Secure Messaging

Britta Hale

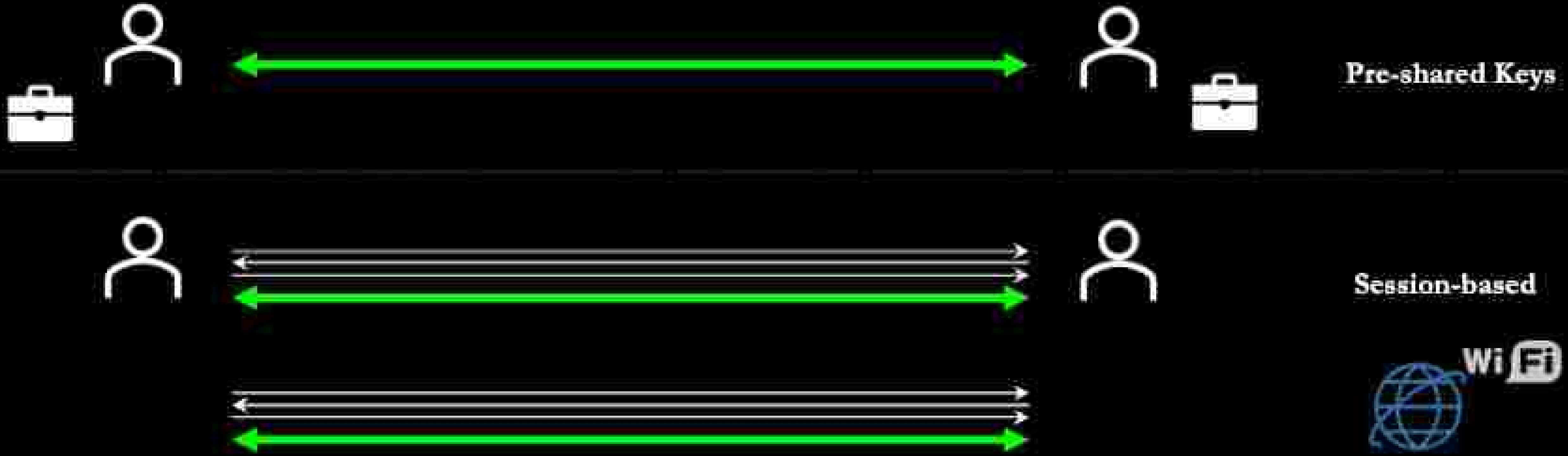
*The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Secure Messaging

Britta Hale

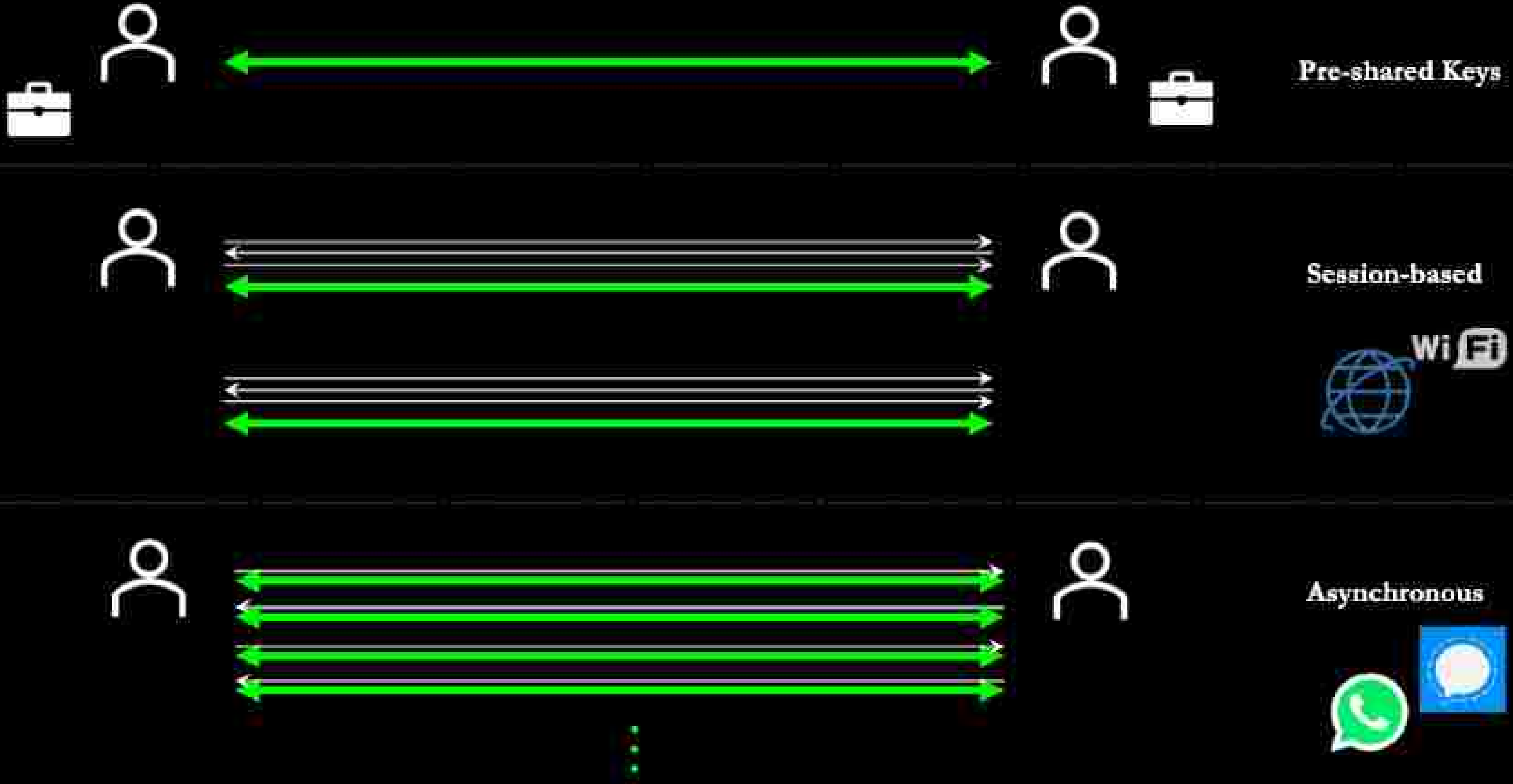
*The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Pre-shared Keys



Pre-shared Keys

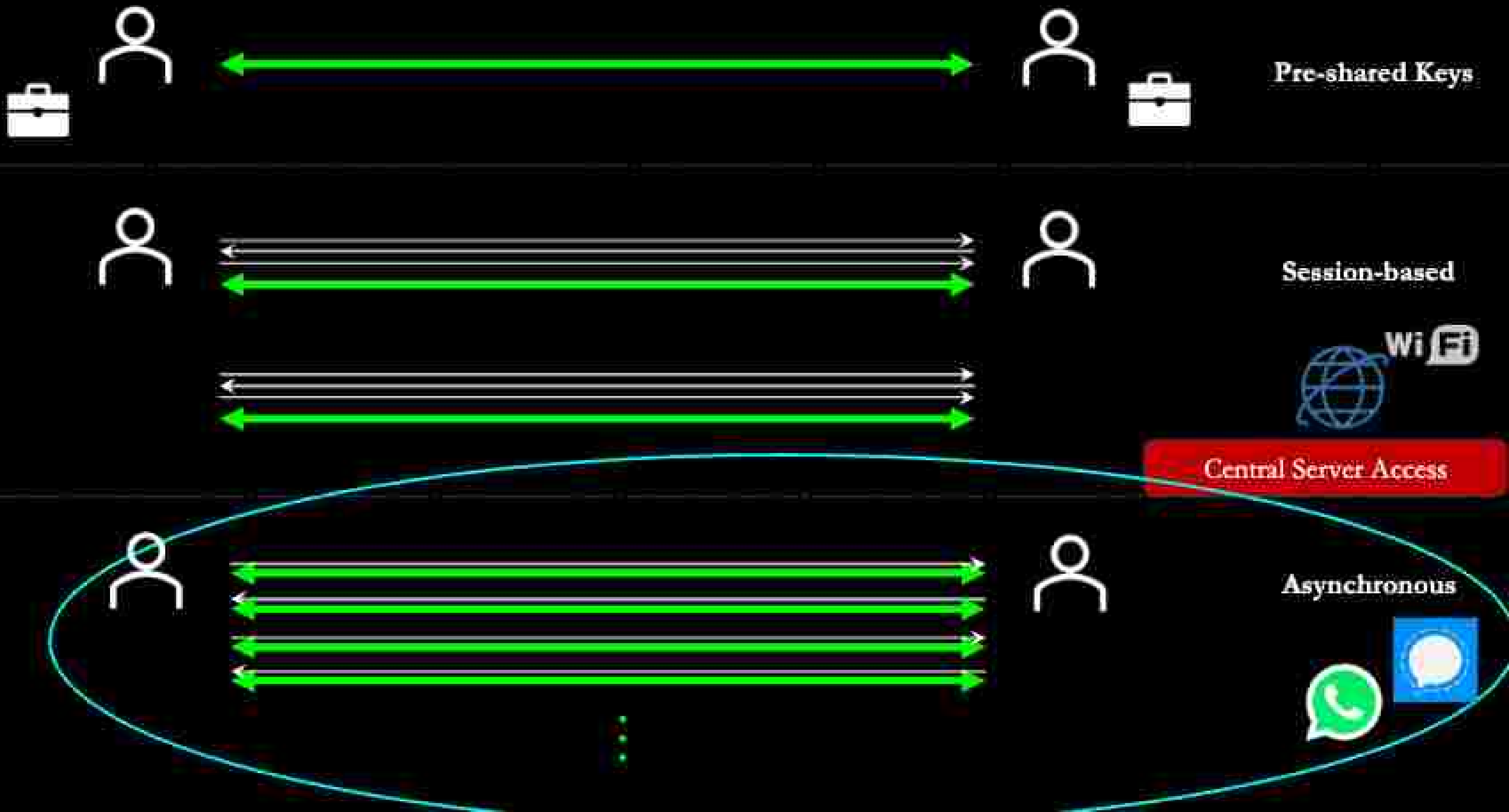
Session-based



Pre-shared Keys

Session-based

Asynchronous



Pre-shared Keys

Session-based

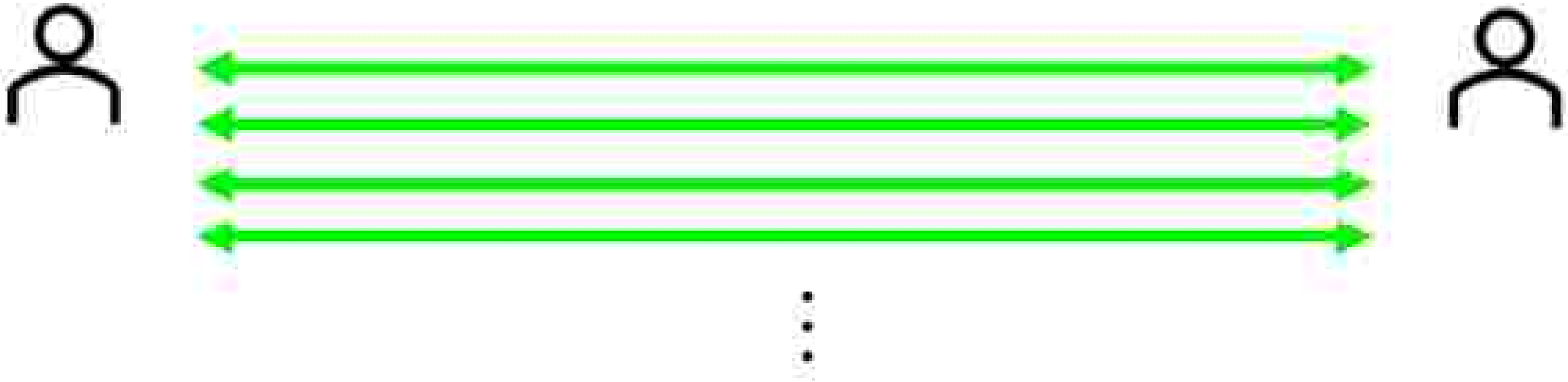
Central Server Access

Asynchronous

Secure End-to-End Messaging

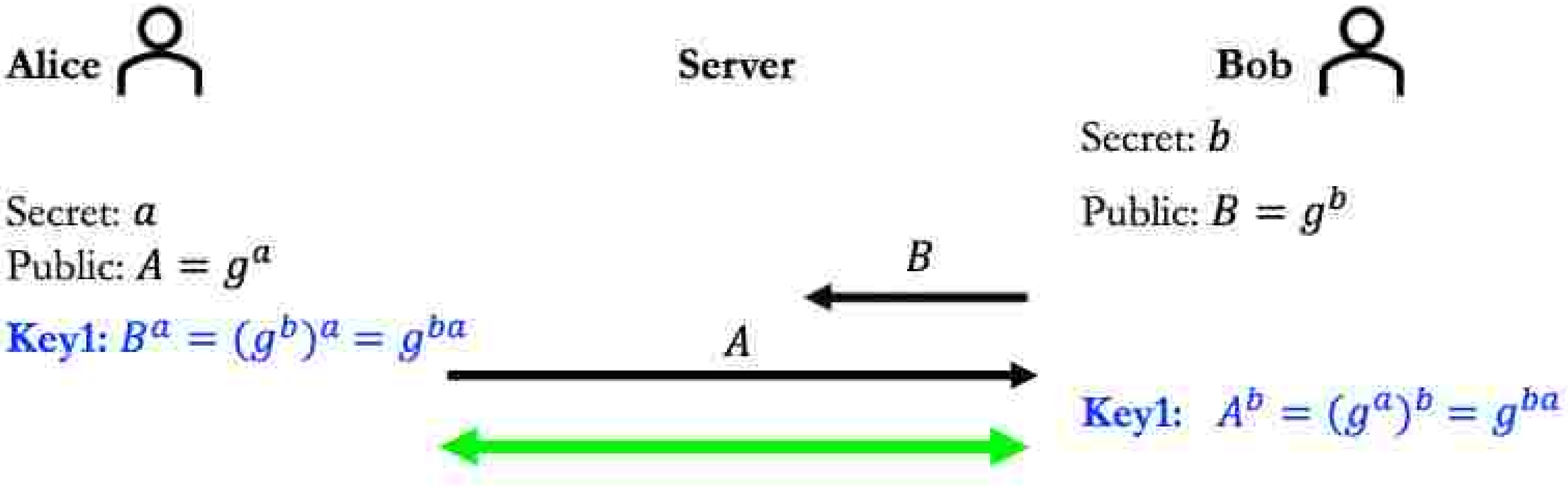
