiding messaging platforms based on onion encryption.

We also define stronger accountability notions, and achieve them for message franking in general.

Code: