

ATAVISM: Private Originator Tracing in E2EE Messaging

Archisman Dutta
(IIT Bombay)

Debayan Gupta
(Ashoka University)

Arup Mondal

Plan for the afternoon

- End-to-End Encrypted Messaging

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

Plan for the afternoon

- **The Dilemma of End-to-End Encrypted Messaging**
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

End-to-End Encrypted Messaging

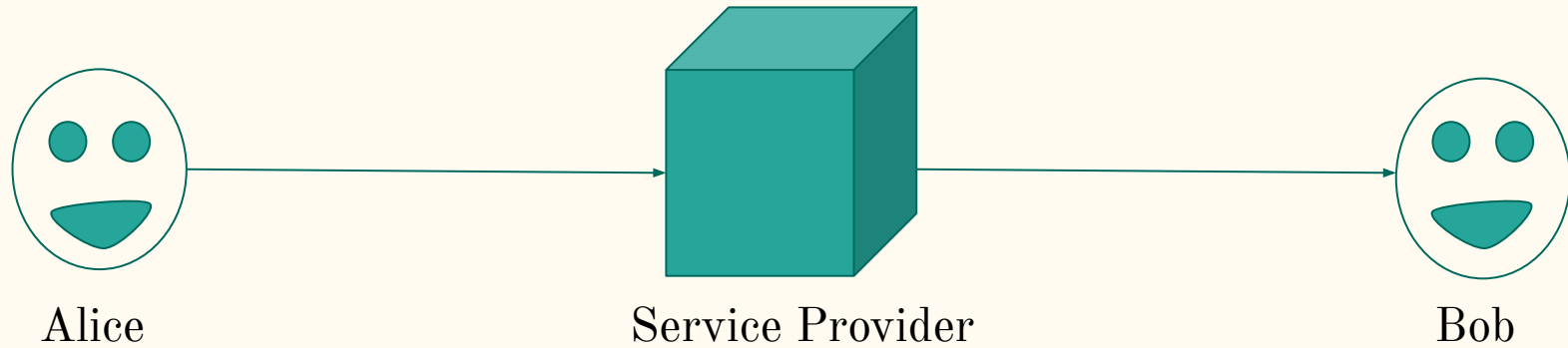


Alice

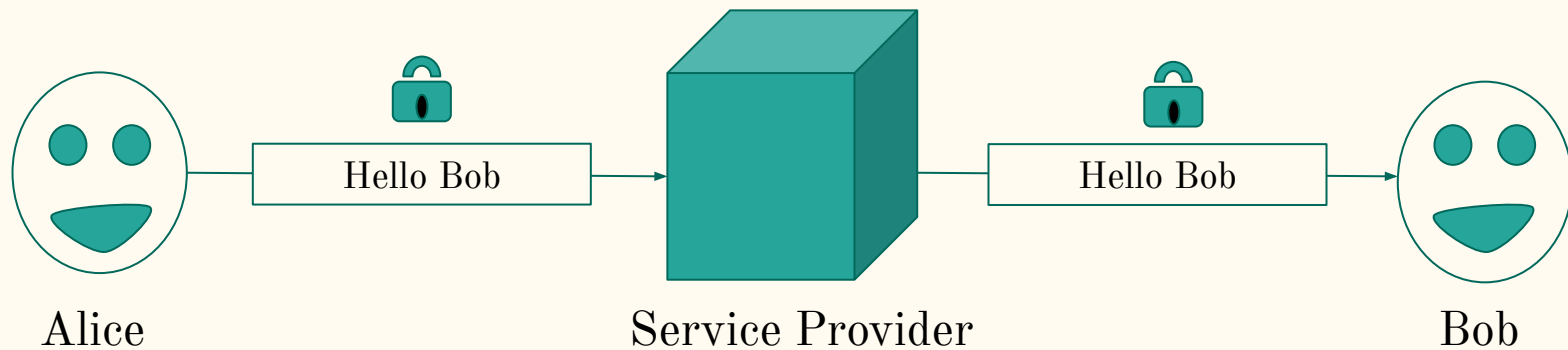


Bob

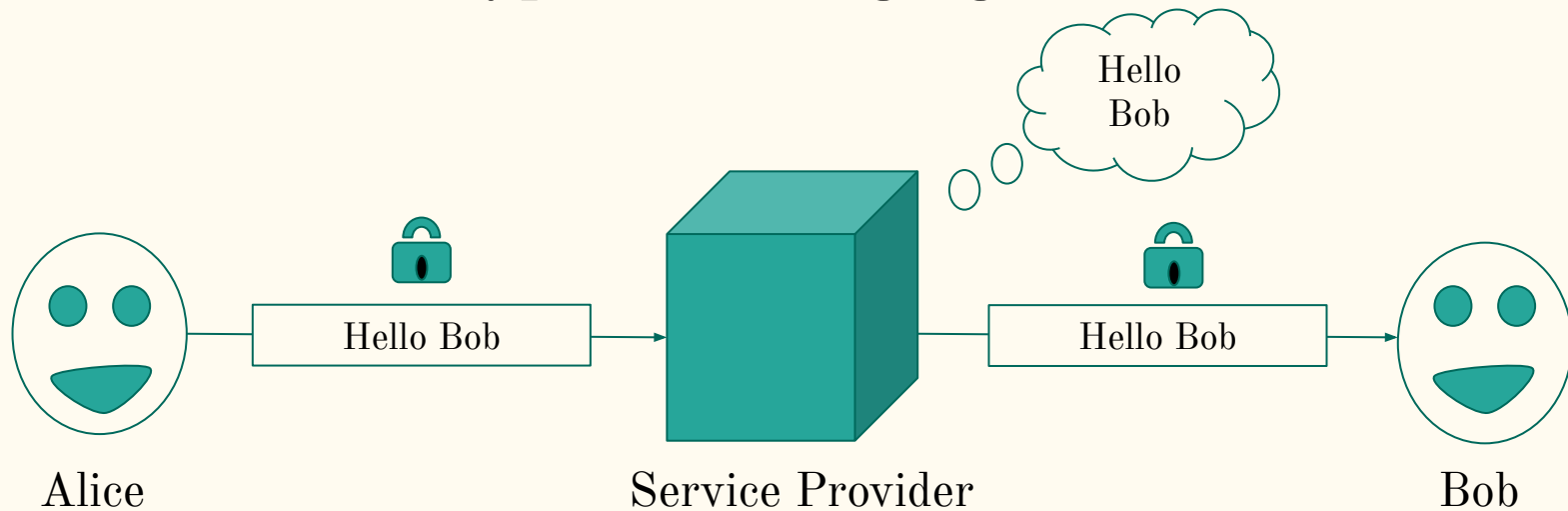
End-to-End Encrypted Messaging



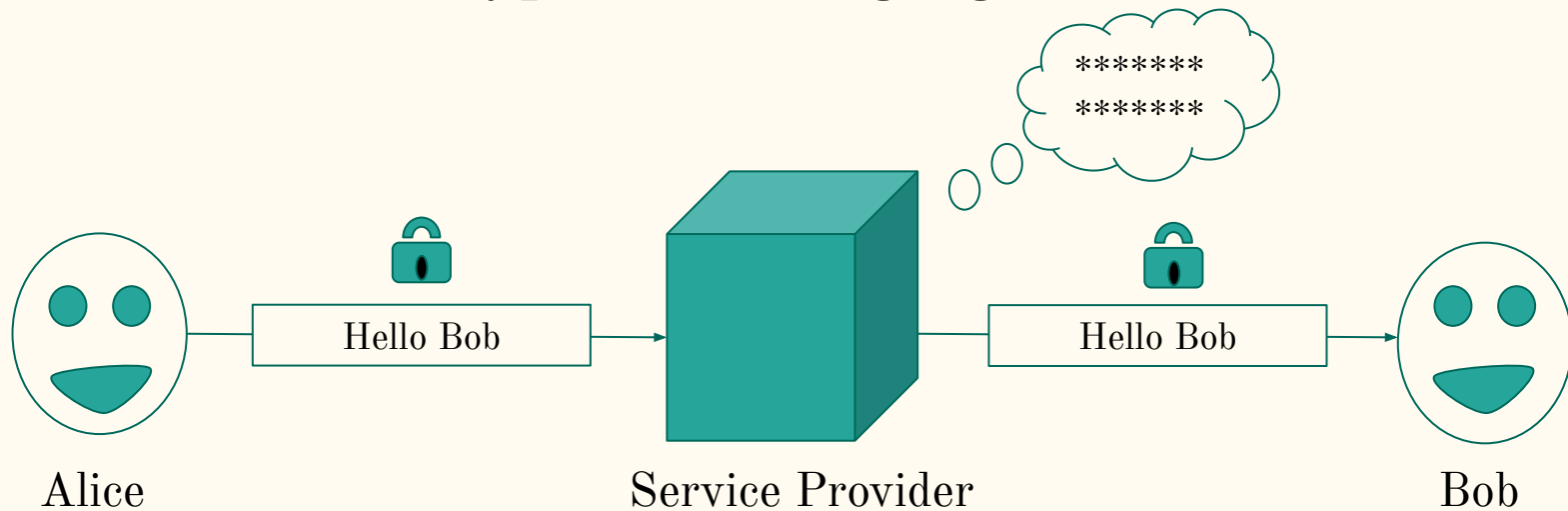
End-to-End Encrypted Messaging



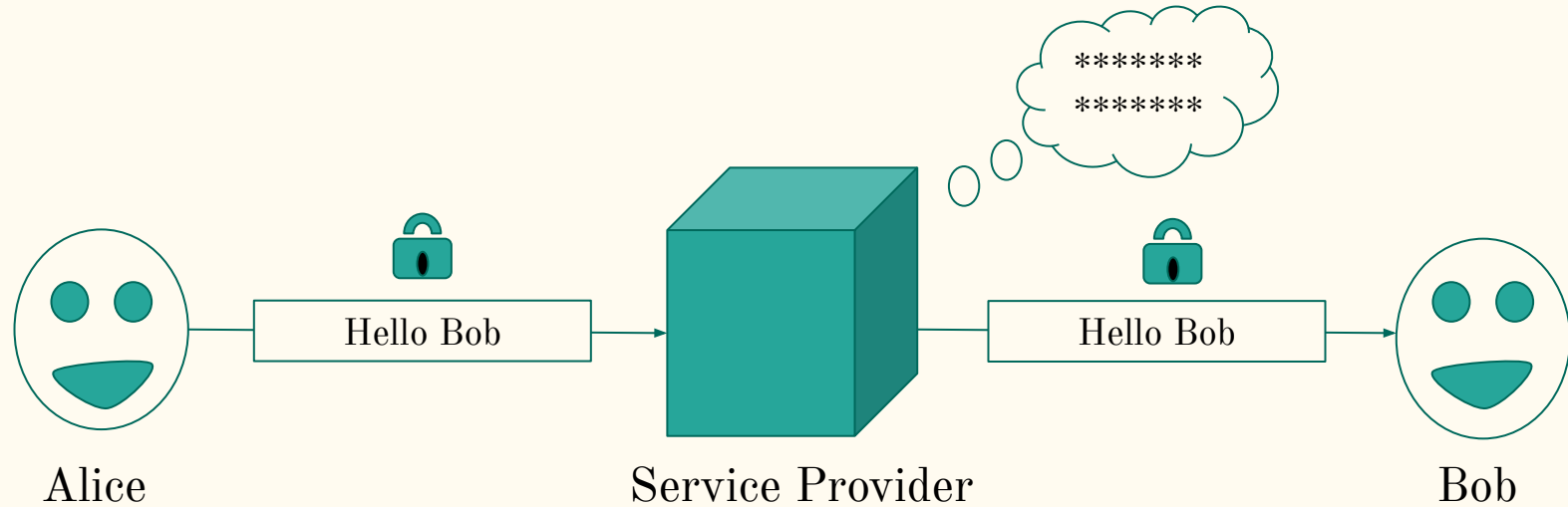
End-to-End Encrypted Messaging



End-to-End Encrypted Messaging

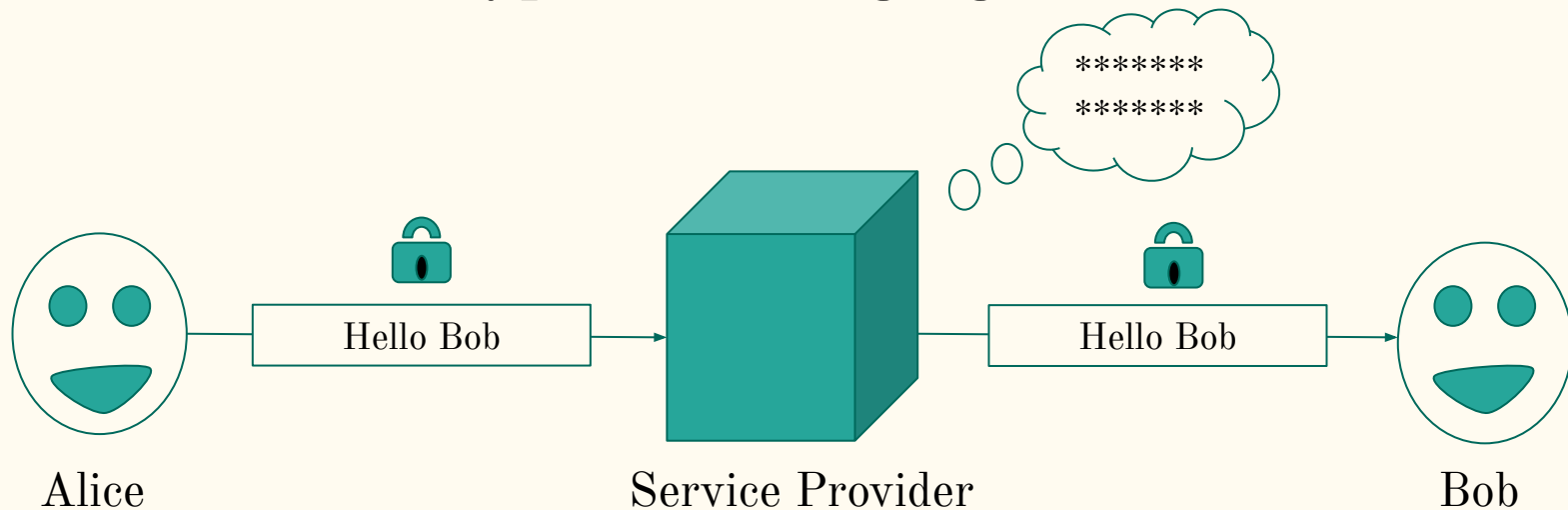


End-to-End Encrypted Messaging



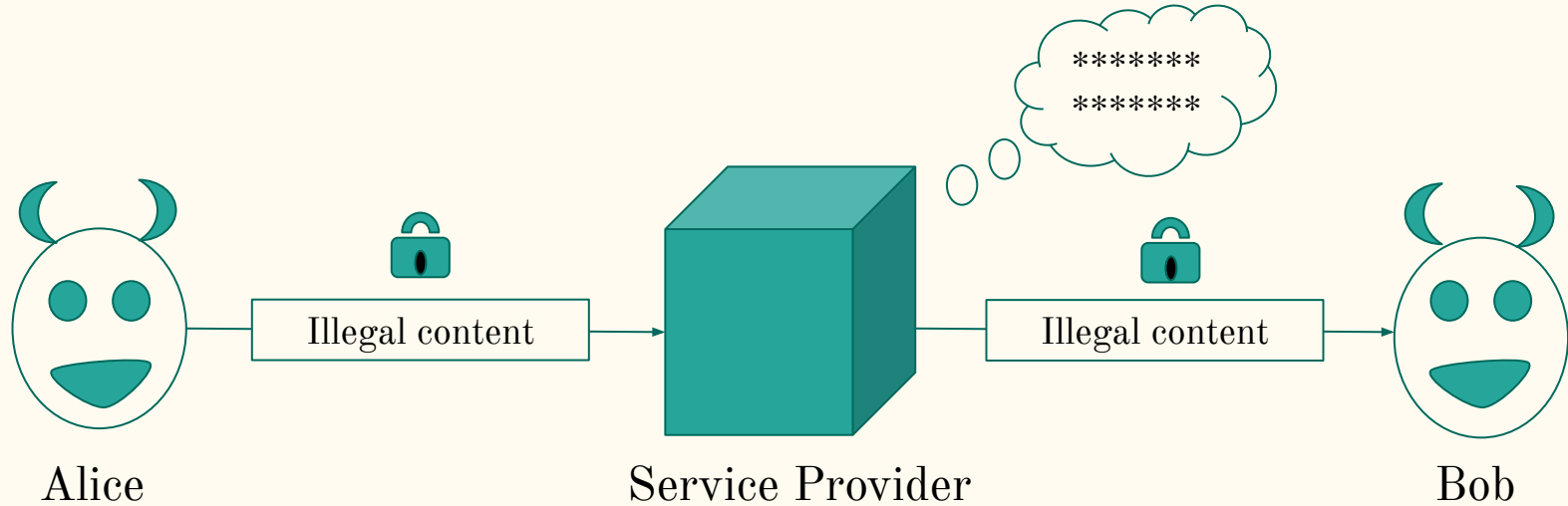
- No one but Alice and Bob – not even the service provider – can decrypt or read the message

End-to-End Encrypted Messaging



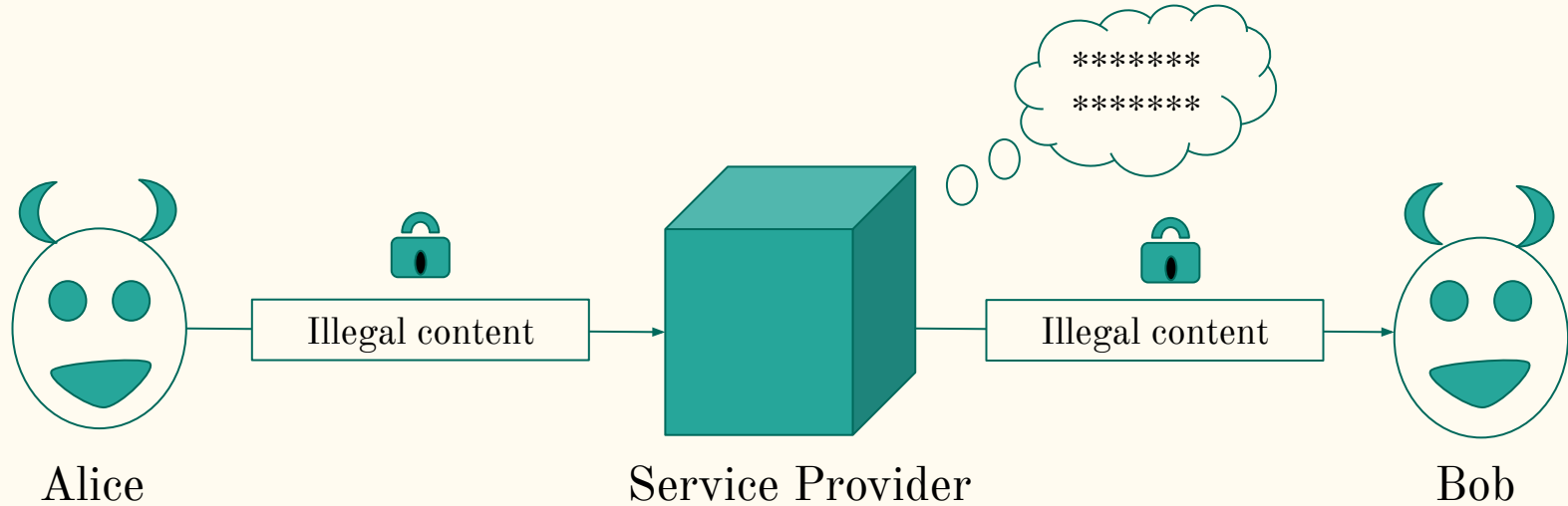
- No one but Alice and Bob – not even the service provider – can decrypt or read the message
- Eg: Signal, Matrix (Element), Session, WhatsApp, Telegram

(The Dilemma of) End-to-End Encrypted Messaging



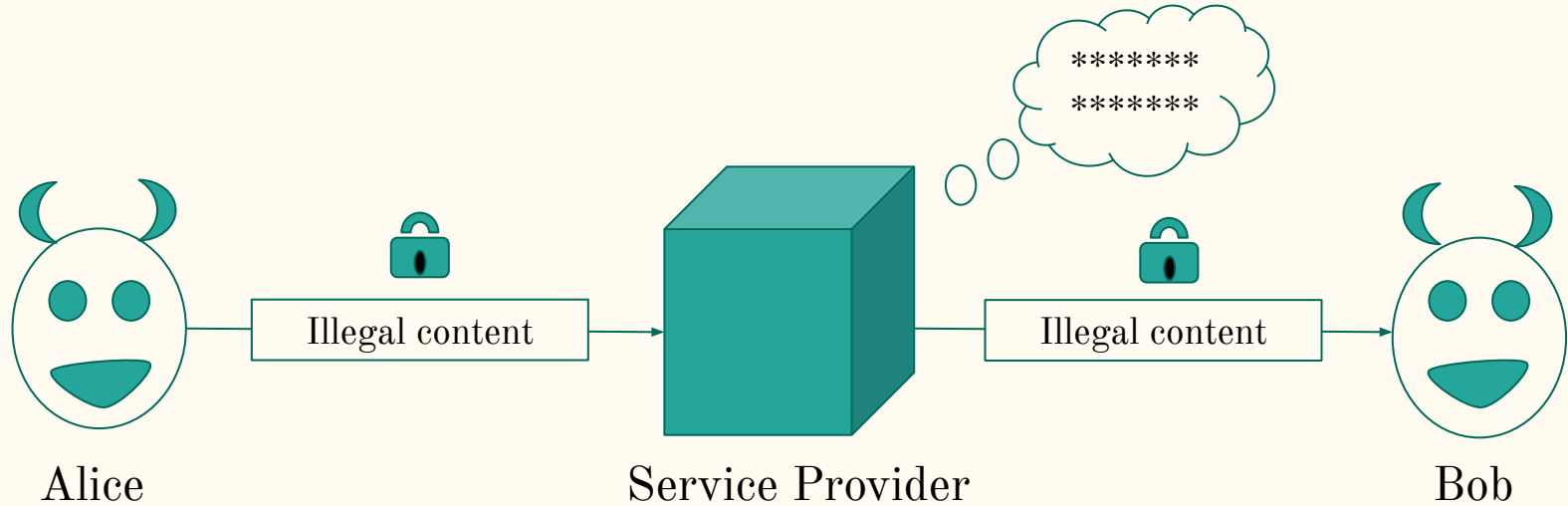
- No one but Alice and Bob – not even the service provider – can decrypt or read the message
- Eg: Signal, Matrix (Element), Session, WhatsApp, Telegram

(The Dilemma of) End-to-End Encrypted Messaging



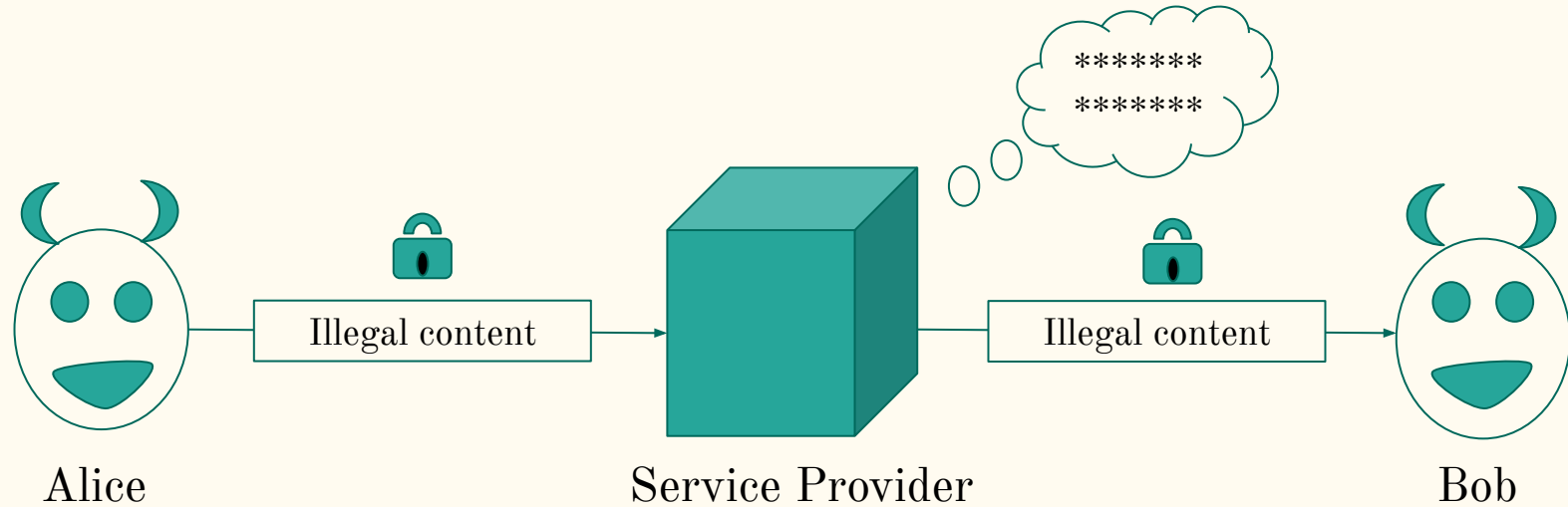
- No one but Alice and Bob – not even the service provider – can decrypt or read the *illegal* message

(The Dilemma of) End-to-End Encrypted Messaging



- No one but Alice and Bob – not even the service provider – can decrypt or read the *illegal* message (eg. misinformation or hate speech)

(The Dilemma of) End-to-End Encrypted Messaging



- No one but Alice and Bob – not even the service provider – can decrypt or read the *illegal* message (eg. misinformation or hate speech)
- Hence, law enforcement cannot regulate misdemeanors on these platforms

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- **India's IT Rules**
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

India's IT Rules (2021)

India's IT Rules (2021)

(2) A significant social media intermediary providing services primarily in the nature of messaging shall enable the identification of the first originator of the information on its computer resource as may be required by a judicial order passed by a court of competent jurisdiction or an order passed under section 69 by the Competent Authority as per the Information Technology (Procedure and Safeguards for interception, monitoring and decryption of information) Rules, 2009, which shall be supported with a copy of such information in electronic form:

India's IT Rules (2021)

(2) A significant social media intermediary providing services primarily in the nature of messaging shall enable the identification of the first originator of the information on its computer resource as may be required by a judicial order passed by a court of competent jurisdiction or an order passed under section 69 by the Competent Authority as per the Information Technology (Procedure and Safeguards for interception, monitoring and decryption of information) Rules, 2009, which shall be supported with a copy of such information in electronic form:

- Government of India wants to trace originator of reported message in E2EE clients

India's IT Rules (2021)

(2) A significant social media intermediary providing services primarily in the nature of messaging shall enable the identification of the first originator of the information on its computer resource as may be required by a judicial order passed by a court of competent jurisdiction or an order passed under section 69 by the Competent Authority as per the Information Technology (Procedure and Safeguards for interception, monitoring and decryption of information) Rules, 2009, which shall be supported with a copy of such information in electronic form:

- Government of India wants to trace originator of reported message in E2EE clients
- Outcome: End-to-end encryption broken?! 🤖

India's IT Rules (2021)

WhatsApp to Delhi HC: Will shut down in India if told to break encryption

TOI Tech Desk / TIMESOFINDIA.COM / Updated: Apr 26, 2024, 11:17 IST

WhatsApp faces shutdown if encryption compromised, protecting user privacy. India's largest market, it challenges IT Rules 2021 for violating privacy rights. Legal battle ongoing in Delhi High Court.

India's IT Rules (2021)

WhatsApp to Delhi HC: Will shut down in India if told to break encryption

TOI Tech Desk / TIMESOFINDIA.COM / Updated: Apr 26, 2024, 11:17 IST

WhatsApp faces shutdown if encryption compromised, protecting user privacy. India's largest market, it challenges IT Rules 2021 for violating privacy rights. Legal battle ongoing in Delhi High Court.

WhatsApp shutdown threat in India: 4 Reasons government says Whatsapp needs to 'follow' IT rules

TOI Tech Desk / TIMESOFINDIA.COM / Updated: May 2, 2024, 10:31 IST

WhatsApp and Facebook challenge IT Rules in Delhi High Court, facing opposition from MeitY. Issues include user rights, fake messages, and accountability to global norms on secondary liability for platforms.

India's IT Rules (2021)

WhatsApp to Delhi HC: Will shut down in India if told to break encryption

TOI Tech Desk / TIMESOFINDIA.COM / Updated: Apr 26, 2024, 11:17 IST

WhatsApp faces shutdown if encryption compromised, protecting user privacy. India's largest market, it challenges IT Rules 2021 for violating privacy rights. Legal battle ongoing in Delhi High Court.

WhatsApp, Meta move Delhi High Court against India's IT rules

A lawyer representing WhatsApp told the Delhi High Court that if the messaging service is forced to break encryption, the service will not be able to function in India

Updated - April 26, 2024 01:21 pm IST

THE HINDU BUREAU

WhatsApp shutdown threat in India: 4 Reasons government says Whatsapp needs to 'follow' IT rules

TOI Tech Desk / TIMESOFINDIA.COM / Updated: May 2, 2024, 10:31 IST

WhatsApp and Facebook challenge IT Rules in Delhi High Court, facing opposition from MeitY. Issues include user rights, fake messages, and accountability to global norms on secondary liability for platforms.

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- **Private Originator Tracing - Overview**
- Related Work
- Security Goals
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

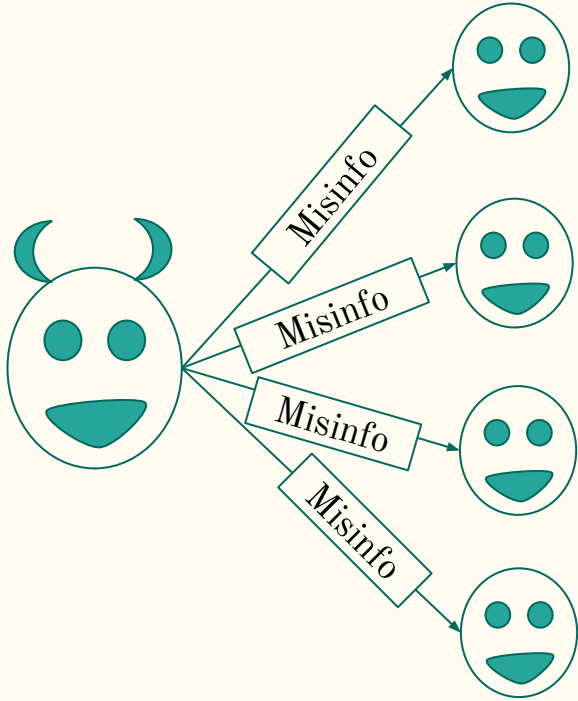
Private Originator Tracing - Overview



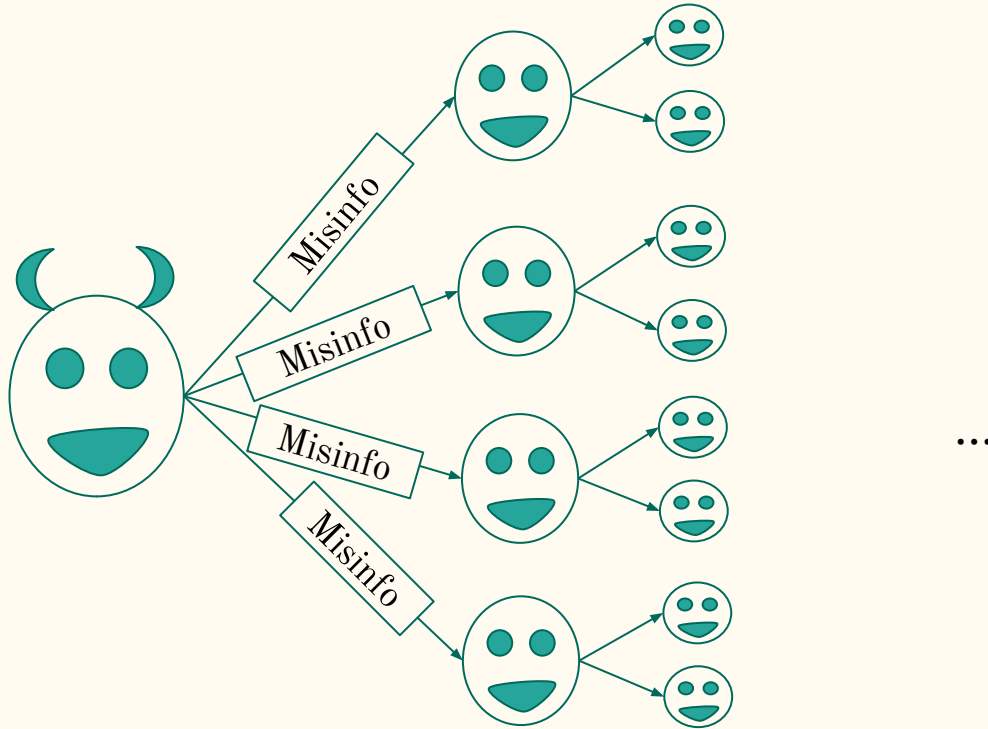
Private Originator Tracing - Overview



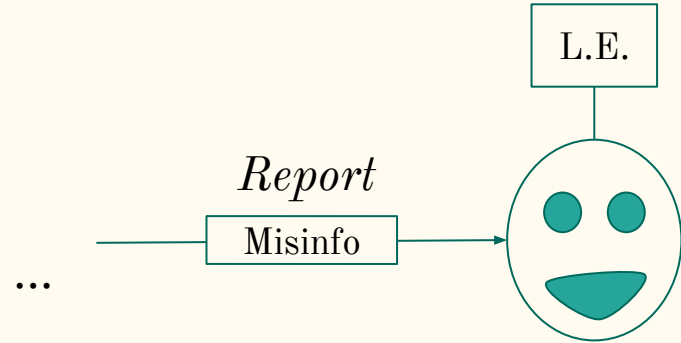
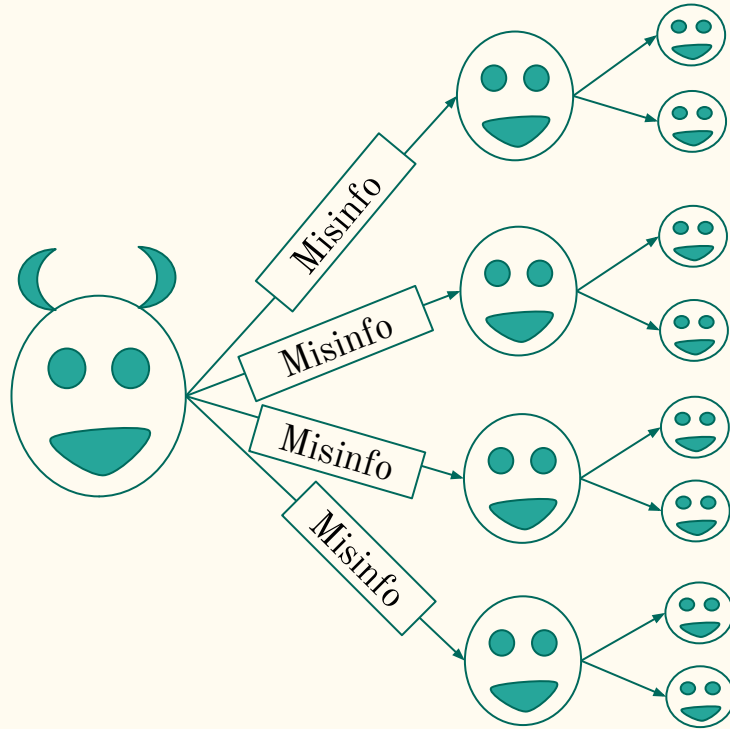
Private Originator Tracing - Overview



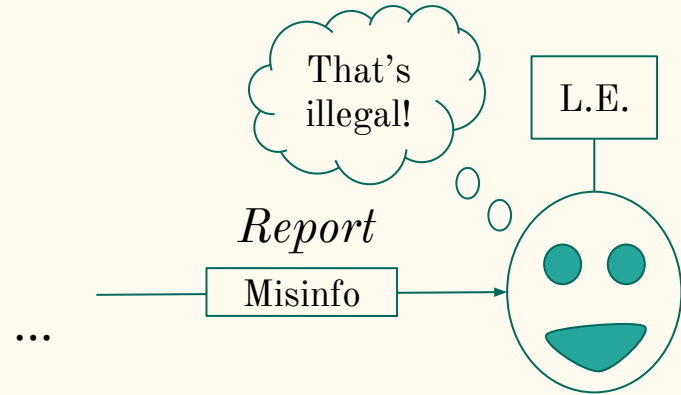
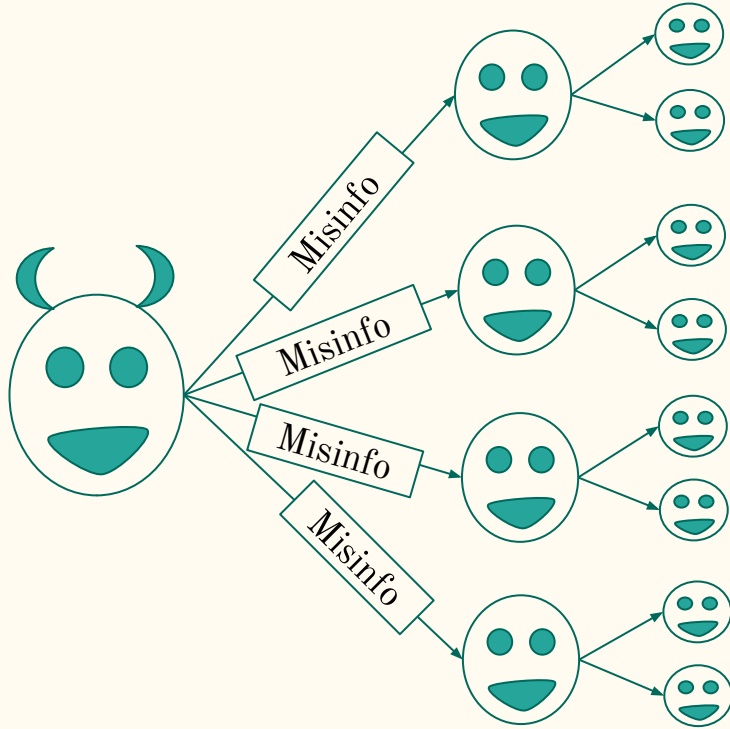
Private Originator Tracing - Overview



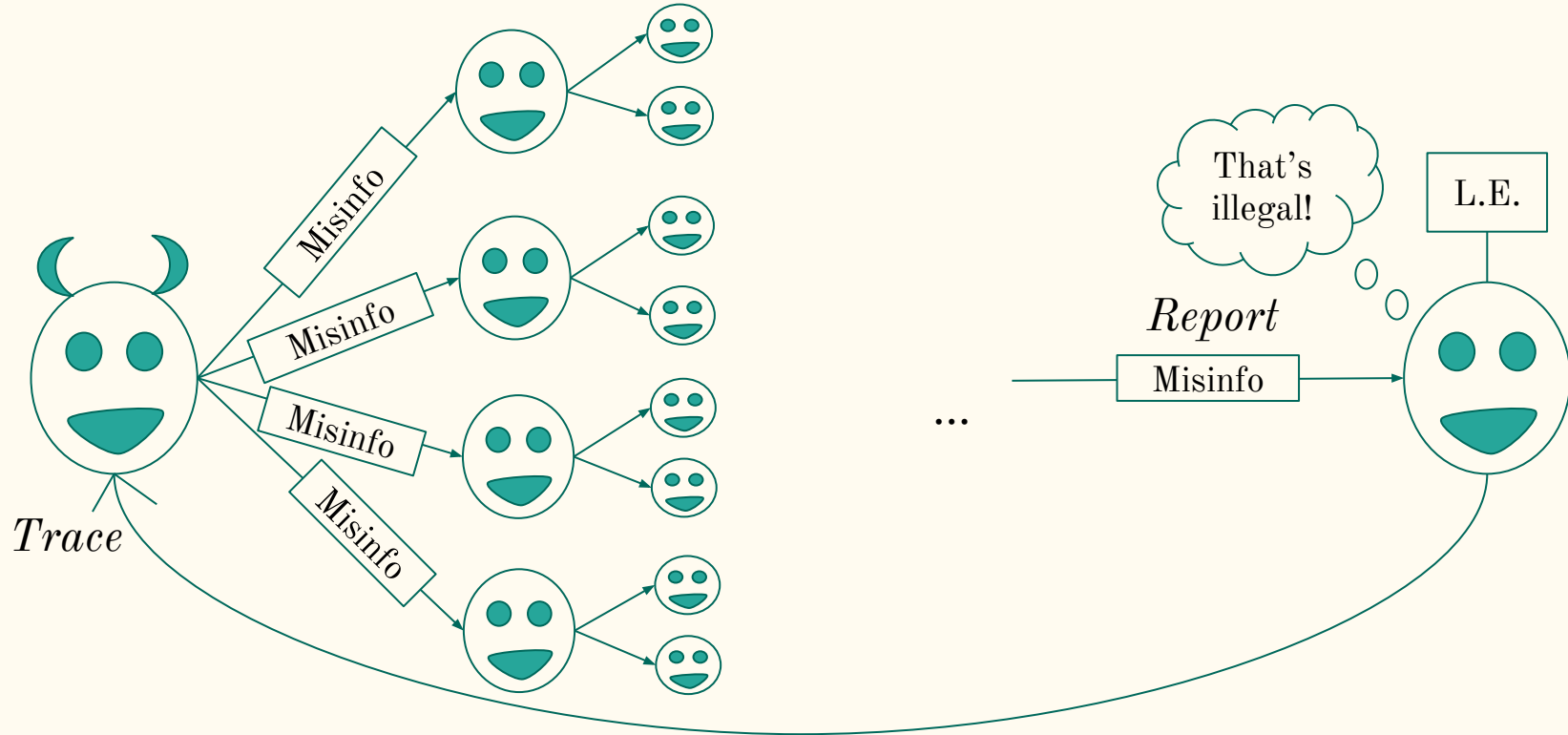
Private Originator Tracing - Overview



Private Originator Tracing - Overview



Private Originator Tracing - Overview



Private Originator Tracing - Overview

- Do NOT break end-to-end encryption at any stage

Private Originator Tracing - Overview

- Do NOT break end-to-end encryption at any stage
- Do NOT violate the privacy of any intermediate party of a forwarding chain

Private Originator Tracing - Overview

- Do NOT break end-to-end encryption at any stage
- Do NOT violate the privacy of any intermediate party of a forwarding chain
- Do NOT trace originators of messages not deemed illegal

Private Originator Tracing - Overview

- Do NOT break end-to-end encryption at any stage
- Do NOT violate the privacy of any intermediate party of a forwarding chain
- Do NOT trace originators of messages not deemed illegal
- Do NOT make messaging servers deviate from standard protocol

Private Originator Tracing - Overview

- Do NOT break end-to-end encryption at any stage
- Do NOT violate the privacy of any intermediate party of a forwarding chain
- Do NOT trace originators of messages not deemed illegal
- Do NOT make messaging servers deviate from standard protocol
- Do NOT make law enforcement have to do a lot of work

Private Originator Tracing - Overview

- Do NOT break end-to-end encryption at any stage
- Do NOT violate the privacy of any intermediate party of a forwarding chain
- Do NOT trace originators of messages not deemed illegal
- Do NOT make messaging servers deviate from standard protocol
- Do NOT complicate affairs for the end user

Private Originator Tracing - Overview

“Can we design a simple, secure, and lightweight protocol which identifies only the originator, induces a realistic workload on law enforcement authorities, is server-immutable, and preserves E2EE otherwise?”

Private Originator Tracing - Overview

“Can we design a simple, secure, and lightweight protocol which identifies only the originator, induces a realistic workload on law enforcement authorities, is server-immutable, and preserves E2EE otherwise?”



Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- **Security Goals**
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

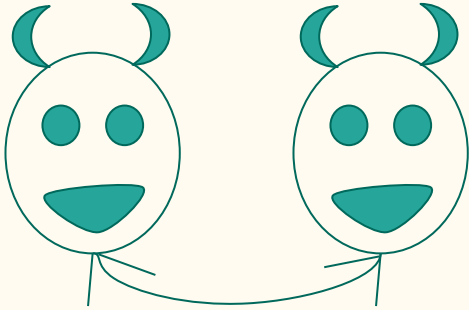
Security Goals

Threat Model:

Security Goals

Threat Model:

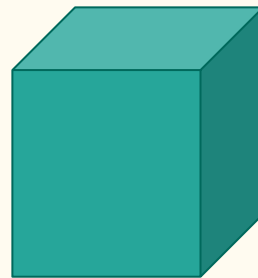
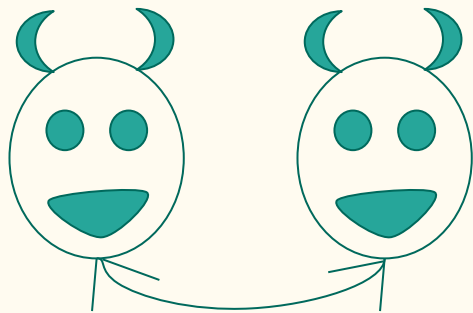
- *Malicious and colluding* users



Security Goals

Threat Model:

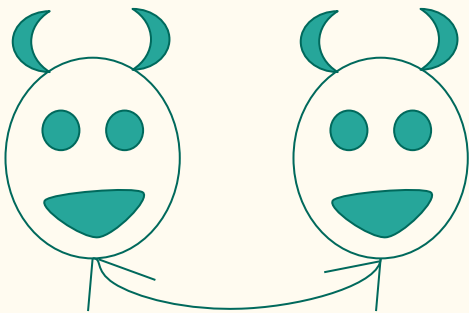
- *Malicious and colluding* users
- *Semi-honest* service provider



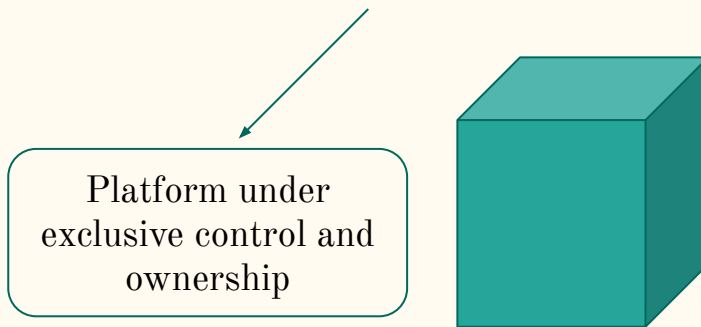
Security Goals

Threat Model:

- *Malicious and colluding* users



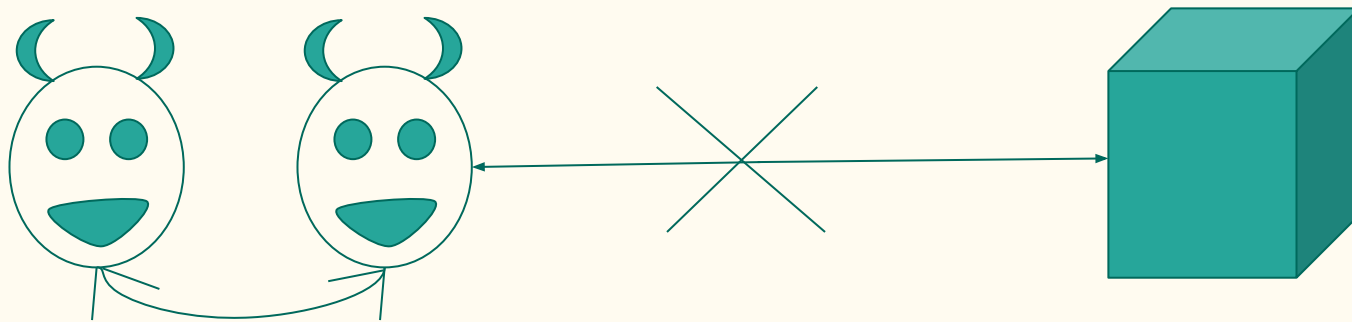
- *Semi-honest* service provider



Security Goals

Threat Model:

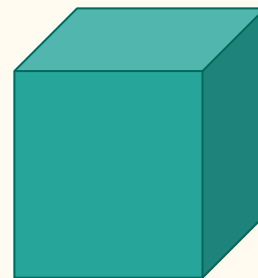
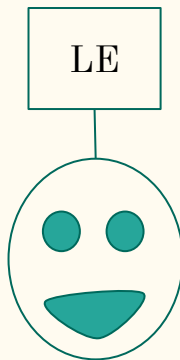
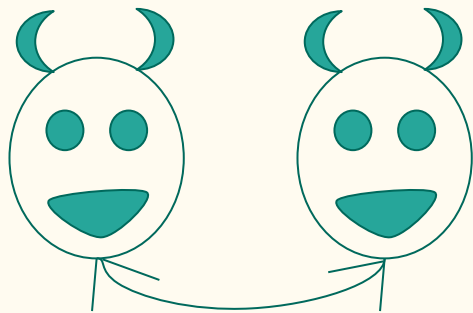
- *Malicious and colluding* users
- *Semi-honest* service provider



Security Goals

Threat Model:

- *Malicious and colluding* users
- *Semi-honest* law enforcement
- *Semi-honest* service provider

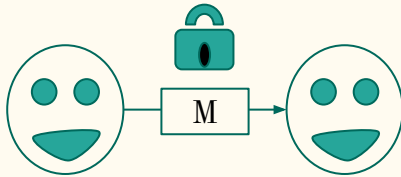


Security Goals

- **Confidentiality**

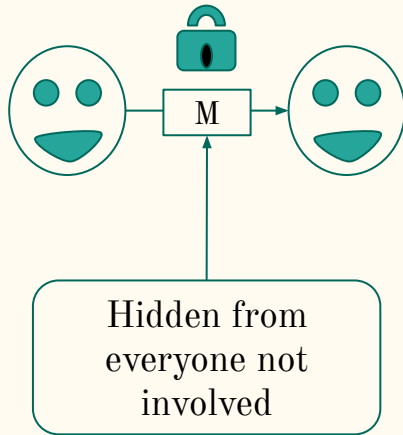
Security Goals

- **Confidentiality**



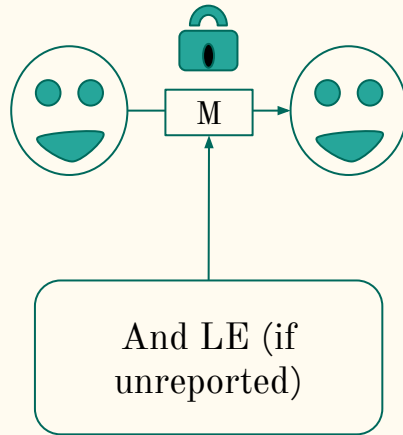
Security Goals

- **Confidentiality**



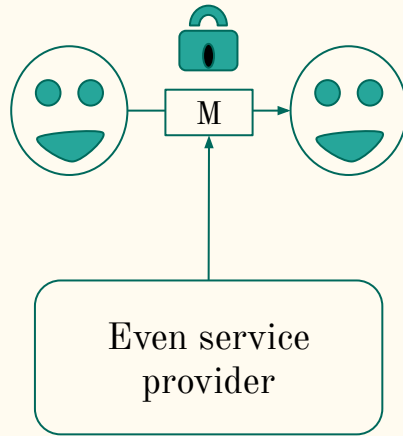
Security Goals

- **Confidentiality**



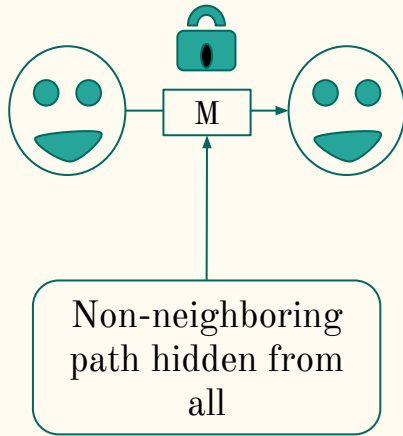
Security Goals

- **Confidentiality**



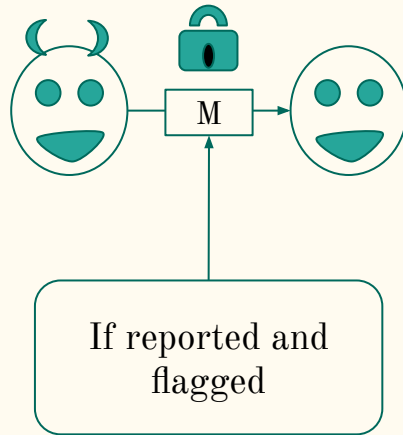
Security Goals

- **Confidentiality**



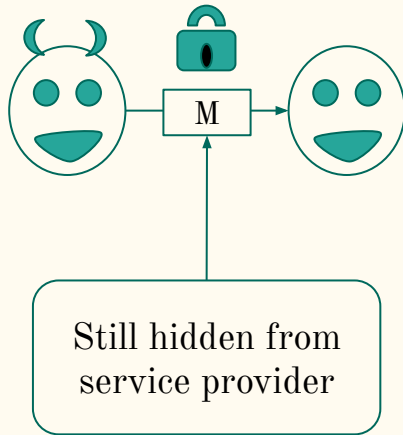
Security Goals

- **Confidentiality**



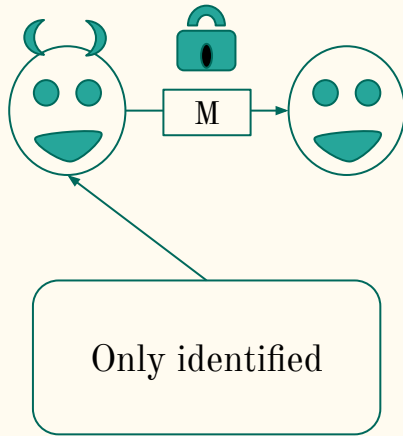
Security Goals

- **Confidentiality**



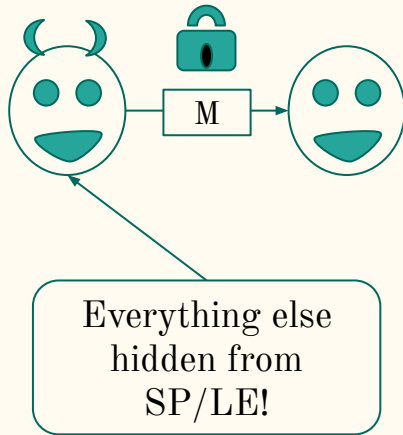
Security Goals

- **Confidentiality**



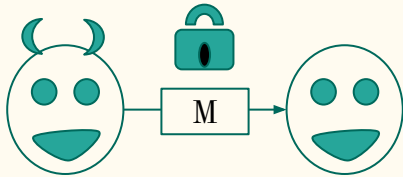
Security Goals

- **Confidentiality**



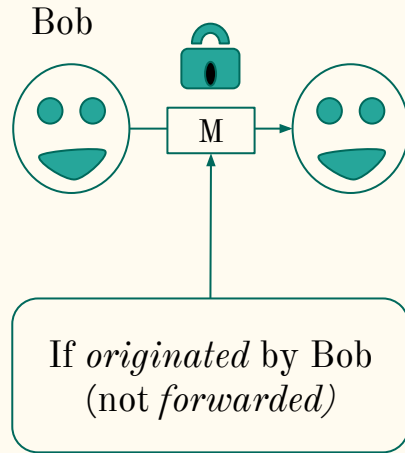
Security Goals

- Confidentiality
- **Accountability**



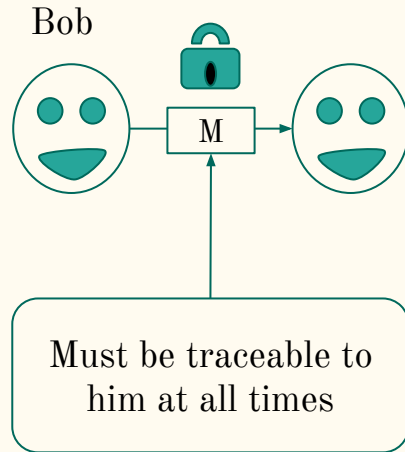
Security Goals

- Confidentiality
- **Accountability**



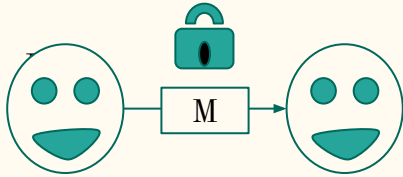
Security Goals

- Confidentiality
- **Accountability**



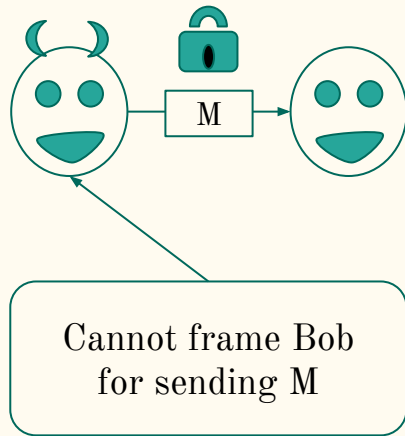
Security Goals

- Confidentiality
- Accountability
- **Unforgeability**



Security Goals

- Confidentiality
- Accountability
- **Unforgeability**

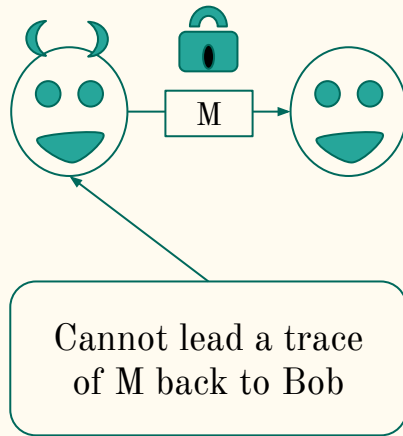


Bob



Security Goals

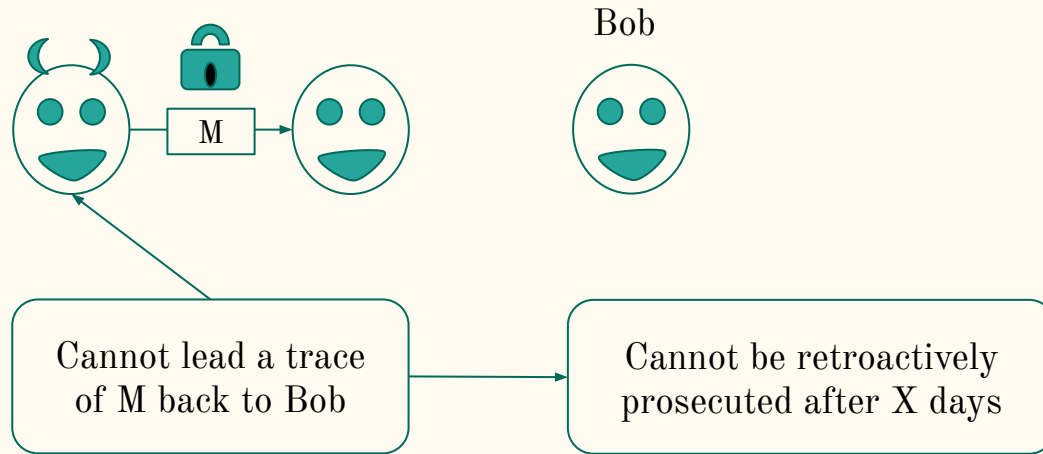
- Confidentiality
- Accountability
- **Unforgeability**



Bob

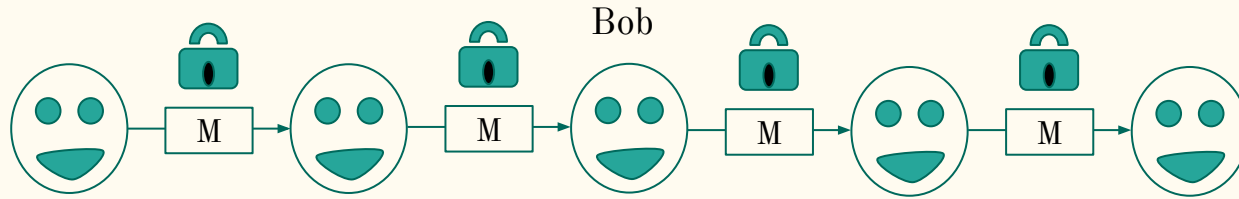
Security Goals

- Confidentiality
- Accountability
- **Unforgeability**



Security Goals

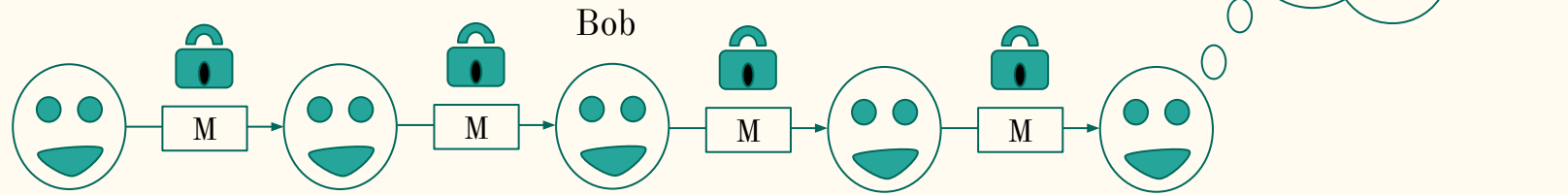
- Confidentiality
- Accountability
- Unforgeability
- **Deniability**



Security Goals

- Confidentiality
- Accountability
- Unforgeability

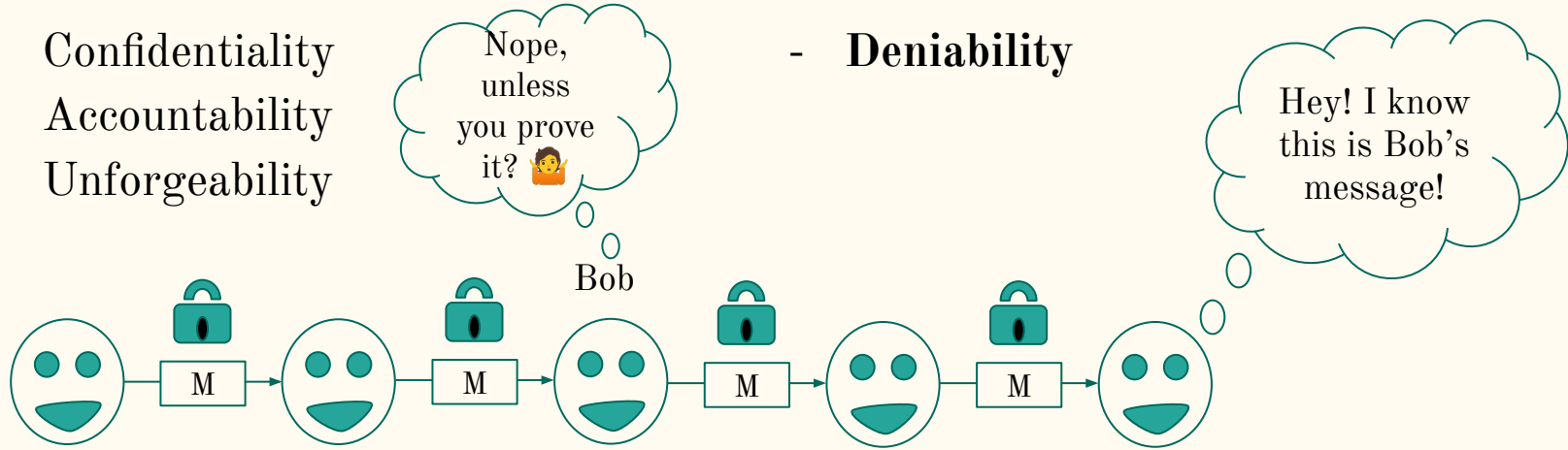
- **Deniability**



Security Goals

- Confidentiality
- Accountability
- Unforgeability

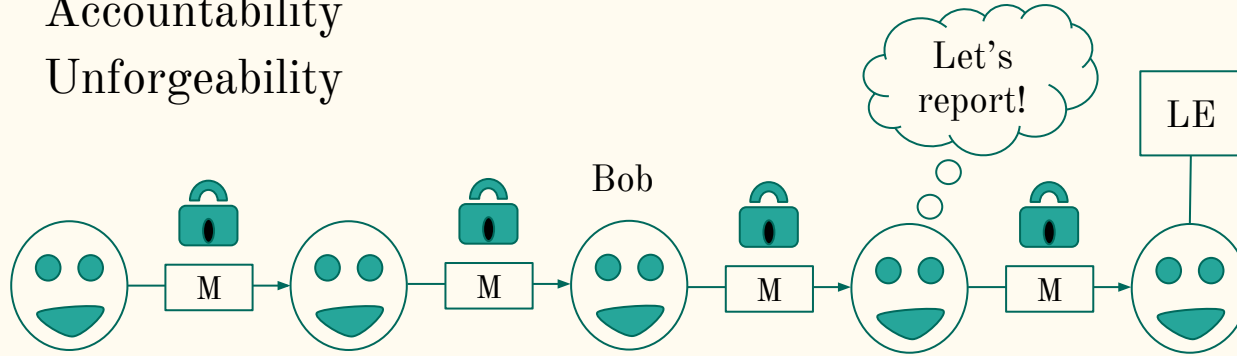
- Deniability



Security Goals

- Confidentiality
- Accountability
- Unforgeability

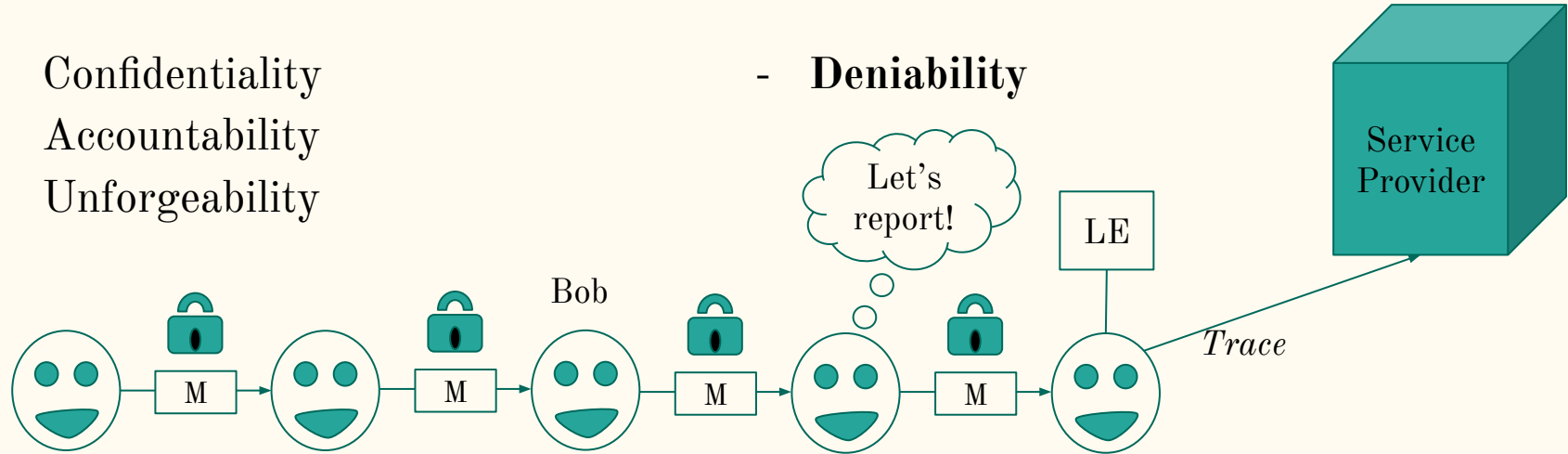
- **Deniability**



Security Goals

- Confidentiality
- Accountability
- Unforgeability

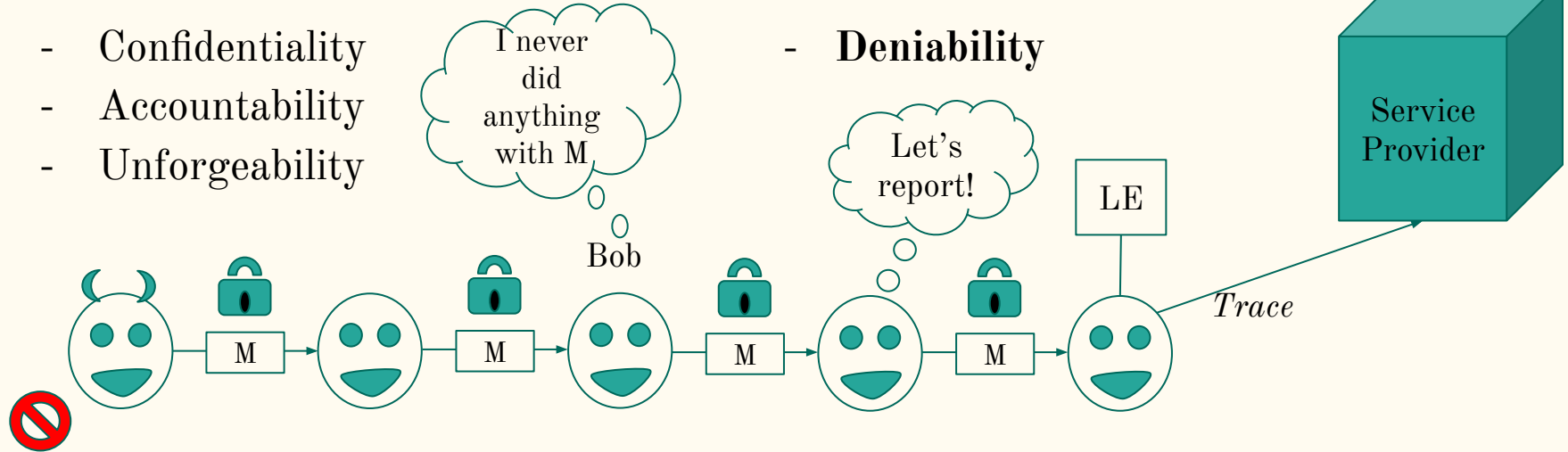
- Deniability



Security Goals

- Confidentiality
- Accountability
- Unforgeability

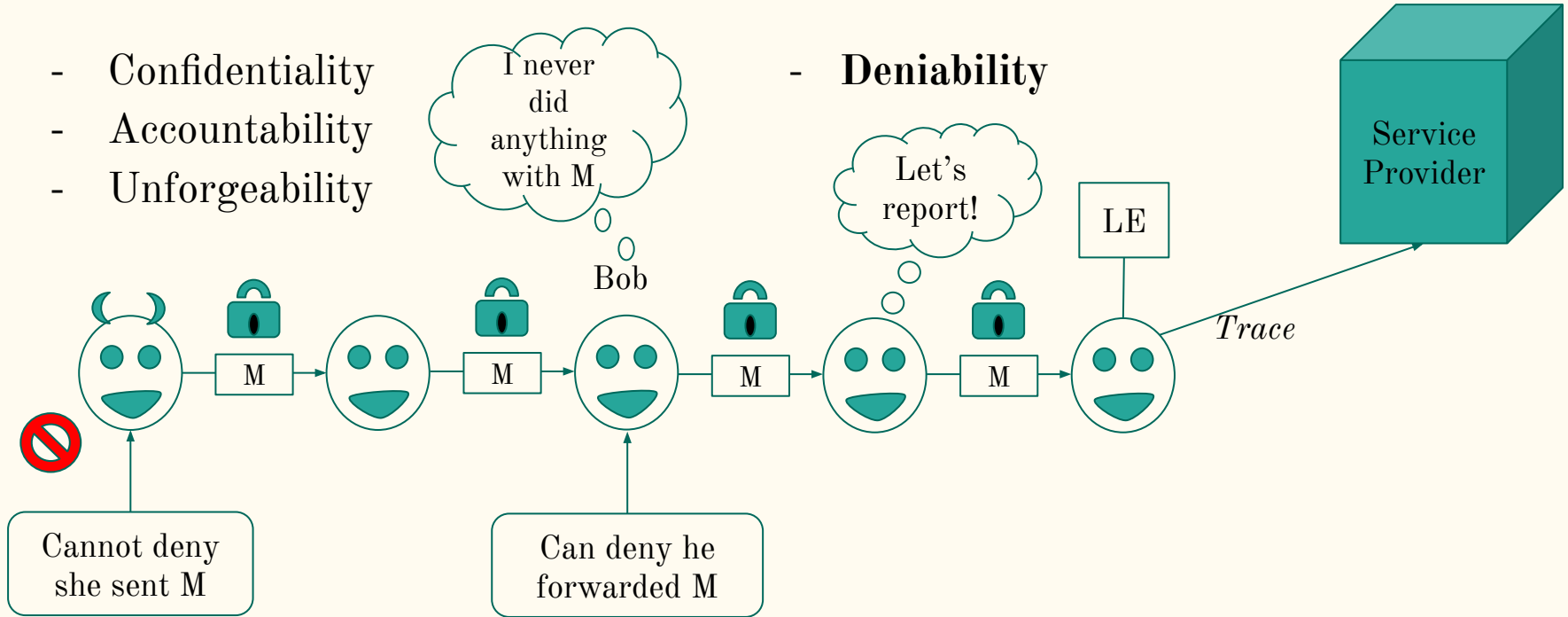
- Deniability



Security Goals

- Confidentiality
- Accountability
- Unforgeability

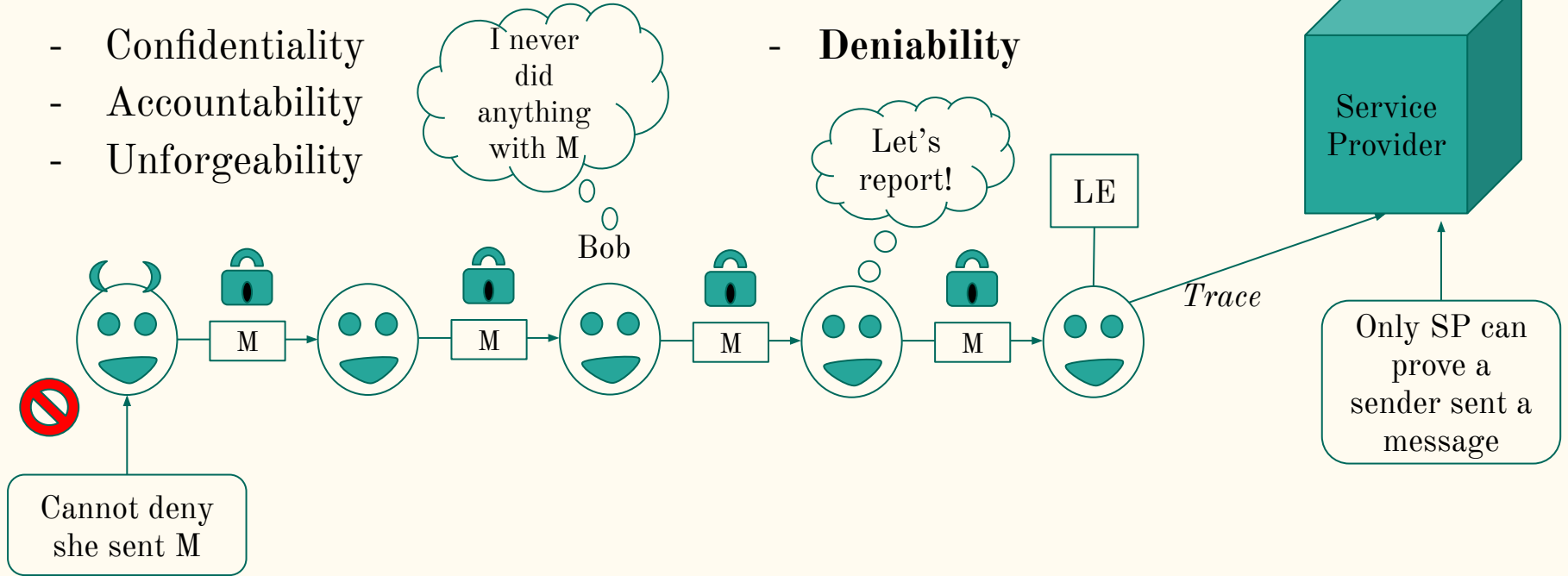
- Deniability



Security Goals

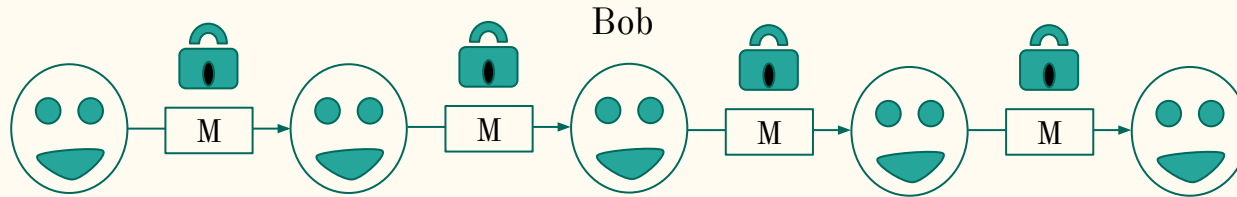
- Confidentiality
- Accountability
- Unforgeability

- Deniability



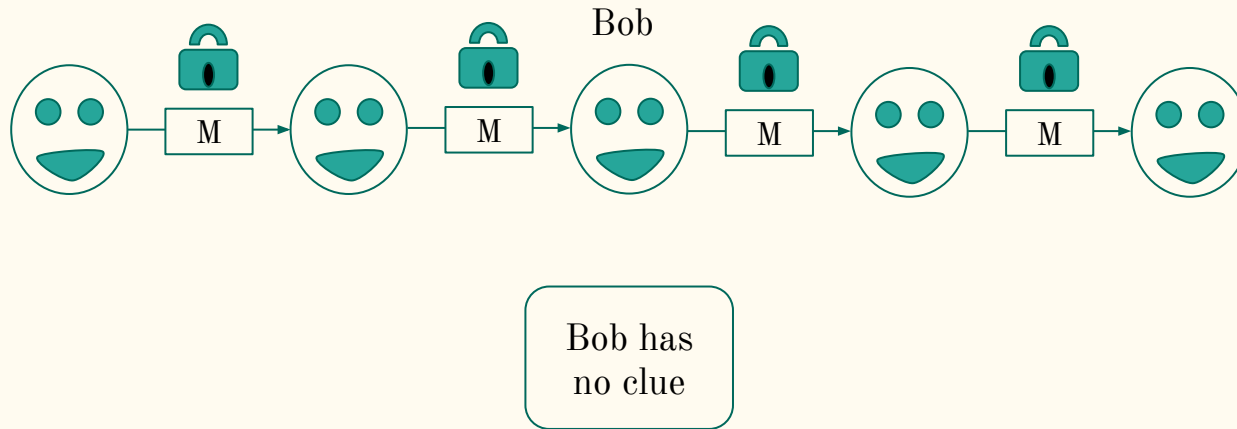
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- **Anonymity**



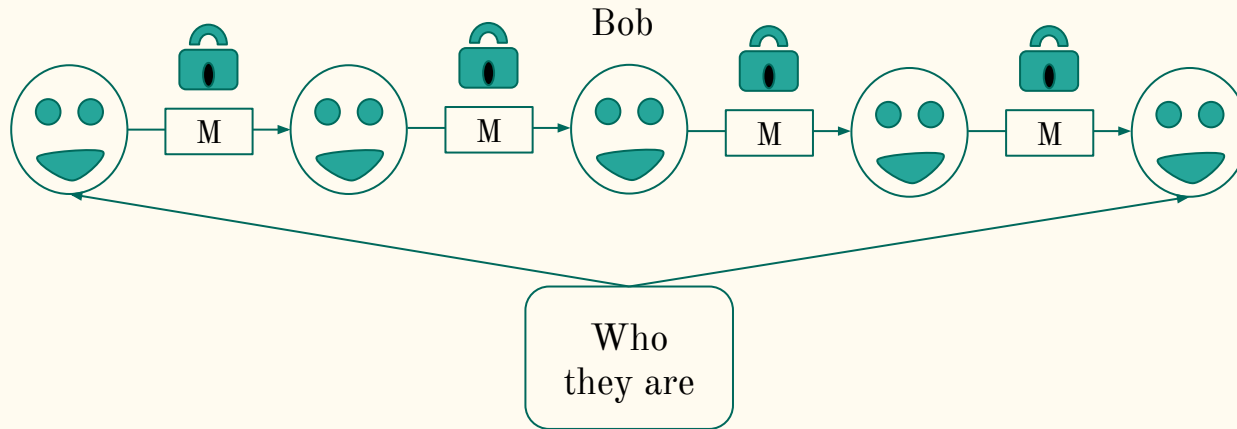
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- **Anonymity**



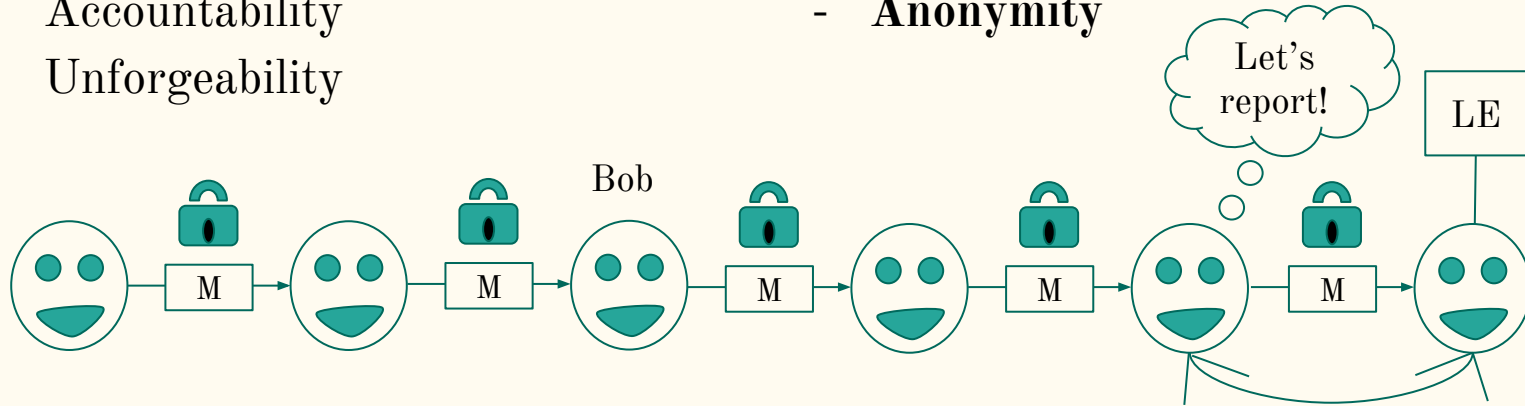
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- **Anonymity**



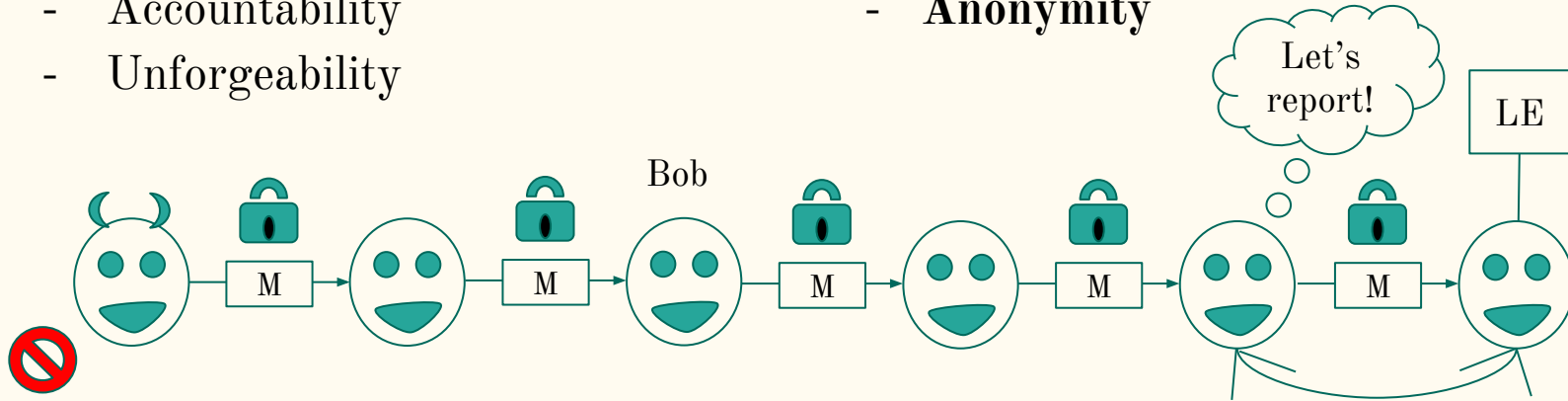
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- **Anonymity**



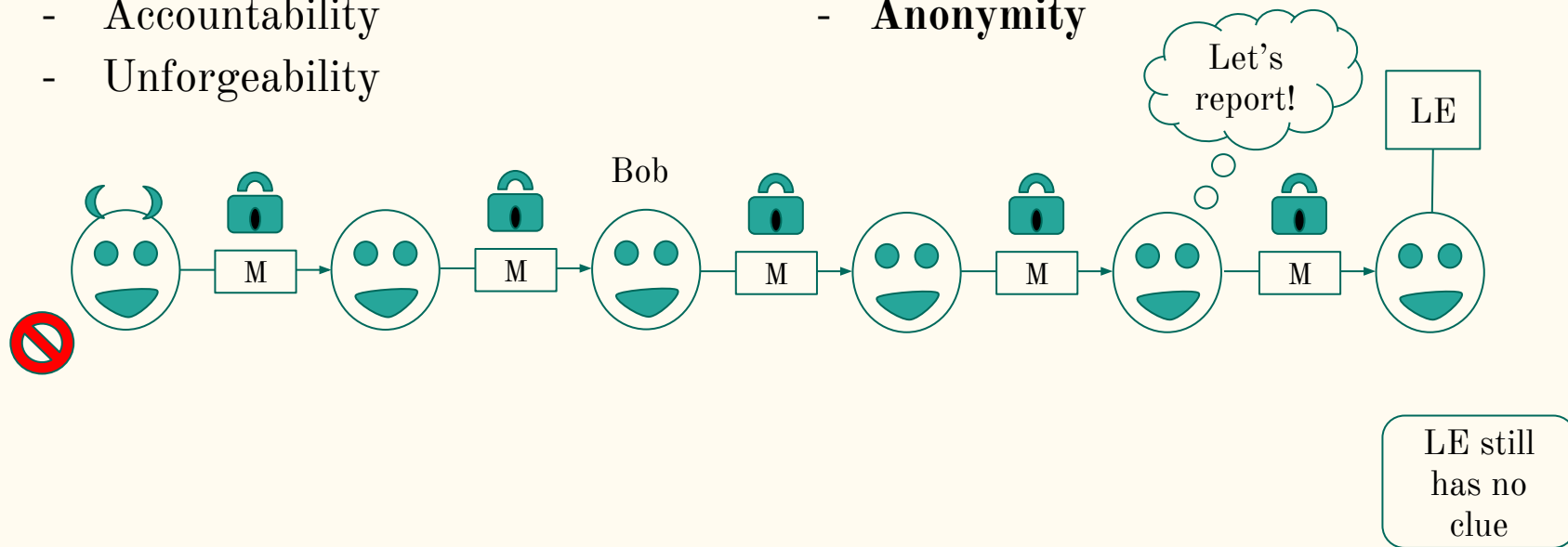
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- **Anonymity**



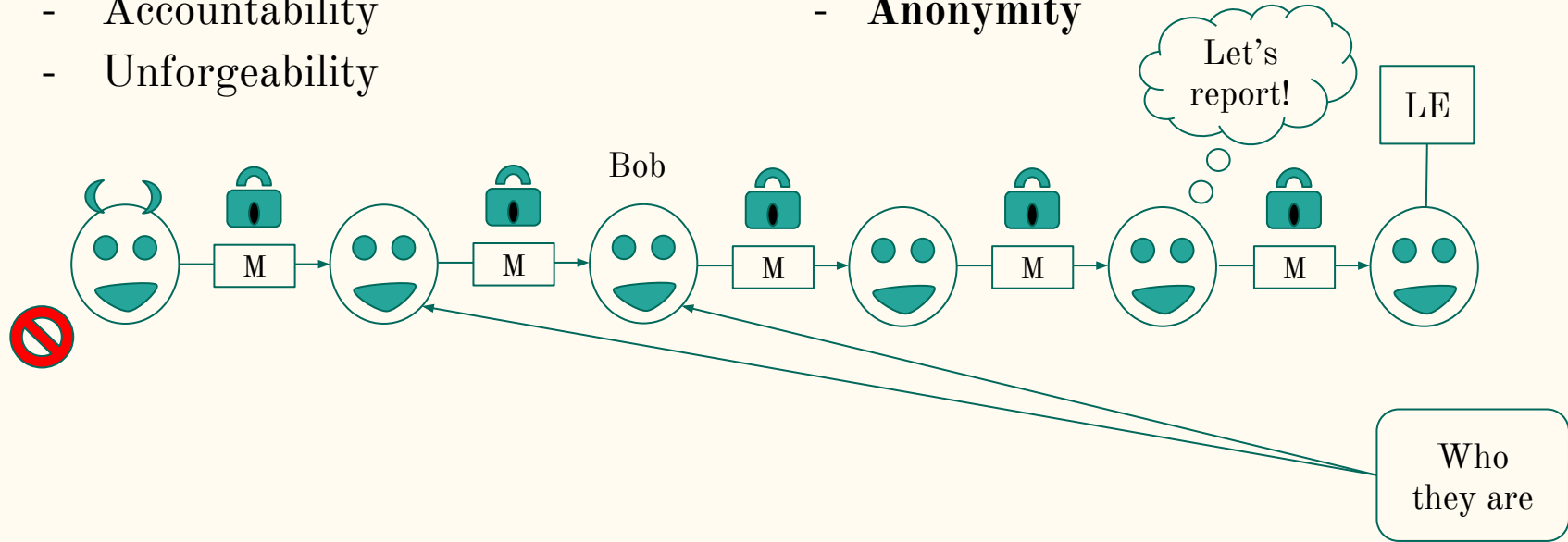
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- **Anonymity**



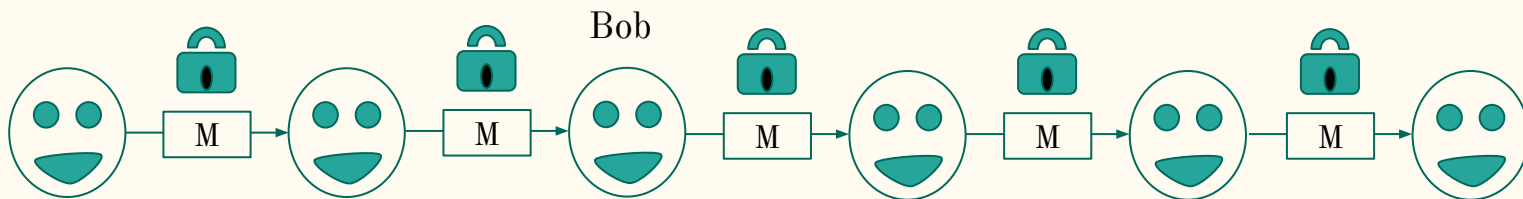
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- **Anonymity**



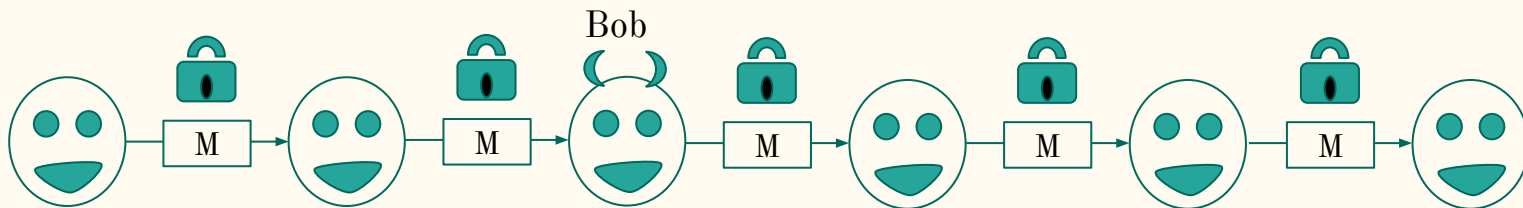
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- Anonymity
- **Forward/backward security**



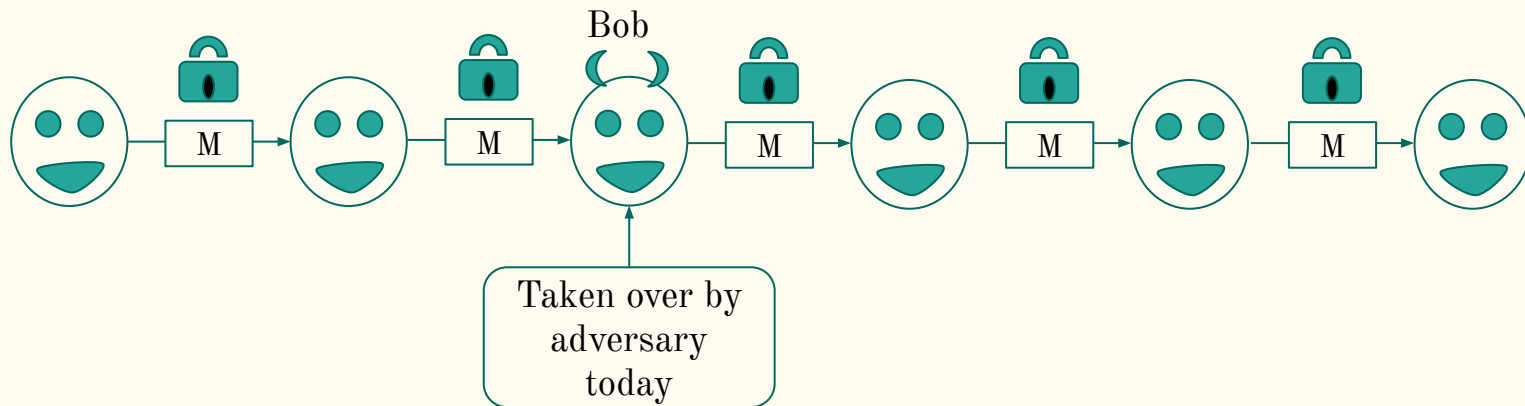
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- Anonymity
- **Forward/backward security**



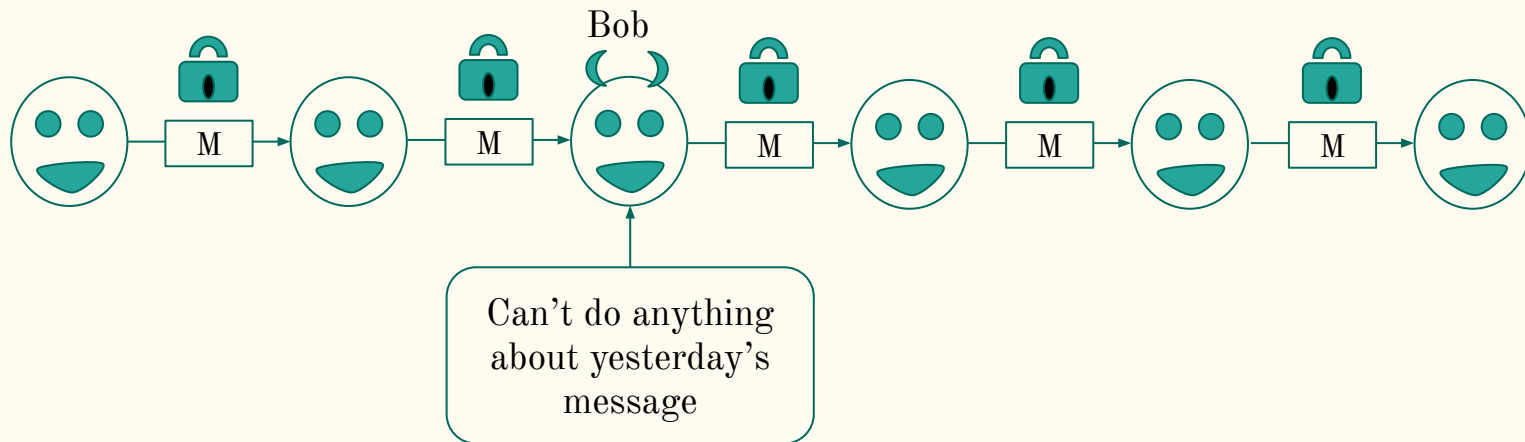
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- Anonymity
- **Forward/backward security**



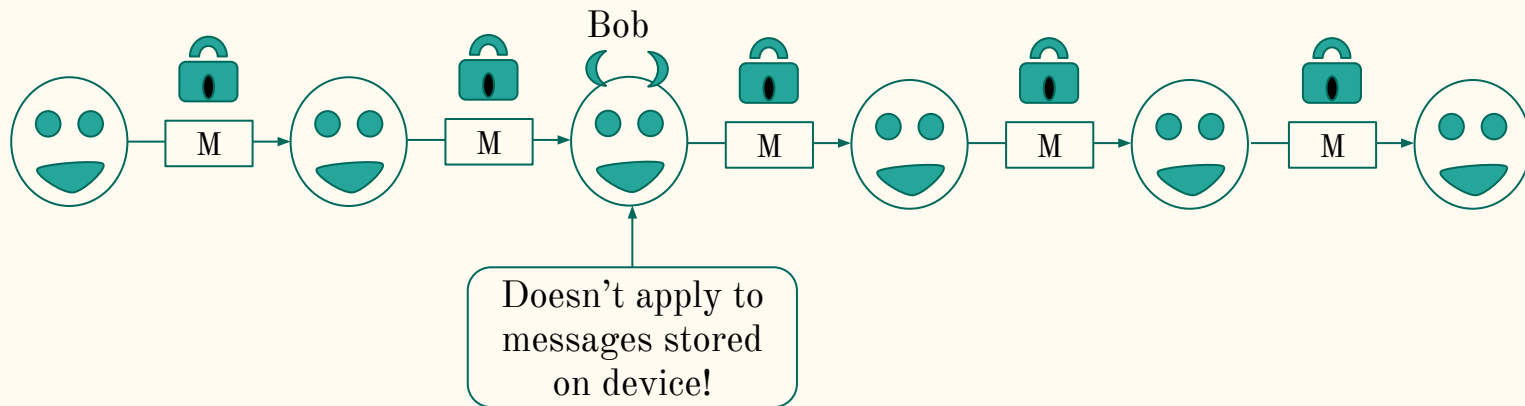
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- Anonymity
- **Forward/backward security**



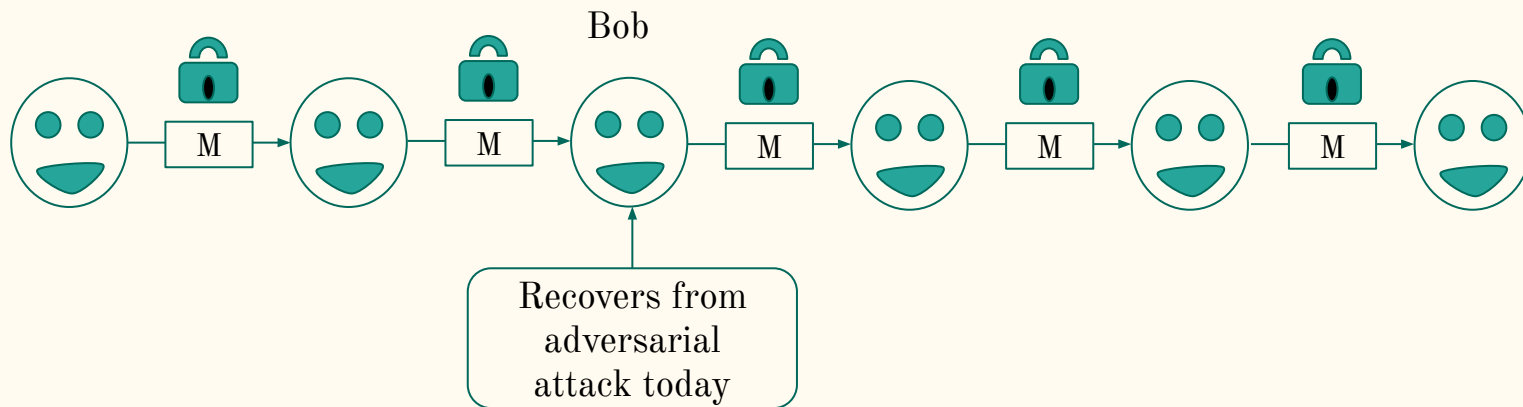
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- Anonymity
- **Forward/backward security**



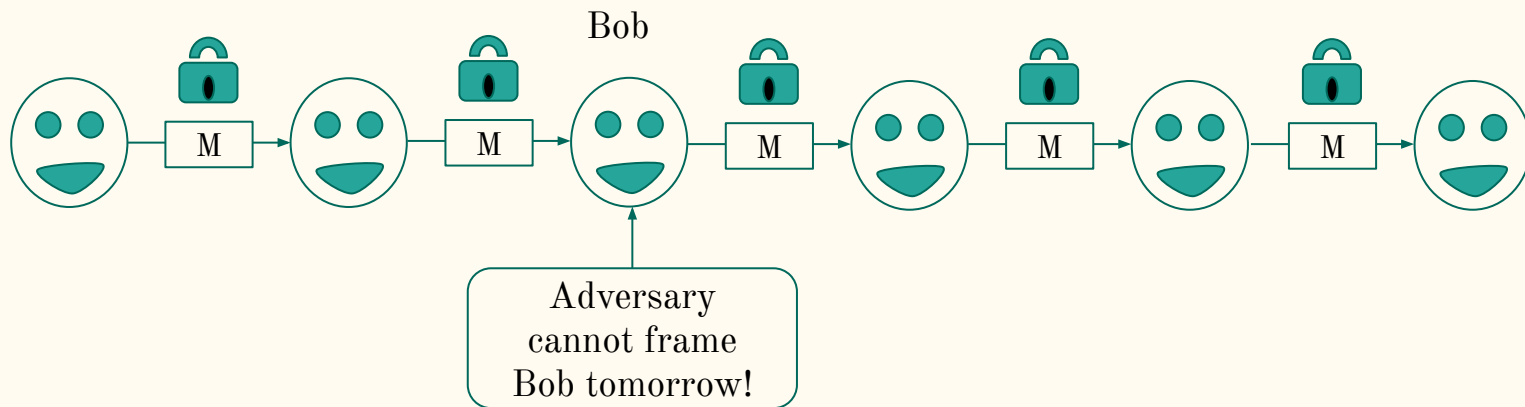
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- Anonymity
- Forward/**backward** security



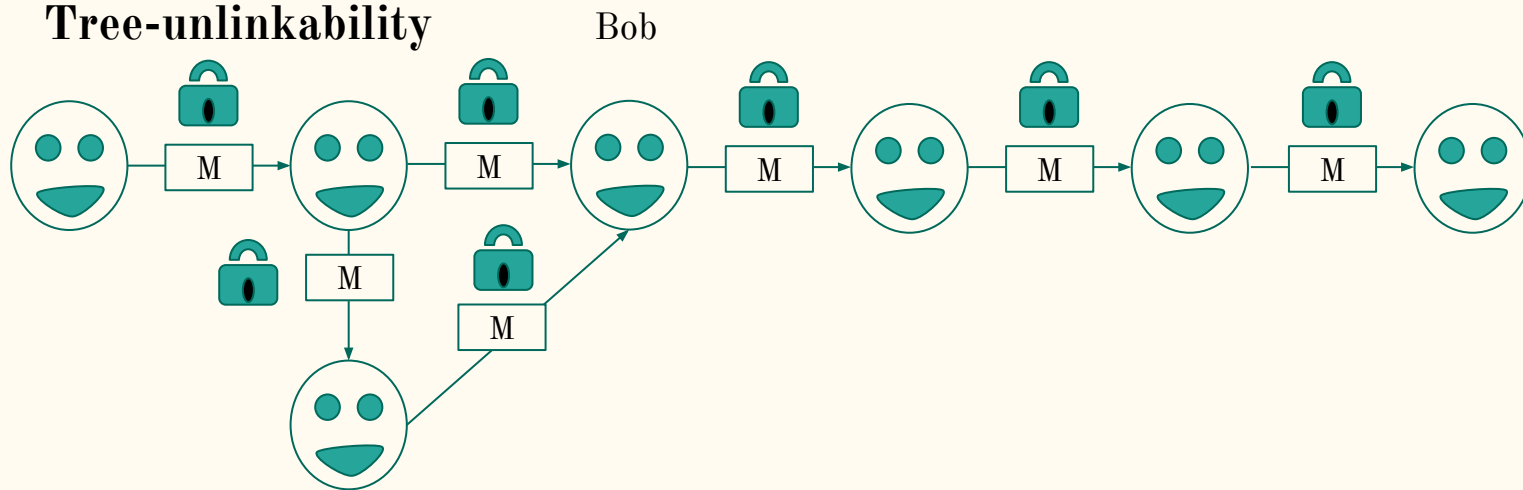
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- Deniability
- Anonymity
- Forward/**backward** security



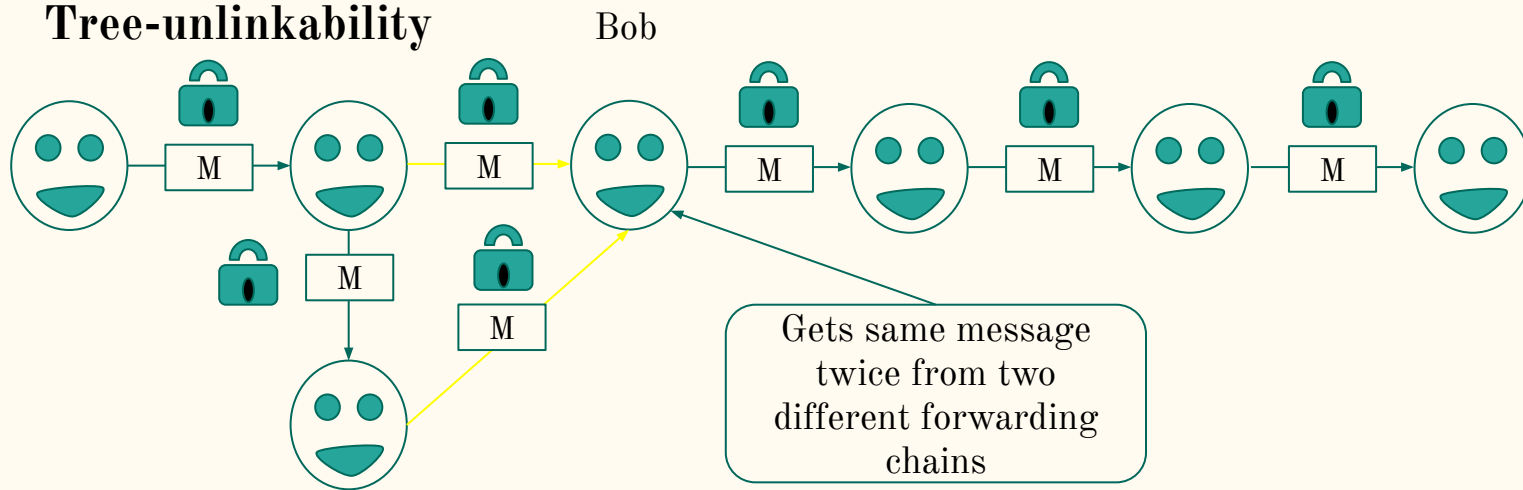
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- **Tree-unlinkability**
- Deniability
- Anonymity
- Forward/backward security



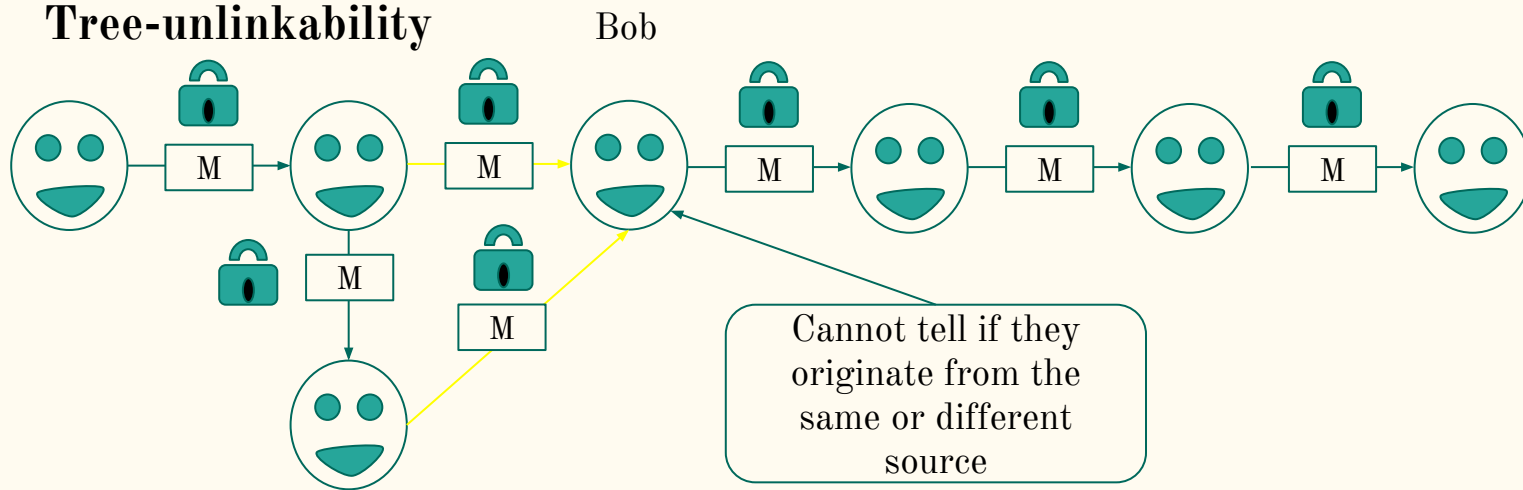
Security Goals

- Confidentiality
- Accountability
- Unforgeability
- **Tree-unlinkability**
- Deniability
- Anonymity
- Forward/backward security



Security Goals

- Confidentiality
- Accountability
- Unforgeability
- **Tree-unlinkability**
- Deniability
- Anonymity
- Forward/backward security



Plan for the afternoon

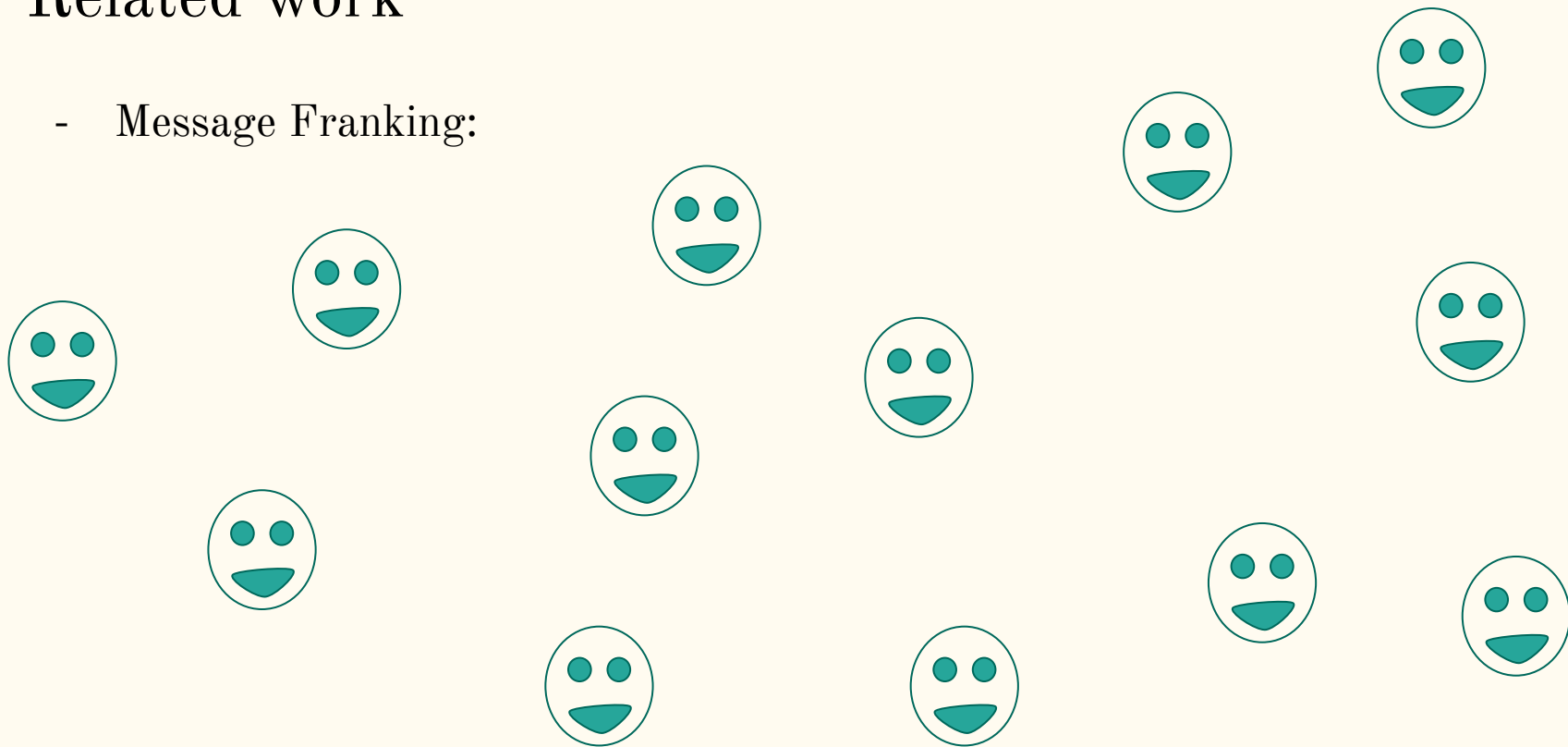
- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- **Related Work**
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

Related work

- Message Franking:

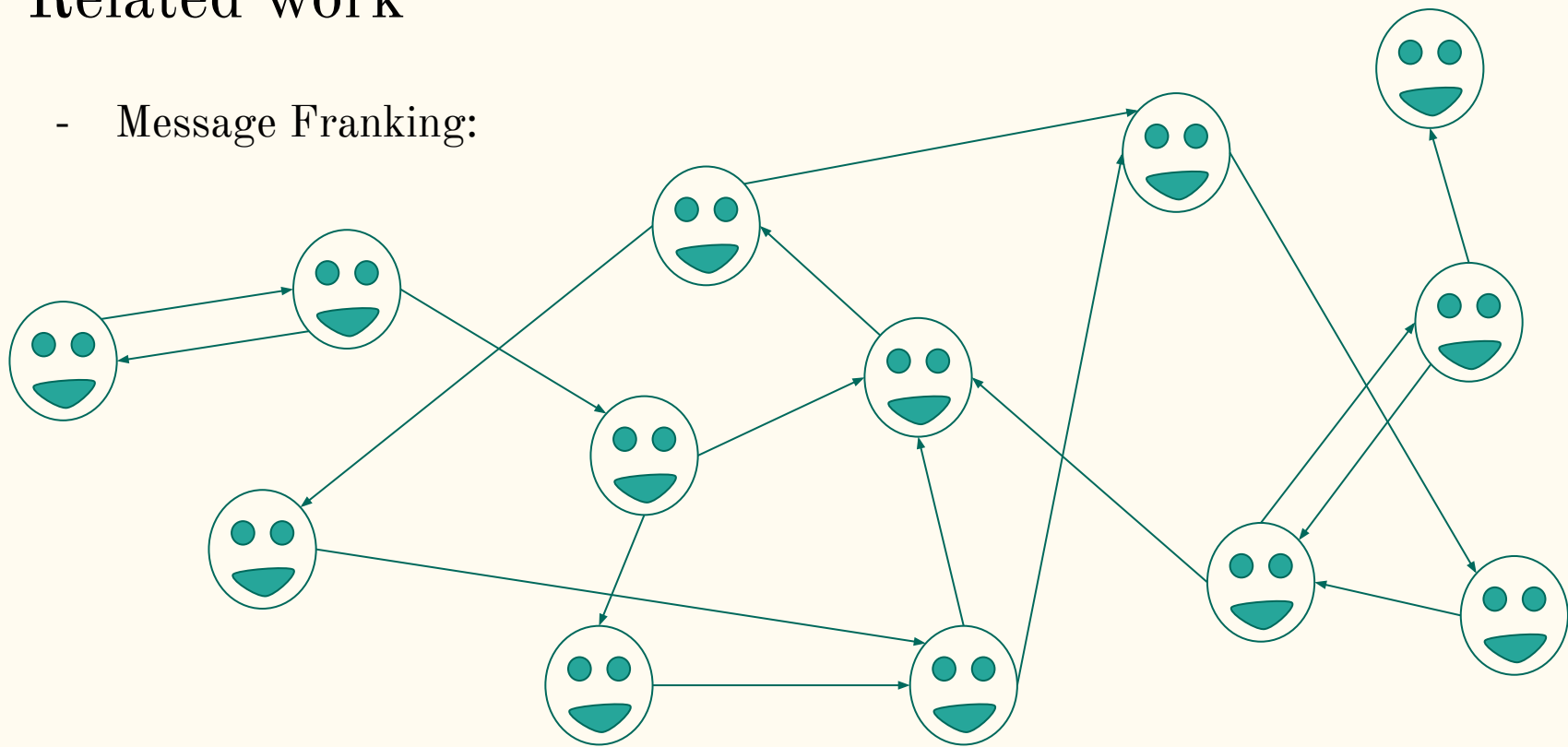
Related work

- Message Franking:



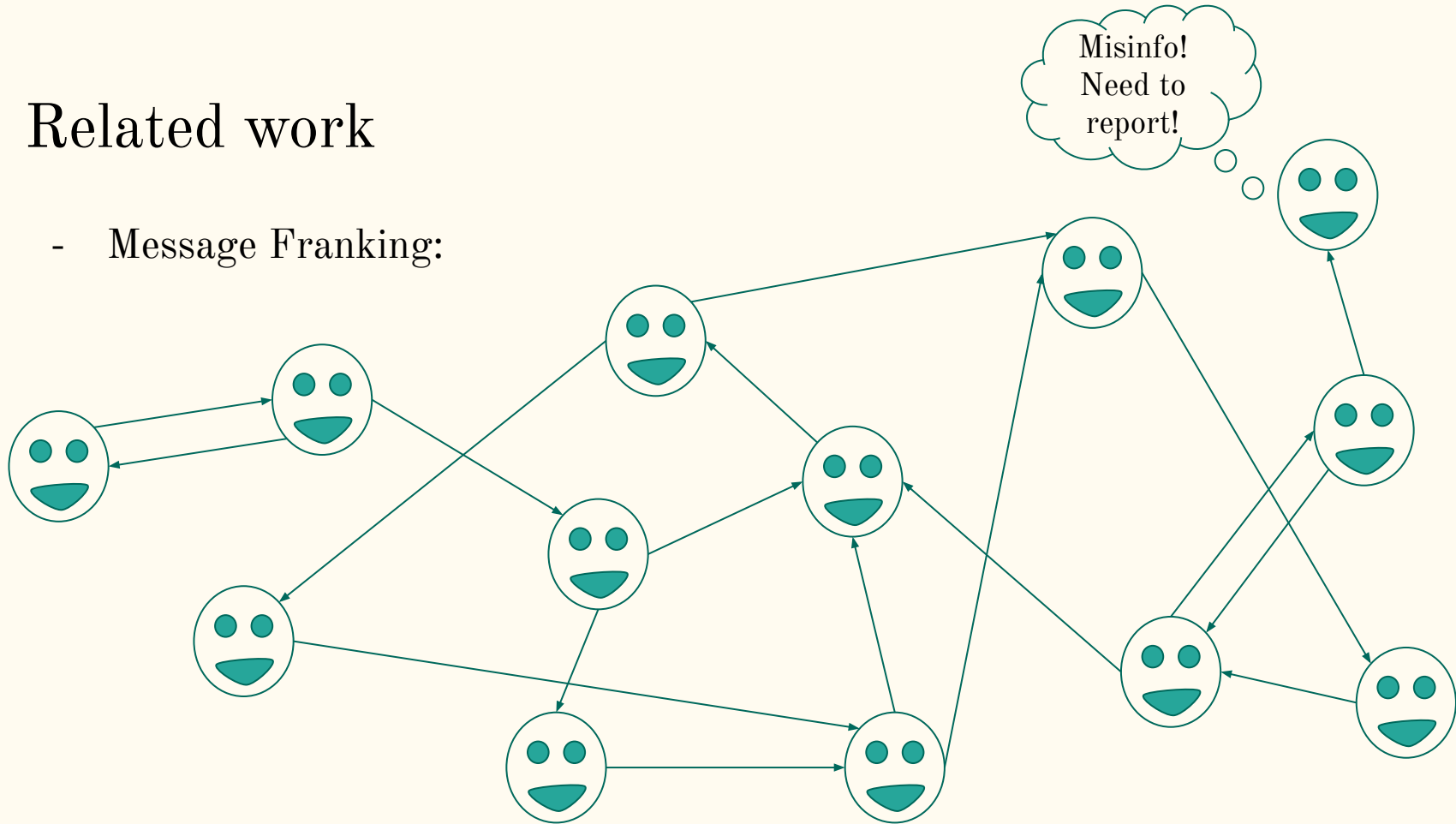
Related work

- Message Franking:



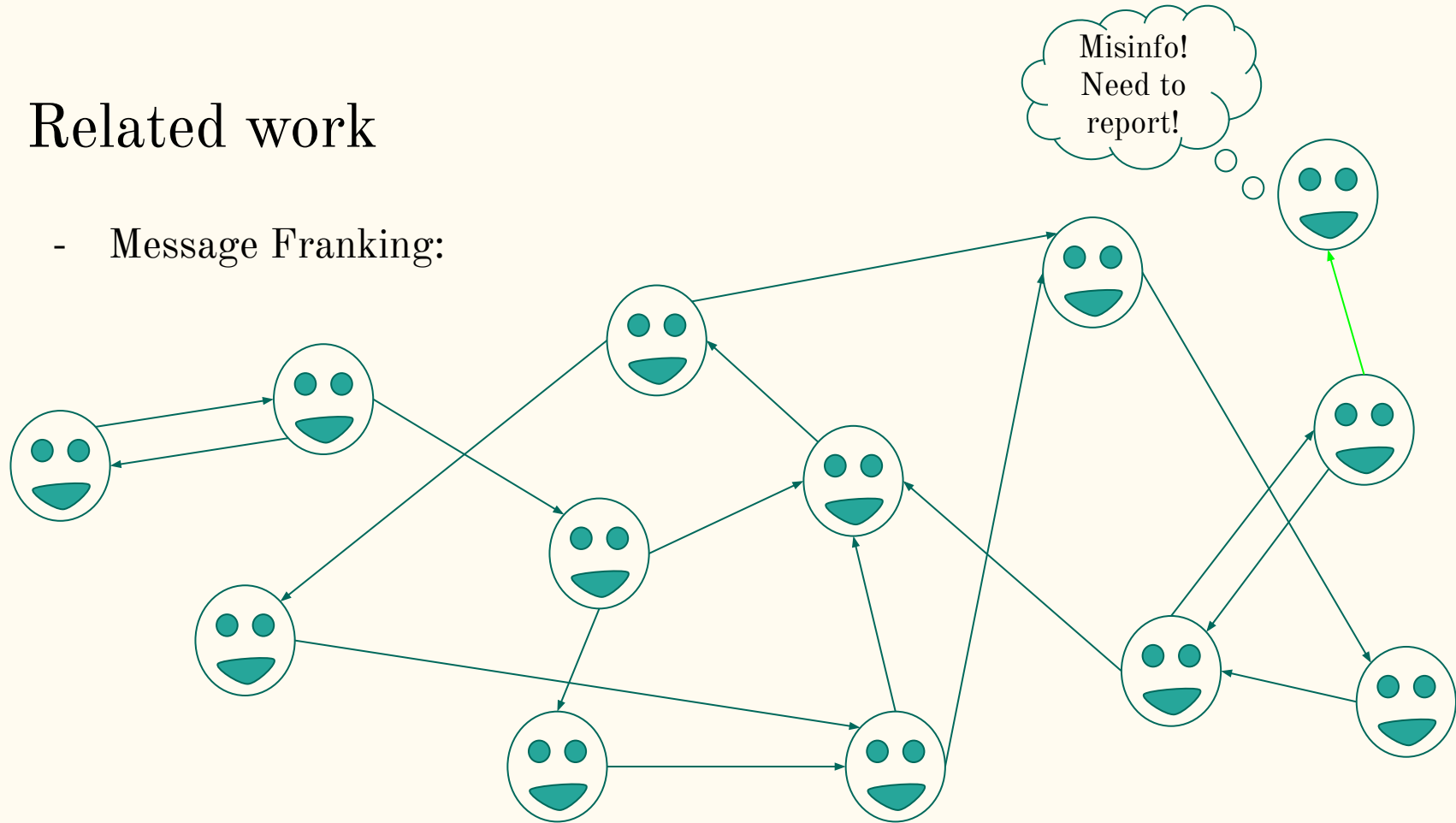
Related work

- Message Franking:



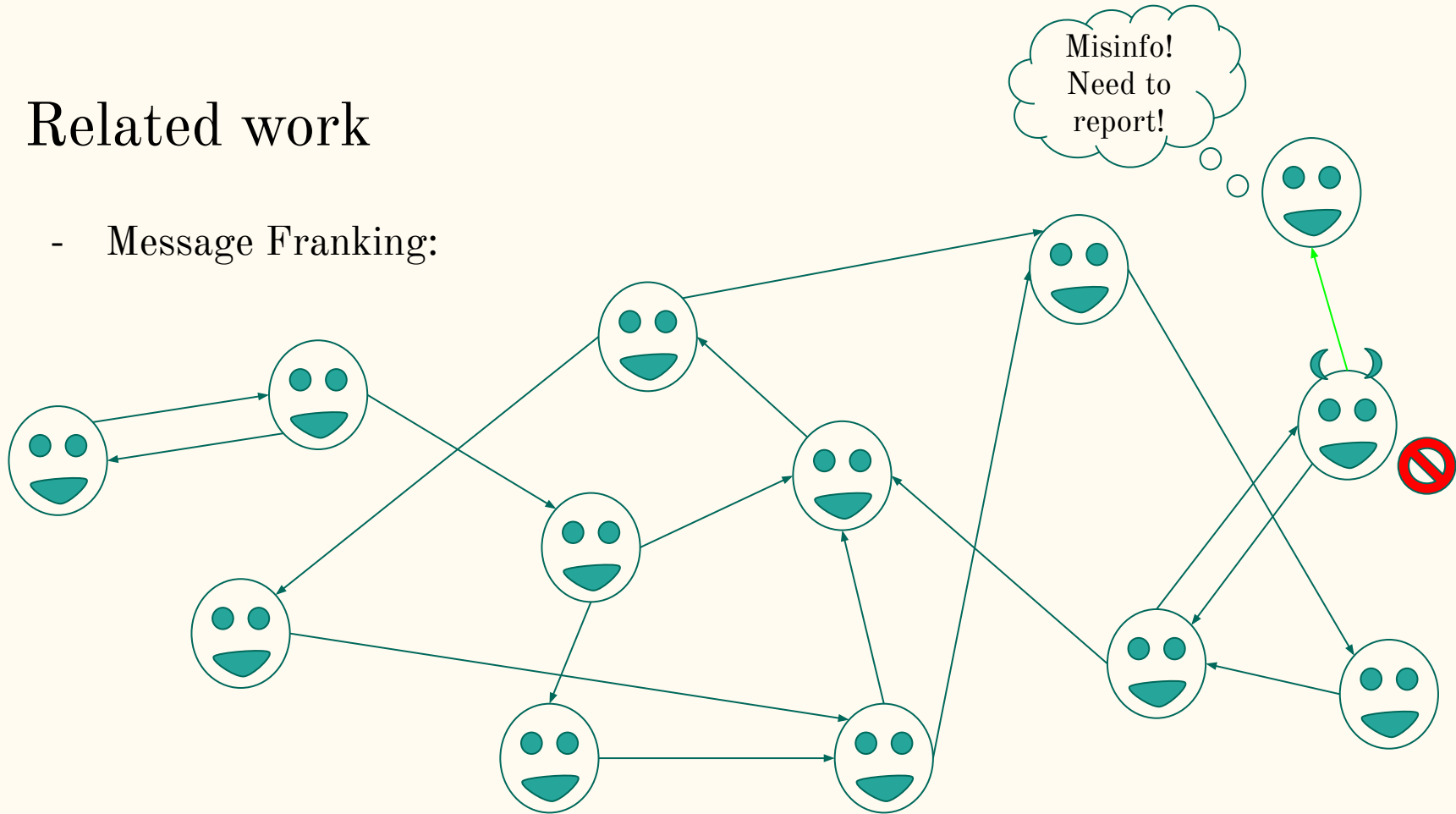
Related work

- Message Franking:



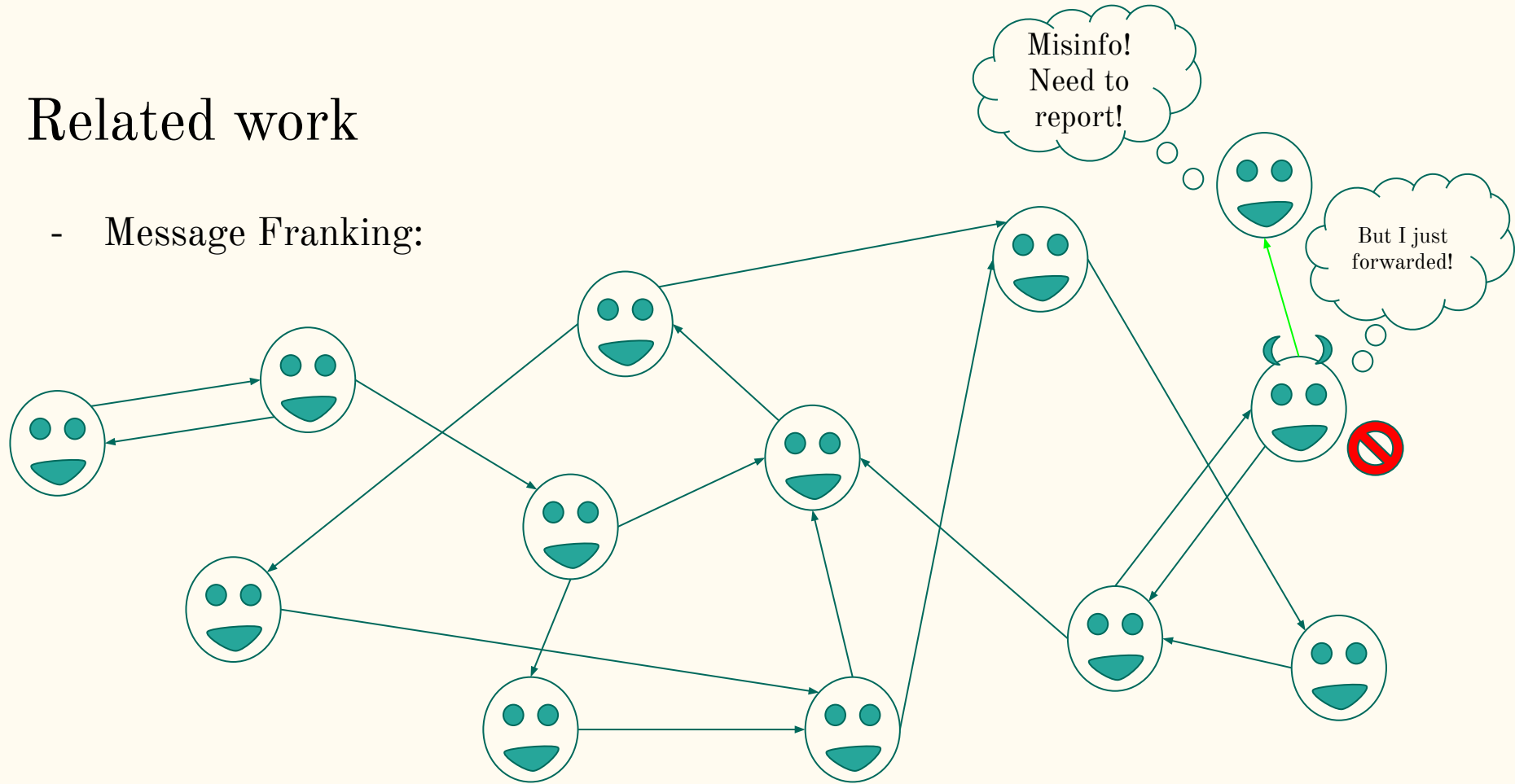
Related work

- Message Franking:



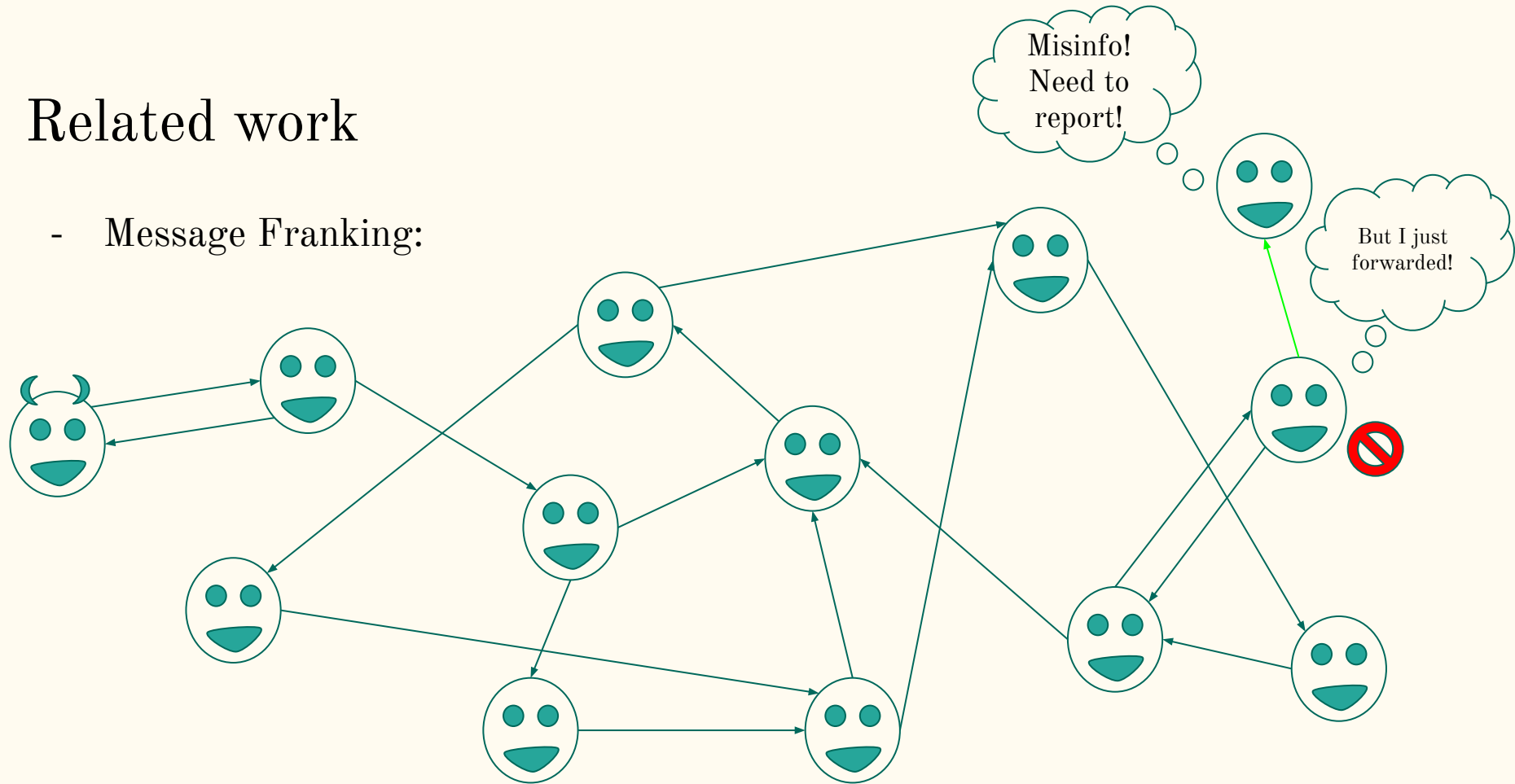
Related work

- Message Franking:



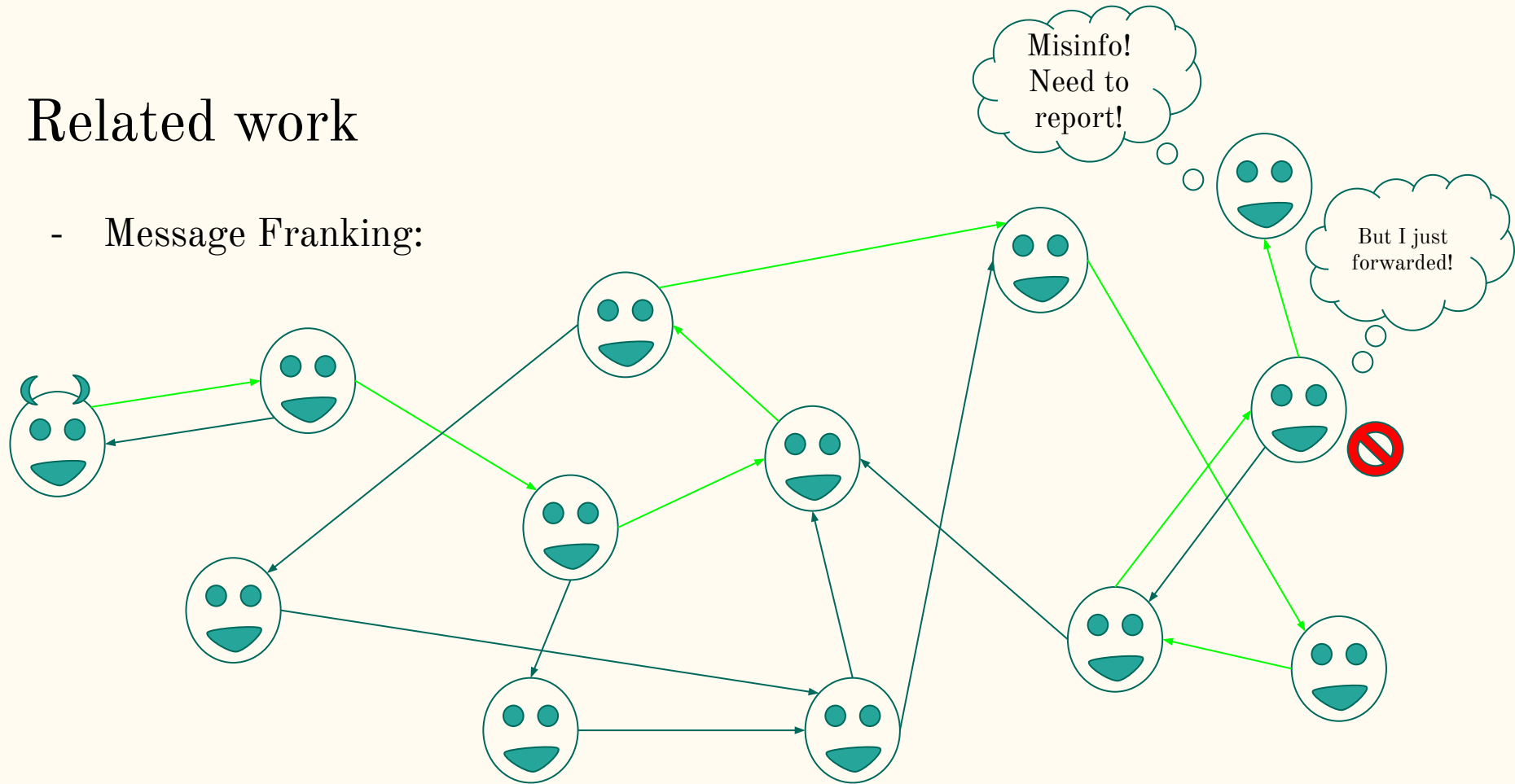
Related work

- Message Franking:



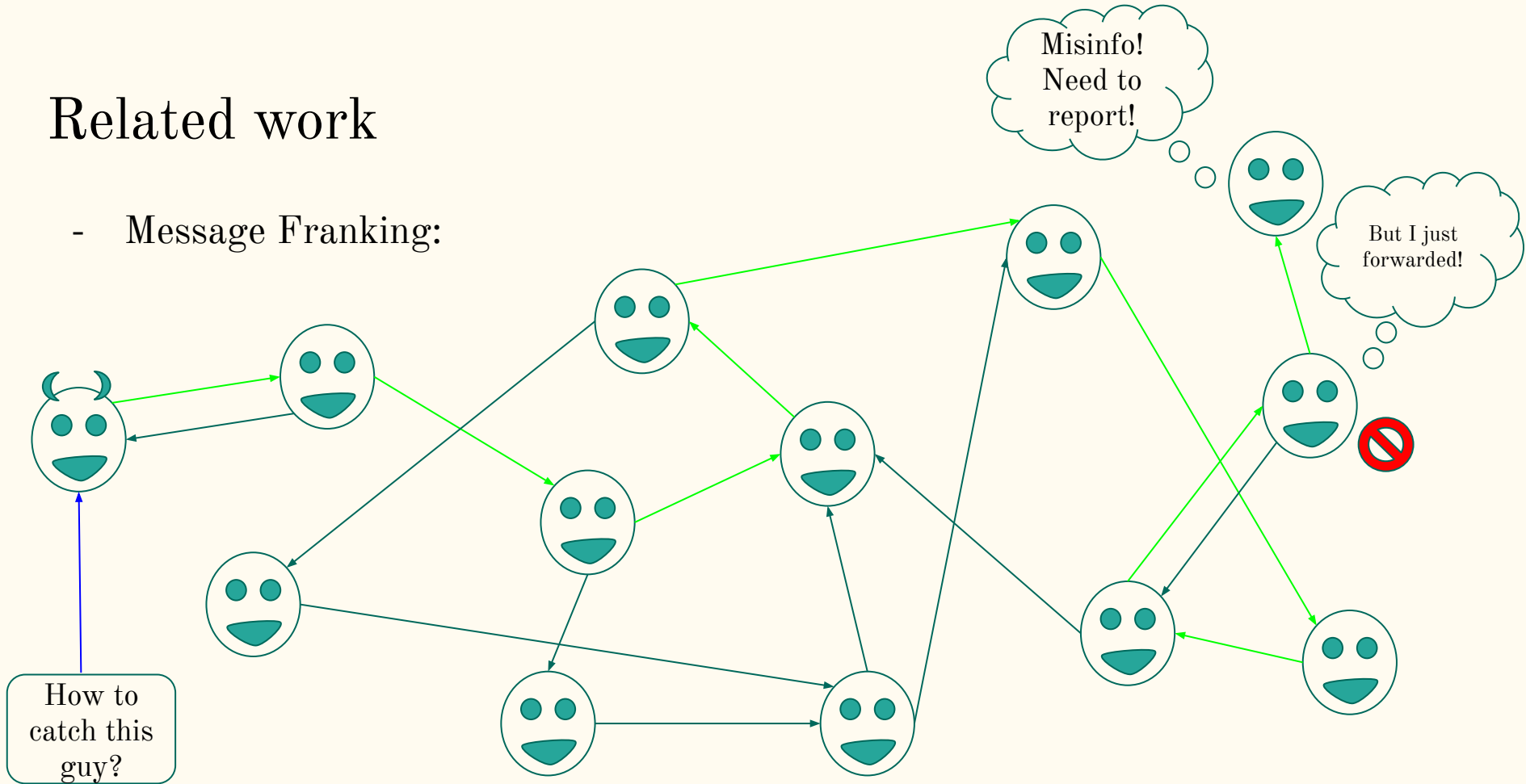
Related work

- Message Franking:



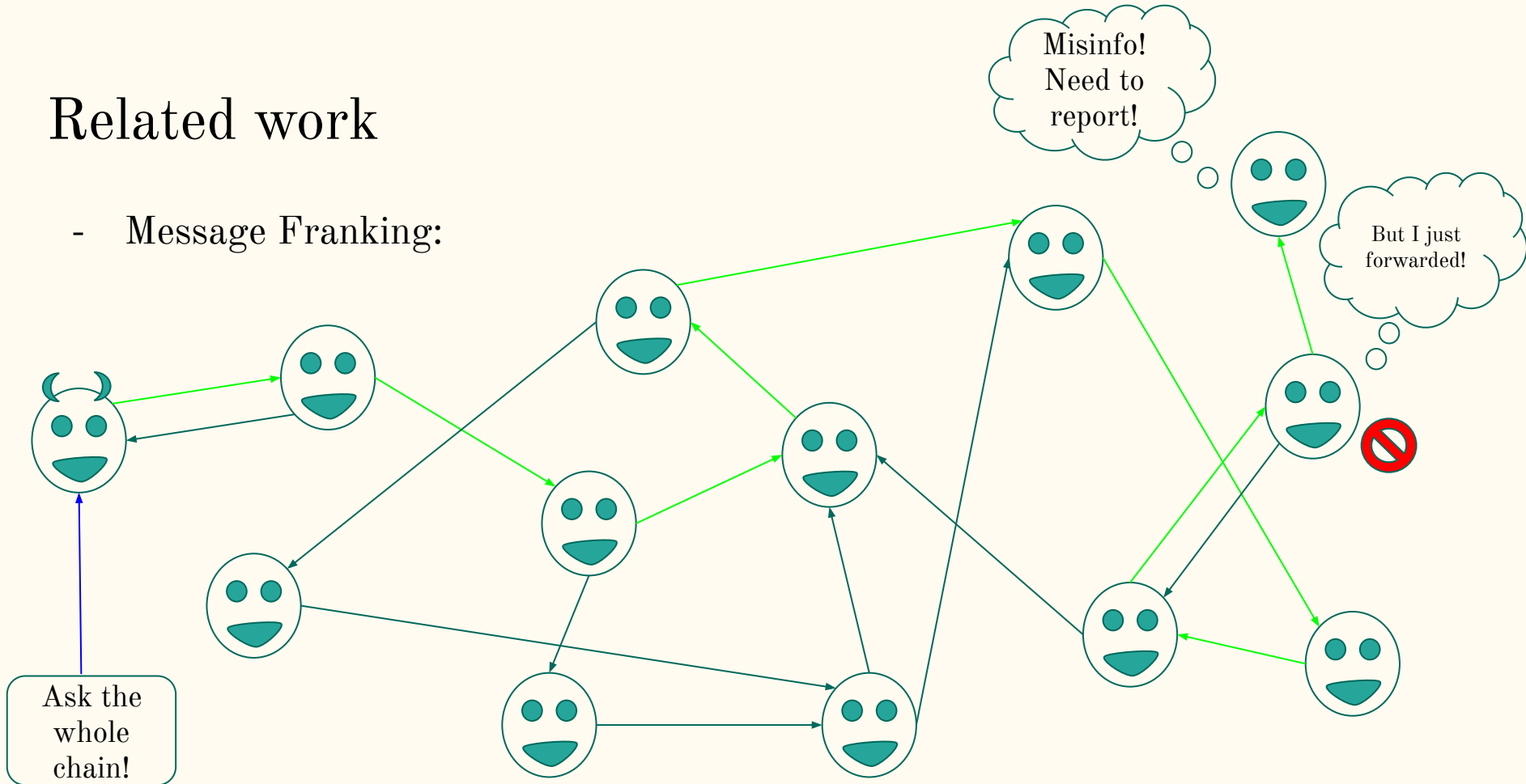
Related work

- Message Franking:



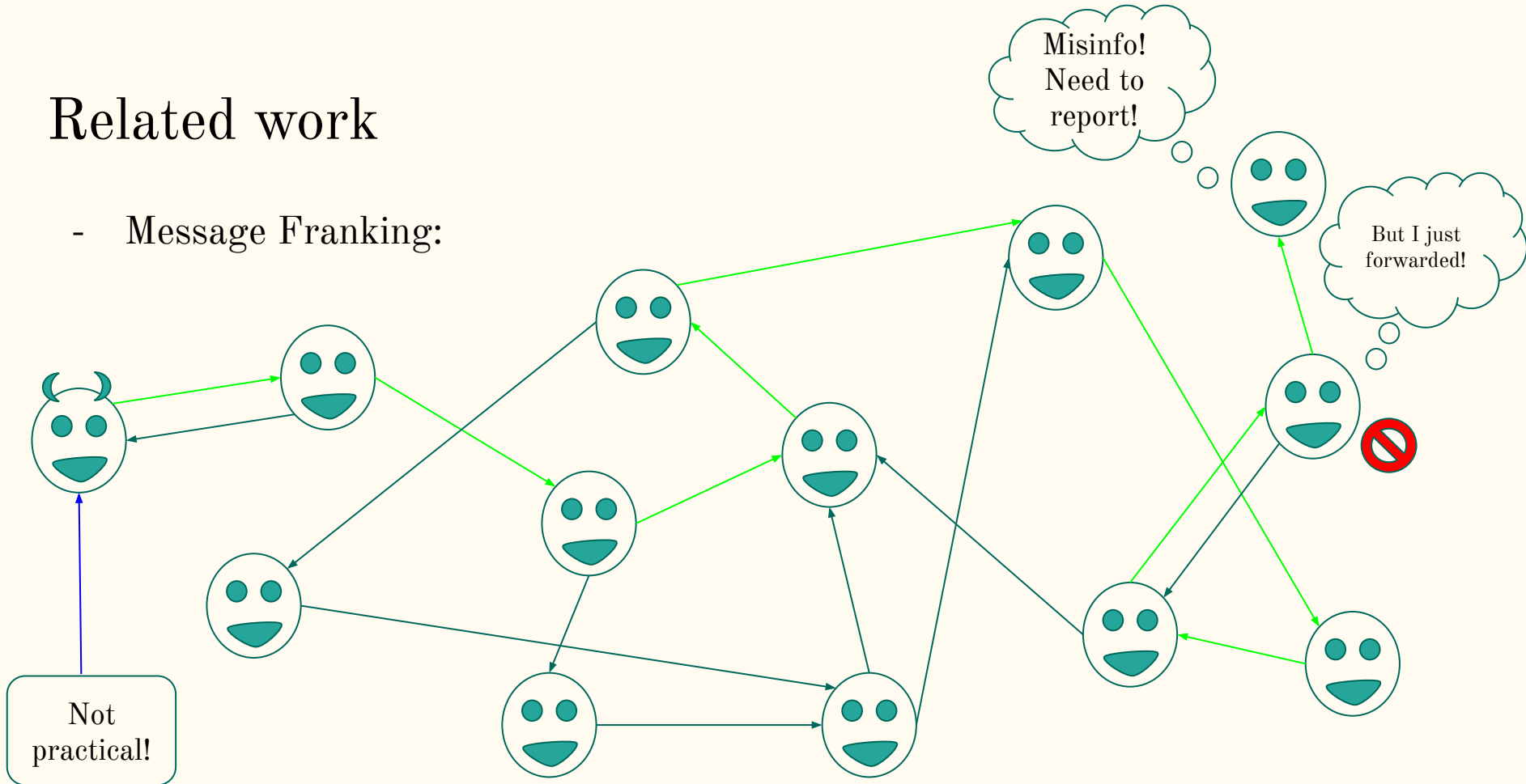
Related work

- Message Franking:



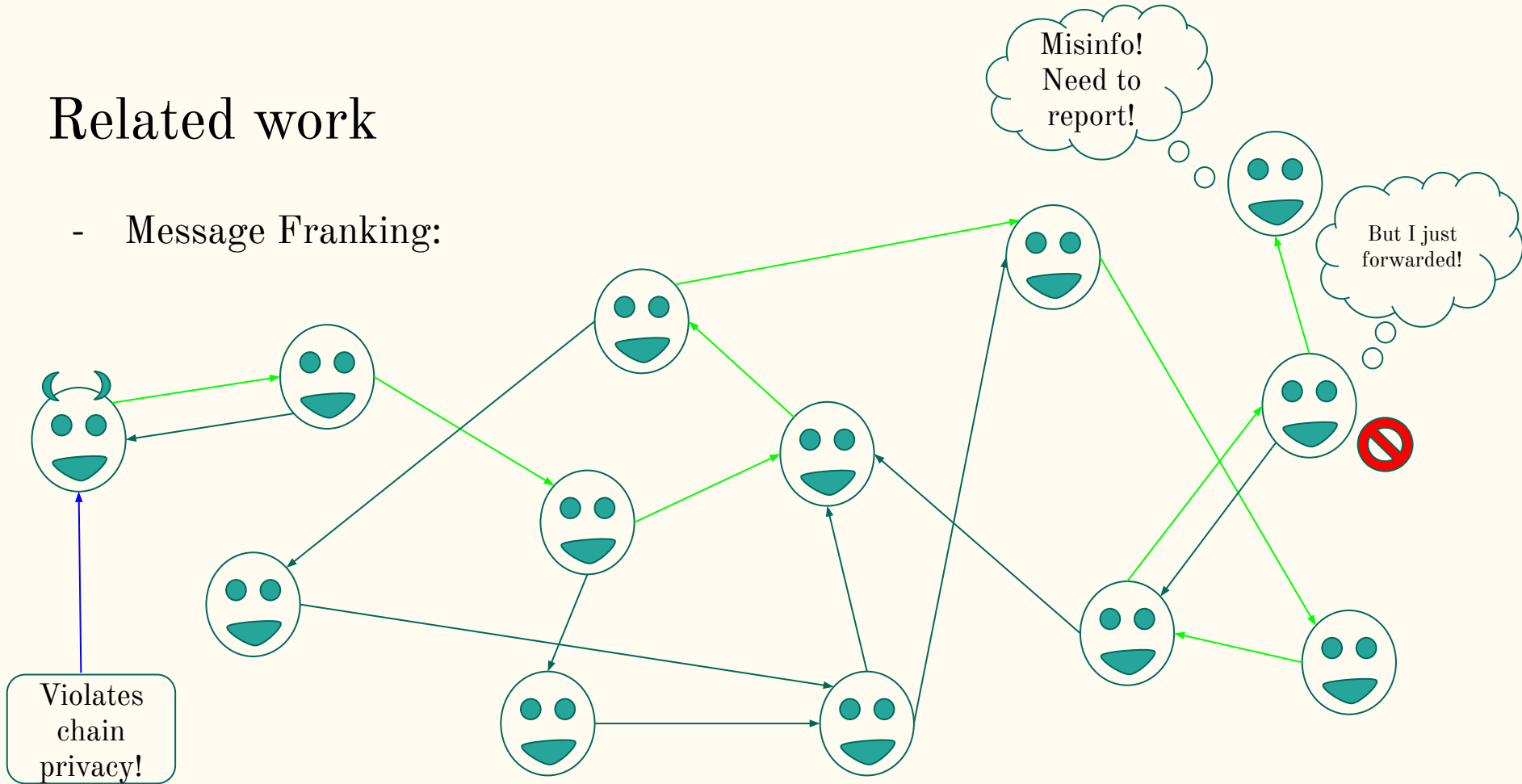
Related work

- Message Franking:



Related work

- Message Franking:



Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders
- Message Traceback [TMR19] [PEB21] [IAV22][LRT⁺22] [KTW22]:

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders
- Message Traceback [TMR19] [PEB21] [IAV22][LRT⁺22] [KTW22] :
 - Can trace entire forwarding tree of reported message

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders
- Message Traceback [TMR19] [PEB21] [IAV22][LRT⁺22] [KTW22] :
 - Can trace entire forwarding tree of reported message
 - Needs additional operations from service provider before message is sent

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders
- Message Traceback [TMR19] [PEB21] [IAV22][LRT⁺22] [KTW22] :
 - Can trace entire forwarding tree of reported message
 - Needs additional operations from service provider before message is sent
 - Law enforcement workload is linear in number of messages, *not* number of reports

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders
- Message Traceback [TMR19] [PEB21] [IAV22][LRT⁺22] [KTW22] :
 - Can trace entire forwarding tree of reported message
 - Needs additional operations from service provider before message is sent
 - Law enforcement workload is linear in number of messages, *not* number of reports
 - User needs to do extra work at the time of sending message

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders
- Message Traceback [TMR19] [PEB21] [IAV22][LRT⁺22] [KTW22] :
 - Can trace entire forwarding tree of reported message
 - Needs additional operations from service provider before message is sent
 - Law enforcement workload is linear in number of messages, *not* number of reports
 - User needs to do extra work at the time of sending message

Can we do better than this?

Related work

- Message Franking [GLR17] [DGR⁺18] [TGL⁺19]:
 - Only traces *immediate* sender, not first originator
 - Needs help of whole chain to trace beyond single party
 - Impractical for large forwarding trees / offline users
 - Violates privacy of forwarders
- Message Traceback [TMR19] [PEB21] [IAV22][LRT⁺22] [KTW22] :
 - Can trace entire forwarding tree of reported message
 - Needs additional operations from service provider before message is sent
 - Law enforcement workload is linear in number of messages, *not* number of reports
 - User needs to do extra work at the time of sending message

Can we do better than this? 

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

We don't care
about the chain,
only the source

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

We operate in
P2P (not star)
network!

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

But with some preprocessing!

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

So *no* server
operations at
runtime!

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

But *some* before
runtime...

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

Tracing data
generation *not* at
runtime

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

But *prior* to it!

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

We fulfill
practically *all*
aspects of E2EE

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

Will later
explain why this
is partial...

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

And why we
think that's
okay!

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

We *don't* need
messaging server
to change its MO

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

LE workload
linear in number
of reports

Related Work

TABLE I: Comparison of Tracing in End-to-End Messaging Systems.

Protocols	Trace Type	Runtime Network [†]	Load Balancing [‡]	Confidentiality	Anonymity	Deniability	Forward Security	Backward Security	Unforgeability	Accountability	Tree Unlinkability ^ℒ	No. of Servers	Storage Required [§]	Server Immutability	LE Workload [¶]
Kamakoti [36]	source/path	P2P	×	○	○	●	●	●	○	○	○	1	small	●	$O(r)$
AMF [44]	source	Star	runtime	●	●	●	○	○	●	●	○	1	small	○	$O(n)$
Traceback [45]	source/path	Star	runtime	●	○	●	○	○	●	●	○	1	large	○	$O(n)$
Peale et al. [42]	source	Star	runtime	●	○	●	●	○	●	●	●	1	large	○	$O(n)$
FACTS [41]	source	Star	runtime	●	◐	●	●	○	●	●	○	2	large	○	$O(n)$
Kenney et al. [40]	source/path	Star	runtime	●	●	●	●	●	●	●	○	1 or 2	large	○	$O(n)$
Hecate [39]	source	Star	preprocessing	●	●	●	●	●	●	●	○	2	small	○	$O(n)$
This Work	source	P2P-wP	preprocessing	●	●	●	●	●	●	●	◐	1 or 2	small	●	$O(r)$

Not number of
messages sent

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- **Private Originator Tracing - Syntax**
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

Run locally by SP to
generate their key
pair

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

Run locally by user
to generate their key
pair

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

Repeat to create as
many key pairs as
needed

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

$$(\text{rc}, \mathbf{U}) \leftarrow \text{UserReg}(S, \text{pk}_S, \text{sk}_S, \mathbf{U}, U_{\text{new}}, \text{pk}_{U_{\text{new}}}, \text{sk}_{U_{\text{new}}})$$

Interactively run by SP and new user,
returns reg. confirm. and updated
membership set $\mathbf{U} \leftarrow \mathbf{U} \cup U_{\text{new}}$

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

$$(\text{rc}, \mathbf{U}) \leftarrow \text{UserReg}(S, \text{pk}_S, \text{sk}_S, U, U_{\text{new}}, \text{pk}_{U_{\text{new}}}, \text{sk}_{U_{\text{new}}})$$

$$\text{ad} \leftarrow \text{Auth}(U_{\text{reg}}, \text{pk}_{U_{\text{reg}}}, \text{rc}, S, \text{sk}_S)$$

Interactively run by SP and reg. user $U_{\text{reg}} \in U$ to auth. and record $(\text{pk}_{U_{\text{reg}}}, U_{\text{reg}})$, returns authoring data $\text{ad} := (\text{pk}_{U_{\text{reg}}}, \text{ath}_{\text{pk}_{U_{\text{reg}}}})$ where $\text{ath}_{\text{pk}_{U_{\text{reg}}}}$ authenticates $\text{pk}_{U_{\text{reg}}}$ using sk_S

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

$$(\text{rc}, U) \leftarrow \text{UserReg}(S, \text{pk}_S, \text{sk}_S, U, U_{\text{new}}, \text{pk}_{U_{\text{new}}}, \text{sk}_{U_{\text{new}}})$$

$$\text{ad} \leftarrow \text{Auth}(U_{\text{reg}}, \text{pk}_{U_{\text{reg}}}, \text{rc}, S, \text{sk}_S)$$

$$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$$

Verify if authoring data is
valid under pk_S

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

$$(\text{rc}, U) \leftarrow \text{UserReg}(S, \text{pk}_S, \text{sk}_S, U, U_{\text{new}}, \text{pk}_{U_{\text{new}}}, \text{sk}_{U_{\text{new}}})$$

$$\text{ad} \leftarrow \text{Auth}(U_{\text{reg}}, \text{pk}_{U_{\text{reg}}}, \text{rc}, S, \text{sk}_S)$$

$$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$$

$$M \leftarrow \text{NewMsg}(U_{\text{send}}, m, \text{sk}_{U_{\text{send}}}, \text{ad} := (\text{pk}_{U_{\text{send}}}, \text{ath}_{\text{pk}_{U_{\text{send}}}}))$$

Run by $U_{\text{send}} \in U$ to create message tuple $M := (m, \text{md}, \text{ad})$

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

$$(\text{rc}, U) \leftarrow \text{UserReg}(S, \text{pk}_S, \text{sk}_S, U, U_{\text{new}}, \text{pk}_{U_{\text{new}}}, \text{sk}_{U_{\text{new}}})$$

$$\text{ad} \leftarrow \text{Auth}(U_{\text{reg}}, \text{pk}_{U_{\text{reg}}}, \text{rc}, S, \text{sk}_S)$$

$$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$$

$$M \leftarrow \text{NewMsg}(U_{\text{send}}, m, \text{sk}_{U_{\text{send}}}, \text{ad} := (\text{pk}_{U_{\text{send}}}, \text{ath}_{\text{pk}_{U_{\text{send}}}}))$$

$$1/0 \leftarrow \text{MVf}(\text{pk}_S, M)$$

Verifies if M is a valid message tuple under pk_S , ad and md

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

$$(\text{rc}, U) \leftarrow \text{UserReg}(S, \text{pk}_S, \text{sk}_S, U, U_{\text{new}}, \text{pk}_{U_{\text{new}}}, \text{sk}_{U_{\text{new}}})$$

$$\text{ad} \leftarrow \text{Auth}(U_{\text{reg}}, \text{pk}_{U_{\text{reg}}}, \text{rc}, S, \text{sk}_S)$$

$$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$$

$$M \leftarrow \text{NewMsg}(U_{\text{send}}, m, \text{sk}_{U_{\text{send}}}, \text{ad} := (\text{pk}_{U_{\text{send}}}, \text{ath}_{\text{pk}_{U_{\text{send}}}}))$$

$$1/0 \leftarrow \text{MVf}(\text{pk}_S, M)$$

$$M \leftarrow \text{RcvMsg}(U_{\text{send}}, M := (m, \text{md}, \text{ad}), U_{\text{rev}})$$

Run by $U_{\text{rev}} \in U$ to receive the tuple M created by $U_{\text{send}} \in U$

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$(\text{pk}_S, \text{sk}_S) \leftarrow \text{SKeyGen}(1^\lambda, S)$$

$$(\text{pk}_U, \text{sk}_U) \leftarrow \text{UKeyGen}(1^\lambda, U)$$

$$(\text{rc}, U) \leftarrow \text{UserReg}(S, \text{pk}_S, \text{sk}_S, U, U_{\text{new}}, \text{pk}_{U_{\text{new}}}, \text{sk}_{U_{\text{new}}})$$

$$\text{ad} \leftarrow \text{Auth}(U_{\text{reg}}, \text{pk}_{U_{\text{reg}}}, \text{rc}, S, \text{sk}_S)$$

$$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$$

$$M \leftarrow \text{NewMsg}(U_{\text{send}}, m, \text{sk}_{U_{\text{send}}}, \text{ad} := (\text{pk}_{U_{\text{send}}}, \text{ath}_{\text{pk}_{U_{\text{send}}}}))$$

$$1/0 \leftarrow \text{MVf}(\text{pk}_S, M)$$

$$M \leftarrow \text{FwdMsg}(U_{\text{rev}}, M := (m, \text{md}, \text{ad}), U_{\text{fwd}})$$

Run by $U_{\text{rev}} \in U$ to forward the received tuple M to $U_{\text{fwd}} \in U$

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$$\text{rep} := (\text{rm}, \text{rd}) \leftarrow \text{Report} (U_{\text{fwd}}, M := (m, \text{md}, \text{ad}), L)$$

Run by $U_{\text{fwd}} \in \mathbf{U}$ to report a message tuple M by sending $\text{rep} := (\text{rm}, \text{rd})$ to L where $\text{rm} := (m, \text{md})$ and $\text{rd} := \text{ad}$ (of corr. M)

$$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$$
$$M \leftarrow \text{NewMsg} (U_{\text{send}}, m, \text{sk}_{U_{\text{send}}}, \text{ad} := (\text{pk}_{U_{\text{send}}}, \text{ath}_{\text{pk}_{U_{\text{send}}}}))$$
$$1/0 \leftarrow \text{MVf}(\text{pk}_S, M)$$
$$M \leftarrow \text{FwdMsg} (U_{\text{rev}}, M := (m, \text{md}, \text{ad}), U_{\text{fwd}})$$

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$\text{rep} := (\text{rm}, \text{rd}) \leftarrow \text{Report} (\text{U}_{\text{fwd}}, \text{M} := (\text{m}, \text{md}, \text{ad}), \text{L})$

$1/0 \leftarrow \text{repVf}(\text{pk}_S, \text{rep})$

Verifies if rep is a valid
report under pk_S

$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$

$\text{M} \leftarrow \text{NewMsg} (\text{U}_{\text{send}}, \text{m},$
 $\text{sk}_{\text{U}_{\text{send}}}, \text{ad} := (\text{pk}_{\text{U}_{\text{send}}}, \text{ath}_{\text{pk}_{\text{U}_{\text{send}}}}))$

$1/0 \leftarrow \text{MVf}(\text{pk}_S, \text{M})$

$\text{M} \leftarrow \text{FwdMsg} (\text{U}_{\text{rev}}, \text{M} := (\text{m},$
 $\text{md}, \text{ad}), \text{U}_{\text{fwd}})$

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$\text{rep} := (\text{rm}, \text{rd}) \leftarrow \text{Report} (\text{U}_{\text{fwd}}, \text{M} := (\text{m}, \text{md}, \text{ad}), \text{L})$

$1/0 \leftarrow \text{repVf}(\text{pk}_S, \text{rep})$

$1/0 \leftarrow \text{rdVf}(\text{pk}_S, \text{rd})$

Verifies if rd has valid reporting data under pk_S

$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$

$\text{M} \leftarrow \text{NewMsg} (\text{U}_{\text{send}}, \text{m}, \text{sk}_{\text{U}_{\text{send}}}, \text{ad} := (\text{pk}_{\text{U}_{\text{send}}}, \text{ath}_{\text{pk}_{\text{U}_{\text{send}}}}))$

$1/0 \leftarrow \text{MVf}(\text{pk}_S, \text{M})$

$\text{M} \leftarrow \text{FwdMsg} (\text{U}_{\text{rev}}, \text{M} := (\text{m}, \text{md}, \text{ad}), \text{U}_{\text{fwd}})$

Private Originator Tracing - Syntax

Tuple of PPT algorithms:

$\text{rep} := (\text{rm}, \text{rd}) \leftarrow \text{Report} (U_{\text{fwd}}, M := (m, \text{md}, \text{ad}), L)$

$1/0 \leftarrow \text{repVf}(\text{pk}_S, \text{rep})$

$1/0 \leftarrow \text{rdVf}(\text{pk}_S, \text{rd})$

$1/0 \leftarrow \text{adVf}(\text{pk}_S, \text{ad})$

$M \leftarrow \text{NewMsg} (U_{\text{send}}, m, \text{sk}_{U_{\text{send}}}, \text{ad} := (\text{pk}_{U_{\text{send}}}, \text{ath}_{\text{pk}_{U_{\text{send}}}}))$

$1/0 \leftarrow \text{MVf}(\text{pk}_S, M)$

$M \leftarrow \text{FwdMsg} (U_{\text{rev}}, M := (m, \text{md}, \text{ad}), U_{\text{fwd}})$

Interactively run by L and S that takes report data rd and returns originator U_{send} of reported message to L (without revealing m to S)

$U_{\text{send}} \leftarrow \text{Trace}(L, \text{rd}, S)$

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- **ATAVISM - a protocol sketch**
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

ATAVISM - a protocol sketch



ATAVISM - a protocol sketch



(pk_s, sk_s)

Preprocessing phase

ATAVISM - a protocol sketch



(pk_s, sk_s)



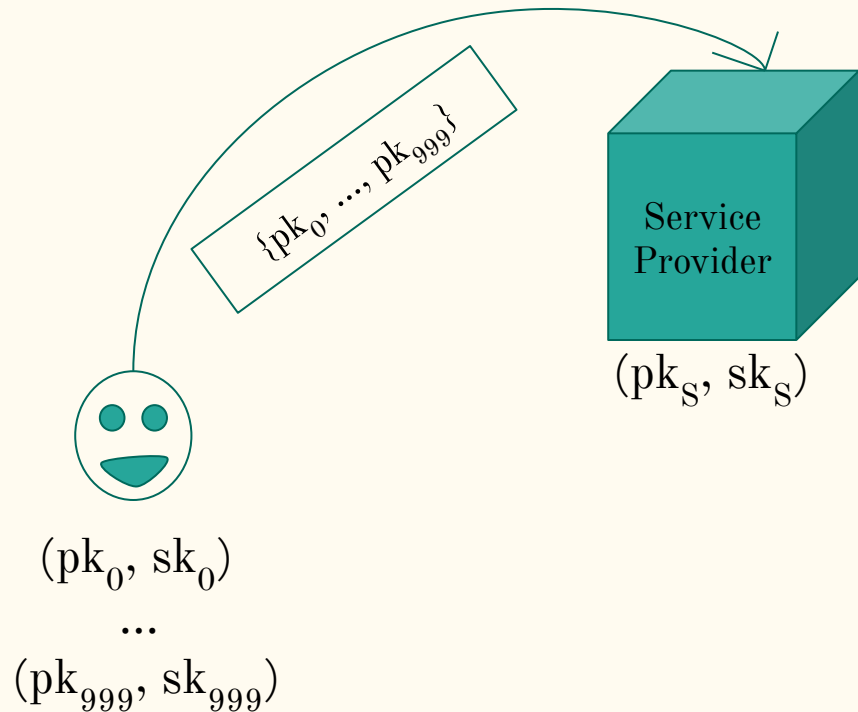
(pk_0, sk_0)

...

(pk_{999}, sk_{999})

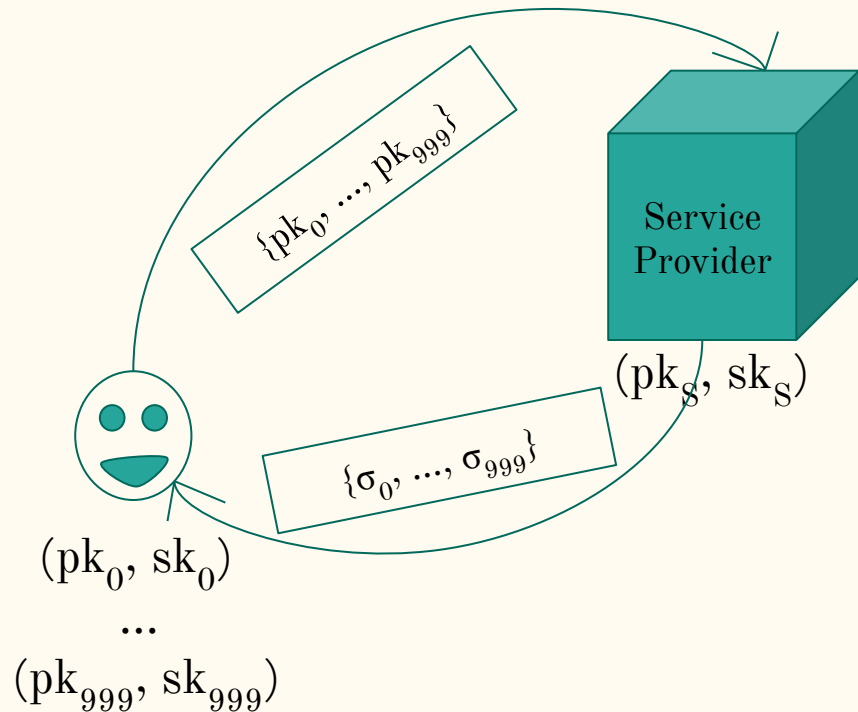
Preprocessing phase

ATAVISM - a protocol sketch



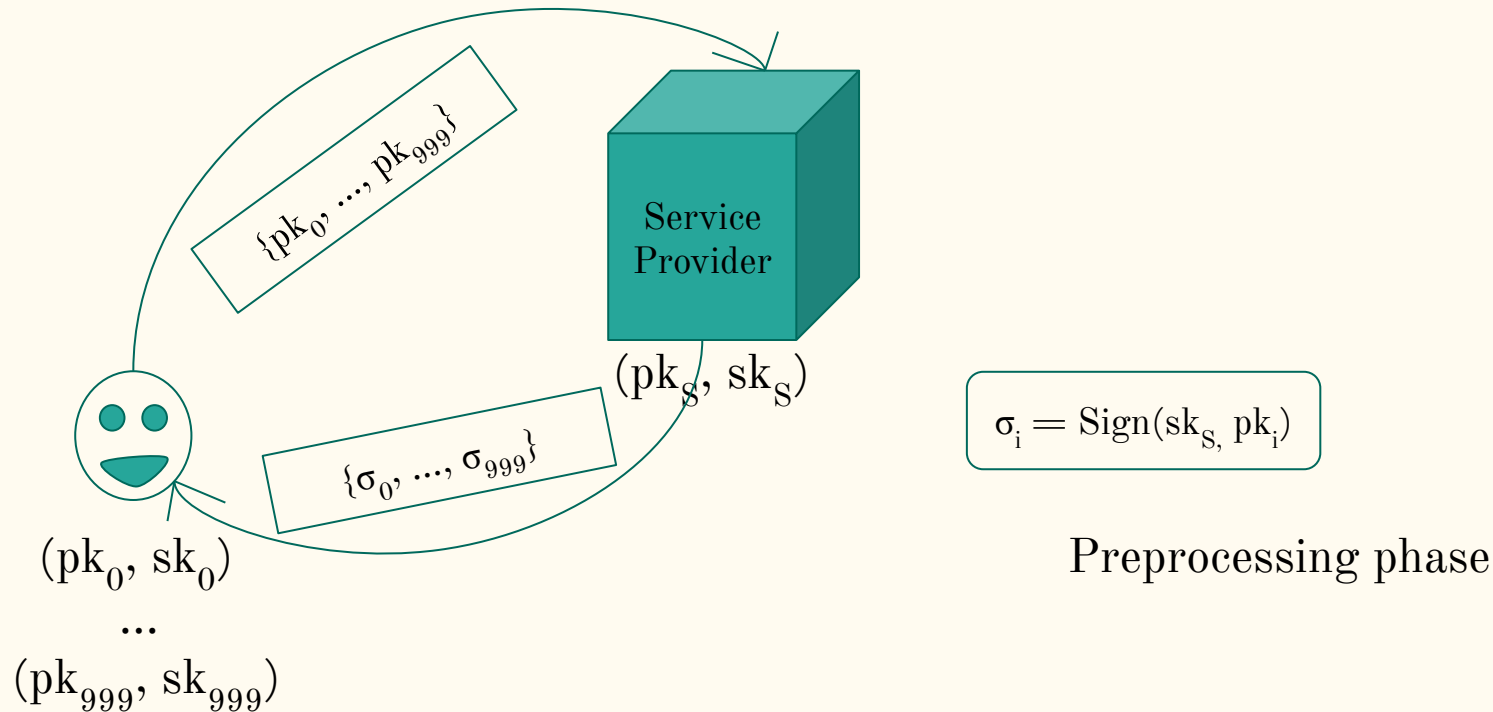
Preprocessing phase

ATAVISM - a protocol sketch

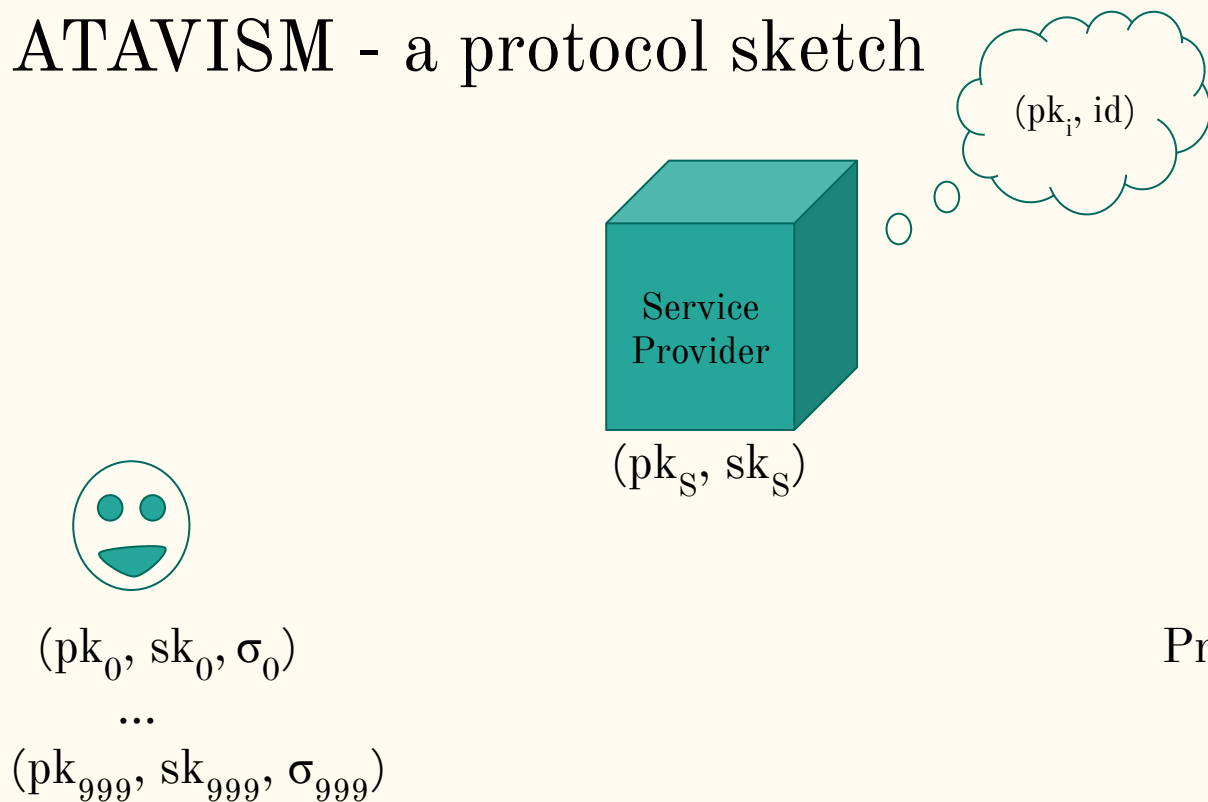


Preprocessing phase

ATAVISM - a protocol sketch

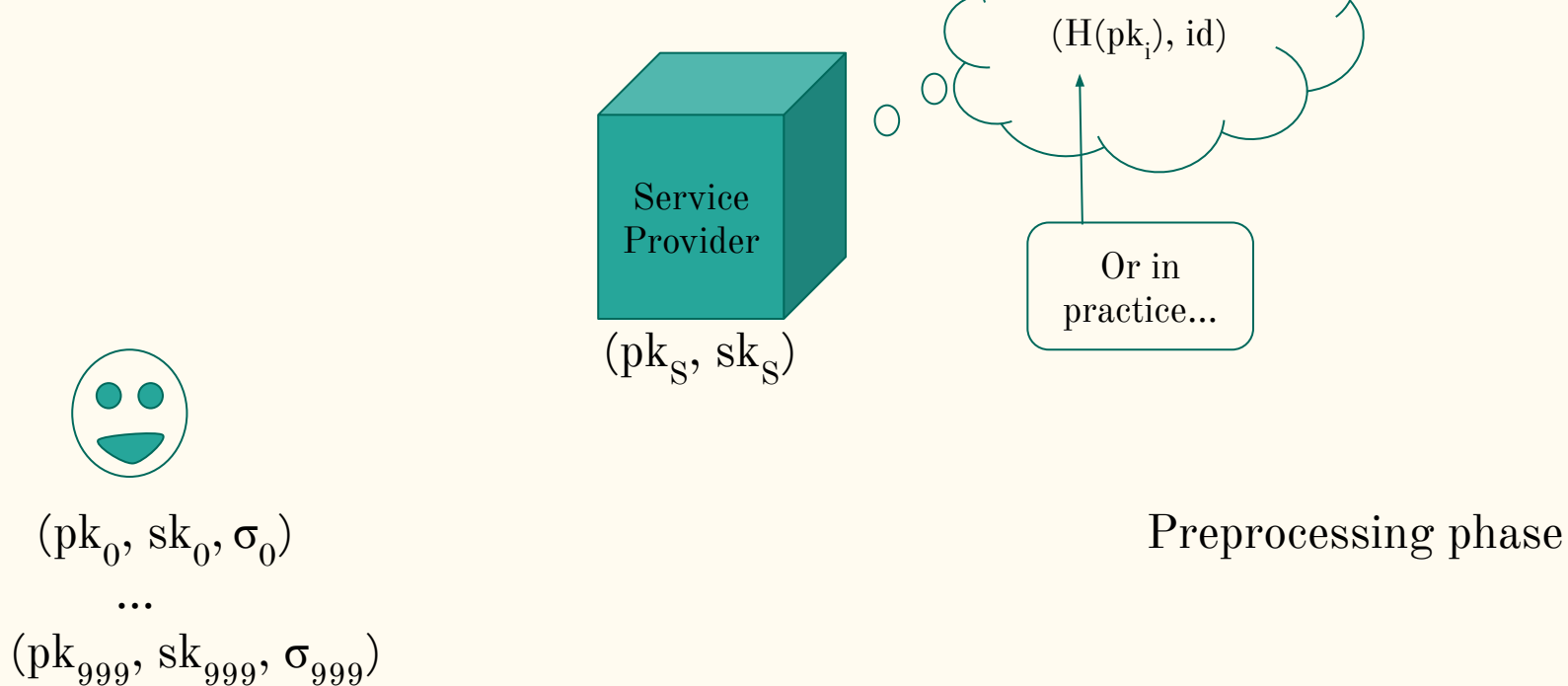


ATAVISM - a protocol sketch

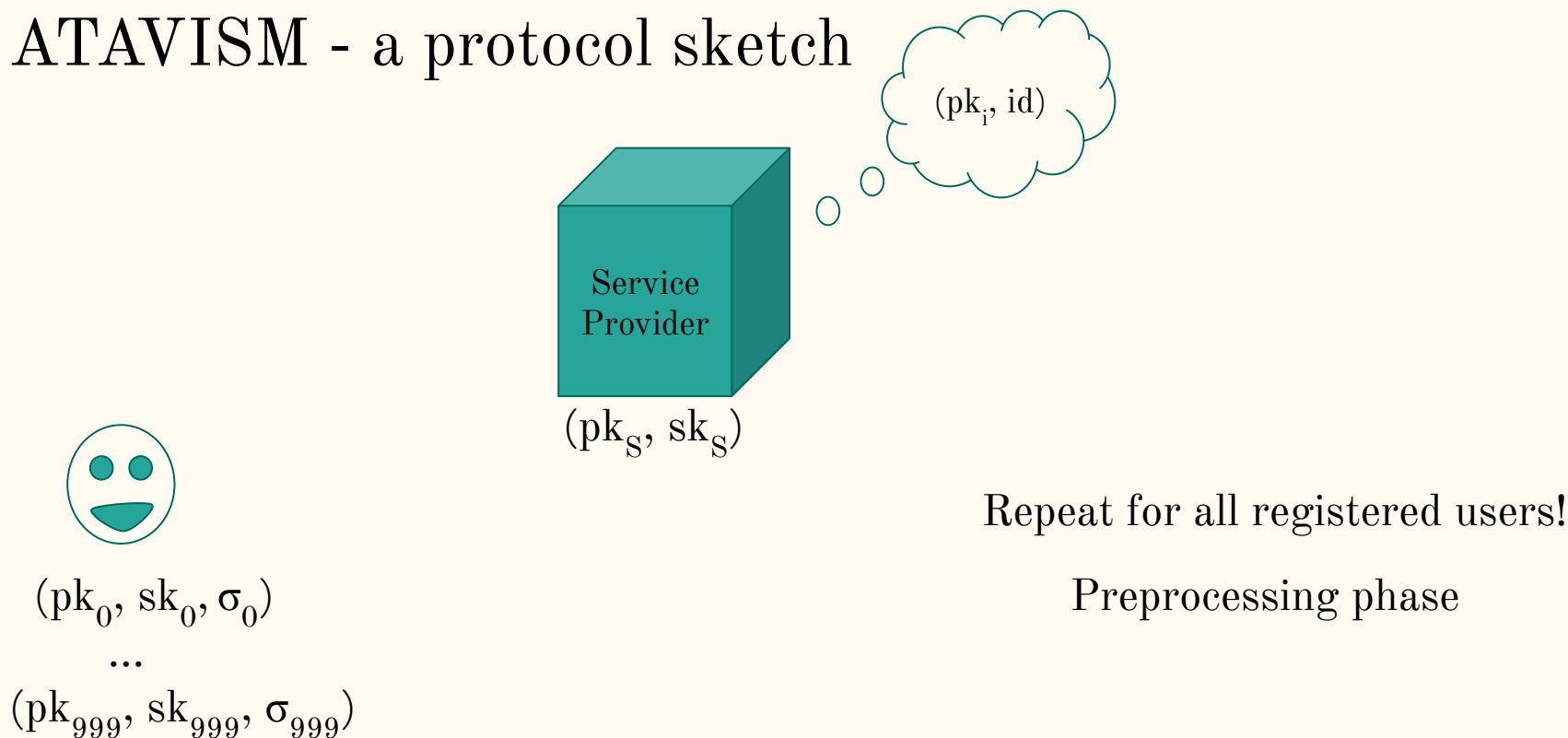


Preprocessing phase

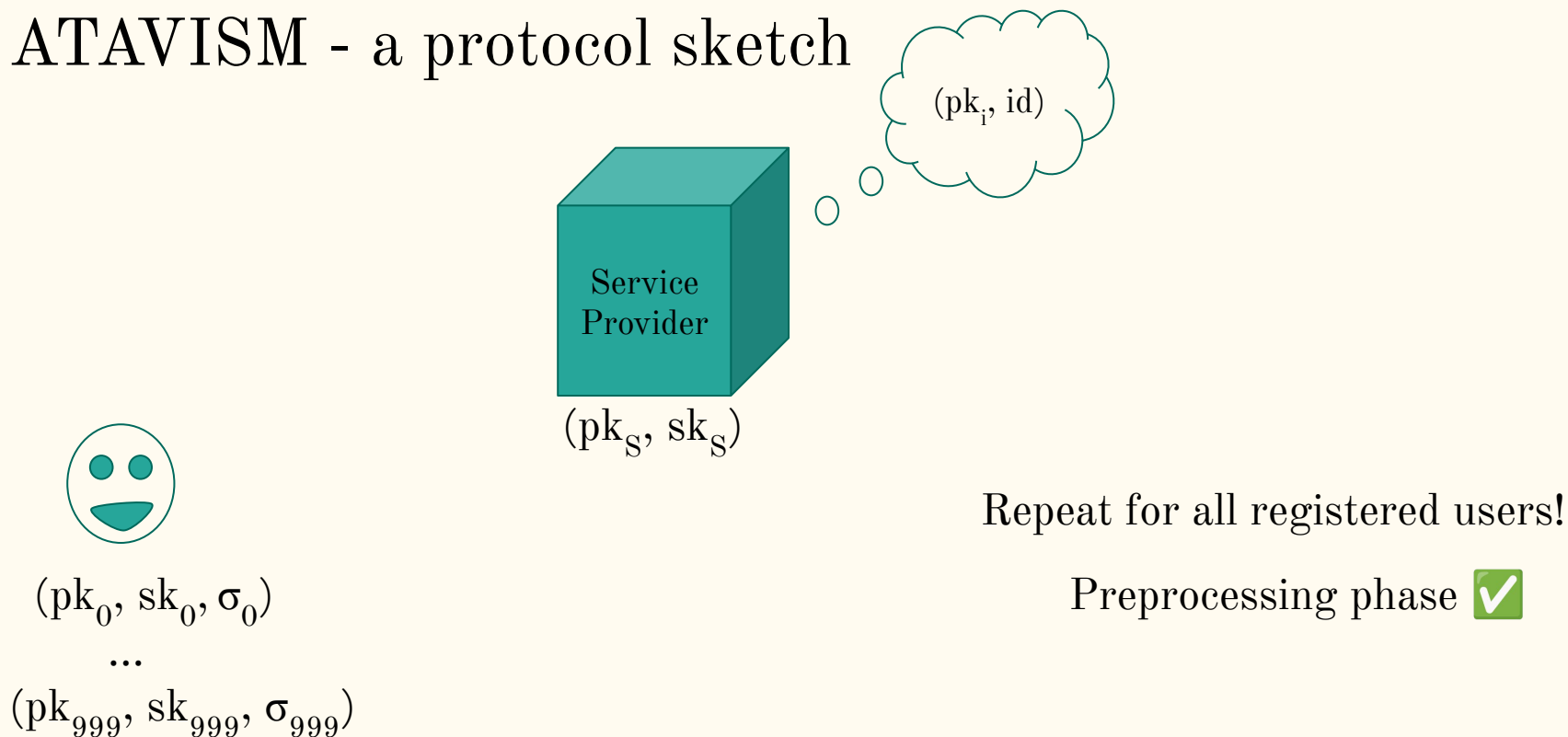
ATAVISM - a protocol sketch



ATAVISM - a protocol sketch

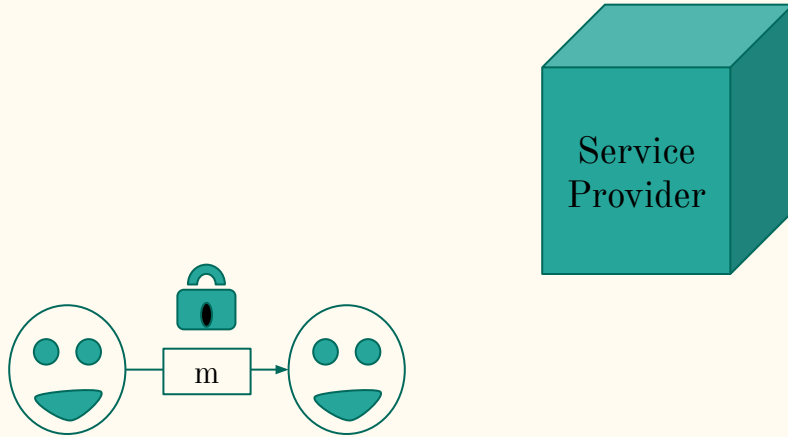


ATAVISM - a protocol sketch



ATAVISM - a protocol sketch

Online phase

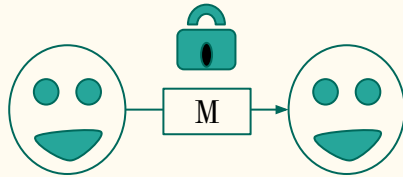


“Hey, do you
want to switch to
using Signal?”

m

ATAVISM - a protocol sketch

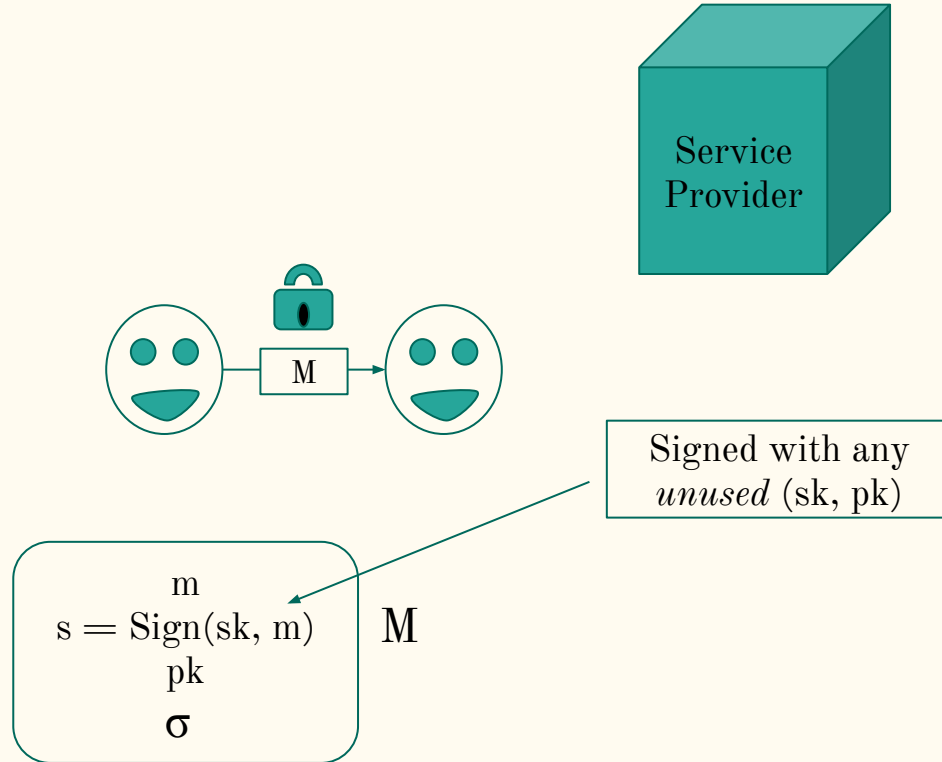
Online phase



$$\begin{array}{c} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma \end{array} \quad M$$

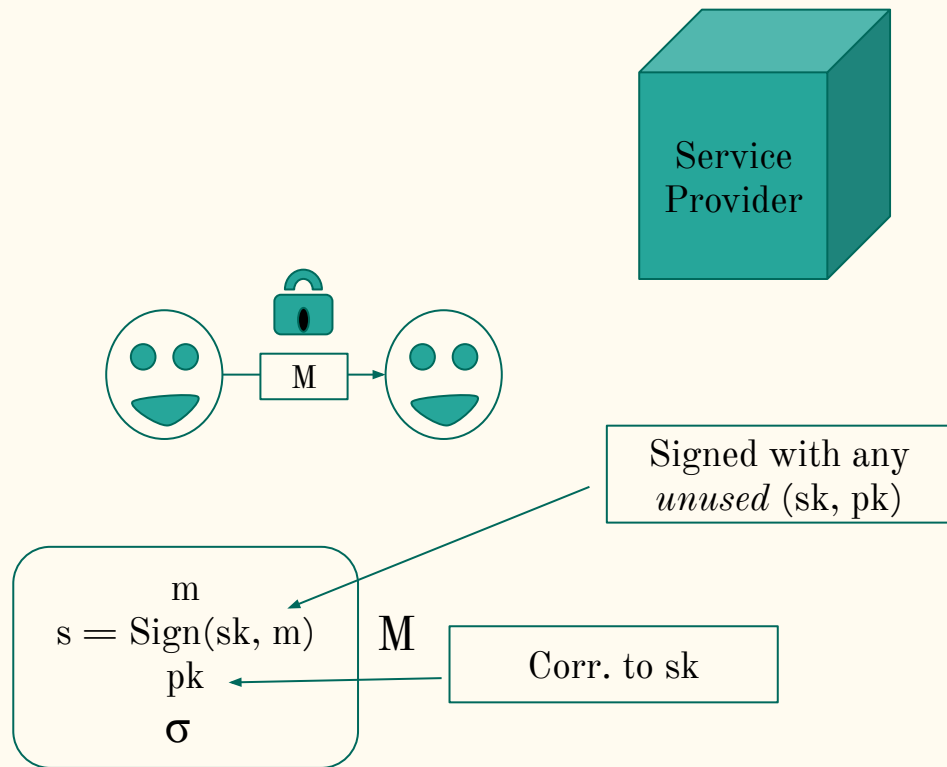
ATAVISM - a protocol sketch

Online phase



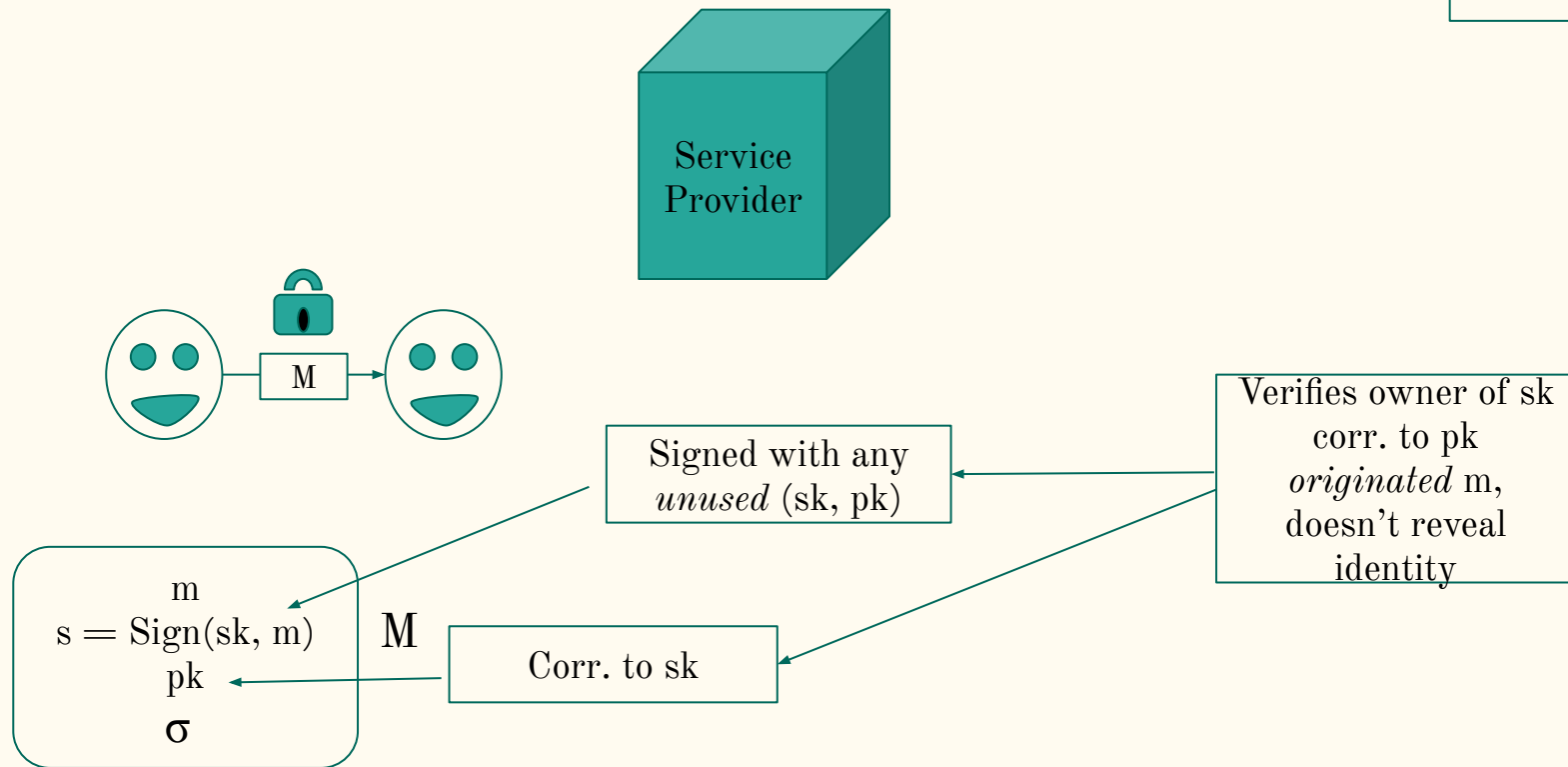
ATAVISM - a protocol sketch

Online phase



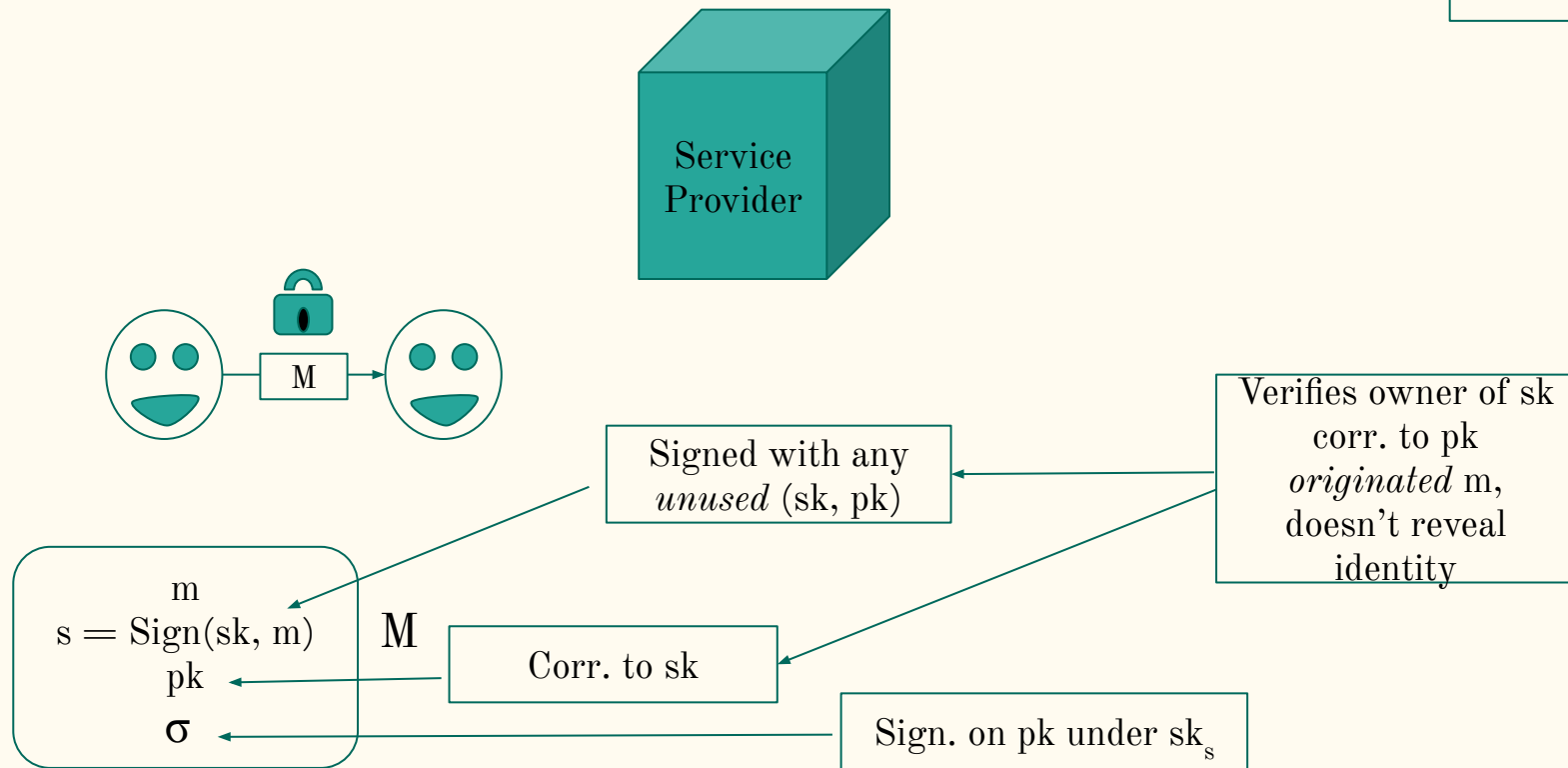
ATAVISM - a protocol sketch

Online phase



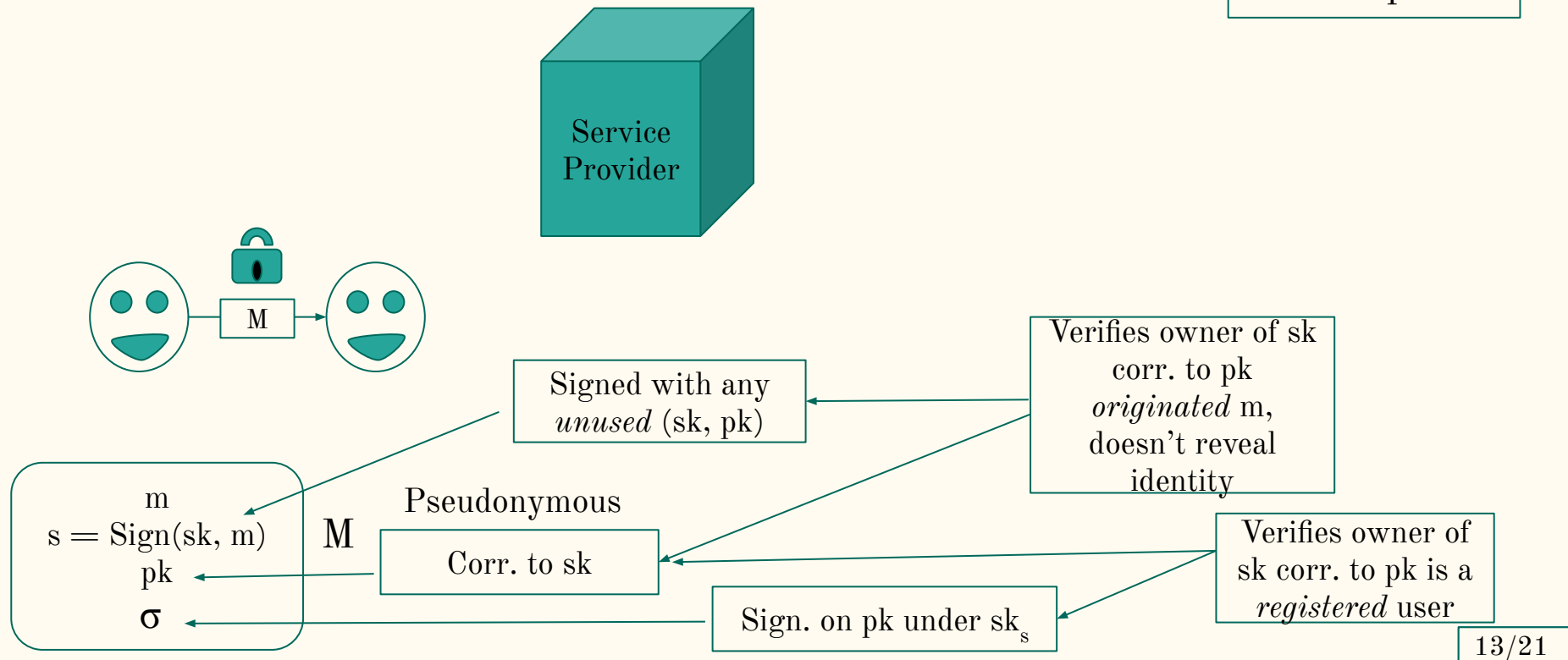
ATAVISM - a protocol sketch

Online phase



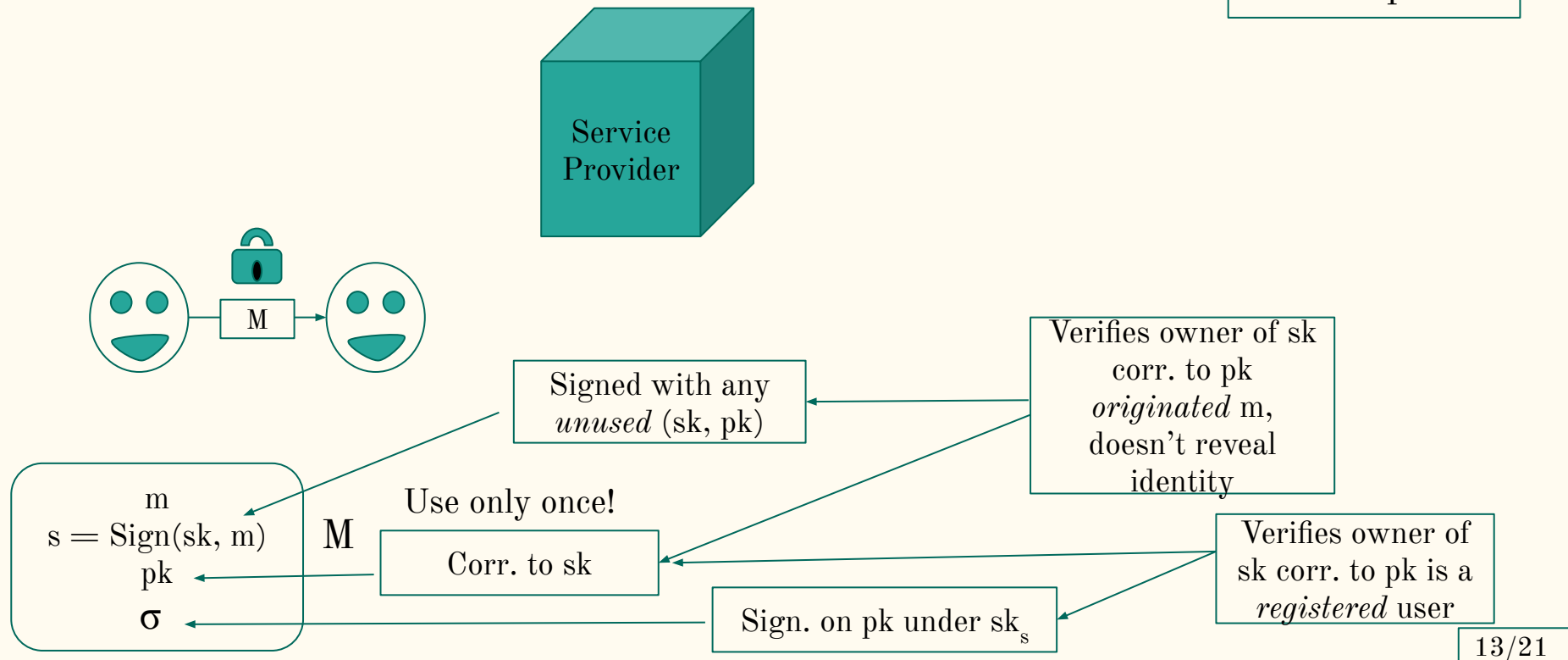
ATAVISM - a protocol sketch

Online phase

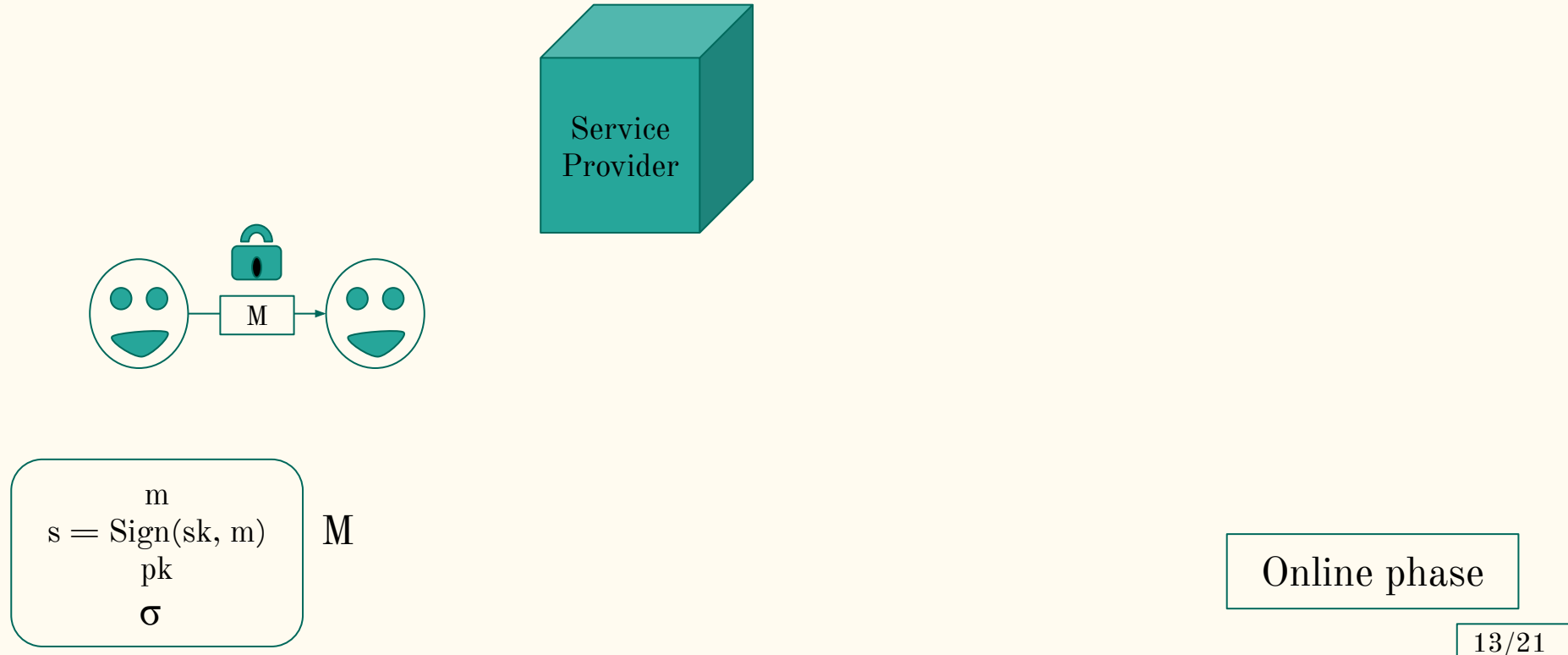


ATAVISM - a protocol sketch

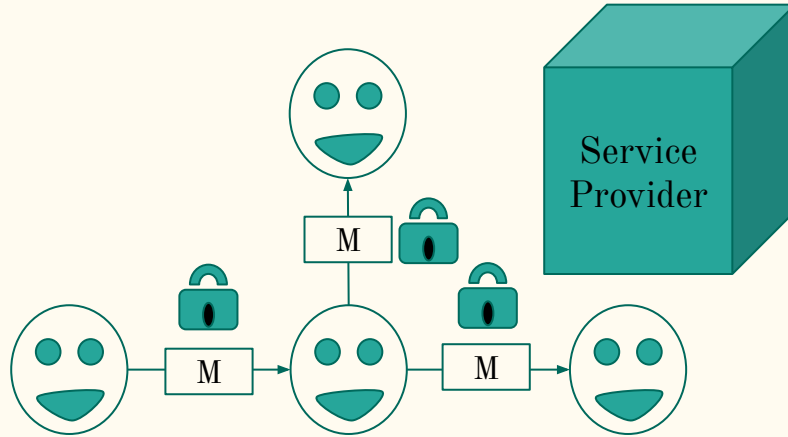
Online phase



ATAVISM - a protocol sketch



ATAVISM - a protocol sketch

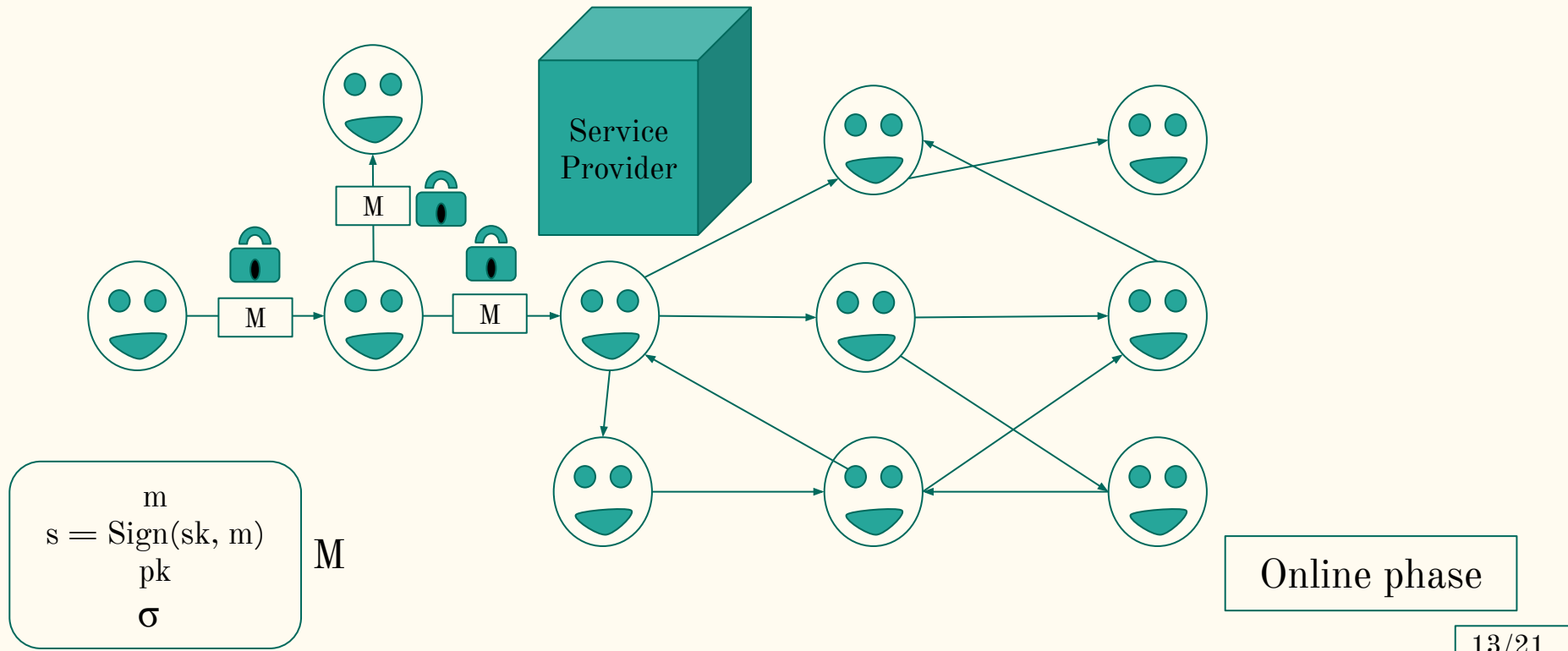


$$\begin{matrix} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma \end{matrix}$$