Secure Messaging

The Signal Protocol



Asynchronous

The Signal Protocol



Alice

Server

Bob

Secret: b

Public: $B = g^b$

Secret: a

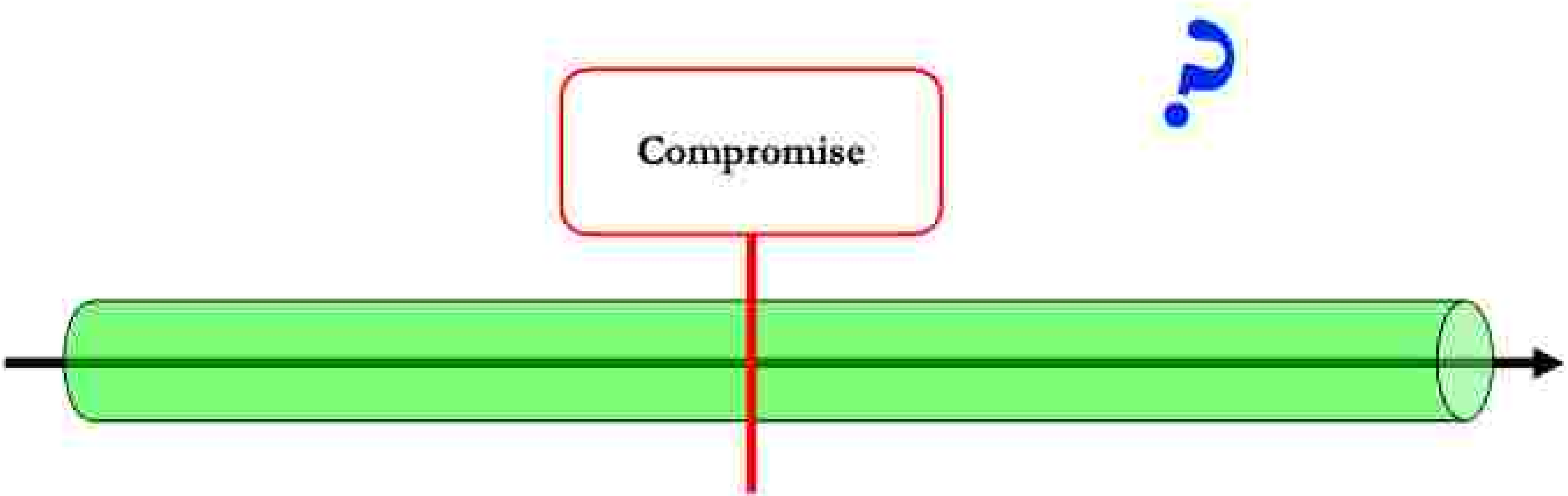
Public: $A = g^a$

B

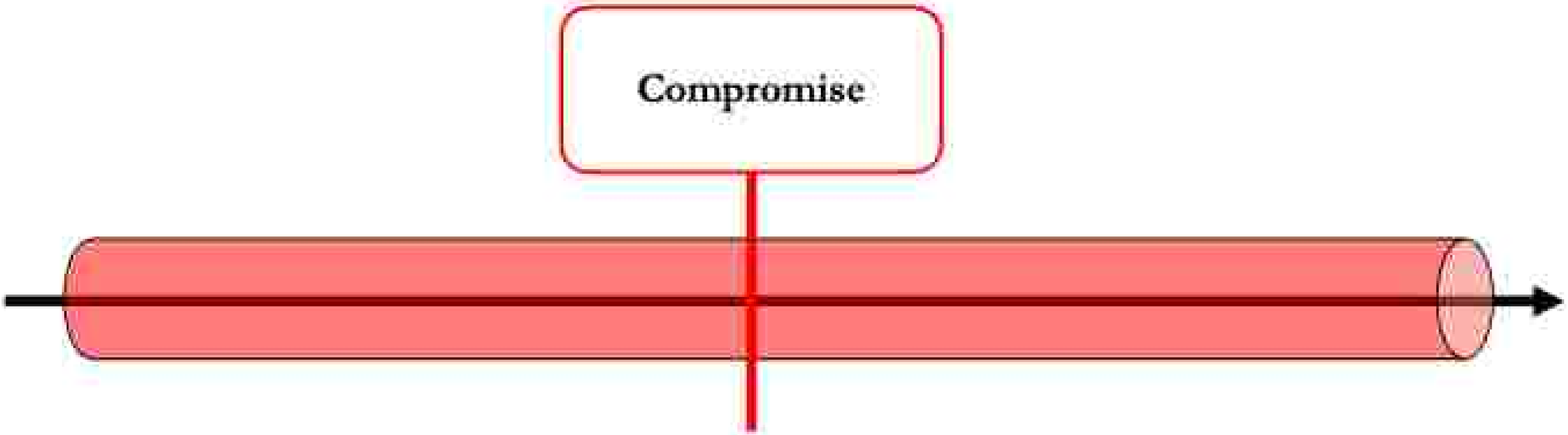
Key1: $B^a = (g^b)^a = g^{ba}$

A

Key1: $A^b = (g^a)^b = g^{ba}$

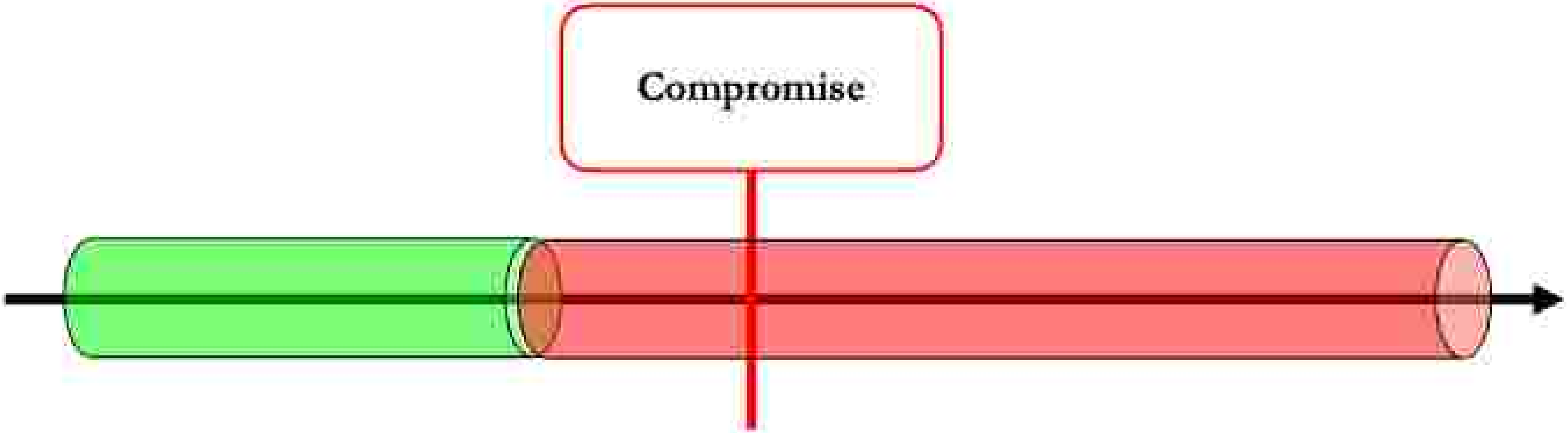


Compromise



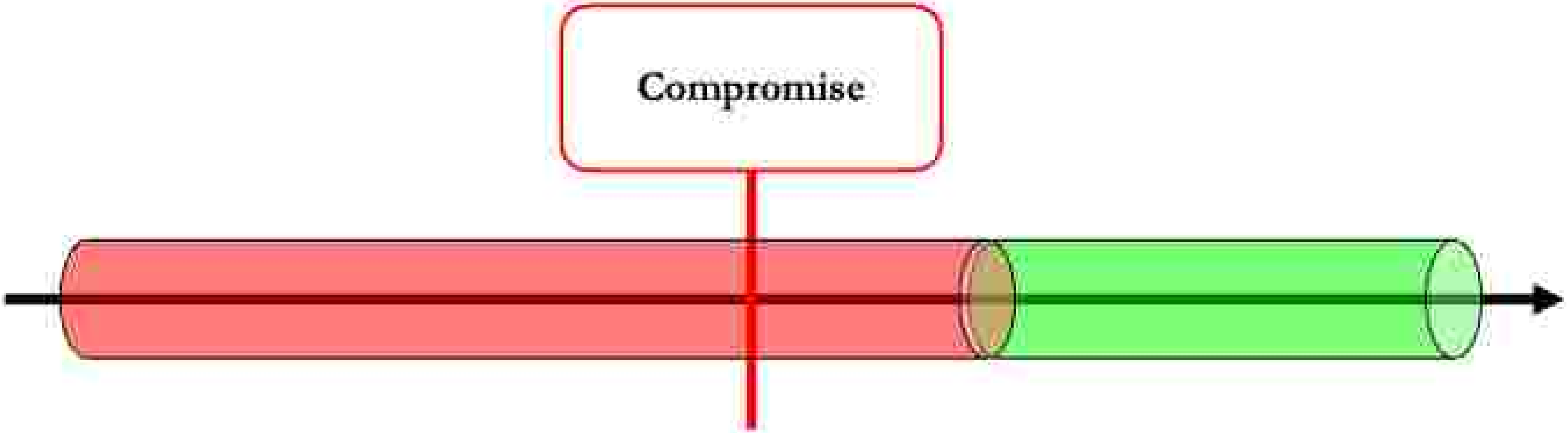
Compromise

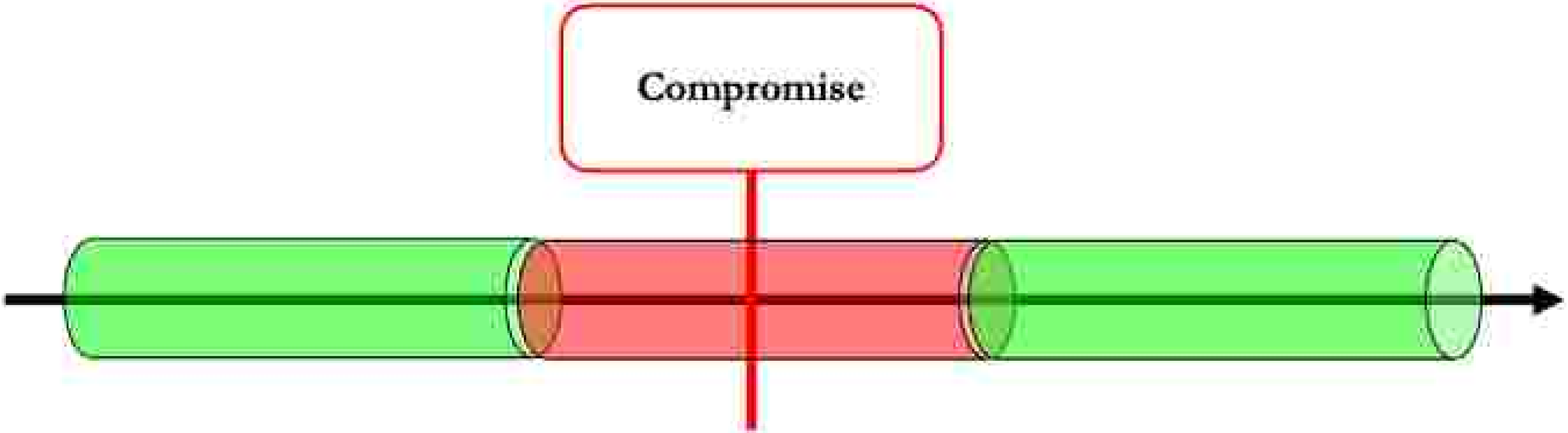
Desired Property #1:
Forward Secrecy



Desired Property #2:

Post-Compromise Security (PCS)





