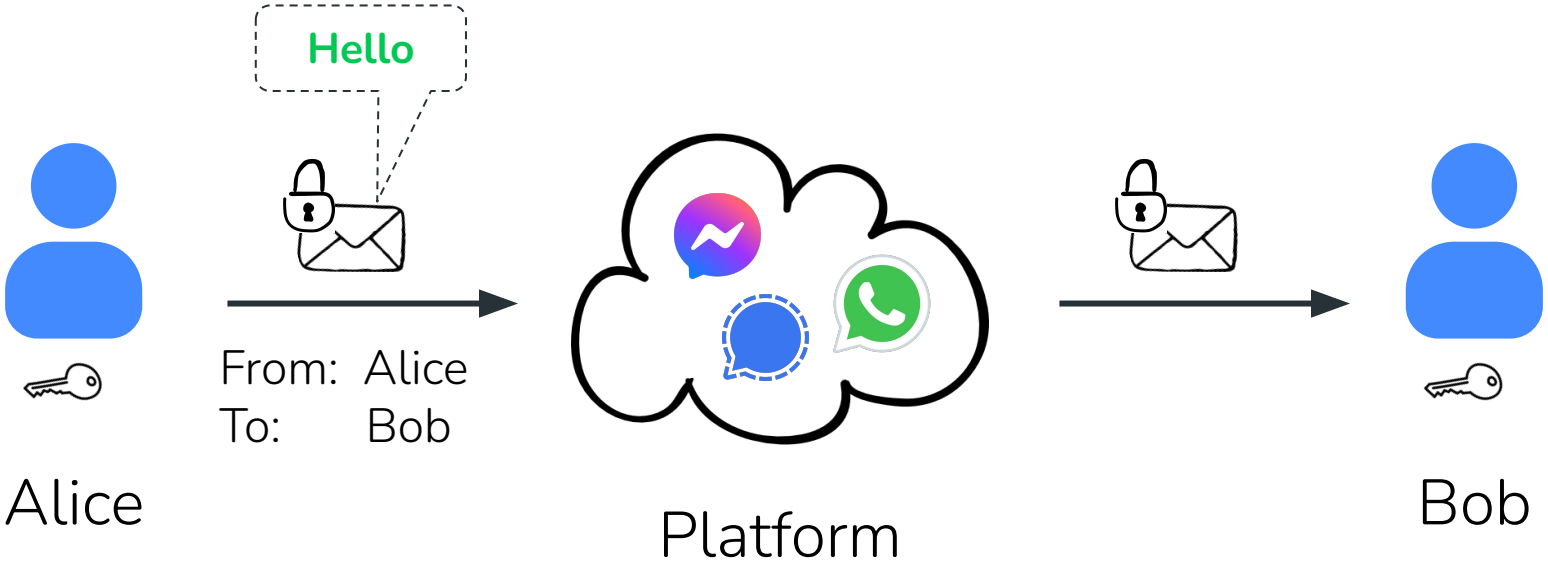
Orca: Blocklisting in Sender-Anonymous Messaging

Nirvan Tyagi Julia Len Ian Miers Tom Ristenpart

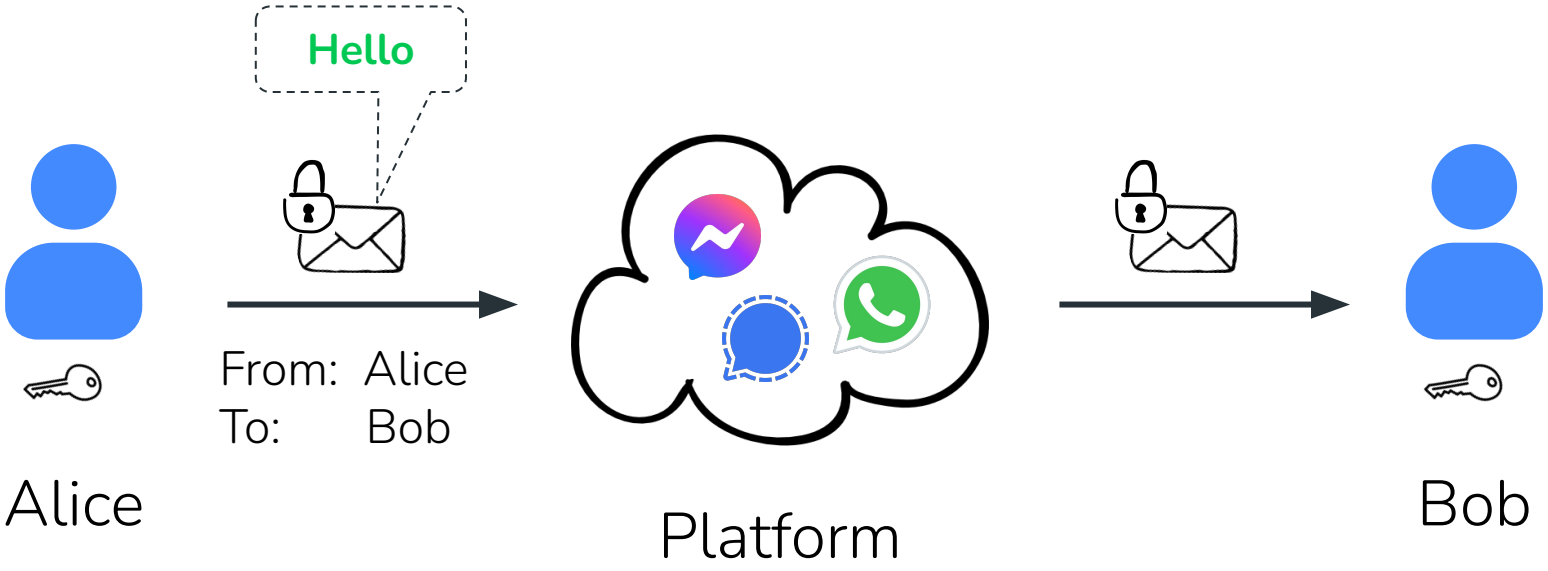
USENIX Security 2022

Setting: End-to-end encrypted messaging



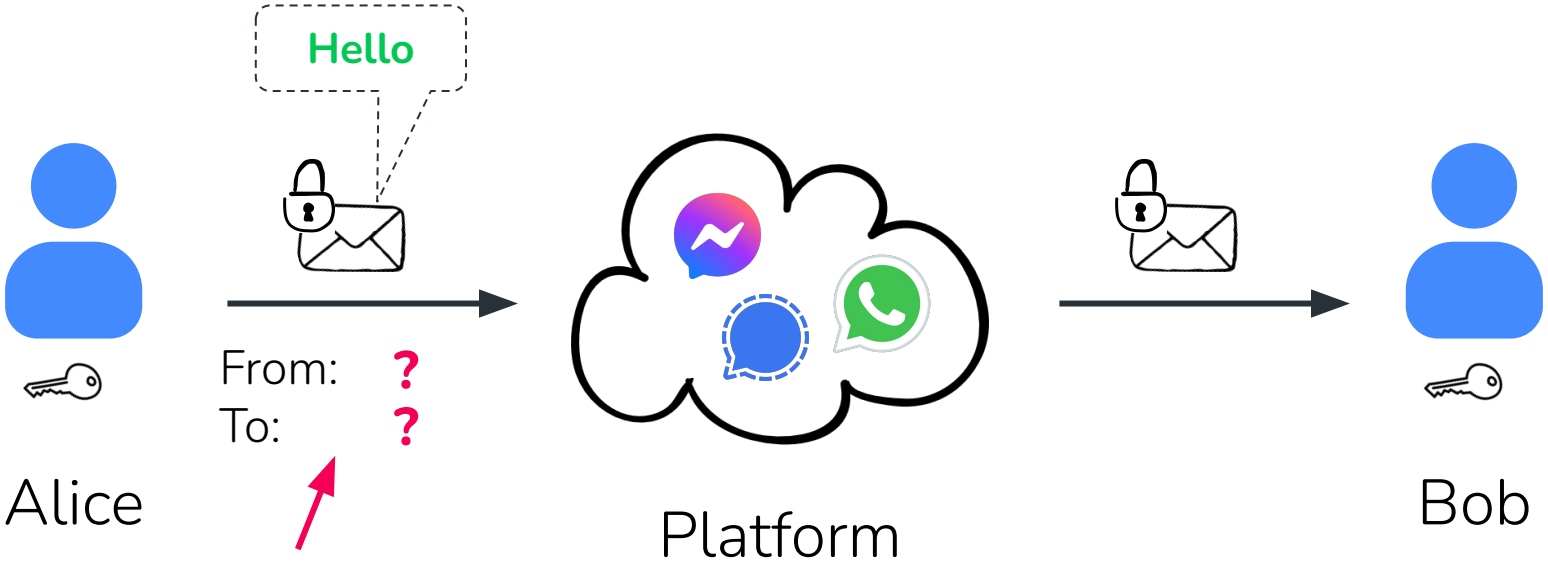
Setting: End-to-end encrypted messaging

- Goal: Confidentiality and Integrity



Setting: End-to-end encrypted messaging

- Goal: Confidentiality and Integrity
- Goal: Conversation participant **metadata privacy**



Metadata privacy is (relatively) expensive!

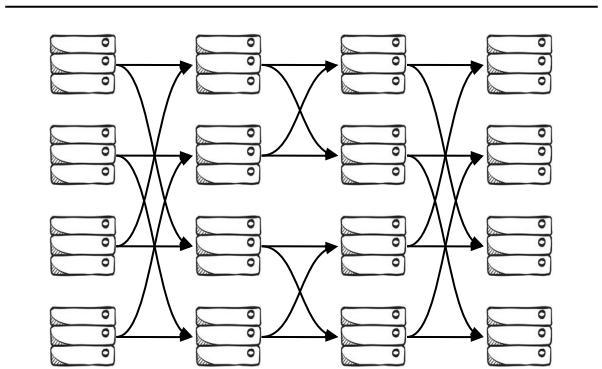
- 3 classes of approaches for metadata privacy of sender and recipient identity

Metadata privacy is (relatively) expensive!

- 3 classes of approaches for metadata privacy of sender and recipient identity

Mixnet

[Dissent OSDI'12]
[Vuvuzela SOSP'15]
[Stadium SOSP'17]
[Loopix USENIX Sec'17]
[Karaoke OSDI'18]

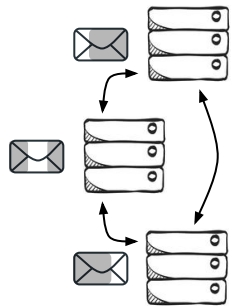


Metadata privacy is (relatively) expensive!

- 3 classes of approaches for metadata privacy of sender and recipient identity

Multiparty Computation (MPC)

[MCMix USENIX Sec'17]
[AsynchroMix CCS'19]
[Blinder CCS'20]
[Clarion NDSS'22]

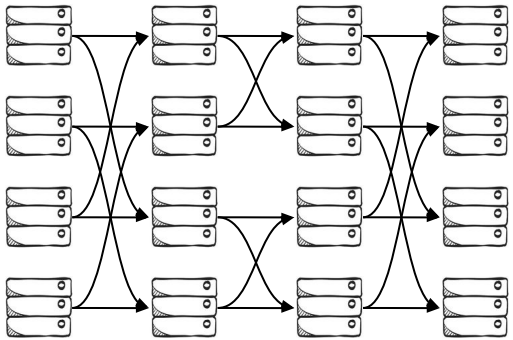


Alice

From: ?
To: ?

Mixnet

[Dissent OSDI'12]
[Vuvuzela SOSP'15]
[Stadium SOSP'17]
[Loopix USENIX Sec'17]
[Karaoke OSDI'18]

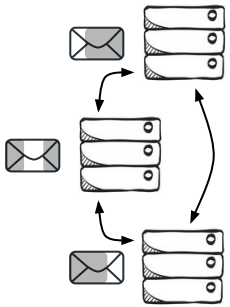


Bob

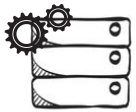
Metadata privacy is (relatively) expensive!

- 3 classes of approaches for metadata privacy of sender and recipient identity

Multiparty Computation (MPC)
[MCMix USENIX Sec'17]
[AsynchroMix CCS'19]
[Blinder CCS'20]
[Clarion NDSS'22]



Private Information Retrieval (PIR)
[Riposte S&P'15]
[Pung OSDI'16]
[Express USENIX Sec'21]

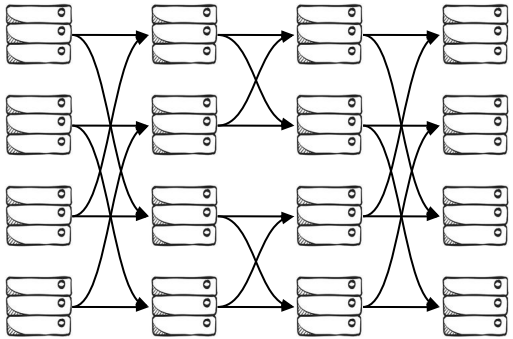


Alice

From: ?
To: ?

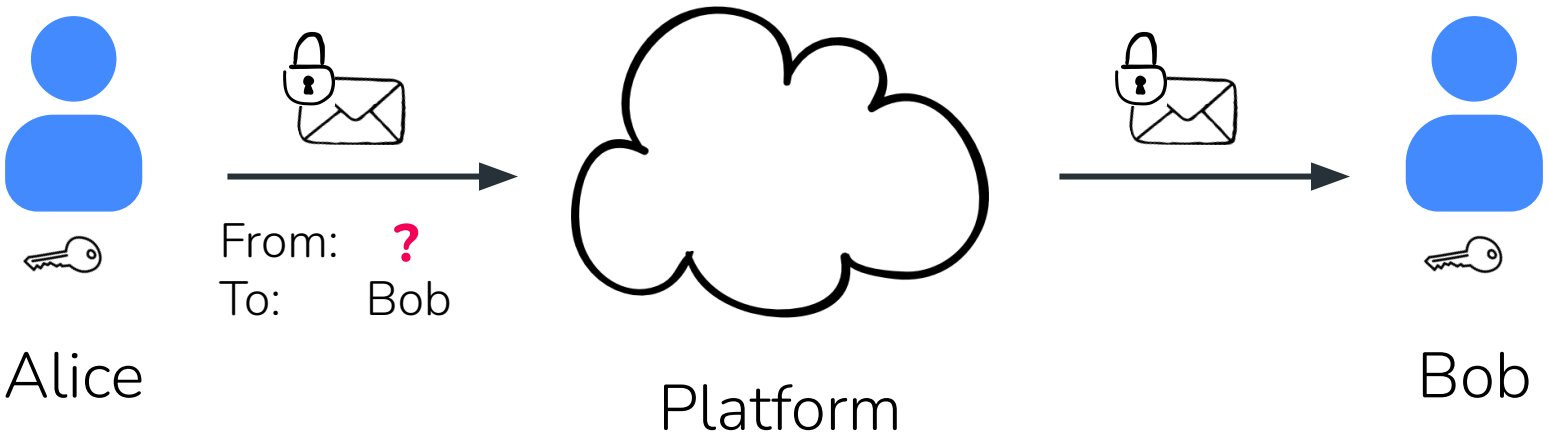
Bob

Mixnet
[Dissent OSDI'12]
[Vuvuzela SOSP'15]
[Stadium SOSP'17]
[Loopix USENIX Sec'17]
[Karaoke OSDI'18]



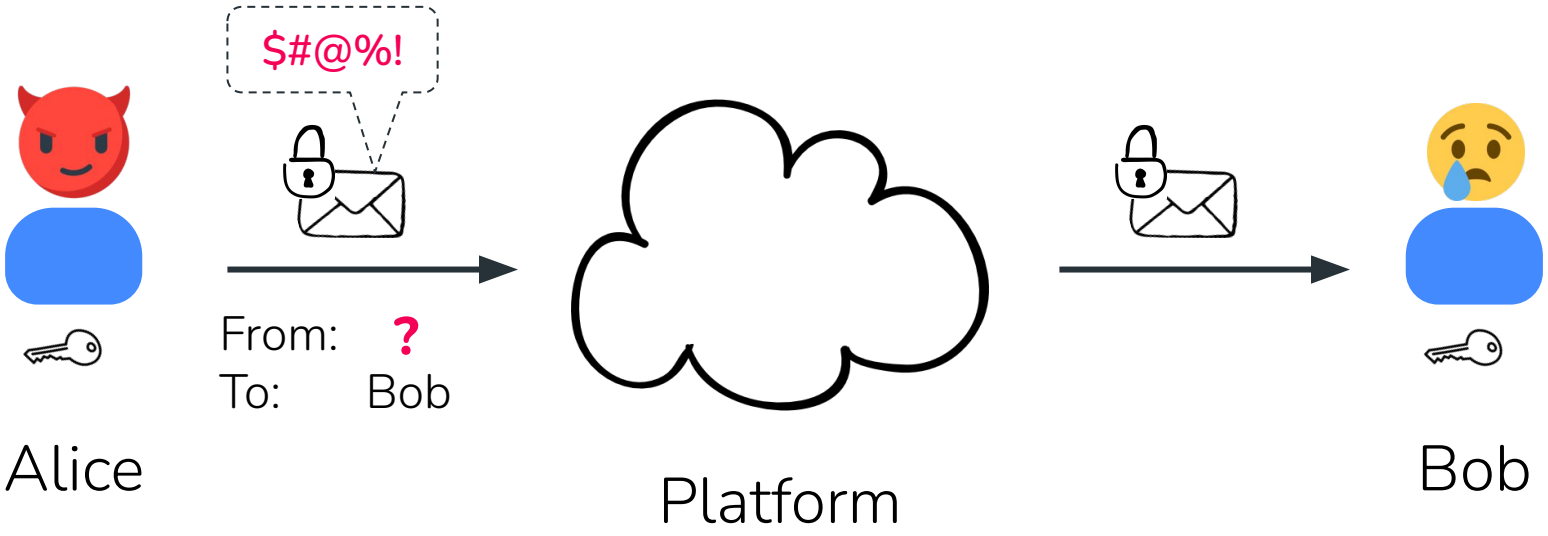
Signal's Sealed Sender: Relaxed metadata privacy

- New messaging protocol released by Signal in 2018
- Focuses on metadata privacy of only sender identity



Problem: Platform unable to filter by sender

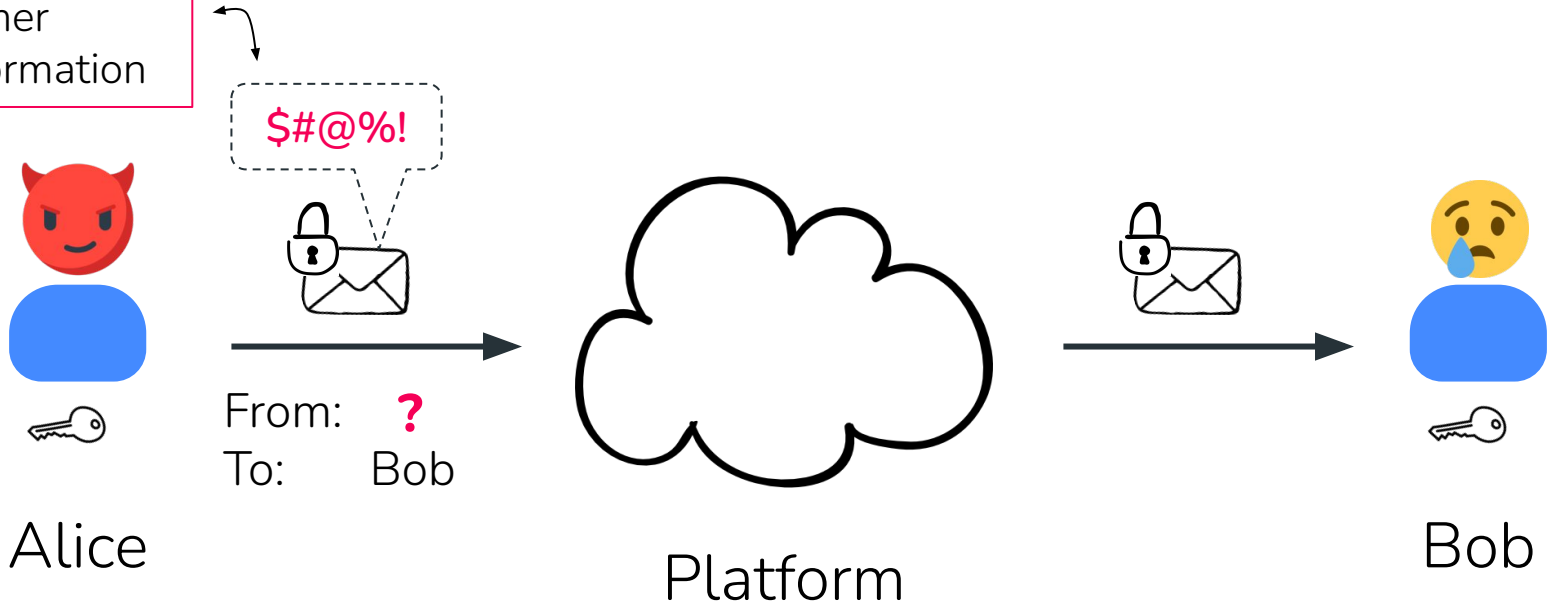
- Modern messaging platforms are expected to perform various message filtering tasks on behalf of the recipient client (e.g., blocking spam / abuse)