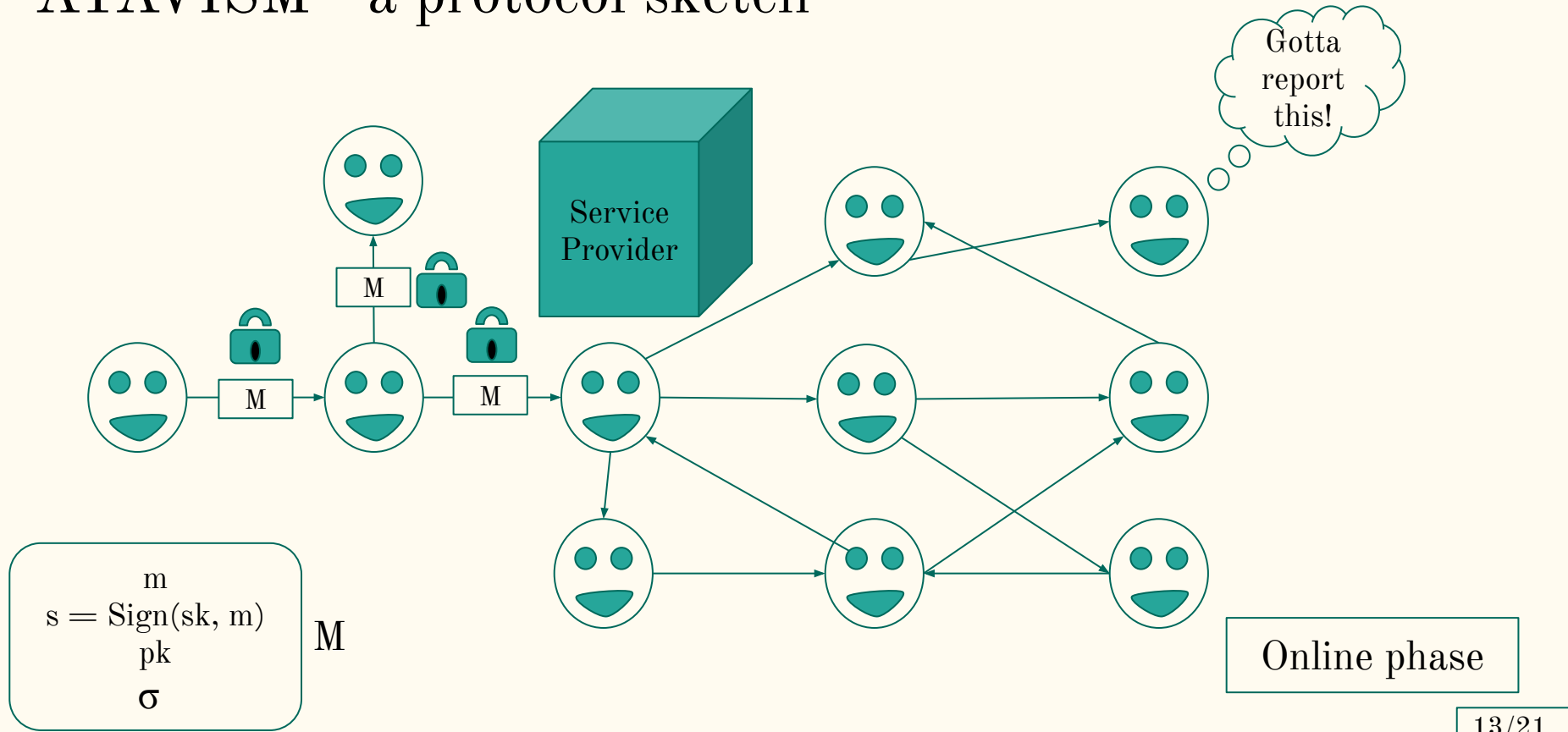
M

Online phase

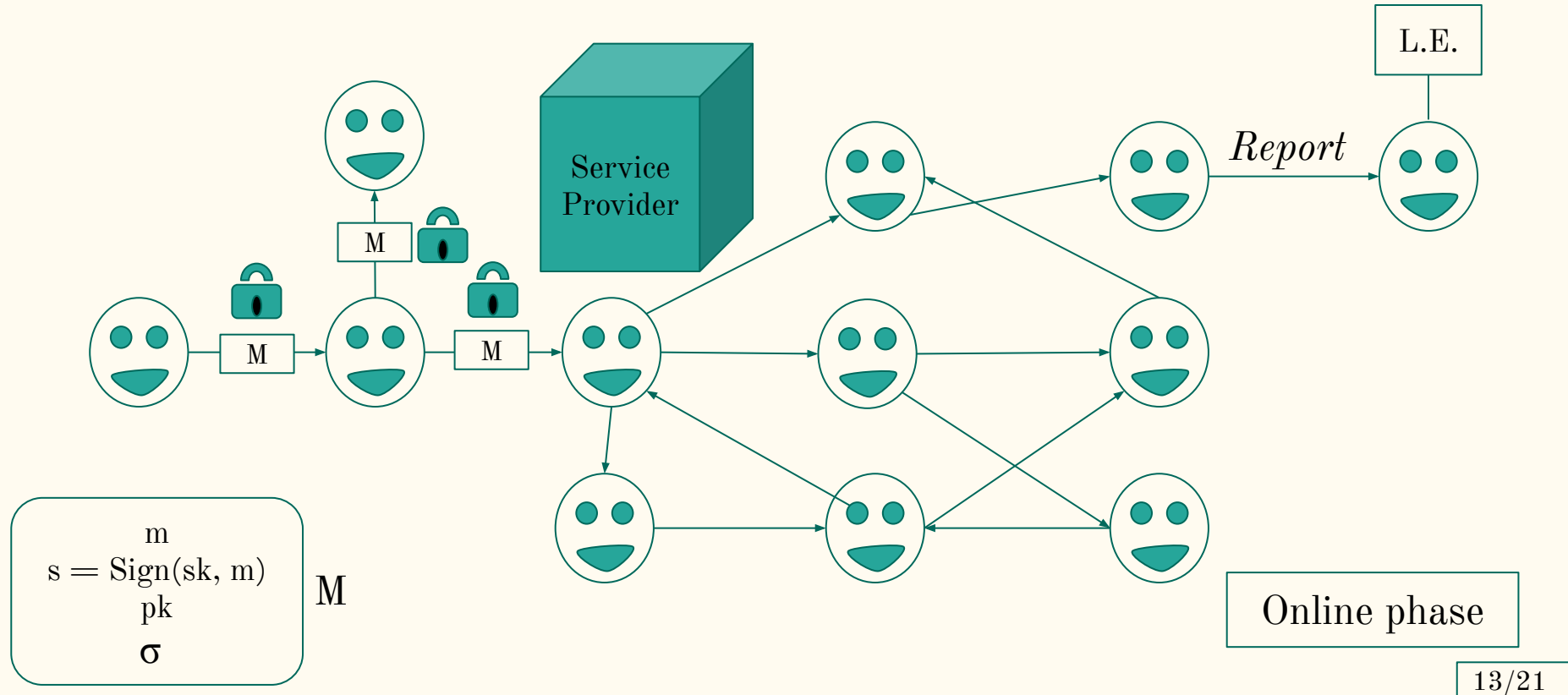
ATAVISM - a protocol sketch



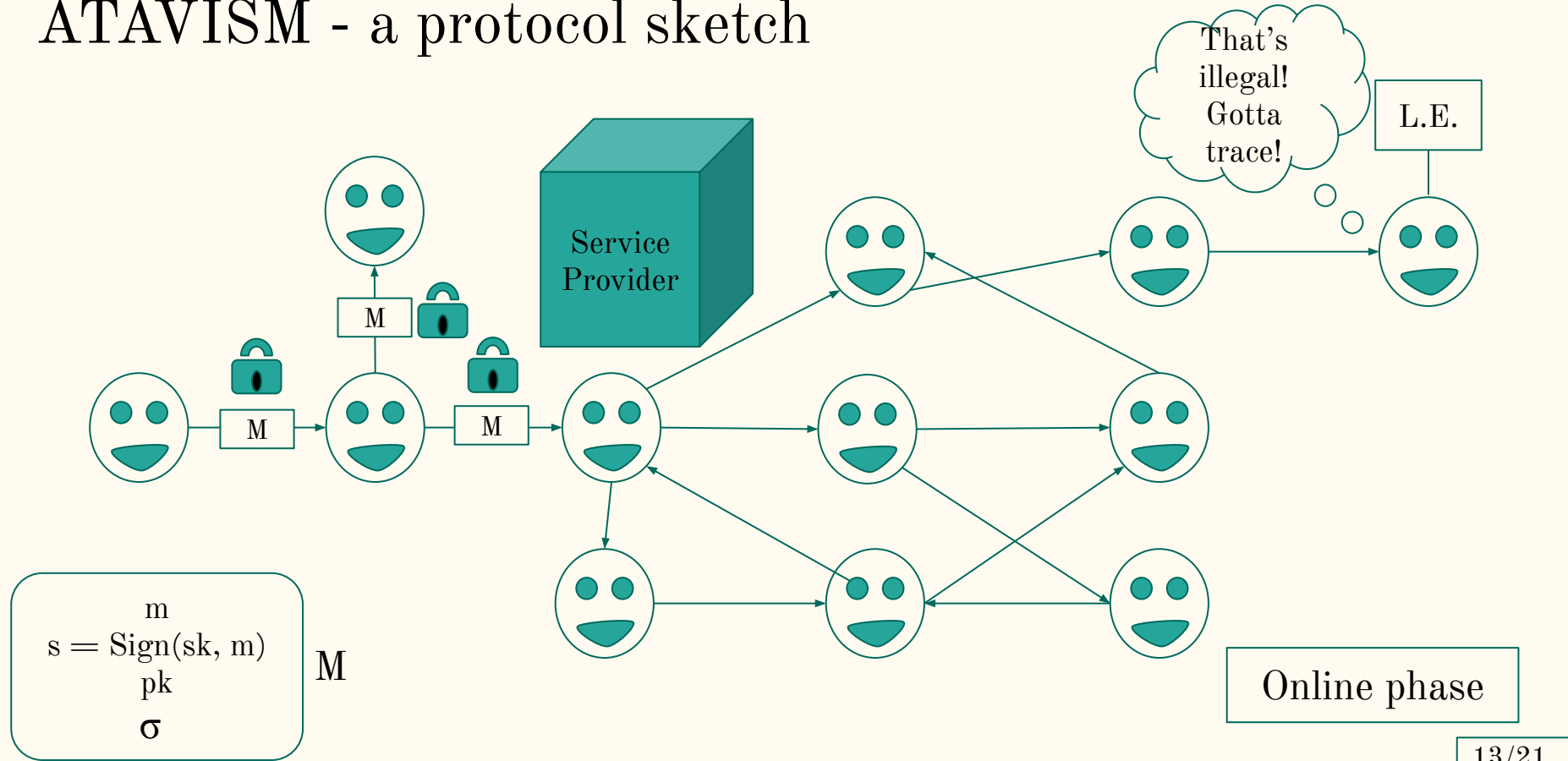
ATAVISM - a protocol sketch



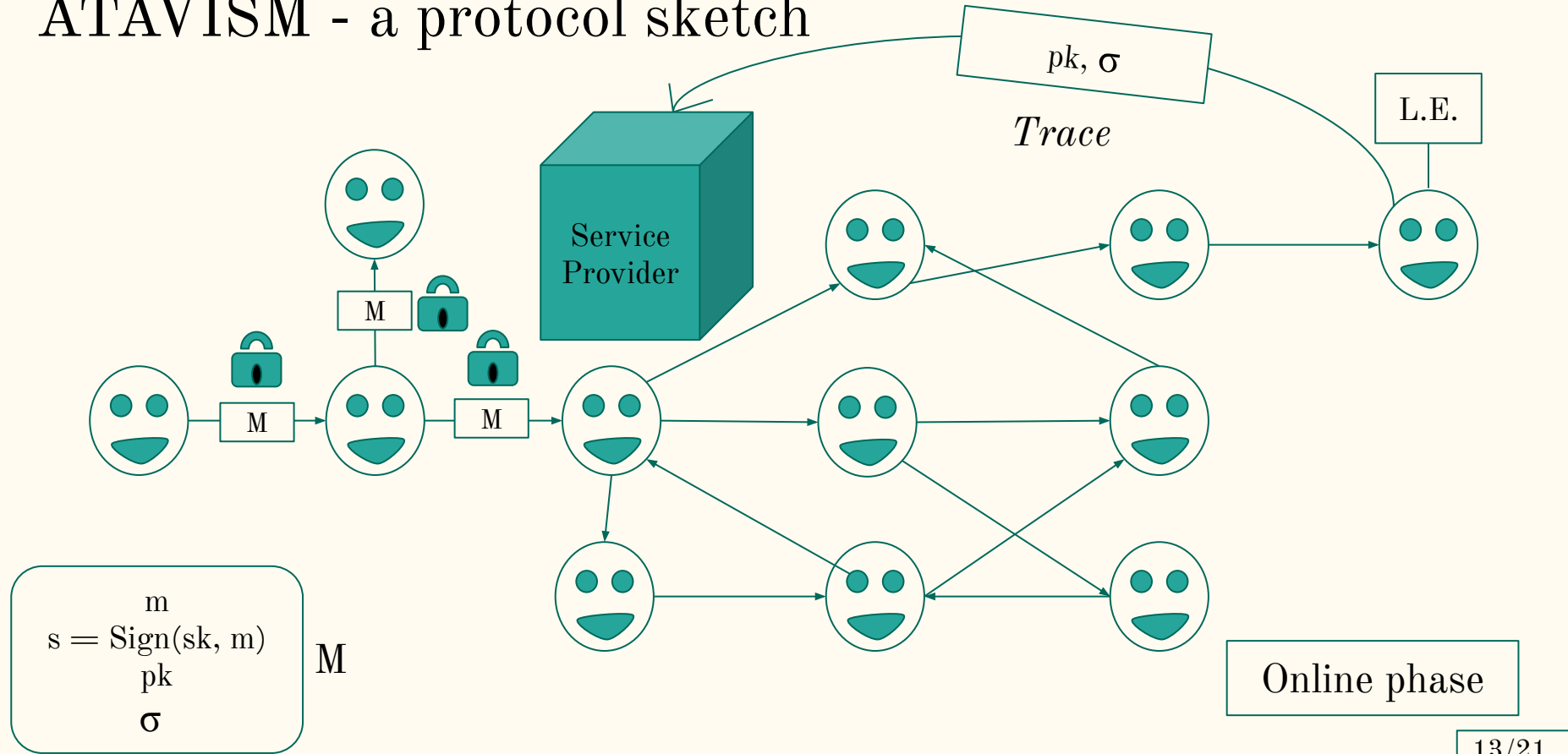
ATAVISM - a protocol sketch



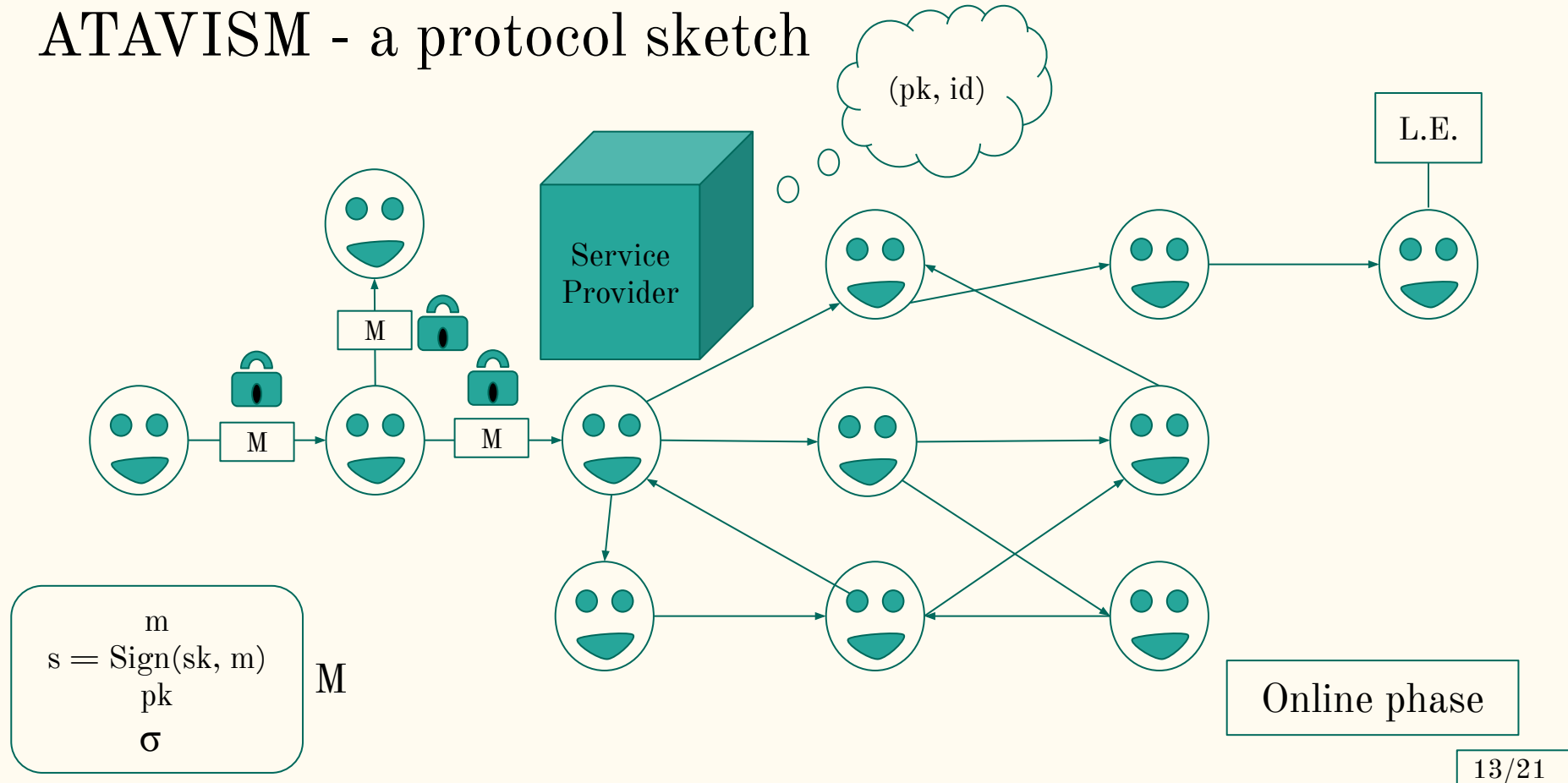
ATAVISM - a protocol sketch



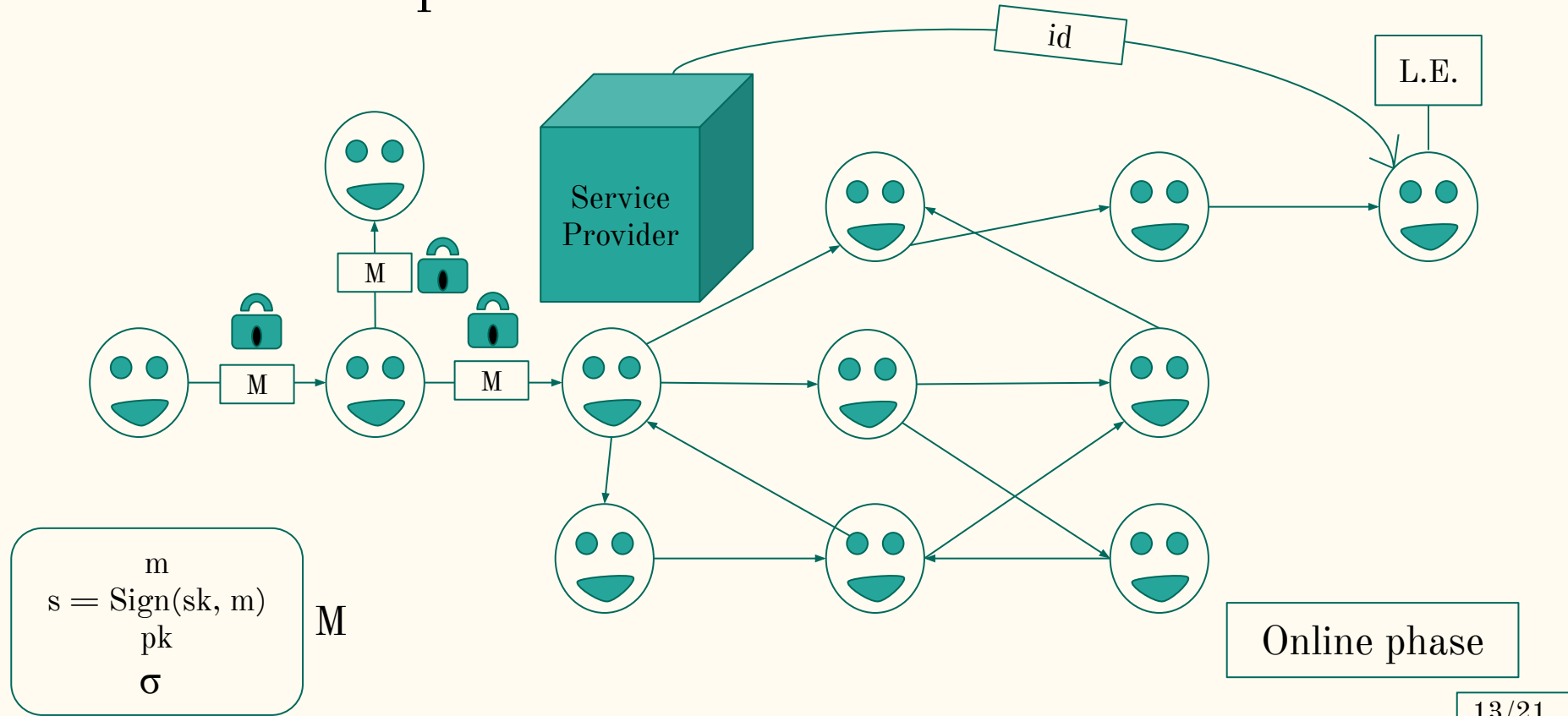
ATAVISM - a protocol sketch



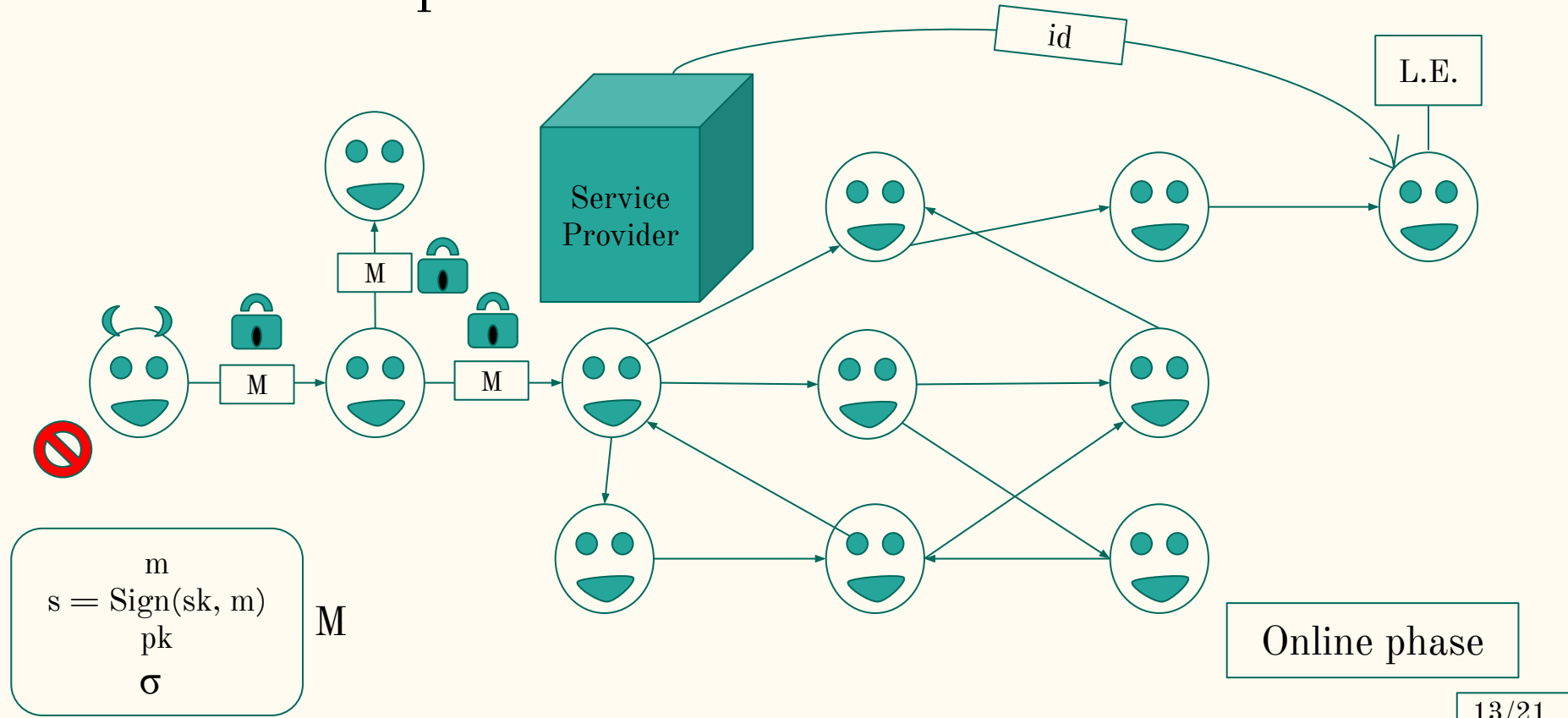
ATAVISM - a protocol sketch



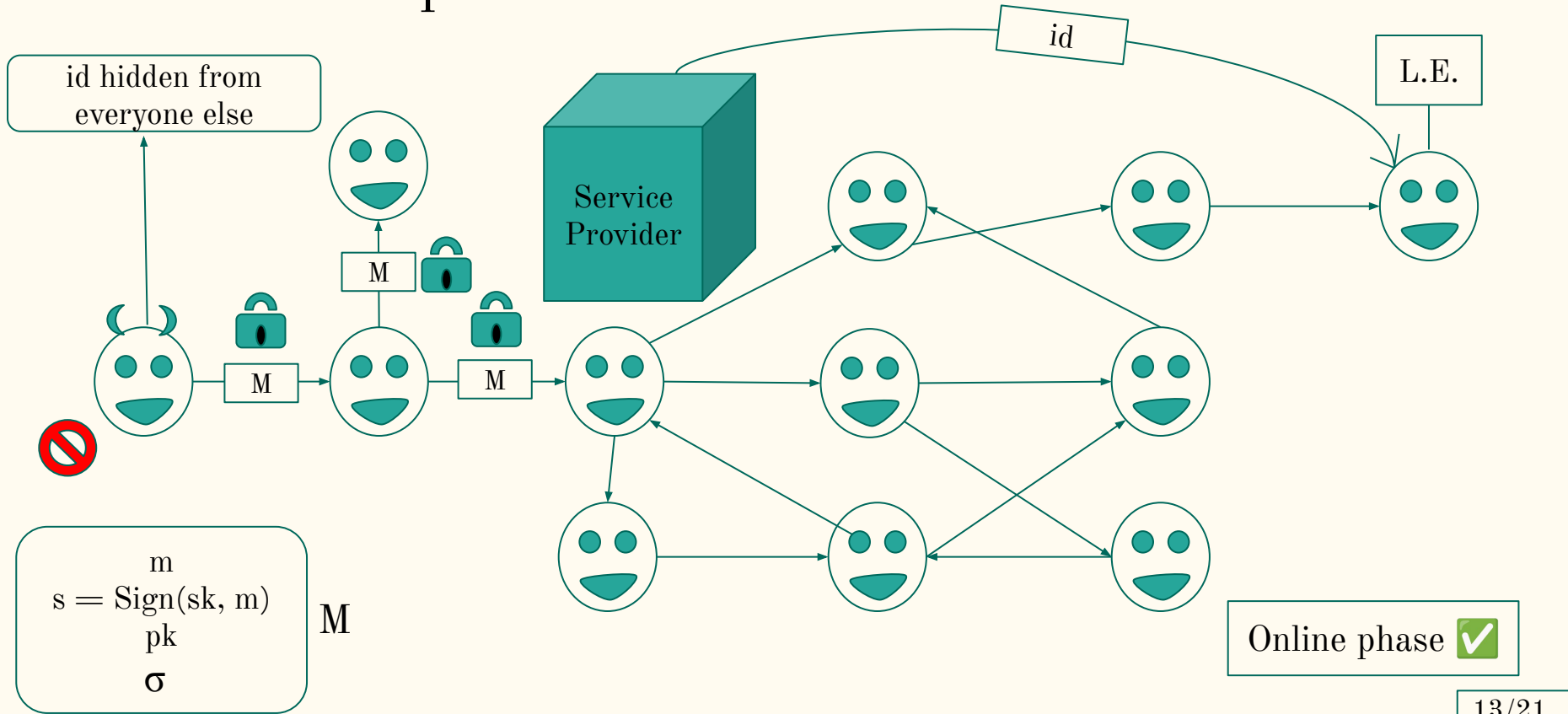
ATAVISM - a protocol sketch



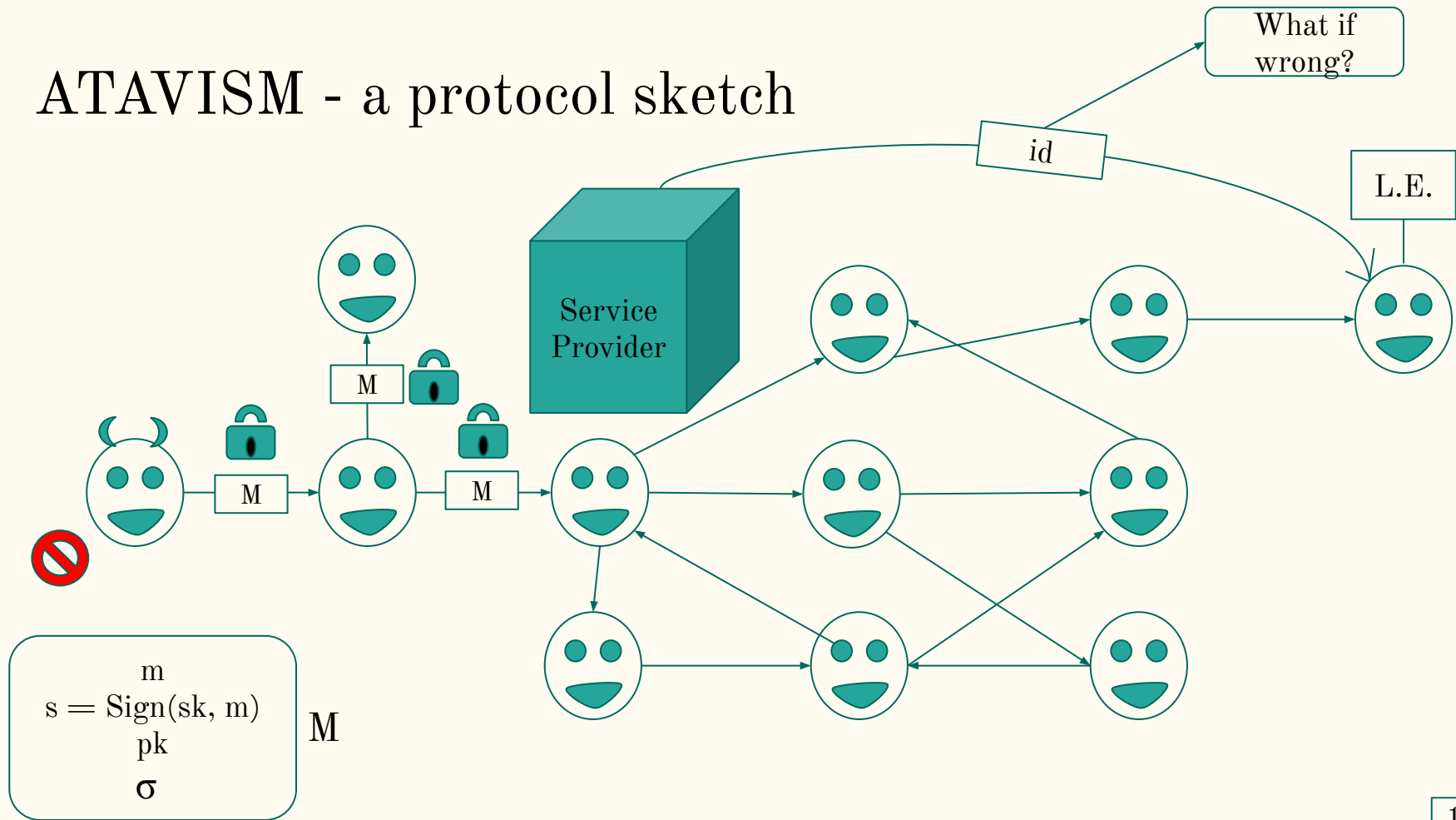
ATAVISM - a protocol sketch



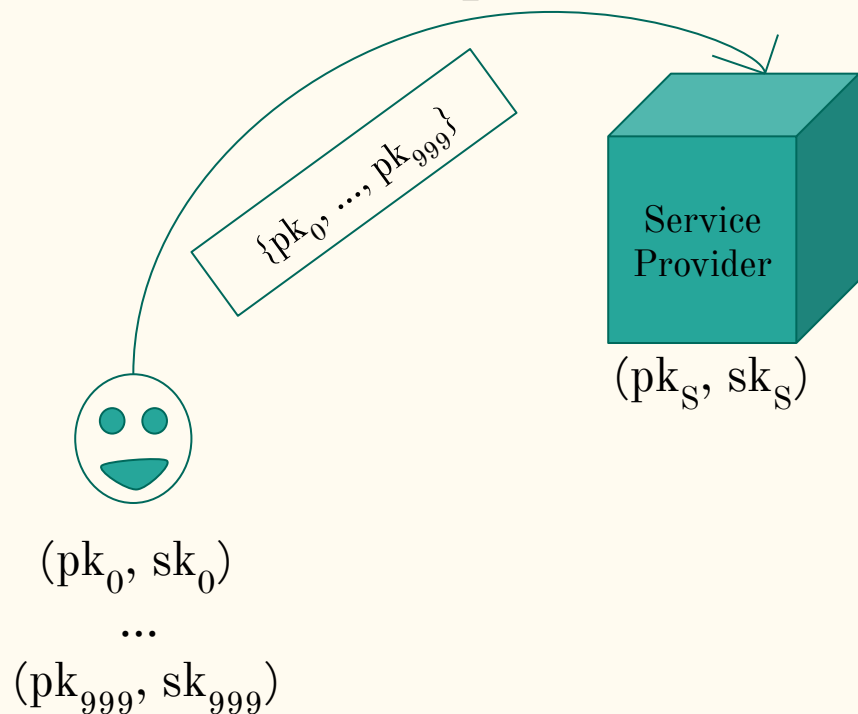
ATAVISM - a protocol sketch



ATAVISM - a protocol sketch

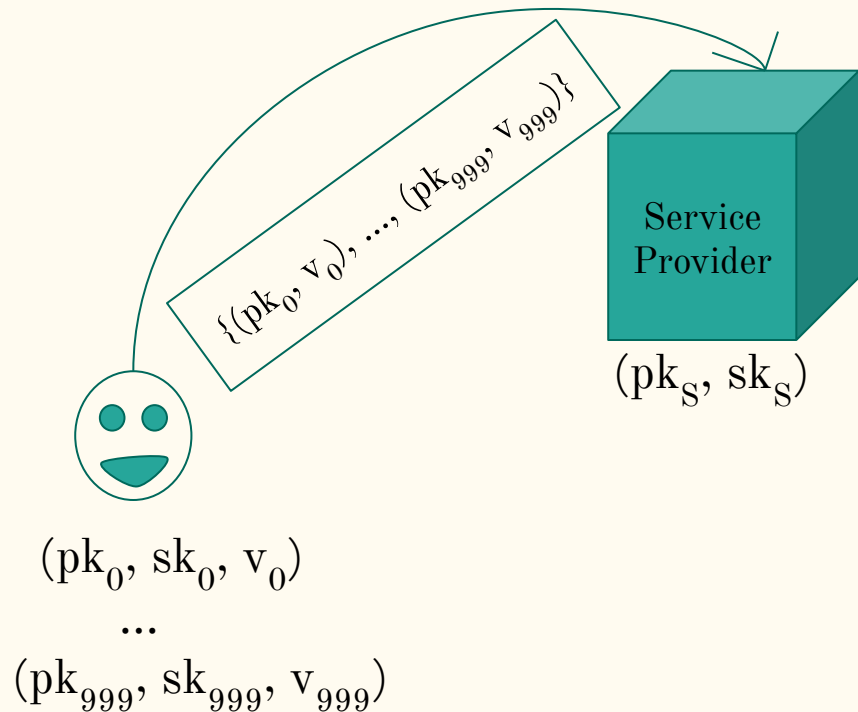


ATAVISM - a protocol sketch



- ✓ Prevent user from registering pk that does not belong to them
- ✓ Prevent service provider from framing user

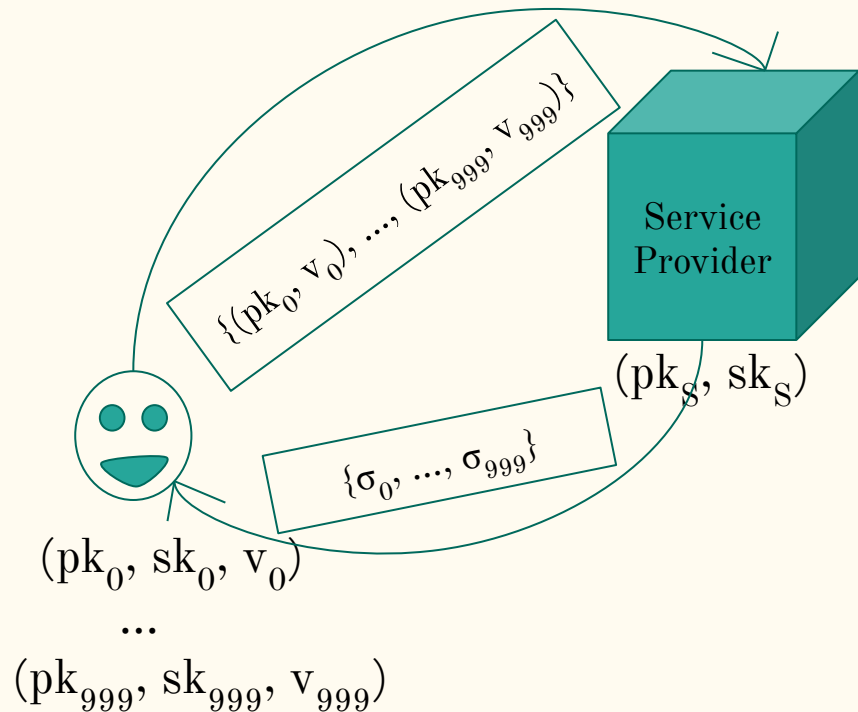
ATAVISM - a protocol sketch



- ✓ Prevent user from registering pk that does not belong to them
- ✓ Prevent service provider from framing user

$$v_i = \text{Sign}(sk_i, pk_i)$$

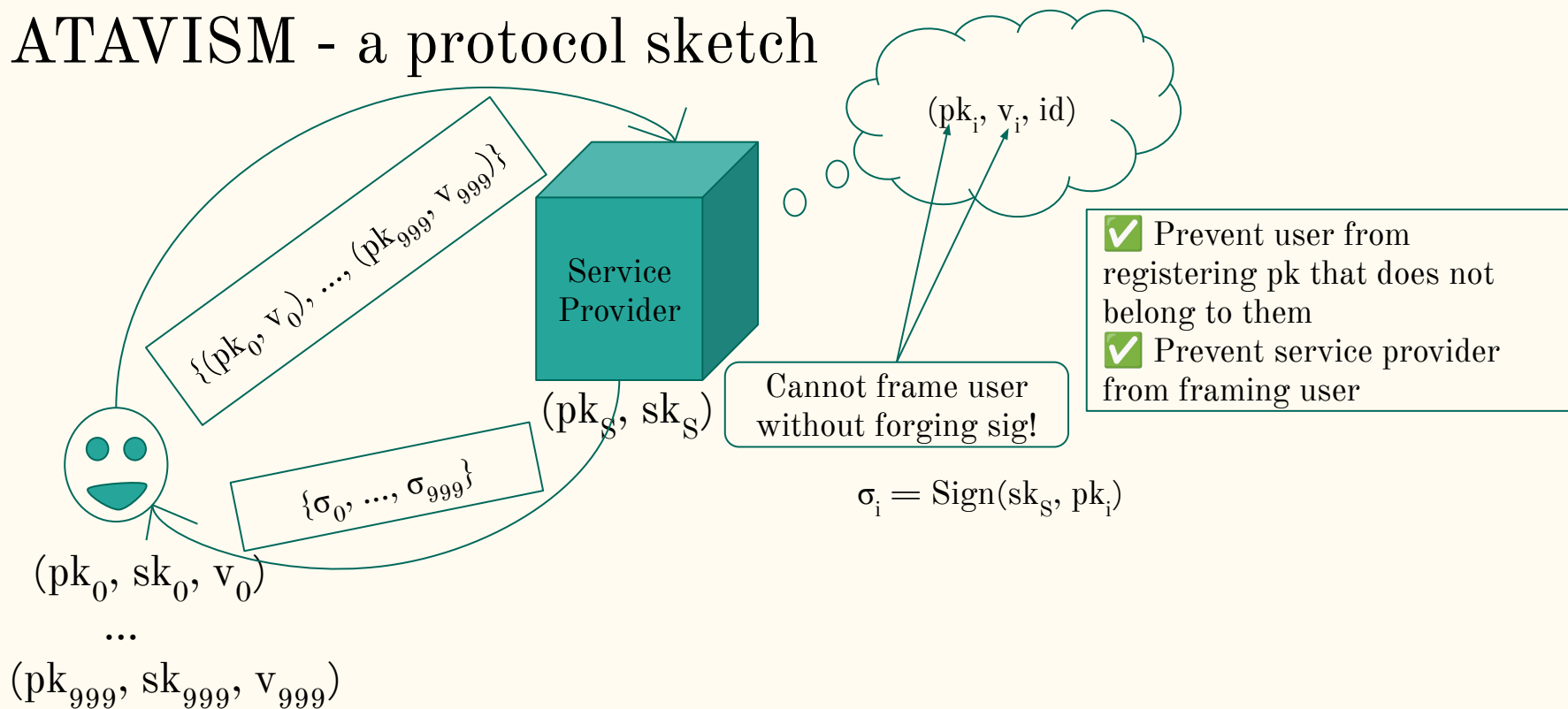
ATAVISM - a protocol sketch



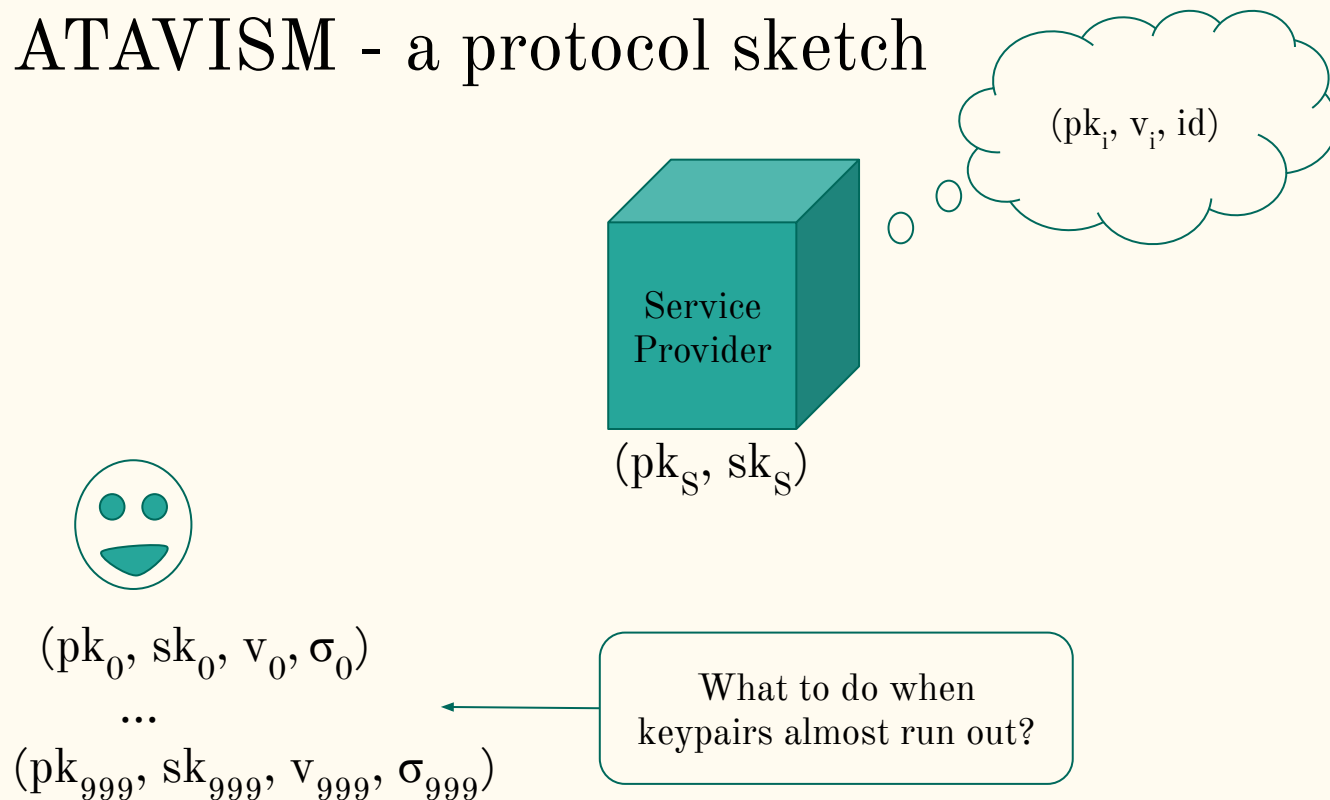
- ✓ Prevent user from registering pk that does not belong to them
- ✓ Prevent service provider from framing user

$$\sigma_i = \text{Sign}(sk_s, pk_i)$$

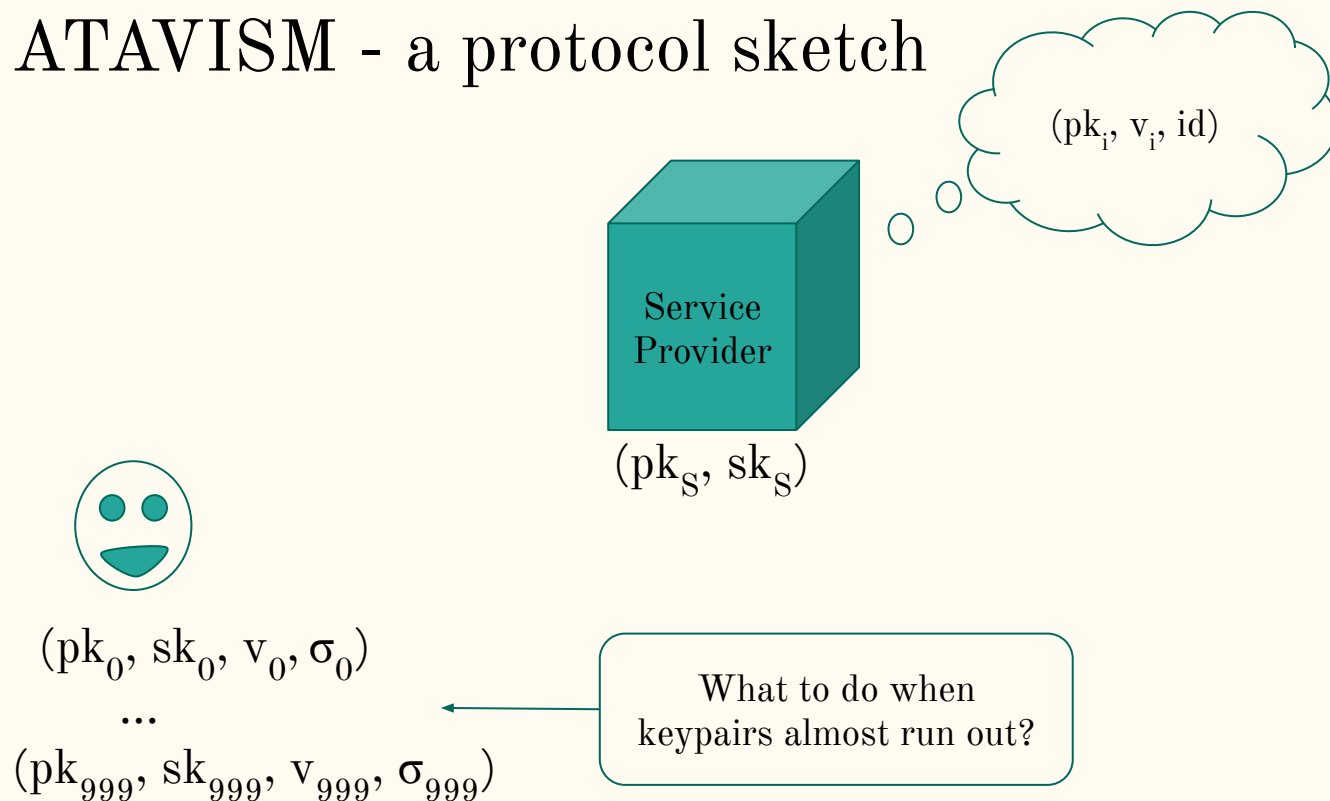
ATAVISM - a protocol sketch



ATAVISM - a protocol sketch

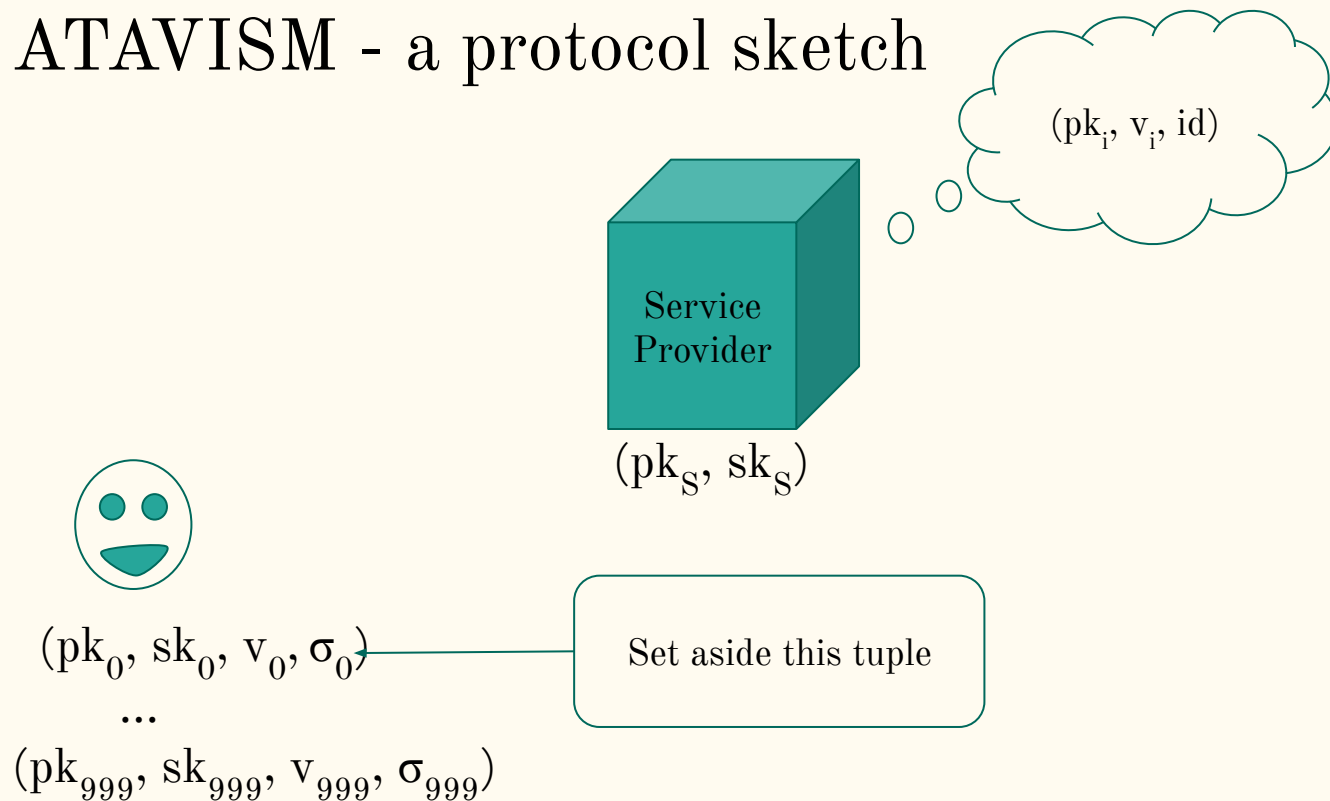


ATAVISM - a protocol sketch



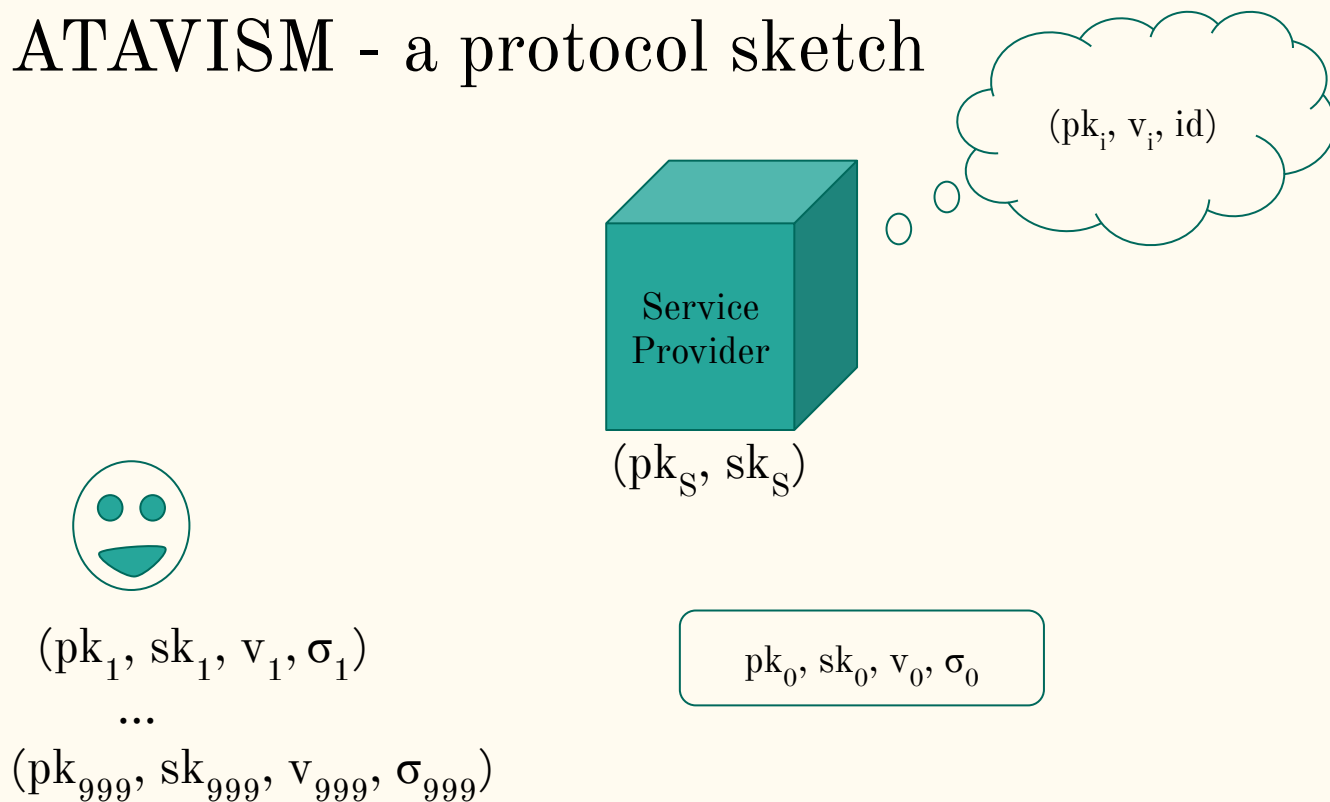
Refresh phase

ATAVISM - a protocol sketch



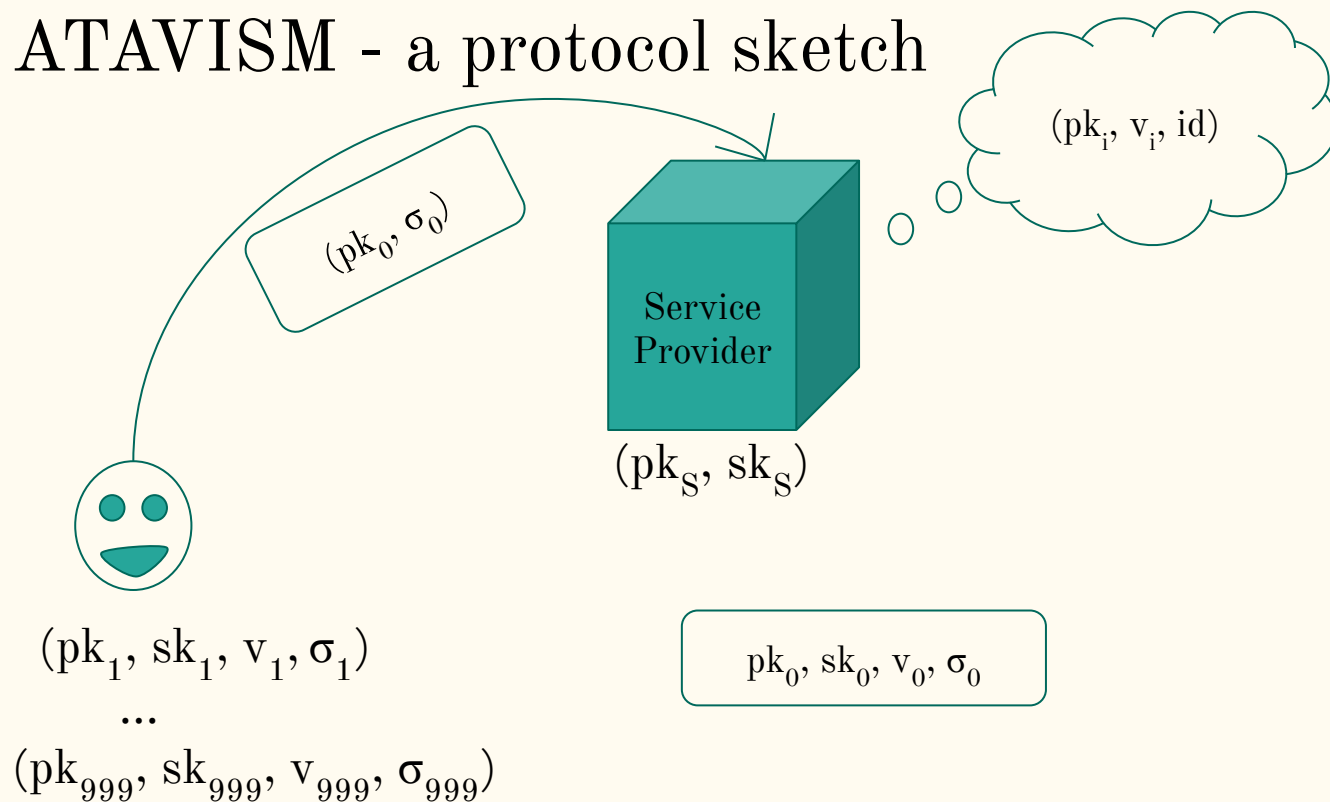
Refresh phase

ATAVISM - a protocol sketch



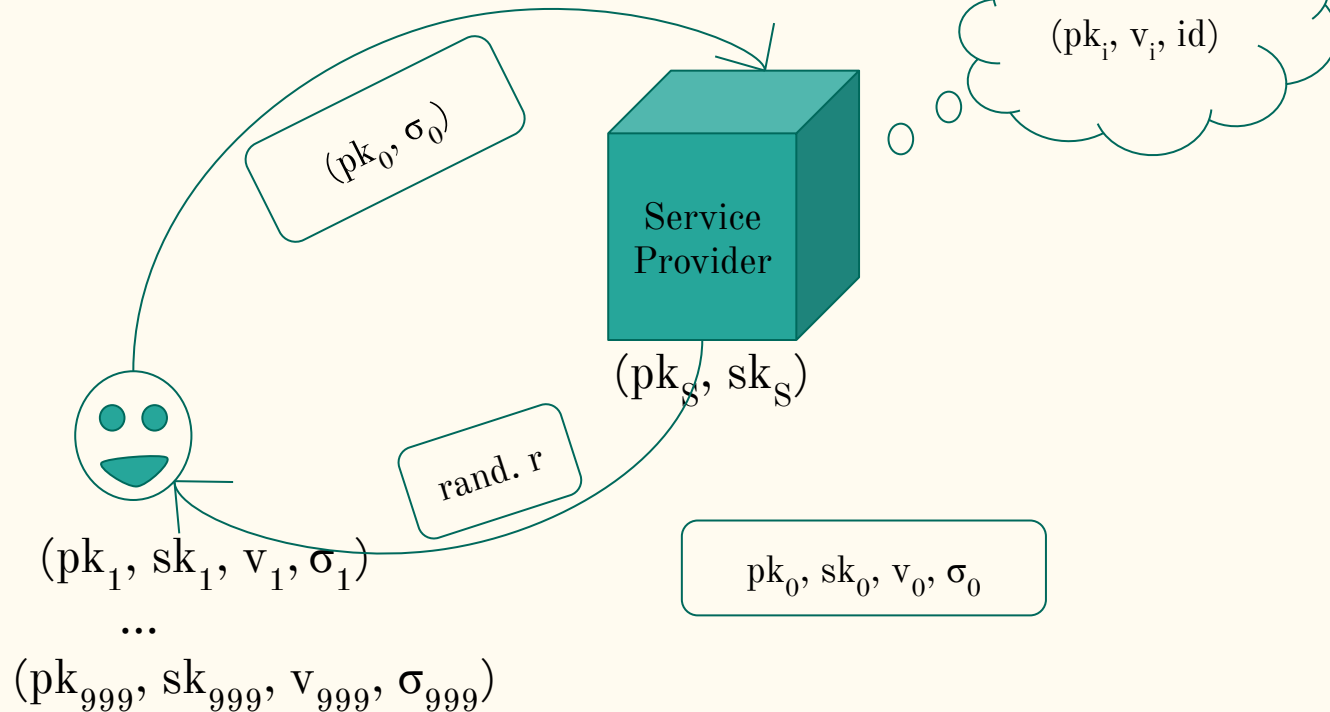
Refresh phase

ATAVISM - a protocol sketch



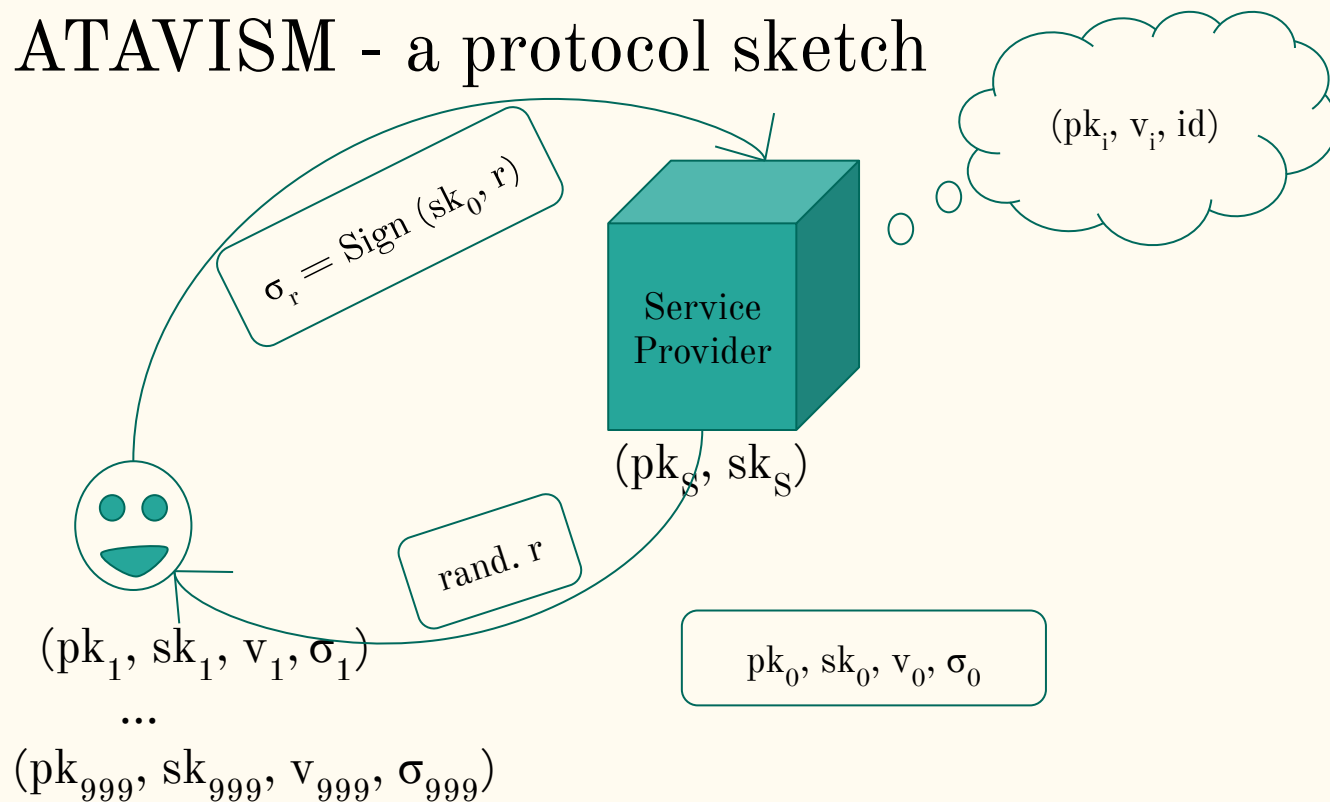
Refresh phase

ATAVISM - a protocol sketch



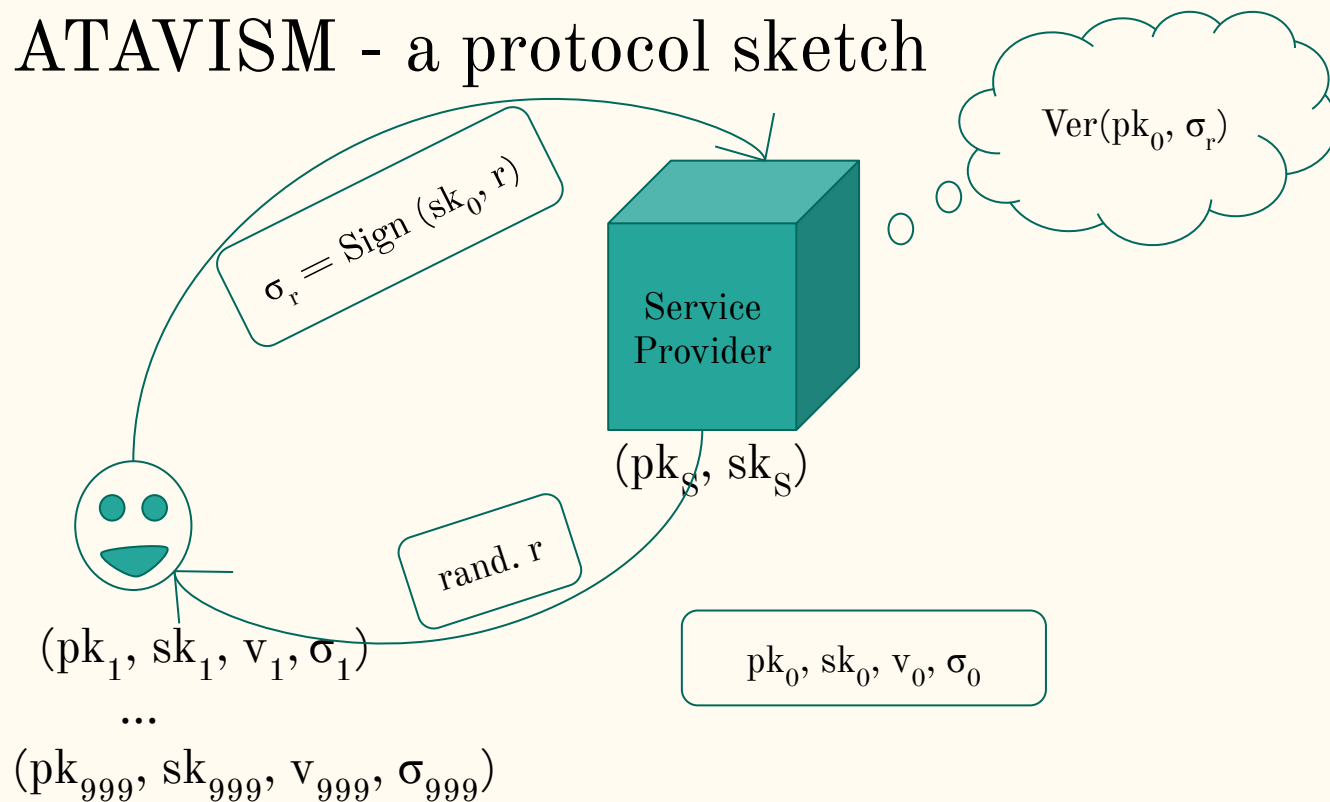
Refresh phase

ATAVISM - a protocol sketch



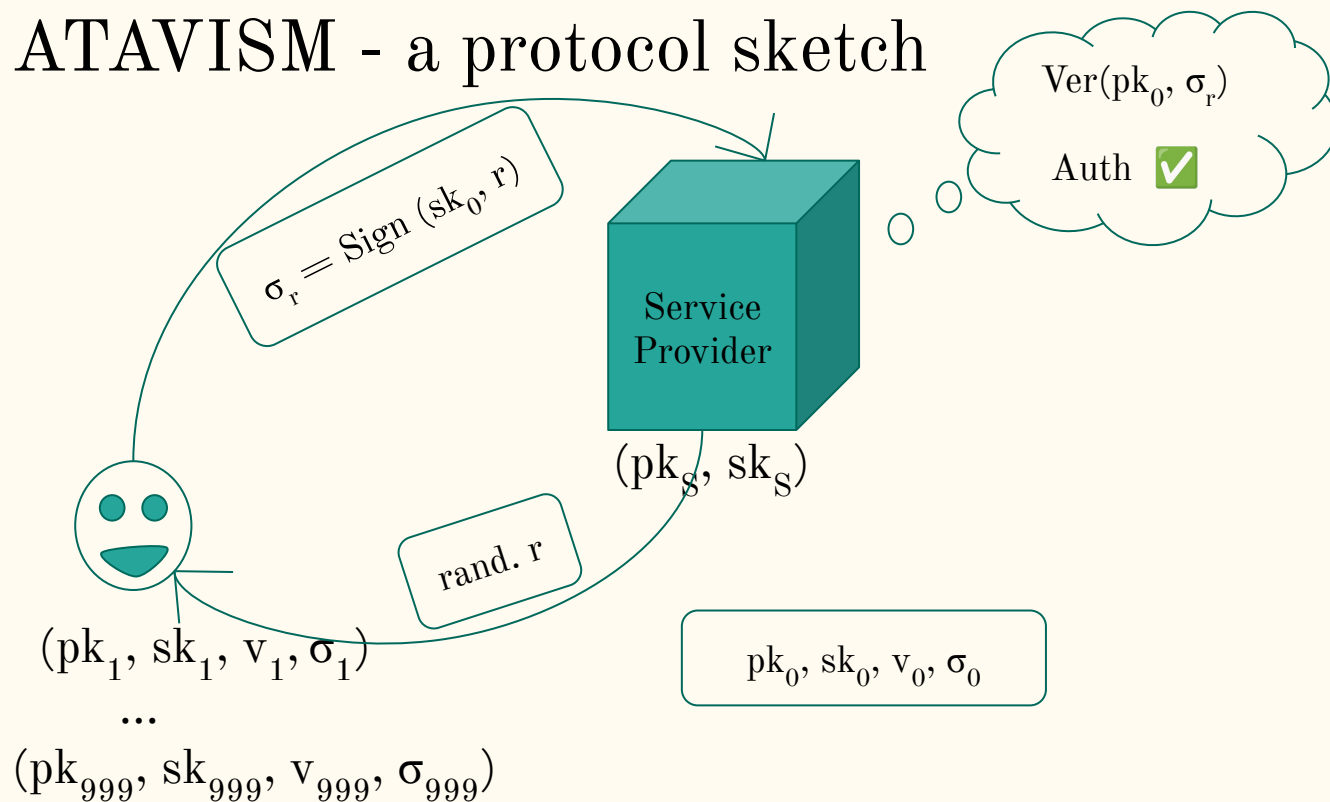
Refresh phase

ATAVISM - a protocol sketch



Refresh phase

ATAVISM - a protocol sketch



Refresh phase

ATAVISM - a protocol sketch



(pk_s, sk_s)



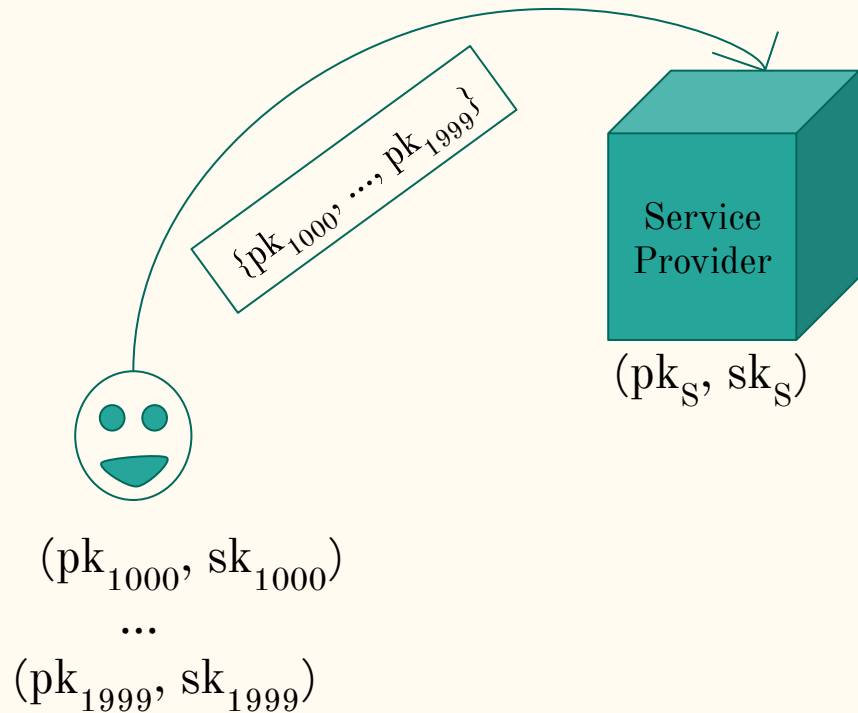
(pk_{1000}, sk_{1000})

...

(pk_{1999}, sk_{1999})