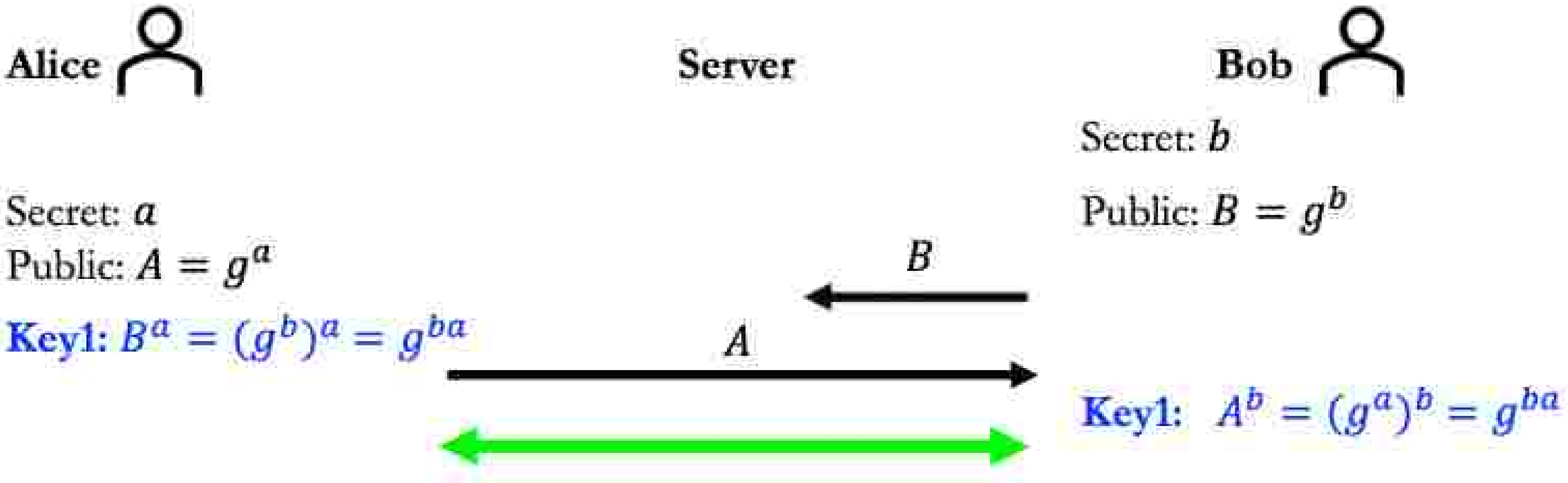
Solution: update keys periodically

Epochs

*Condition: adversary is passive for one epoch

The Signal Protocol

The Signal Protocol



Alice

Server

Bob

Secret: b

Public: $B = g^b$

Secret: a

Public: $A = g^a$

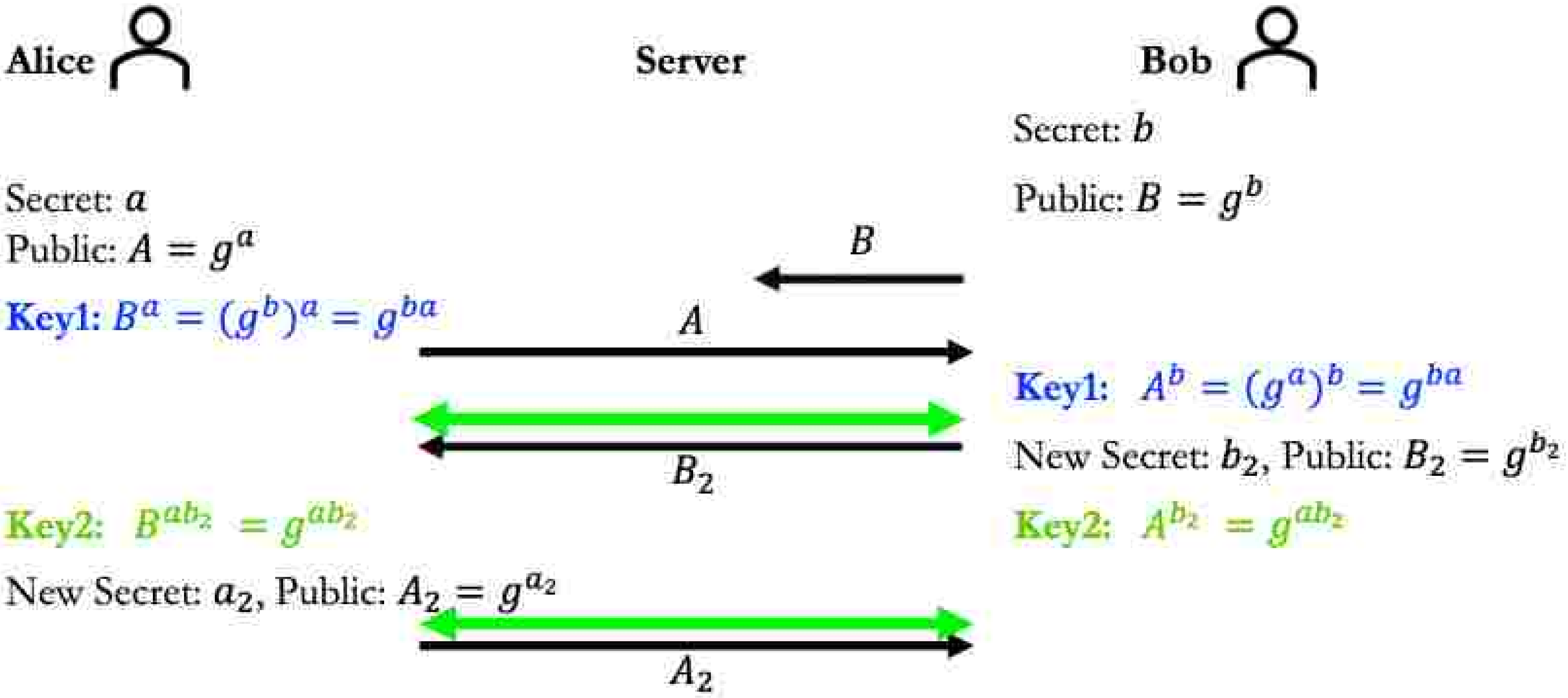
B

Key1: $B^a = (g^b)^a = g^{ba}$

A

Key1: $A^b = (g^a)^b = g^{ba}$

The Signal Protocol