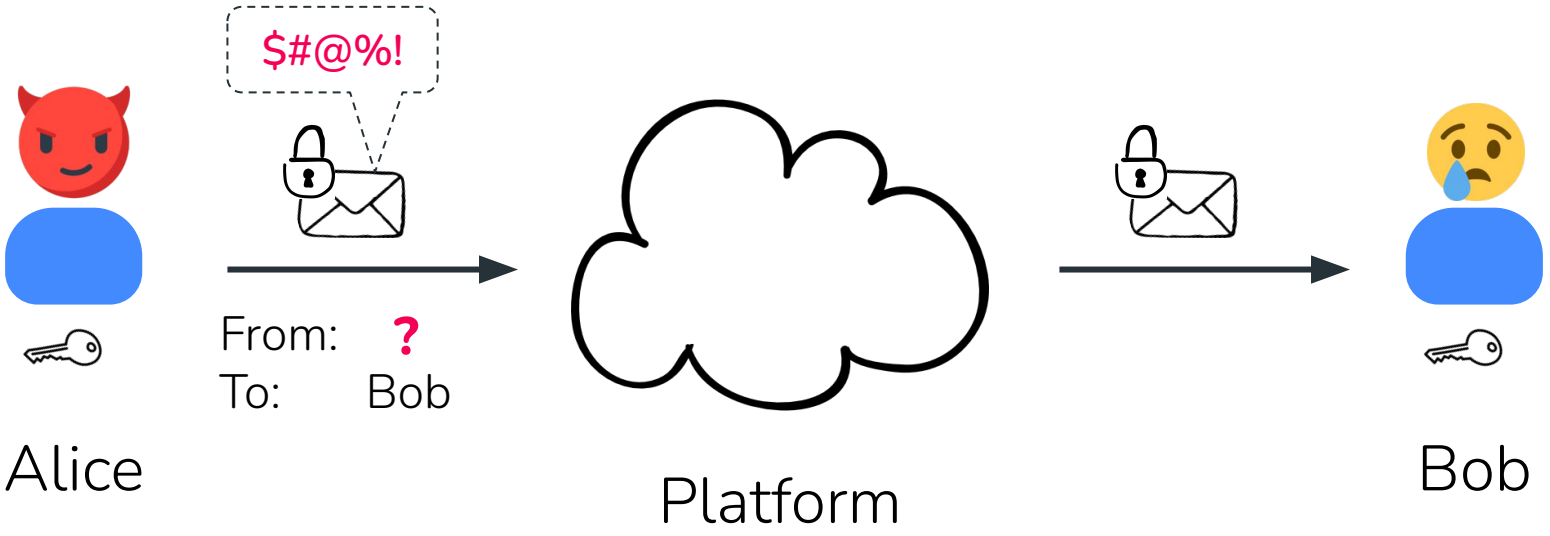
Problem: Platform unable to filter by sender

- Modern messaging platforms are expected to perform various message filtering tasks on behalf of the recipient client (e.g., blocking spam / abuse)

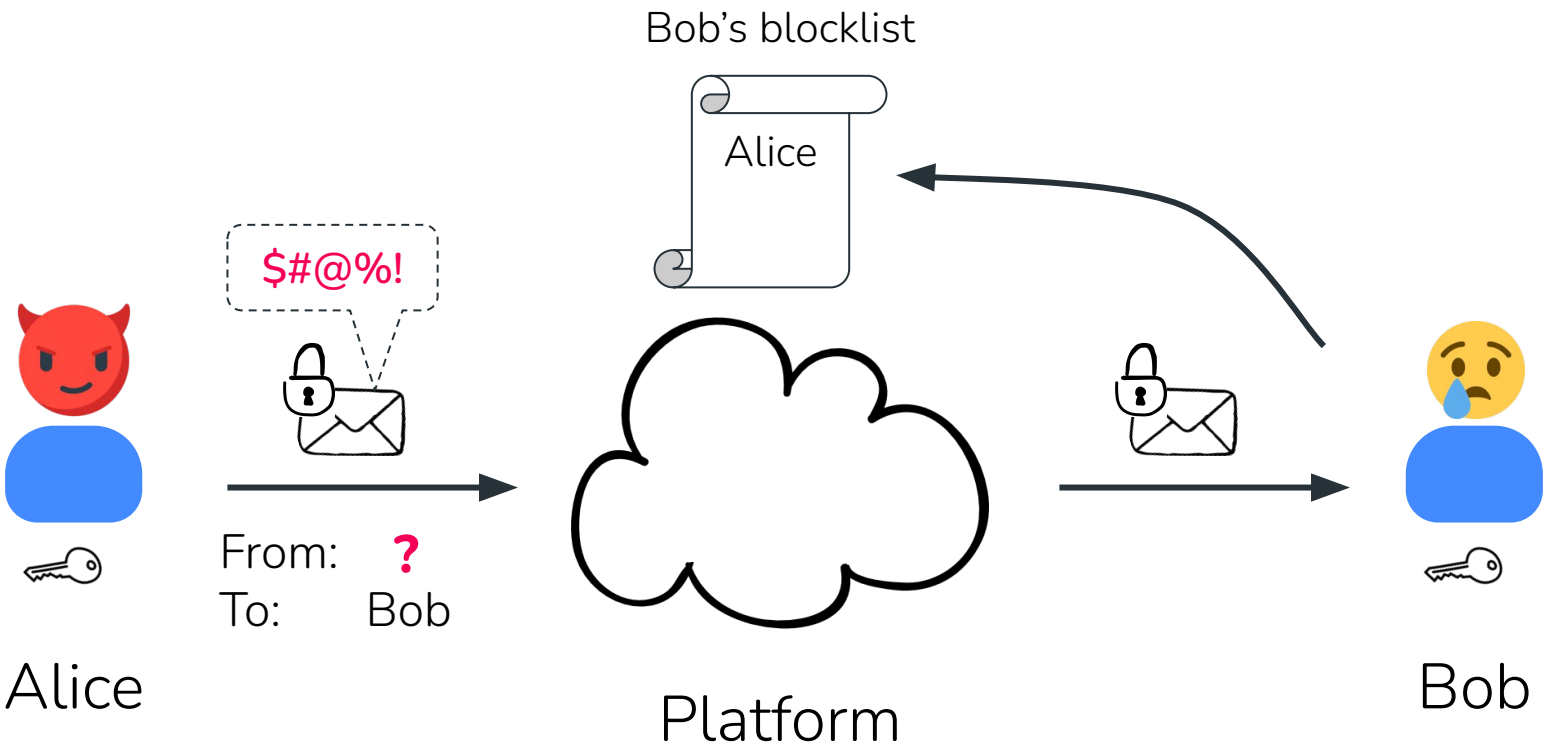
Online bully
Spammer
Misinformation



Abuse mitigation: Sender blocklists

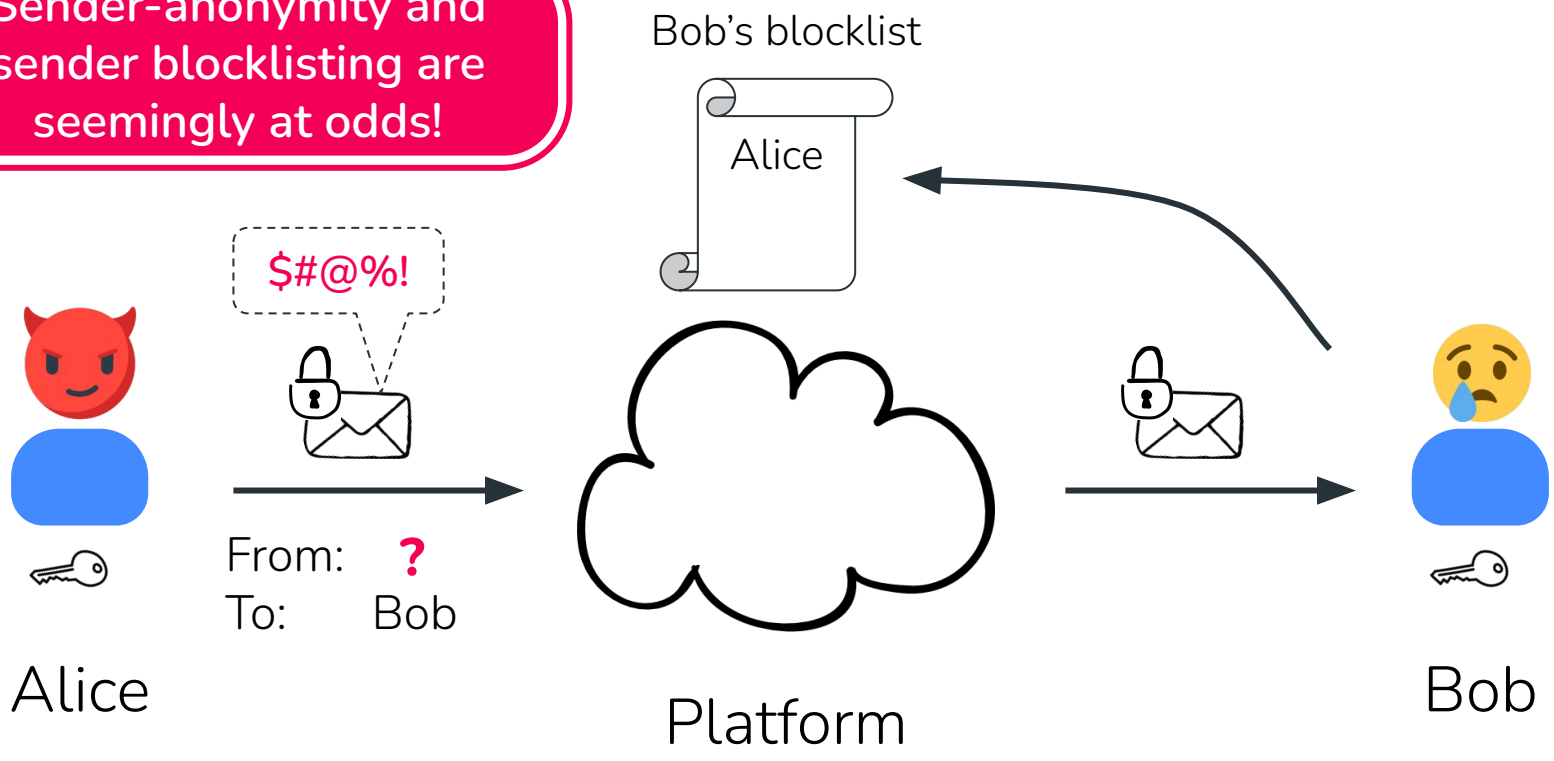


Abuse mitigation: Sender blocklists



Abuse mitigation: Sender blocklists

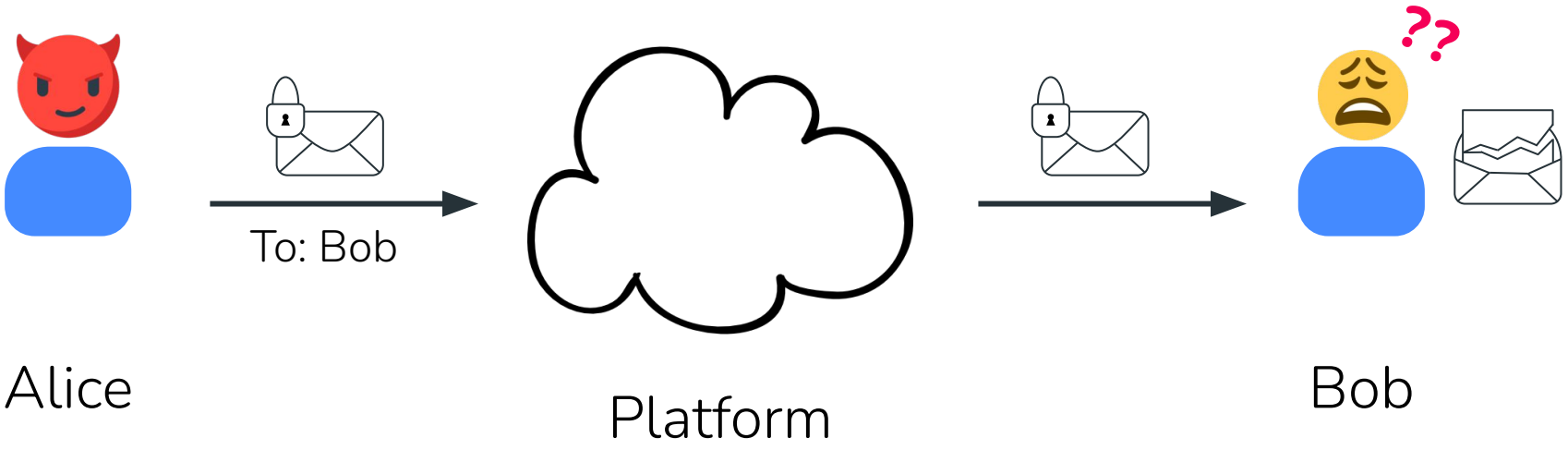
Sender-anonymity and sender blocklisting are seemingly at odds!



Outline

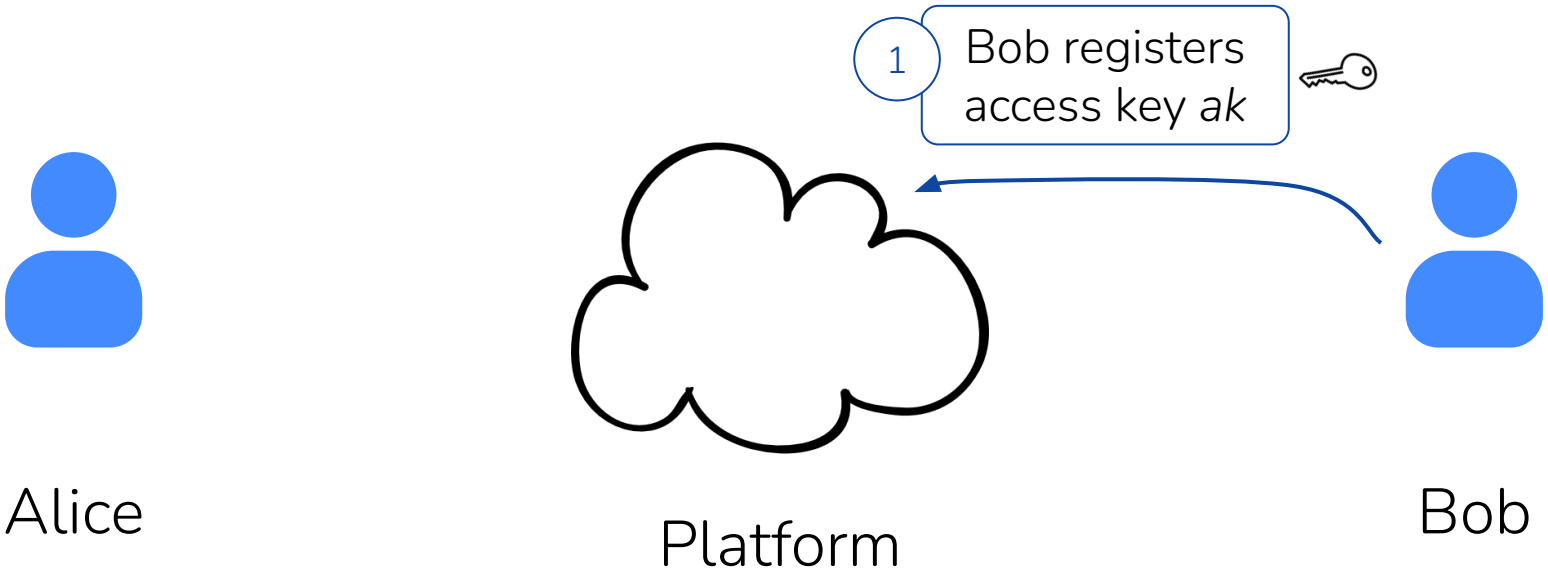
- **Contribution**: Blocklisting for sender-anonymous messaging
- Identifying weaknesses in Signal's sealed sender protocol
  - Requires non-sender-anonymous communication to initialize
  - Admits untraceable battery-draining (griefing) attack
- Orca: a sender-anonymous blocklisting protocol
  - Group signature scheme for sender-anonymous initialization
  - Efficient one-time-use authentication tokens from algebraic MACs

Background: Sealed Sender



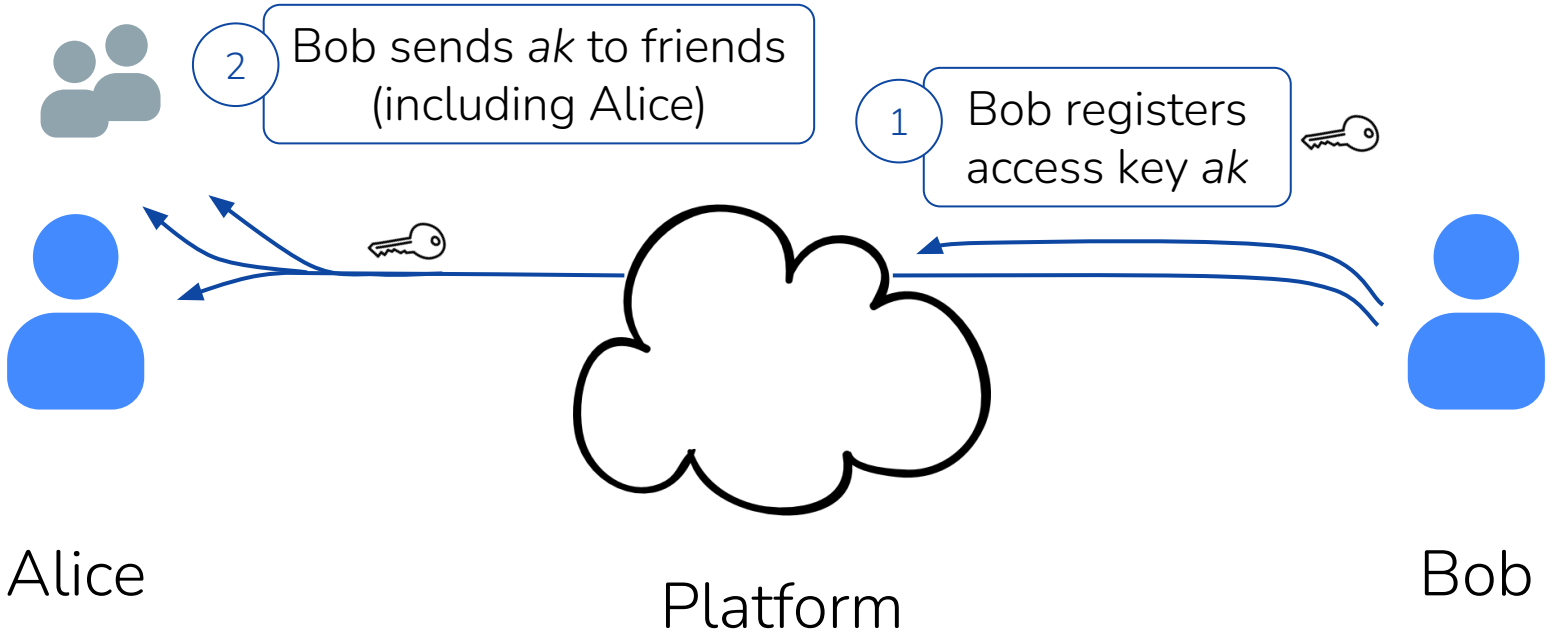
Background: Sealed Sender

- Sealed Sender protects recipients by requiring sender to show recipient's "access key"



