Generic Anonymity Wrapper for Messaging Protocols

Lea Thiemt
FAU Erlangen-Nürnberg

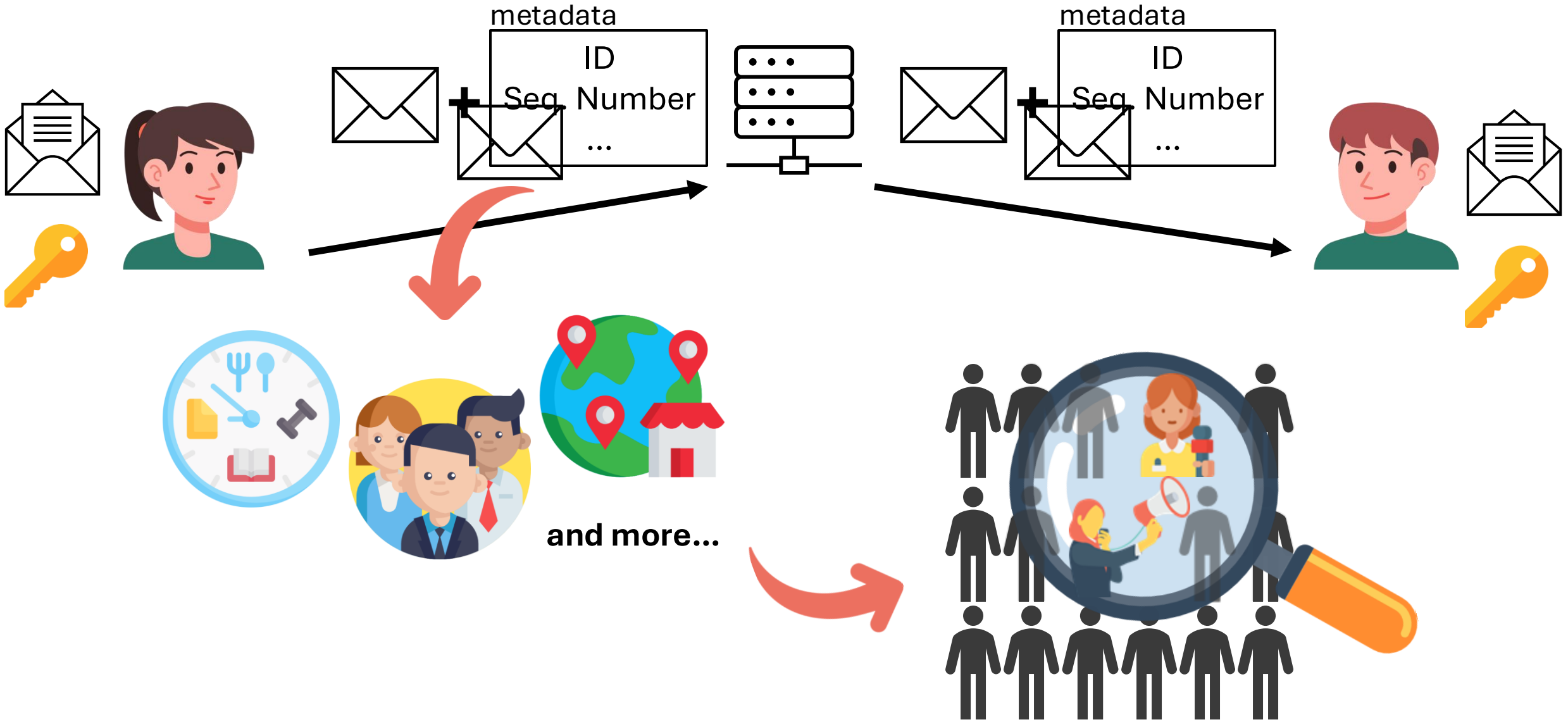
Paul Rösler
FAU Erlangen-Nürnberg

Alexander Bienstock
J.P. Morgan AI Research and J.P. Morgan AlgoCRYPT CoE

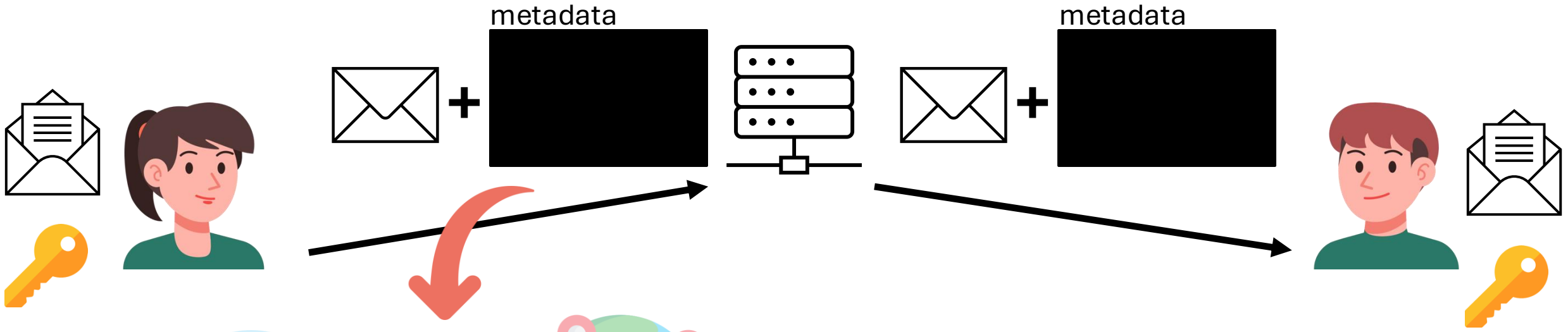
Rolfe Schmidt
Signal Messenger

Yevgeniy Dodis
New York University

Metadata in Messaging Protocols



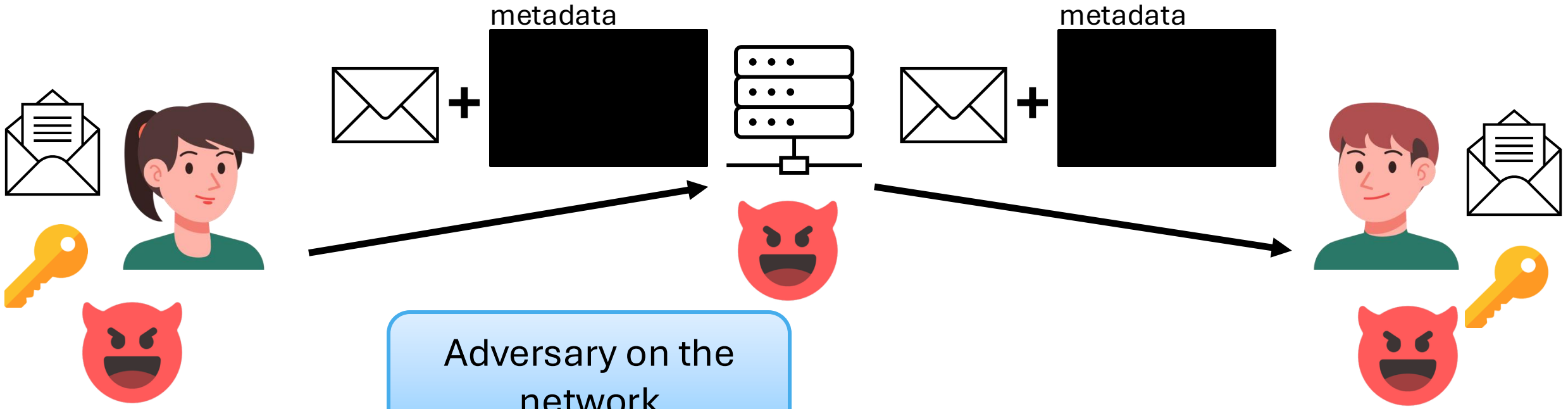
Metadata in Messaging Protocols



and more...