Alice

Server

Bob

Secret: b

Public: $B = g^b$

Secret: a

Public: $A = g^a$

B

Key1: $B^a = (g^b)^a = g^{ba}$

A

Key1: $A^b = (g^a)^b = g^{ba}$

New Secret: b_2, Public: $B_2 = g^{b_2}$

B_2

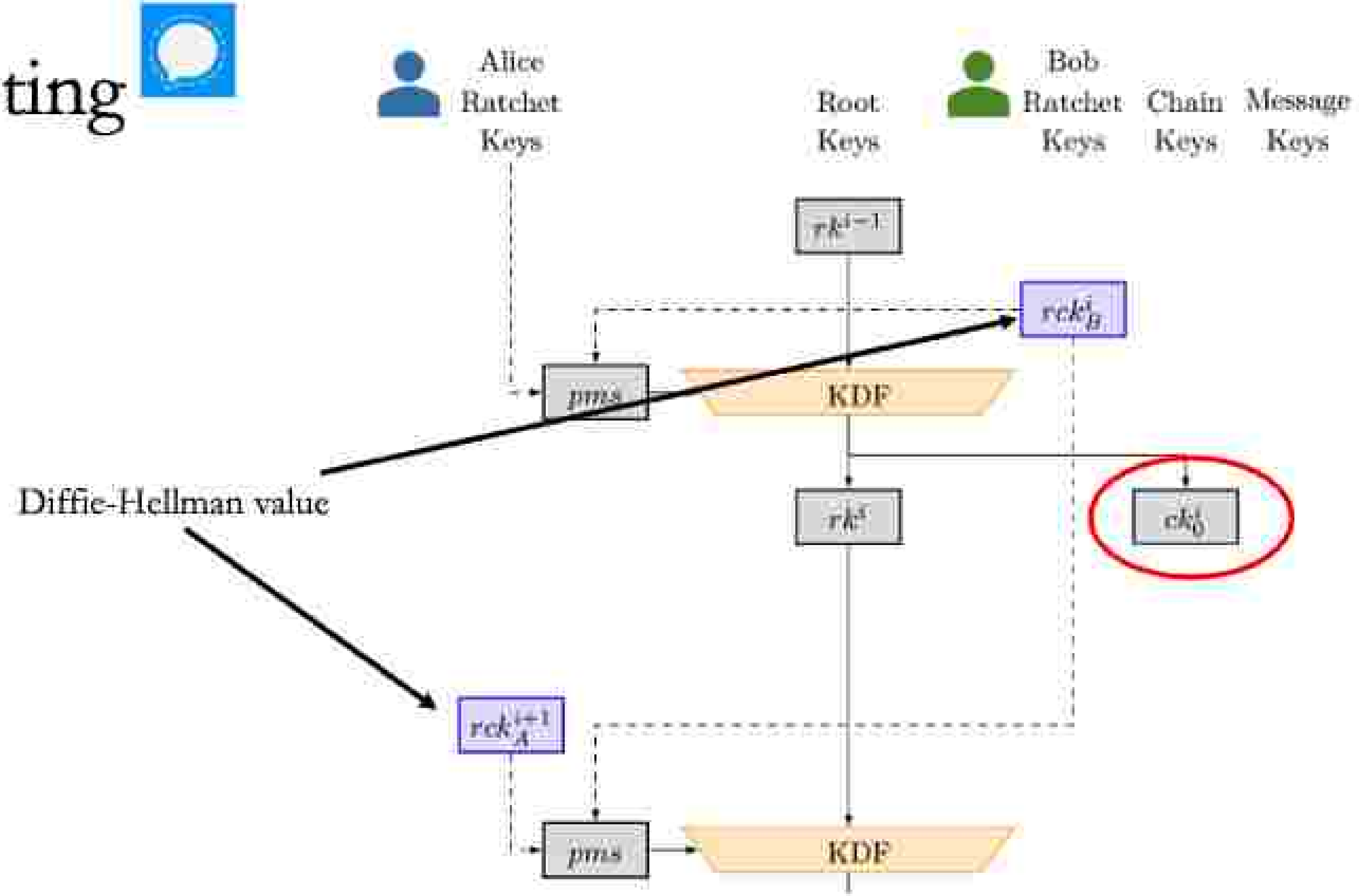
Key2: $B^{ab_2} = g^{ab_2}$

Key2: $A^{b_2} = g^{ab_2}$

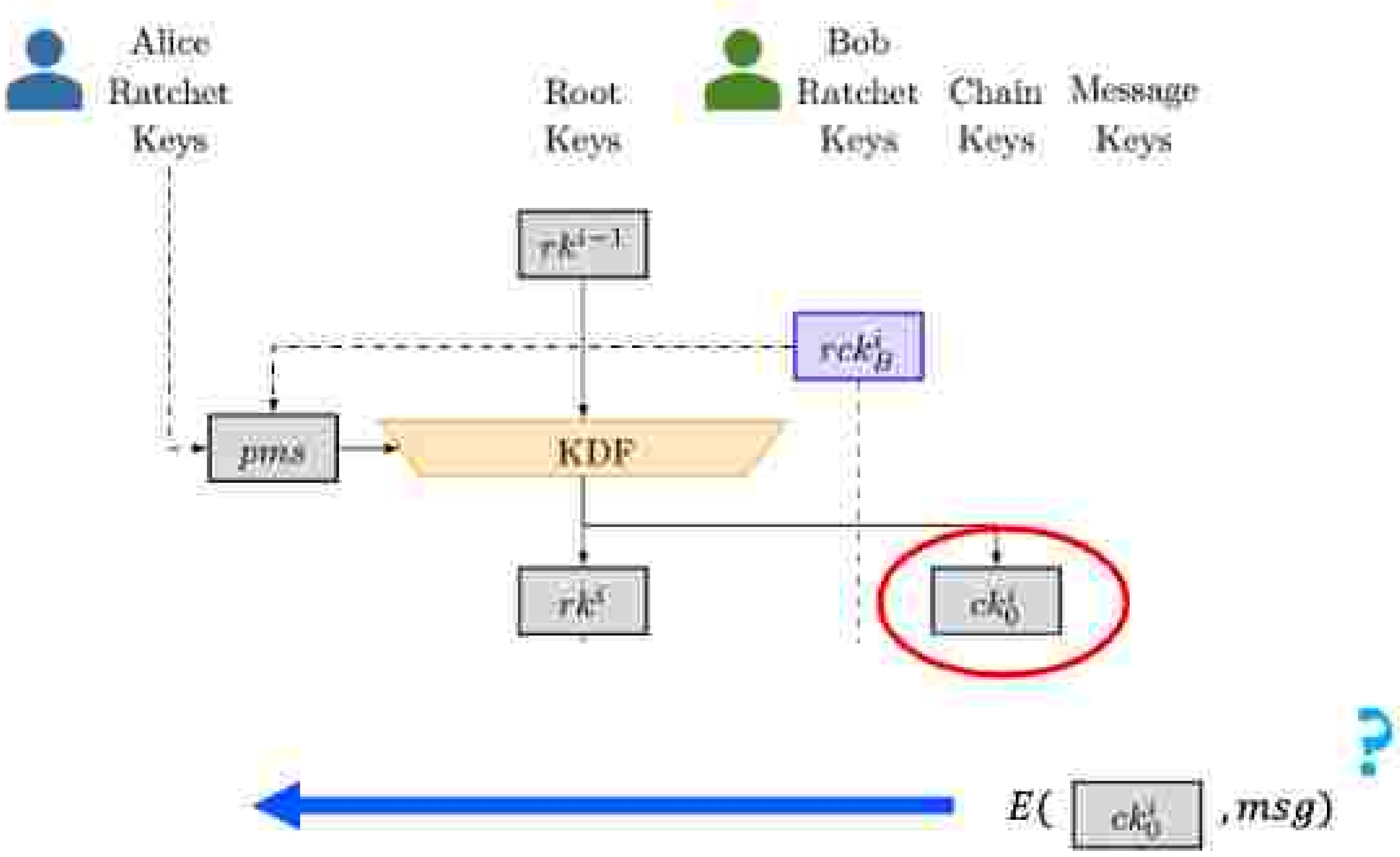
New Secret: a_2, Public: $A_2 = g^{a_2}$