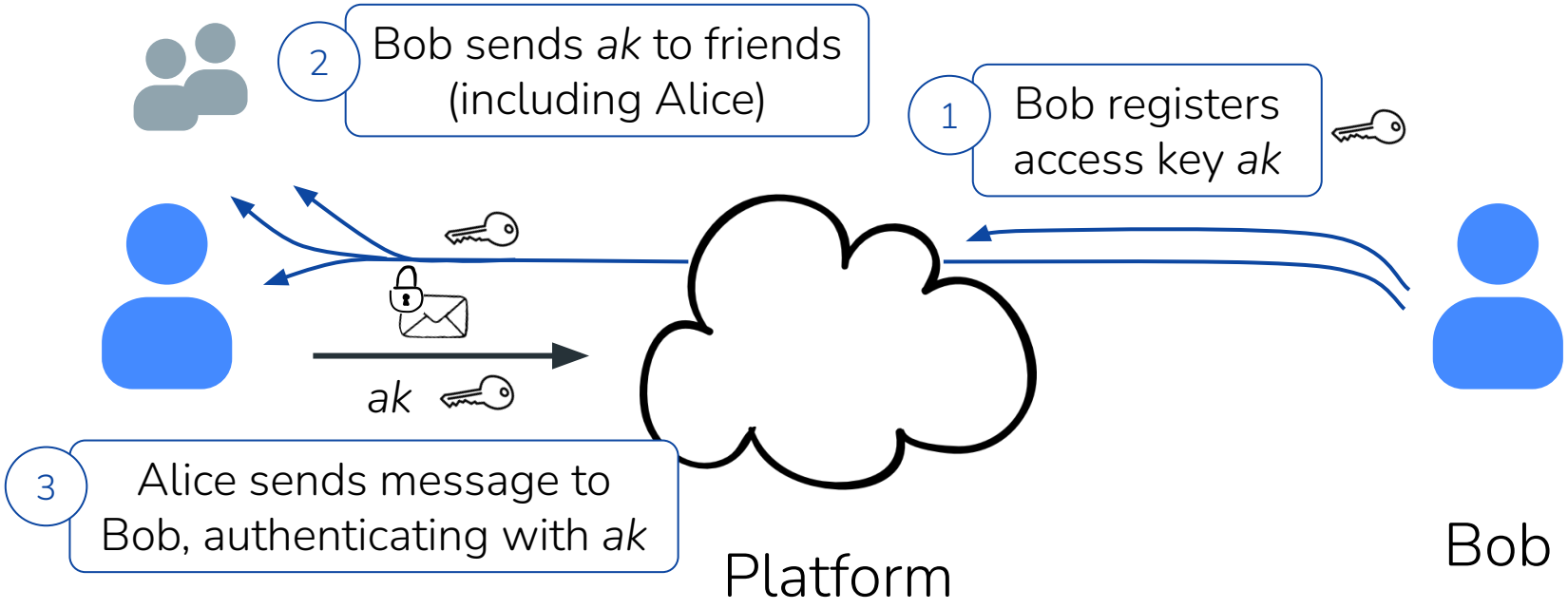
Background: Sealed Sender

- Sealed Sender protects recipients by requiring sender to show recipient's "access key"



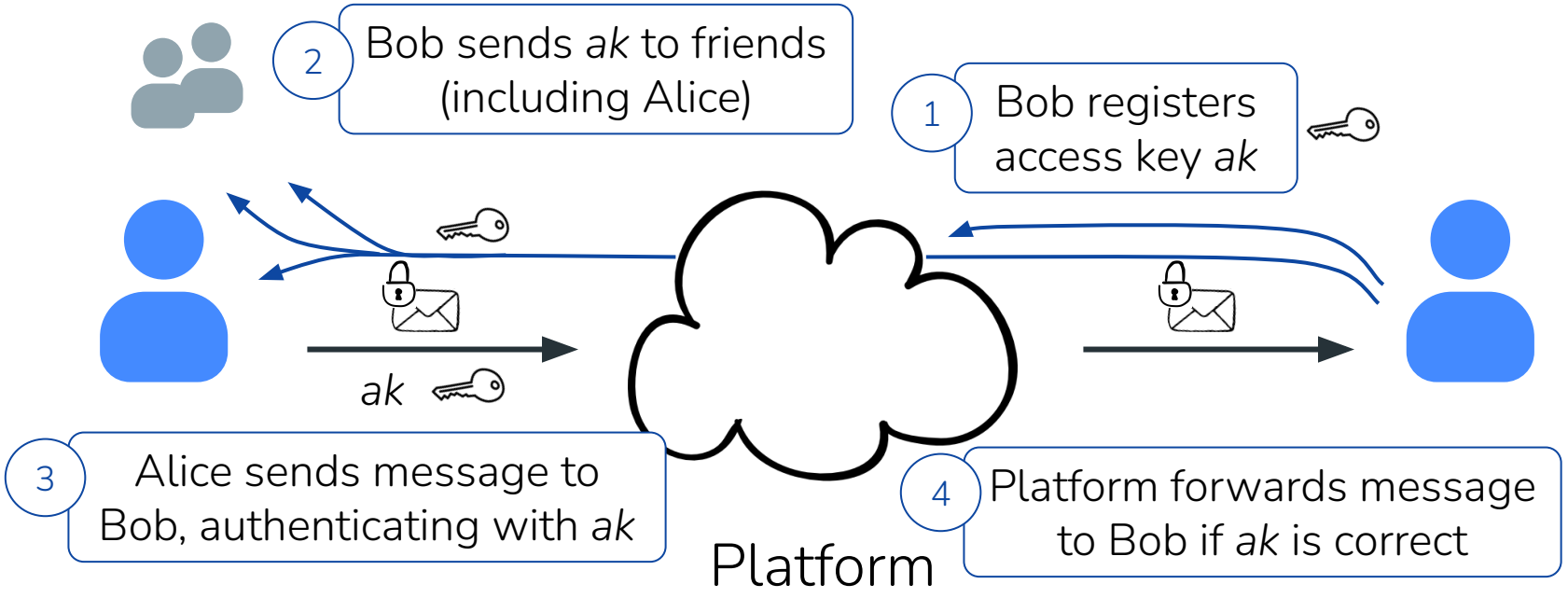
Background: Sealed Sender

- Sealed Sender protects recipients by requiring sender to show recipient's "access key"



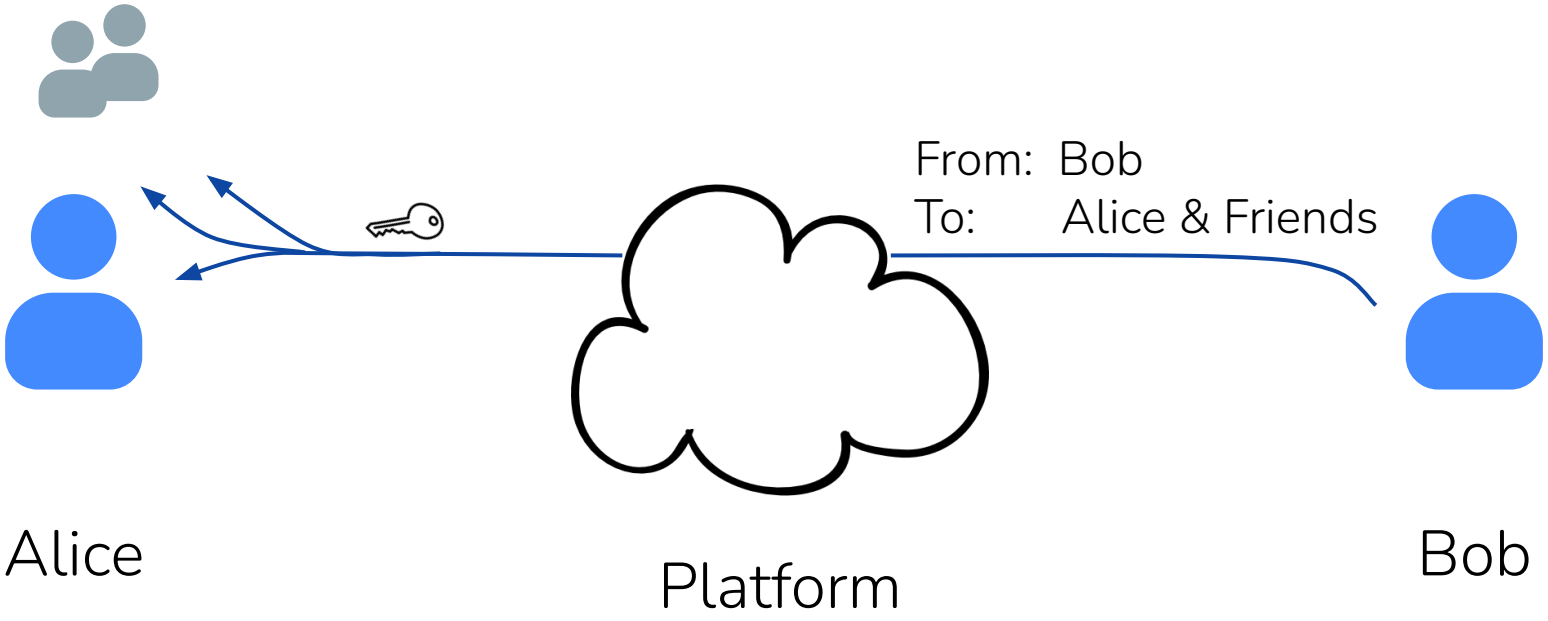
Background: Sealed Sender

- Sealed Sender protects recipients by requiring sender to show recipient's "access key"



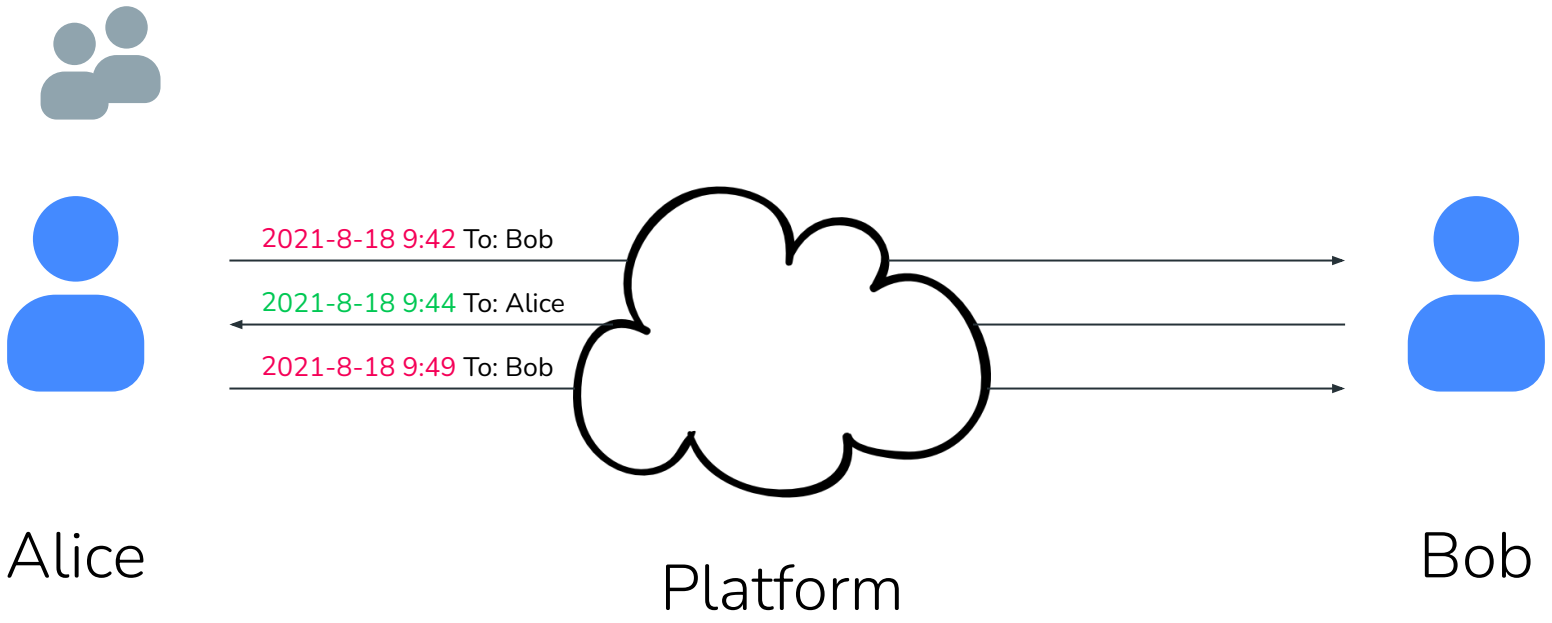
Weaknesses in Sealed Sender design

1. Initial distribution of access key done over non-sender-anonymous channel



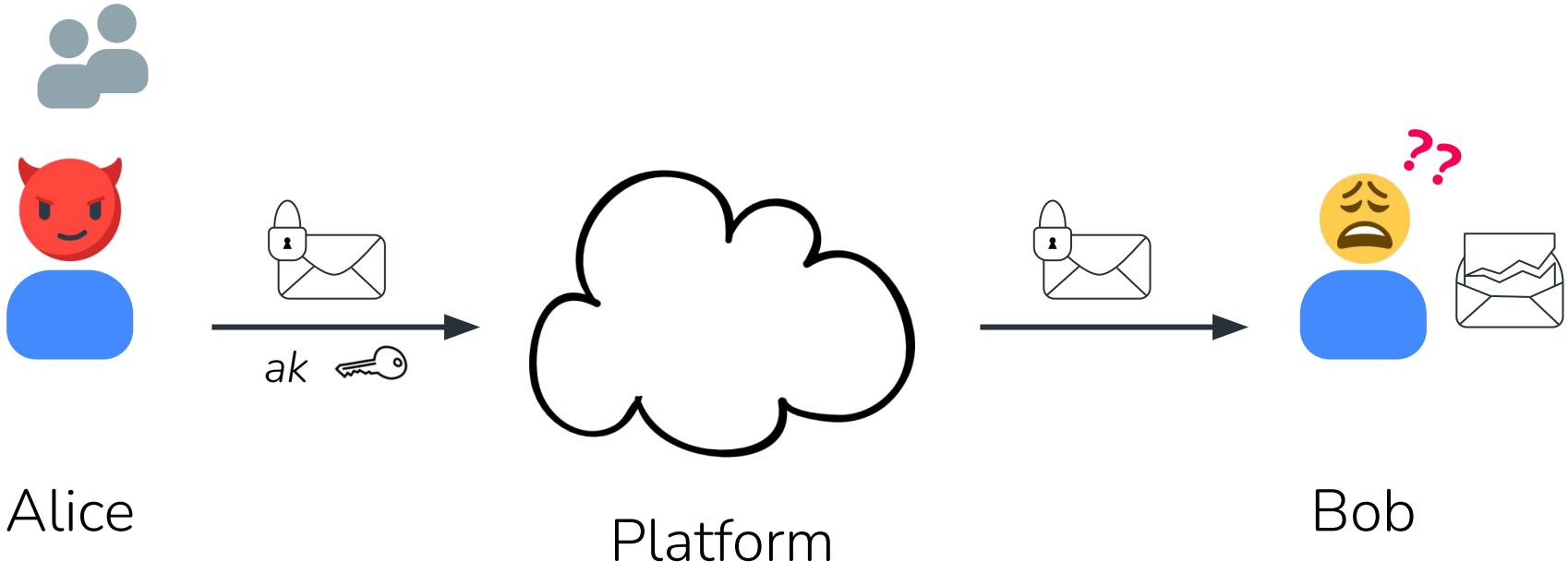
Weaknesses in Sealed Sender design

1. Initial distribution of access key done over non-sender-anonymous channel
2. Timing of sender-anonymous messages can leak conversation patterns



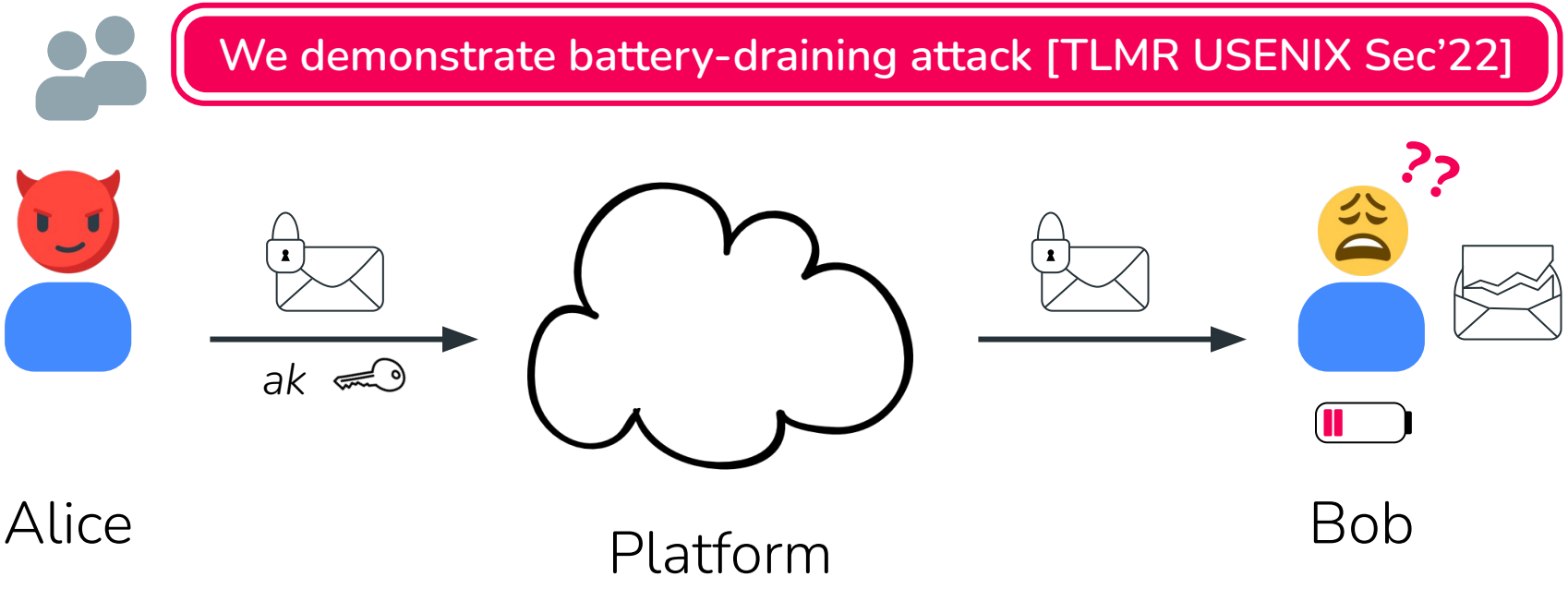
Weaknesses in Sealed Sender design

1. Initial distribution of access key done over non-sender-anonymous channel
2. Timing of sender-anonymous messages can leak conversation patterns
3. Untraceable griefing attack mountable by compromised (or malicious) friend



Weaknesses in Sealed Sender design

1. Initial distribution of access key done over non-sender-anonymous channel
2. Timing of sender-anonymous messages can leak conversation patterns
3. Untraceable griefing attack mountable by compromised (or malicious) friend



Weaknesses in Sealed Sender design

1. Initial distribution of access key done over non-sender-anonymous channel
2. Timing of sender-anonymous messages can leak conversation patterns
3. Untraceable griefing attack mountable by compromised (or malicious) friend

Weaknesses in Sealed Sender design

1. Initial distribution of access key done over non-sender-anonymous channel
2. Timing of sender-anonymous messages can leak conversation patterns
3. Untraceable griefing attack mountable by compromised (or malicious) friend

Orca [TLMR USENIX Sec'22] addresses (1) & (3)

Weaknesses in Sealed Sender design

1. Initial distribution of access key done over non-sender-anonymous channel
2. Timing of sender-anonymous messages can leak conversation patterns
3. Untraceable griefing attack mountable by compromised (or malicious) friend

Orca [TLMR USENIX Sec'22] addresses (1) & (3)

[MKARW NDSS'21] evaluates (2) and proposes some partial countermeasures that are compatible with Orca

Outline

- **Contribution**: Blocklisting for sender-anonymous messaging
- Identifying weaknesses in Signal's sealed sender protocol
  - Requires non-sender-anonymous communication to initialize
  - Admits untraceable battery-draining (griefing) attack
- Orca: a sender-anonymous blocklisting protocol
  - Group signature scheme for sender-anonymous initialization
  - Efficient one-time-use authentication tokens from algebraic MACs

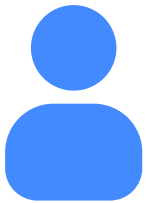
Building block: Group signatures

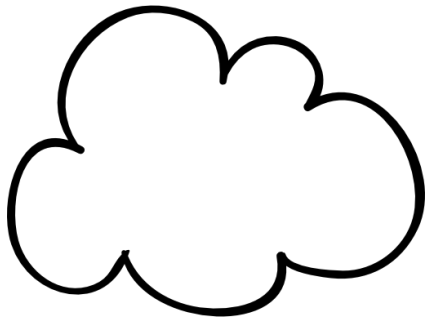
[CvH EUROCRYPT'91]

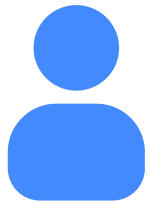
- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke

Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke







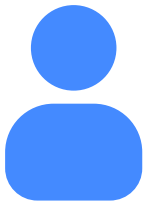
Alice

Platform

Bob

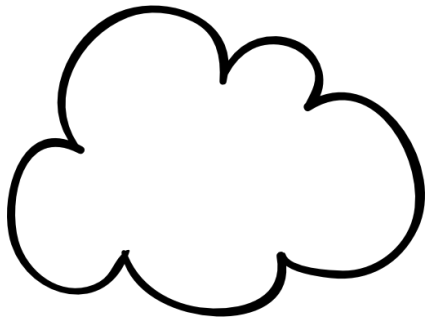
Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke



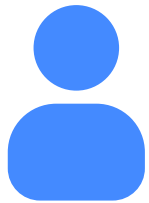
Alice

Group member



Platform

Group manager



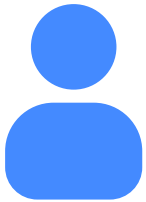
Bob

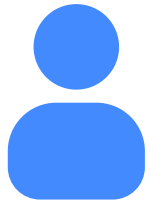
Opening authority

Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke

1 Users register with platform as both group members and opening authorities





Alice
Group member

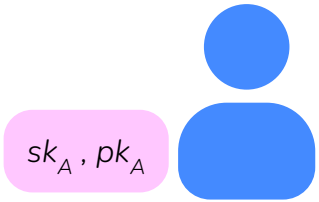
Platform
Group manager

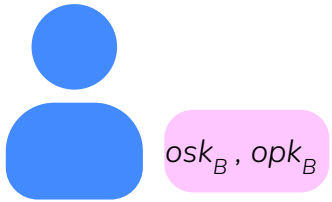
Bob
Opening authority

Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke

1 Users register with platform as both group members and opening authorities





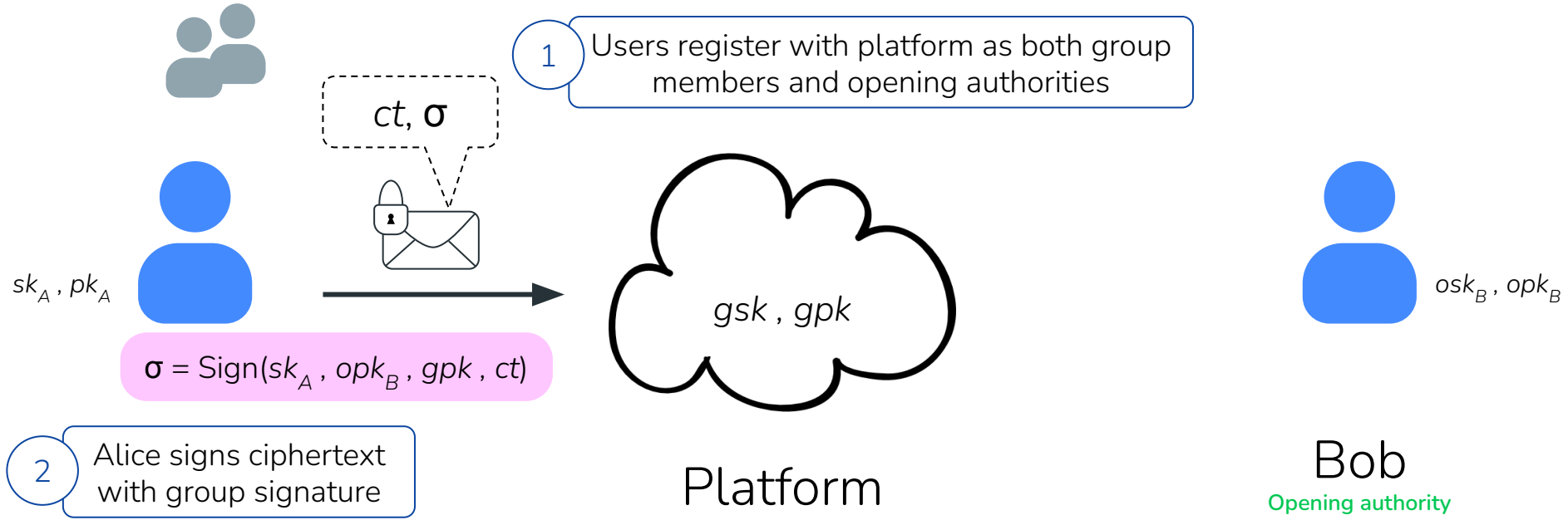
Alice
Group member

Platform
Group manager

Bob
Opening authority

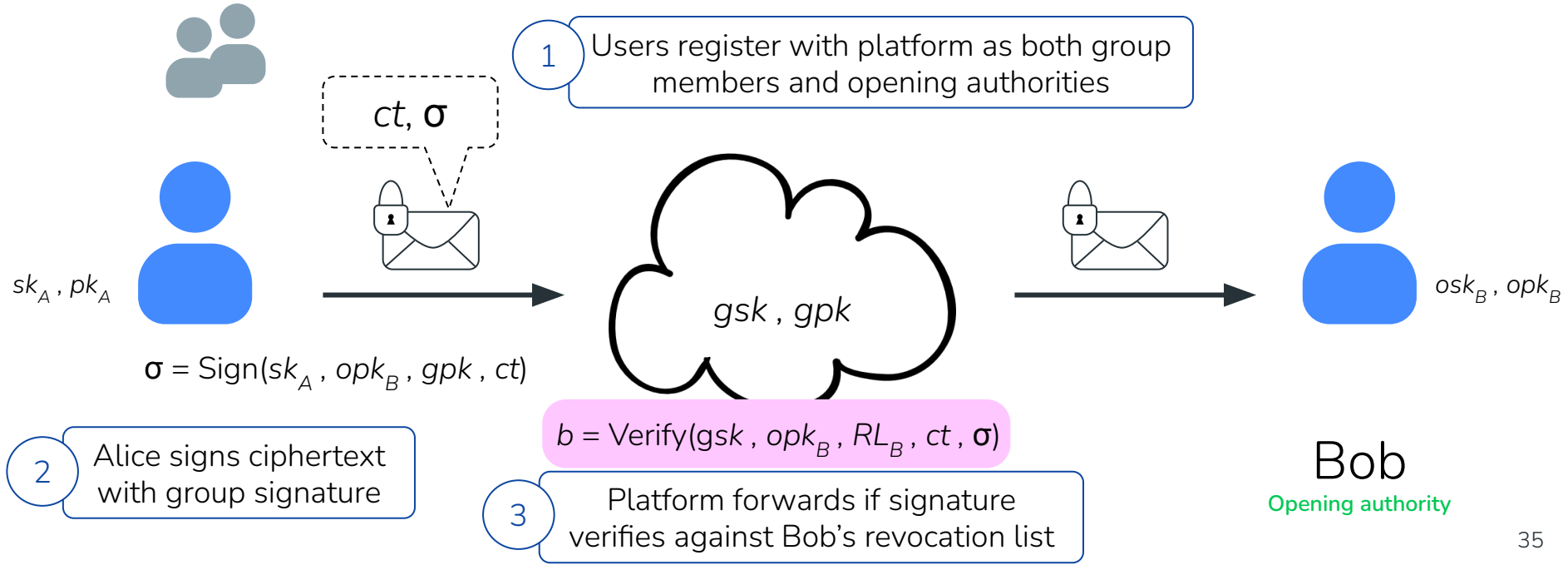
Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke



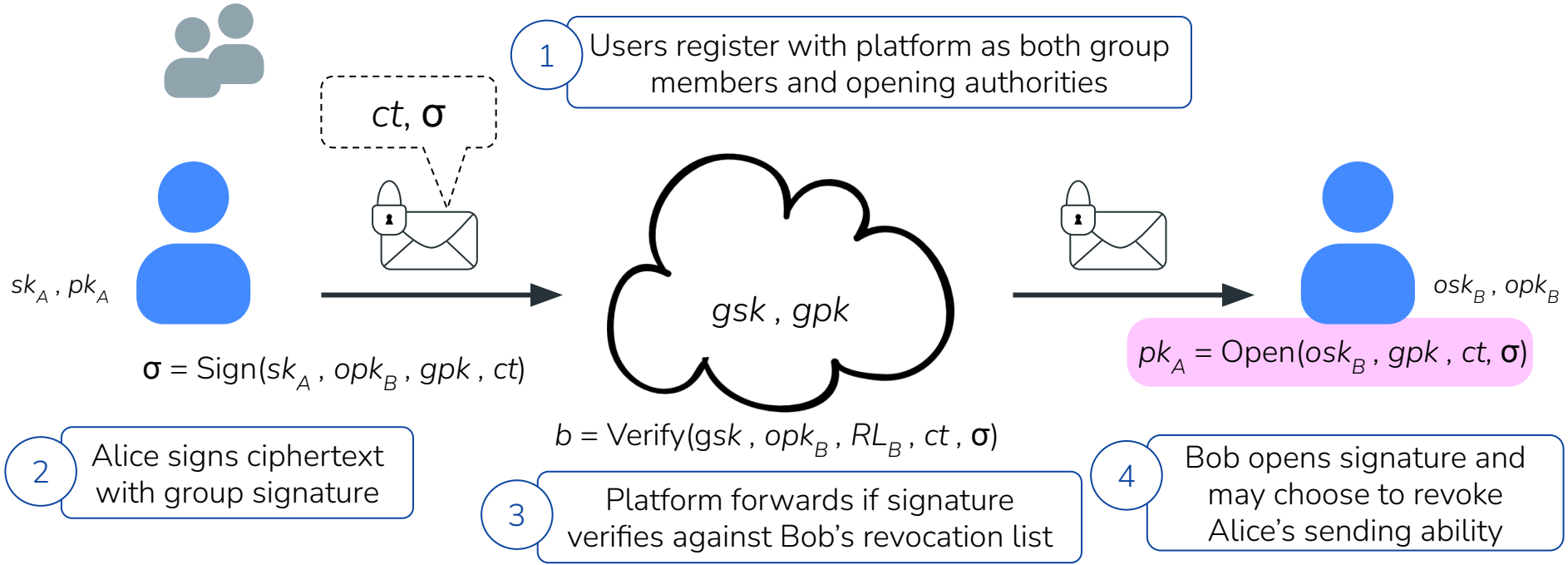
Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke



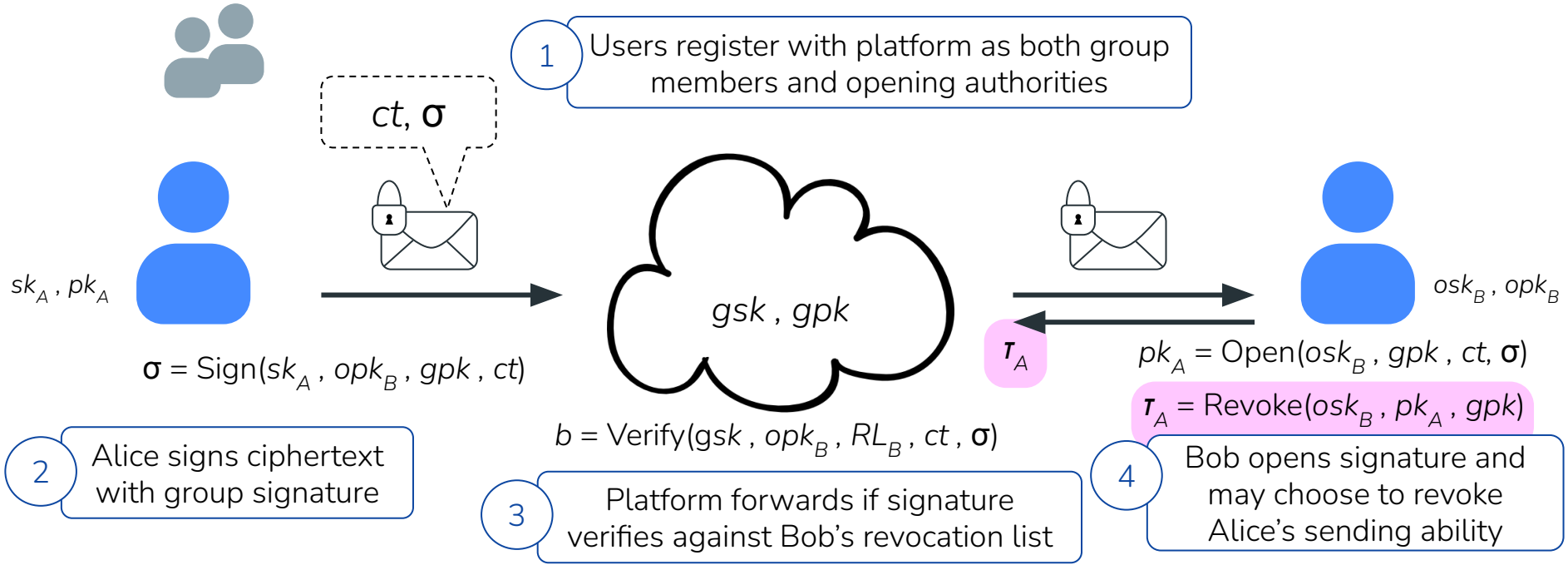
Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke

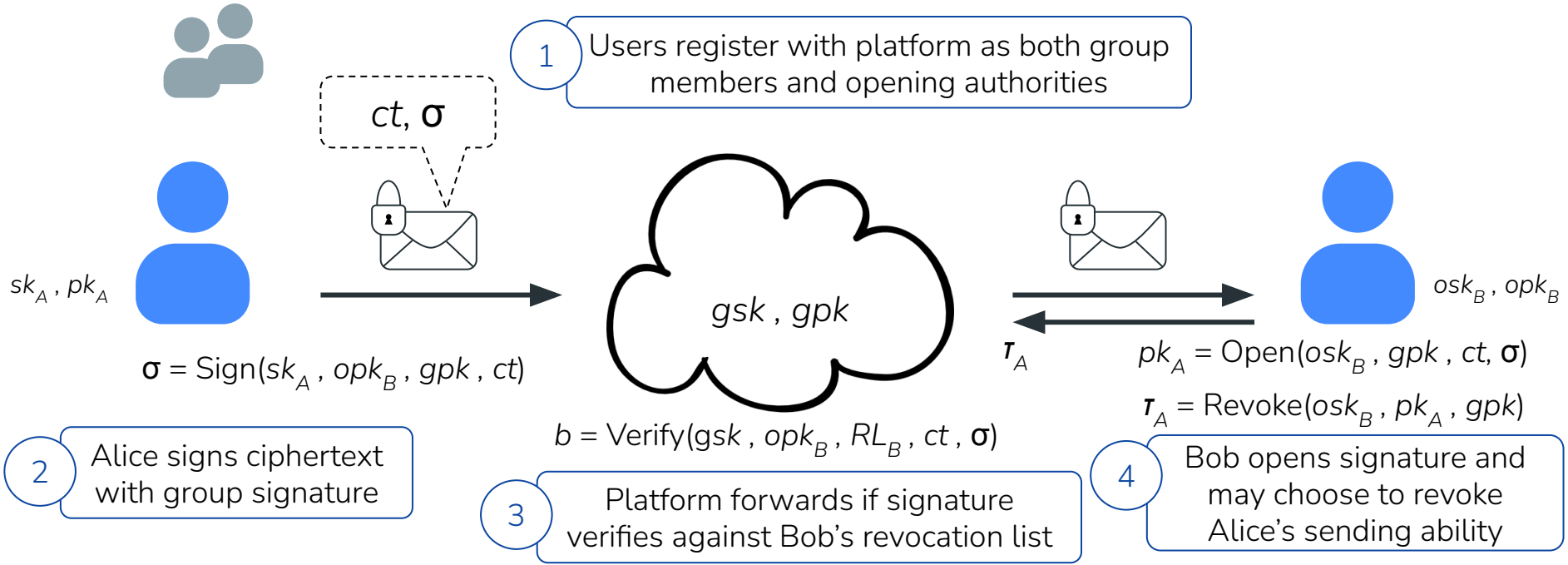


Building block: Group signatures

- **Group manager** manages membership of group
- **Group members** can sign messages anonymously on behalf of the group
- **Opening authority** can open group signature to learn identity of signer, and revoke

