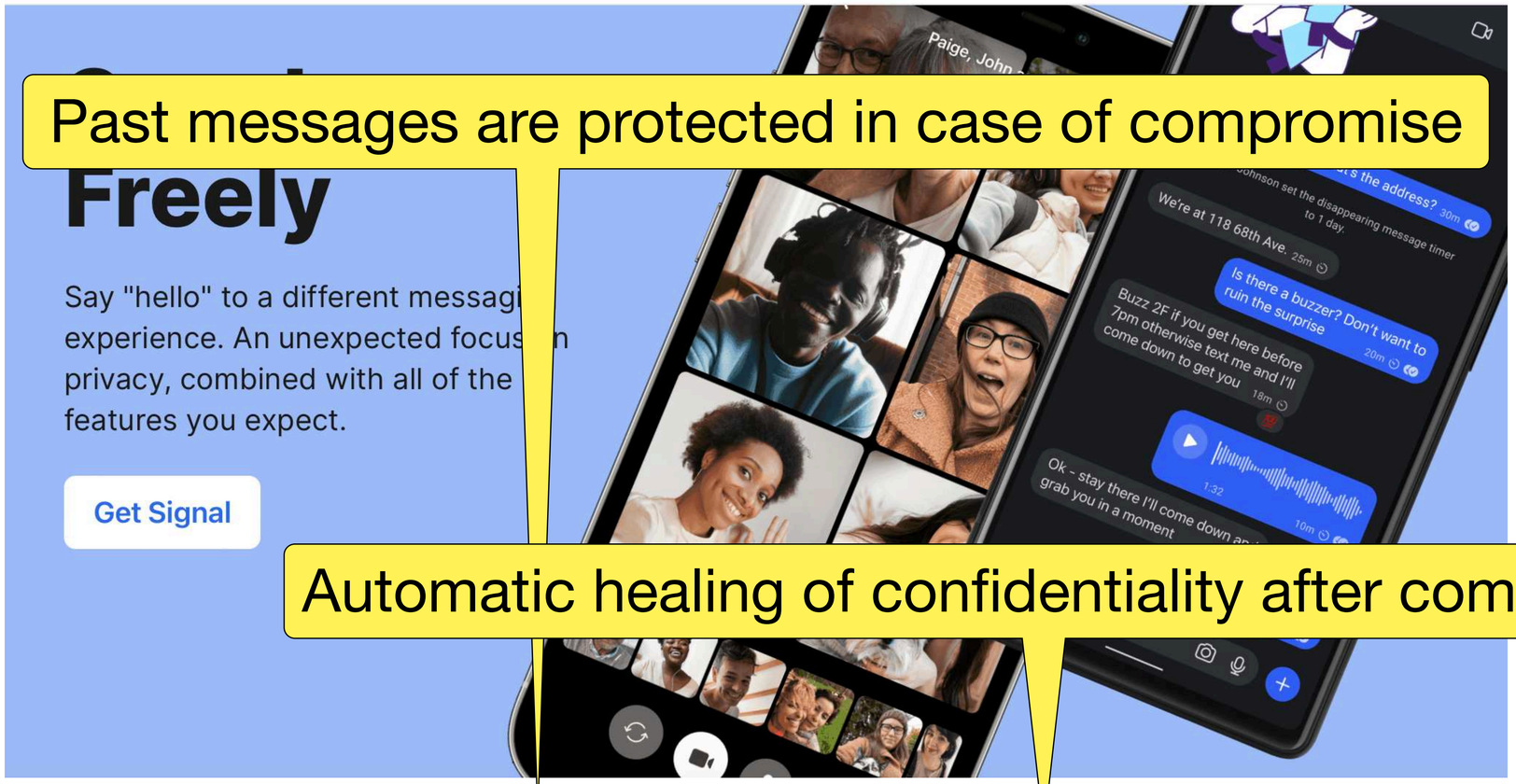
Beyond the ratchet: practical challenges in secure messaging

PhD oral exam of Simone Colombo, 11th of December 2024



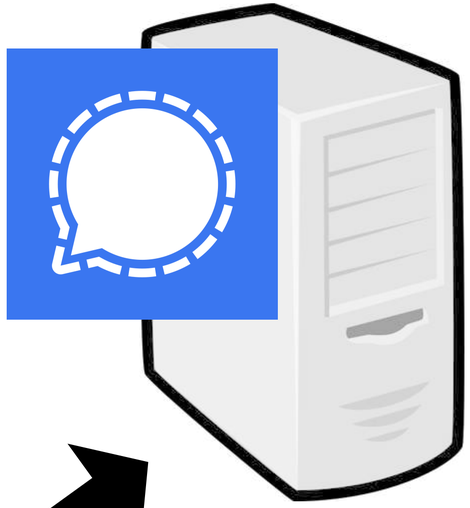
Past messages are protected in case of compromise

Automatic healing of confidentiality after compromise

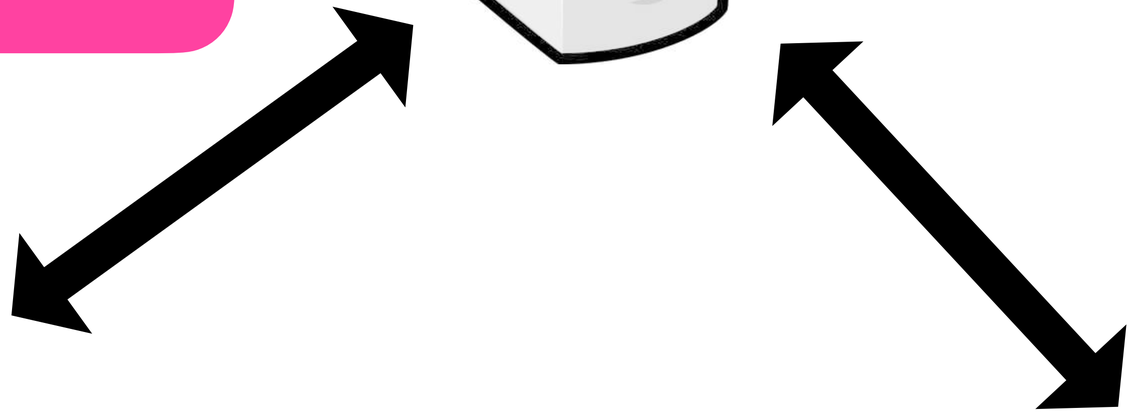
end-to-end encryption, forward secrecy, post-compromise security

Despite these important improvements, several challenges remain

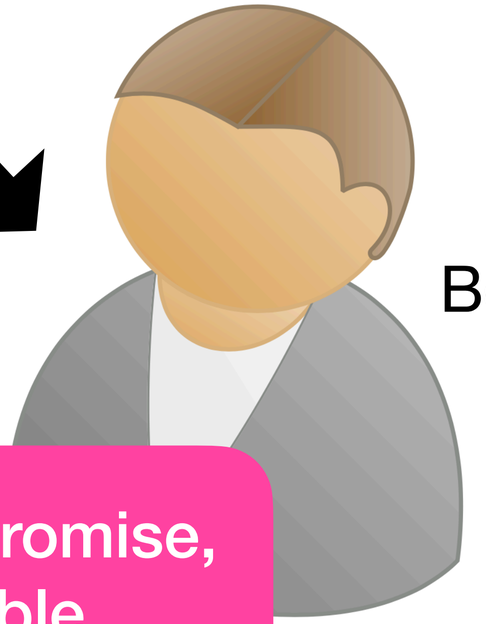
Messages are confidential, but metadata are not



Parties can report messages and no practical protection exists



Alice



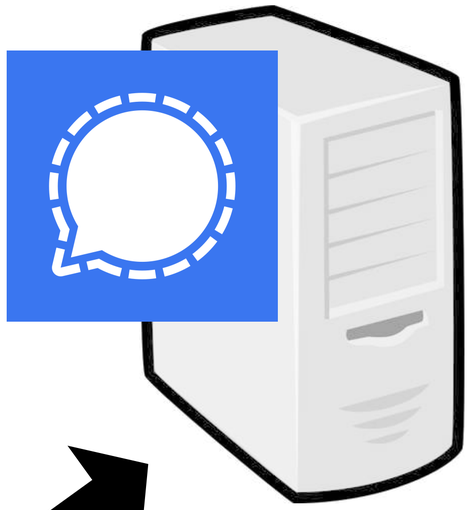
Bob

Confidentiality even in case of state compromise, but detecting such attacks is impossible

Chapter 3

1. Metadata protection during key retrieval

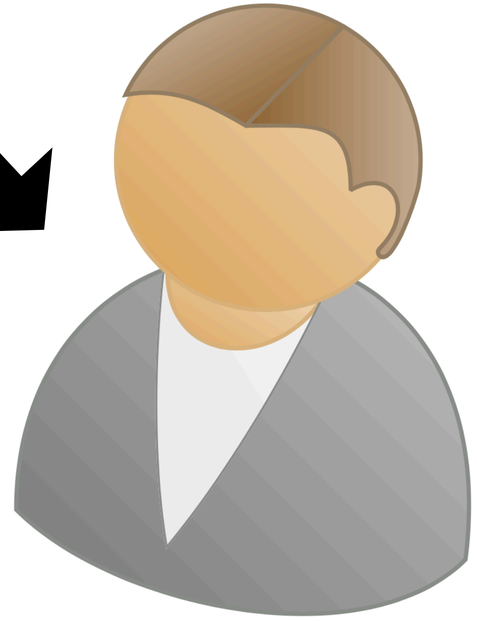
[**C**NCGWF23]



Chapter 4

3. Real-world deniability

[C**C**HD25]



Chapter 2

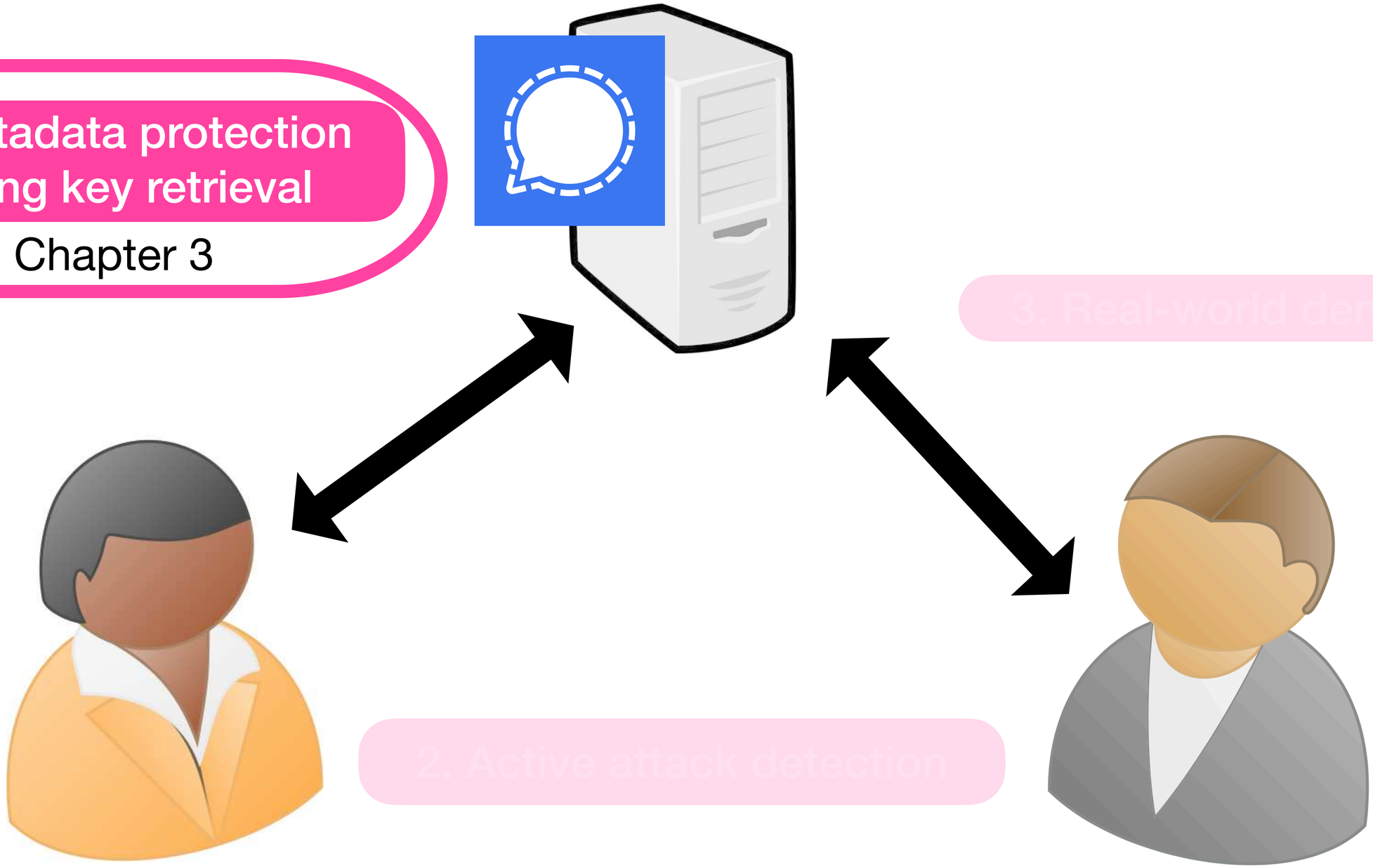
2. Active attack detection

[BC**C**HDV23]