Goal: Generic Wrapper Protocol for Anonymity

Threat Model

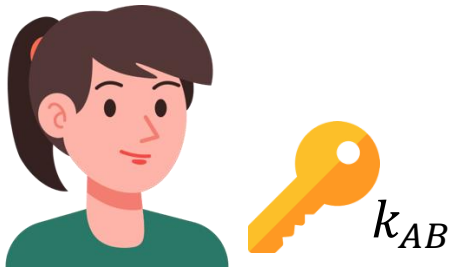


Adversary on the network

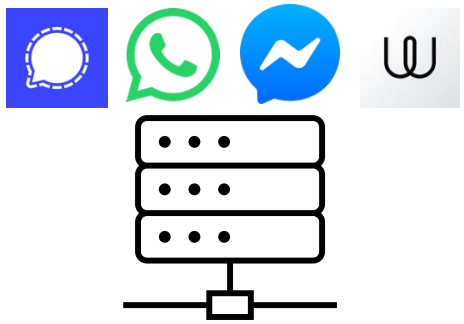
Temporary state exposure

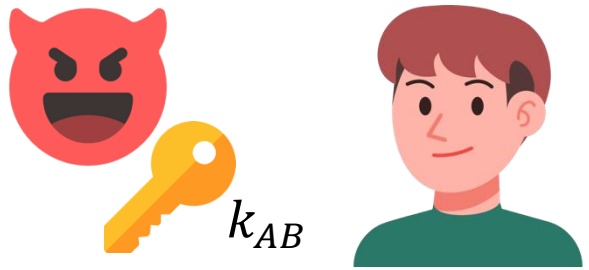
Goal: Generic Wrapper Protocol for Anonymity