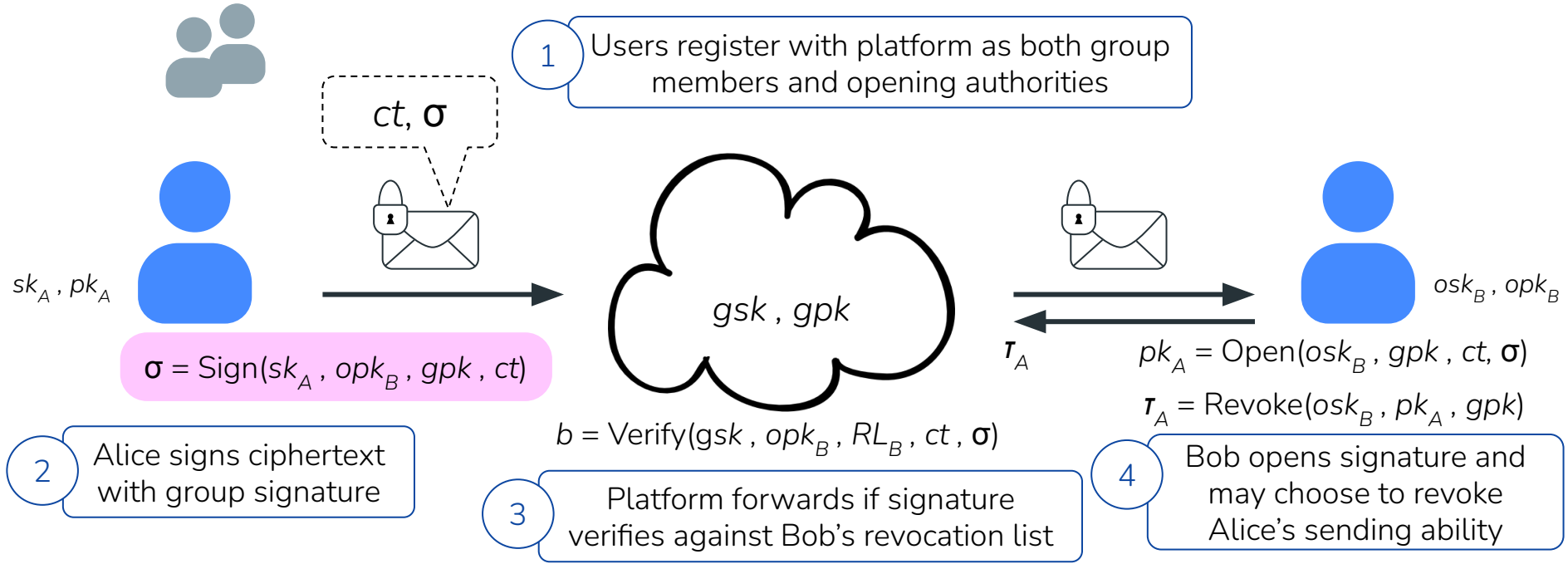


Building block: Group signatures



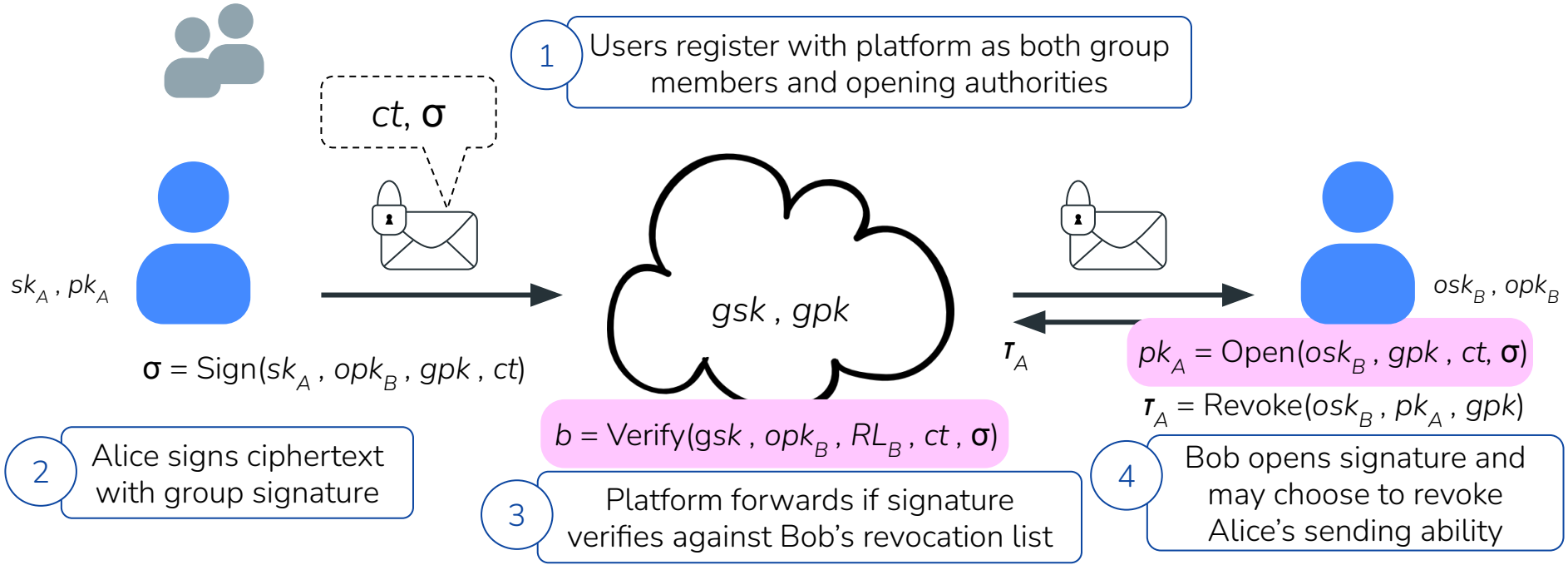
Building block: Group signatures

- Group signatures are **anonymous**



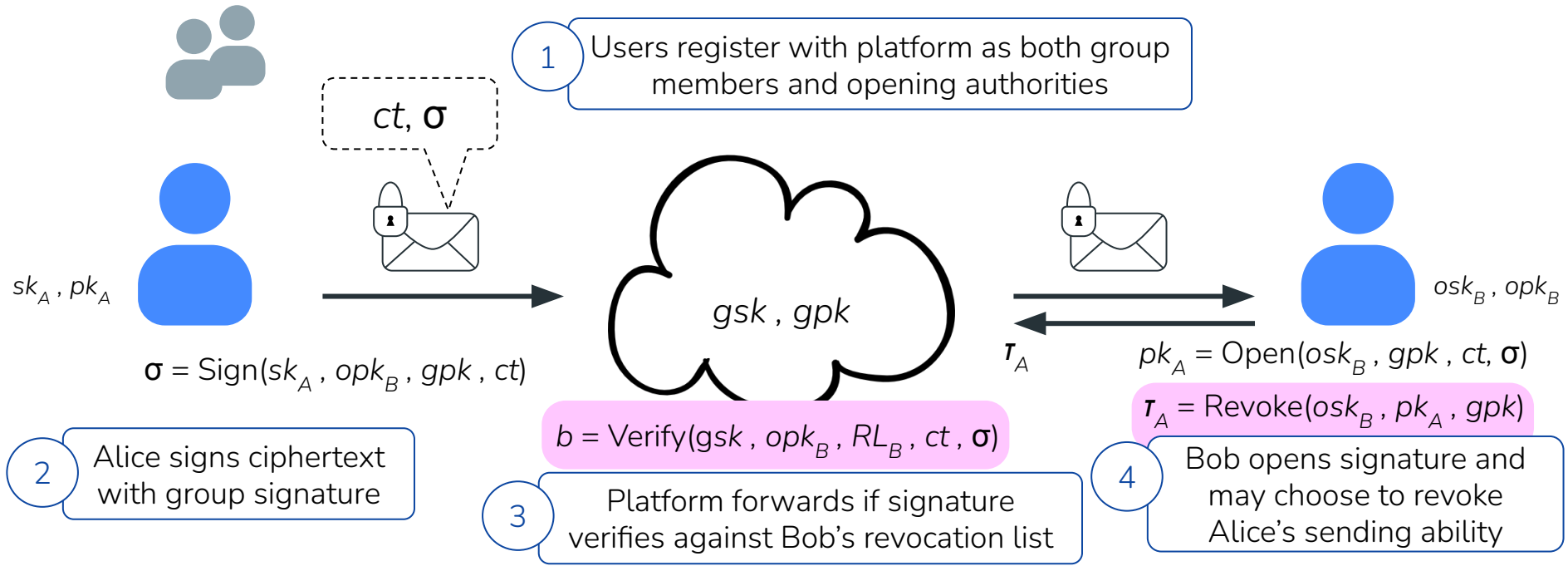
Building block: Group signatures

- Group signatures are anonymous, **traceable**



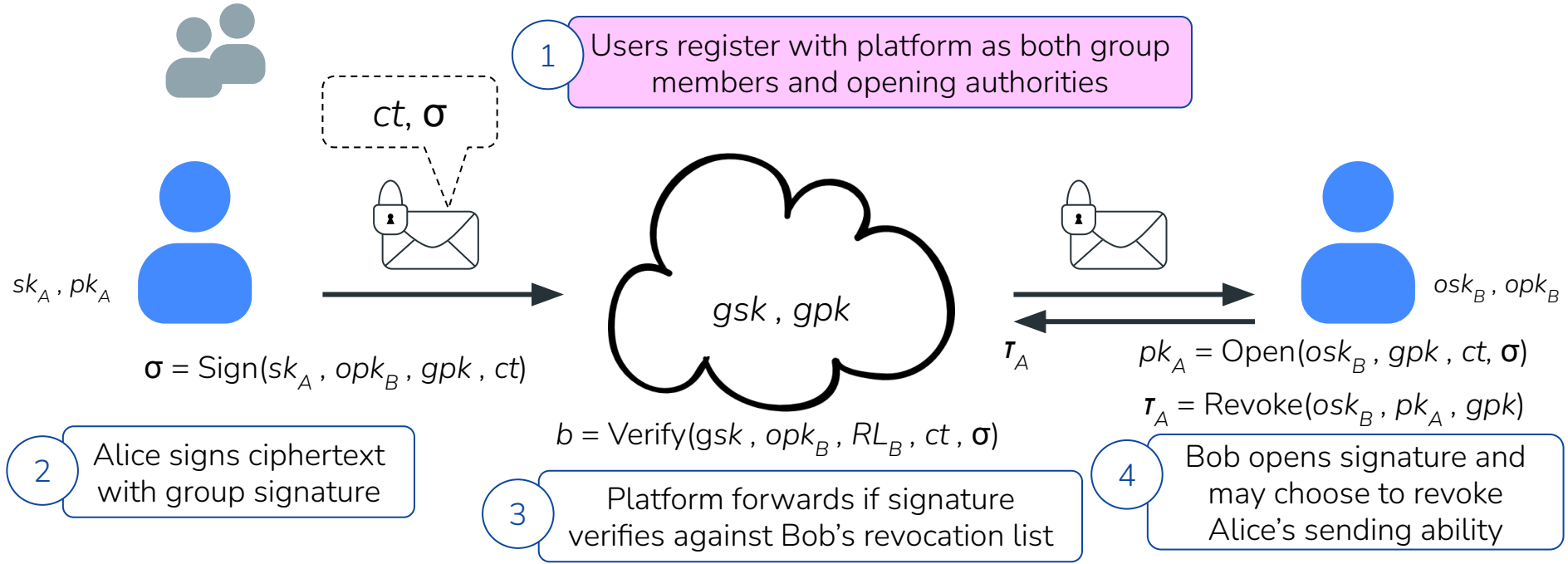
Building block: Group signatures

- Group signatures are anonymous, traceable, and **revocable**



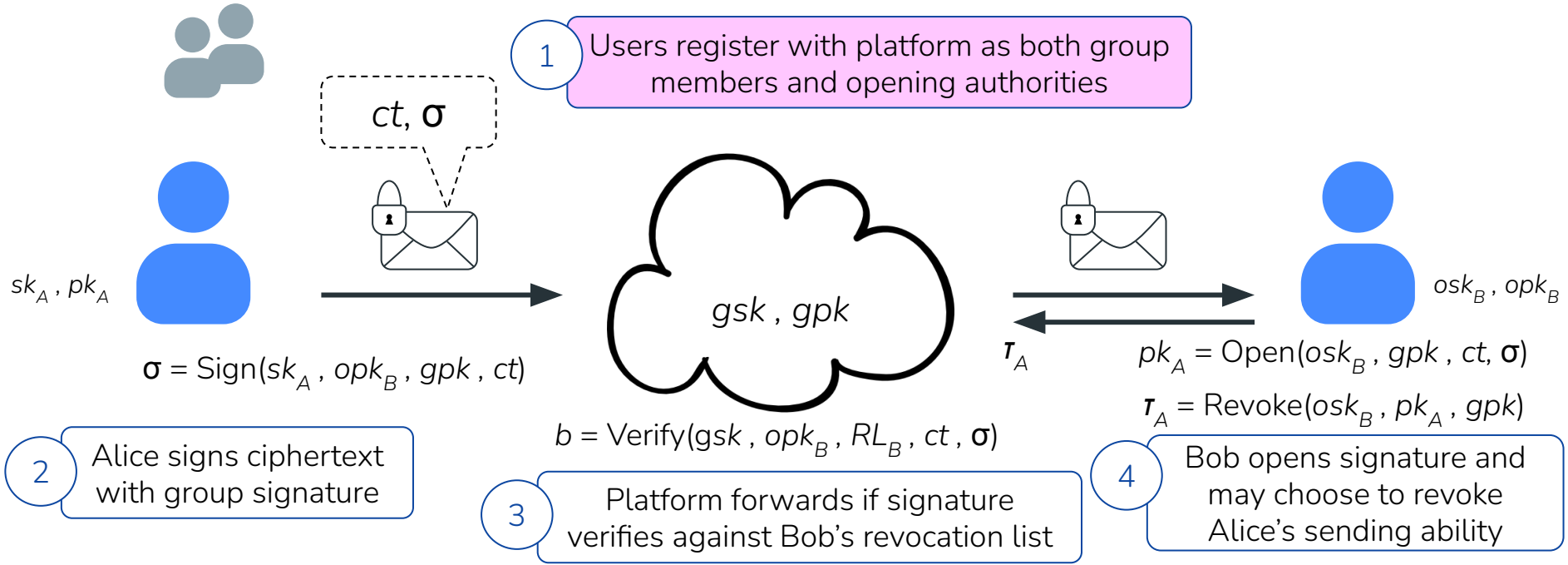
Building block: Group signatures

- Group signatures are anonymous, traceable, and revocable
- Group signatures do not require initialization over non-sender-anonymous channels



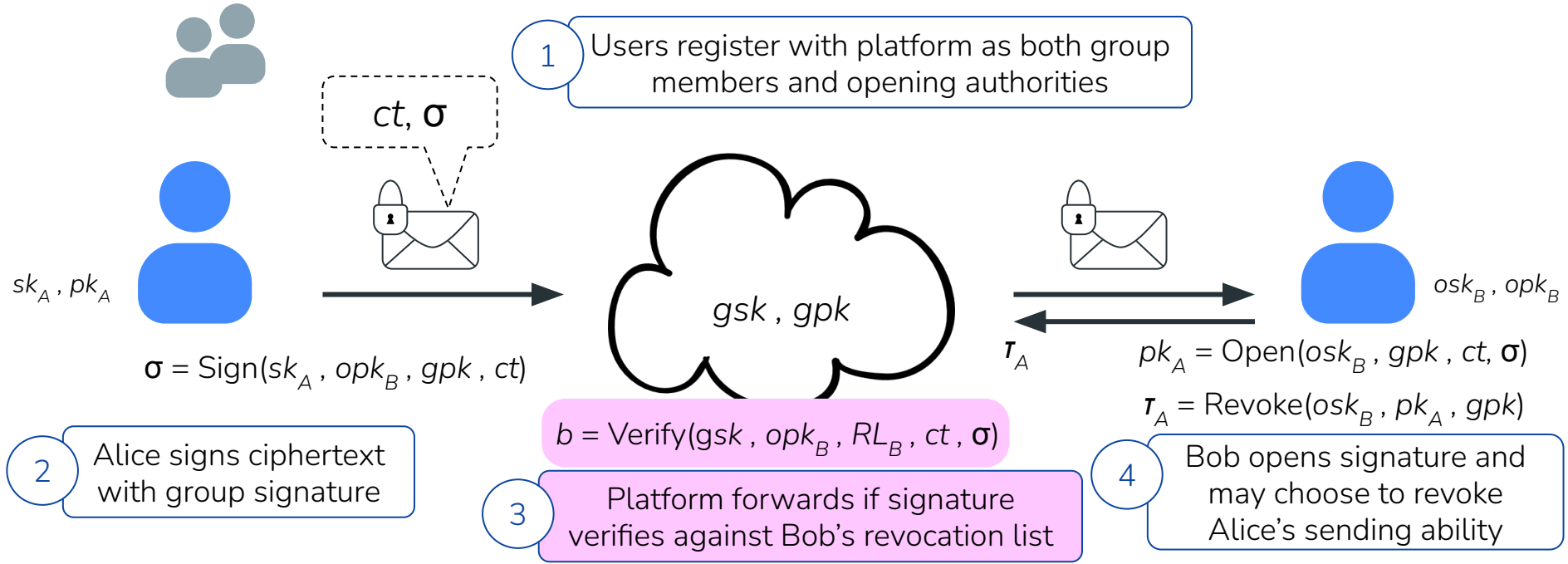
Building block: Group signatures

- Group signatures are anonymous, traceable, and revocable
- Group signatures do not require initialization over non-sender-anonymous channels
- Contribution: **Multi-opener group signatures**



Building block: Group signatures

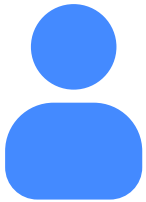
- Group signatures are anonymous, traceable, and revocable
- Group signatures do not require initialization over non-sender-anonymous channels
- Contribution: Multi-opener group signatures, **Keyed-verification group signatures**



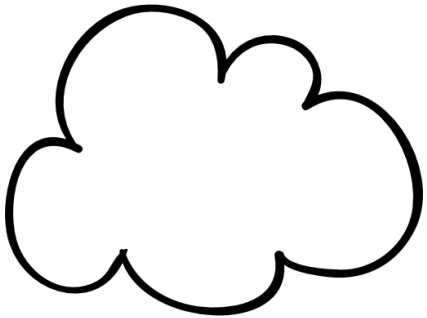
Outline

- **Contribution**: Blocklisting for sender-anonymous messaging
- Identifying weaknesses in Signal's sealed sender protocol
  - Requires non-sender-anonymous communication to initialize
  - Admits untraceable battery-draining (griefing) attack
- Orca: a sender-anonymous blocklisting protocol
  - Group signature scheme for sender-anonymous initialization
  - Efficient one-time-use authentication tokens from algebraic MACs

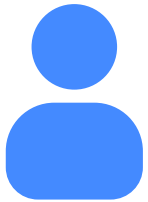
Building block: One-time use access tokens