4. Summary and conclusions

1. Metadata protection during key retrieval

Chapter 3

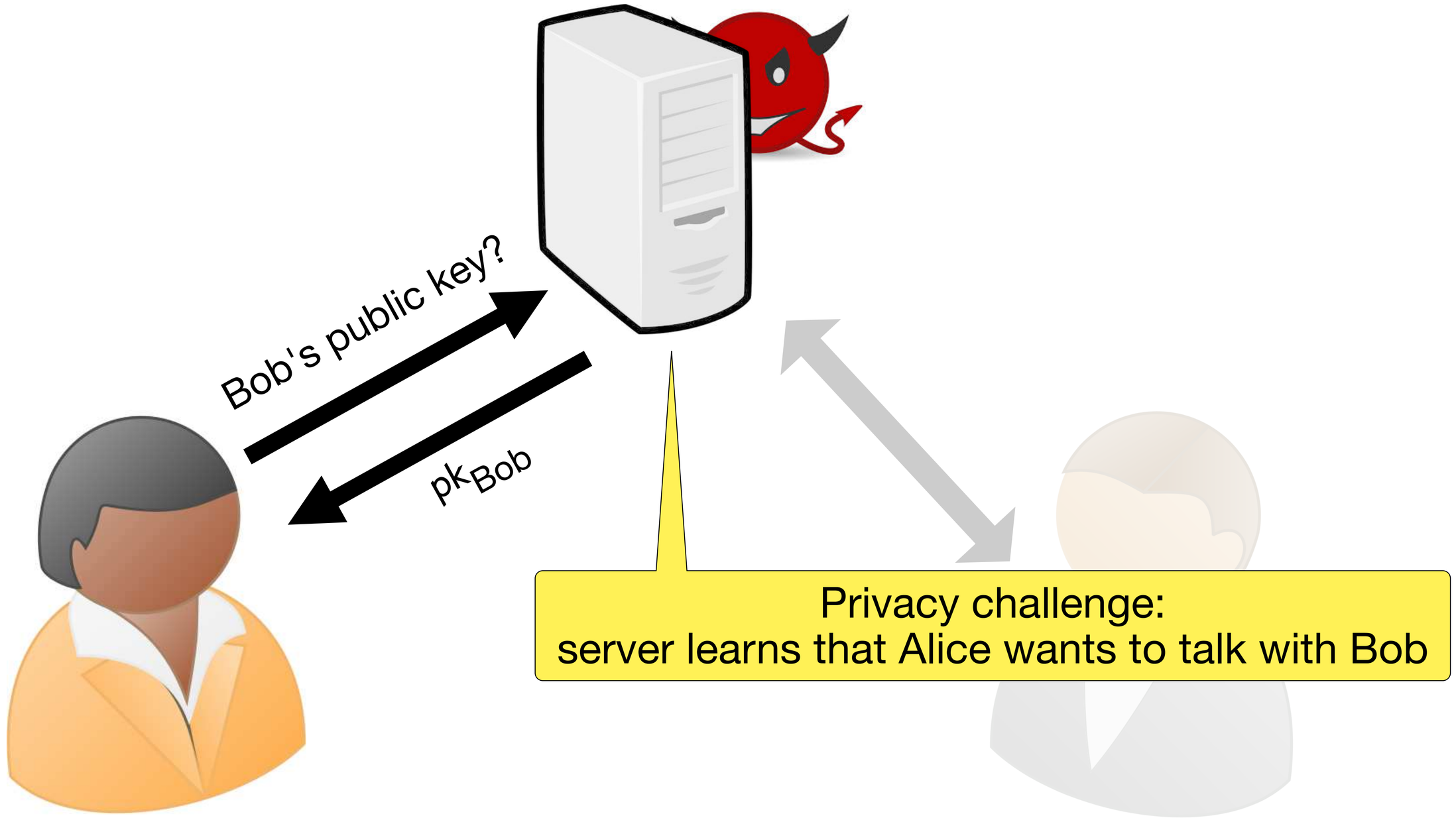


3. Real-world deniability

2. Active attack detection

4. Summary and conclusions

Authenticated private information retrieval
Simone Colombo, Kirill Nikitin, Henry Corrigan-Gibbs, David J. Wu, Bryan Ford
USENIX Security 2023

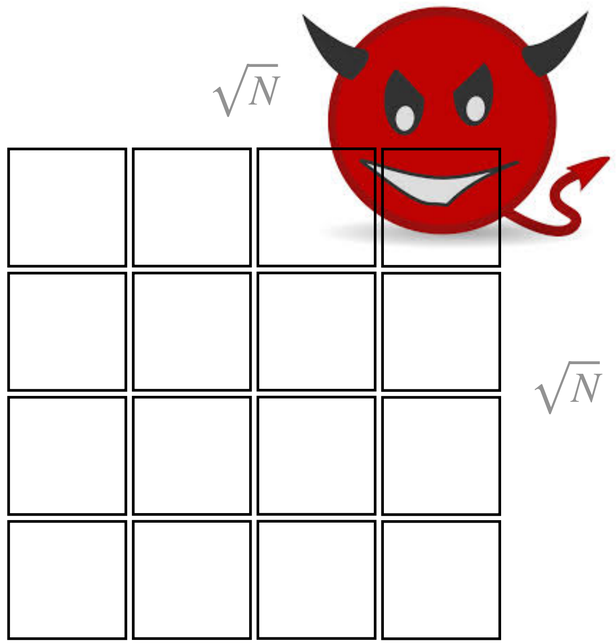

Bob's public key?
pk_{Bob}
Privacy challenge:
server learns that Alice wants to talk with Bob

Private information retrieval (PIR) [CGKS95]

holds index $i \in \{1,\ldots,N\}$

learns d_i

holds database $d \in \mathbb{F}^N$



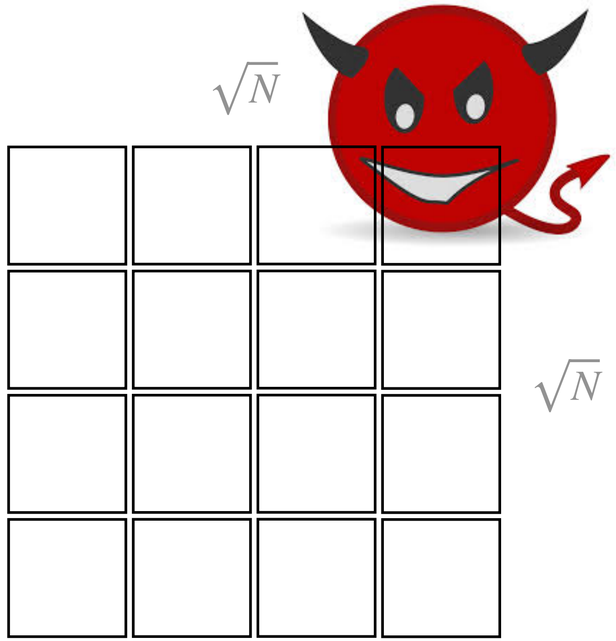
learns nothing

Private information retrieval (PIR) [CGKS95,WYGVZ17]

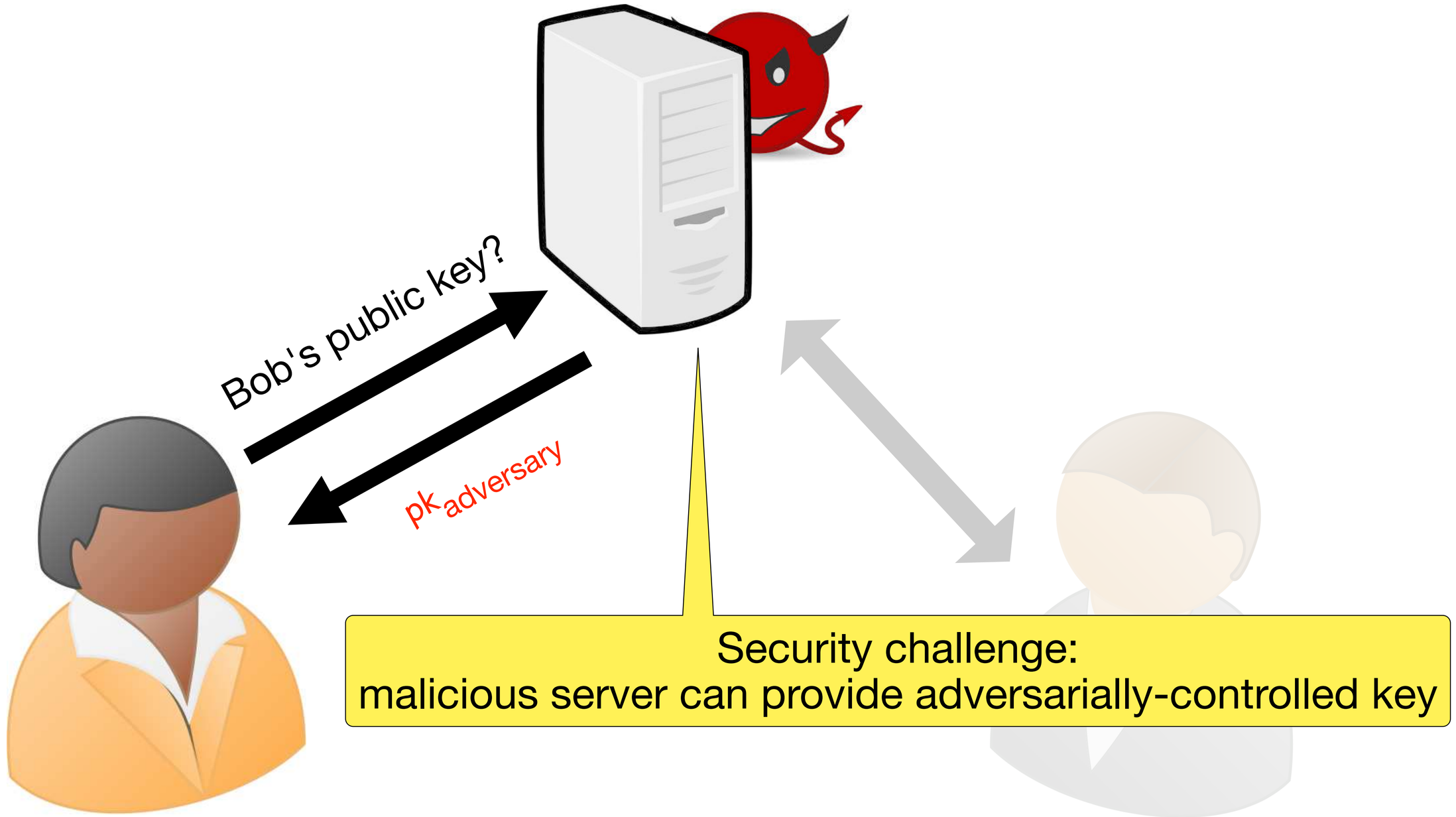
holds function f: $\mathbb{F}^N \to \mathbb{F}$

learns $f(d)$

holds database $d \in \mathbb{F}^N$



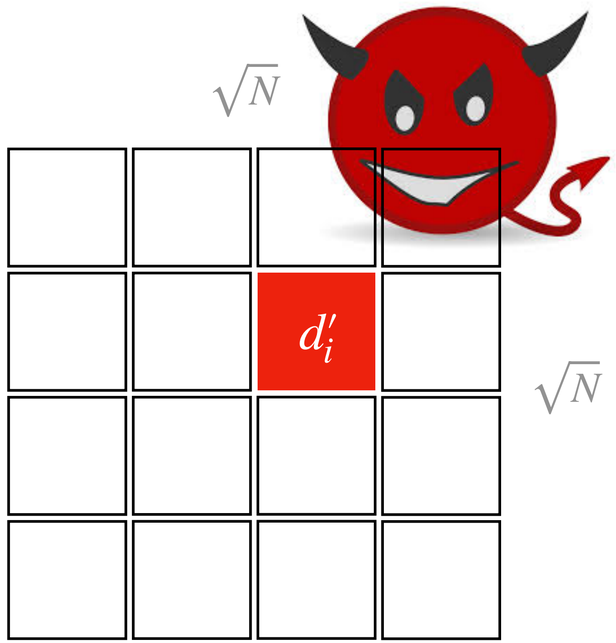
learns nothing


Bob's public key?
pk adversary
Security challenge:
malicious server can provide adversarially-controlled key

PIR does not consider integrity

holds index $i \in \{1,\ldots,N\}$

holds database $d \in \mathbb{F}^N$



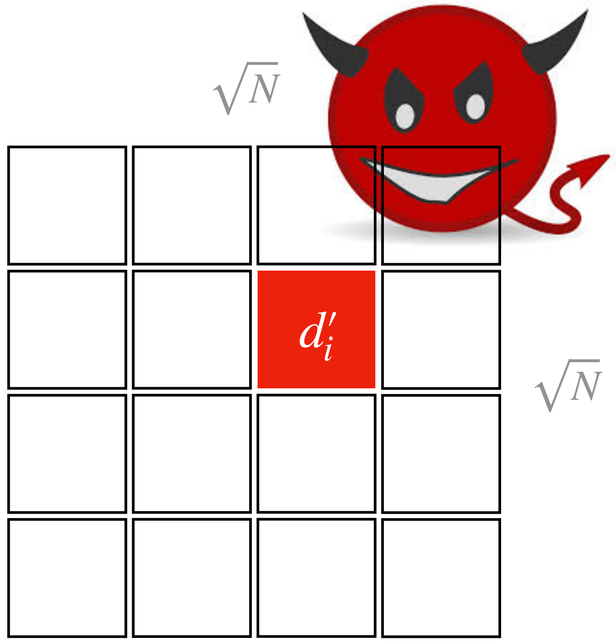
learns wrong d'_i

learns nothing

PIR does not consider integrity

holds index $i \in \{1,\ldots,N\}$

holds database $d \in \mathbb{F}^N$



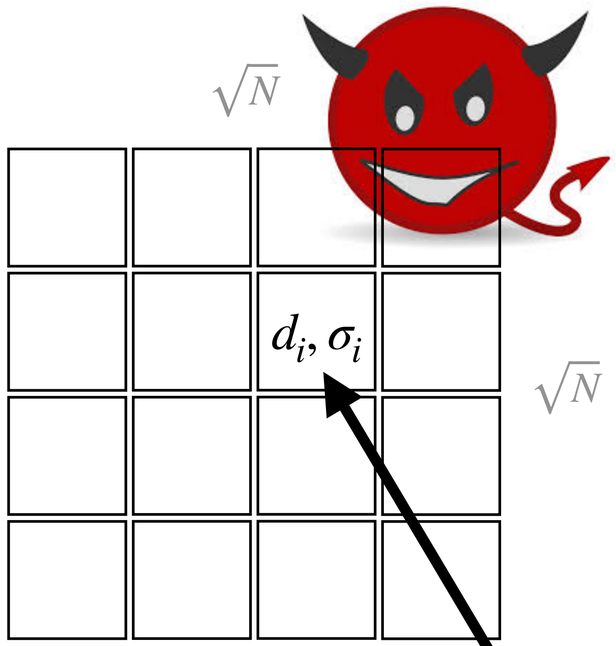
learns wrong $\mathsf{pk}_{\mathsf{adversary}}$