Secure communication



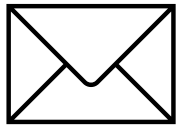
k_{AB}



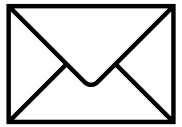


k_{AB}

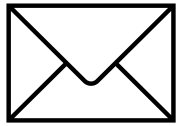
$c_1 = aenc_{k_{AB}}(m_1)$



c_1



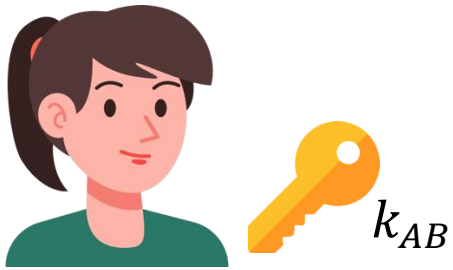
c_2



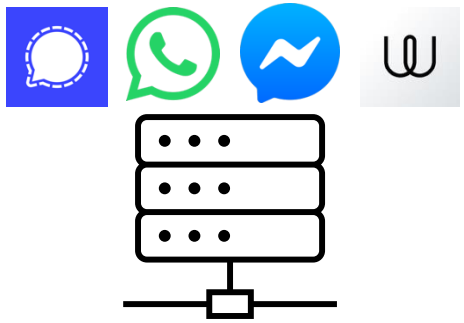
c_3

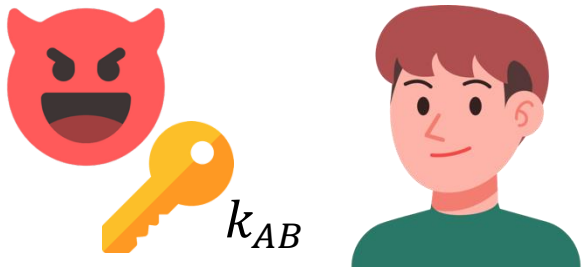
$m_1 = adec_{k_{AB}}(c_1)$

Secure communication



k_{AB}

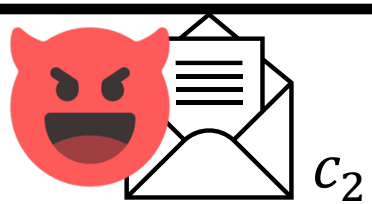




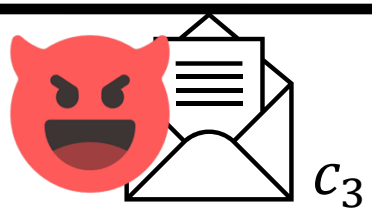
k_{AB}

$c_1 = aenc_{k_{AB}}(m_1)$

c_1



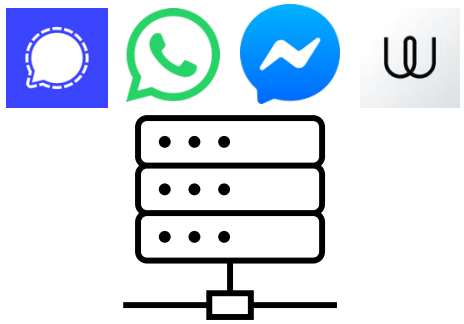
c_2



c_3

$m_1 = adec_{k_{AB}}(c_1)$

Secure communication



ck_0

$ck_1 \quad k_1$

$ck_2 \quad k_2$

$ck_3 \quad k_3$

$\cdots$

$ck_0^* \quad k_0^*$

$c_1 = aenc_{k_1}(m_1)$

$c_2 = aenc_{k_2}(m_2)$

$c_3 = aenc_{k_3}(m_3)$

c_1

c_2

c_3

$m_1 = adec_{k_1}(c_n)$

$m_2 = adec_{k_2}(c_2)$

$m_3 = adec_{k_3}(c_3)$

ck_0

$ck_1 \quad k_1$

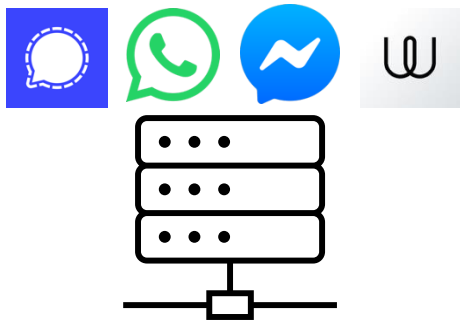
$ck_2 \quad k_2$

$ck_3 \quad k_3$

$\cdots$

$ck_0^* \quad k_0^*$

Secure communication



ck_0

$ck_1 \quad k_1$

$ck_2 \quad k_2$

$ck_3 \quad k_3$

$\cdots$

$ck_0^* \quad k_0^*$

$c_1 = aenc_{k_1}(m_1)$

$c_2 = aenc_{k_2}(m_2)$

$c_3 = aenc_{k_3}(m_3)$

c_3

c_2

$m_1 = adec_{k_1}(c_n)$

$m_2 = adec_{k_2}(c_2)$

$m_3 = adec_{k_3}(c_3)$

ck_0

$ck_1 \quad k_1$

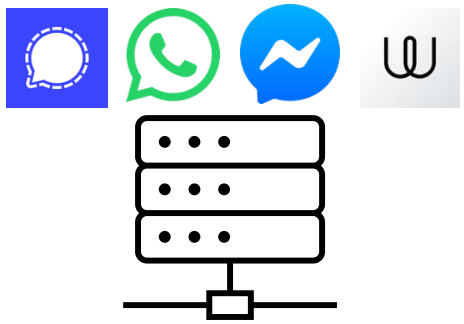
$ck_2 \quad k_2$

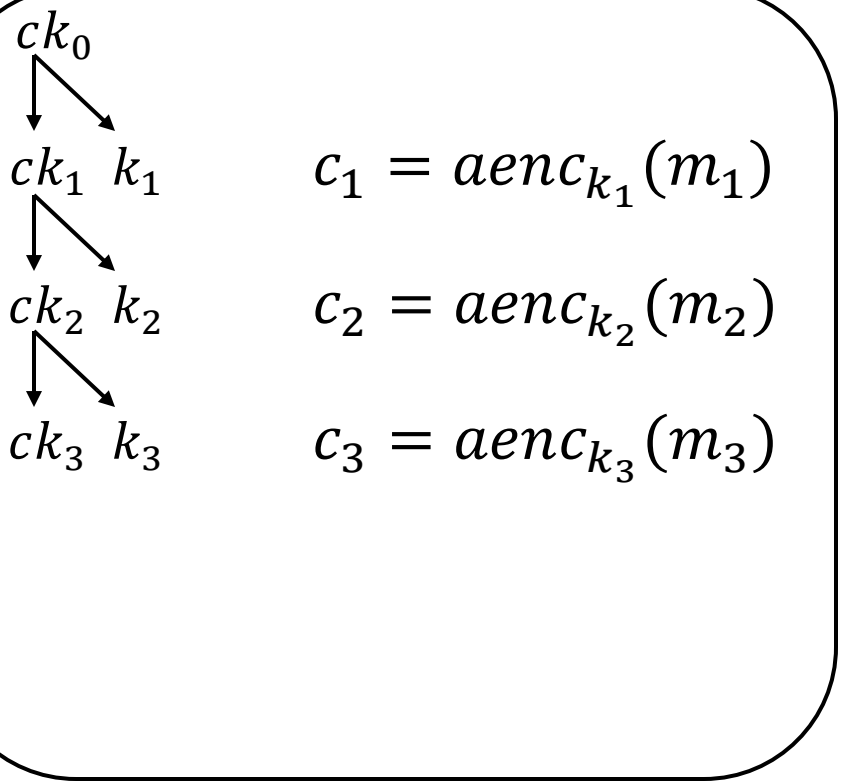
$ck_3 \quad k_3$

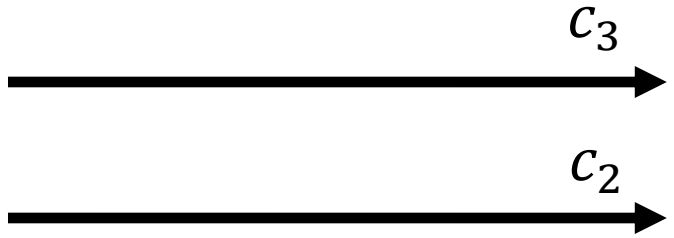
$\cdots$

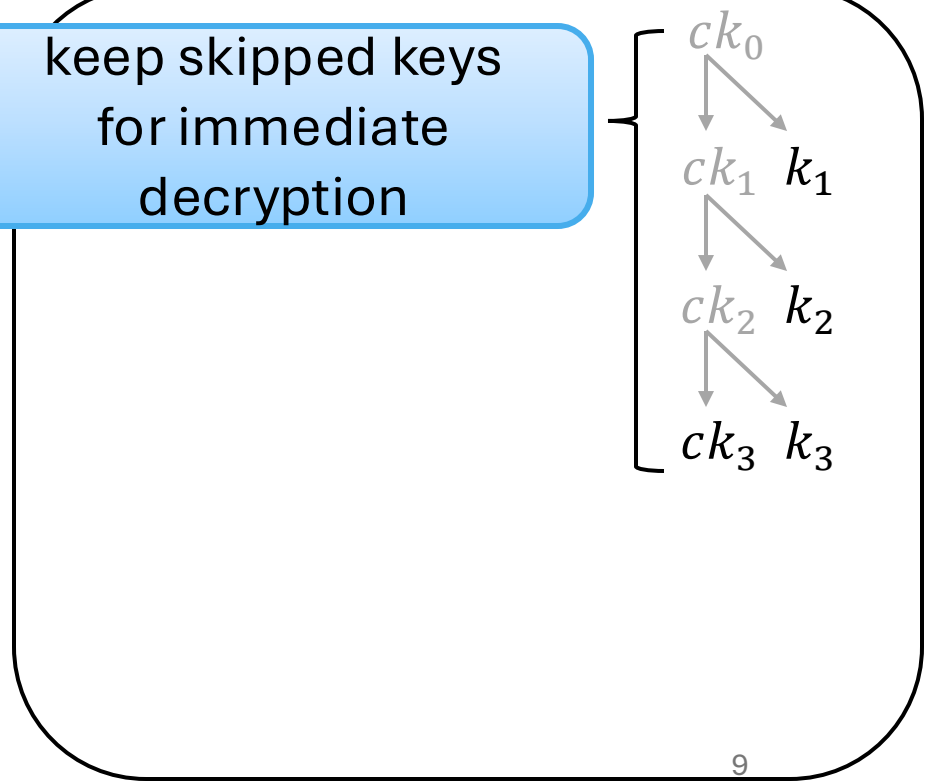
$ck_0^* \quad k_0^*$

Secure communication

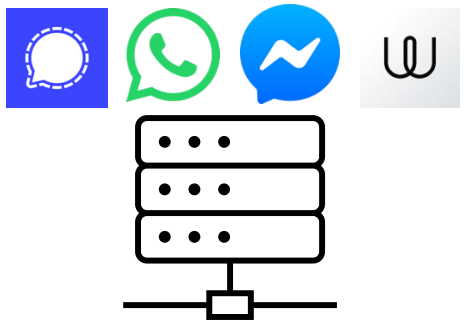


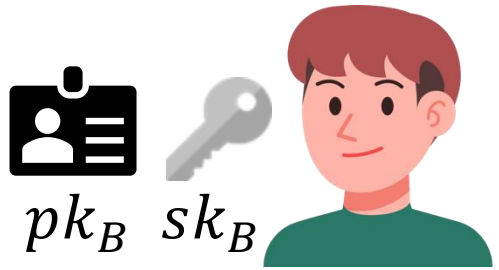






Double Ratchet: Metadata





ck_0

ck_1 k_1

ck_2 k_2

ck_3 k_3

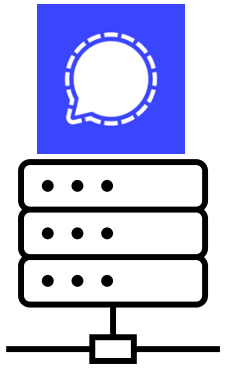
...

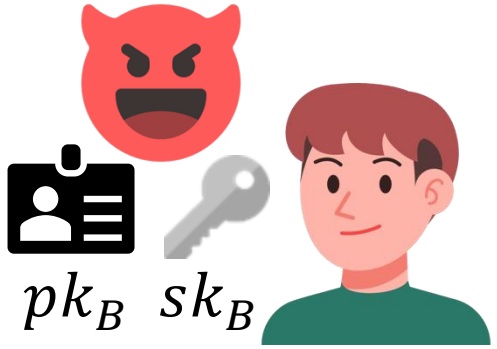
$c_3 = aenc_{k_3}(m_3)$

c_3 + index=3, pk_A

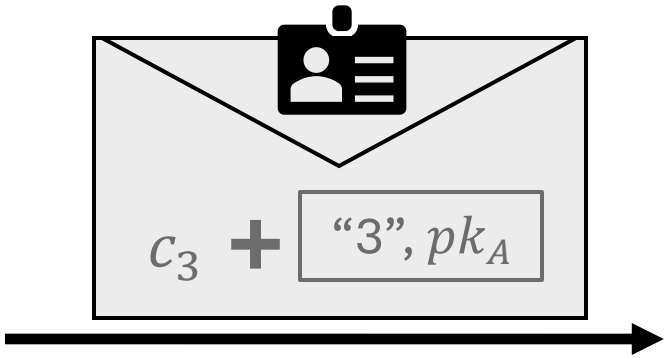
index	key
1	k_1
2	k_2
3	k_3
...	...