Alice

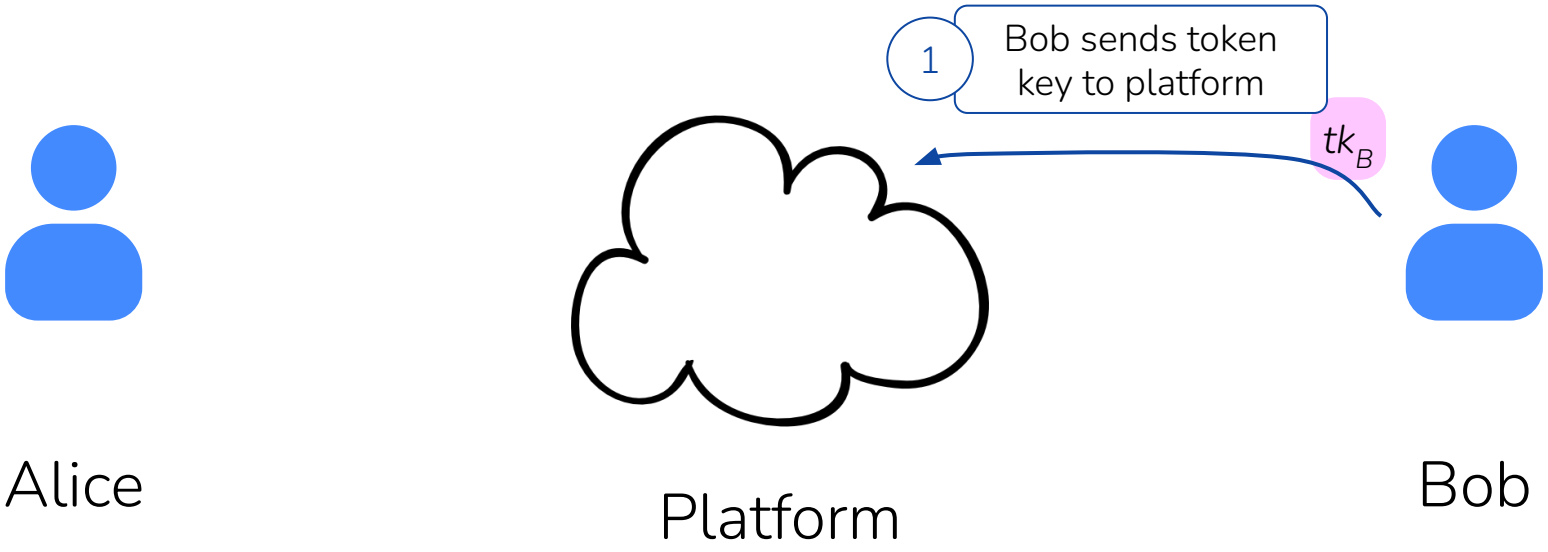


Platform

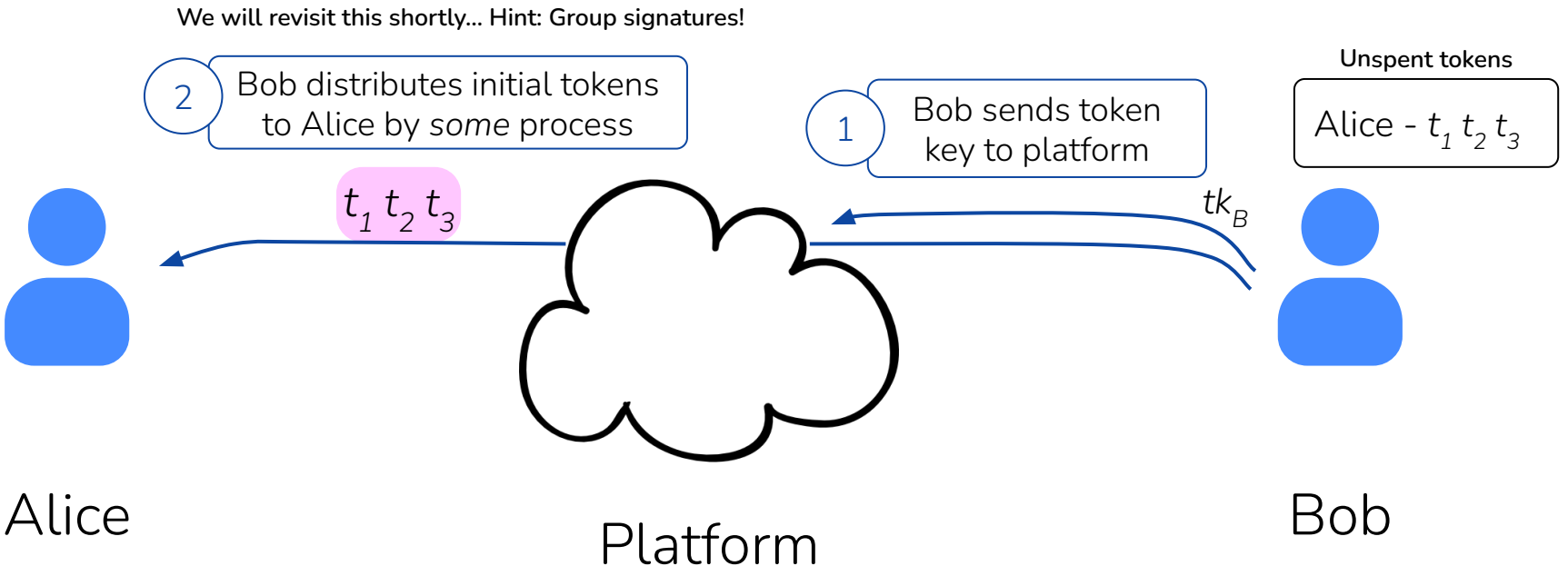


Bob

Building block: One-time use access tokens



Building block: One-time use access tokens

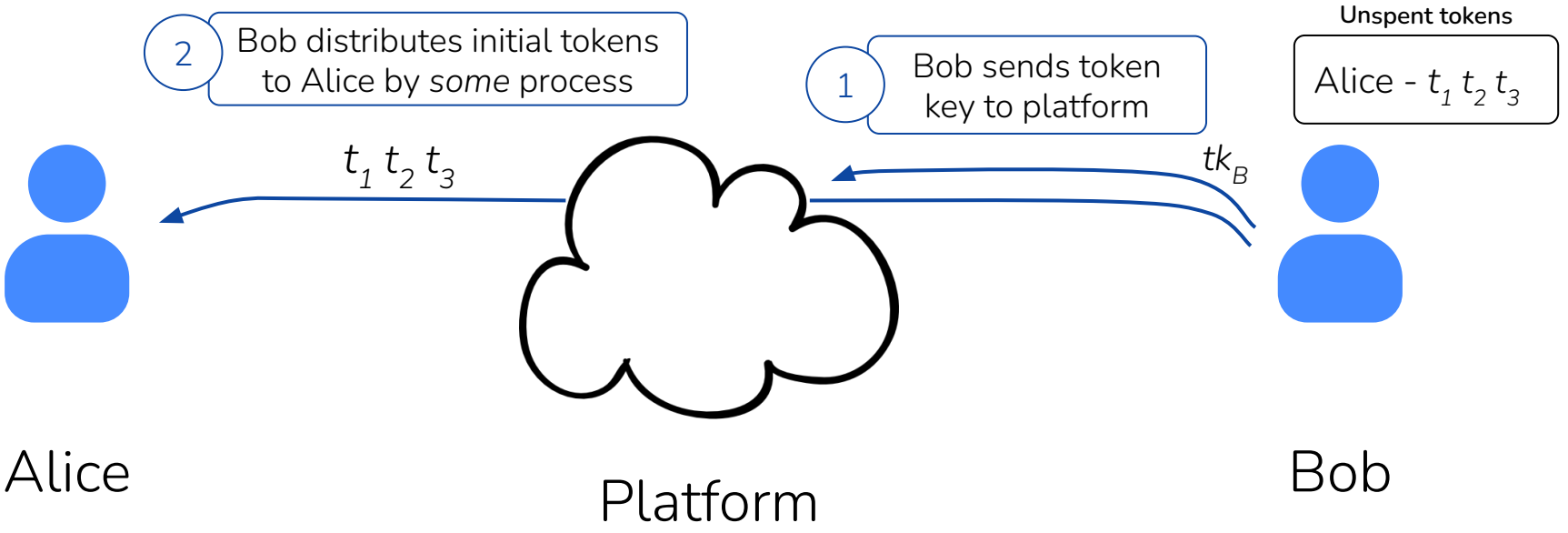


Building block: One-time use access tokens

Tokens are generated by sampling a random input x and computing the MAC

$y = \mathrm{MAC}_{tk}(x)$

$t = (x, y)$

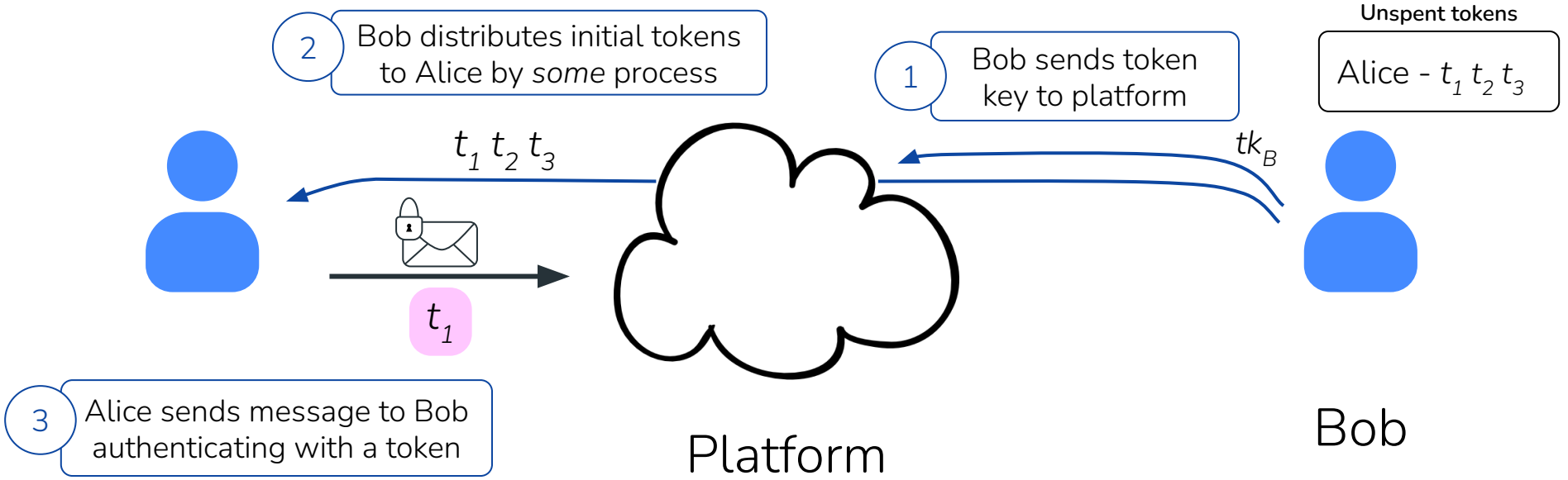


Building block: One-time use access tokens

Tokens are generated by sampling a random input x and computing the MAC

$y = \mathrm{MAC}_{tk}(x)$

$t = (x, y)$

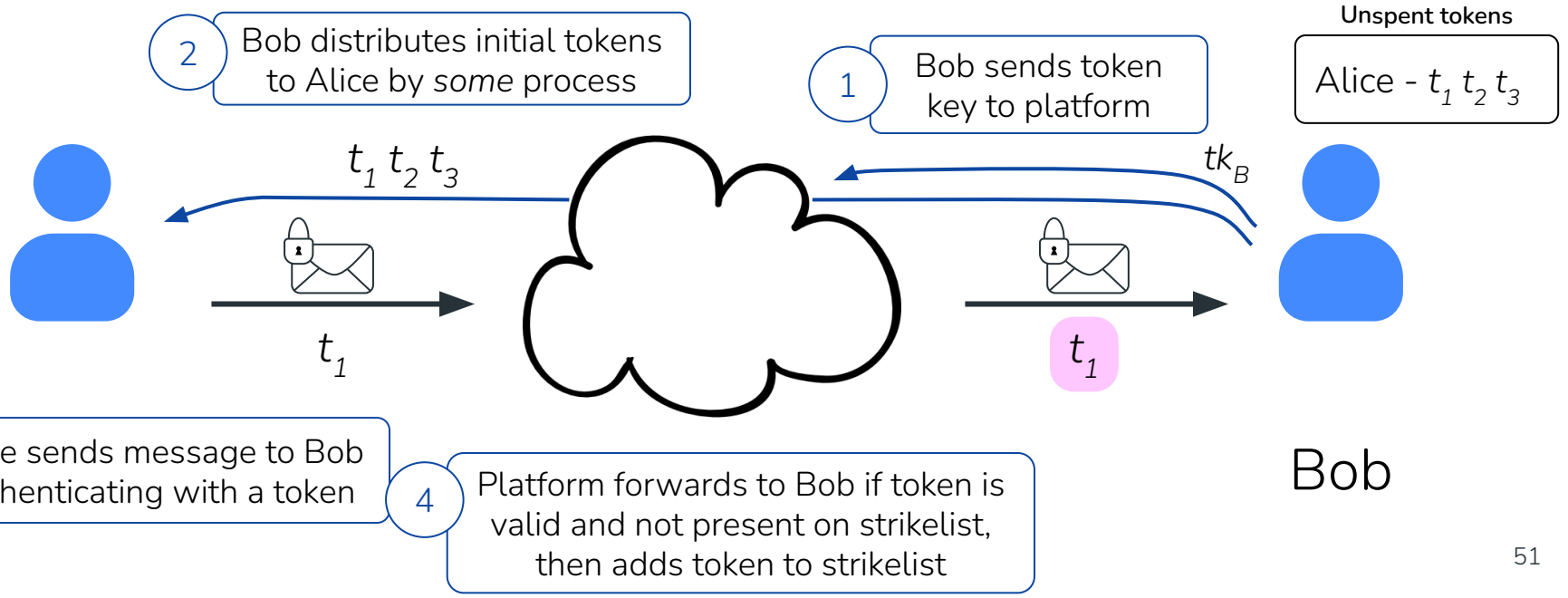


Building block: One-time use access tokens

Tokens are generated by sampling a random input x and computing the MAC

$y = \mathrm{MAC}_{tk}(x)$

$t = (x, y)$

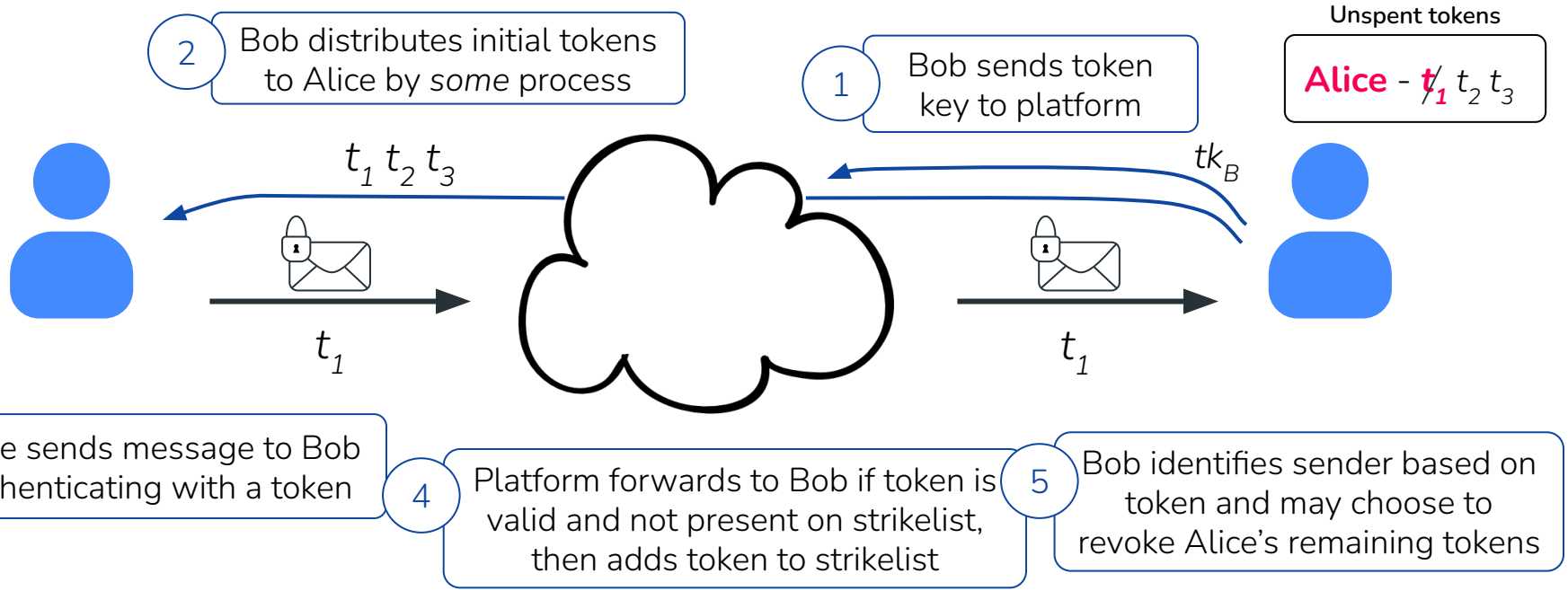


Building block: One-time use access tokens

Tokens are generated by sampling a random input x and computing the MAC

$y = \mathrm{MAC}_{tk}(x)$

$t = (x, y)$

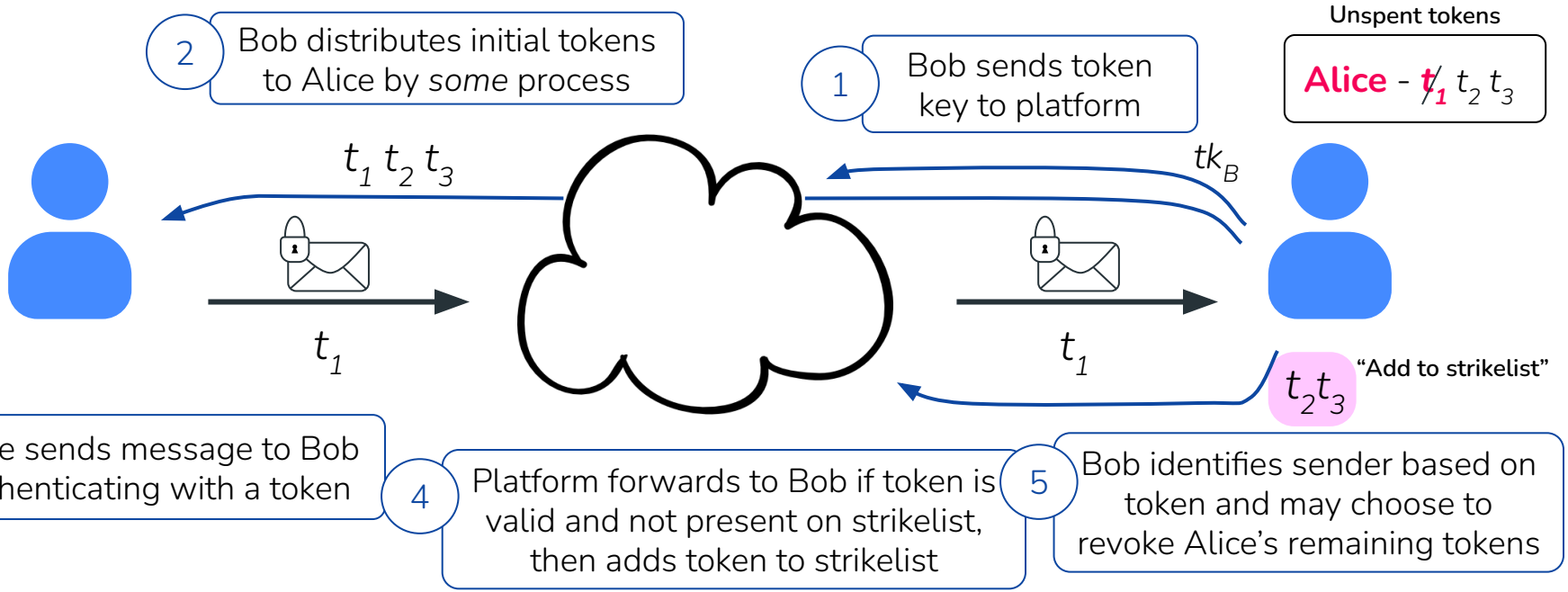


Building block: One-time use access tokens

Tokens are generated by sampling a random input x and computing the MAC

$y = \mathrm{MAC}_{tk}(x)$

$t = (x, y)$

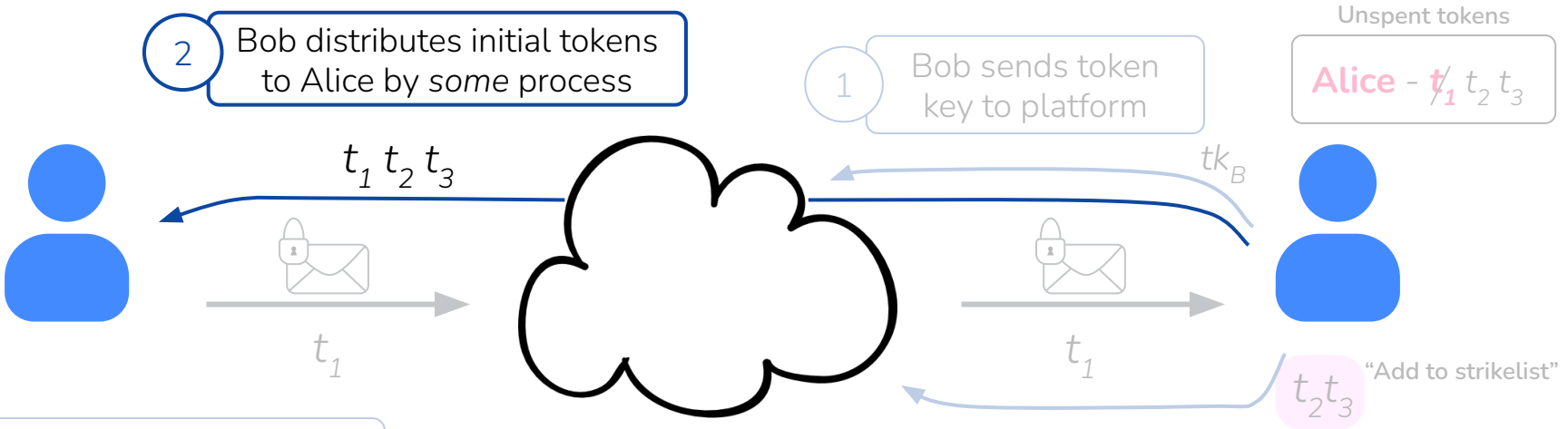


Building block: One-time use access tokens

Tokens are generated by sampling a random input x and computing the MAC

$y = \mathrm{MAC}_{tk}(x)$

$t = (x, y)$

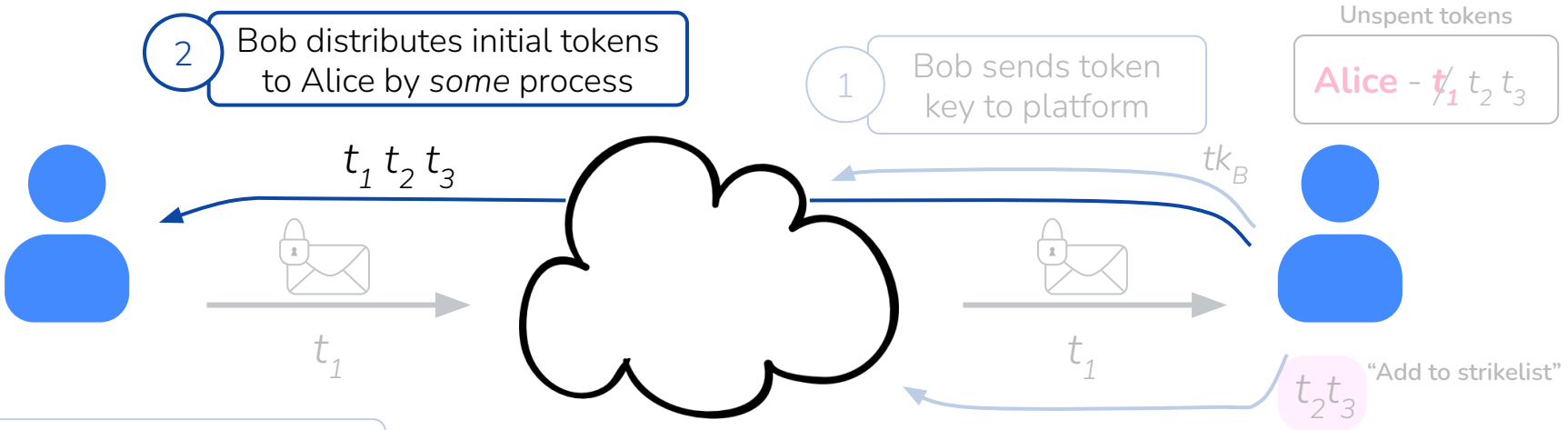


Building block: One-time use access tokens

Do not want to rely on non-sender-anonymous channels!

generated by sampling a
t x and computing the MAC

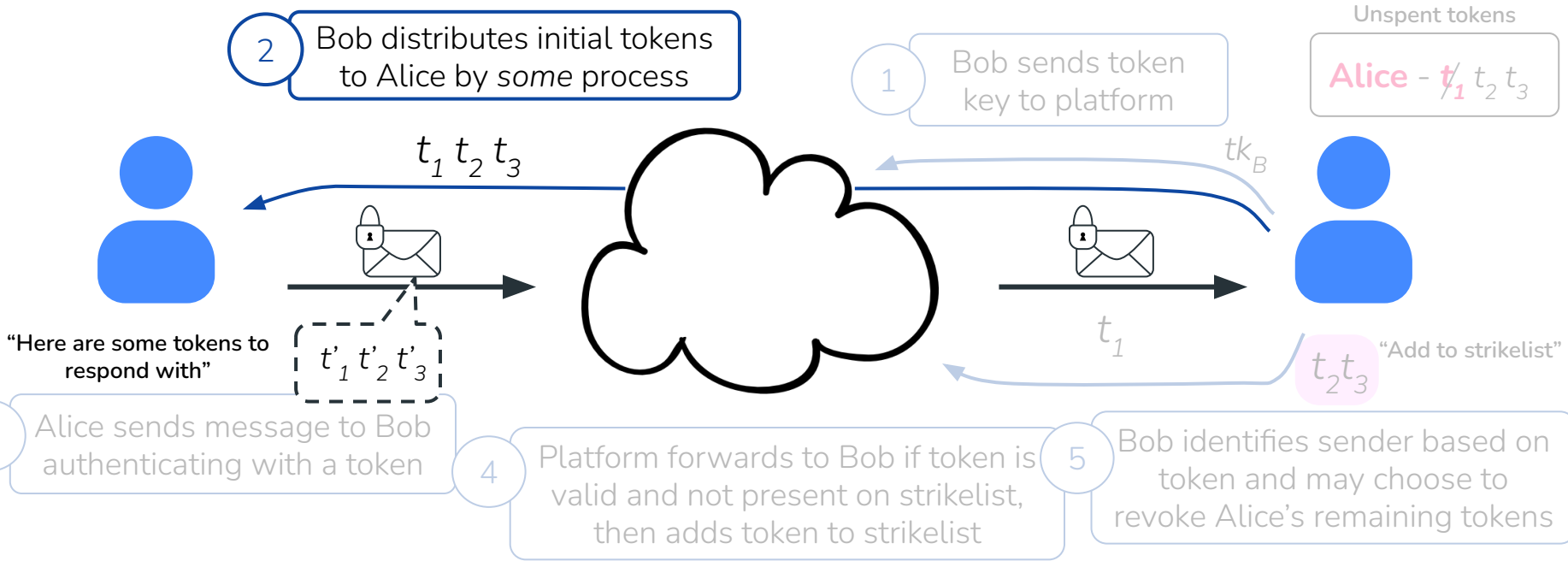
$y = \mathrm{MAC}_{tk}(x)$
$t = (x, y)$



Building block: One-time use access tokens

Do not want to rely on non-sender-anonymous channels!