learns nothing

PIR and authentication are not enough

holds index $i \in \{1,\ldots,N\}$

holds database $d \in \mathbb{F}^N$



if Verify(pk, d_i, σ_i) = ⊤ return d_i
else abort

E.g. under Signal's secret key d_i, σ_i

PIR and authentication are not enough

holds index $i \in \{1,\ldots,N\}$

holds database $d \in \mathbb{F}^N$

Our contribution: authenticated private information retrieval

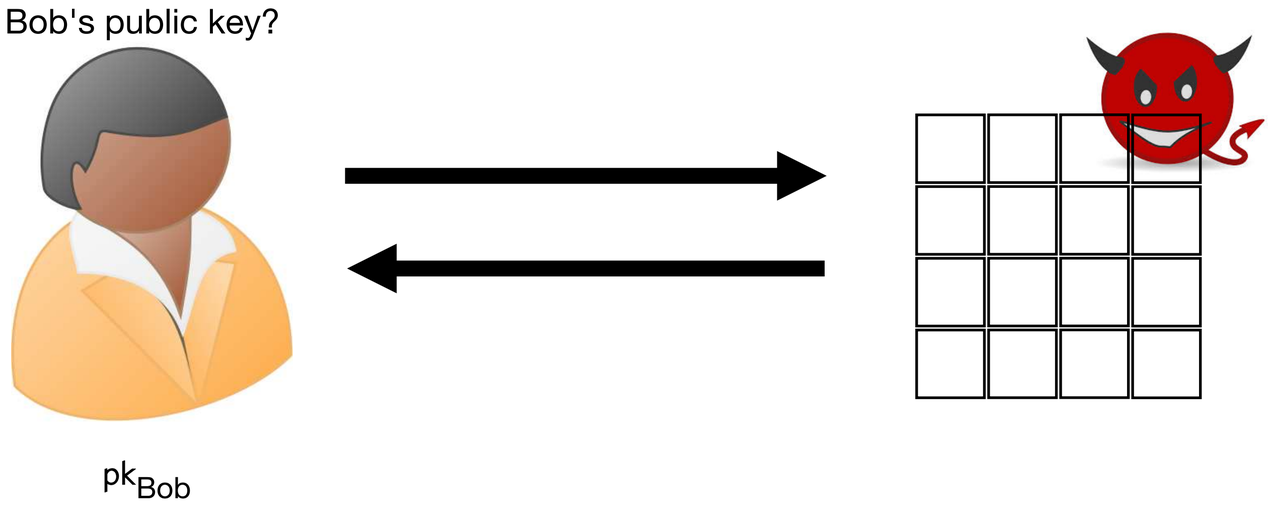
- First definition of authenticated PIR in multi-server and single-server settings.
- Multi-server schemes to fetch records and evaluate functions on database.
- Two single-server schemes to fetch single-bit records.
- Implementation and evaluation of all the schemes that we propose.
- Keyd, a PGP public-key directory service that builds on authenticated PIR.

if Verify(pk, d_i, σ_i) = 1 return d_i

else abort

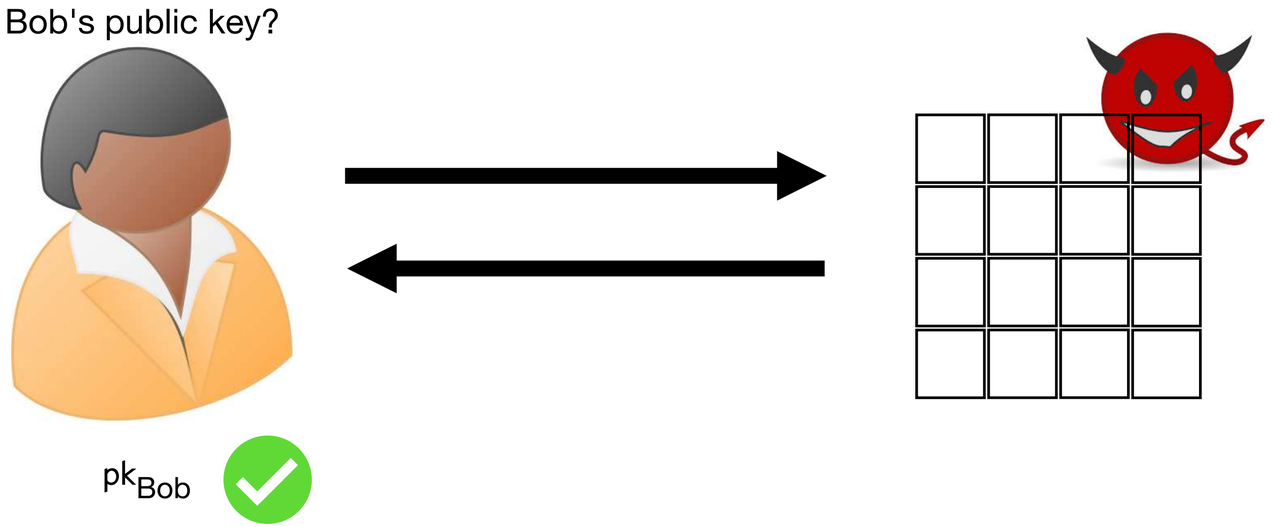
by revealing she queries the i[th] entry:
selective-failure attack [KS06].

Authenticated PIR properties



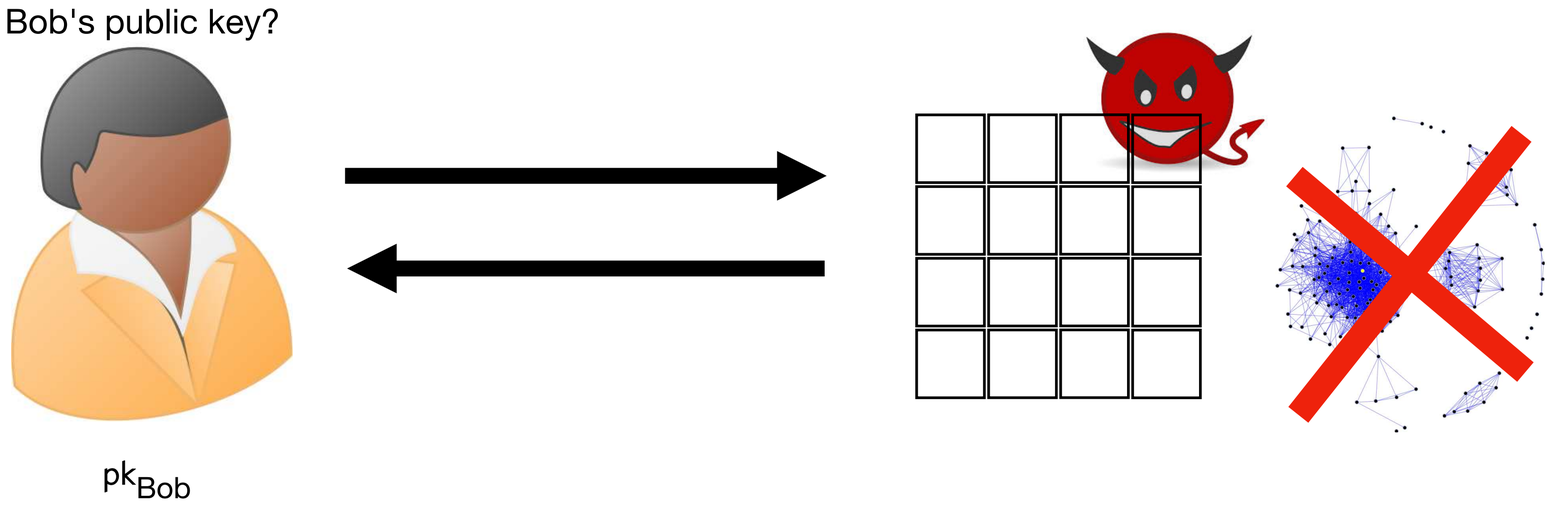
- Efficiency: Total communication is sublinear in the size of the database.

Authenticated PIR properties



- Efficiency: Total communication is sublinear in the size of the database.
- Correctness: If client and server are honest, the client recovers pk_{Bob}.

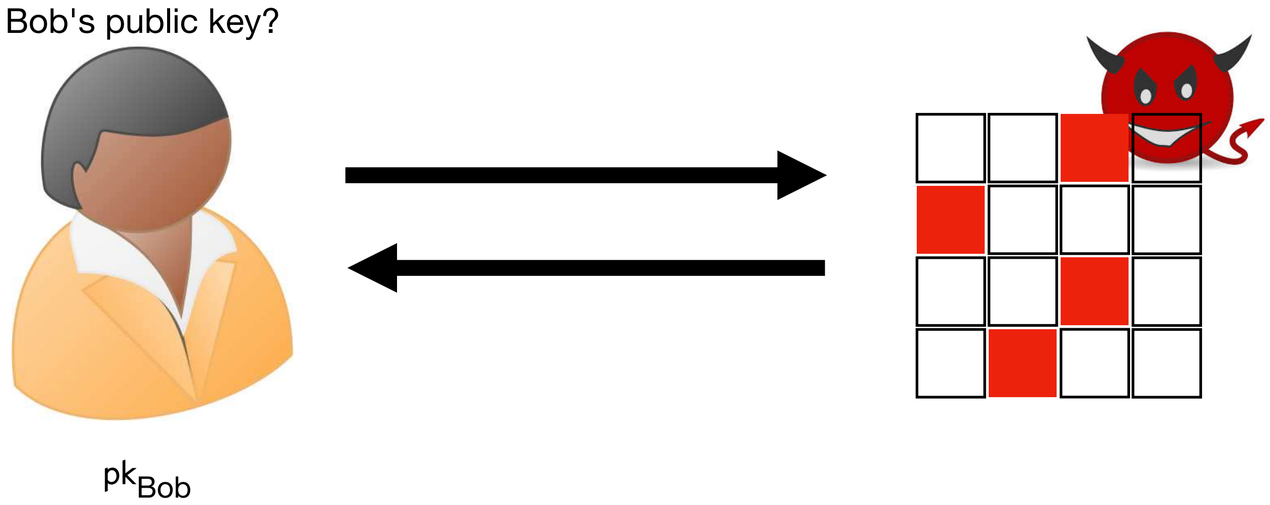
Authenticated PIR properties



- Efficiency: Total communication is sublinear in the size of the database.
- Correctness: If client and server are honest, the client recovers pk_{Bob}.
- Privacy: The server(s) learns nothing about the content of the client's query, even if the server(s) learns whether the client aborted during reconstruction.

Selective-failure attacks

Authenticated PIR properties

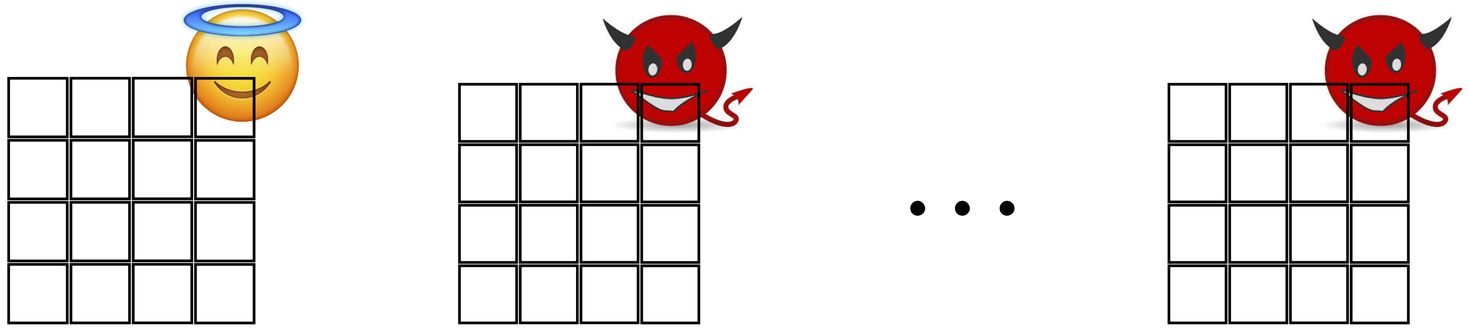


- Efficiency: Total communication is sublinear in the size of the database.
- Correctness: If client and server are honest, the client recovers pk_{Bob}.
- Privacy: The server(s) learn nothing about the content of the client's query, even if the server(s) learn whether the client aborted during reconstruction.
- Integrity: The client either outputs the authentic pk_{Bob} or aborts, except with negligible probability.

What does it mean "authentic pk_{Bob}"?

How to define authentic data?

Honest server’s view of the database.



Multi-server schemes

(1) Multi-servers, single-record query

Given a Merkle-tree scheme, on a database of size N

- the per-query communication is $O(\log N)$, same as unauthenticated PIR,
- the integrity error is negligible.

(2) Two-servers, single-record and aggregate queries

Given PRG and a field $\mathbb{F}$, on a database of size N

- the per-query communication is $O(\log N)$, same as unauthenticated PIR,
- the integrity error is $1/|\mathbb{F}|$

This talk (roughly)

Classic multi-server PIR [CGKS95]

= secret shares

pk$_{Bob}$ is in d_{22}, i.e., 2nd column

privacy

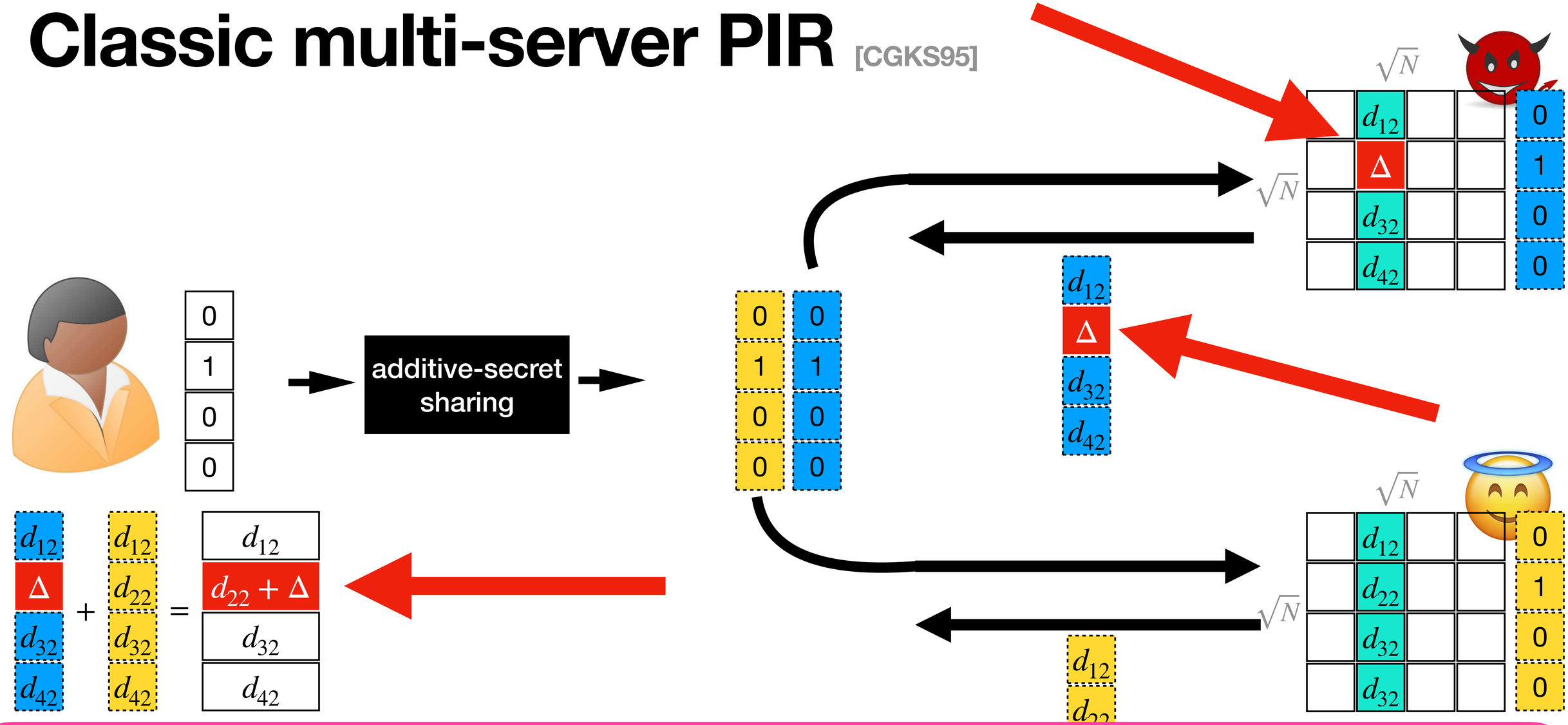
additive-secret sharing

correctness

communication $O(\sqrt{N})$

= secret shares

Classic multi-server PIR [CGKS95]