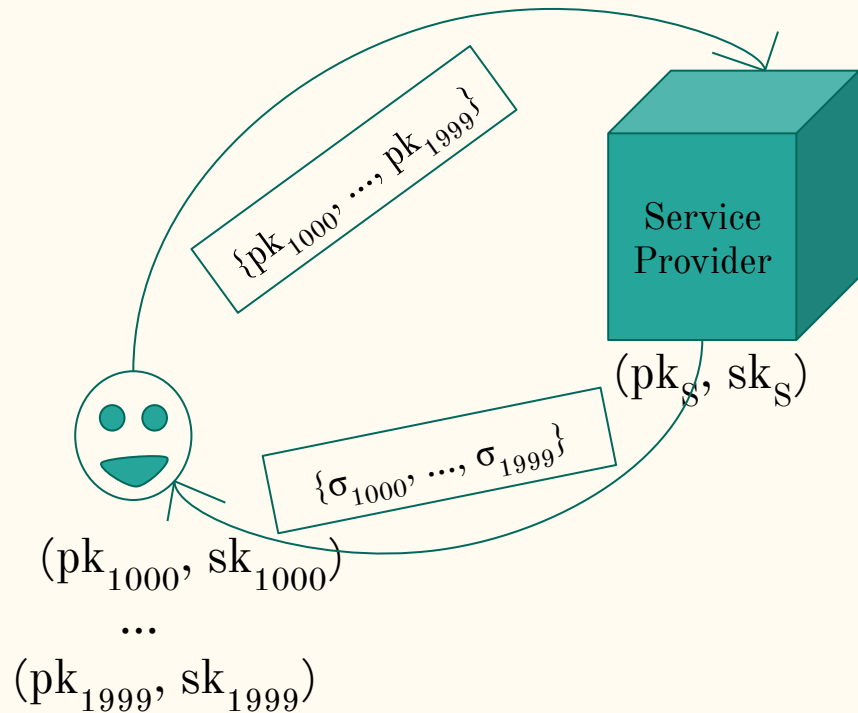
Refresh phase

ATAVISM - a protocol sketch



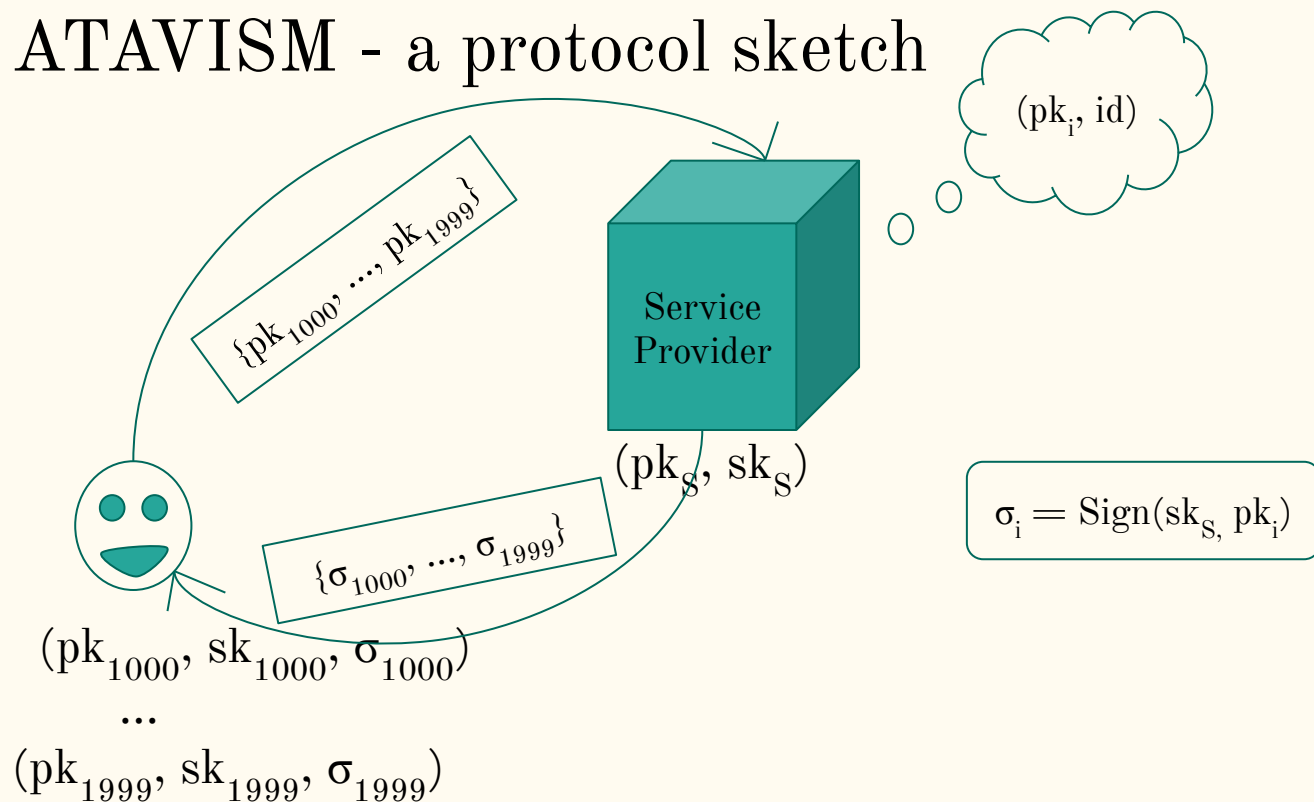
Refresh phase

ATAVISM - a protocol sketch



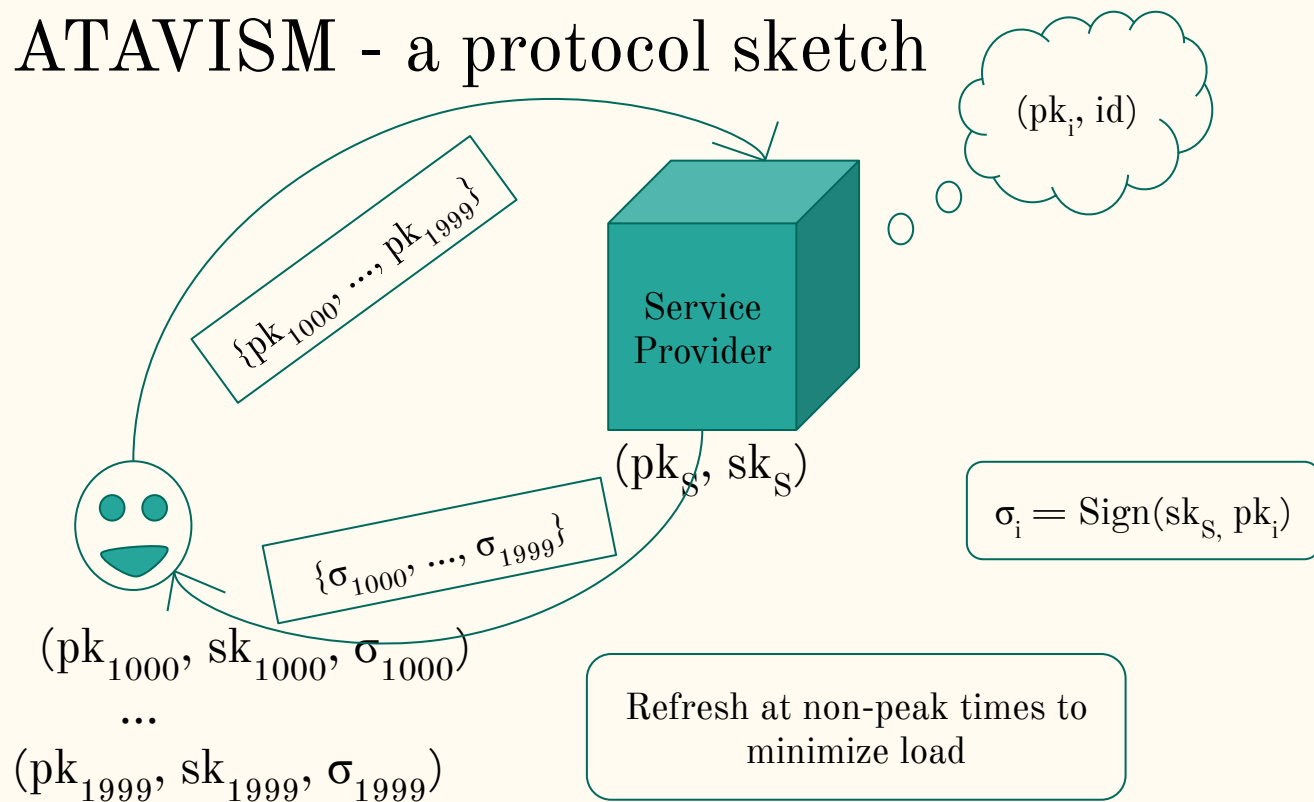
Refresh phase

ATAVISM - a protocol sketch



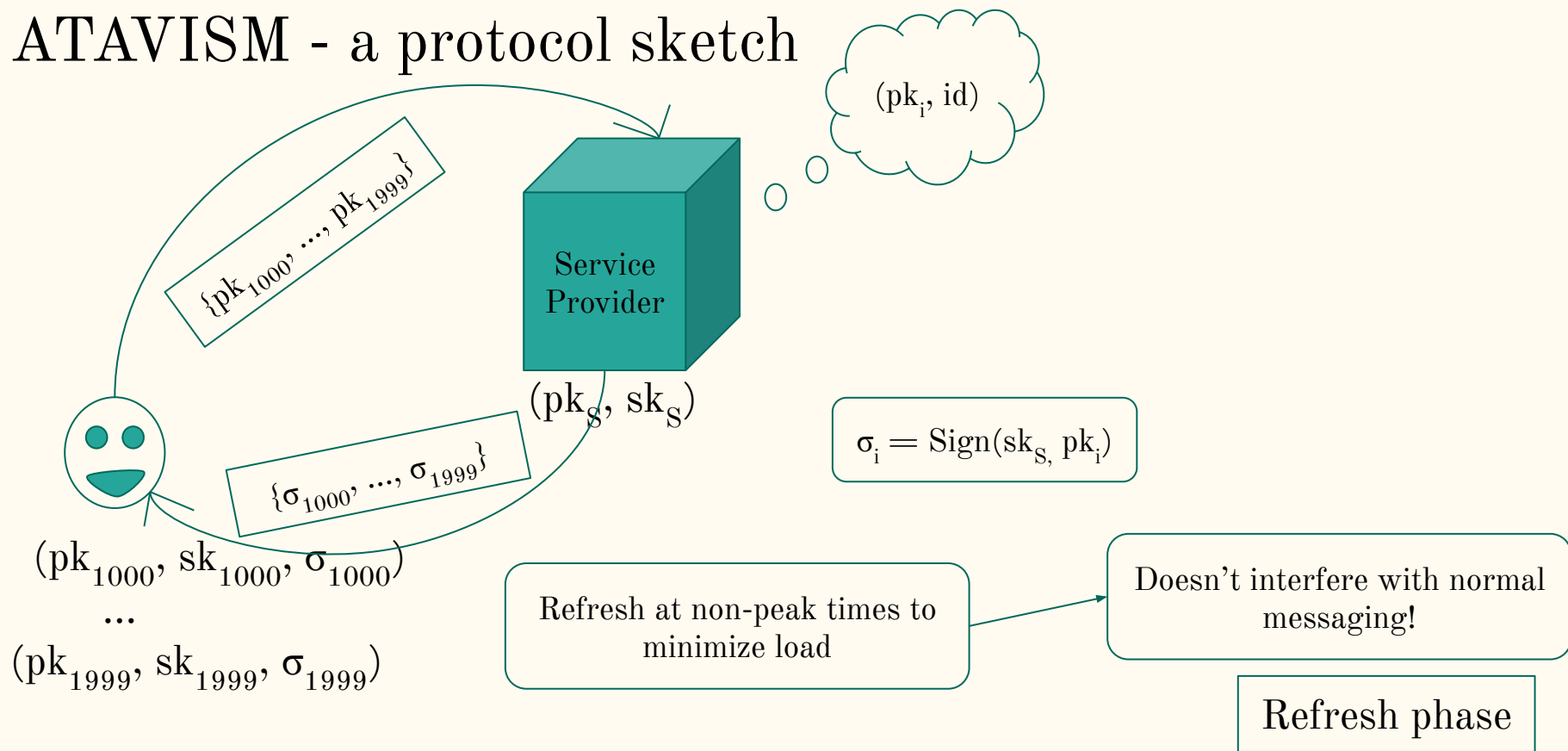
Refresh phase

ATAVISM - a protocol sketch

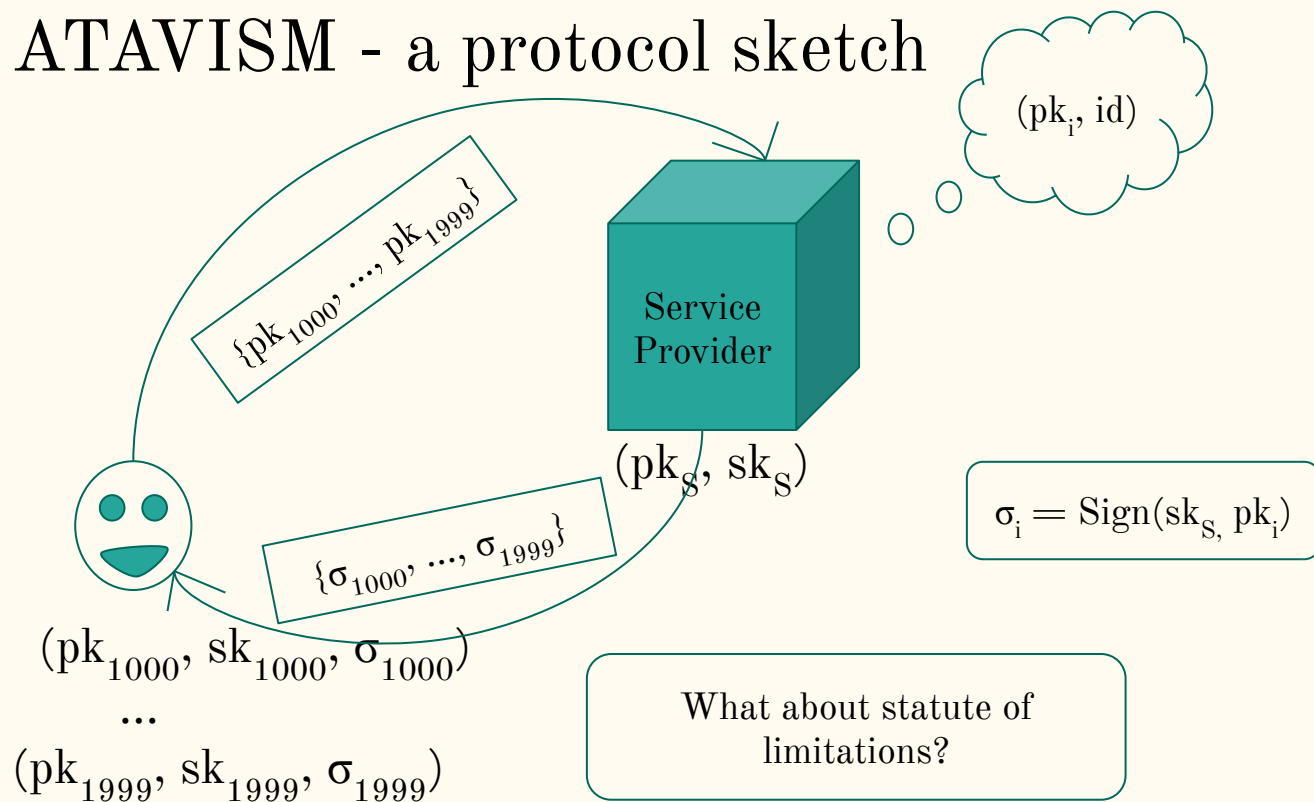


Refresh phase

ATAVISM - a protocol sketch

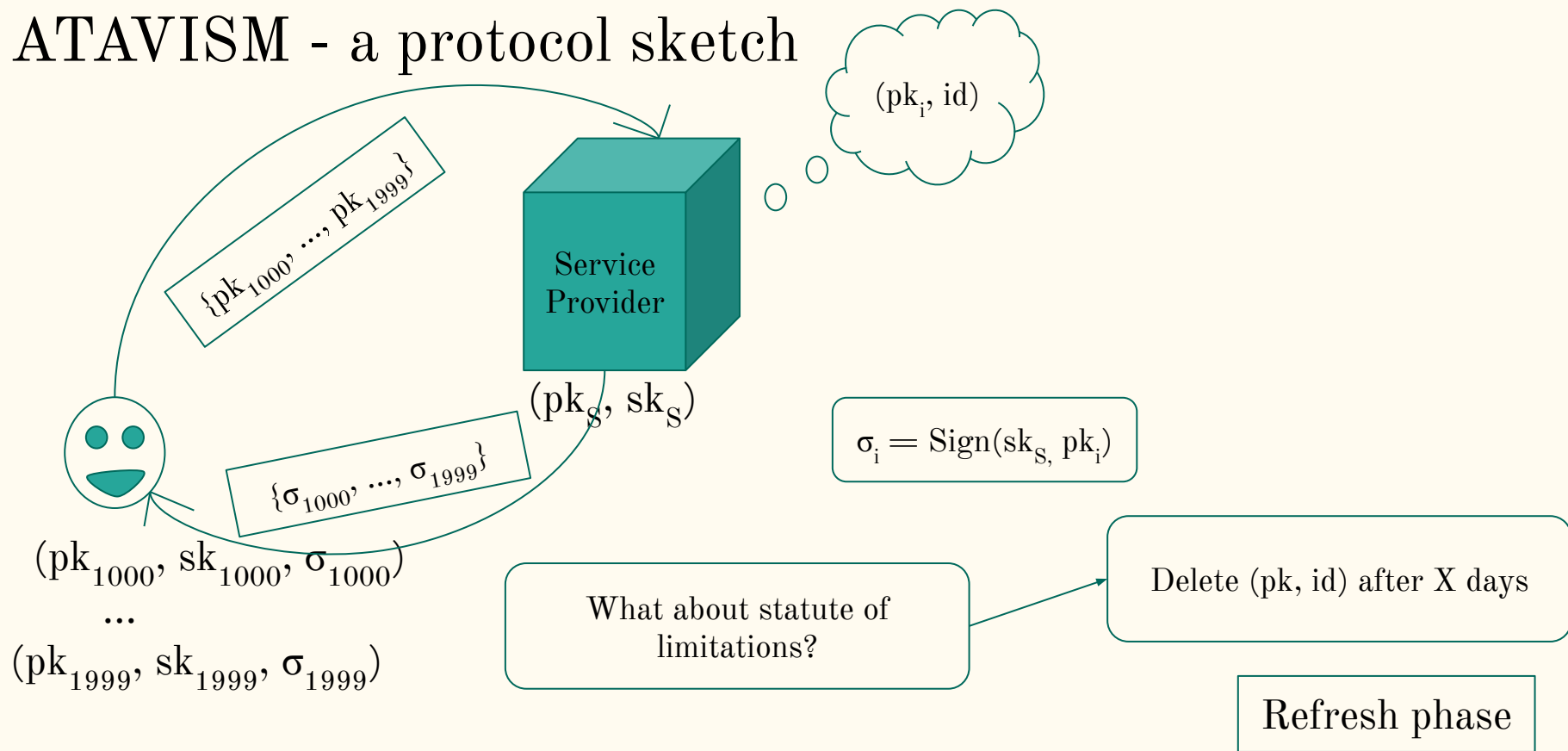


ATAVISM - a protocol sketch

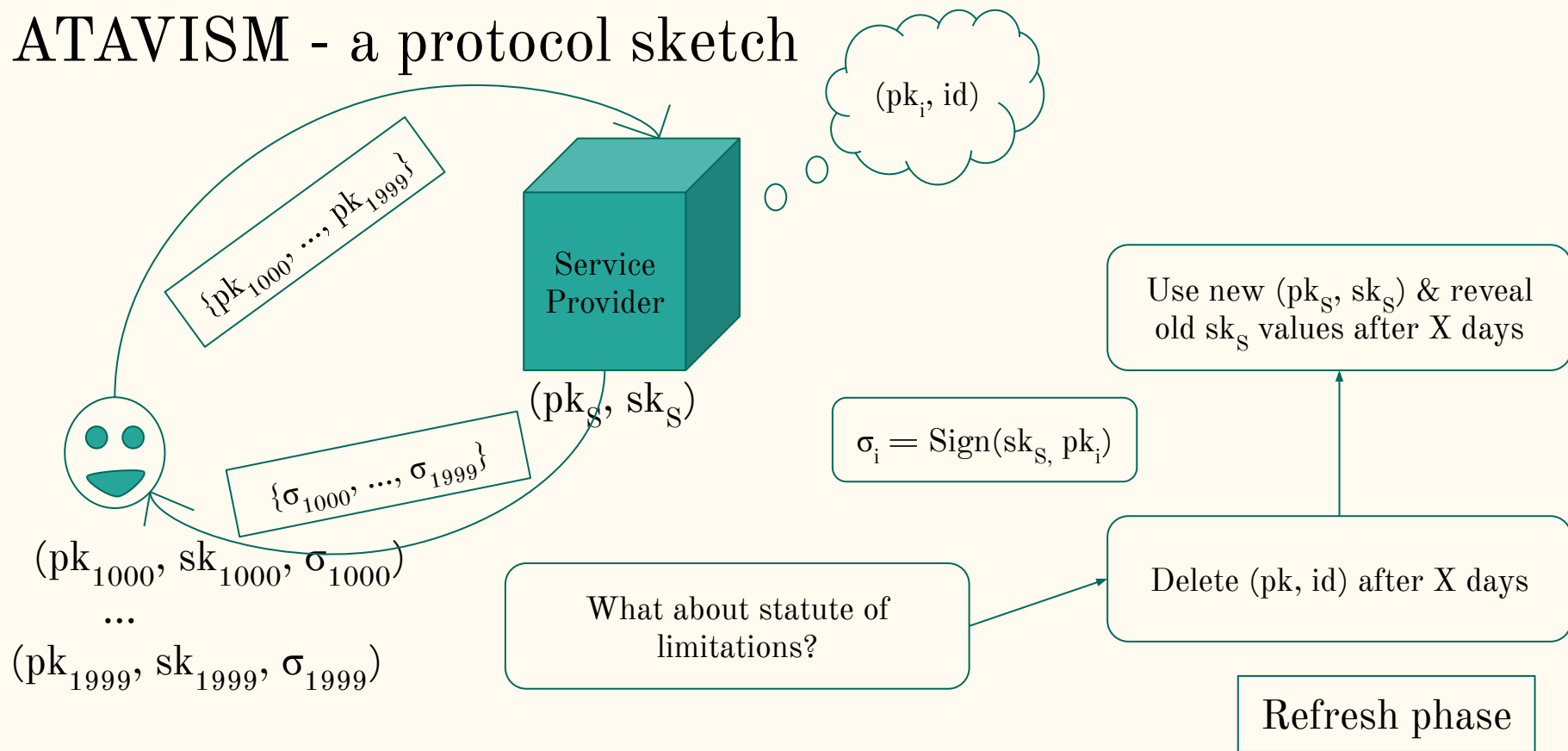


Refresh phase

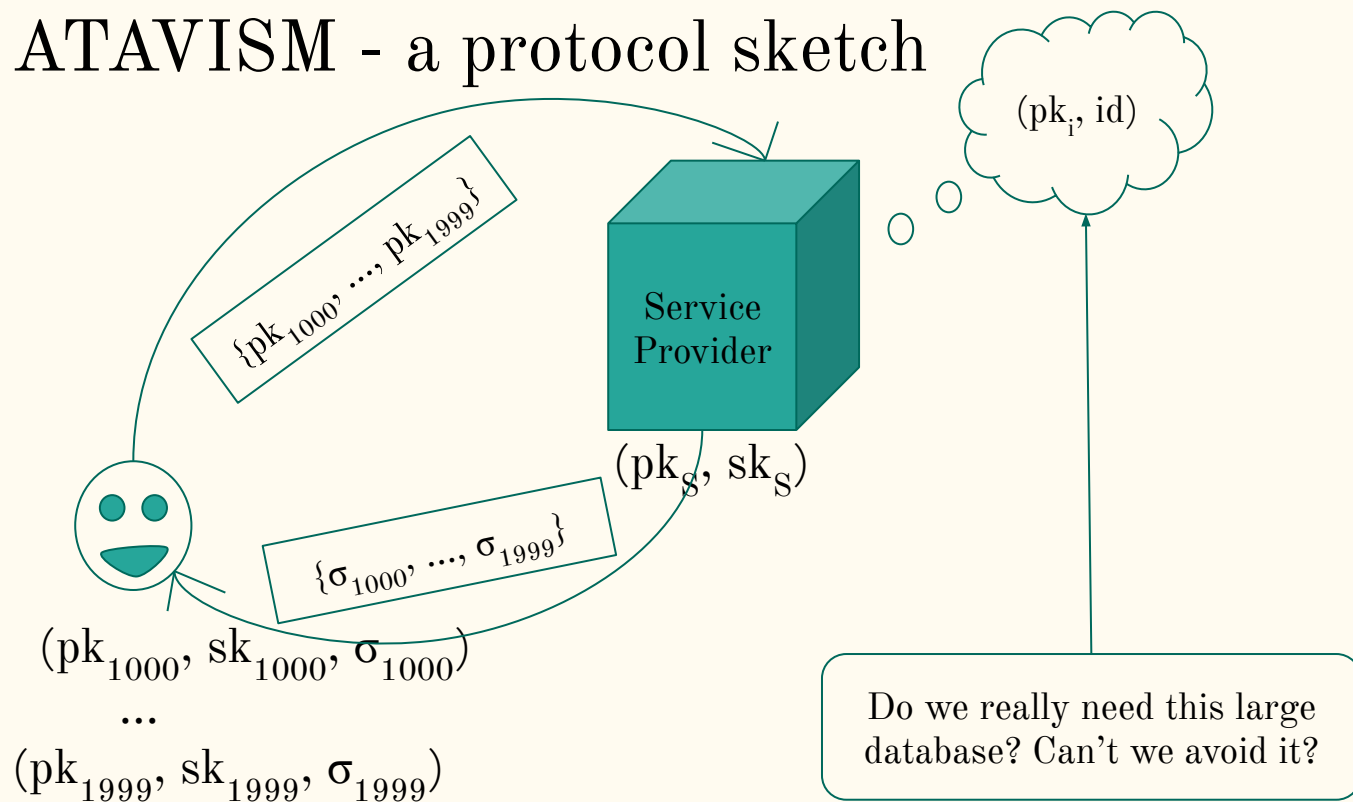
ATAVISM - a protocol sketch



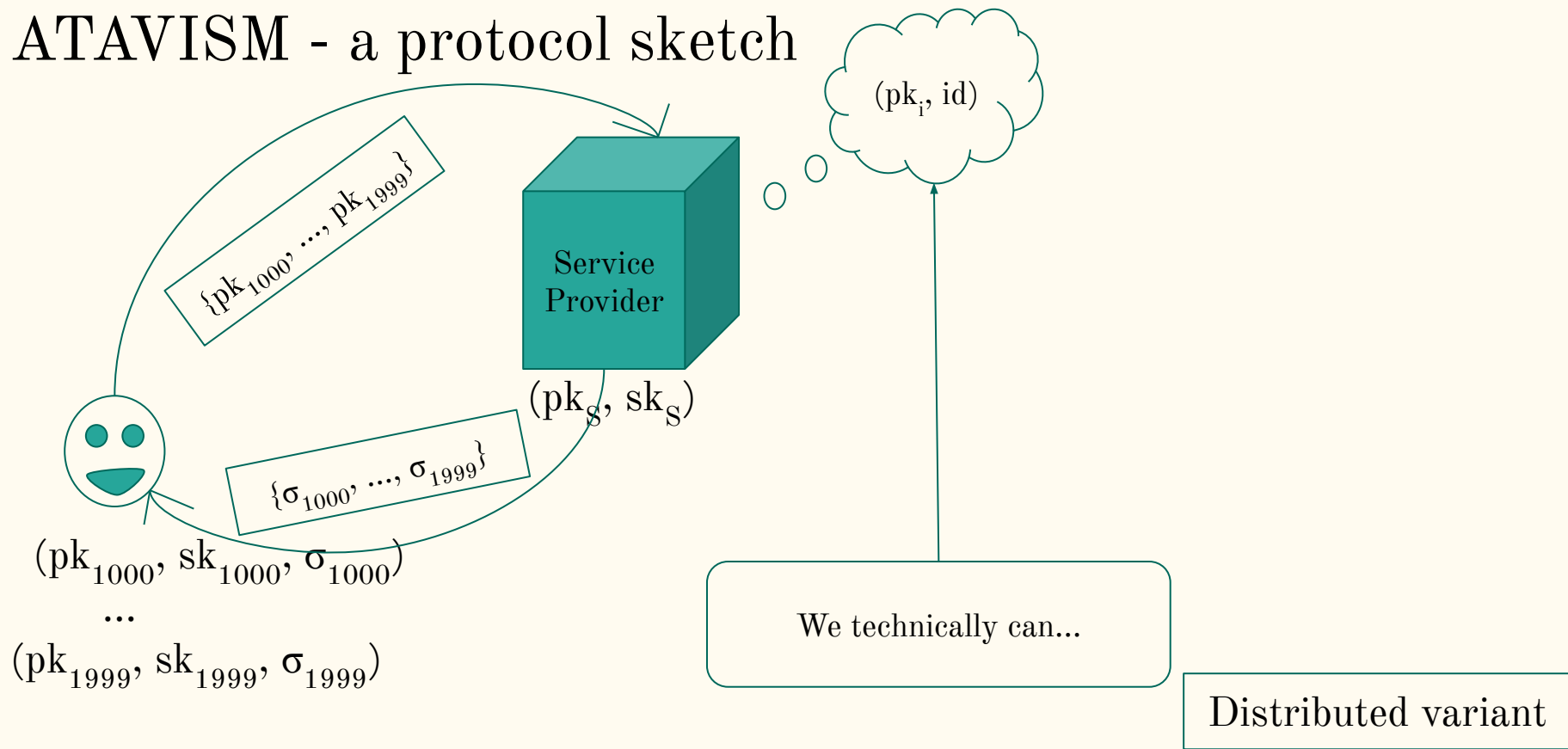
ATAVISM - a protocol sketch



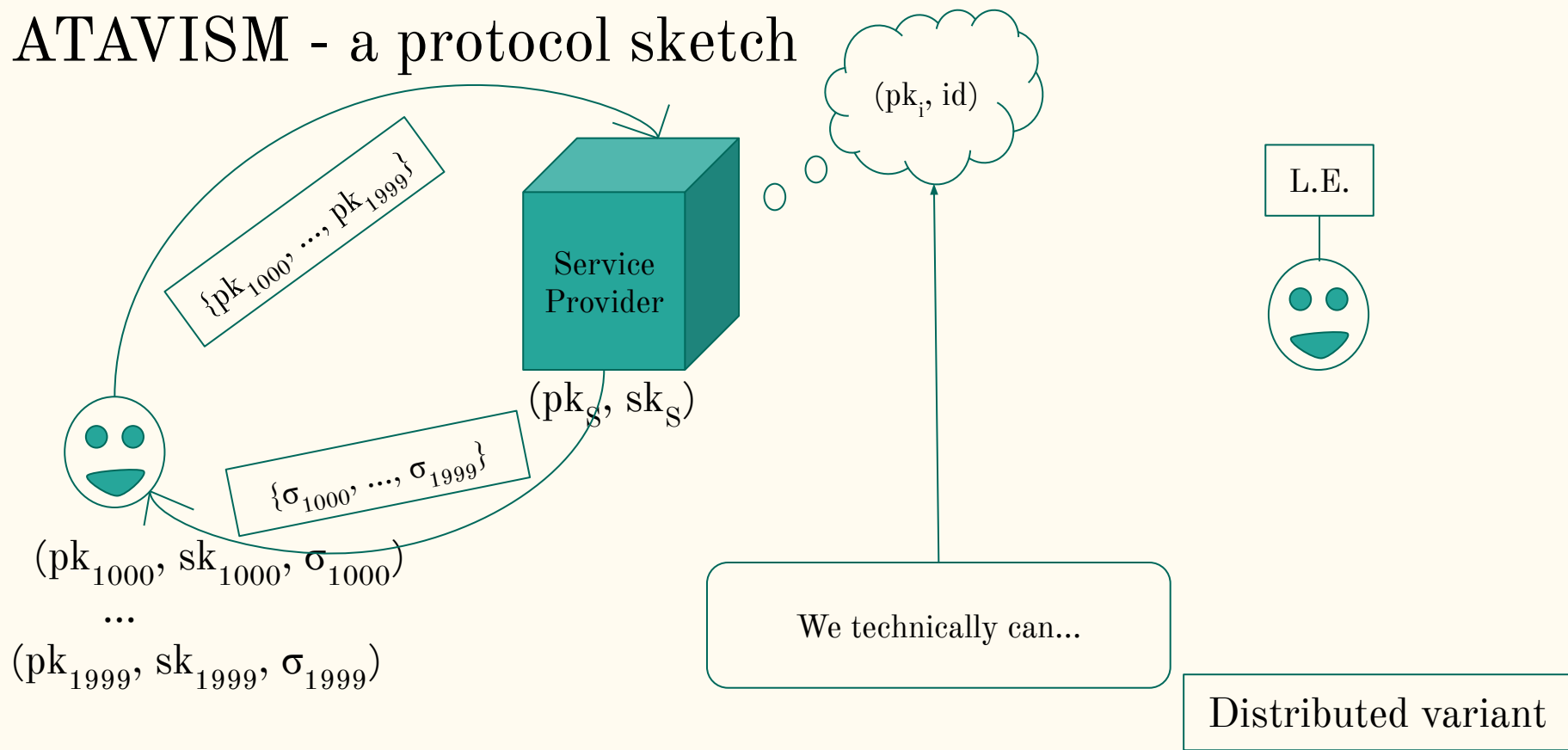
ATAVISM - a protocol sketch



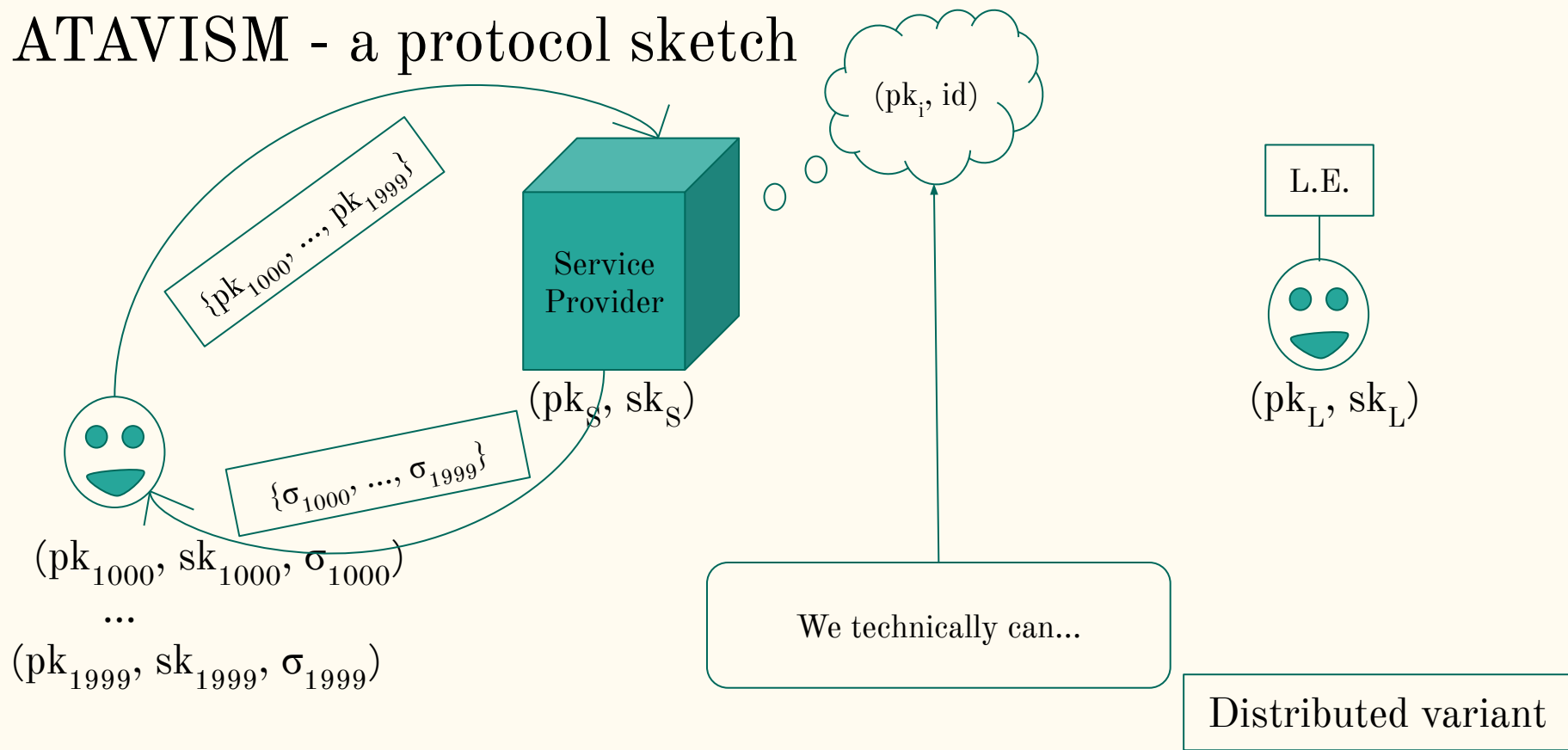
ATAVISM - a protocol sketch



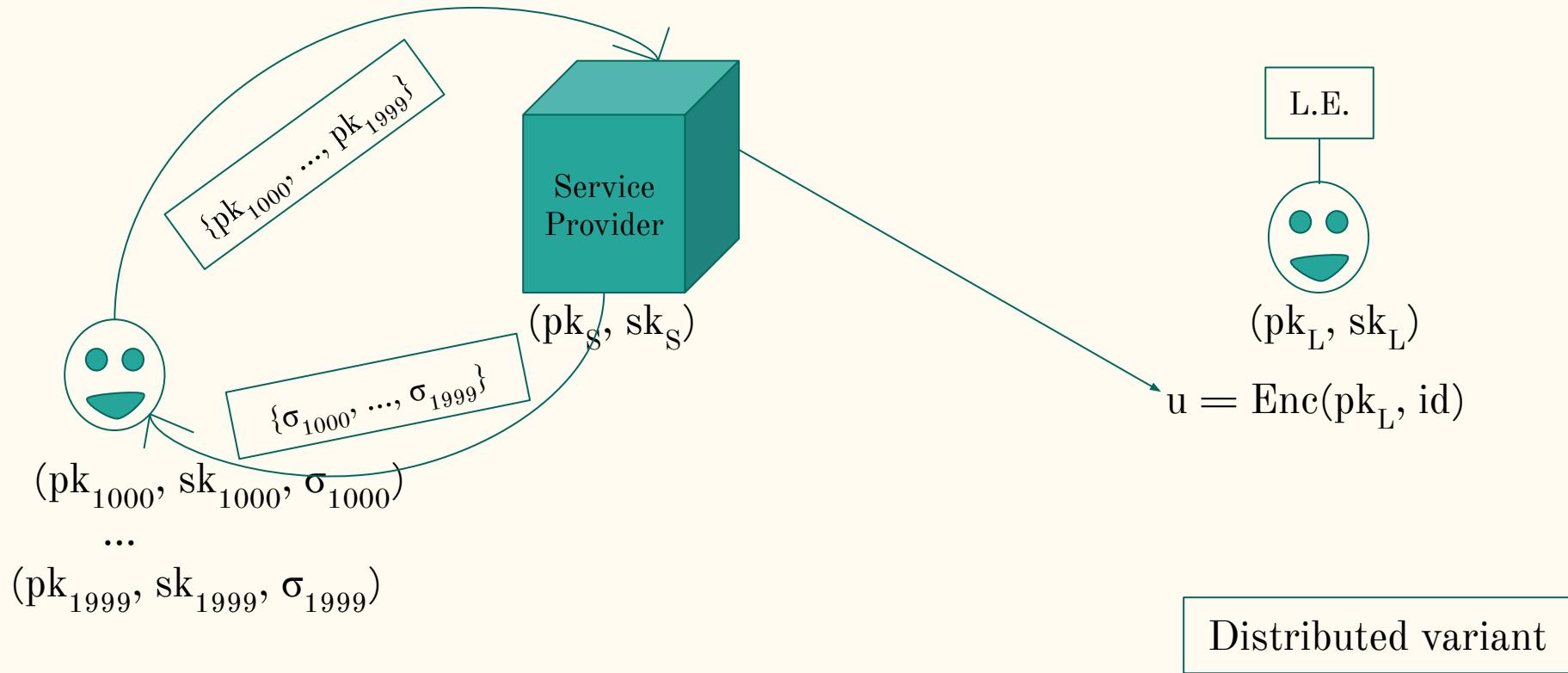
ATAVISM - a protocol sketch



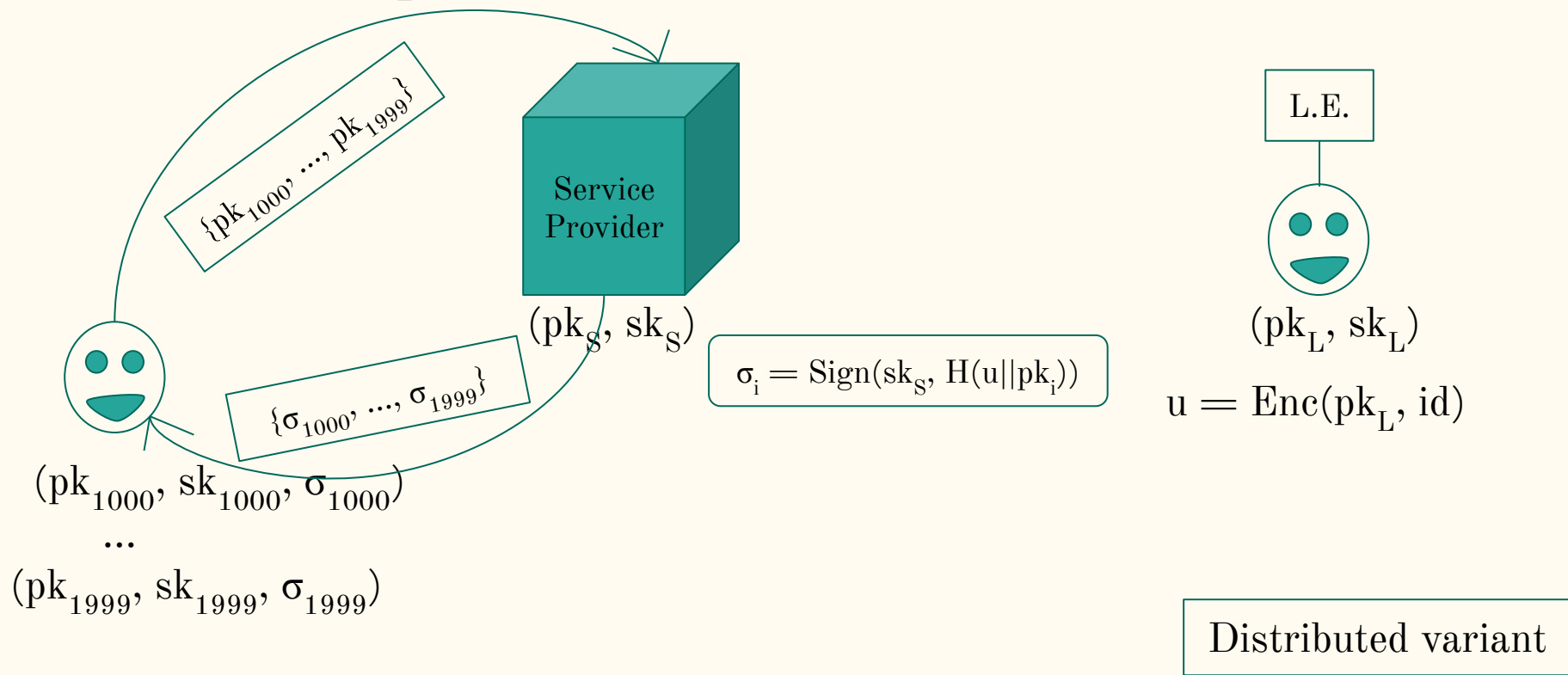
ATAVISM - a protocol sketch



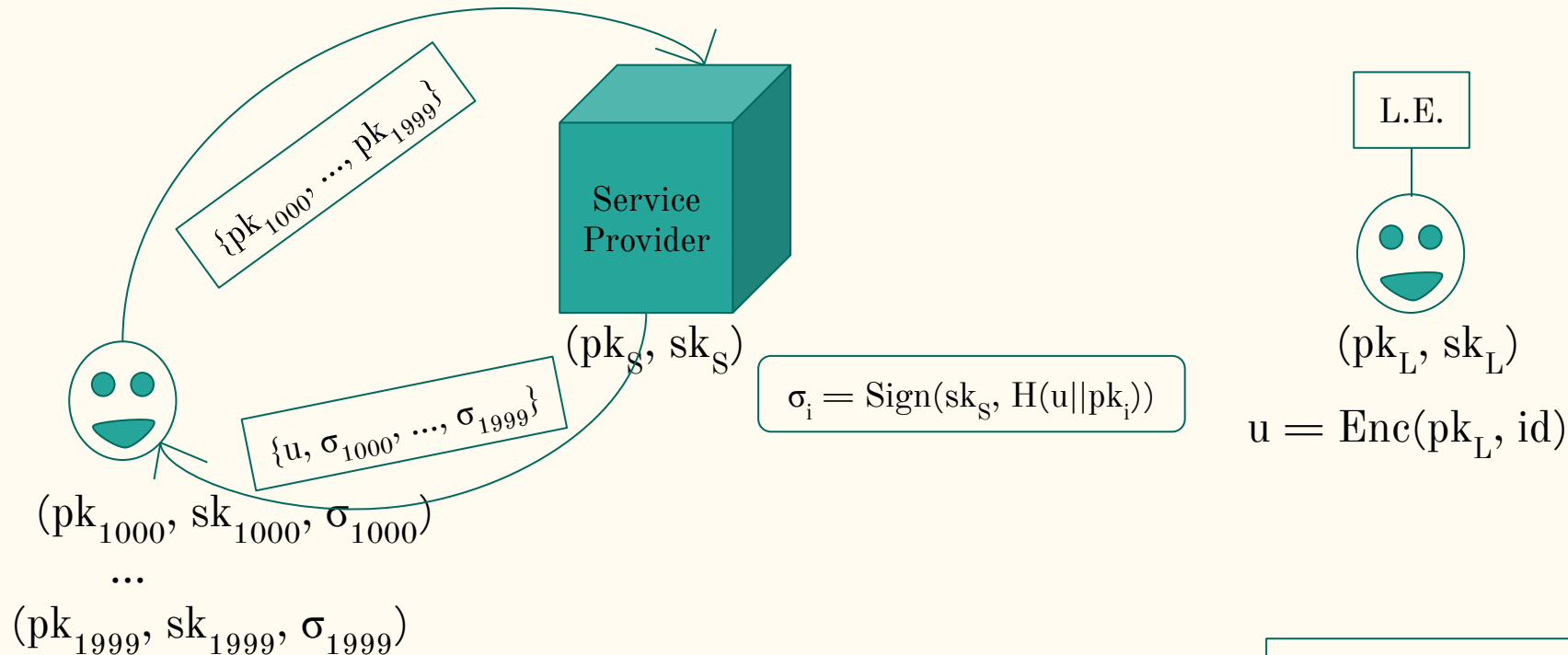
ATAVISM - a protocol sketch



ATAVISM - a protocol sketch

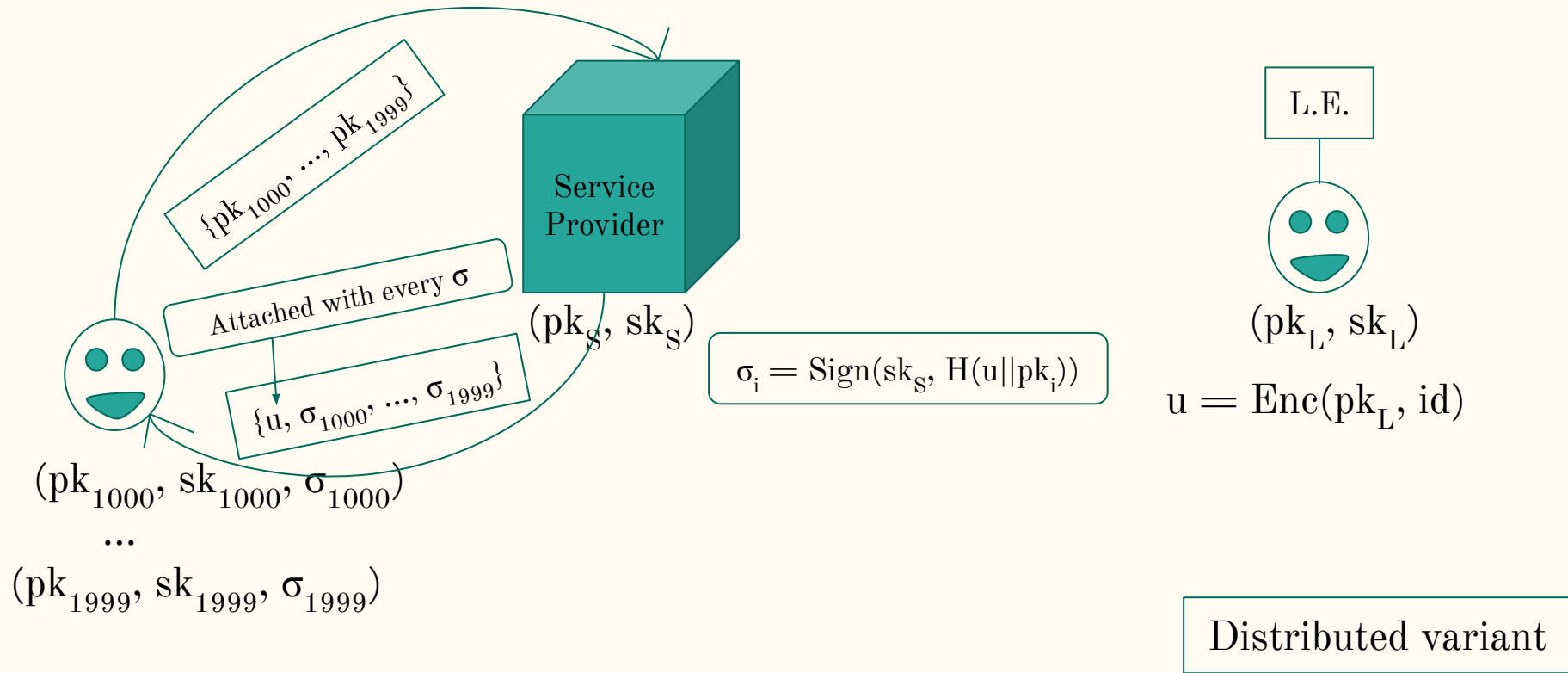


ATAVISM - a protocol sketch

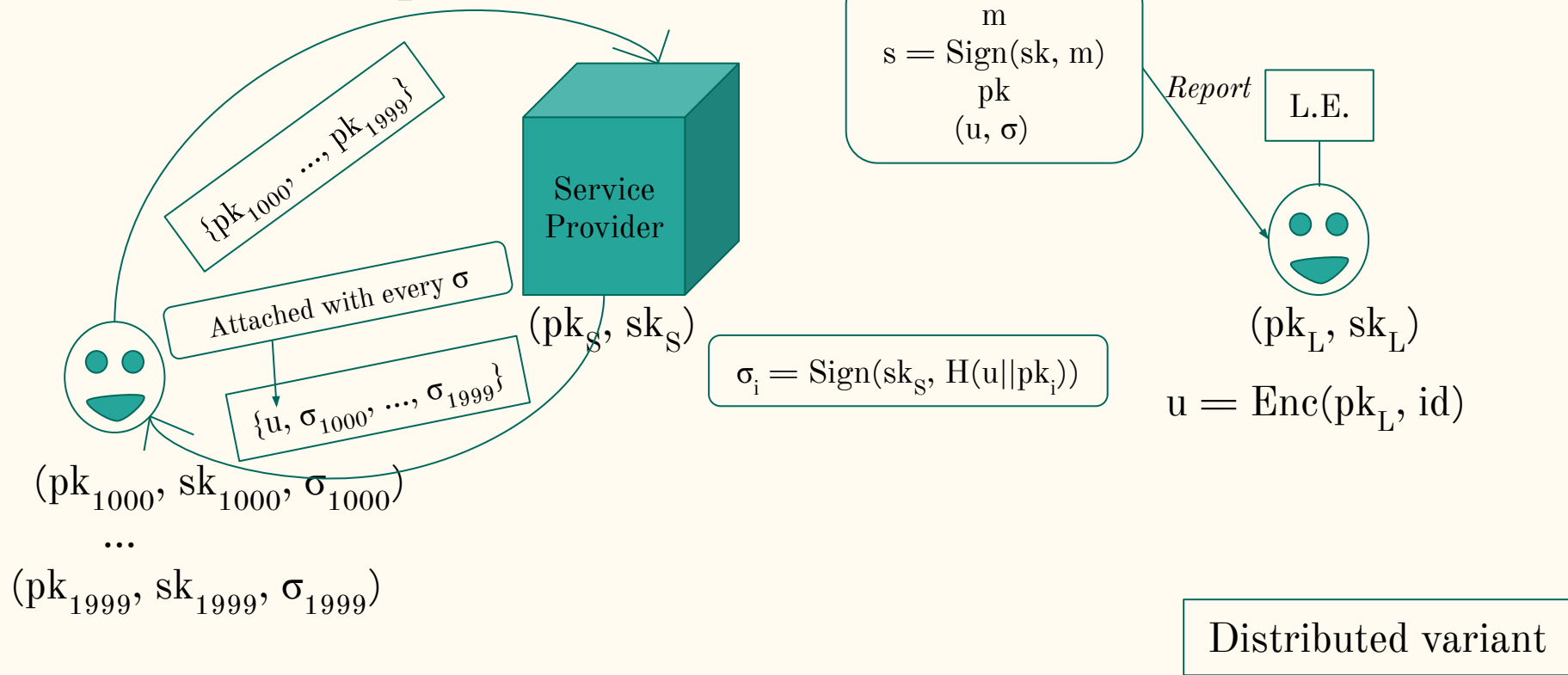


Distributed variant

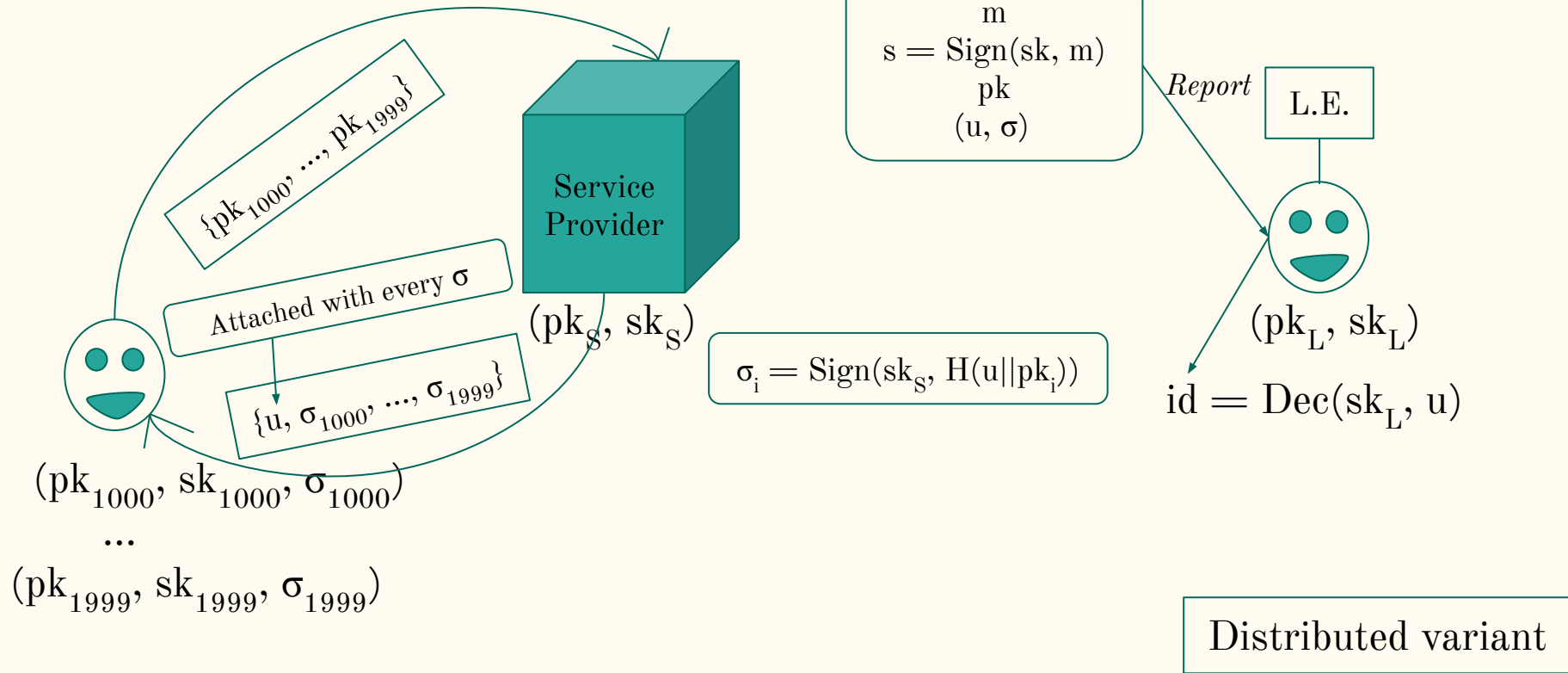
ATAVISM - a protocol sketch



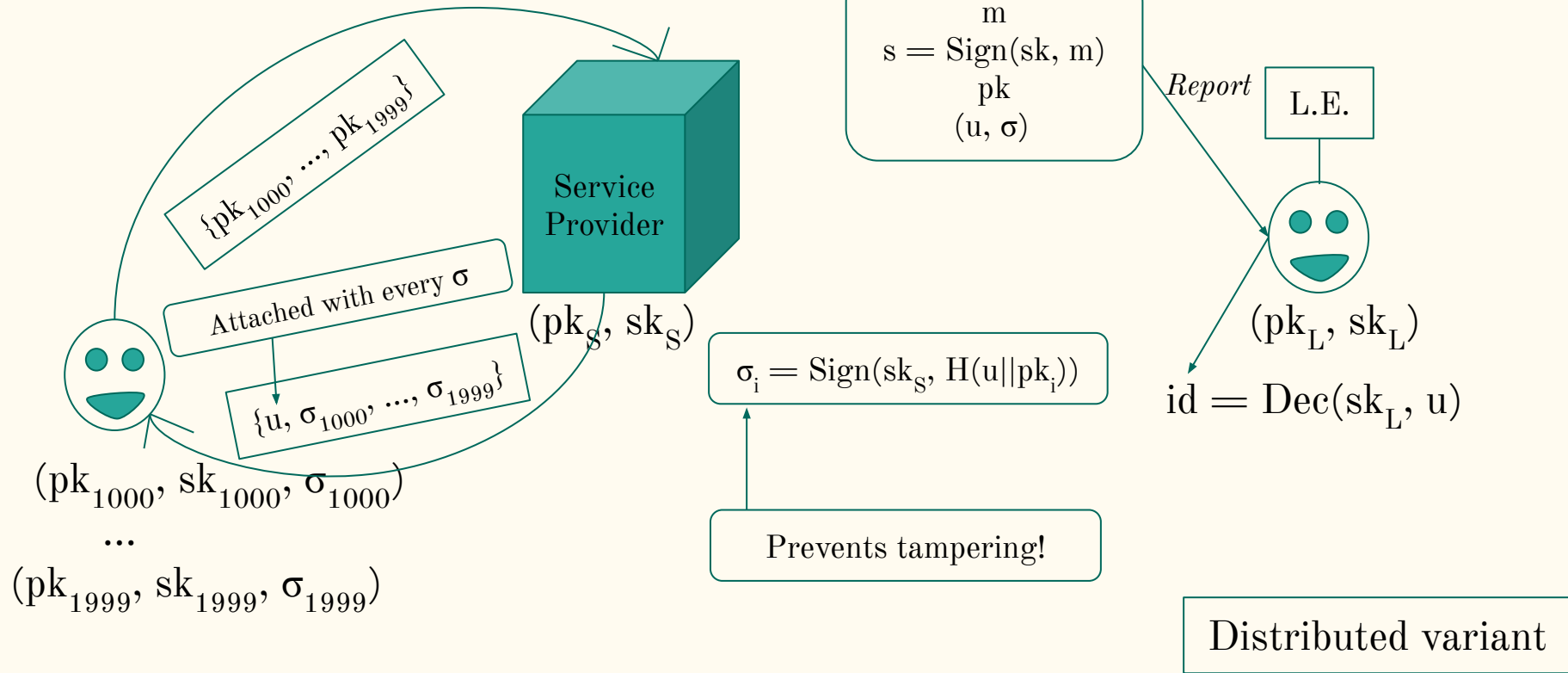
ATAVISM - a protocol sketch



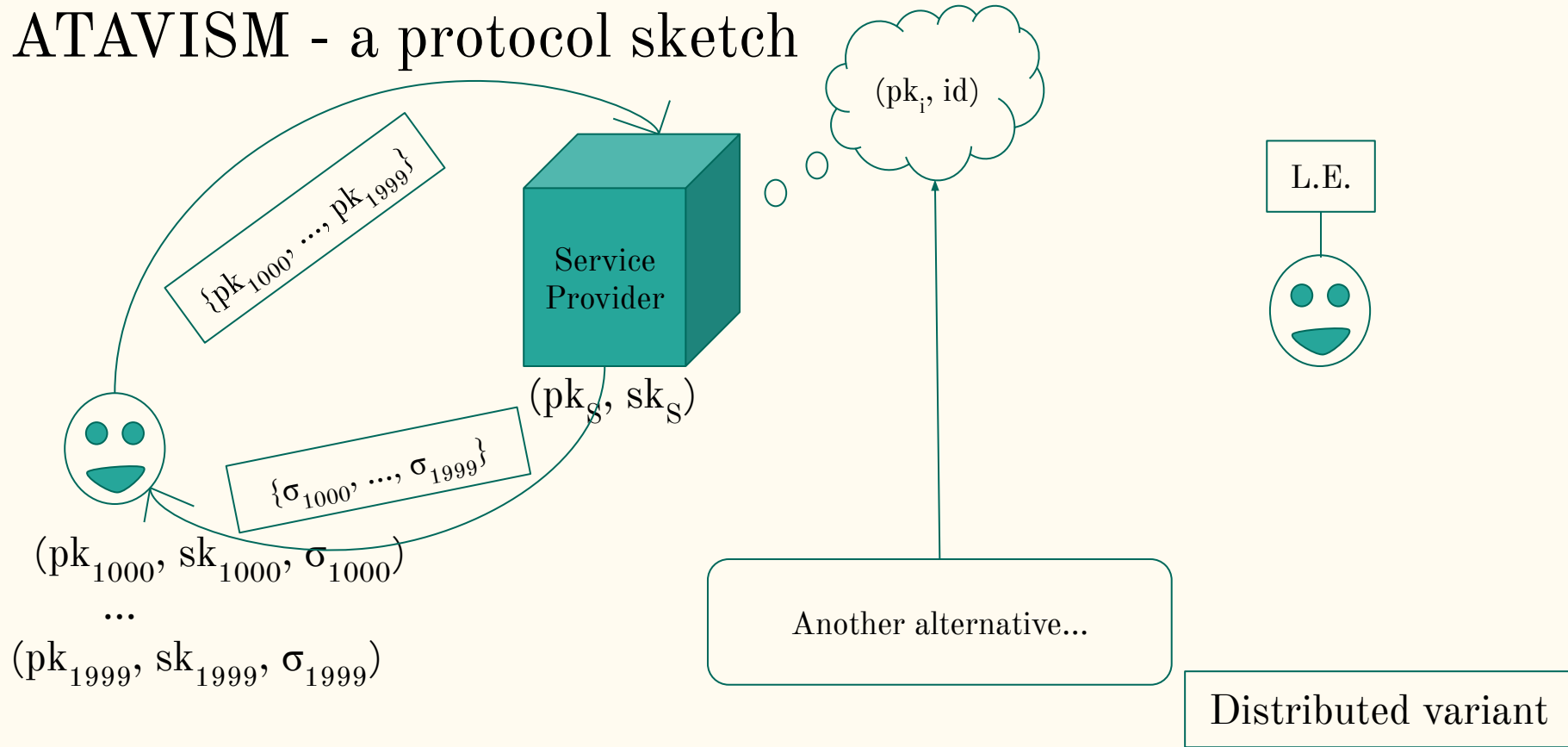
ATAVISM - a protocol sketch



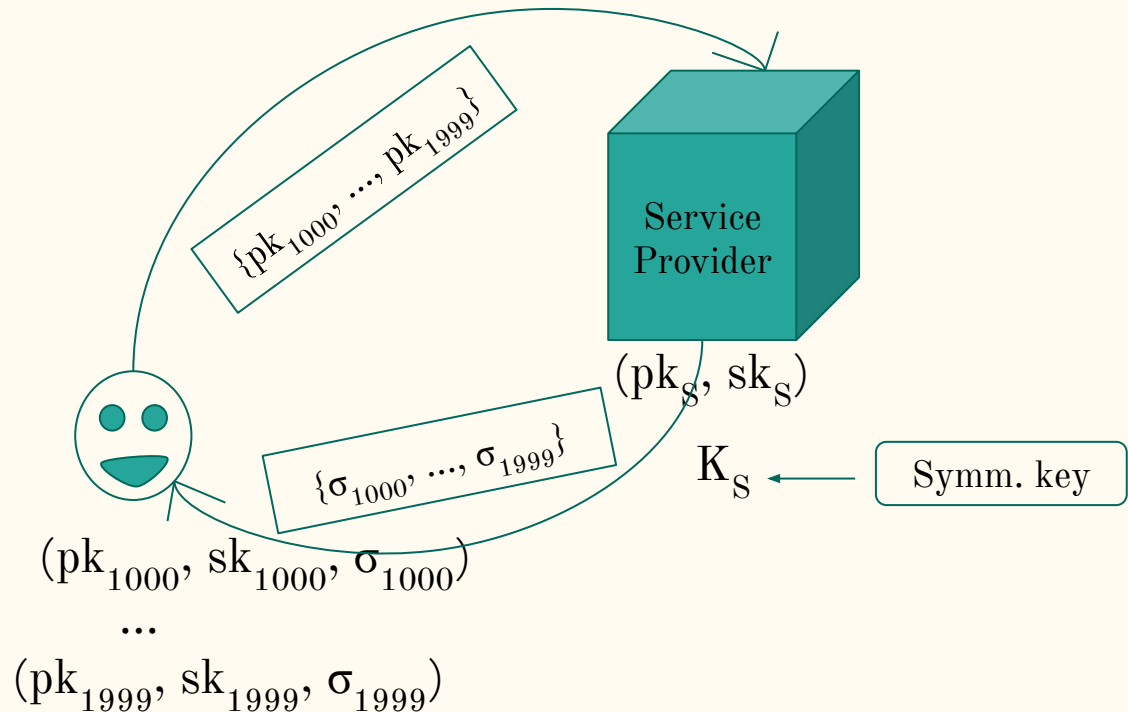
ATAVISM - a protocol sketch



ATAVISM - a protocol sketch

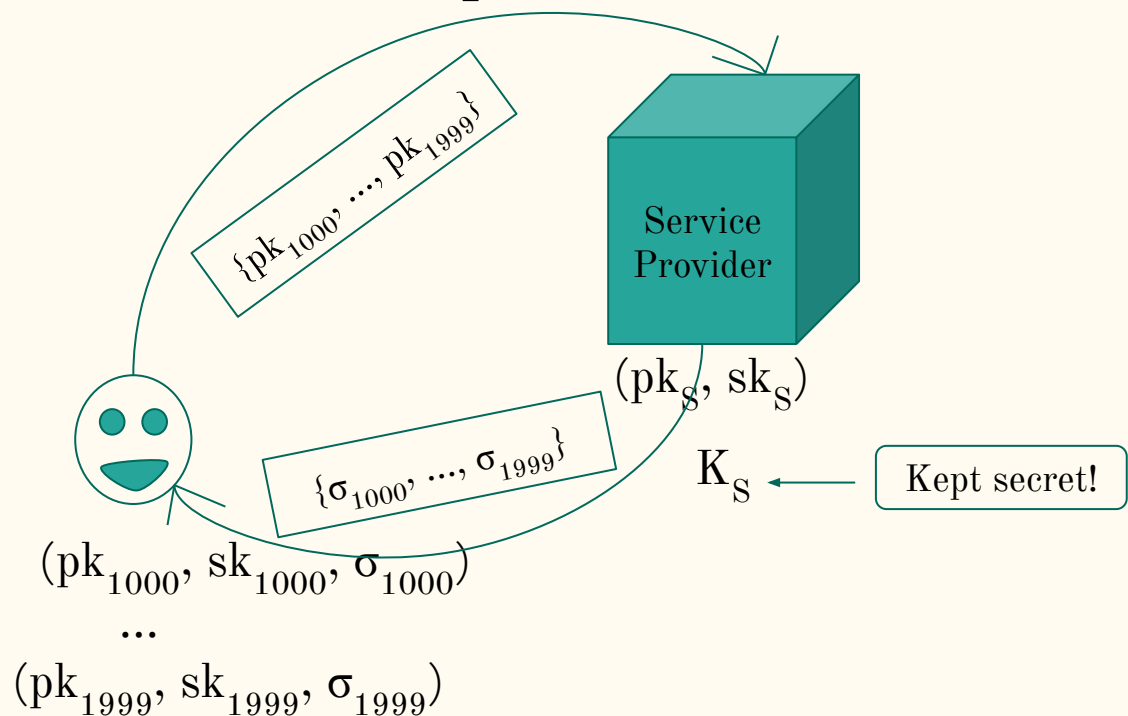


ATAVISM - a protocol sketch



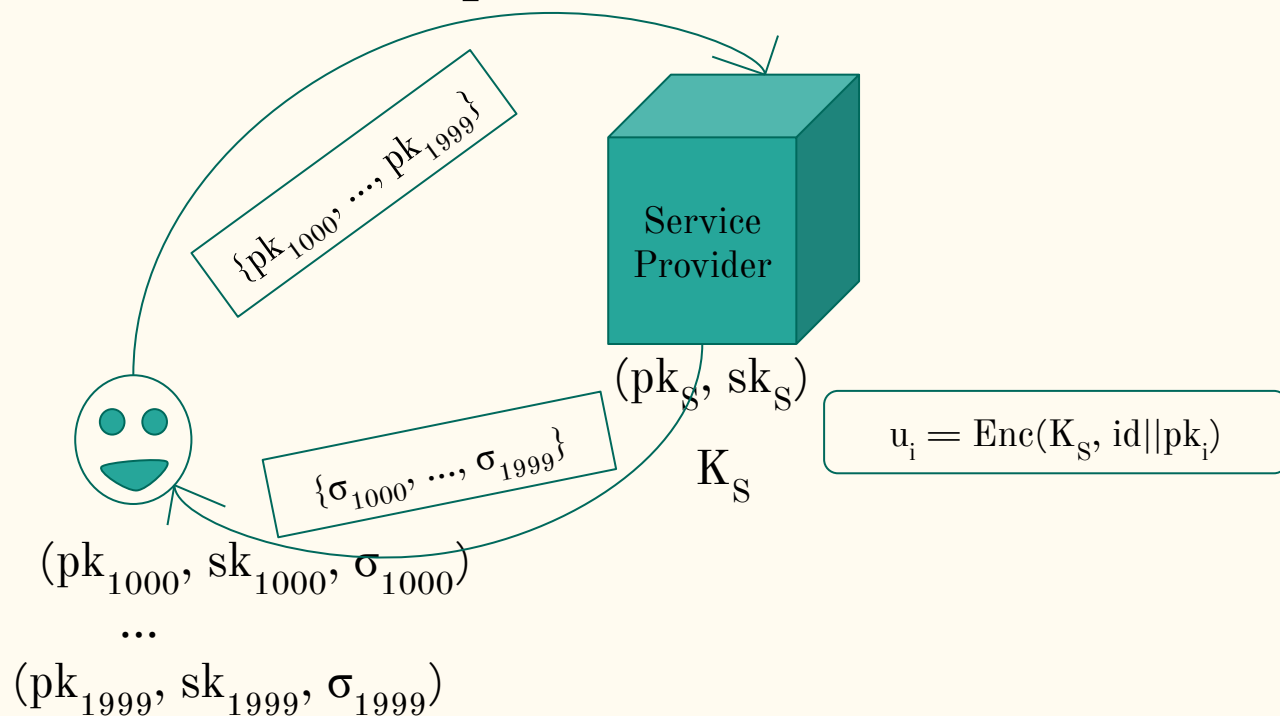
Distributed variant

ATAVISM - a protocol sketch



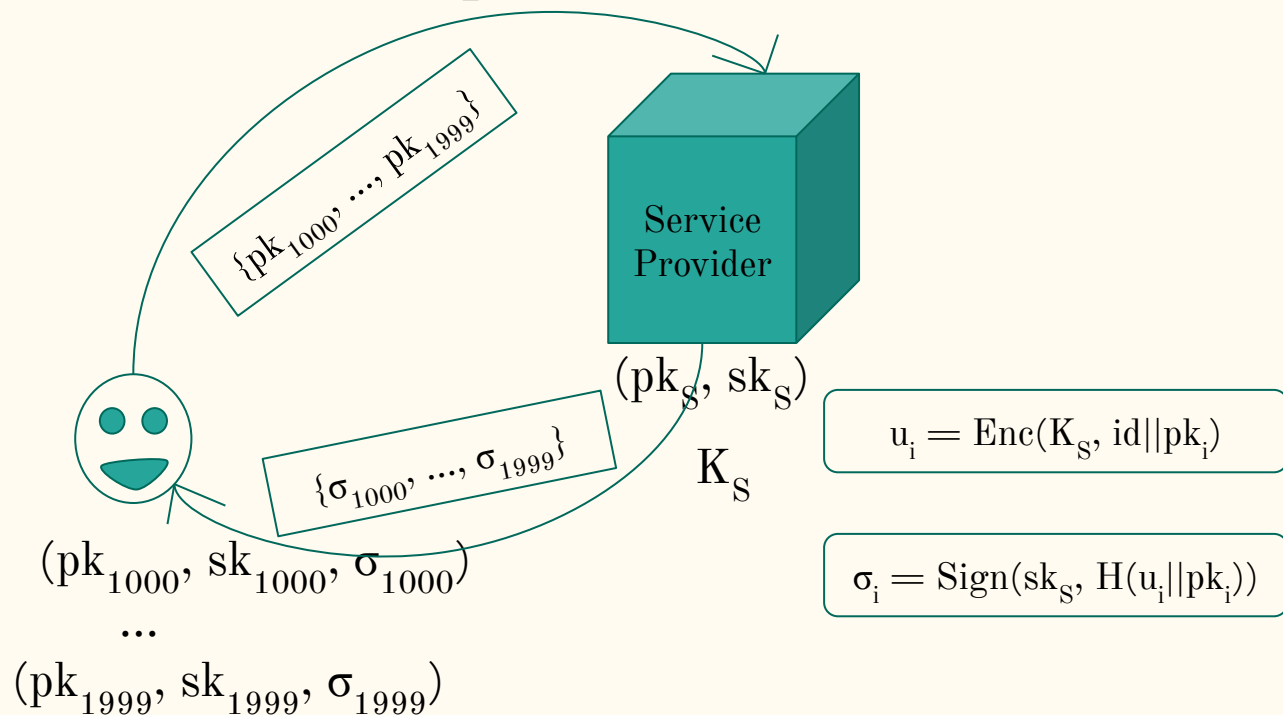
Distributed variant

ATAVISM - a protocol sketch



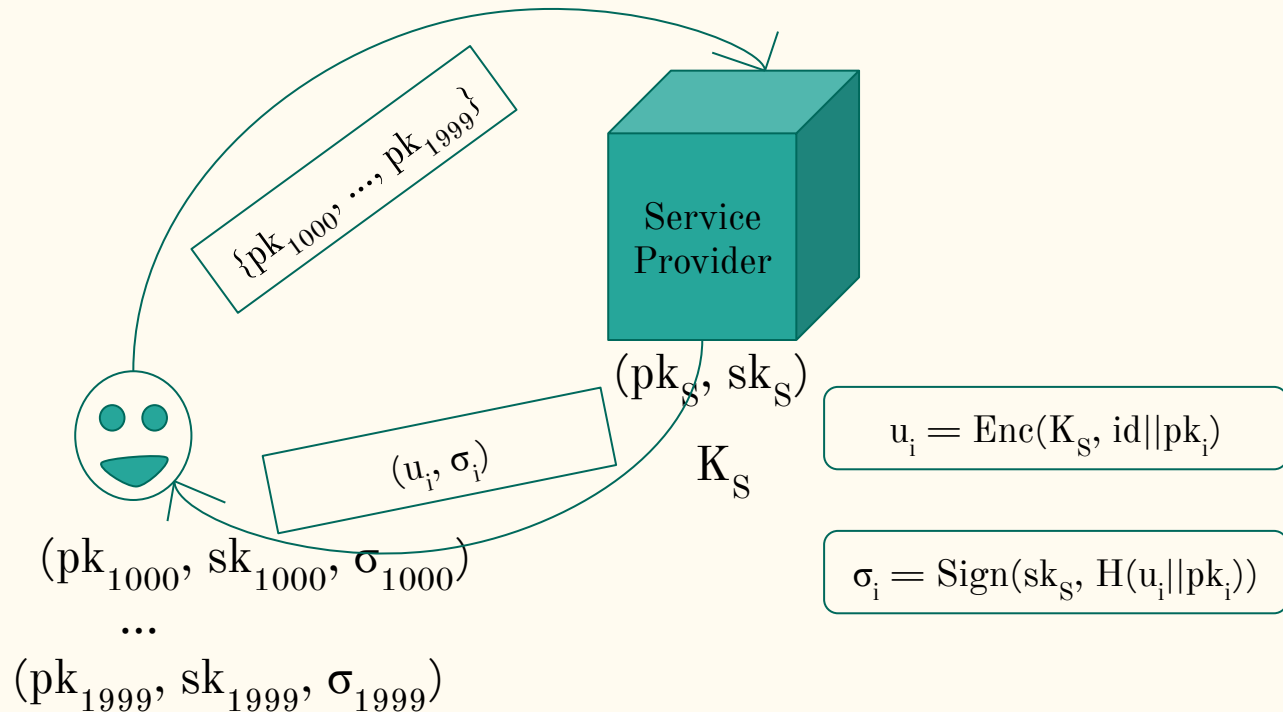
Distributed variant

ATAVISM - a protocol sketch



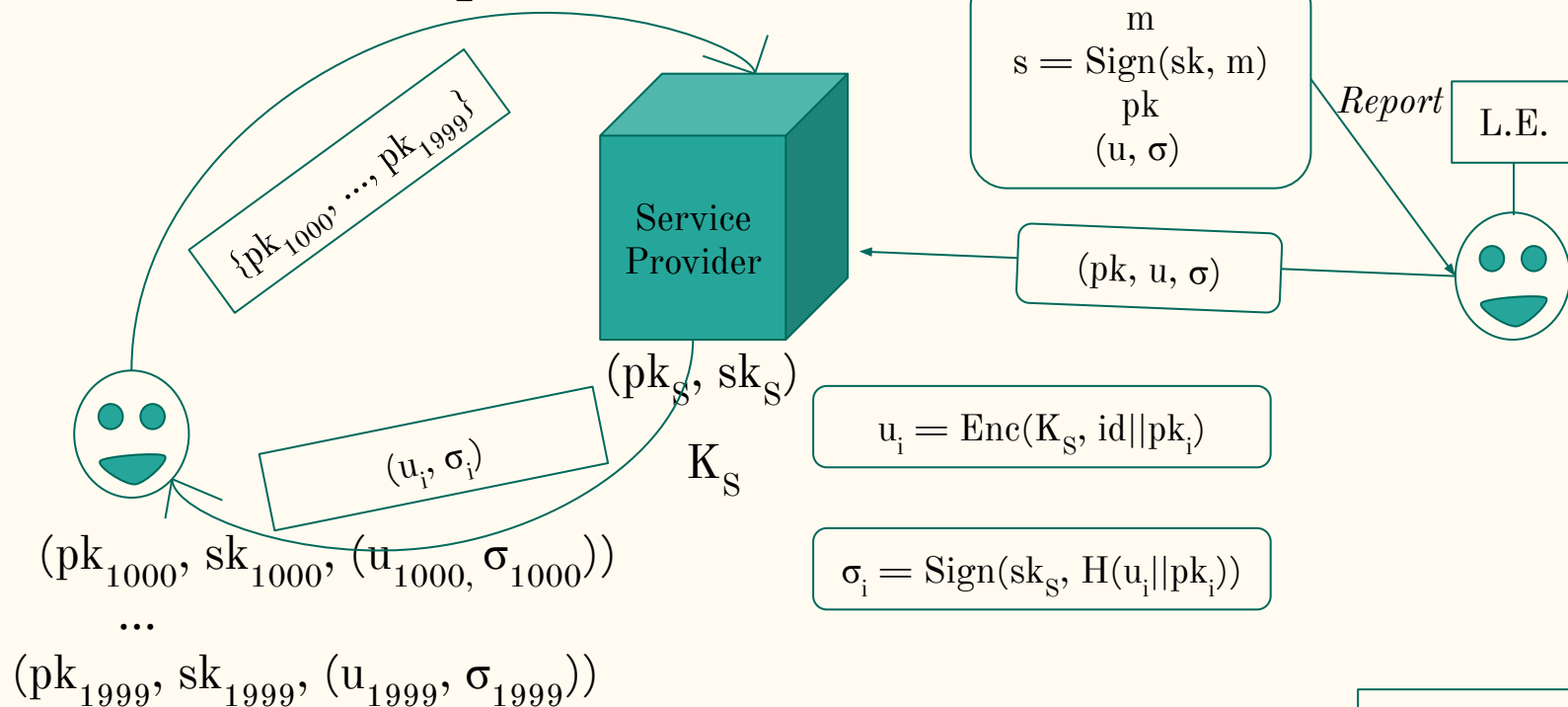
Distributed variant

ATAVISM - a protocol sketch



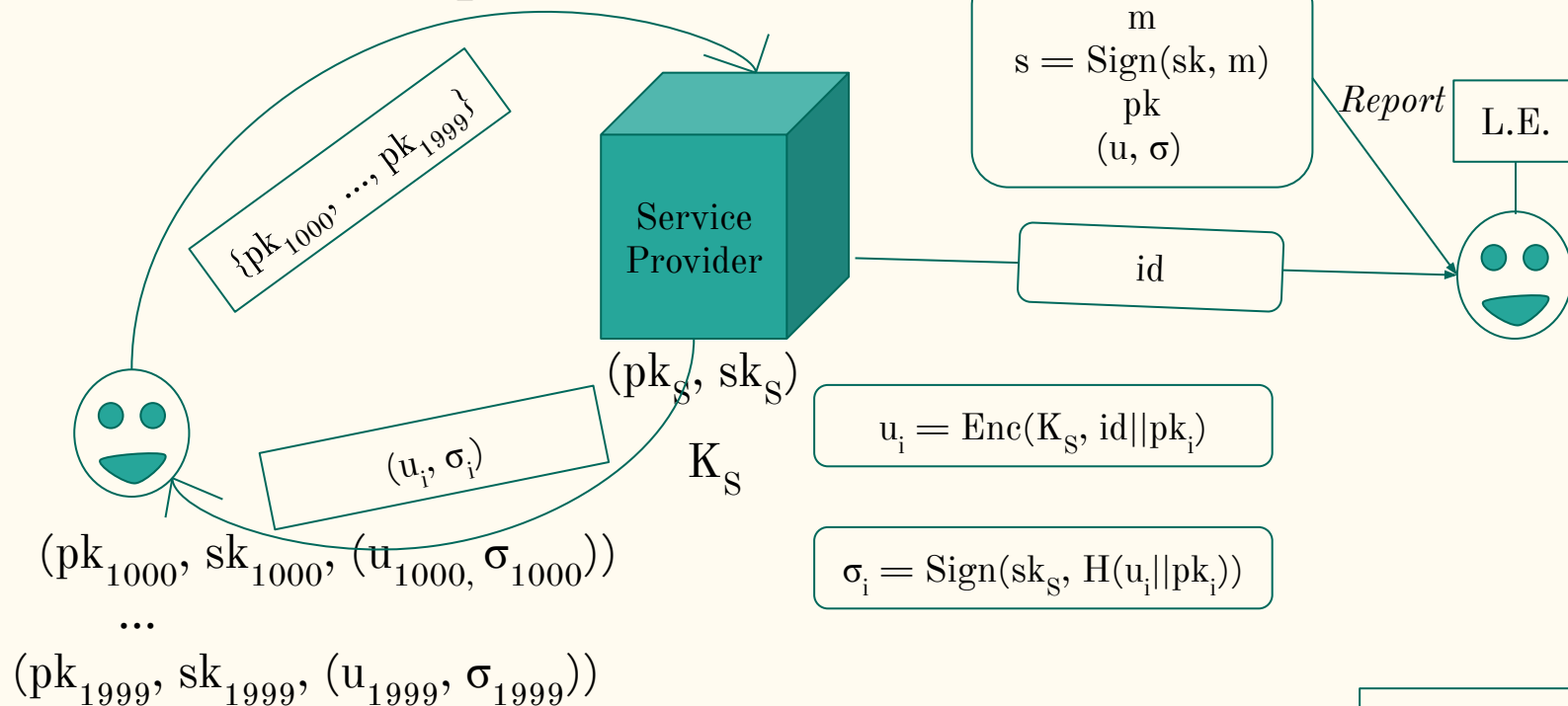
Distributed variant

ATAVISM - a protocol sketch



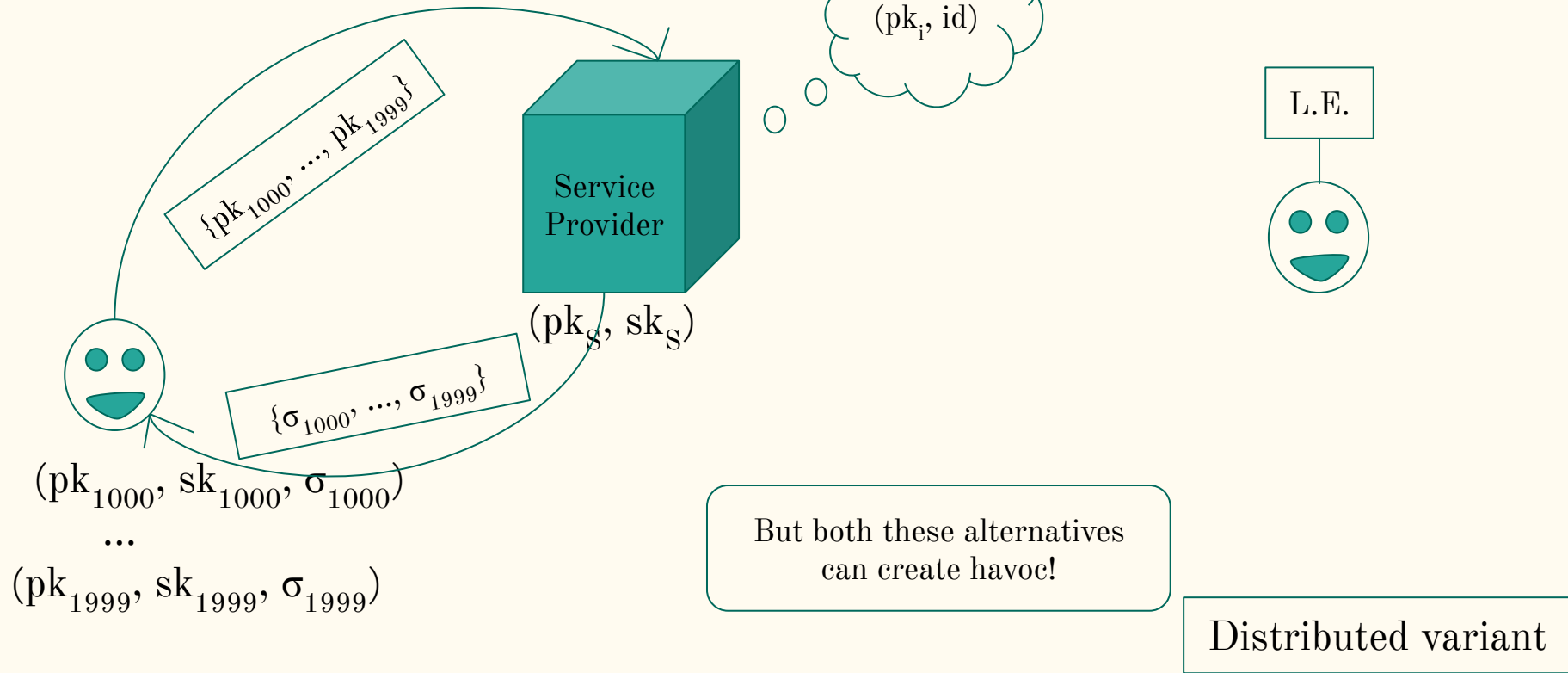
Distributed variant

ATAVISM - a protocol sketch

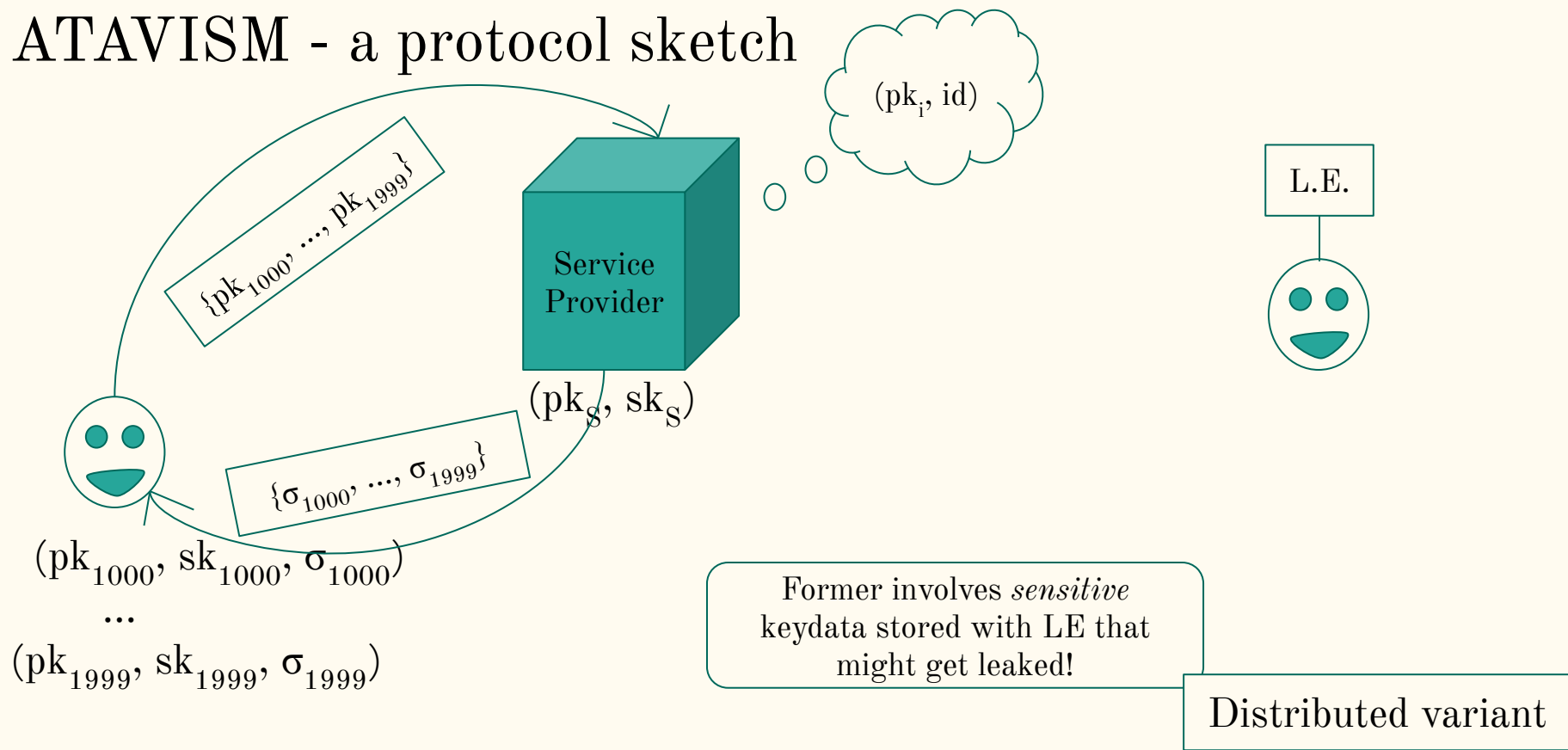


Distributed variant

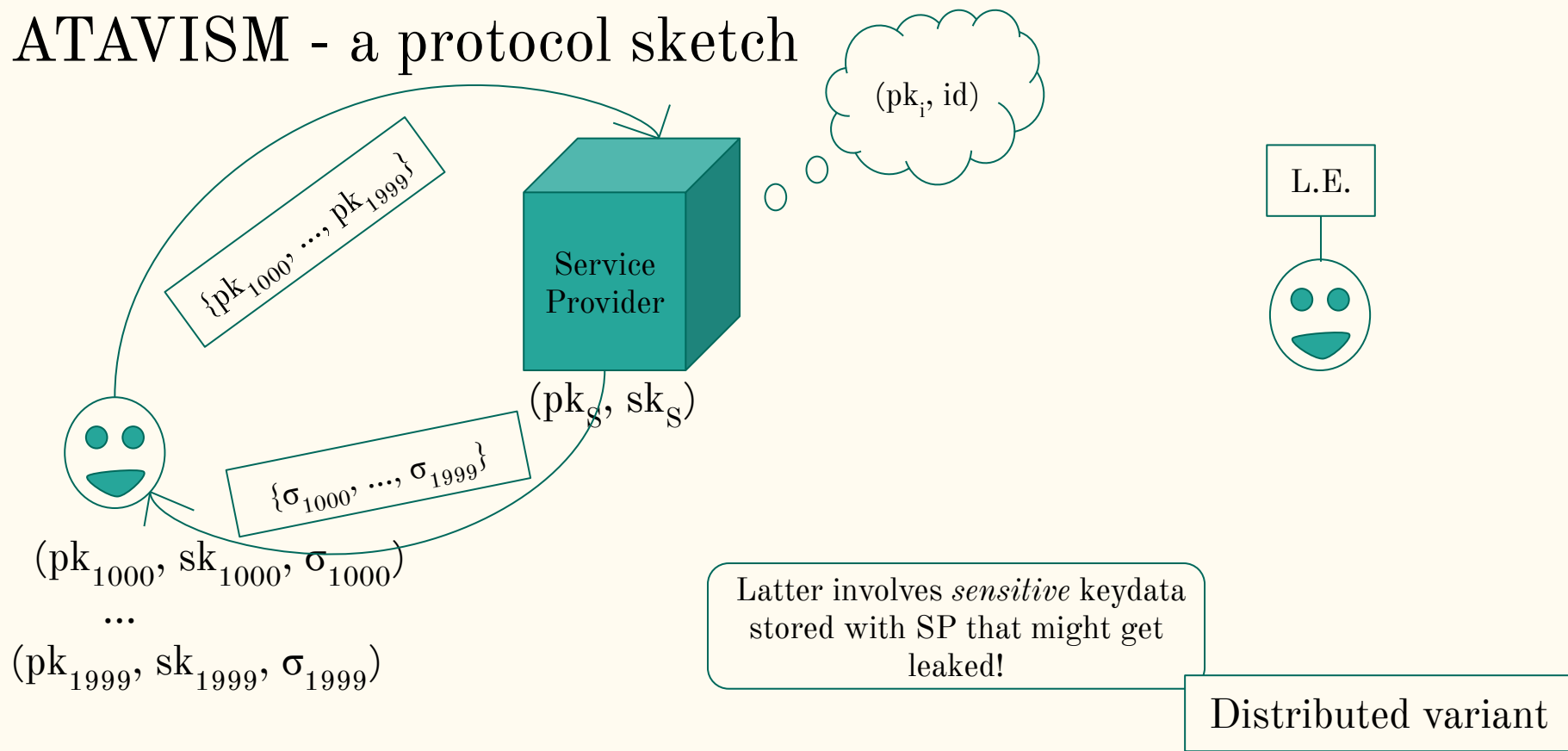
ATAVISM - a protocol sketch



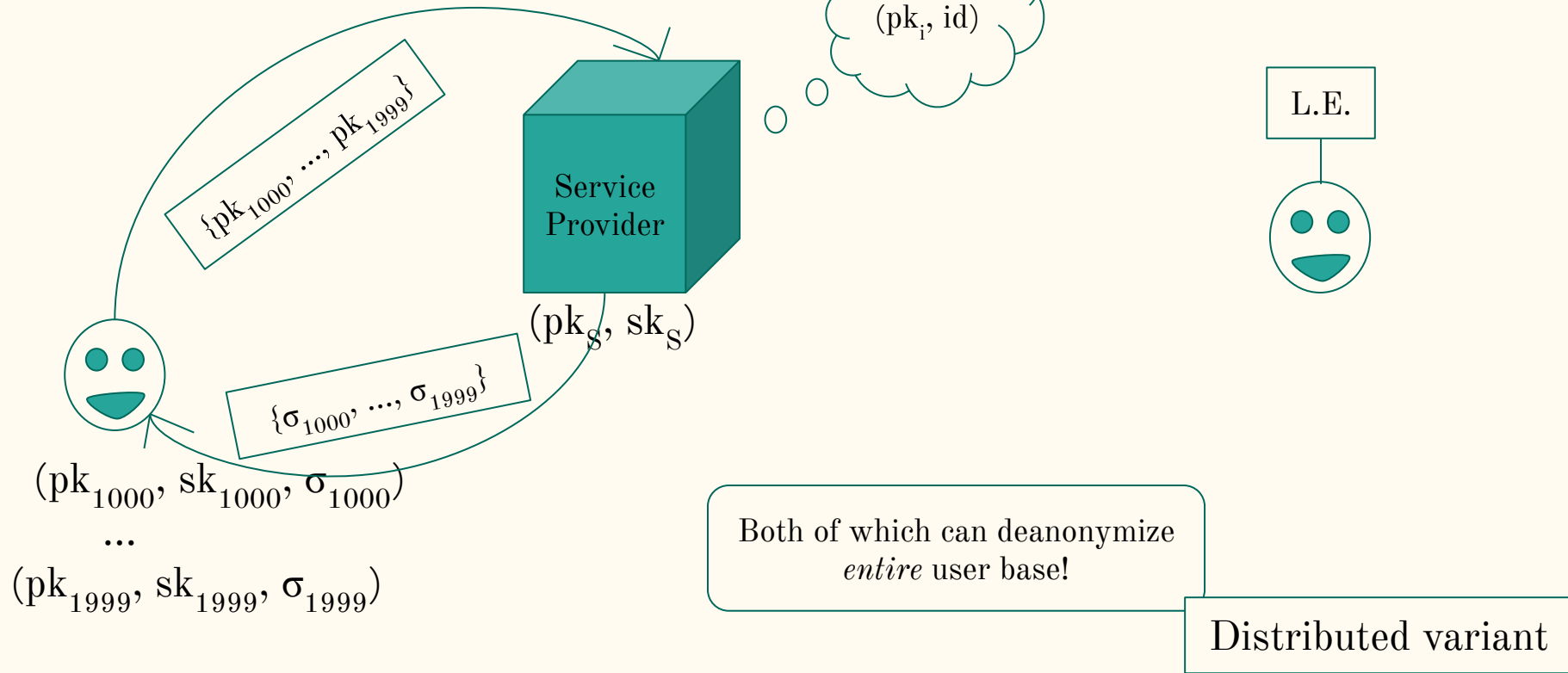
ATAVISM - a protocol sketch



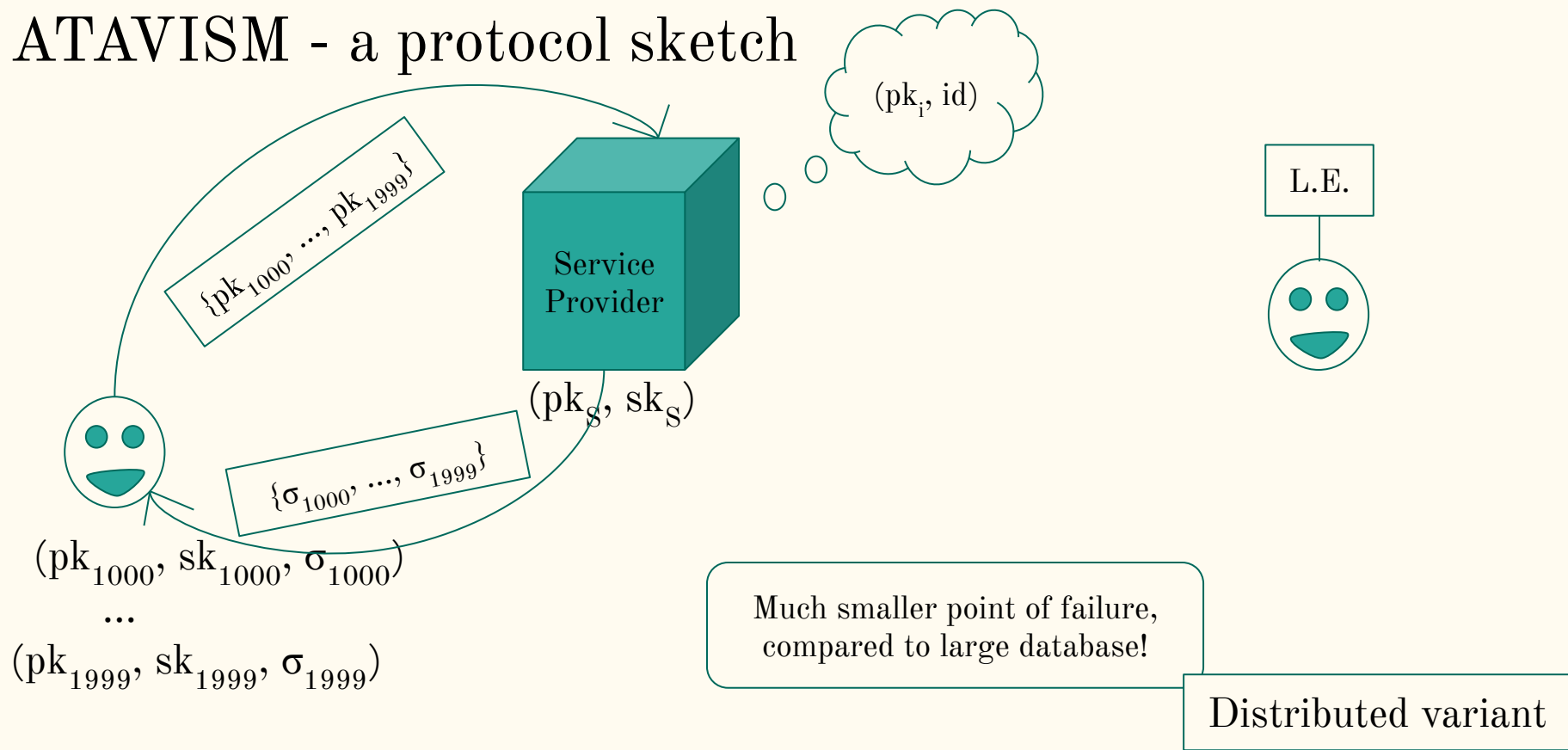
ATAVISM - a protocol sketch



ATAVISM - a protocol sketch



ATAVISM - a protocol sketch



Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- **Security Analysis - Overview**
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- Future Work and Conclusion

Security Analysis

- **Confidentiality**

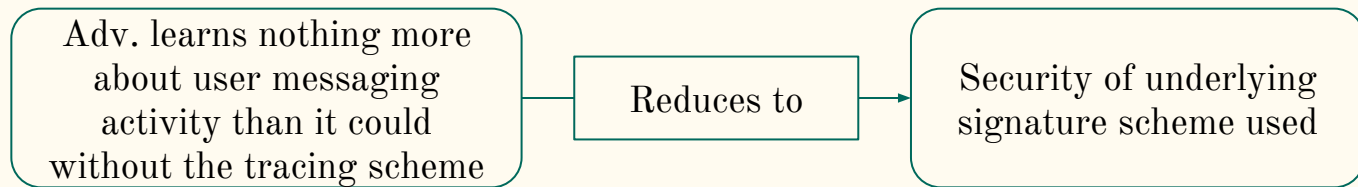
Security Analysis

- **Confidentiality**

Adv. learns nothing more
about user messaging
activity than it could
without the tracing scheme

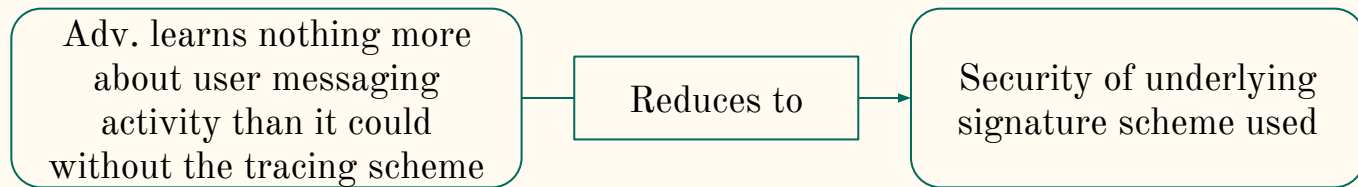
Security Analysis

- **Confidentiality**



Security Analysis

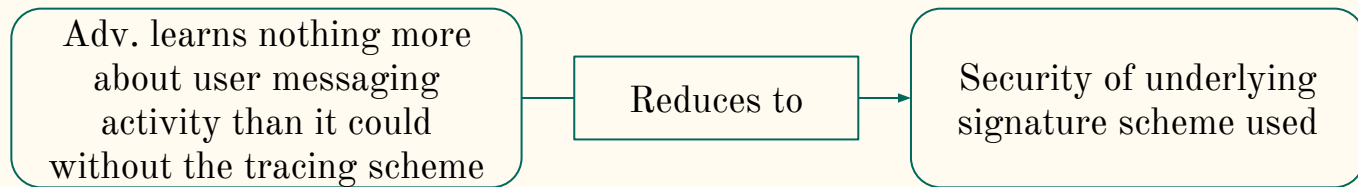
- **Confidentiality**



How?

Security Analysis

- **Confidentiality**



$$\begin{aligned} & m \\ s &= \text{Sign}(\text{sk}, m) \\ & \text{pk} \\ \sigma &= \text{Sign}(\text{sk}_s, \text{pk}) \end{aligned}$$

Security Analysis

- **Confidentiality**

Adv. learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

Security of underlying
signature scheme used

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

As in normal E2EE, no
extra power to adv.

Security Analysis

- **Confidentiality**

Adv. learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

Security of underlying
signature scheme used

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

Should only reveal that m is
authored by owner of sk corr. to pk

Security Analysis

- **Confidentiality**

Adv. learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

Security of underlying
signature scheme used

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

No one but SP knows who it
belongs to

Security Analysis

- **Confidentiality**

Adv. learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