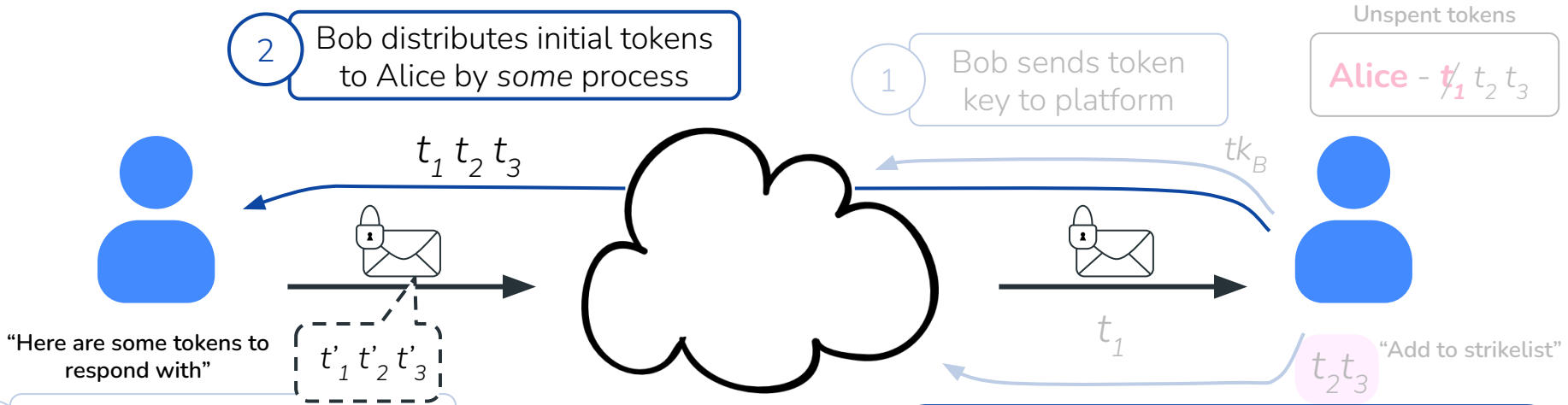
After initial batch is distributed, future tokens can be replenished in the regular flow of conversation



Building block: One-time use access tokens

Do not want to rely on non-sender-anonymous channels!

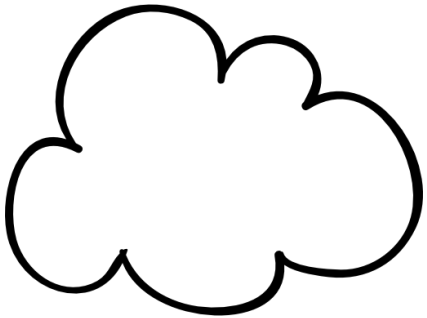
After initial batch is distributed, future tokens can be replenished in the regular flow of conversation



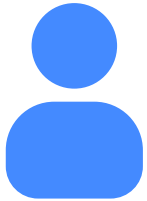
Orca: Hybrid of OTU tokens + Group signatures

- Use group signature to send initial batch of one-time-use (OTU) tokens

Alice



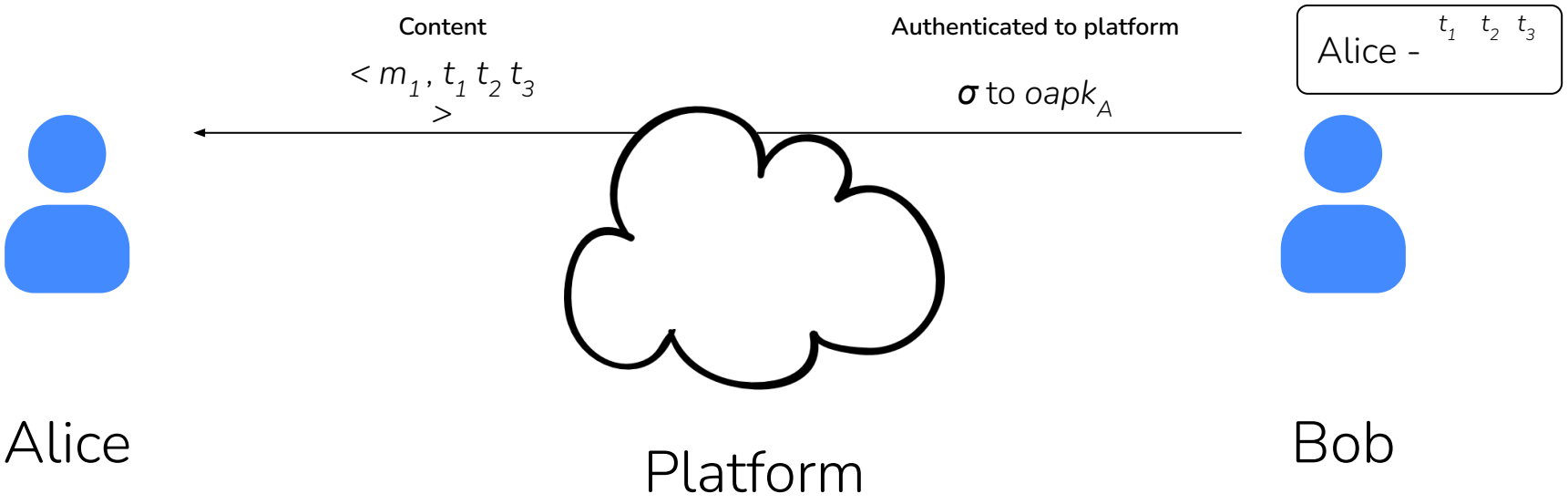
Platform



Bob

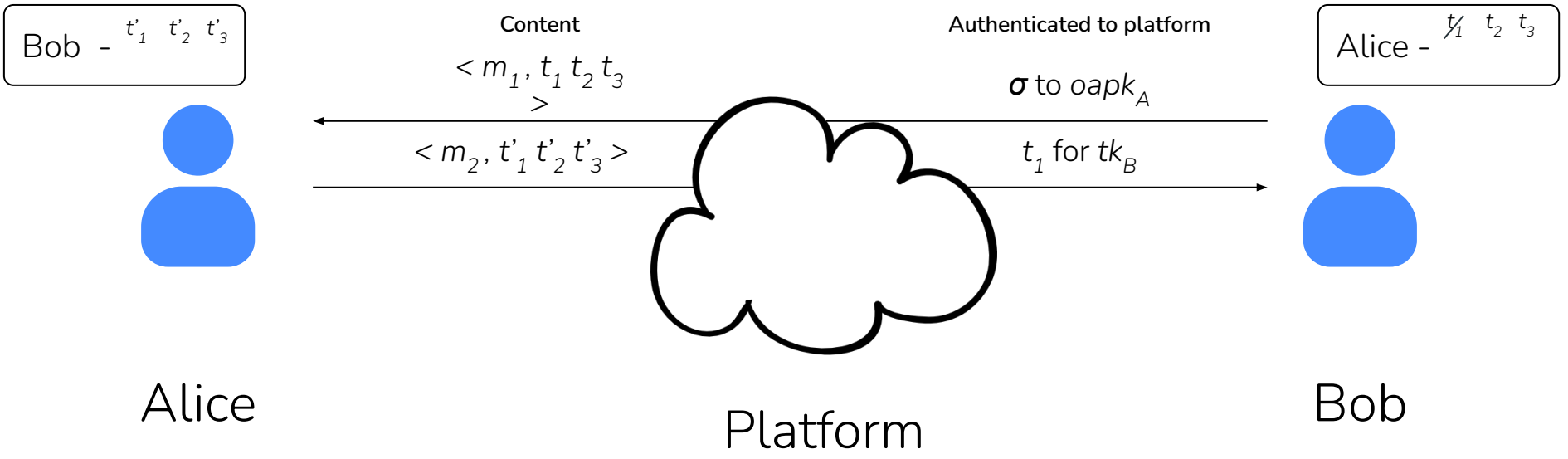
Orca: Hybrid of OTU tokens + Group signatures

- Use group signature to send initial batch of one-time-use (OTU) tokens



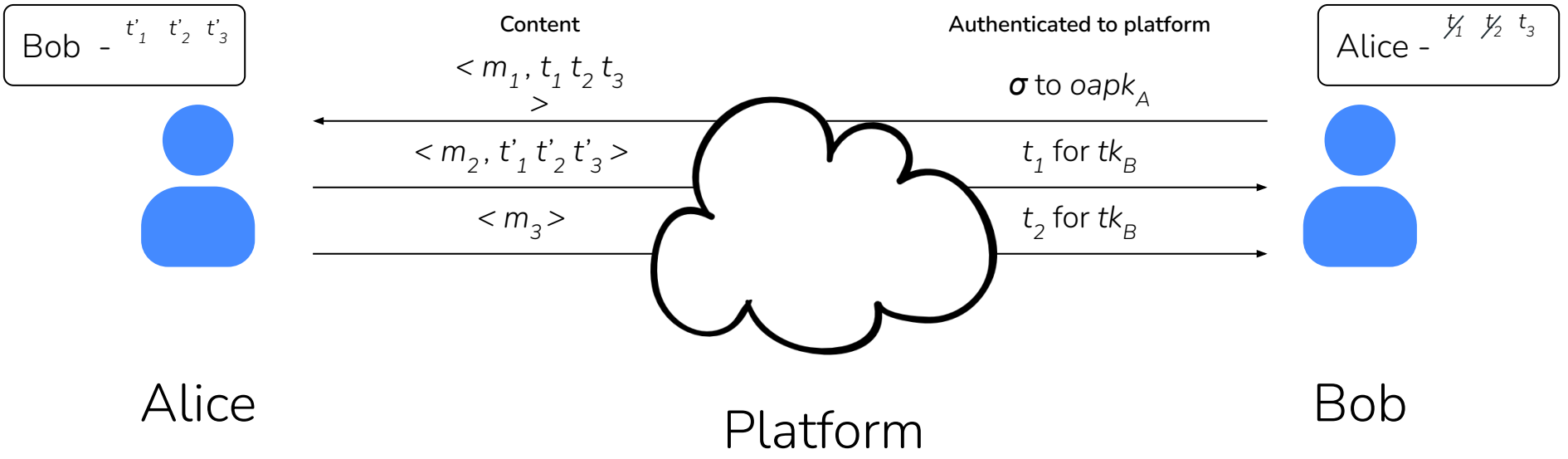
Orca: Hybrid of OTU tokens + Group signatures

- Use group signature to send initial batch of one-time-use (OTU) tokens



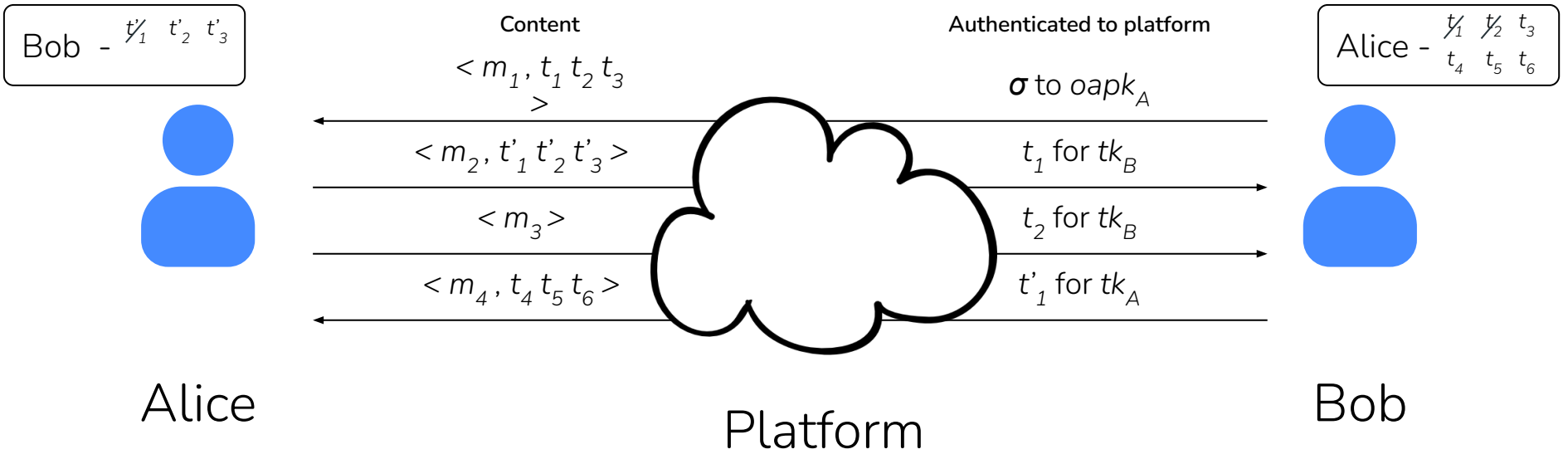
Orca: Hybrid of OTU tokens + Group signatures

- Use group signature to send initial batch of one-time-use (OTU) tokens



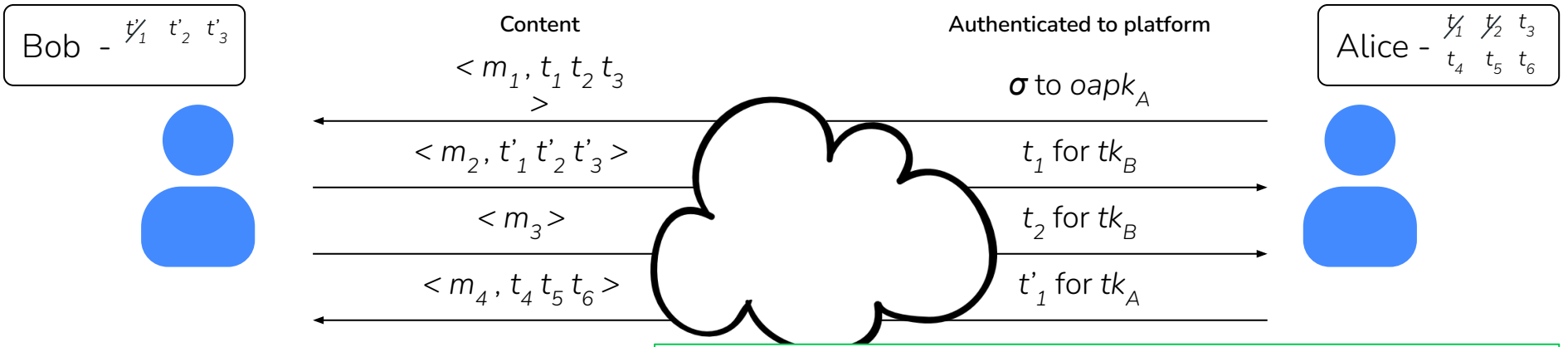
Orca: Hybrid of OTU tokens + Group signatures

- Use group signature to send initial batch of one-time-use (OTU) tokens



Orca: Hybrid of OTU tokens + Group signatures

- Use group signature to send initial batch of one-time-use (OTU) tokens



Alice

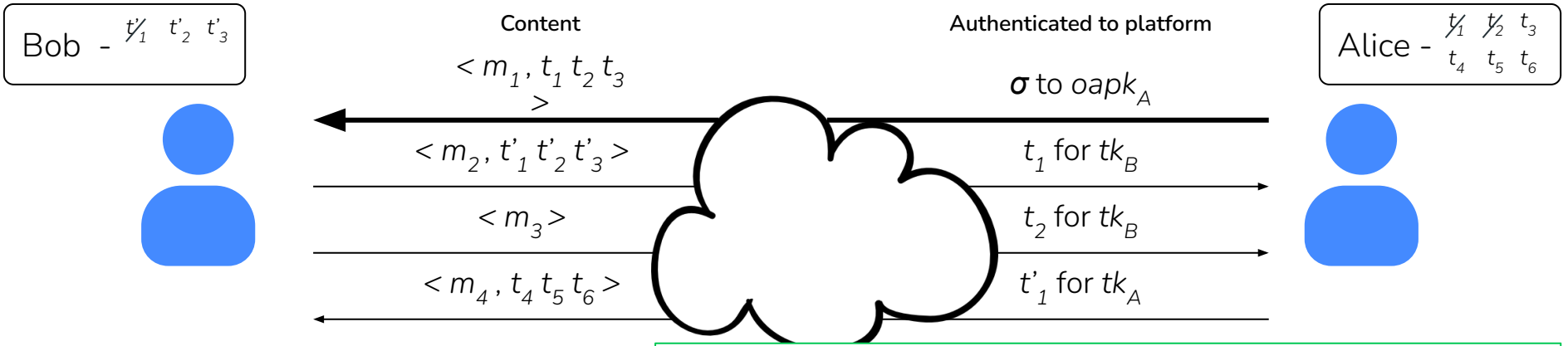
Best of both worlds!

Sender-anonymous initialization via group signatures and efficiency via one-time-use tokens

Orca: Hybrid of OTU tokens + Group signatures

- Use group signature to send initial batch of one-time-use (OTU) tokens

Group signatures until first response? Remedied by oblivious token minting protocol. See paper.



Alice

Best of both worlds!

Sender-anonymous initialization via group signatures and efficiency via one-time-use tokens

Impact

- Open source: Implemented to confirm practicality of solution
  - “Steady-state” costs of authenticating via OTU tokens add little overhead
  - Initialization costs of group signature and token minting
    - ~ 200 ms computation for both platform and client
- Disclosed findings to Signal
  - Advising on possible partial mitigations

Open source: `https://github.com/nirvantyagi/orca`

Archive: `https://ia.cr/2021/1380`

Summary

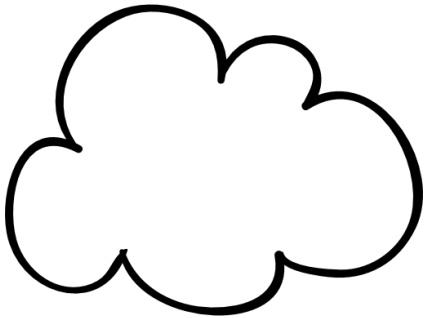
- **Contribution**: Blocklisting for sender-anonymous messaging
- Identifying weaknesses in Signal's sealed sender protocol
  - Requires non-sender-anonymous communication to initialize
  - Admits untraceable battery-draining (griefing) attack
- Orca: a sender-anonymous blocklisting protocol
  - Group signature scheme for sender-anonymous initialization
  - Efficient one-time-use authentication tokens from algebraic MACs

Open source: `https://github.com/nirvantyagi/orca`

Archive: `https://ia.cr/2021/1380`

Back-up slides

Inference attack: Interleaving messages



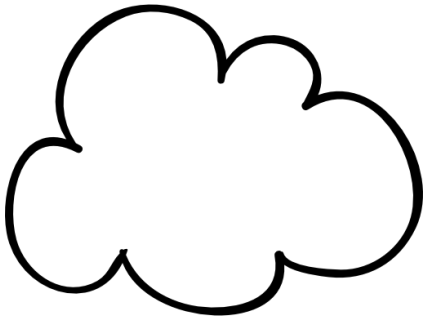
Platform

Observations

Time	Recipient
2021-8-18 8:08:59	Bob
2021-8-18 8:14:02	Claire
2021-8-18 8:20:19	Alice
2021-8-18 8:22:25	Dave
2021-8-18 8:29:55	Claire
2021-8-18 8:31:38	Dave
2021-8-18 8:46:24	Claire
2021-8-18 8:57:11	Dave
2021-8-18 9:06:41	Claire
2021-8-18 9:09:56	Bob
2021-8-18 9:14:39	Dave
2021-8-18 9:17:29	Claire
2021-8-18 9:20:12	Alice
2021-8-18 9:20:30	Dave
2021-8-18 9:28:54	Claire
2021-8-18 9:37:15	Dave
2021-8-18 9:42:08	Bob

[MKARW NDSS'21]

Inference attack: Interleaving messages



Platform

Observations

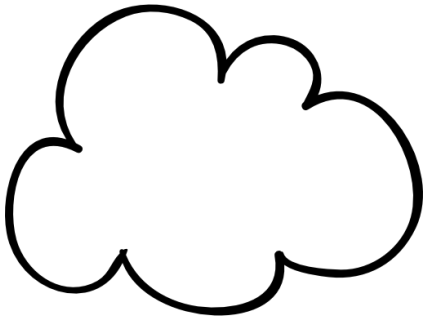
Time	Recipient
2021-8-18 8:08:59	Bob
2021-8-18 8:14:02	Claire
2021-8-18 8:20:19	Alice
2021-8-18 8:22:25	Dave
2021-8-18 8:29:55	Claire
2021-8-18 8:31:38	Dave
2021-8-18 8:46:24	Claire
2021-8-18 8:57:11	Dave
2021-8-18 9:06:41	Claire
2021-8-18 9:09:56	Bob
2021-8-18 9:14:39	Dave
2021-8-18 9:17:29	Claire
2021-8-18 9:20:12	Alice
2021-8-18 9:20:30	Dave
2021-8-18 9:28:54	Claire
2021-8-18 9:37:15	Dave
2021-8-18 9:42:08	Bob

Bob & Claire?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

[MKARW NDSS'21]

Inference attack: Interleaving messages



Platform

Observations

Time	Recipient
2021-8-18 8:08:59	Bob
2021-8-18 8:14:02	Claire
2021-8-18 8:20:19	Alice
2021-8-18 8:22:25	Dave
2021-8-18 8:29:55	Claire
2021-8-18 8:31:38	Dave
2021-8-18 8:46:24	Claire
2021-8-18 8:57:11	Dave
2021-8-18 9:06:41	Claire
2021-8-18 9:09:56	Bob
2021-8-18 9:14:39	Dave
2021-8-18 9:17:29	Claire
2021-8-18 9:20:12	Alice
2021-8-18 9:20:30	Dave
2021-8-18 9:28:54	Claire
2021-8-18 9:37:15	Dave
2021-8-18 9:42:08	Bob

Bob & Claire?