Key idea: two correlated queries, one for data and one to authenticate

Our contribution

Authenticated multi-server PIR

samples random $\alpha \in_R \mathbb{F}$

0	0
1	α
0	0
0	0

additive-secret sharing

0	0	0	0
1	α	1	α
0	0	0	0
0	0	0	0

0	0
1	α
0	0
0	0

$\sqrt{N}$

	d_{12}		0
	d_{22}		1
	d_{32}		0
	d_{42}		0

0	0
1	α
0	0
0	0

$\sqrt{N}$

	d_{12}		0
	d_{22}		1
	d_{32}		0
	d_{32}		0

Authenticated multi-server PIR

samples random $\alpha \in_R \mathbb{F}$

0	0
1	α
0	0
0	0

additive-secret sharing

0	0	0	0
1	α	1	α
0	0	0	0
0	0	0	0

0	0	0	0
1	α	1	α
0	0	0	0
0	0	0	0

d_{12}	αd_{21}	0	0
d_{22}	αd_{22}	1	α
0	0	0	0
1	α	0	0
0	0	0	0
0	0	0	0

d_{12}	αd_{12}	0	0
d_{22}	αd_{22}	0	0
d_{23}	αd_{23}	1	α
d_{24}	αd_{24}	0	0

$\sqrt{N}$

	d_{12}		0
	d_{22}		α
	d_{32}		0
	d_{42}		0

$\sqrt{N}$

	d_{12}		0
	d_{22}		α
	d_{32}		0
	d_{42}		0

d_{12}		d_{12}			αd_{12}		αd_{12}		αd_{12}
d_{22}	+	d_{22}	=	d_{22}	αd_{22}	+	αd_{22}	=	αd_{22}
d_{32}		d_{32}		α_{32}	αd_{32}		αd_{32}		αd_{32}
d_{42}		d_{42}		d_{42}	αd_{42}		αd_{42}		αd_{42}

Authenticated multi-server PIR integrity

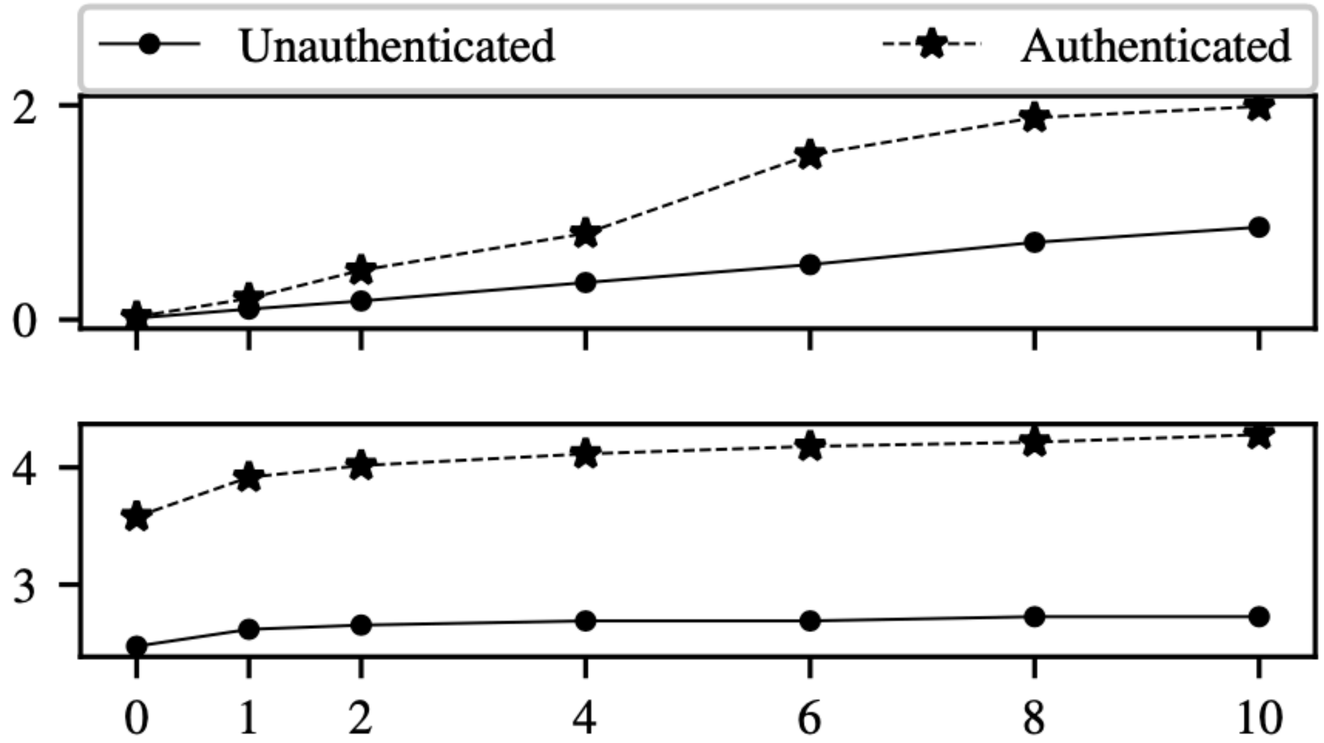
if $\alpha \cdot \left(\begin{matrix} d_{12} \\ d_{22} \\ d_{32} \\ d_{42} \end{matrix} + \begin{matrix} d_{12} \\ d_{22} \\ d_{32} \\ d_{42} \end{matrix} \right) = \begin{matrix} \alpha d_{12} \\ \alpha d_{22} \\ \alpha d_{32} \\ \alpha d_{42} \end{matrix} + \begin{matrix} \alpha d_{12} \\ \alpha d_{22} \\ \alpha d_{32} \\ \alpha d_{42} \end{matrix}$

return second element of $\begin{matrix} d_{12} \\ d_{22} \\ d_{32} \\ d_{42} \end{matrix} + \begin{matrix} d_{12} \\ d_{22} \\ d_{32} \\ d_{32} \end{matrix} = \begin{matrix} d_{12} \\ d_{22} \\ d_{32} \\ d_{42} \end{matrix}$

else abort

communication $O(\sqrt{N})$, function secret sharing reduces to $O(\lambda \log N)$ [BGI16]

Evaluation: single-record queries (Merkle)



User time [s]

maximum 3x

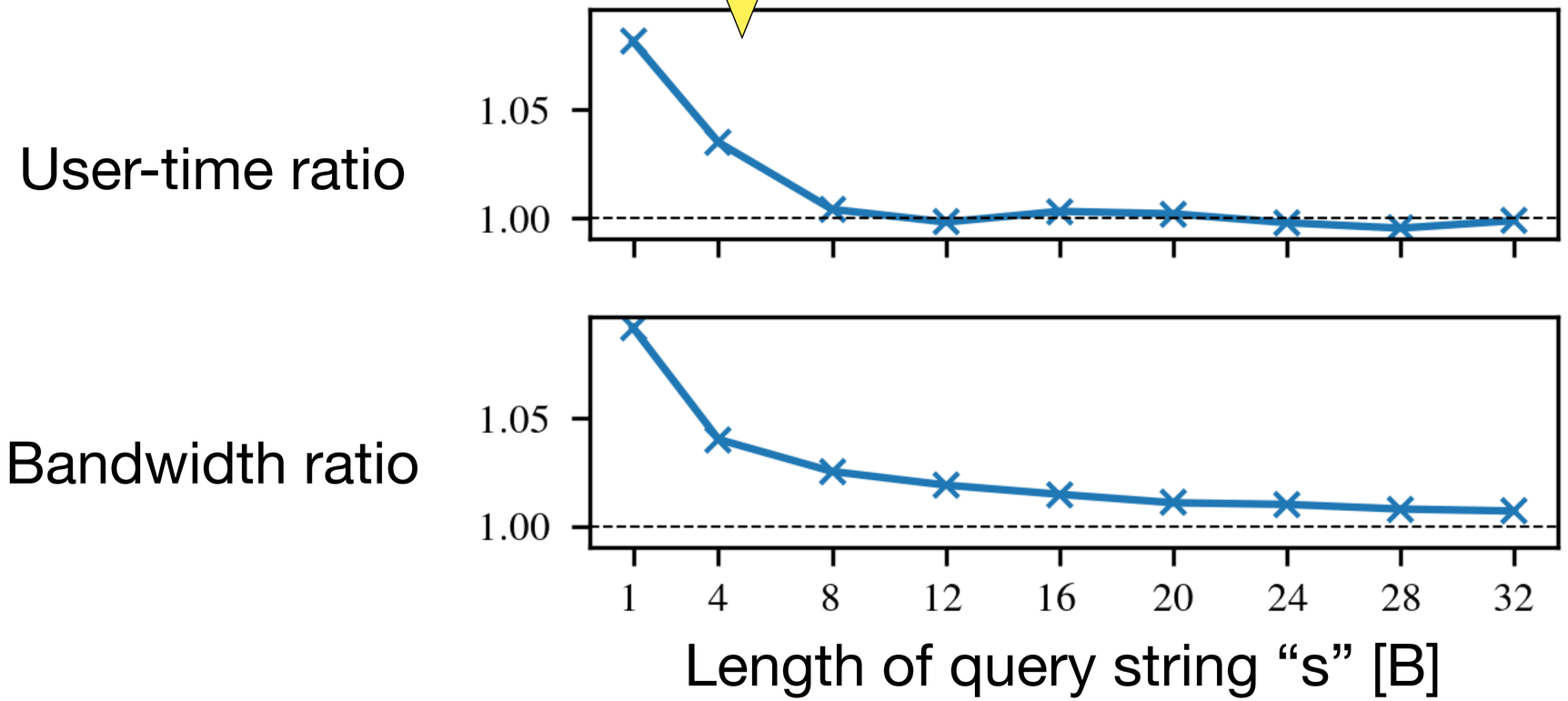
Bandwidth [KiB]

maximum 1.6x

Cost of retrieving a 1KiB record

Evaluation: aggregate queries

ratio of authenticated and classic unauthenticated PIR



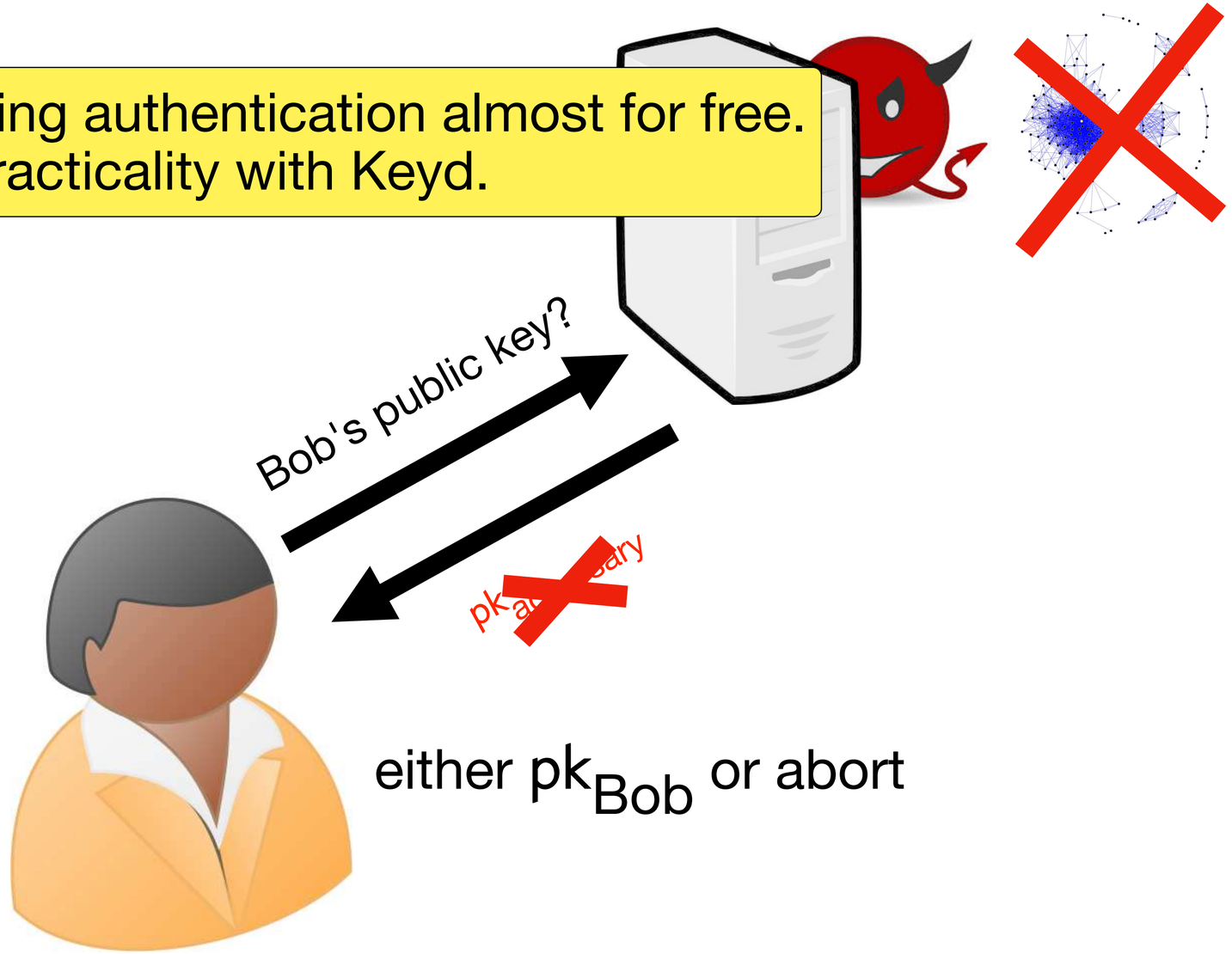
SELECT COUNT(*) FROM keys WHERE email LIKE “%s”