A_2

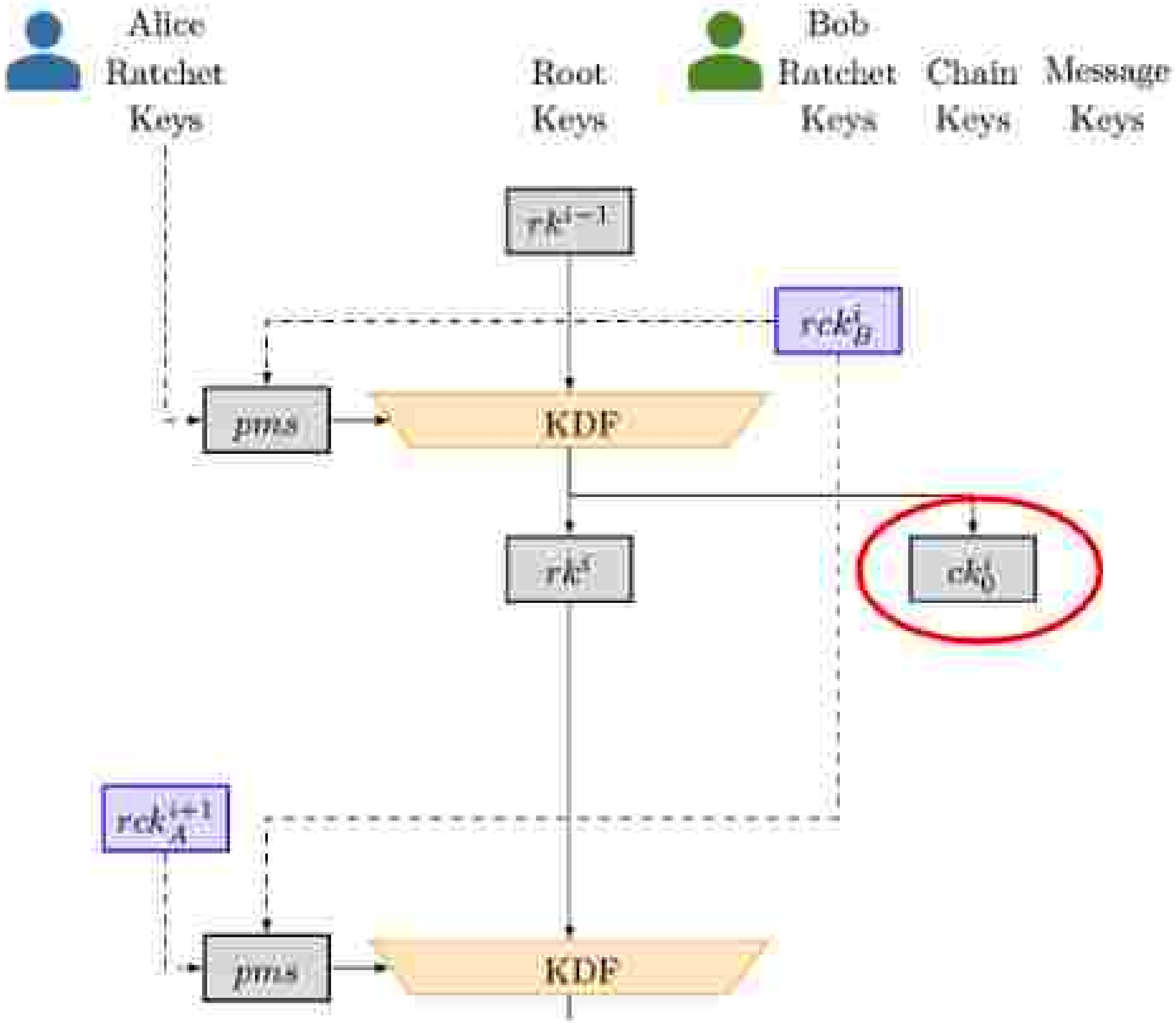
Ratcheting


Alice
Ratchet
Keys
Root
Keys
Bob
Ratchet
Keys
Chain
Keys
Message
Keys
rk^{i-1}
rck_B^i
pms
KDF
Diffie-Hellman value
rk^i
ck_0^i
rck_A^{i+1}
pms
KDF

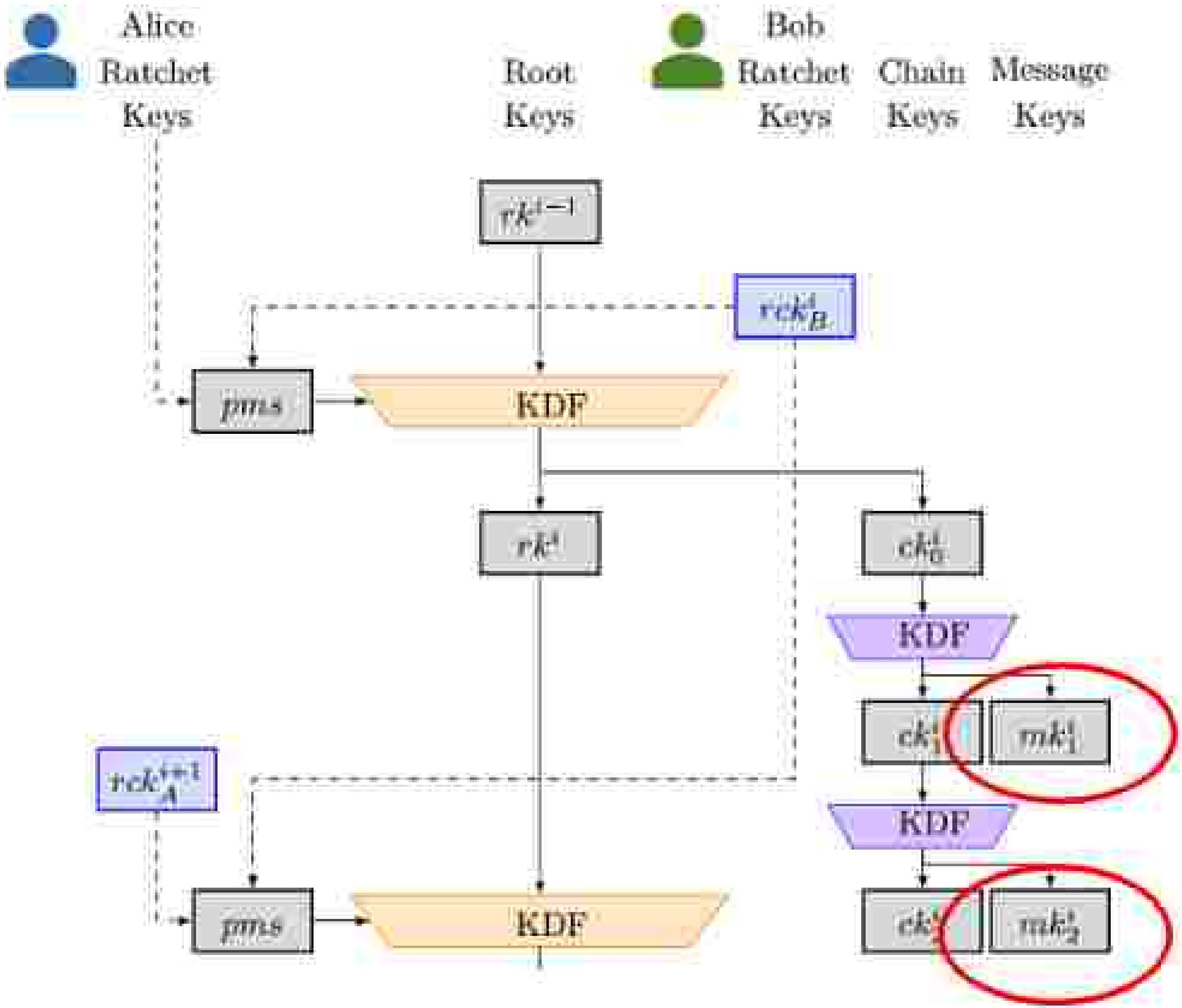
Ratcheting


Alice Ratchet Keys
Root Keys
Bob Ratchet Keys
Chain Keys
Message Keys
rk^{i-1}
rck_B^i
pms
KDF
rk^i
ck_0^i
$E(\boxed{ck_0^i}, msg)$
?