Security of underlying
signature scheme used

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

And LE, if message is traced

Security Analysis

- **Confidentiality**

Adv. learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

Security of underlying
signature scheme used

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

Should only reveal that the sender
of m is registered with SP

Security Analysis

- Confidentiality

SP learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

E2EE

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

SP cannot read
anything either way

Security Analysis

- **Confidentiality**

SP learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

E2EE

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

Unless LE initiates trace
of reported message

Security Analysis

- **Confidentiality**

SP learns nothing more
about user messaging
activity than it could
without the tracing scheme

Reduces to

E2EE

m
 $s = \text{Sign}(sk, m)$
 pk
 $\sigma = \text{Sign}(sk_s, pk)$

Even then, m stays hidden
from SP!

Security Analysis

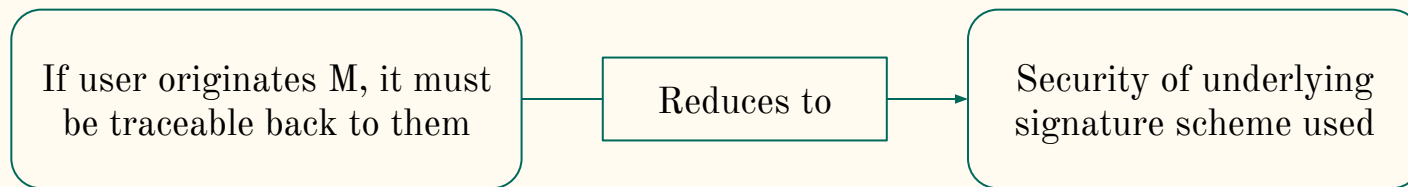
- **Accountability**

If user originates M, it must
be traceable back to them

$$\begin{array}{l} m \\ s = \text{Sign}(\text{sk}, m) \\ pk \\ \sigma = \text{Sign}(\text{sk}_s, pk) \end{array} M$$

Security Analysis

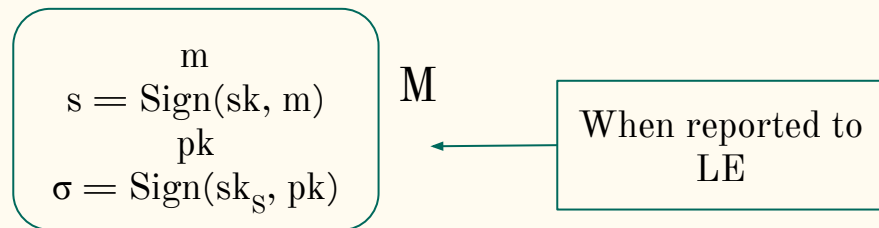
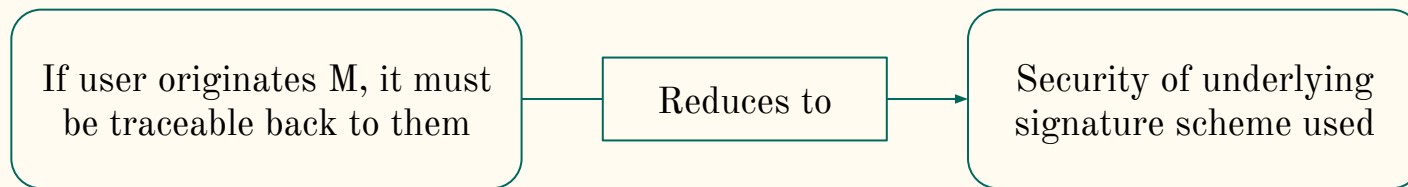
- **Accountability**



$$\begin{array}{l} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

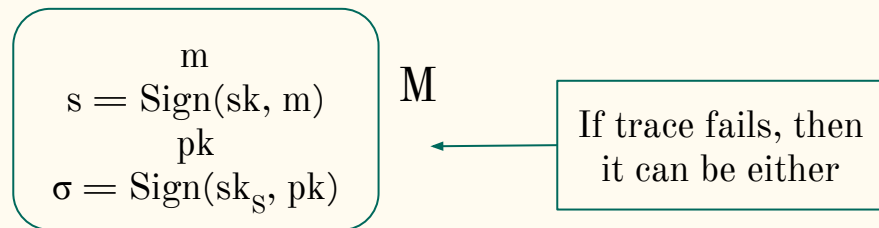
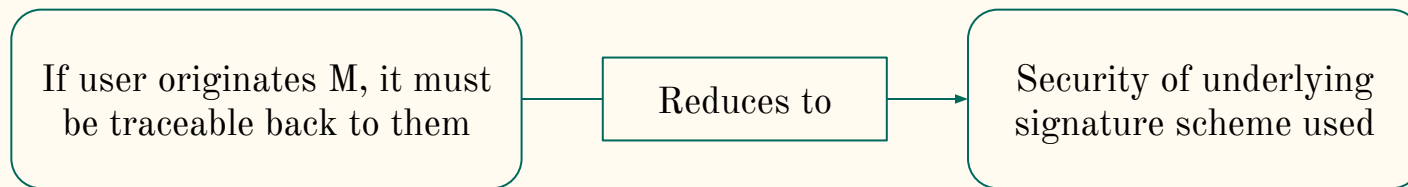
Security Analysis

- **Accountability**



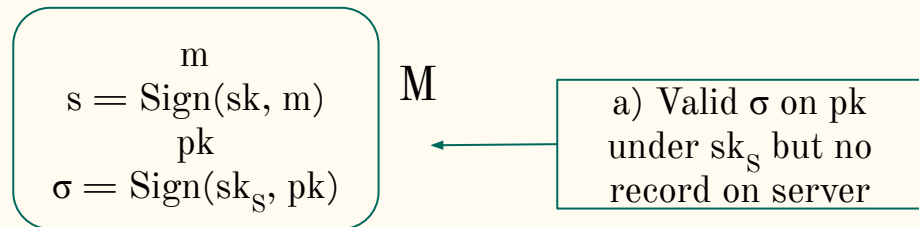
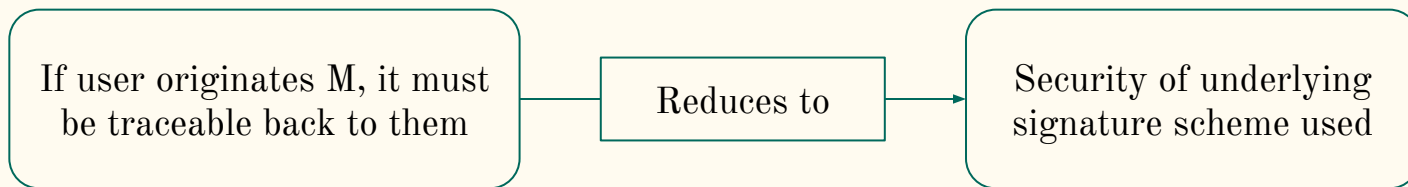
Security Analysis

- **Accountability**



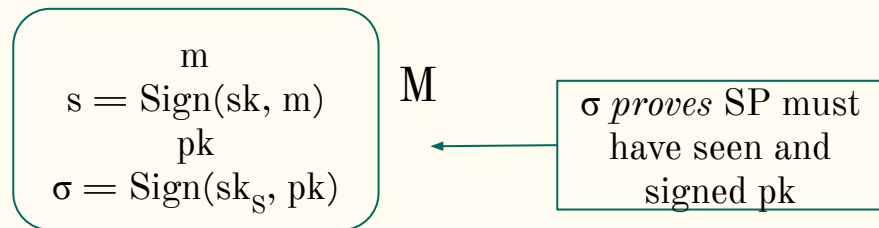
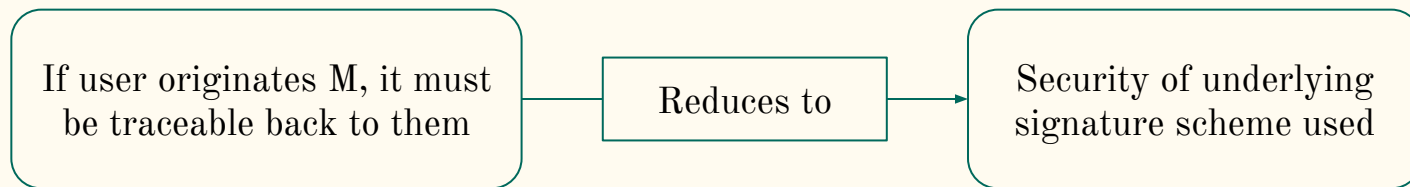
Security Analysis

- Accountability



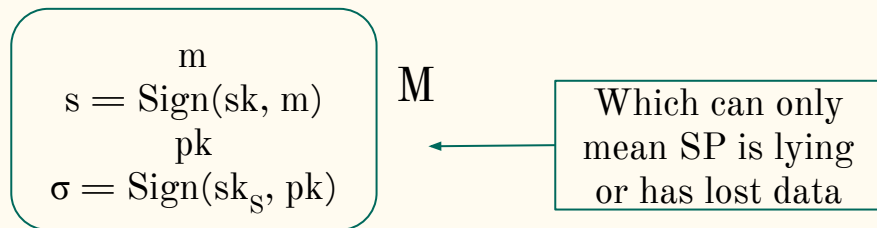
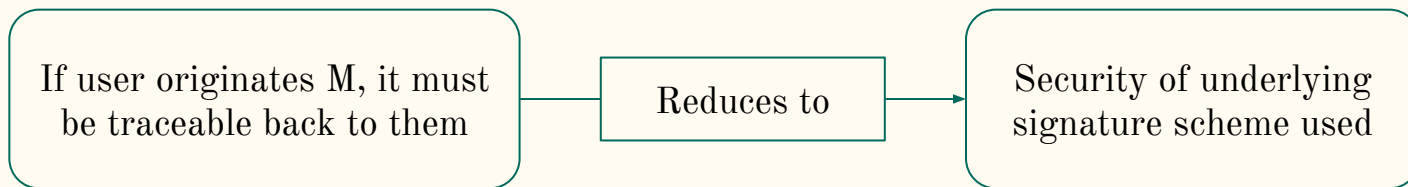
Security Analysis

- **Accountability**



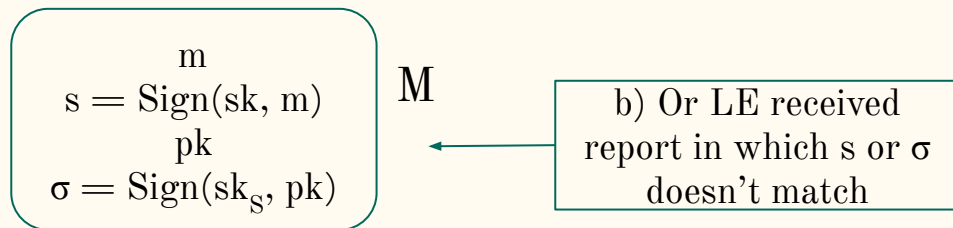
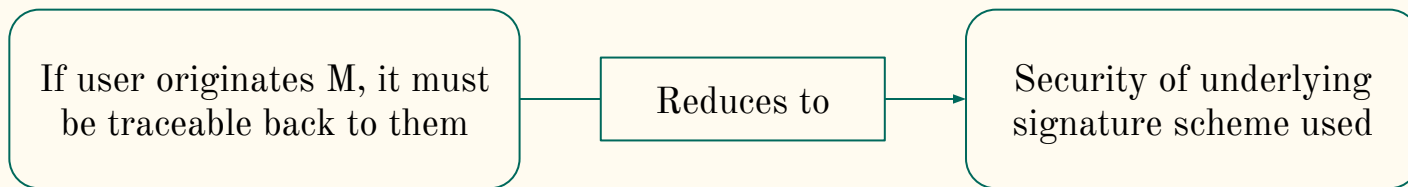
Security Analysis

- **Accountability**



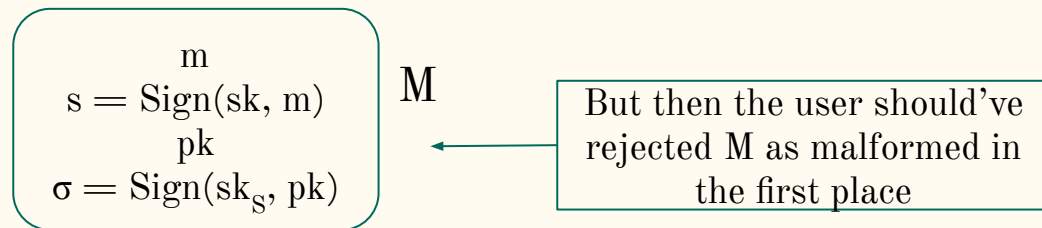
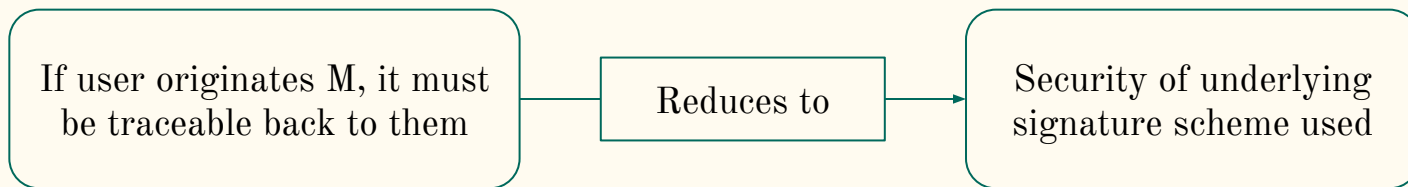
Security Analysis

- **Accountability**



Security Analysis

- **Accountability**



Security Analysis

- **Unforgeability**

Adv. cannot implicate user
in sending a message they
did not actually send

$$\begin{array}{l} m \\ s = \text{Sign}(\text{sk}, m) \\ pk \\ \sigma = \text{Sign}(\text{sk}_s, pk) \end{array} \quad M$$

Security Analysis

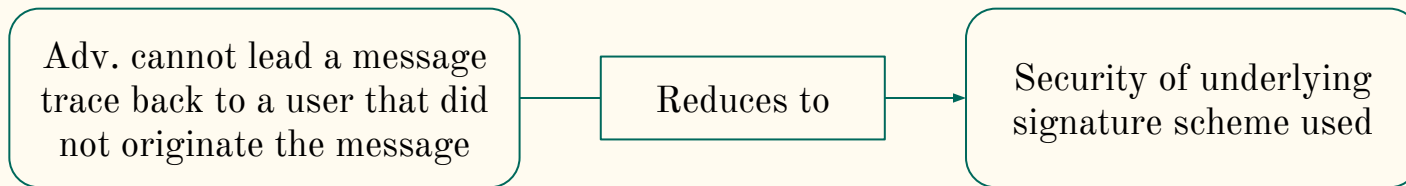
- **Unforgeability**

Adv. cannot lead a message
trace back to a user that did
not originate the message

$$\begin{array}{l} m \\ s = \text{Sign}(\text{sk}, m) \\ pk \\ \sigma = \text{Sign}(\text{sk}_s, pk) \end{array} \quad M$$

Security Analysis

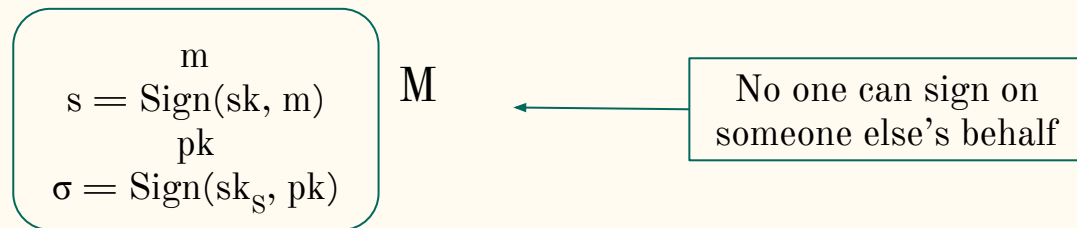
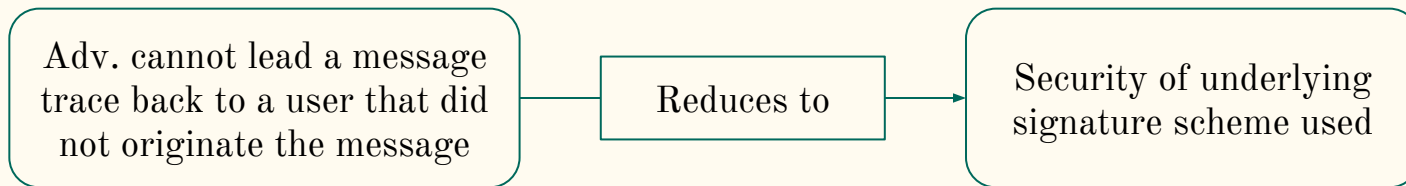
- **Unforgeability**



$$\begin{array}{l} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

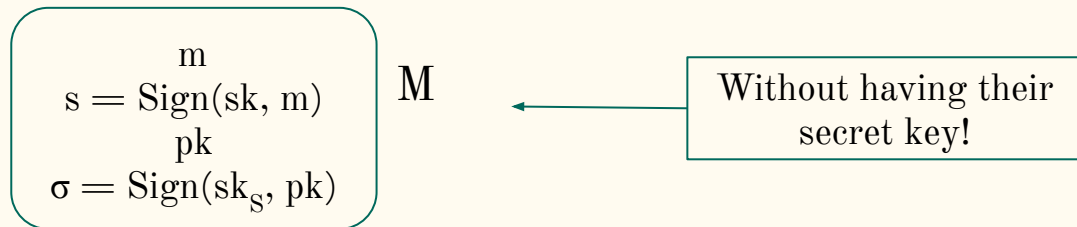
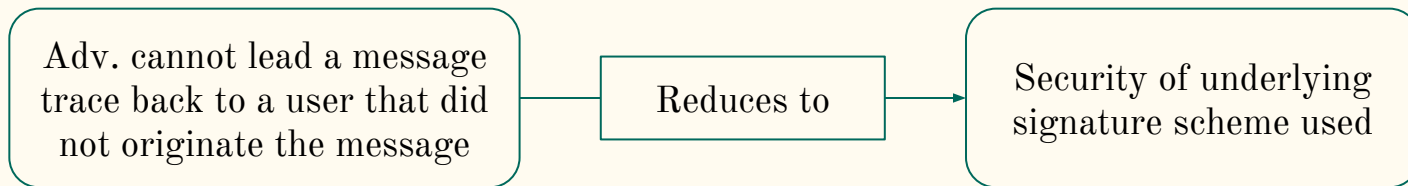
Security Analysis

- **Unforgeability**



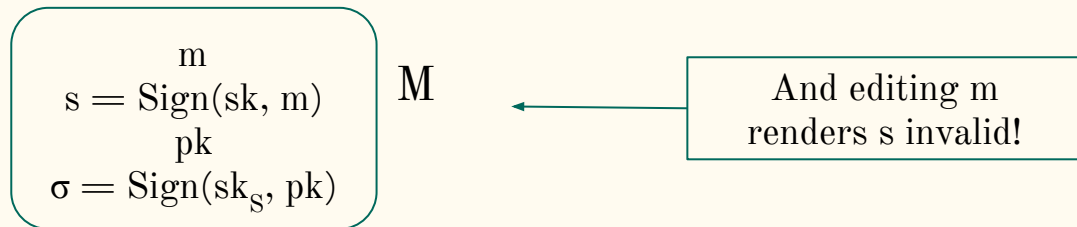
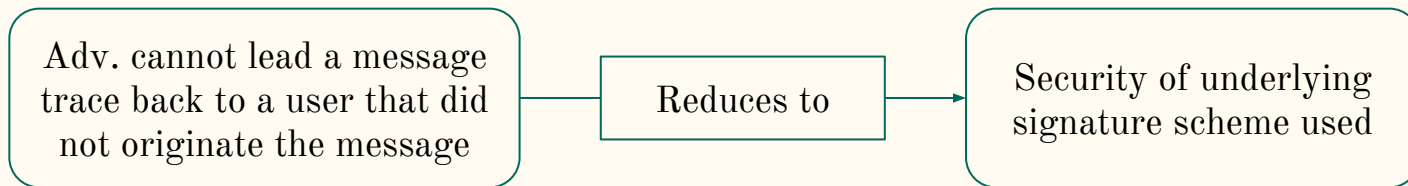
Security Analysis

- **Unforgeability**



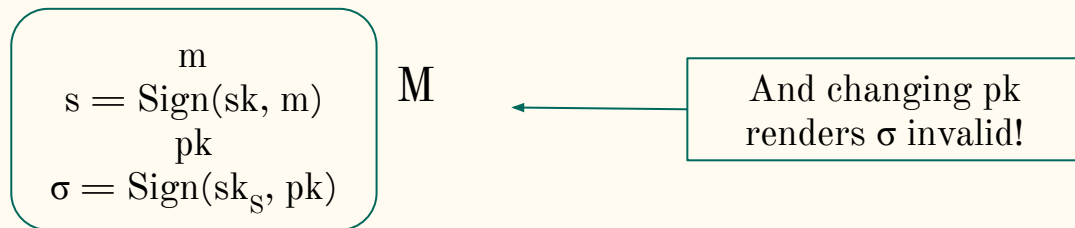
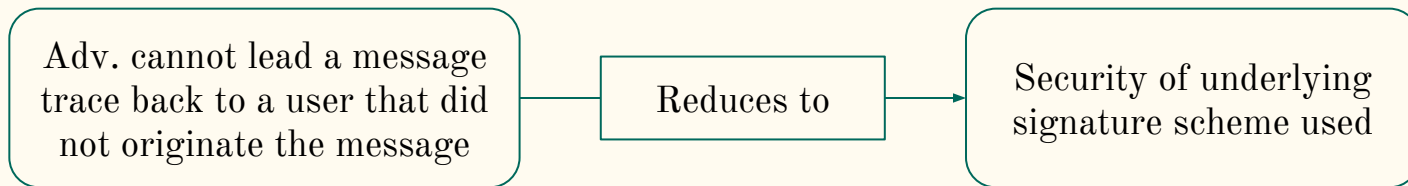
Security Analysis

- **Unforgeability**



Security Analysis

- **Unforgeability**



Security Analysis

- Unforgeability

Adv. cannot lead a message
trace back to a user that did
not originate the message

Reduces to

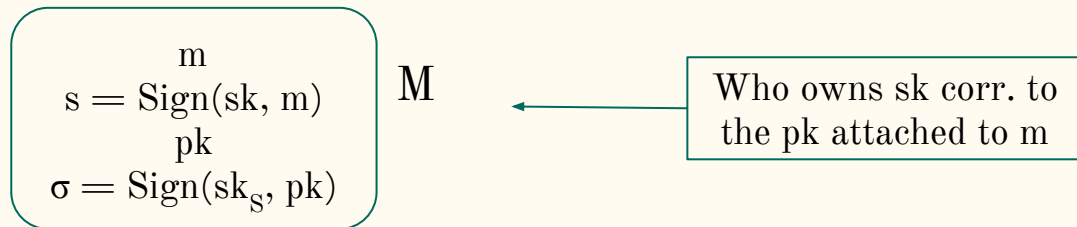
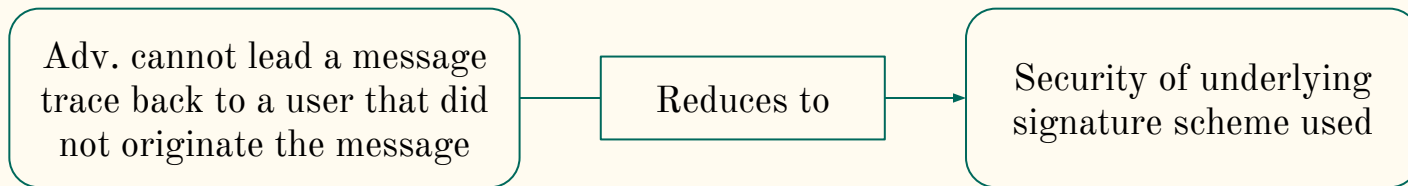
Security of underlying
signature scheme used

$$\begin{array}{c} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

Both sigs tie m to the
true originator of m

Security Analysis

- Unforgeability



Security Analysis

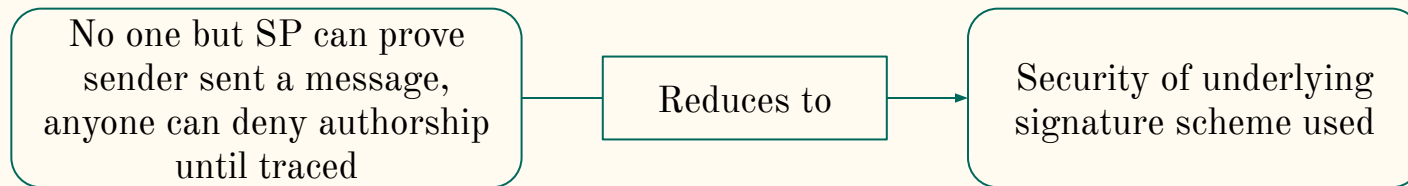
- **Deniability**

No one but SP can prove
sender sent a message,
anyone can deny authorship
until traced

$$\begin{array}{l} m \\ s = \text{Sign}(\text{sk}, m) \\ pk \\ \sigma = \text{Sign}(\text{sk}_s, pk) \end{array} \quad M$$