Count emails that end with string “s”

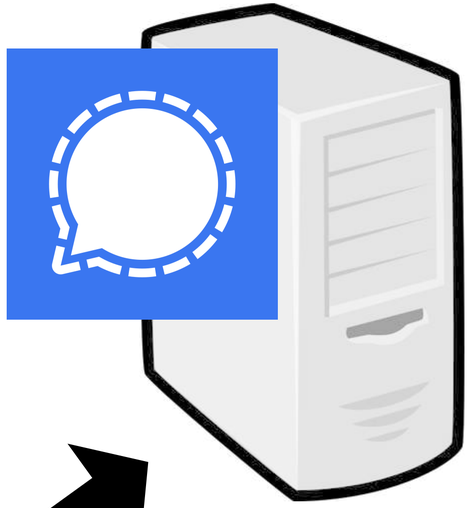

DEMO TIME

Summary of metadata protection during key retrieval

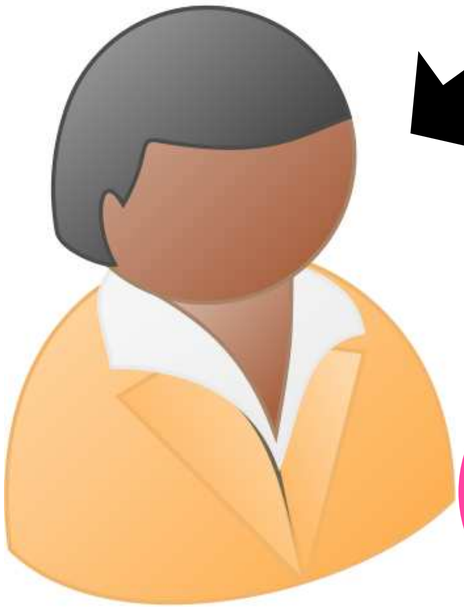
In multi-server setting authentication almost for free.
We demonstrate practicality with Keyd.

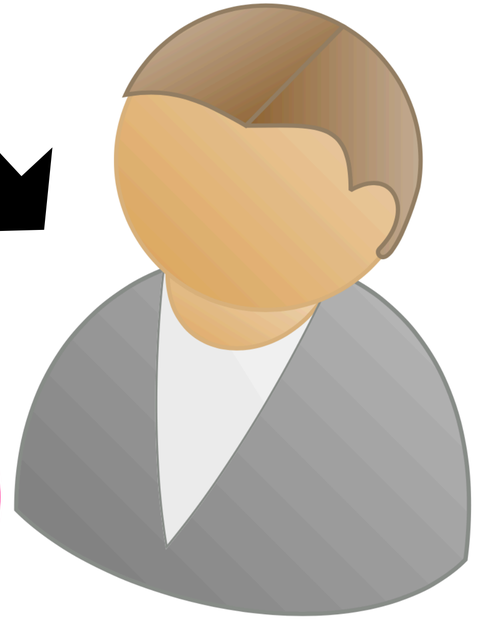


1. Metadata protection during key retrieval



3. Real-world deniability



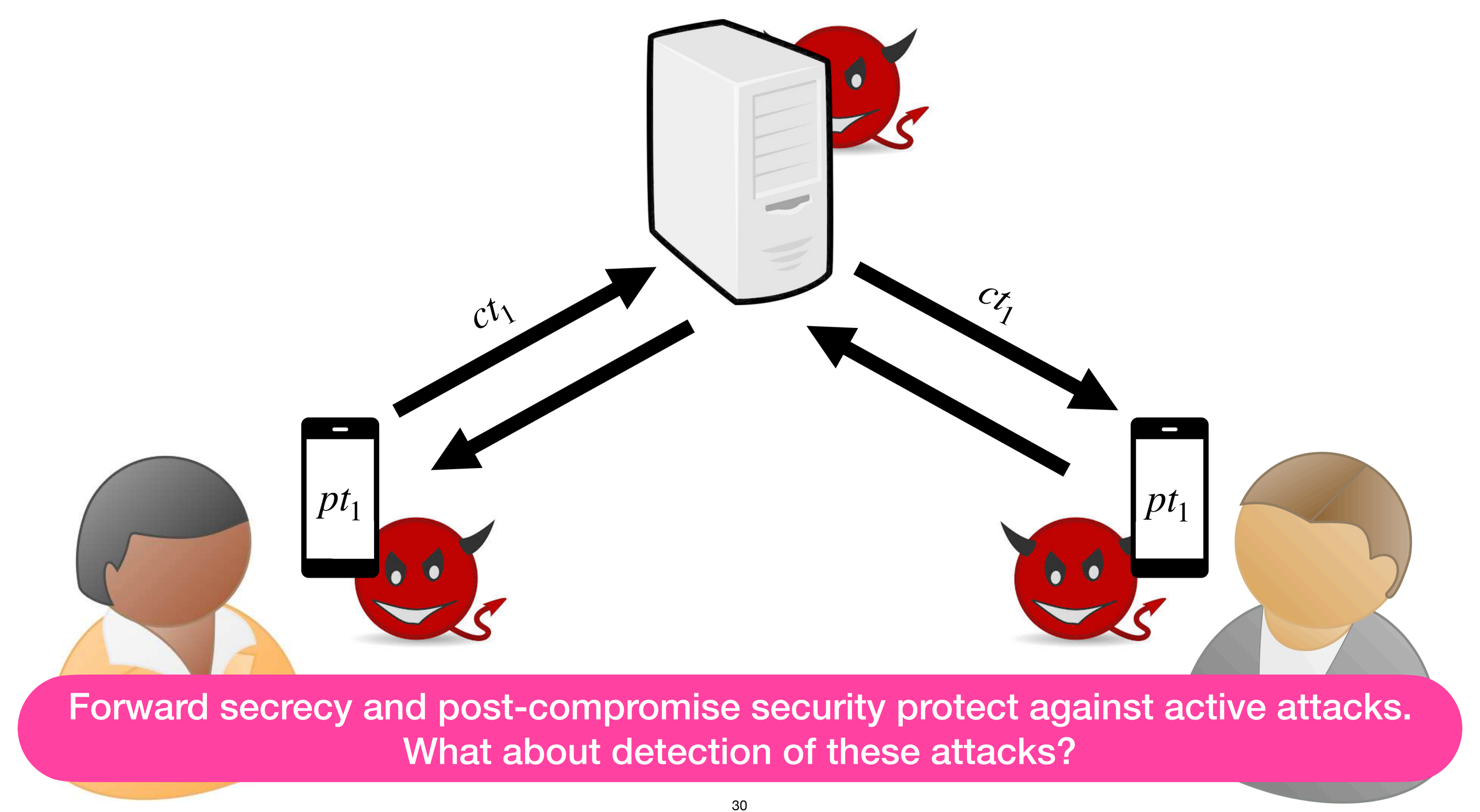


2. Active attack detection

Chapter 2

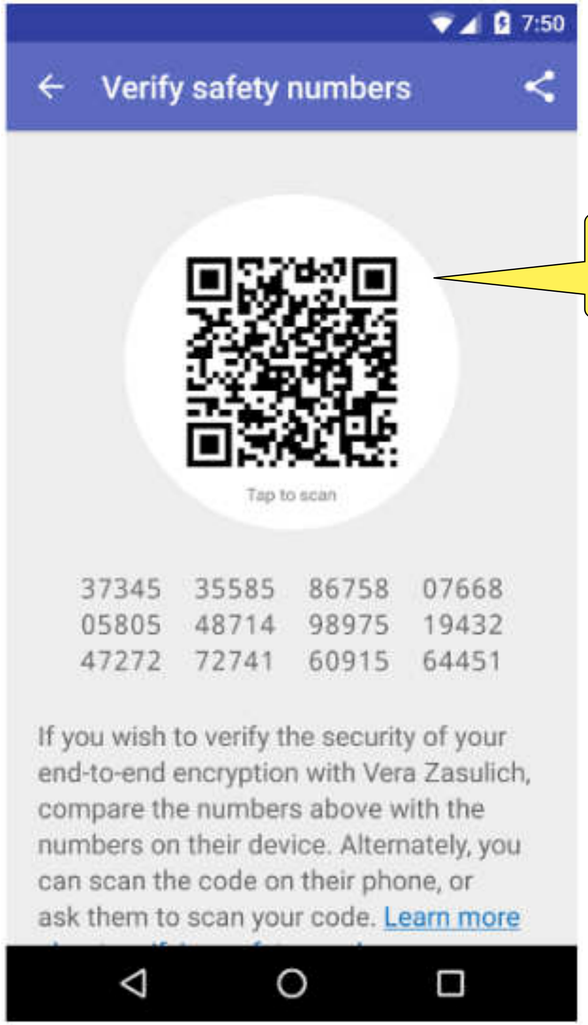
4. Summary and conclusions

On Active Attack Detection in Messaging with Immediate Decryption
Khashayar Barooti, Daniel Collins, Simone Colombo, Loïs Huguenin-Dumittan, Serge Vaudenay
CRYPTO 2023



Forward secrecy and post-compromise security protect against active attacks.
What about detection of these attacks?

Signal’s safety numbers



Only protect initial key exchange

Active attacks

In our model the adversary can expose states, control randomness and invoke algorithms via oracles

I go to the protest

Don’t go to the protest!

ling thanks to PCS, but attac

ct_1

Send(pt_2)

ct_2

ct_2'

Recv(ct_2') → pt_2'

Concretely, the adversary steals Alice’s phone and copies the cryptographic state

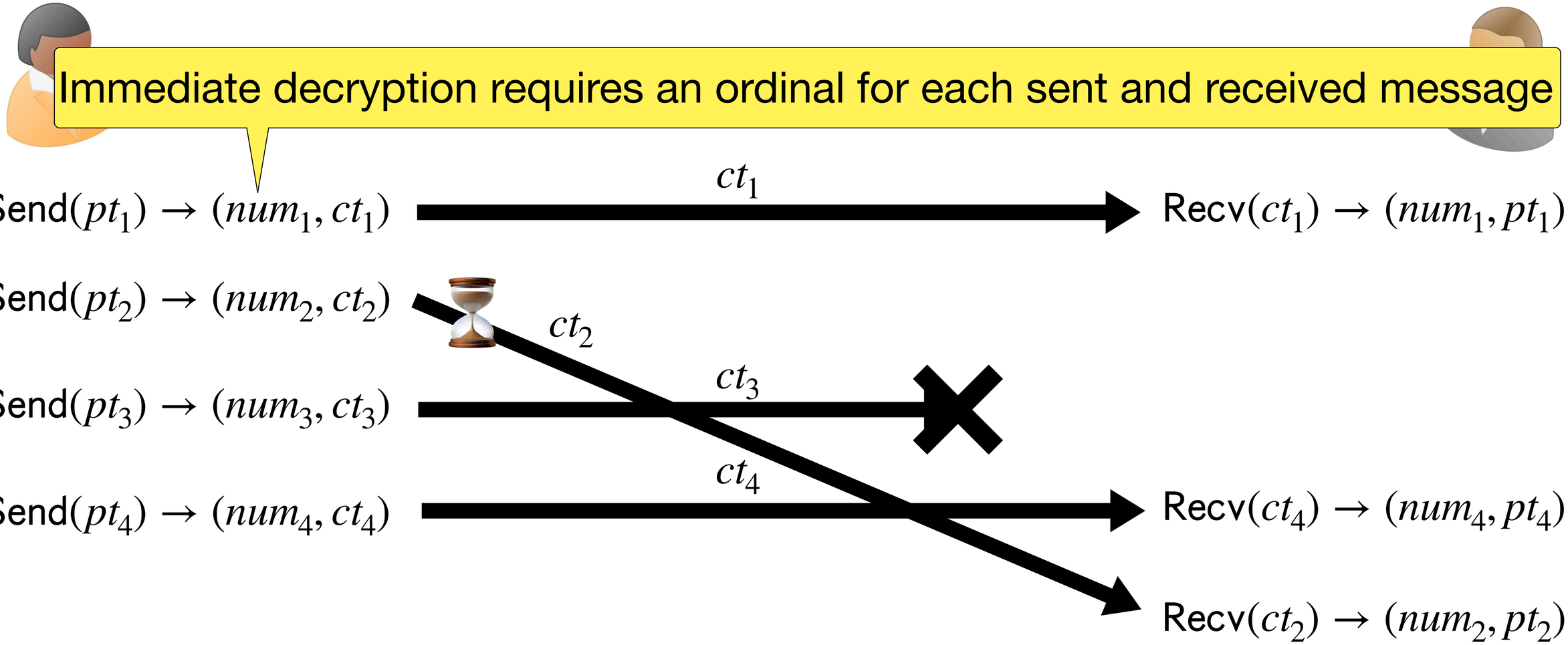
ct_4

Send(pt_4)

Recv(ct_4) → pt_4

Immediate decryption (ID) [ACD19]

Schemes support out-of-order delivery and message loss at the protocol level.



Immediate decryption (ID) [ACD19]

Schemes support out-of-order delivery and message loss at the protocol level.

Notation simplified for the rest of the talk

Our contribution: active attack detection with immediate decryption

- New authenticated ratcheted communication primitive.
- In-band detection with immediate decryption with r-RID and s-RID notions.
- Out-of-band detection with ID with new r-UNF and s-UNF security notions.
- Optimisations for s-RID and s-UNF security towards practicality.