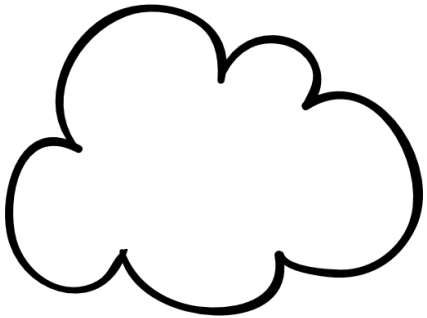
Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

Bob & Alice?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

[MKARW NDSS'21]

Inference attack: Interleaving messages



Platform

Observations

Time	Recipient
2021-8-18 8:08:59	Bob
2021-8-18 8:14:02	Claire
2021-8-18 8:20:19	Alice
2021-8-18 8:22:25	Dave
2021-8-18 8:29:55	Claire
2021-8-18 8:31:38	Dave
2021-8-18 8:46:24	Claire
2021-8-18 8:57:11	Dave
2021-8-18 9:06:41	Claire
2021-8-18 9:09:56	Bob
2021-8-18 9:14:39	Dave
2021-8-18 9:17:29	Claire
2021-8-18 9:20:12	Alice
2021-8-18 9:20:30	Dave
2021-8-18 9:28:54	Claire
2021-8-18 9:37:15	Dave
2021-8-18 9:42:08	Bob

Bob & Claire?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

Bob & Alice?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

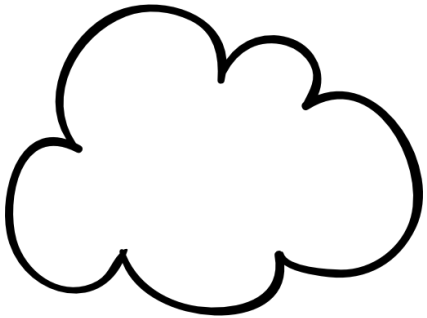
Dave & Claire?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

[MKARW NDSS'21]

Inference attack: Interleaving messages

Attack efficacy varies based on number of conversation participants, frequency of response, participation balance, etc.



Platform

2021-8-18 8:14:02	Claire
2021-8-18 8:20:19	Alice
2021-8-18 8:22:25	Dave
2021-8-18 8:29:55	Claire
2021-8-18 8:31:38	Dave
2021-8-18 8:46:24	Claire
2021-8-18 8:57:11	Dave
2021-8-18 9:06:41	Claire
2021-8-18 9:09:56	Bob
2021-8-18 9:14:39	Dave
2021-8-18 9:17:29	Claire
2021-8-18 9:20:12	Alice
2021-8-18 9:20:30	Dave
2021-8-18 9:28:54	Claire
2021-8-18 9:37:15	Dave
2021-8-18 9:42:08	Bob

Bob & Claire?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

Bob & Alice?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

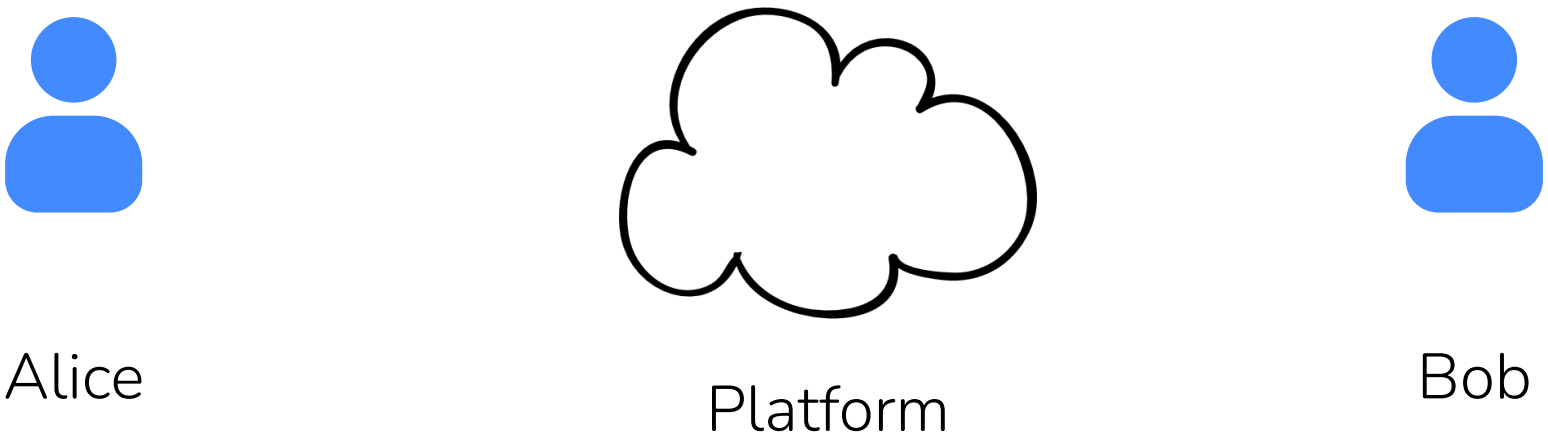
Dave & Claire?

Recipient
Bob
Claire
Alice
Dave
Claire
Dave
Claire
Dave
Claire
Bob
Dave
Claire
Alice
Dave
Claire
Dave
Bob

[MKARW NDSS'21]

Orca: Hybrid of OTU tokens + Group signatures

- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels

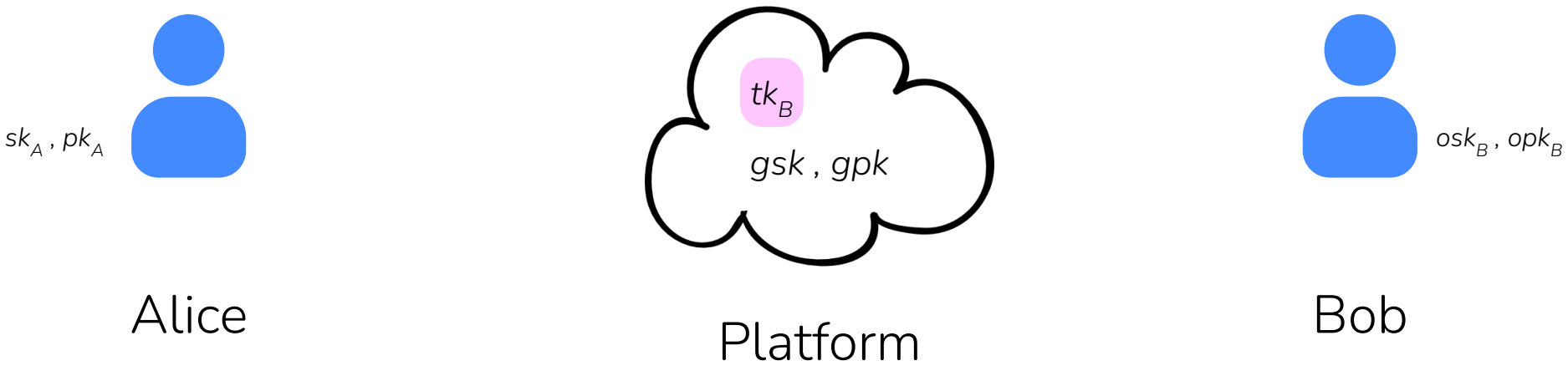


Orca: Hybrid of OTU tokens + Group signatures

- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels

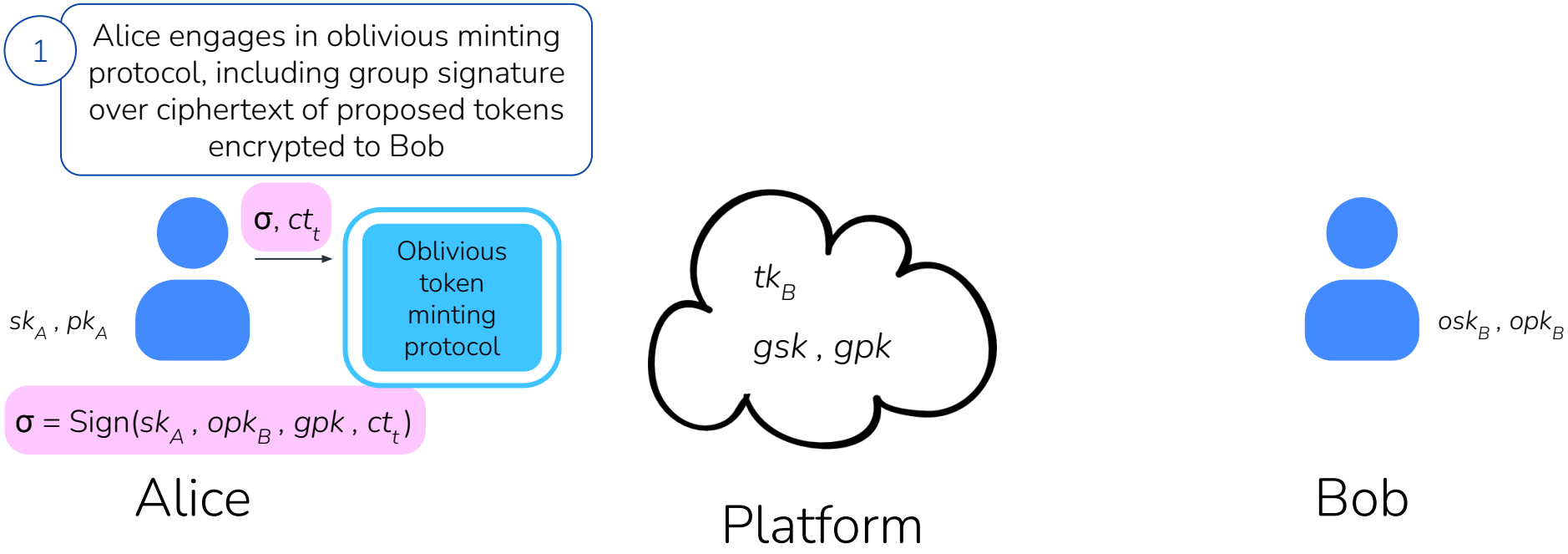
Orca: Hybrid of OTU tokens + Group signatures

- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels



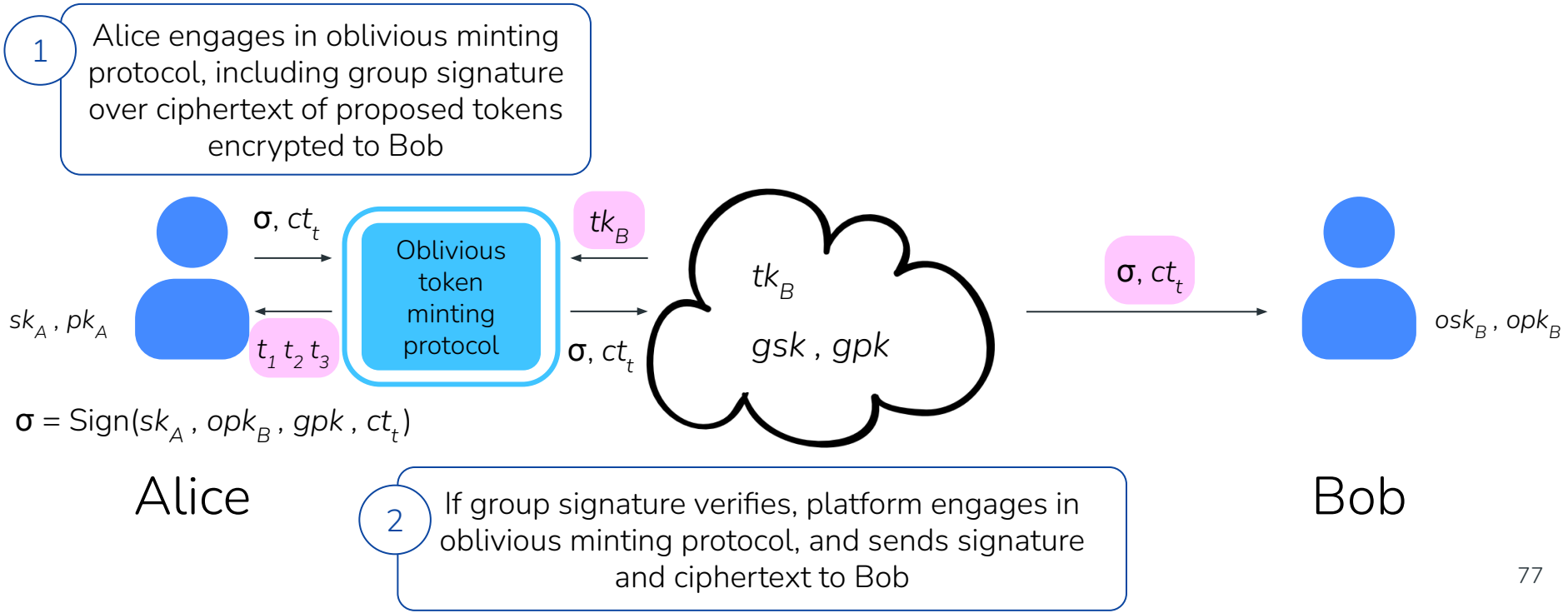
Orca: Hybrid of OTU tokens + Group signatures

- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels



Orca: Hybrid of OTU tokens + Group signatures

- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels



Orca: Hybrid of OTU tokens + Group signatures

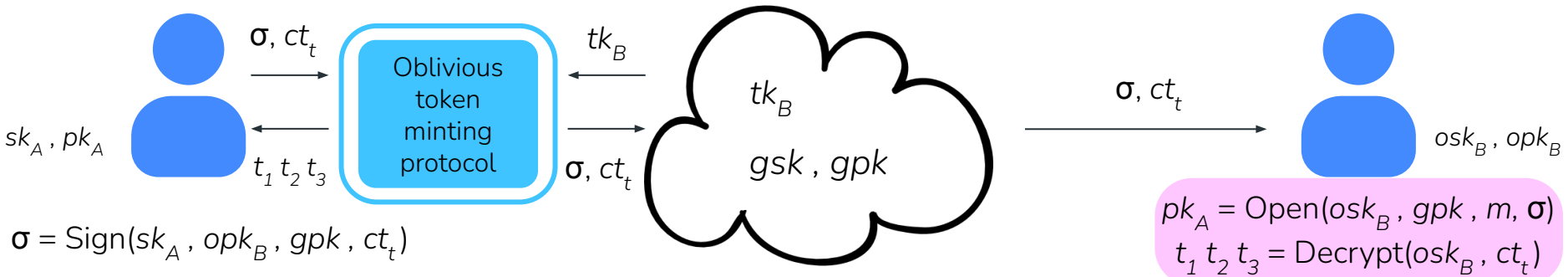
- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels

1 Alice engages in oblivious minting protocol, including group signature over ciphertext of proposed tokens encrypted to Bob

3 Bob decrypts ciphertext to get tokens, opens signature to identify Alice, then stores tokens to identify future messages from Alice

Unspent tokens

Alice - t_1 t_2 t_3



σ = Sign(sk_A , opk_B , gpk , ct_t)

pk_A = Open(osk_B , gpk , m, σ)

t_1 t_2 t_3 = Decrypt(osk_B , ct_t)

Alice

Bob

2 If group signature verifies, platform engages in oblivious minting protocol, and sends signature and ciphertext to Bob

Orca: Hybrid of OTU tokens + Group signatures

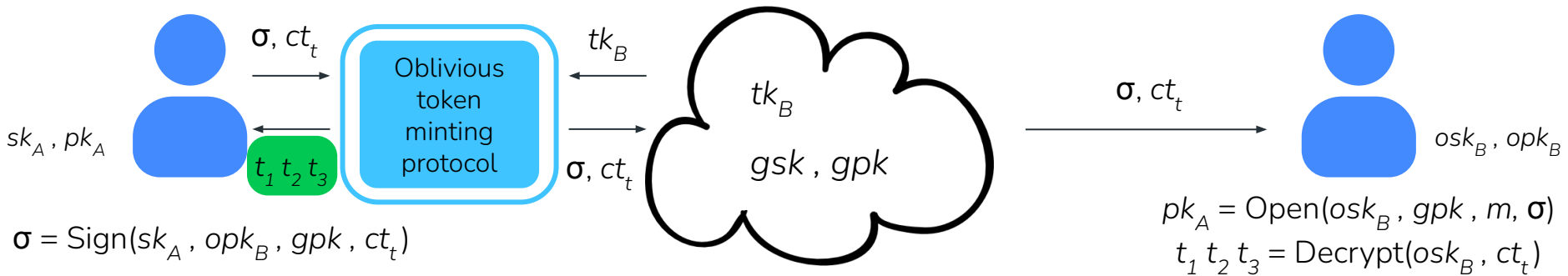
- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels

1 Alice engages in oblivious minting protocol, including group signature over ciphertext of proposed tokens encrypted to Bob

3 Bob decrypts ciphertext to get tokens, opens signature to identify Alice, then stores tokens to identify future messages from Alice

Unspent tokens

Alice - t_1 t_2 t_3



σ = Sign(sk_A , opk_B , gpk , ct_t)

pk_A = Open(osk_B , gpk , m, σ)

t_1 t_2 t_3 = Decrypt(osk_B , ct_t)

Alice

Bob

2 If group signature verifies, platform engages in oblivious minting protocol, and sends signature and ciphertext to Bob

Orca: Hybrid of OTU tokens + Group signatures

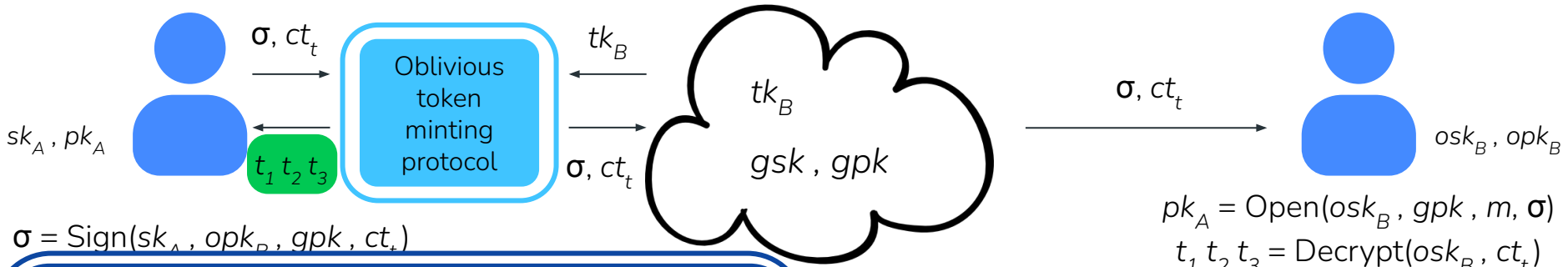
- Oblivious token minting protocol authenticated by group signatures to obtain first batch of tokens without use of non-sender-anonymous channels

1 Alice engages in oblivious minting protocol, including group signature over ciphertext of proposed tokens encrypted to Bob

3 Bob decrypts ciphertext to get tokens, opens signature to identify Alice, then stores tokens to identify future messages from Alice

Unspent tokens

Alice - t_1 t_2 t_3



σ = Sign(sk_A , opk_B , gpk , ct_t)

pk_A = Open(osk_B , gpk , m, σ)

t_1 t_2 t_3 = Decrypt(osk_B , ct_t)

Bob

platform engages in
and sends signature
t to Bob

Note: We used algebraic MACs for OTU tokens instead of OPRF-based tokens for algebraic proving properties