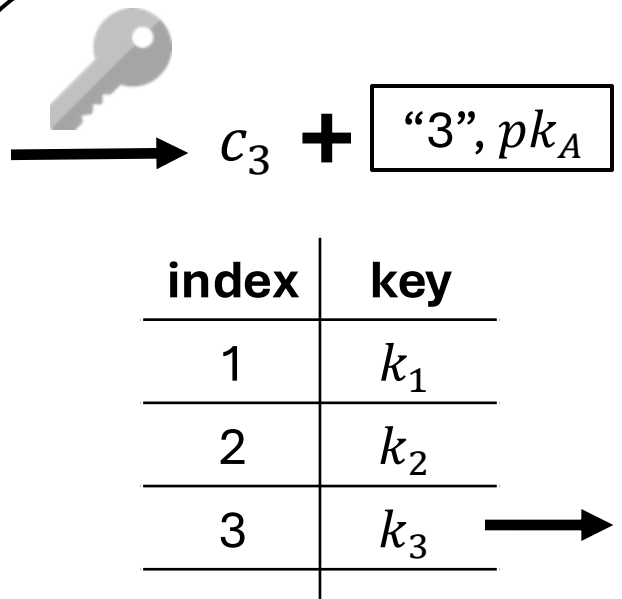
Hiding Metadata: Sealed Sender





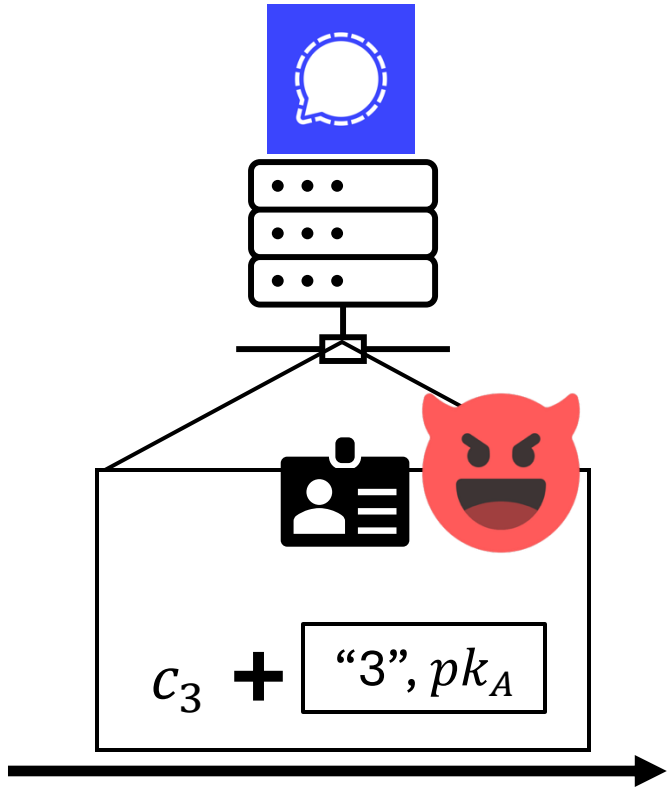
$$c_3 = aenc_{k_3}(m_3)$$



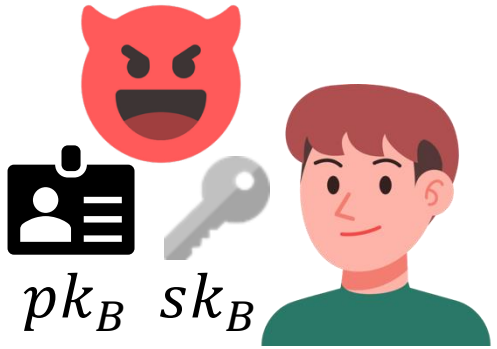


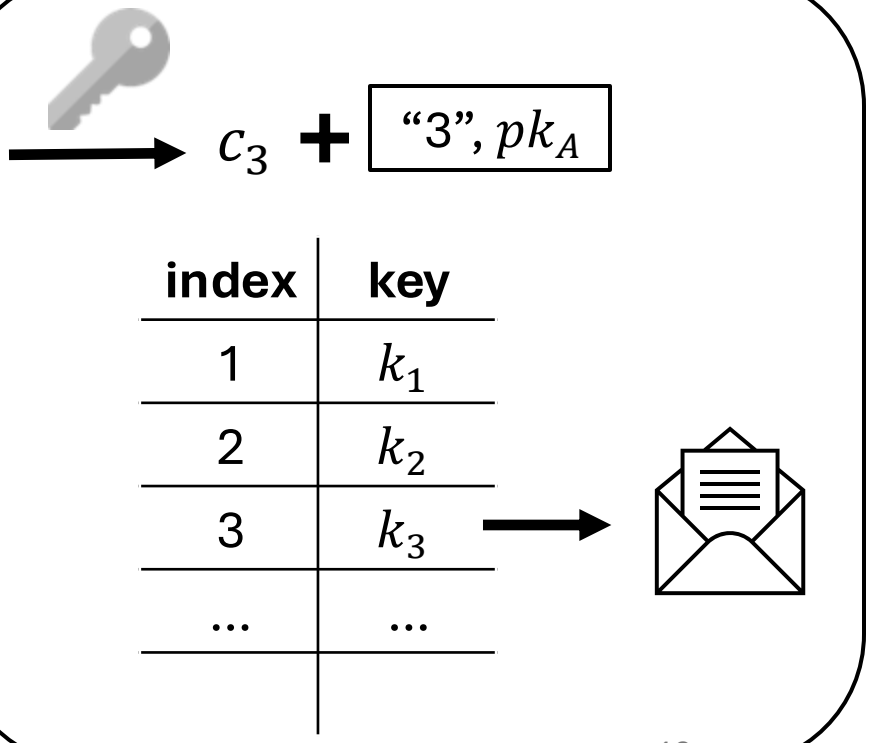
Hiding Metadata: Sealed Sender

$$c_3 = aenc_{k_3}(m_3)$$

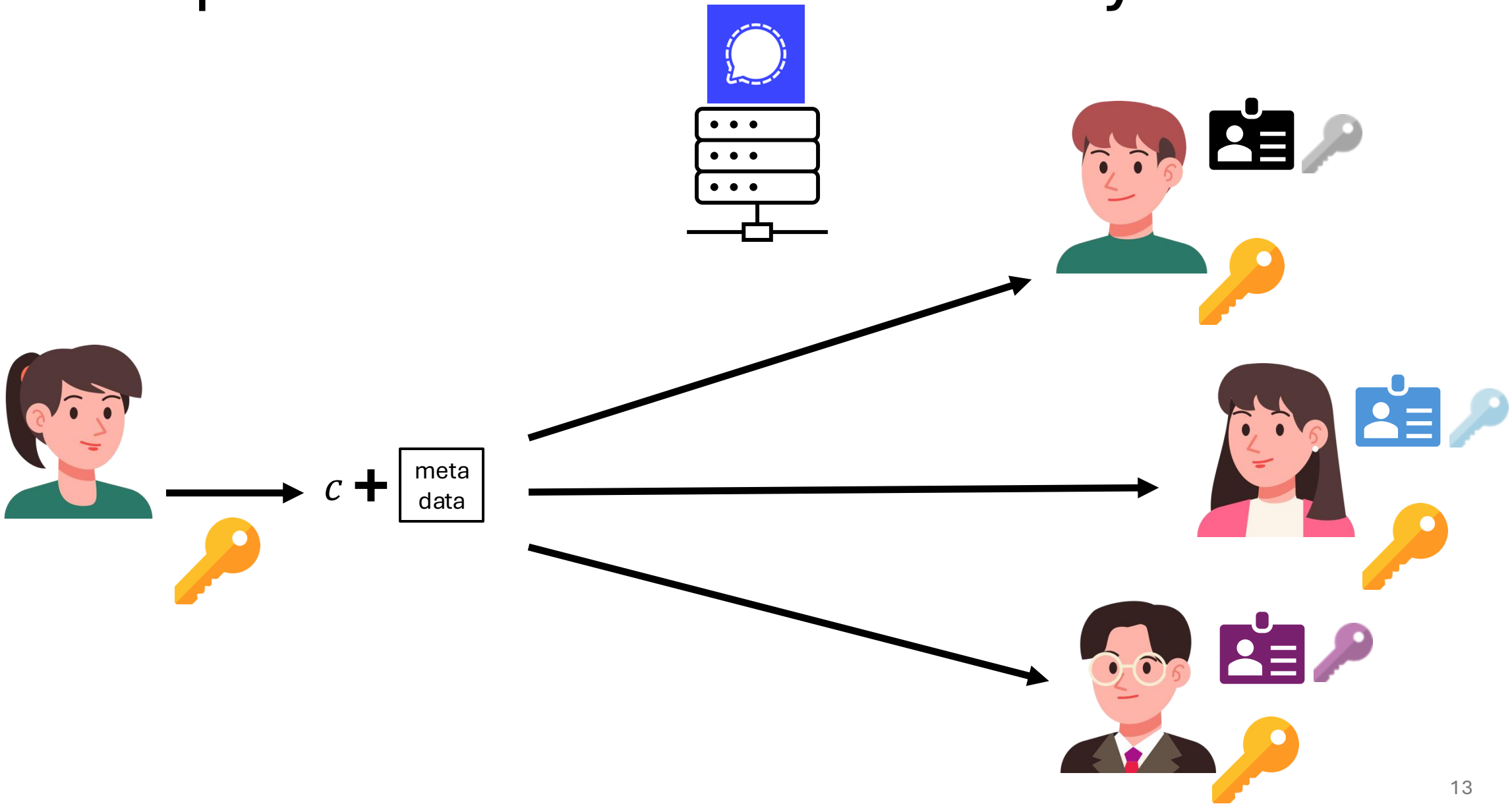


No anonymity guarantees after compromise