$\mathsf{Recv}(ct_2) \rightarrow (num_2, \quad)$

Out-of-band active attack detection

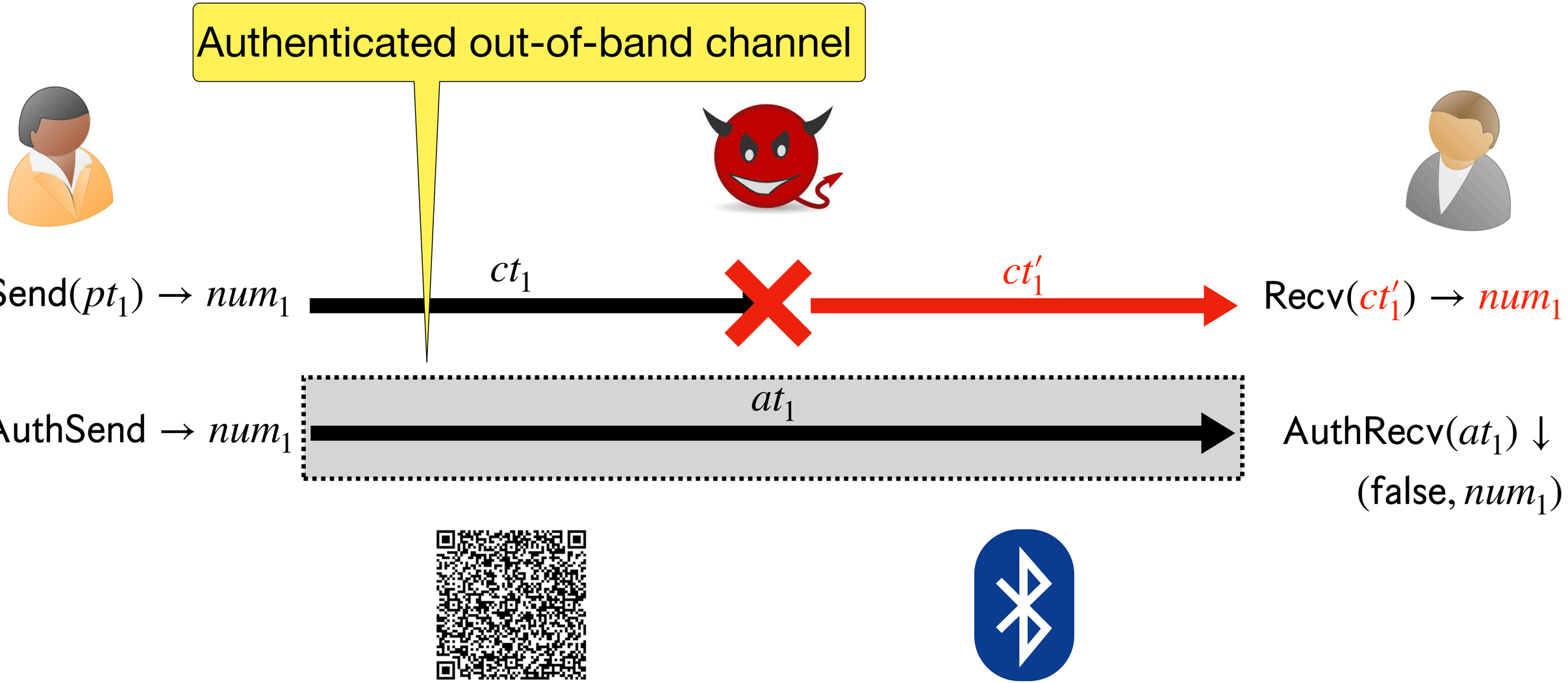
$\text{Send}(pt_1) \rightarrow num_1$ ct_1 ct'_1 $\text{Recv}(ct'_1) \rightarrow num_1$

$\text{Send}(pt_2) \rightarrow num_2$ ct_2 ct'_2 $\text{Recv}(ct'_2) \rightarrow num_2$

$\text{Send}(pt_3) \rightarrow num_3$ ct_3 ct'_3 $\text{Recv}(ct'_3) \rightarrow num_3$

If the adversary blocks all messages, we must use an out-of-band channel

Out-of-band active attack detection



In-band active attack detection

$\text{Send}(pt_1) \rightarrow num_1$ ct_1 ct_1' $\text{Recv}(ct_1') \rightarrow num_1$

$\text{Send}(pt_2) \rightarrow num_2$ ct_2 ct_2' $\text{Recv}(ct_2') \rightarrow num_2$

If one honest message goes through, in-band detection is possible

$\text{Send}(pt_3) \rightarrow num_3$ ct_3 $\text{Recv}(ct_3) \rightarrow num_3$

We define in-band active attack detection through r-RID and s-RID security

r-RID security

If Bob receives a forgery with num, then Bob rejects all Alice's messages with $num' > num$.

Bob **doesn't accept** as Alice's message has $num_3 > num_2$

$\mathsf{Send}(pt_1) \to num_1$

ct_1

ct_2'

$\mathsf{Recv}(ct_2') \to num_2$

ct_3

$\mathsf{Send}(pt_3) \to num_3$

$\mathsf{Recv}(ct_3) \to \perp$

Bob **accepts** as Alice's message has $num_1 < num_2$

$\mathsf{Recv}(ct_1) \to num_1$

s-RID security

If Bob receives a forgery with num at time t, then Alice rejects all Bob's messages sent after time t.

Alice doesn't accept as sent by Bob **after** the forgery

ct_1

$\mathsf{Send}(pt_1) \to num_1$

ct_2'

$\mathsf{Recv}(ct_2') \to num_2$

t

ct_3

$\mathsf{Recv}(pt_3) \to \perp$

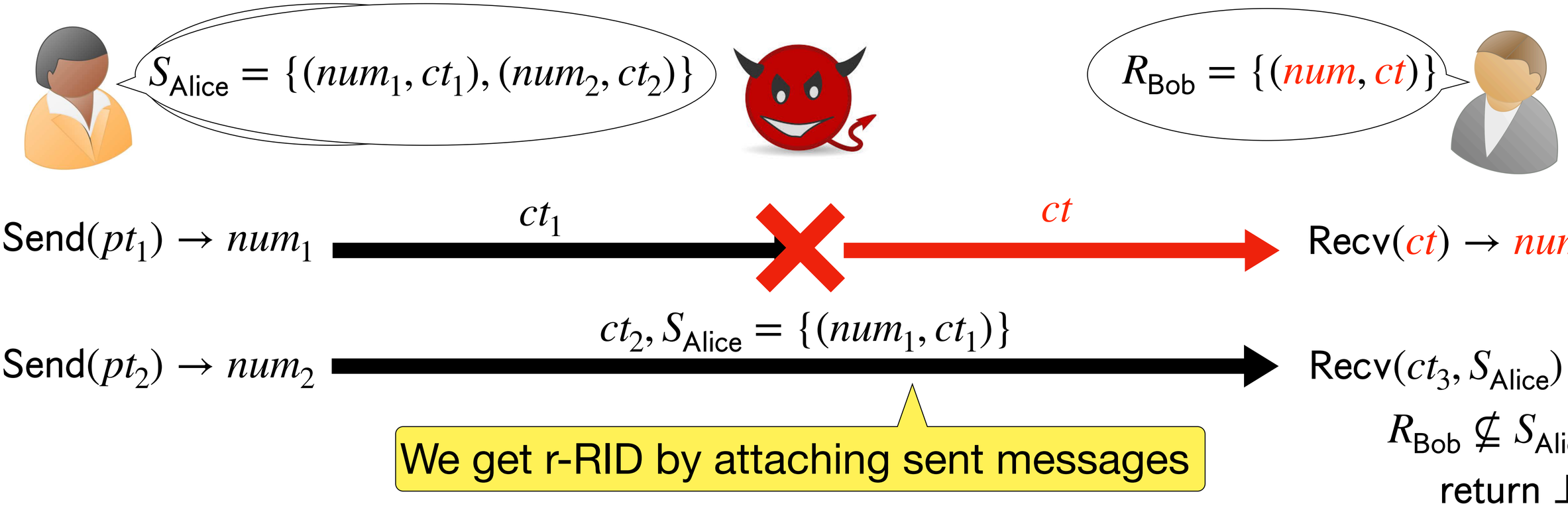
$\mathsf{Send}(pt_3) \to num_3$

$\mathsf{Recv}(pt_1) \to num_1$

Alice accepts since sent by Bob **before** the forgery

A simple RID construction (r-RID + s-RID)

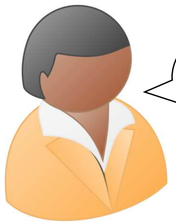
Attach all sent and received ciphertexts to every ciphertext and check at reception.



A simple RID construction (r-RID + s-RID)

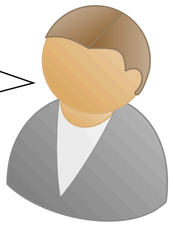
Attach all sent and received messages to every ciphertexts and check at reception.

This approach works supports immediate decryption



$S_{\text{Alice}} = \{(num_1, ct_1)\}$

$R_{\text{Bob}} = \{(num, ct)\}$



$\text{Send}(pt_1) \rightarrow num_1$

ct_1

ct

$\text{Recv}(ct) \rightarrow num$

$ct_2, R_{\text{Bob}} = \{(num, ct)\}$

$\text{Recv}(ct_2, R_{\text{Bob}})$:

$R_{\text{Bob}} \nsubseteq S_{\text{Alice}}$

return $\perp$

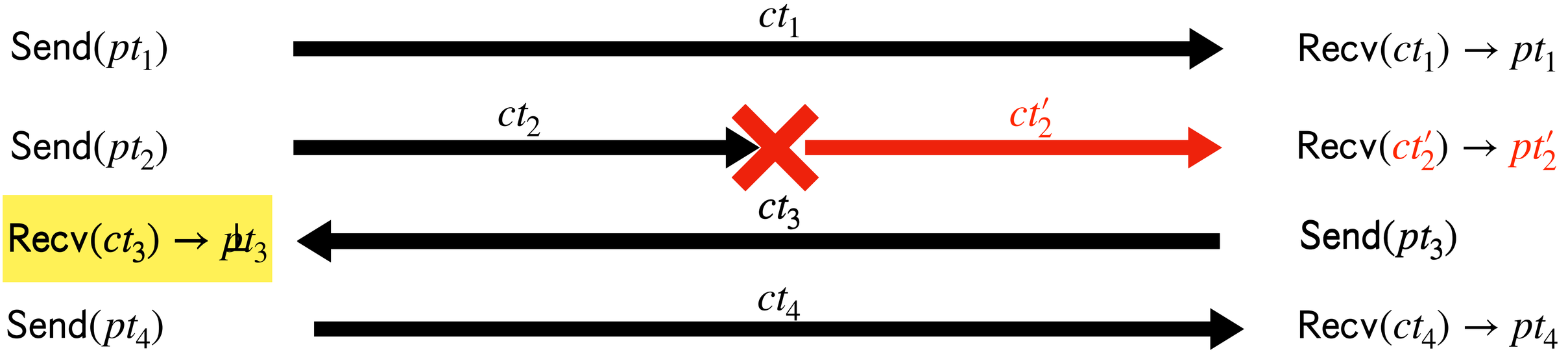
$\text{Send}(pt_2) \rightarrow num_2$

We get s-RID by attaching received messages

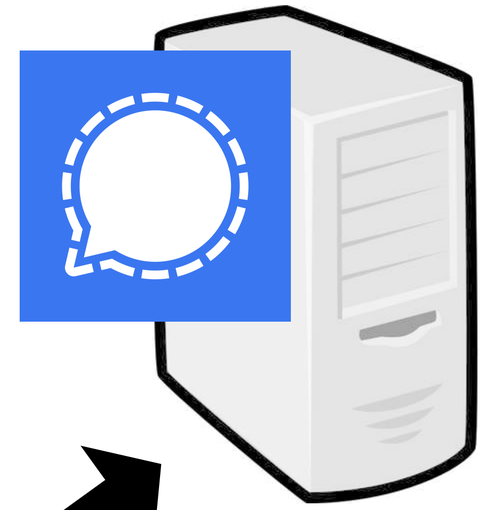
Summary of active attack detection

If adversary blocks in-band channel, parties can use out-of-band authenticated one.
s-RID is likely practical and we propose optimisations to reduce overhead.



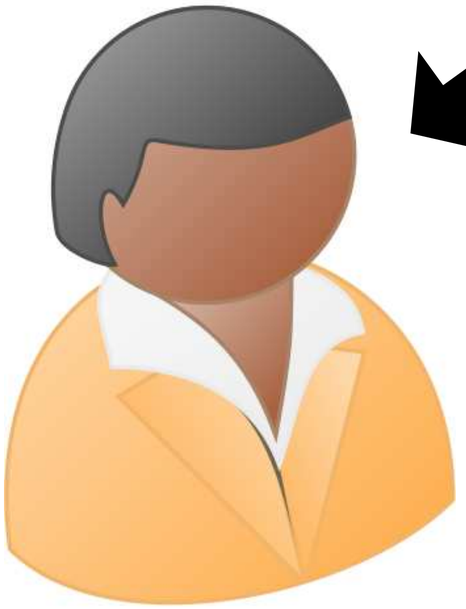


1. Metadata protection during key retrieval

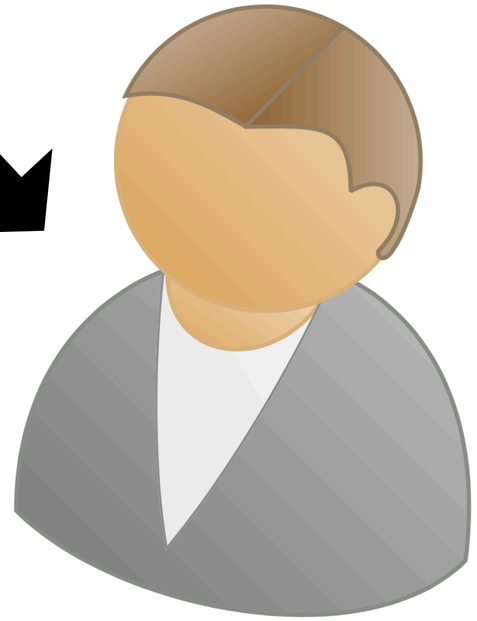


3. Real-world deniability

Chapter 4



2. Active attack detection



4. Summary and conclusions

Real-World Deniability in Messaging
Daniel Collins, Simone Colombo, Loïs Huguenin-Dumittan
PETS 2025, RWC 2023

Let’s go to the protest!

Our contribution: analysis of deniability’s impracticality, and solutions to achieve real-world deniability.

- Model to analyze real-world deniability in messaging.
- Technical case studies: Signal application and DKIM protected email.
- Legal case study: 140 Swiss court cases that use WhatsApp as evidence.
- Discussion on deniability and on how to achieve real-world deniability.