Security Analysis

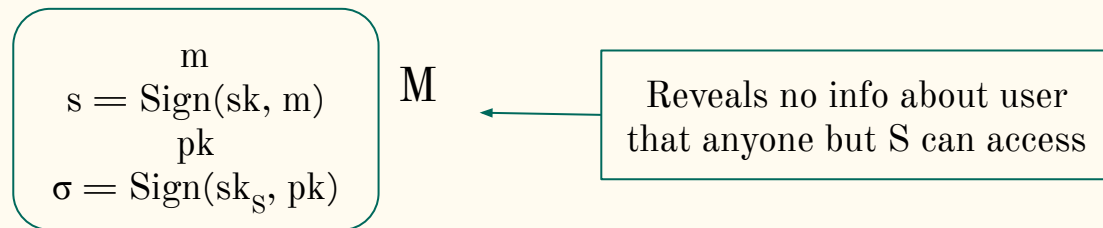
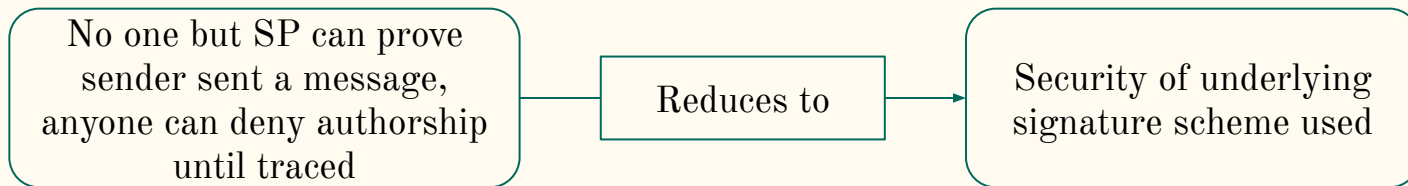
- **Deniability**



$$\begin{array}{l} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

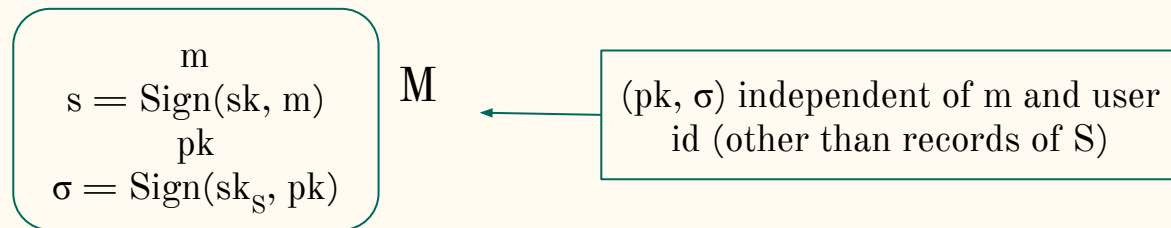
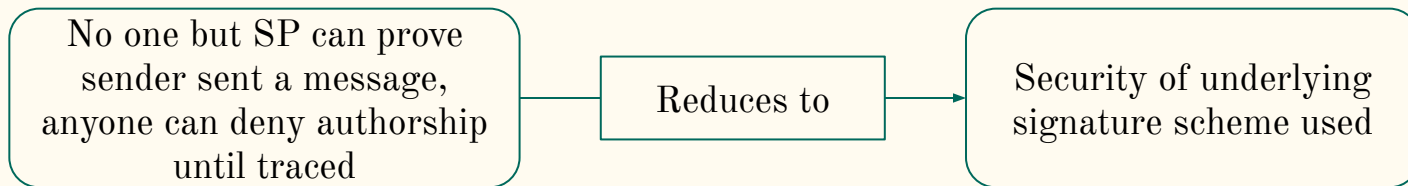
Security Analysis

- Deniability



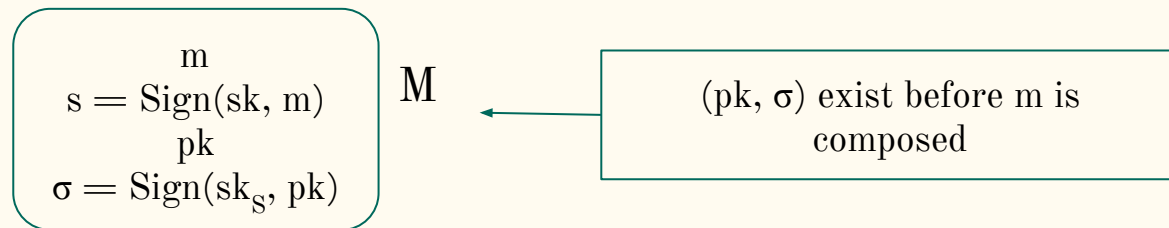
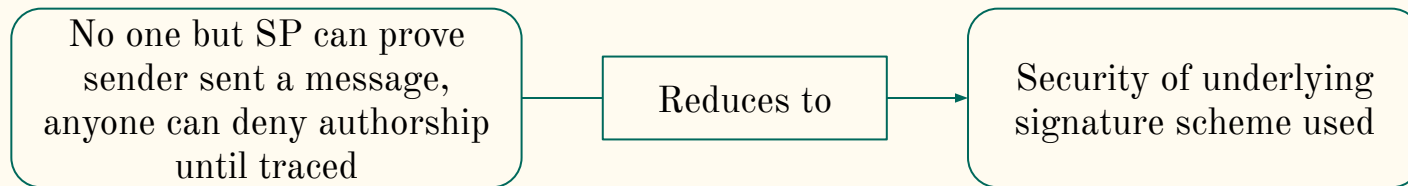
Security Analysis

- Deniability



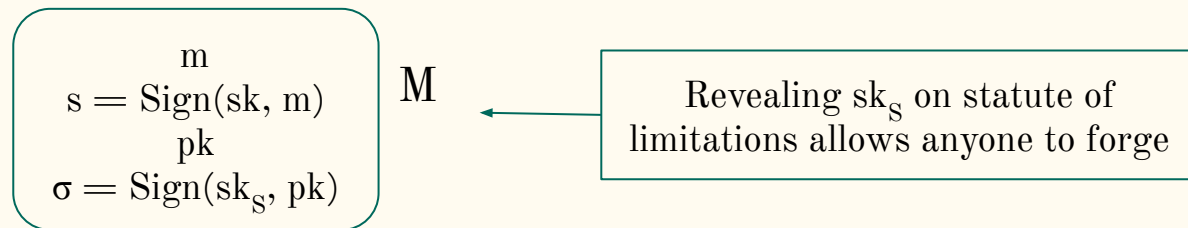
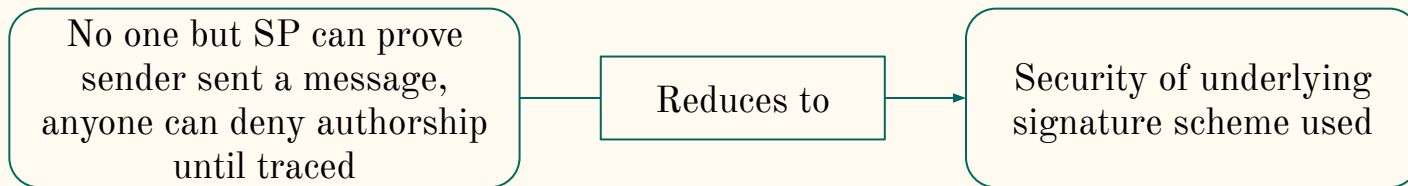
Security Analysis

- Deniability



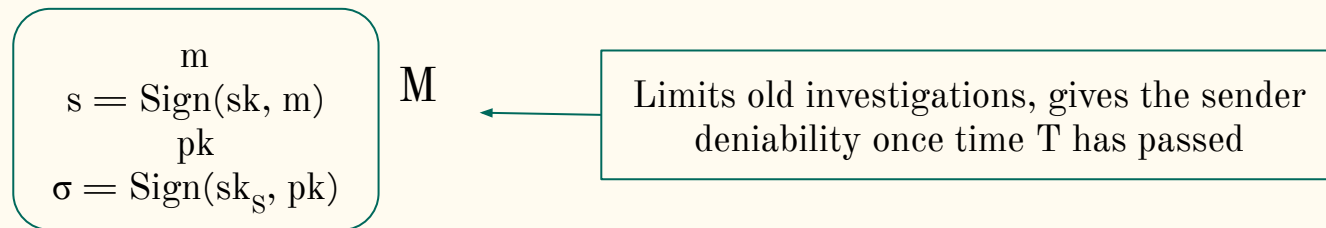
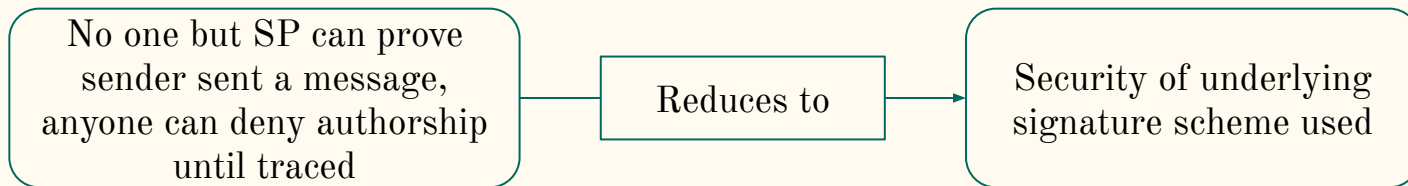
Security Analysis

- Deniability



Security Analysis

- Deniability



Security Analysis

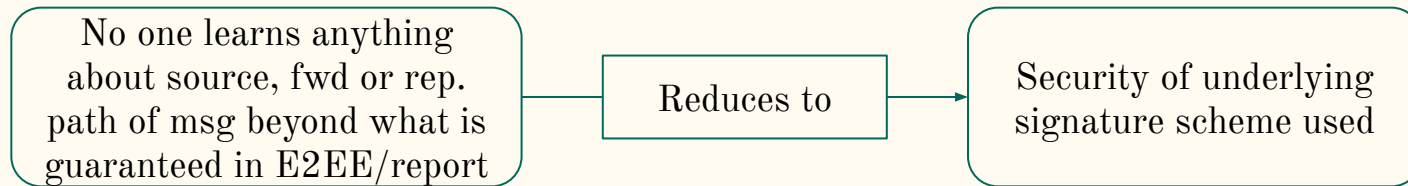
- **Anonymity**

No one learns anything
about source, fwd or rep.
path of msg beyond what is
guaranteed in E2EE/report

$$\begin{array}{l} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} M$$

Security Analysis

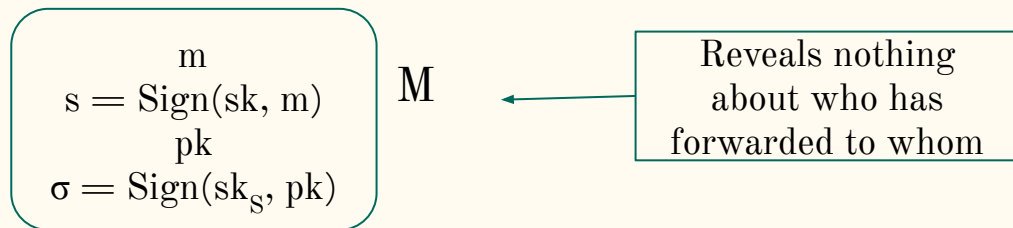
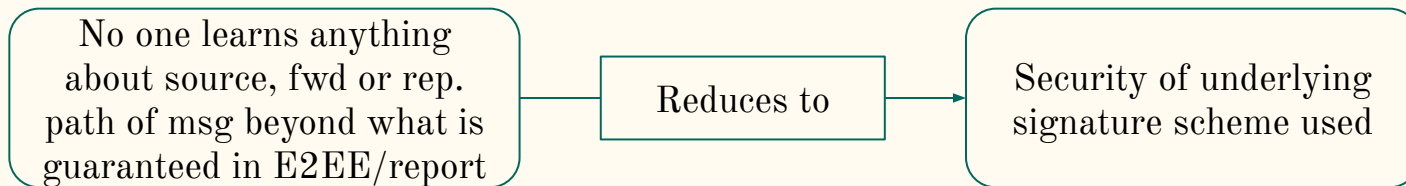
- **Anonymity**



$$\begin{array}{l} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

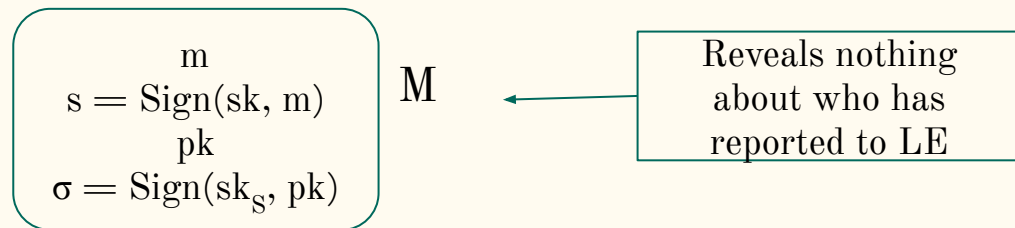
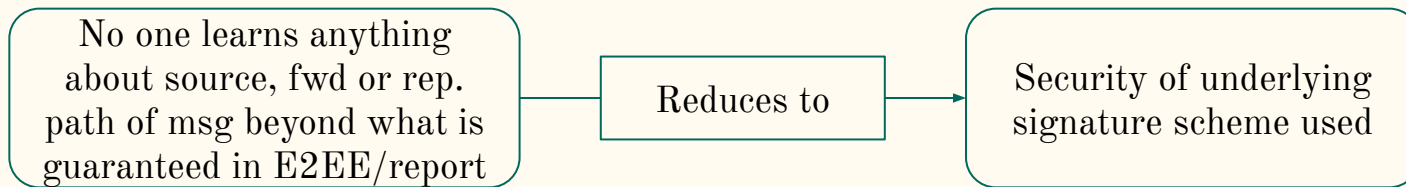
Security Analysis

- **Anonymity**



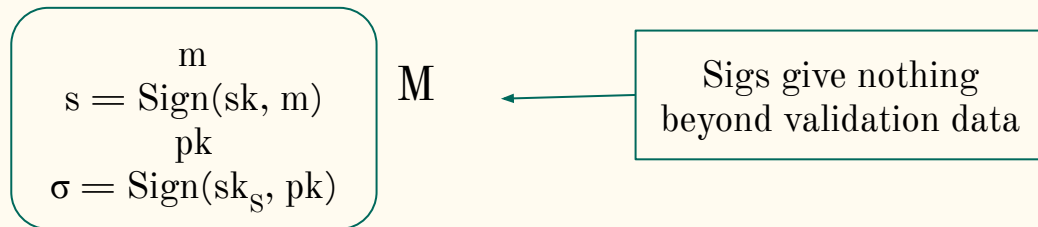
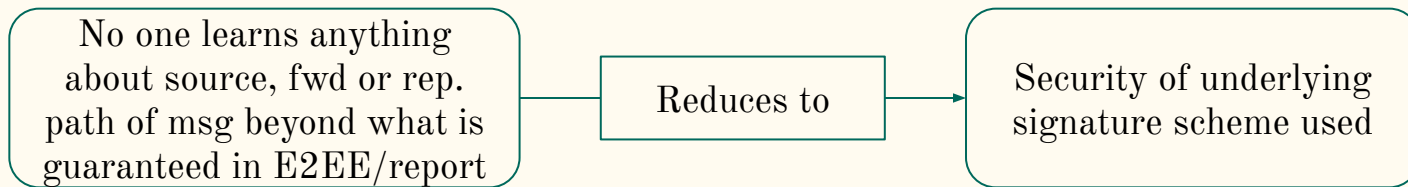
Security Analysis

- **Anonymity**



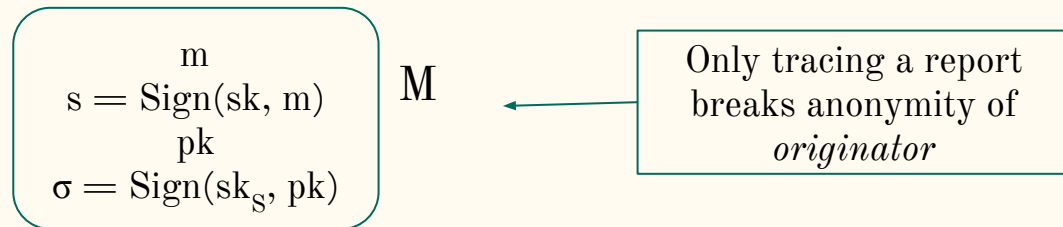
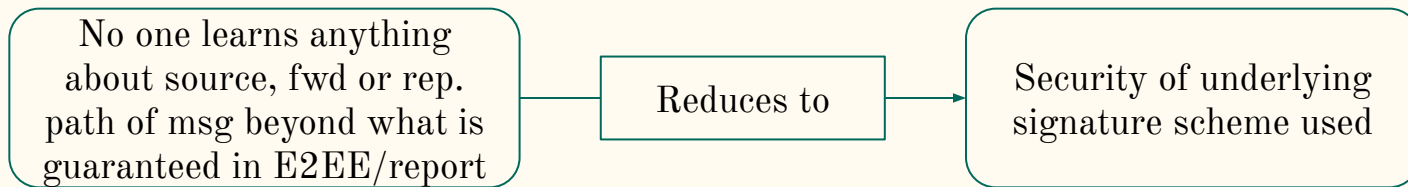
Security Analysis

- **Anonymity**



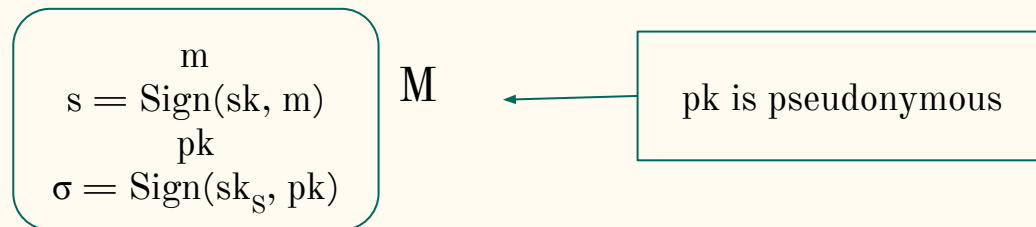
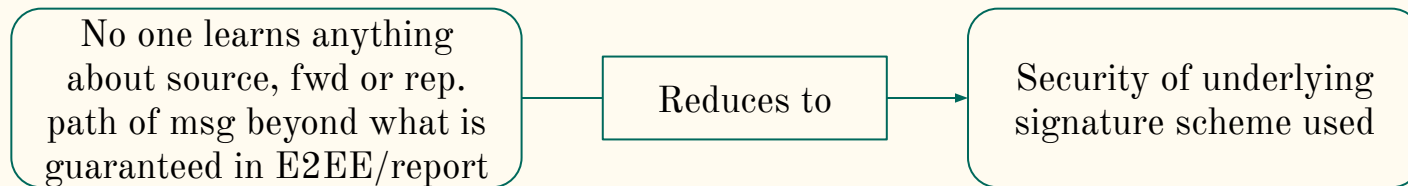
Security Analysis

- **Anonymity**



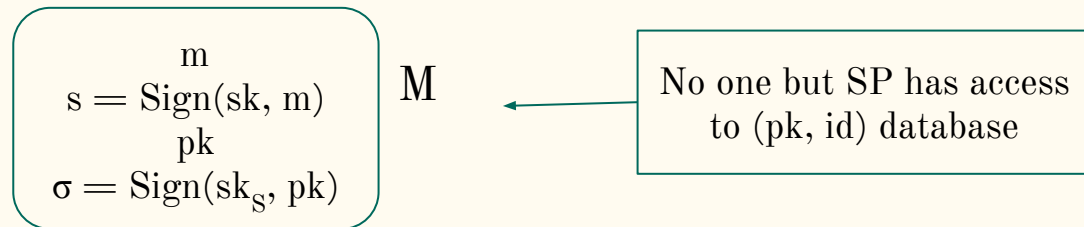
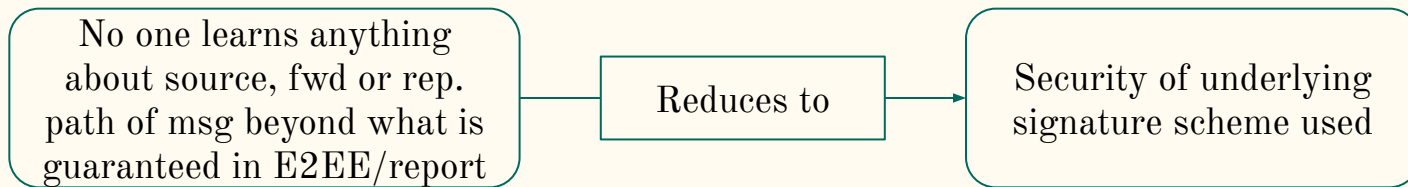
Security Analysis

- **Anonymity**



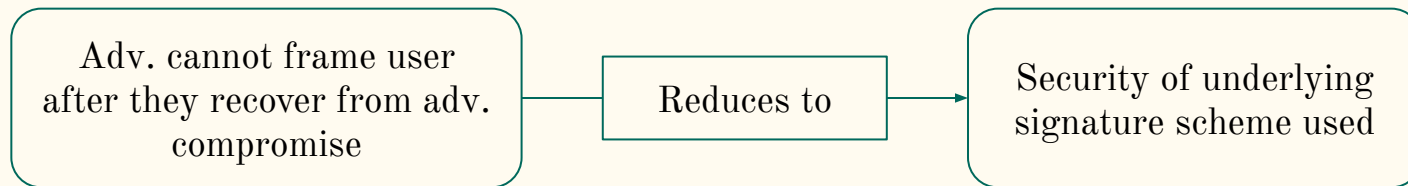
Security Analysis

- **Anonymity**



Security Analysis

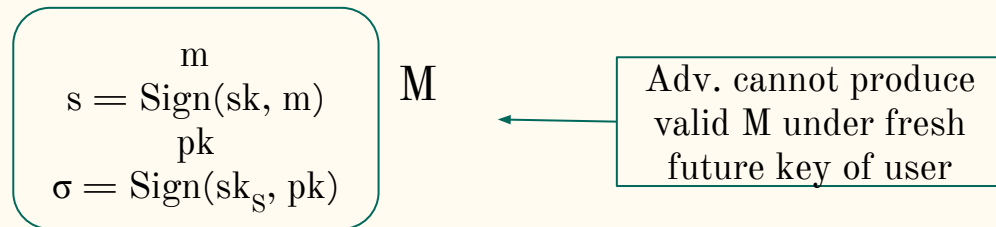
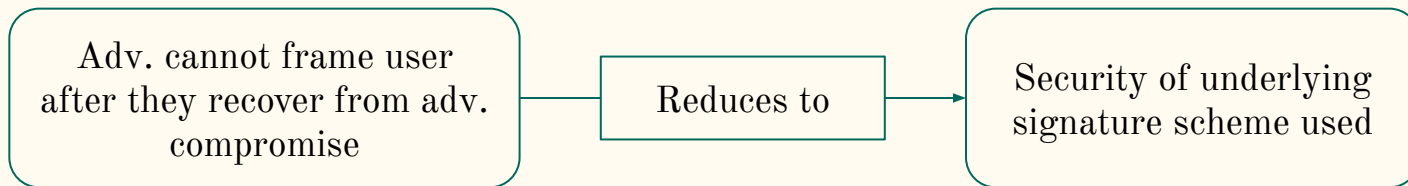
- **Backward**/forward security



$$\begin{array}{l} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

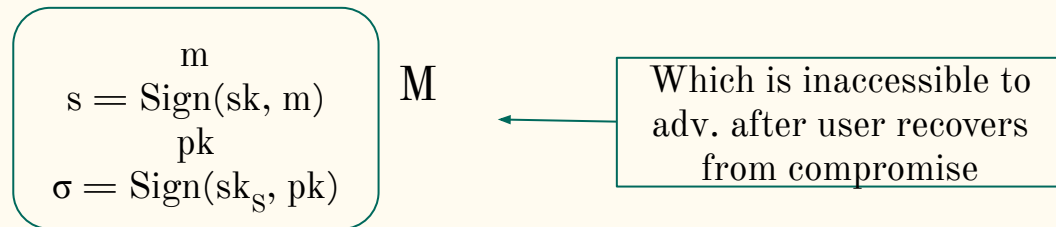
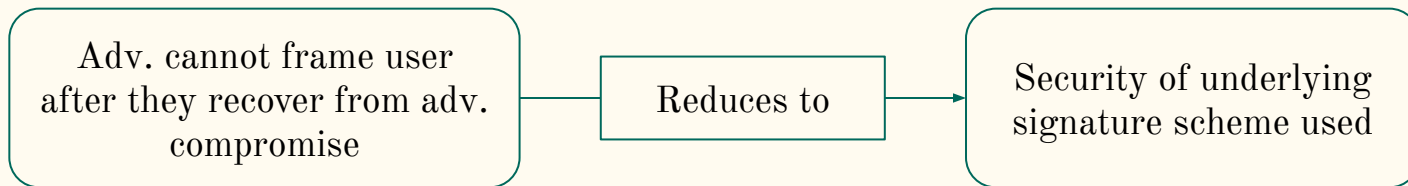
Security Analysis

- **Backward/forward security**



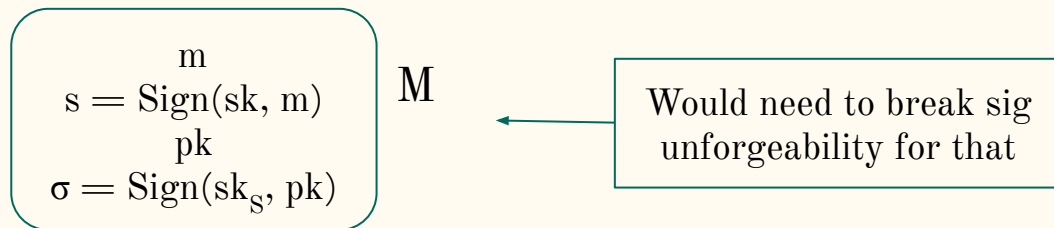
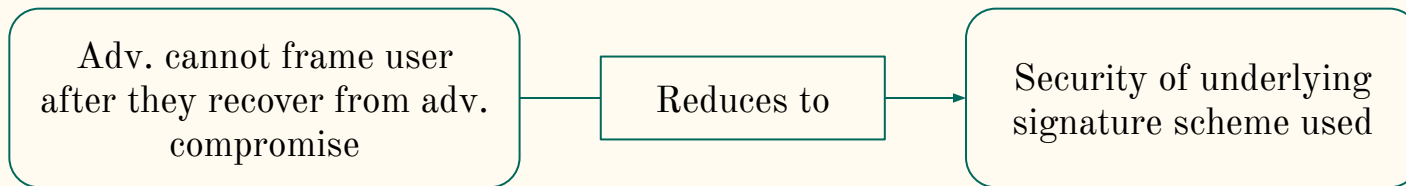
Security Analysis

- **Backward/forward security**



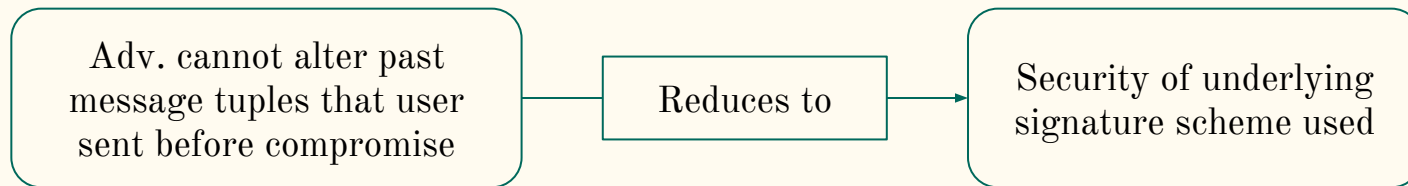
Security Analysis

- **Backward/forward security**



Security Analysis

- Backward/**forward** security



$$\begin{array}{l} m \\ s = \text{Sign}(\text{sk}, m) \\ pk \\ \sigma = \text{Sign}(\text{sk}_s, pk) \end{array} \quad M$$

Security Analysis

- Backward/**forward** security

Adv. cannot alter past message tuples that user sent before compromise

Reduces to

Security of underlying signature scheme used

$$\begin{array}{c} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

Adv. cannot produce valid M under past key of user (pre-compromise)

Security Analysis

- Backward/**forward** security

Adv. cannot alter past
message tuples that user
sent before compromise

Reduces to

Security of underlying
signature scheme used

$$\begin{array}{c} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

Assuming user discards
used keys already!

Security Analysis

- Backward/**forward** security

Adv. cannot alter past
message tuples that user
sent before compromise

Reduces to

Security of underlying
signature scheme used

$$\begin{array}{c} m \\ s = \text{Sign}(sk, m) \\ pk \\ \sigma = \text{Sign}(sk_s, pk) \end{array} \quad M$$

Without access to corr.
sk, adv. must break sig.
to alter M

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- **Benchmarking ATAVISM**
- Tradeoffs and Limitations
- Future Work and Conclusion

Benchmarking ATAVISM

Benchmarking ATAVISM

- Prototype implemented in Typescript

Benchmarking ATAVISM

- Prototype implemented in Typescript
- Session data stored in Postgres-12 database

Benchmarking ATAVISM

- Prototype implemented in Typescript
- Session data stored in Postgres-12 database
- Ed25519 signatures used in implementation

Benchmarking ATAVISM

- Prototype implemented in Typescript
- Session data stored in Postgres-12 database
- Ed25519 signatures used in implementation
- Signal's Double Ratchet algorithm used for encryption

Benchmarking ATAVISM

- Prototype implemented in Typescript
- Session data stored in Postgres-12 database
- Ed25519 signatures used in implementation
- Signal's Double Ratchet algorithm used for encryption
- Tested on system: Ryzen 9 7940HS with 16GB ram running NixOS

Benchmarking ATAVISM

- Prototype implemented in Typescript
- Session data stored in Postgres-12 database
- Ed25519 signatures used in implementation
- Signal's Double Ratchet algorithm used for encryption
- Tested on system: Ryzen 9 7940HS with 16GB ram running NixOS
- We also test on *thin clients* (Poco C65 phones) connected to a remote EU server to test real-world deployability

Benchmarking ATAVISM

- Prototype implemented in Typescript
- Session data stored in Postgres-12 database
- Ed25519 signatures used in implementation
- Signal's Double Ratchet algorithm used for encryption
- Tested on system: Ryzen 9 7940HS with 16GB ram running NixOS
- We also test on *thin clients* (Poco C65 phones) connected to a remote EU server to test real-world deployability
- Barebones Rust implementation used in benchmarking for fairer comparison

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

- Everything tested on same system for fairer comparison!

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

Uses symmetric
crypto

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

Likely
incomparable

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

Still operates in
star network!

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

We do P2P w/
preprocessing

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

Constant time
database lookup

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

Practically a few
microseconds

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

Small overhead
of signatures
and keys

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

Storing a lot of
keys for many
million users?

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

Not too much of an issue!

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

TABLE IV: Execution Time on thin clients (in ms).

Protocols	UKeyGen	NewMsg	RcvMsg
ATAVISM (Figure 1)	0.56	17.4	18.8

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

TABLE IV: Execution Time on thin clients (in ms).

Protocols	UKeyGen	NewMsg	RcvMsg
ATAVISM (Figure 1)	0.56	17.4	18.8

Main overhead is
bandwidth latency

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

TABLE IV: Execution Time on thin clients (in ms).

Protocols	UKeyGen	NewMsg	RcvMsg
ATAVISM (Figure 1)	0.56	17.4	18.8

Still small enough
to be practical on
phones

Benchmarking ATAVISM

TABLE II: Comparison of Execution Time (in ms)

Protocols	UKeyGen	Auth	NewMsg	RcvMsg	Trace
AMF [44]	0.017	—	—	0.2	0.2
Path traceback [45]	0.014	—	—	0.005	—
Tree traceback [45]	0.04	—	—	0.0113	—
Tree-linkable [42]	—	—	0.1	0.23	0.06
Tree-unlinkable [42]	—	—	1.3	2.14	1.7
Hecate [39]	—	0.06	0.03	0.19	0.2
ATAVISM	0.078	0.016	0.0154	0.013	$O(1)$

TABLE III: Comparison of Storage Space (in B).

Protocols	Send (B)	Receive (B)	Trace (B)
AMF [44]	489	489	489
Tree-linkable [42]	256	320	160
Tree-unlinkable [42]	712	1688	648
Hecate [39]	380	484	380
ATAVISM (Figure 1)	160	160	96

TABLE IV: Execution Time on thin clients (in ms).

Protocols	UKeyGen	NewMsg	RcvMsg
ATAVISM (Figure 1)	0.56	17.4	18.8

No other protocol
gives results for
thin clients!

Plan for the afternoon

- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- **Tradeoffs and Limitations**
- Future Work and Conclusion

Tradeoffs and Limitations

- Ethical considerations?

Tradeoffs and Limitations

- Ethical considerations? We assume honest law enforcement!

Tradeoffs and Limitations

- Ethical considerations? Might have a chilling effect on free speech!

Tradeoffs and Limitations

- Ethical considerations? We operate in the context of  legislation.

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!

Tradeoffs and Limitations


- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message 😈

Tradeoffs and Limitations


- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message



Empowered to act
ethically!

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message



Not focusing on
criminal content

Tradeoffs and Limitations


- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message



Like CSAM

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message



Since criminals can
move to less
regulated platforms

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - ATAVISM only traces user ID (eg. phone number)

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs?

Tradeoffs and Limitations


- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? We did the math!

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp

Tradeoffs and Limitations


- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp



Doesn't grow over time. Why?

Tradeoffs and Limitations


- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp



Statute of
limitations!

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp



SP expected to
delete database and
key-rotate after
time T

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp
- Public key signing?

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp
- Public key signing? Can be done asynchronously, not needed at runtime!

Tradeoffs and Limitations

- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp
- Public key signing? Do it when network load is low!

Tradeoffs and Limitations

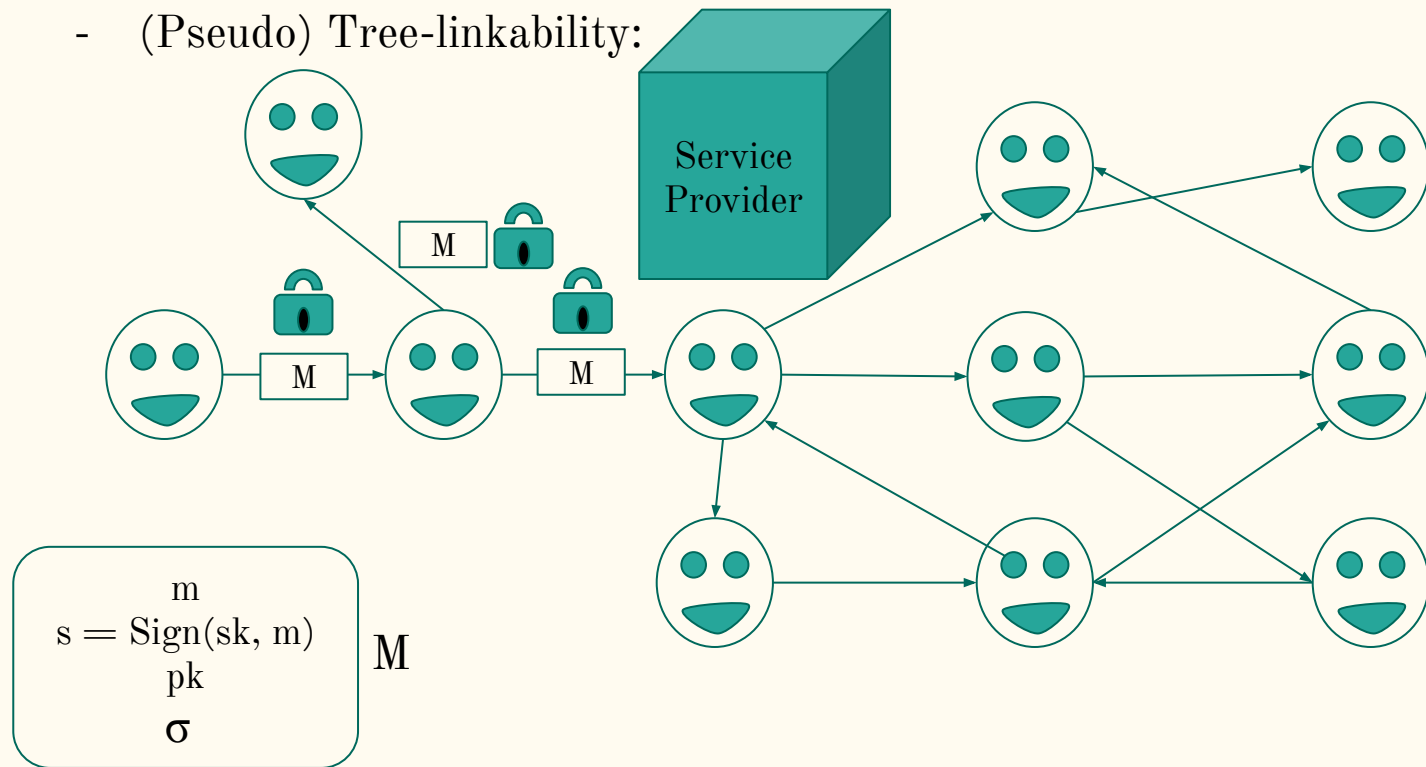
- Ethical considerations? Cannot help against oppressive government policy on what *is* and *isn't* ruled illegal!
- We want to mainly combat fake news and hate speech - cannot do anything if LE flags an otherwise innocuous message
- Cannot help against social engineering - a bad guy can take over an honest person's SIM card
- Storage costs? Need $\sim 288\text{TB}$ for $>2\text{B}$ users sending $\sim 100\text{B}$ messages/day on WhatsApp
- Public key signing? Still want to remove the need to produce so many keypairs and signatures!

Tradeoffs and Limitations

- (Pseudo) Tree-linkability:

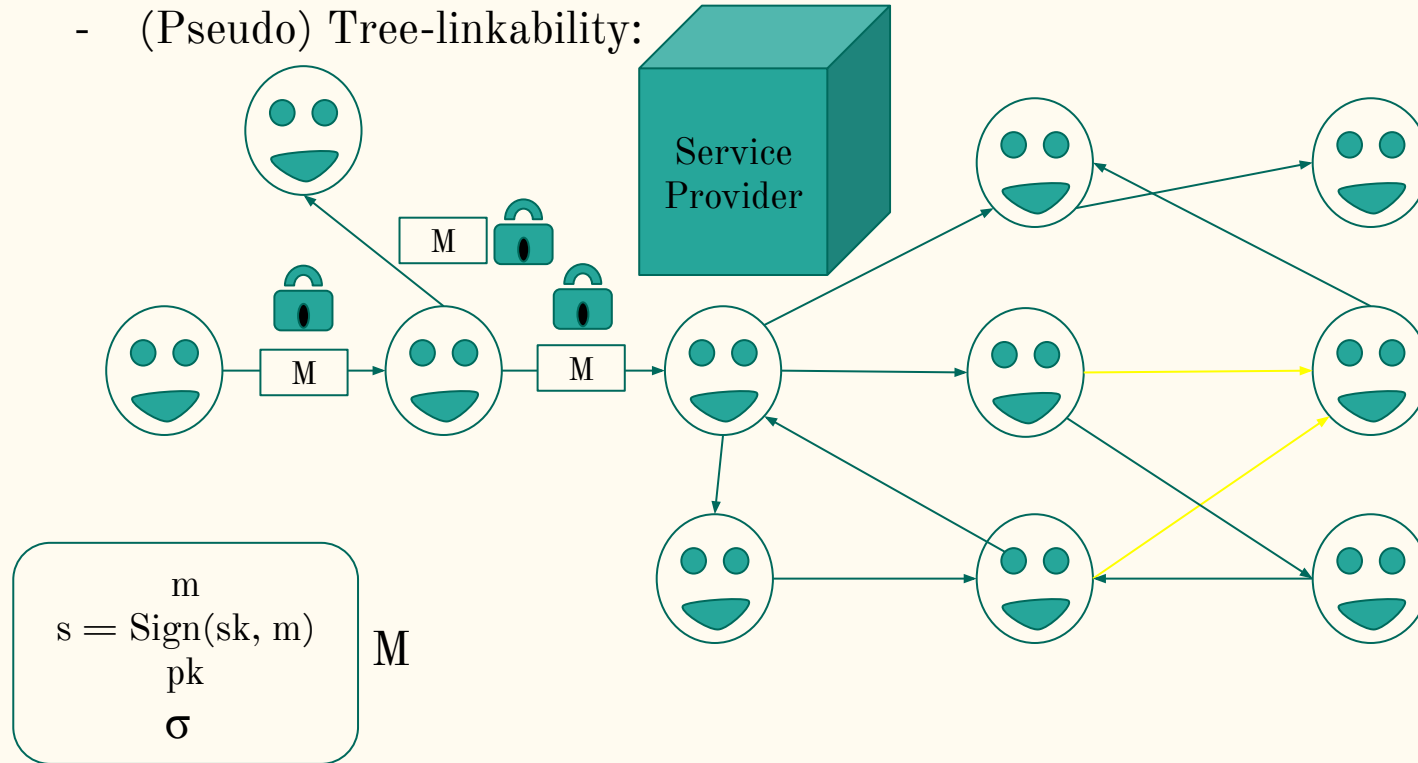
Tradeoffs and Limitations

- (Pseudo) Tree-linkability:



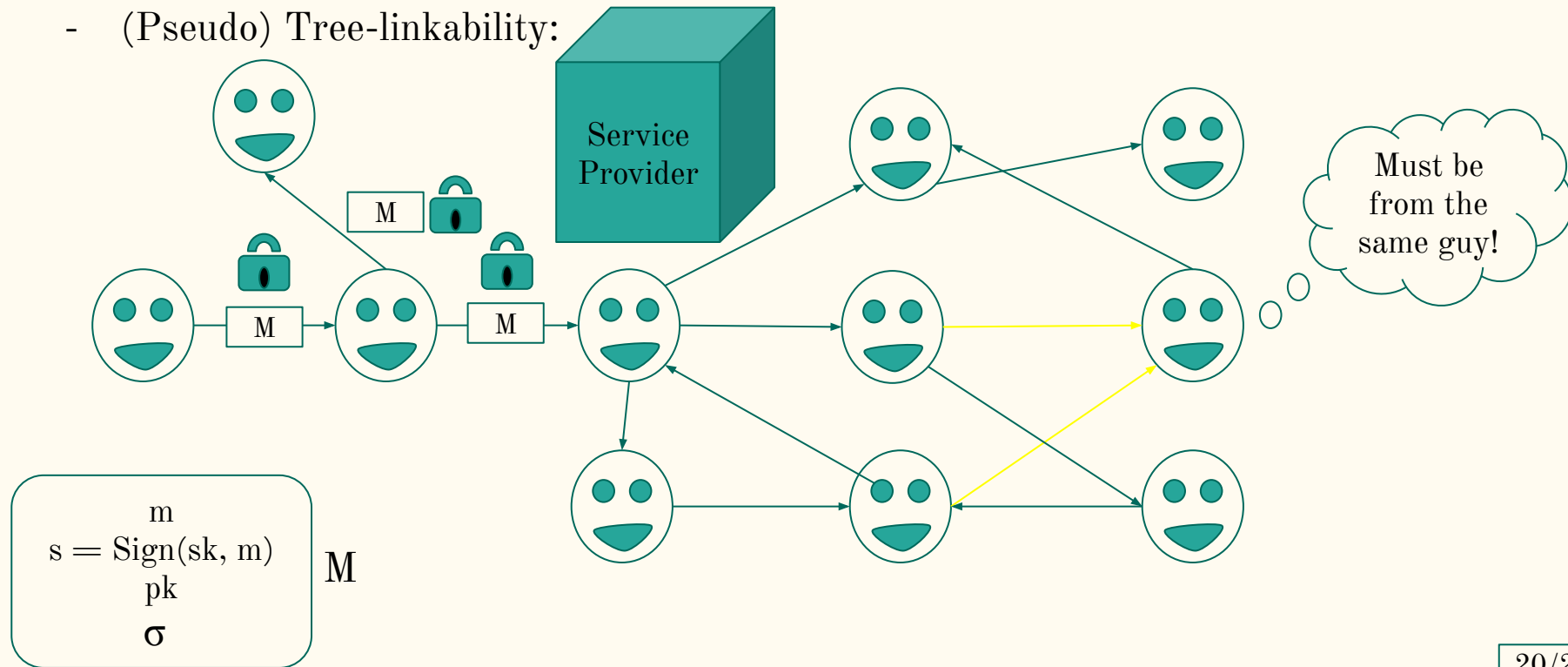
Tradeoffs and Limitations

- (Pseudo) Tree-linkability:



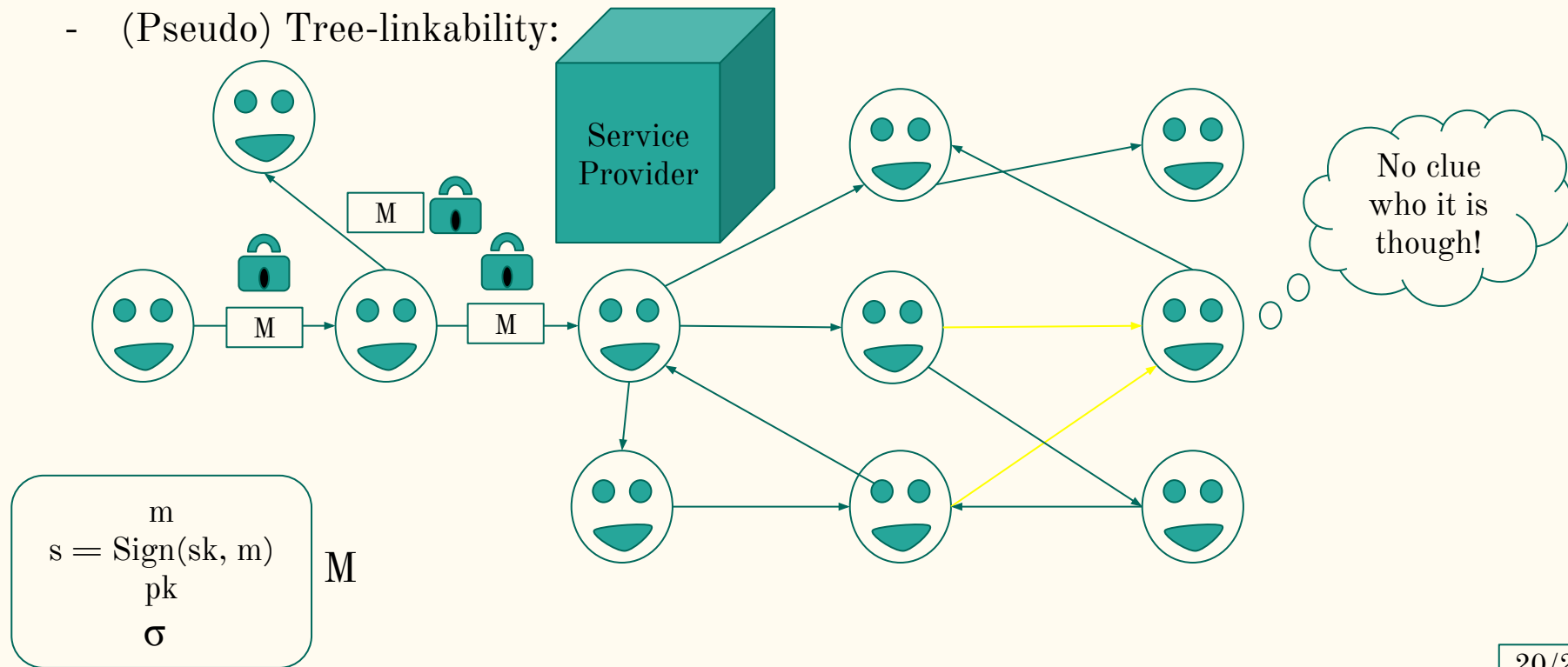
Tradeoffs and Limitations

- (Pseudo) Tree-linkability:



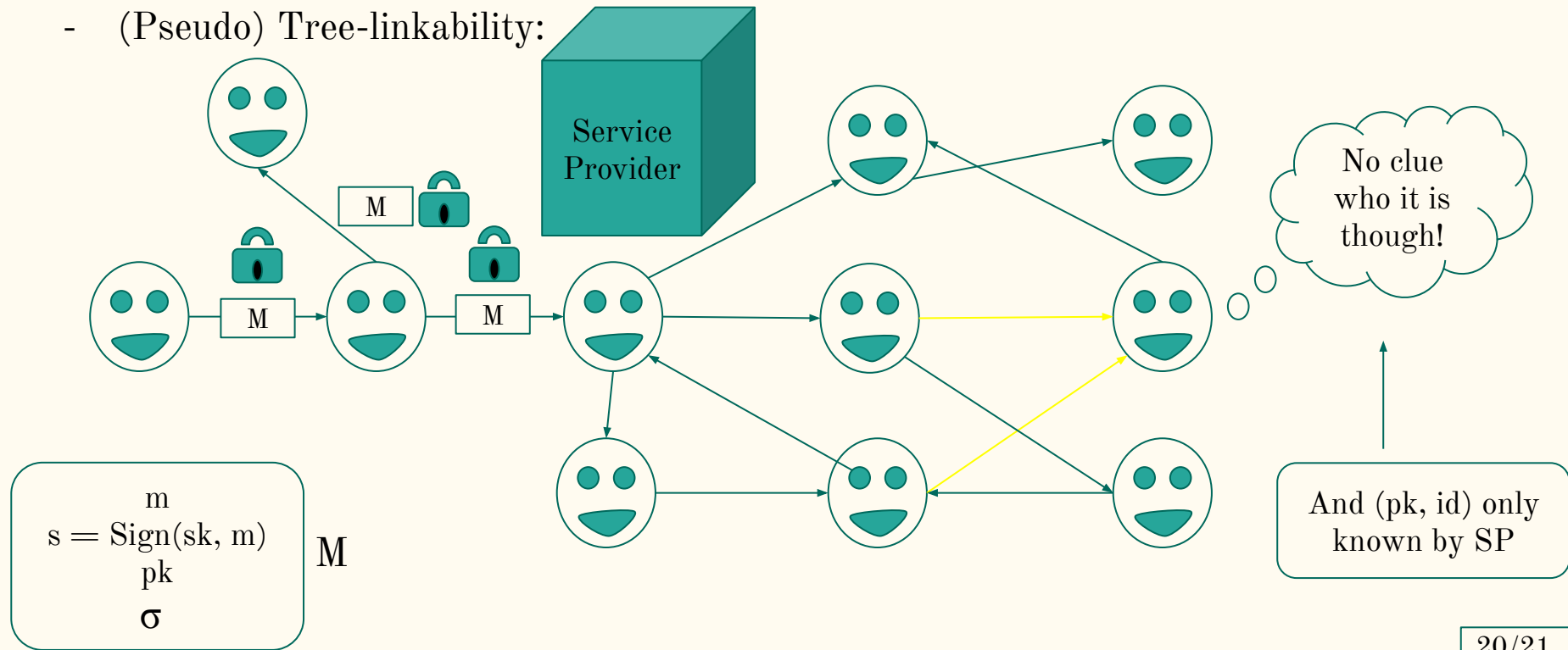
Tradeoffs and Limitations

- (Pseudo) Tree-linkability:



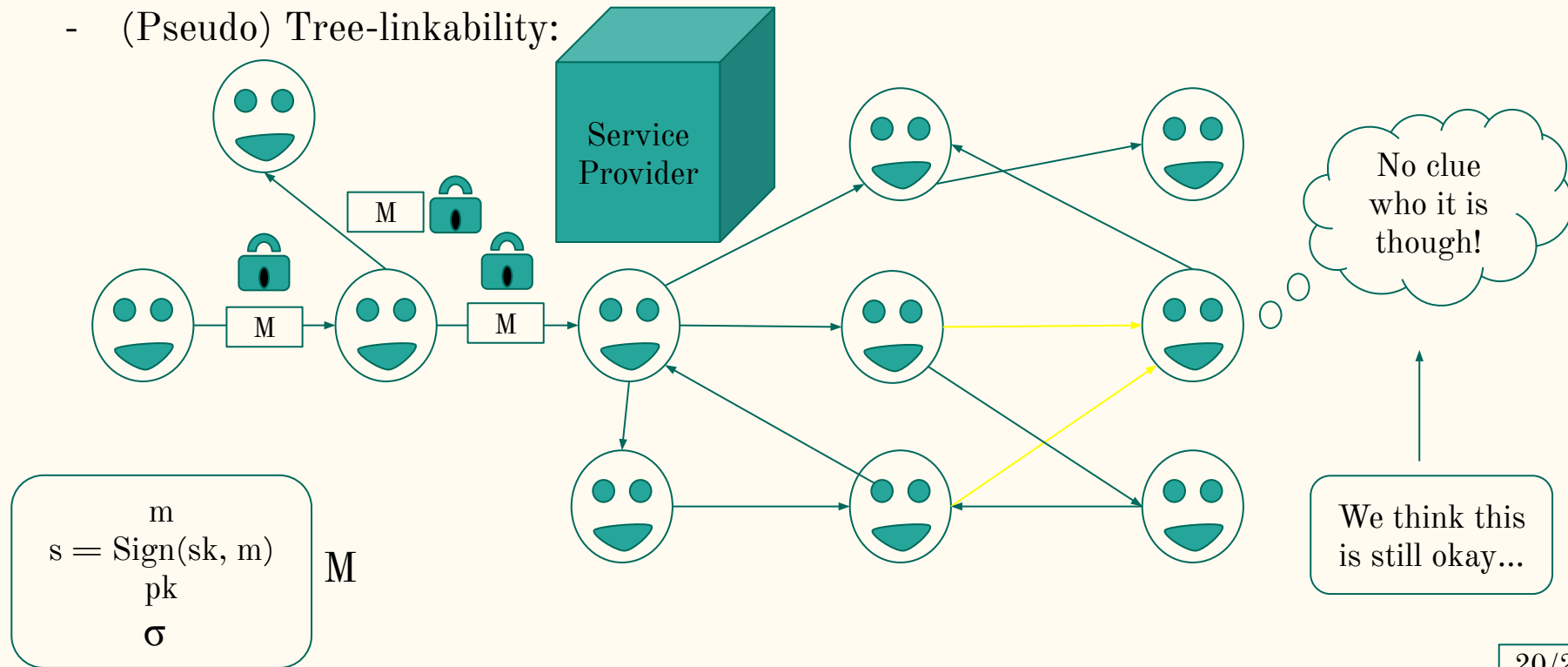
Tradeoffs and Limitations

- (Pseudo) Tree-linkability:



Tradeoffs and Limitations

- (Pseudo) Tree-linkability:



Plan for the afternoon


- The Dilemma of End-to-End Encrypted Messaging
- India's IT Rules
- Private Originator Tracing - Overview
- Security Goals
- Related Work
- Private Originator Tracing - Syntax
- ATAVISM - a protocol sketch
- Security Analysis - Overview
- Benchmarking ATAVISM
- Tradeoffs and Limitations
- **Future Work and Conclusion**

Future Work and Conclusion

- *Semi-honest* \rightarrow *malicious* service provider

Future Work and Conclusion

- *Semi-honest* ➔ *malicious* service provider




Assumed since we *are*
relying on service to work
properly

Future Work and Conclusion

- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement

Future Work and Conclusion

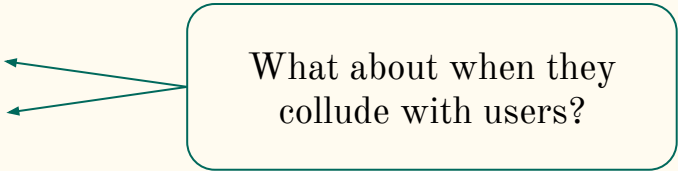
- *Semi-honest* ➡ *malicious* service provider
- *Semi-honest* ➡ *malicious* law enforcement



Can we build *any*
safeguards at all?

Future Work and Conclusion

- *Semi-honest* ➡ *malicious* service provider
- *Semi-honest* ➡ *malicious* law enforcement



What about when they
collude with users?

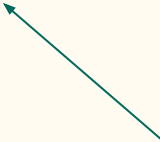
The diagram consists of a light blue rounded rectangle on the right side of the slide. Two teal arrows originate from the left side of this rectangle and point towards the two list items in the 'List-Group' block. The top arrow points to the 'malicious service provider' part of the first item, and the bottom arrow points to the 'malicious law enforcement' part of the second item.

Future Work and Conclusion

- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh

Future Work and Conclusion

- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh



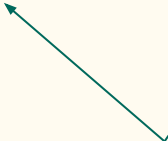
Looking into
structure-preserving
signatures over
equivalence classes

Future Work and Conclusion

- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh
- Pseudo tree-linkability \Rightarrow Full tree unlinkability

Future Work and Conclusion

- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh
- Pseudo tree-linkability \Rightarrow Full tree unlinkability



Looking into *invisible*
and unlinkable
sanitizable signatures

Future Work and Conclusion

- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh
- Pseudo tree-linkability \Rightarrow Full tree unlinkability
- Filter spam reports to LE?

Future Work and Conclusion

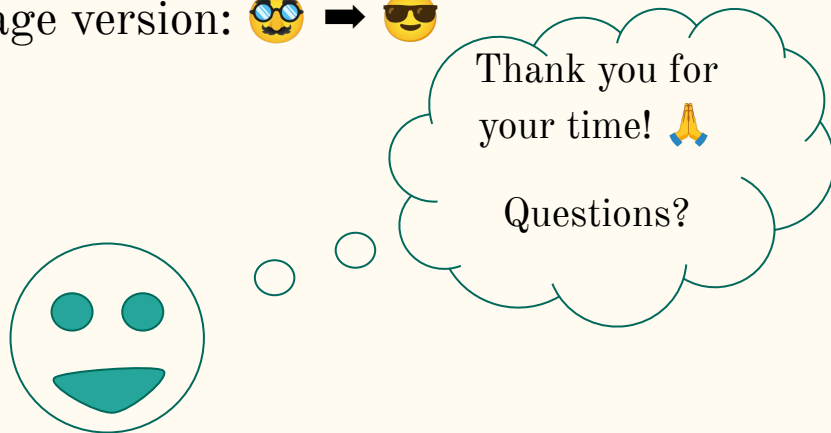
- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh
- Pseudo tree-linkability \Rightarrow Full tree unlinkability
- Filter spam reports to LE? Thought not technically illegal!

Future Work and Conclusion

- *Semi-honest* ➡ *malicious* service provider
- *Semi-honest* ➡ *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh
- Pseudo tree-linkability ➡ Full tree unlinkability
- Filter spam reports to LE? Thought not technically illegal!
- Distributed storage version: 🧐 ➡ 😎

Future Work and Conclusion

- *Semi-honest* \Rightarrow *malicious* service provider
- *Semi-honest* \Rightarrow *malicious* law enforcement
- Optimize server involvement in preprocessing/refresh
- Pseudo tree-linkability \Rightarrow Full tree unlinkability
- Filter spam reports to LE? Thought not technically illegal!
- Distributed storage version: 🧐 \Rightarrow 😎



References

- Indian IT Rules, 2021, Available: <https://www.meity.gov.in/static/uploads/2024/02/Information-Technology-Intermediary-Guidelines-and-Digital-Media-Ethics-Code-Rules-2021-updated-06.04.2023-.pdf>
- P. Grubbs, J. Lu, and T. Ristenpart, “Message franking via committing authenticated encryption,” in *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, ser. Lecture Notes in Computer Science, J. Katz and H. Shacham, Eds., vol. 10403. Springer, 2017, pp. 66–97.
- Y. Dodis, P. Grubbs, T. Ristenpart, and J. Woodage, “Fast message franking: From invisible salamanders to encryptment,” in *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, ser. Lecture Notes in Computer Science, H. Shacham and A. Boldyreva, Eds., vol. 10991. Springer, 2018, pp. 155–186.
- N. Tyagi, P. Grubbs, J. Len, I. Miers, and T. Ristenpart, “Asymmetric message franking: Content moderation for metadata-private end-to-end encryption,” in *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, ser. Lecture Notes in Computer Science, A. Boldyreva and D. Micciancio, Eds., vol. 11694. Springer, 2019, pp. 222–250.
- N. Tyagi, I. Miers, and T. Ristenpart, “Traceback for end-to-end encrypted messaging,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, Eds. ACM, 2019, pp. 413–421.
- R. Issa, N. Alhaddad, and M. Varia, “Hecate: Abuse reporting in secure messengers with sealed sender,” in *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, K. R. B. Butler and K. Thomas, Eds. USENIX Association, 2022, pp. 2335–2352.
- C. Peale, S. Eskandarian, and D. Boneh, “Secure complaint-enabled source-tracking for encrypted messaging,” in *CCS ’21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, Y. Kim, J. Kim, G. Vigna, and E. Shi, Eds. ACM, 2021, pp. 1484–1506.
- L. Liu, D. S. Roche, A. Theriault, and A. Yerukhimovich, “Fighting fake news in encrypted messaging with the fuzzy anonymous complaint tally system (FACTS),” in *29th Annual Network and Distributed System Security Symposium, NDSS 2022, San Diego, California, USA, April 24-28, 2022*. The Internet Society, 2022.
- E. Kenney, Q. Tang, and C. Wu, “Anonymous traceback for end-to-end encryption,” in *Computer Security - ESORICS 2022 - 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26-21, 2022, Proceedings, Part II*, ser. Lecture Notes in Computer Science, V. Atluri, R. D. Pietro, C. D. Jensen, and W. Meng, Eds., vol. 13555. Springer, 2022, pp. 42–62.