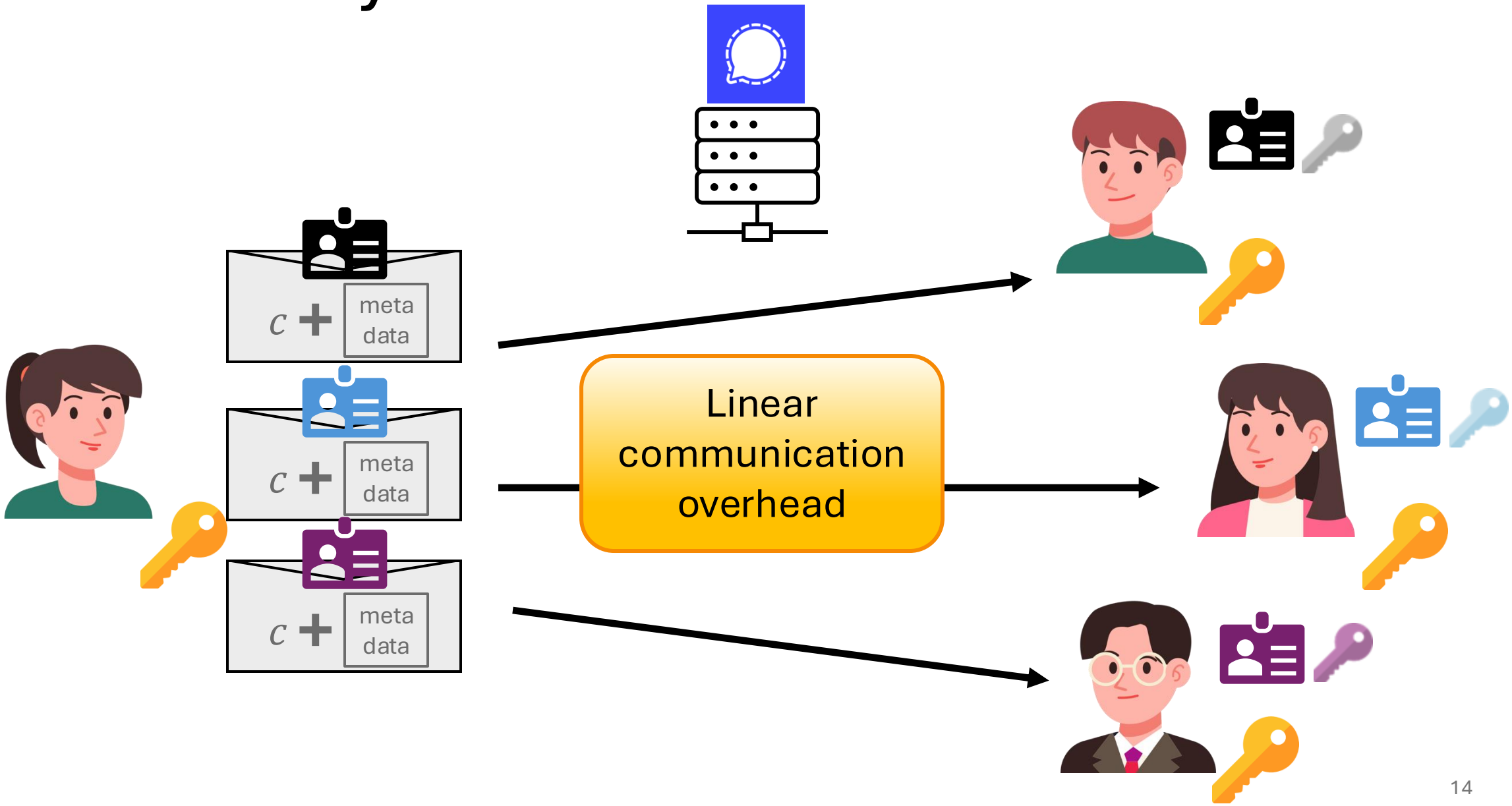




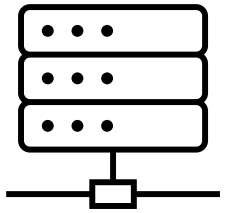
Group Communication: Sender Keys

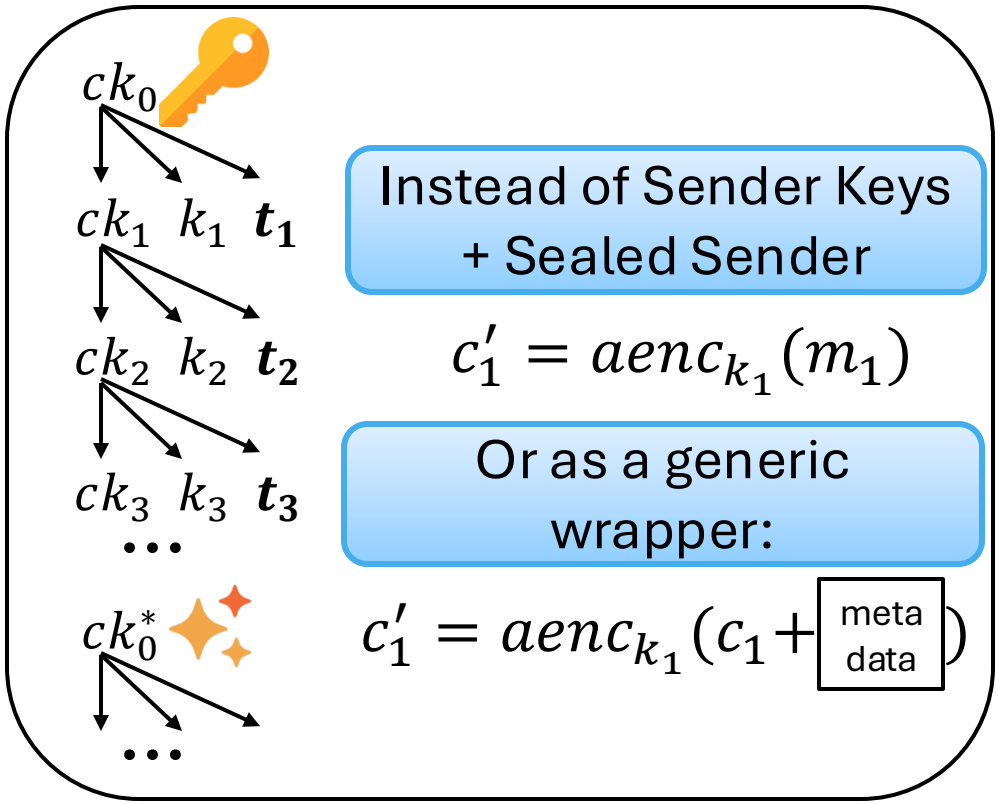


Sender Keys + Sealed Sender



Anonymity Wrapper





Instead of Sender Keys + Sealed Sender

$$c_1' = aenc_{k_1}(m_1)$$

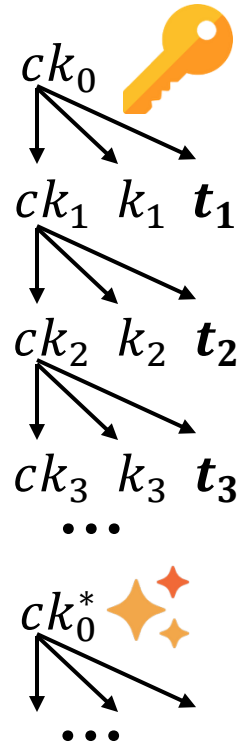
Or as a generic wrapper:

$$c_1' = aenc_{k_1}(c_1 + \boxed{\text{meta data}})$$

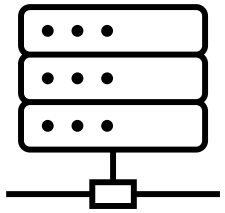
c_1', t_1

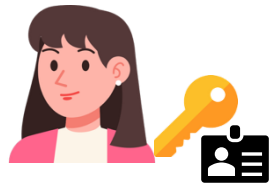
tag	key
t_1	k_1
t_2	k_2
t_3	k_3
…	…

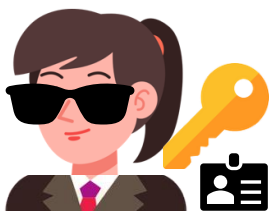
$$m_1 = adec_{k_1}(c_1)$$

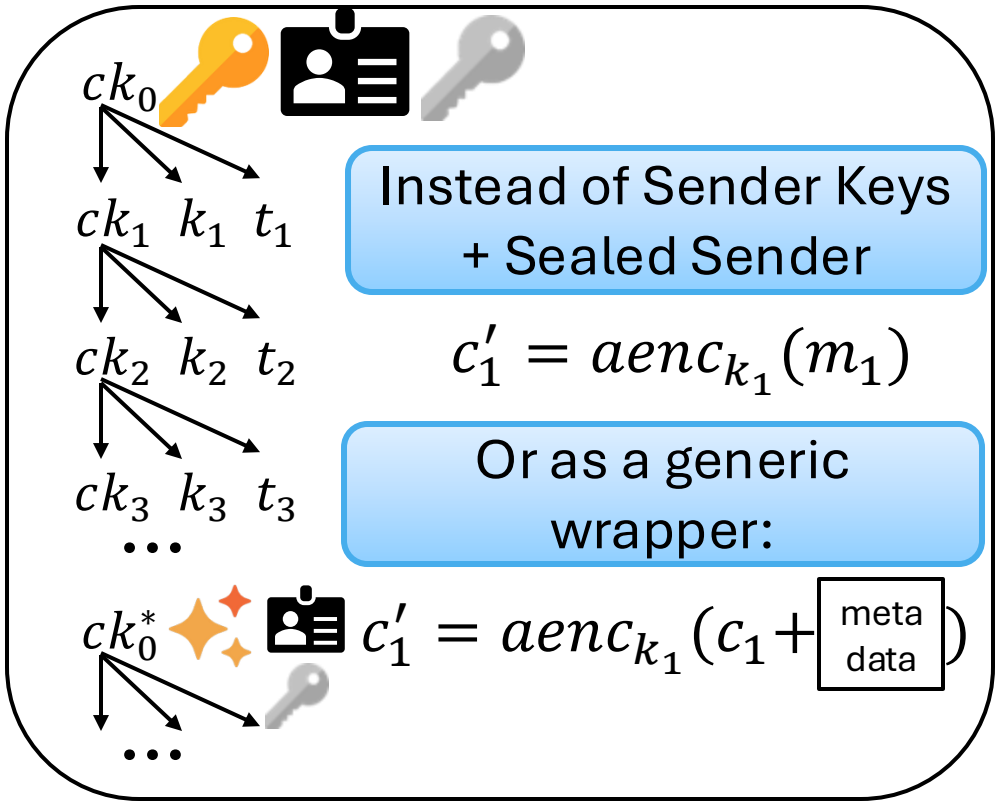


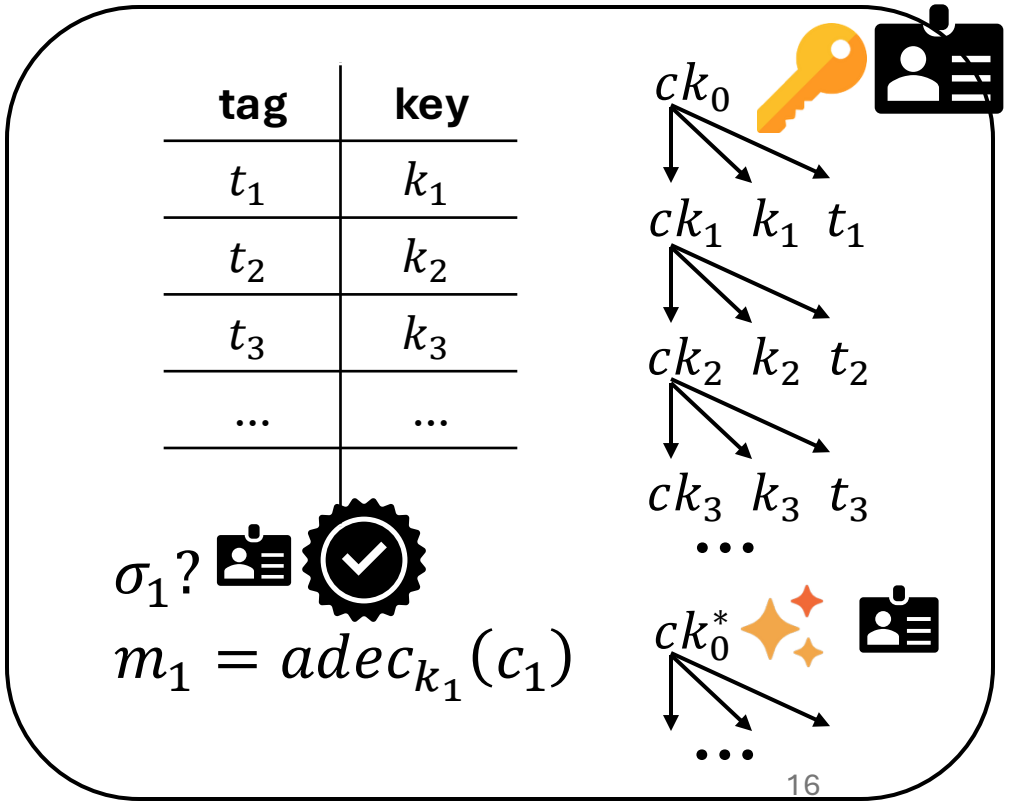
Anonymity Wrapper: Sender Authentication









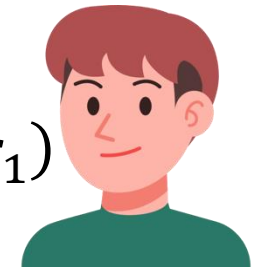


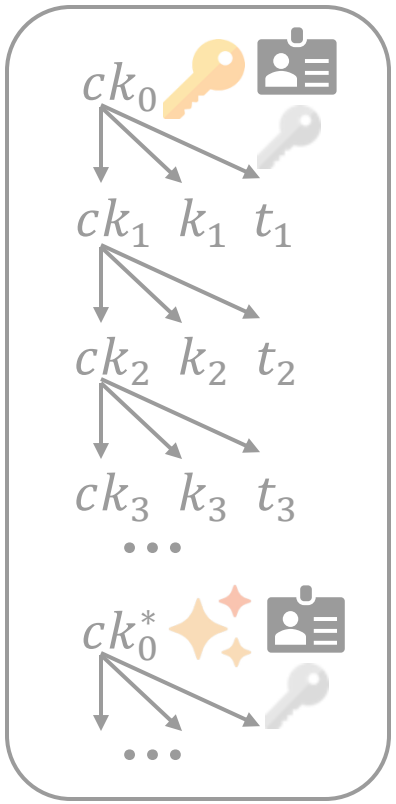
Anonymity Wrapper: Performance

$c_1' = aenc_{k_1}(c_1 + \boxed{\text{meta data}})$

c_1', t_1, σ_1

$m_1 = adec_{k_1}(c_1)$

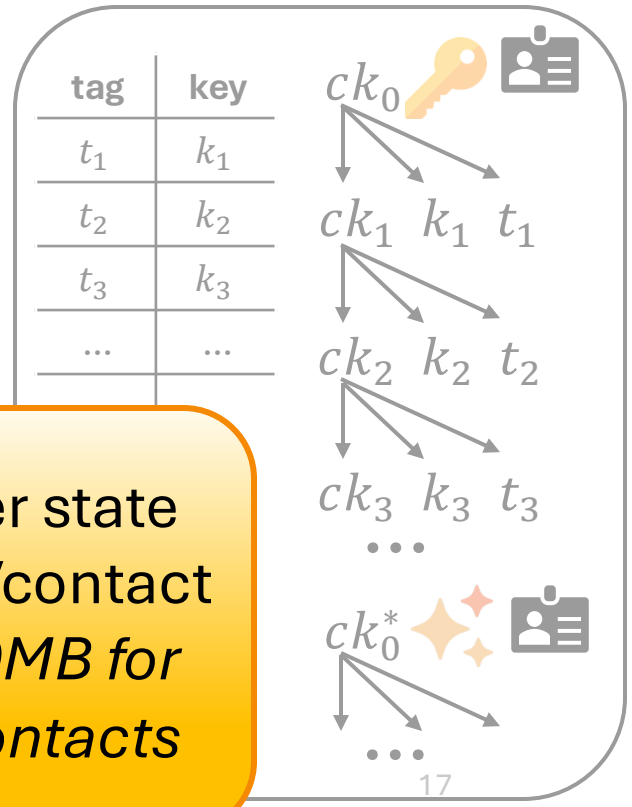




Encryption (Decryption) (µs)					
Group Size	2	5	10	100	1000
AW					
SS+SK					

Sender state
+ 192 bytes

	Ciphertext size (bytes)
AW	
SS+SK	



Receiver state
+ 600KB/contact
i.e., 600MB for 1000 contacts

Receiver State: Out-of-Order Delivery

ck_0

ck_1 k_1 t_1

ck_2 k_2 t_2

ck_0^*

ck_1^* k_1^* t_1^*

c_1, t_1

c_2, t_2

No message index to advance key chain!