Technical case study: Signal

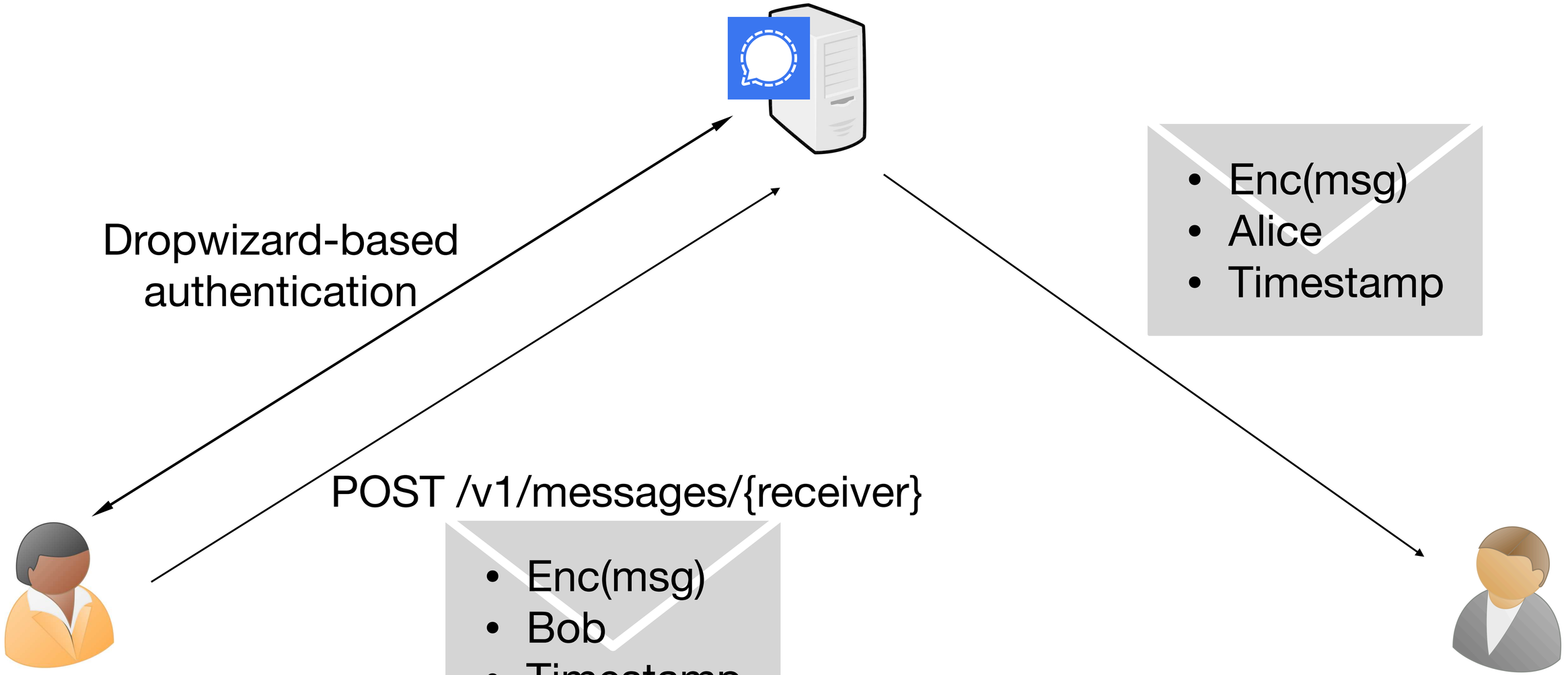
Signal claims to provide deniability and recent works show it achieves some form of cryptographic deniability [VGIK20, FJ24].

On the Cryptographic Deniability of the Signal Protocol

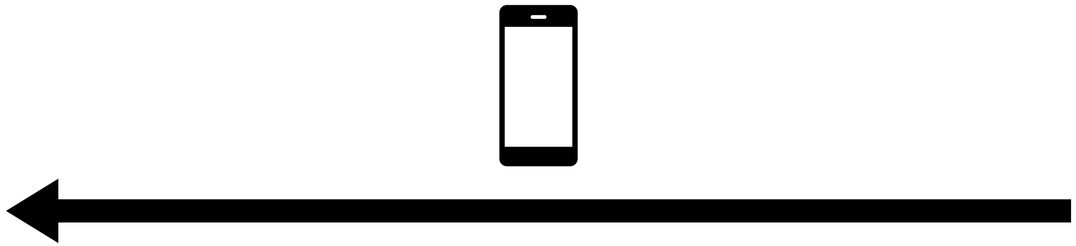
Nihal Vatandas[1], Rosario Gennaro[1], Bertrand Ithurburn[1], and Hugo Krawczyk[2]

Is this sufficient in practice?

Signal with classic authentication



Classic authentication hinders deniability



Look at my phone,
Alice sent me this message

If Bob's phone contains Alice's message, then

If server logs, even worse

- either Alice really sent it after authenticating with the server, or
- Bob modified the local message database.

Signal is technically undeniable unless Bob knows how to tamper with the device.
What about the legal impact of deniability?

Legal case study methodology

- Manual analysis of 341 penal cases in Switzerland that mention “WhatsApp”.
- Research questions:
  - Do judges in Swiss courts use WhatsApp as evidence?
  - When they do, is their usage contested by any of the parties involved?
  - What are the reasons used to dispute the legal validity of such messages?
  - How do judges respond to these disputes?

No mention of Signal in cases

Legal case study results

Deniability is not invoked in these two cases

Total cases	N/A	Evidence	Contested	Rejected
341	201 (59%)	140 (41%)	2 (0.6%)	0

Yadav et al. [YGS23] reach similar results in an analysis of US court cases

Cryptographic deniability fails technically and (likely) legally: what to do?

A possible solution

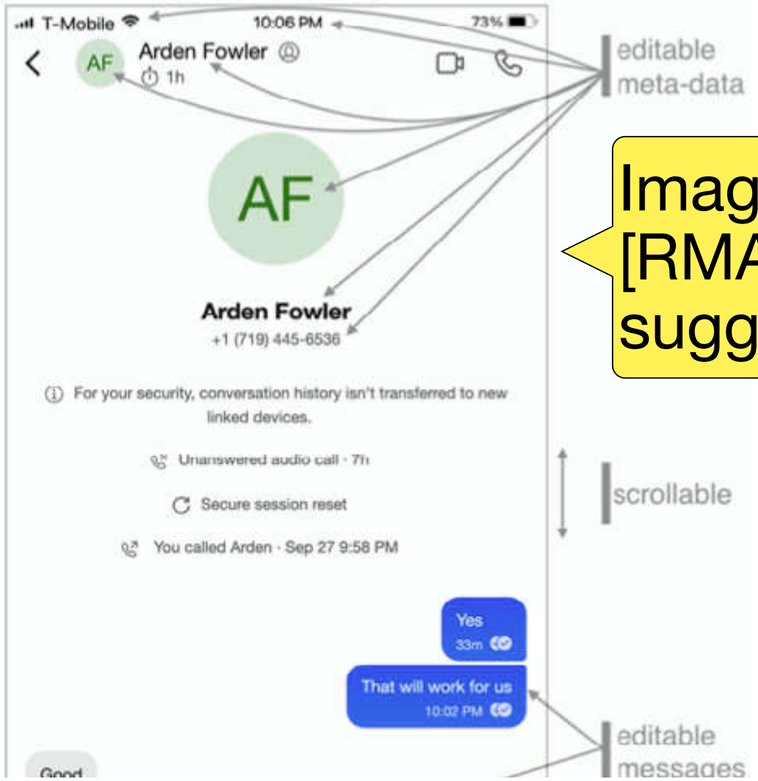


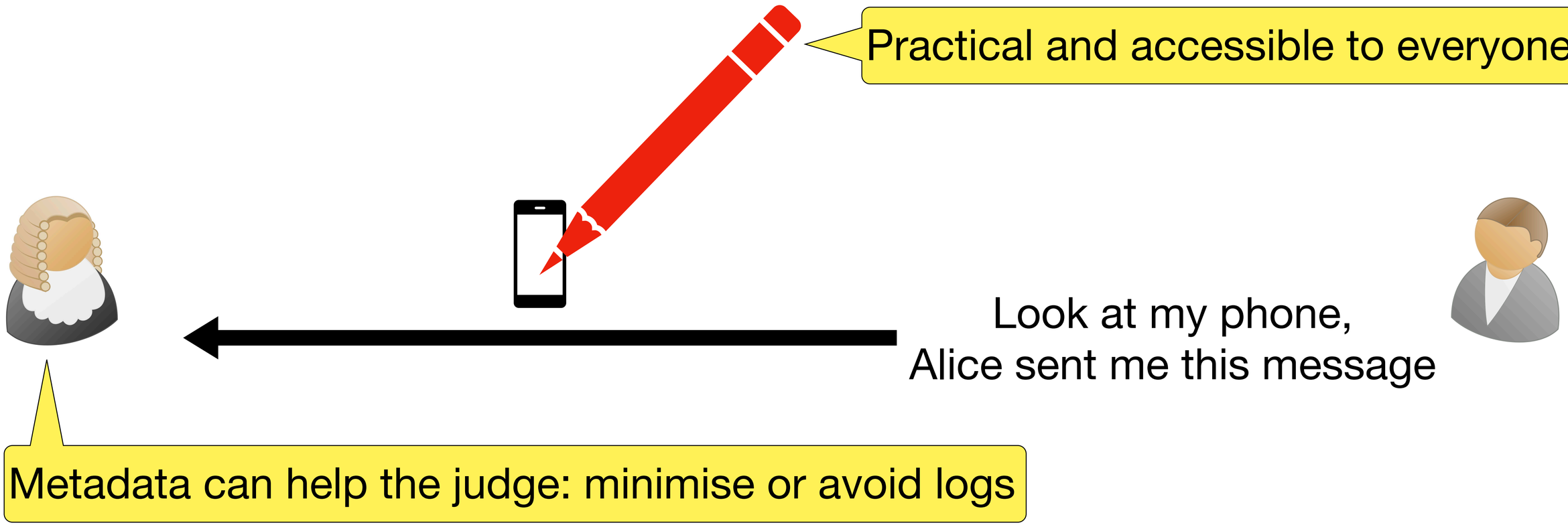
Image from Reitinger et al. [RMAMM23], who independently suggest it could improve deniability

- either Alice really sent it after authenticating with the server, or
- Bob modified the local message database.

Artifact: Reproduced

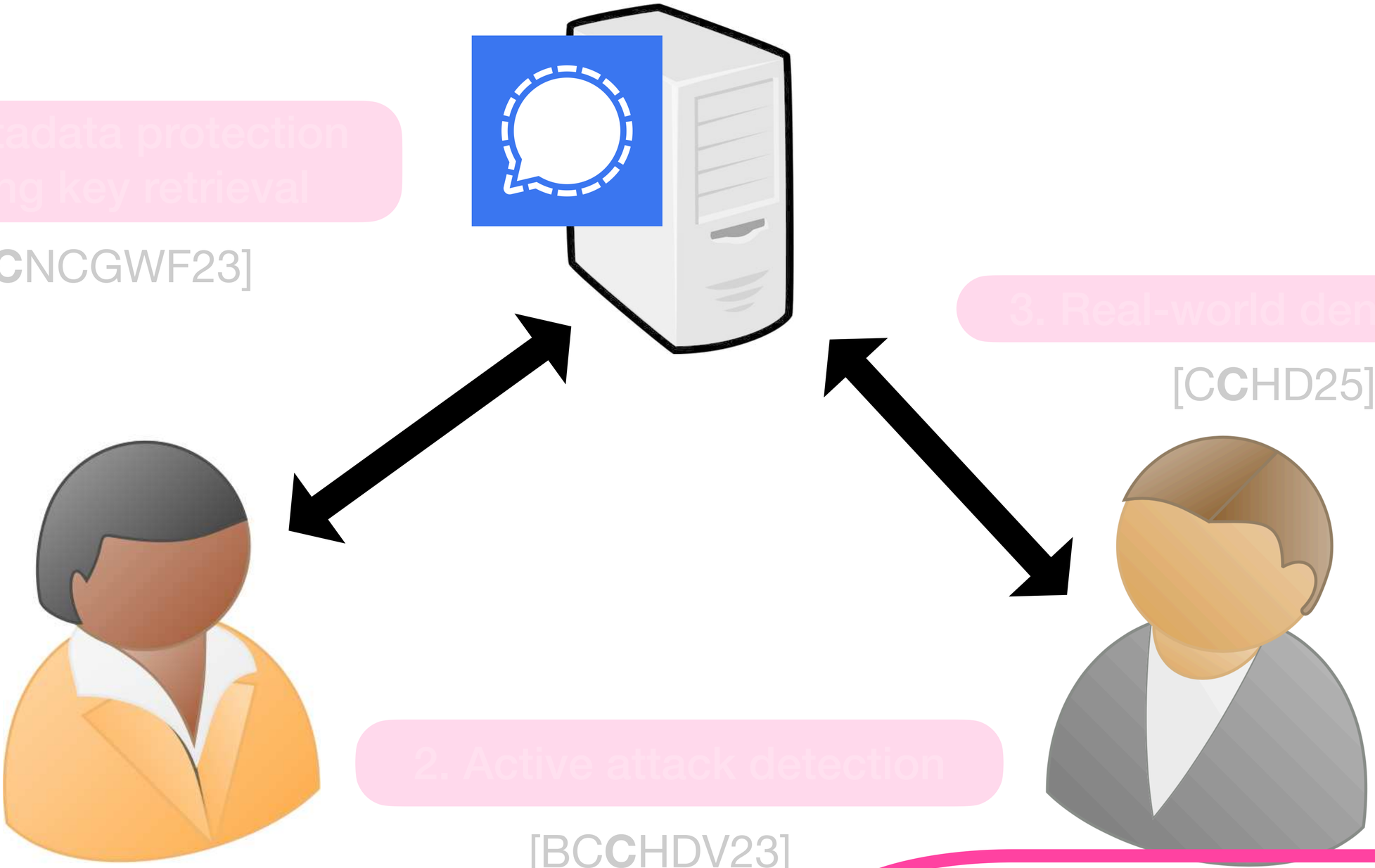
As well as available and functional

Summary of real-world deniability



1. Metadata protection during key retrieval

[CNCGWF23]



3. Real-world deniability

[CCHD25]