tag	key
t_1	k_1
t_2	k_2
t_3	k_3
t_4	k_4
…	…

t_2 not in receiver state
→ Continue key chain?
→ Discard…?
→ Pre-compute keys ☺

ck_0

ck_1 k_1 t_1

ck_2 k_2 t_2

ck_3 k_3 t_3

…

ck_0^*

ck_1^* k_1^* t_1^*

State size: Receiver stores 2000 future pre-computed keys for each session + up to 2000 skipped keys

Receiver State: Pre-computed Keys

ck_0

ck_1 k_1 t_1

ck_2 k_2 t_2

ck_0^*

ck_1^* k_1^* t_1^*

c_1^*

Last tag: t_3

tag	key
t_2	k_2
t_3	k_3
…	…
t_{2003}	t_{2003}

← unnecessary

ck_0

ck_1 k_1 t_1

ck_2 k_2 t_2

ck_3 k_3 t_3

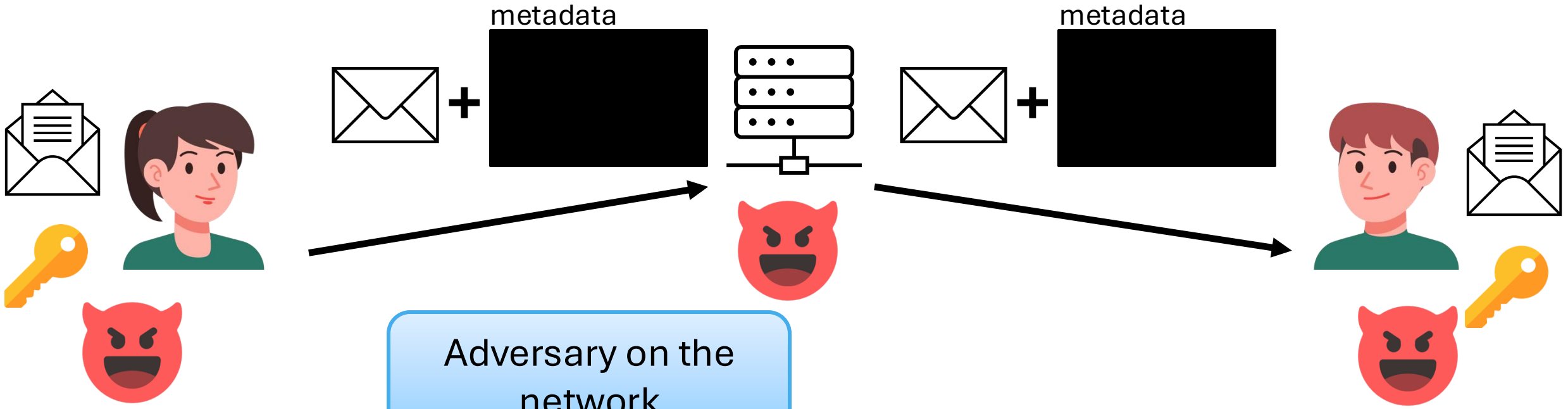
…

ck_0^*

ck_1^* k_1^* t_1^*

…

Threat Model



Goal: Generic Wrapper Protocol for Anonymity

Metadata in Receiver State

- State exposure should not reveal past communication

But this state is too large, how can we improve?

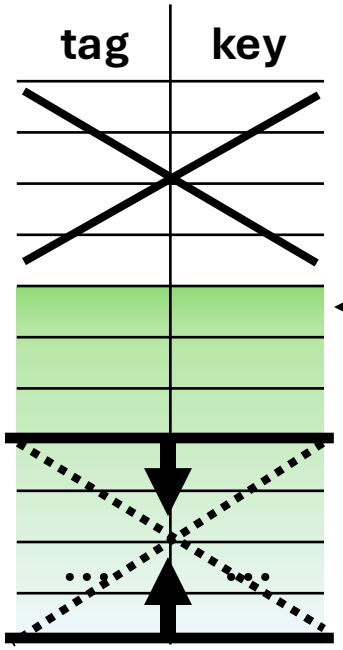
tag	key
...	...

- Only store key derivation order *(no indices)*
- Constant size (may insert dummies)

Improving Size of Receiver State

- Up to 1/3 of receiver state are dummies
- Most pre-computed keys only needed for worst case

Storage size:
38KB/contact
3.8MB/1000 contacts



most likely
to be received

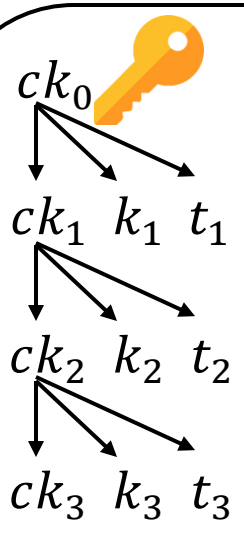
Don‘t store dummies

Store only
“*whether tag is receivable*“
(Counting Bloom Filter)

Conclusion

Encryption (Decryption) (µs)					
Group Size	2	5	10	100	1000
AW	21 (32)				
SS+SK	131 (61)	283 (61)	343 (61)	1,116 (61)	6,052 (61)

	Message size (bytes)
AW	155
SS+SK	440 + 68 * group_size



Instead of Sender Keys + Sealed Sender

$$c_1' = aenc_{k_1}(m_1)$$

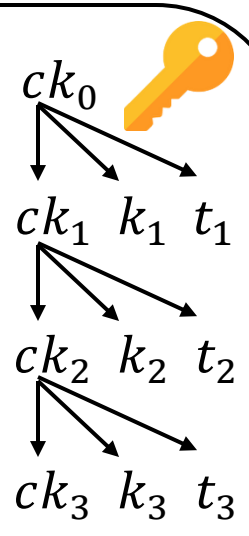
Or as a generic wrapper:

$$c_1' = aenc_{k_1}(c_1 + \boxed{\text{meta data}})$$

c_1', t_1

- Signal considers deploying this protocol

tag	**key**
t_1	k_1
t_2	k_2
t_3	k_3
…	…



$$m_1 = adec_{k_1}(c_1)$$