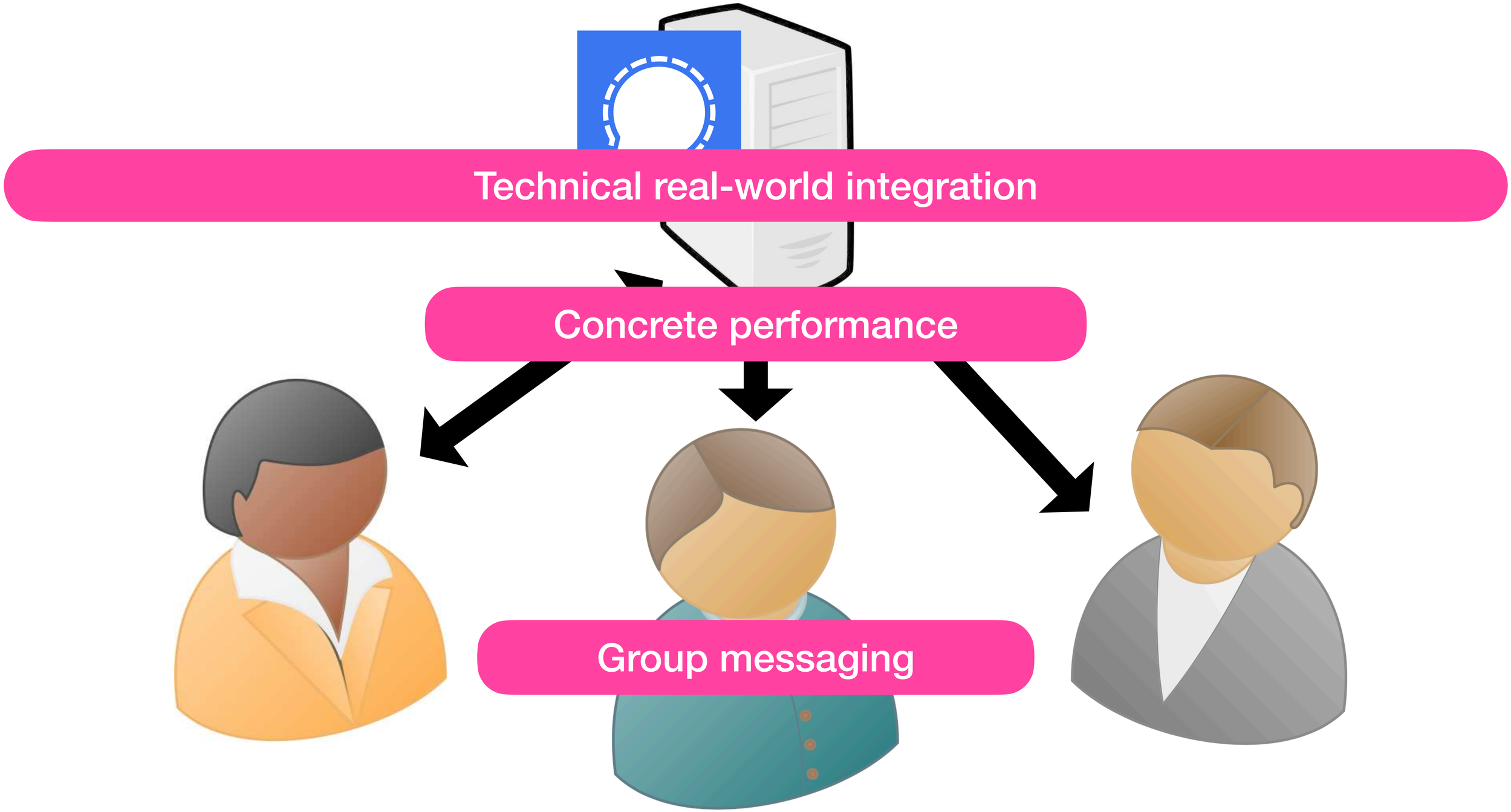
2. Active attack detection

[BCCHDV23]

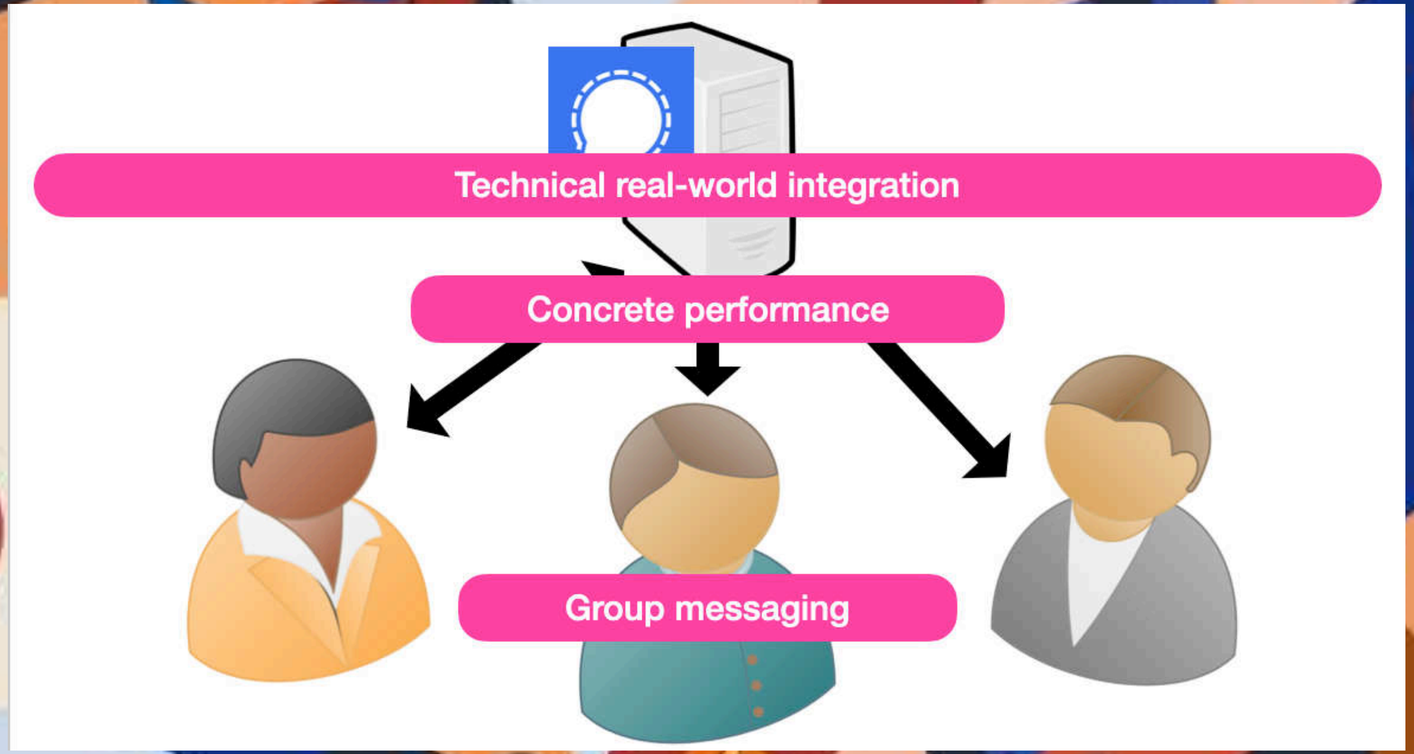
4. Summary and conclusions

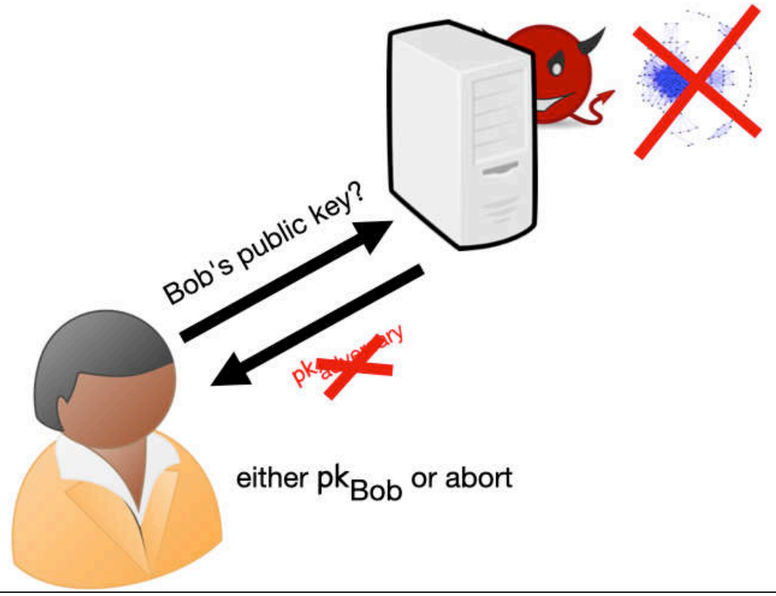
Summary of contributions

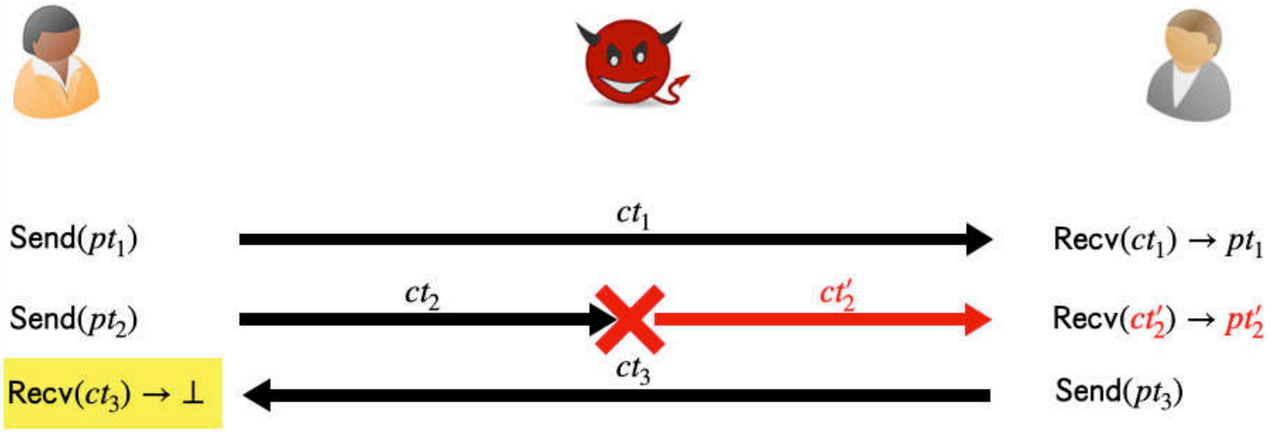
	Before	After
Metadata protection	Classic PIR without integrity protection	Authenticated PIR provides privacy and integrity
Active attack detection	Detection without immediate decryption, partial detection or additional rounds.	Efficient detection while supporting immediate decryption
Deniability	Only cryptographic deniability	Cryptographic deniability fails technically and legally: local messages modification can work

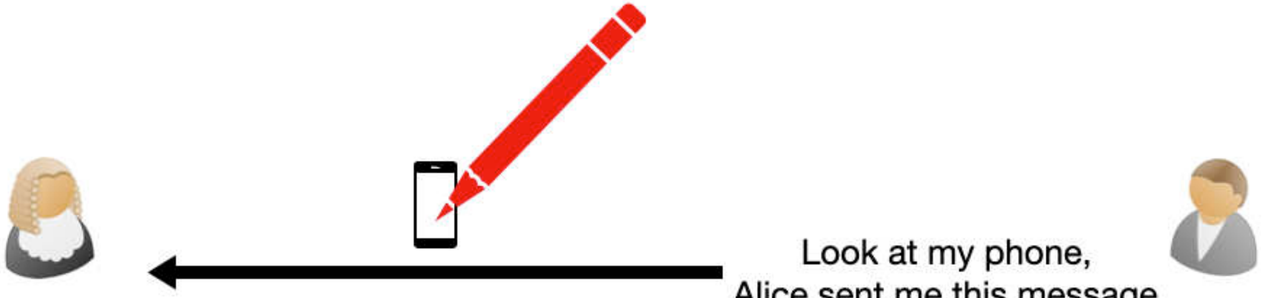

Technical real-world integration
Concrete performance
Group messaging

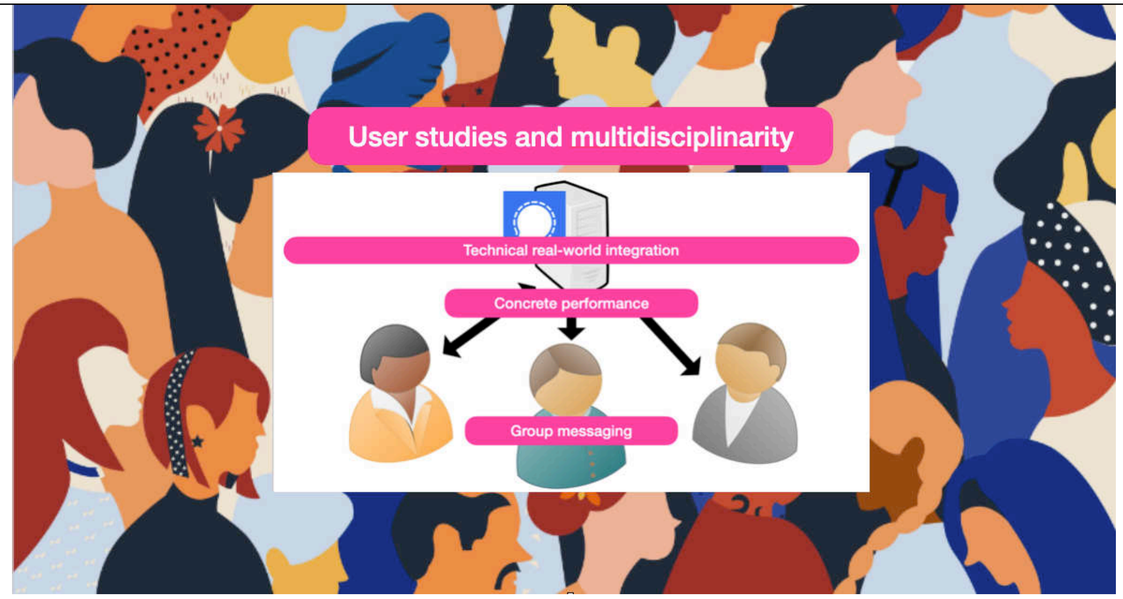
User studies and multidisciplinarity




Bob's public key?
pk adversary
either pk_{Bob} or abort


Send(pt_1)
ct_1
Recv(ct_1) → pt_1
Send(pt_2)
ct_2
ct_2'
Recv(ct_2') → pt_2'
Recv(ct_3) → ⊥
ct_3
Send(pt_3)


Look at my phone,
Alice sent me this message


User studies and multidisciplinarity
Technical real-world integration
Concrete performance
Group messaging