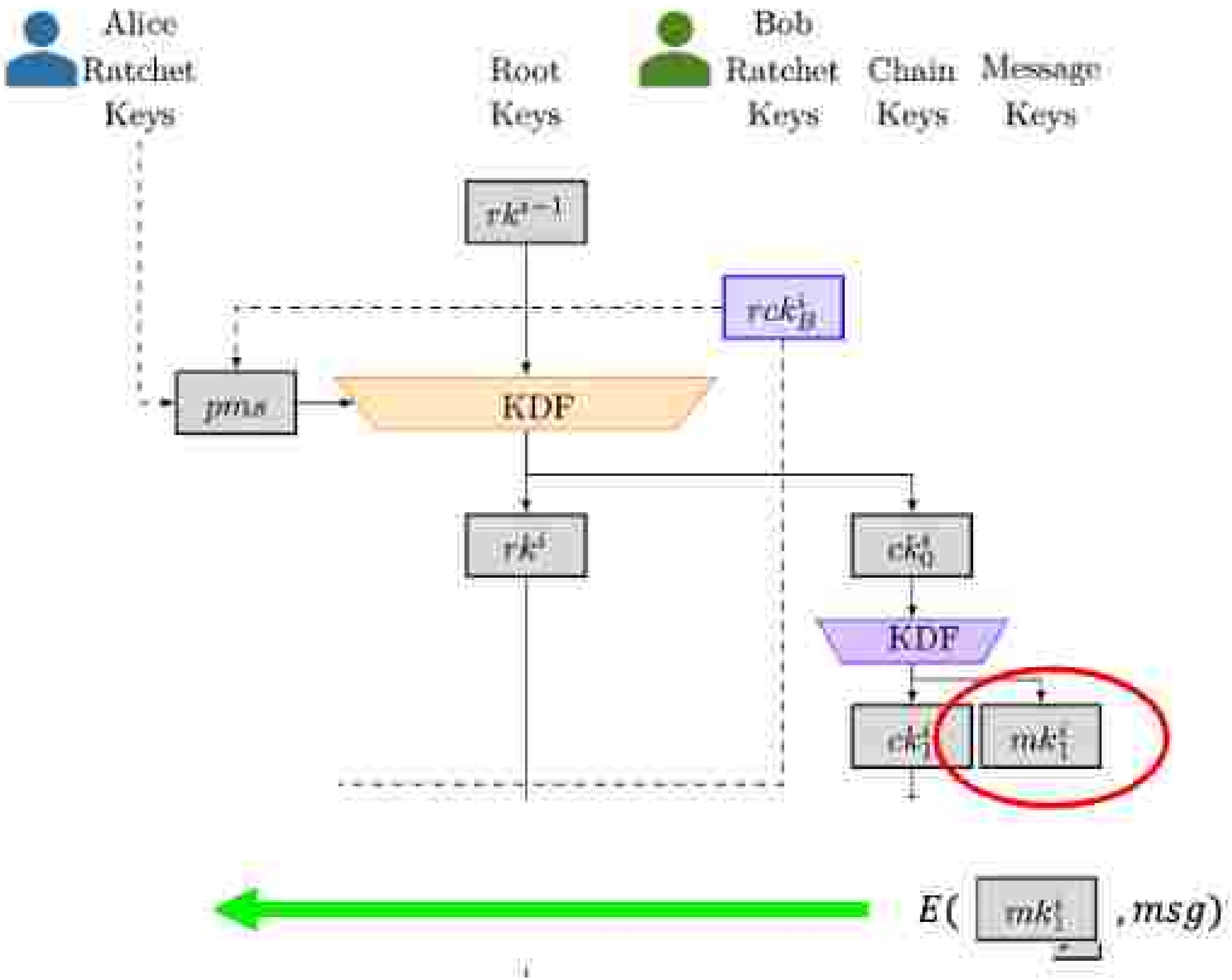
Ratcheting


Alice
Ratchet
Keys
Root
Keys
Bob
Ratchet
Keys
Chain
Keys
Message
Keys
rk^{i-1}
rck_B^i
pms
KDF
rk^i
ck_0^i
rck_A^{i+1}
pms
KDF

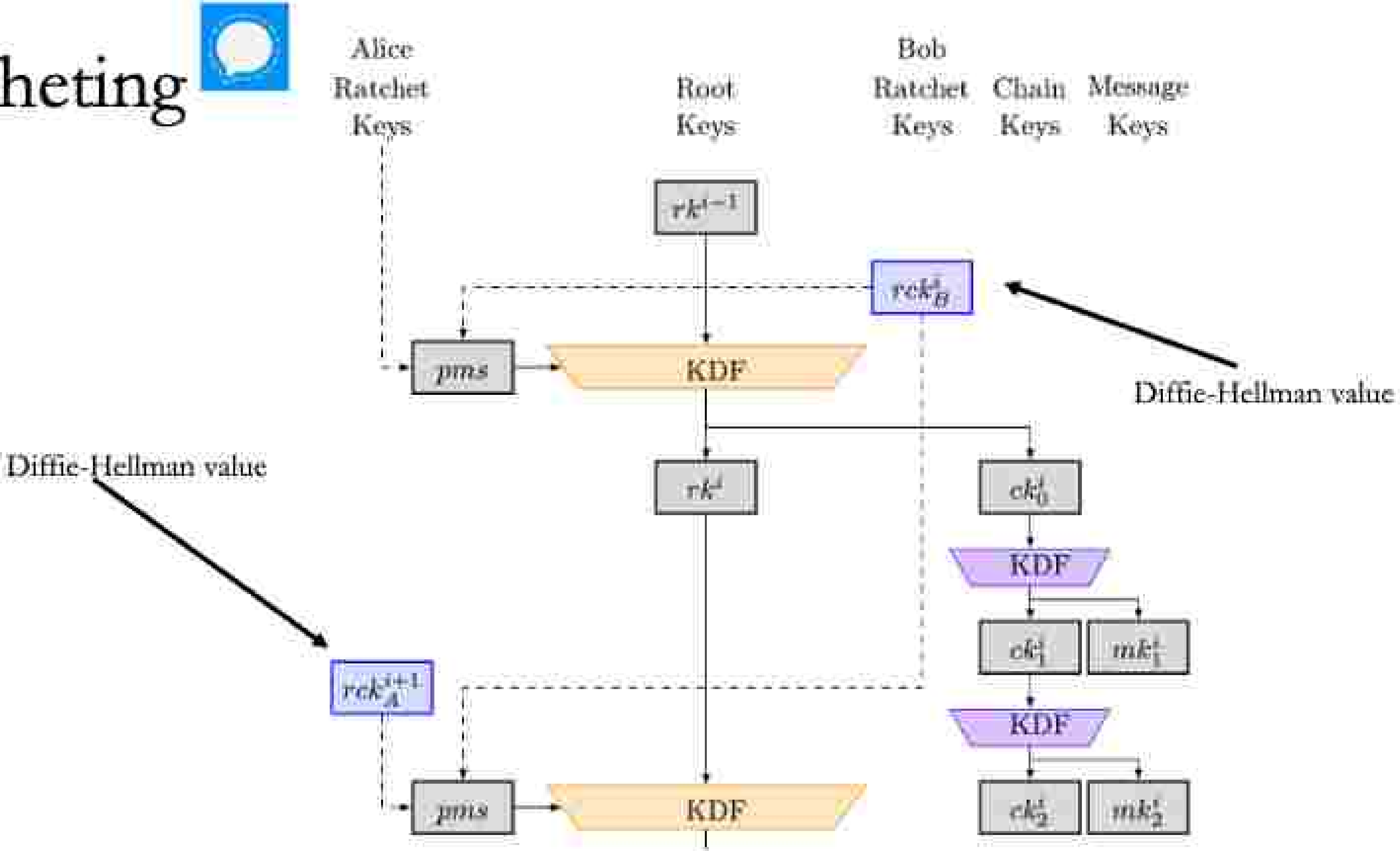
Ratcheting


Alice Ratchet Keys
Root Keys
Bob Ratchet Keys
Chain Keys
Message Keys
rk^{i-1}
rck_B^i
pms
KDF
rk^i
ck_0^i
KDF
ck_1^i
mk_1^i
rck_A^{i+1}
KDF
pms
KDF
ck_2^i
mk_2^i

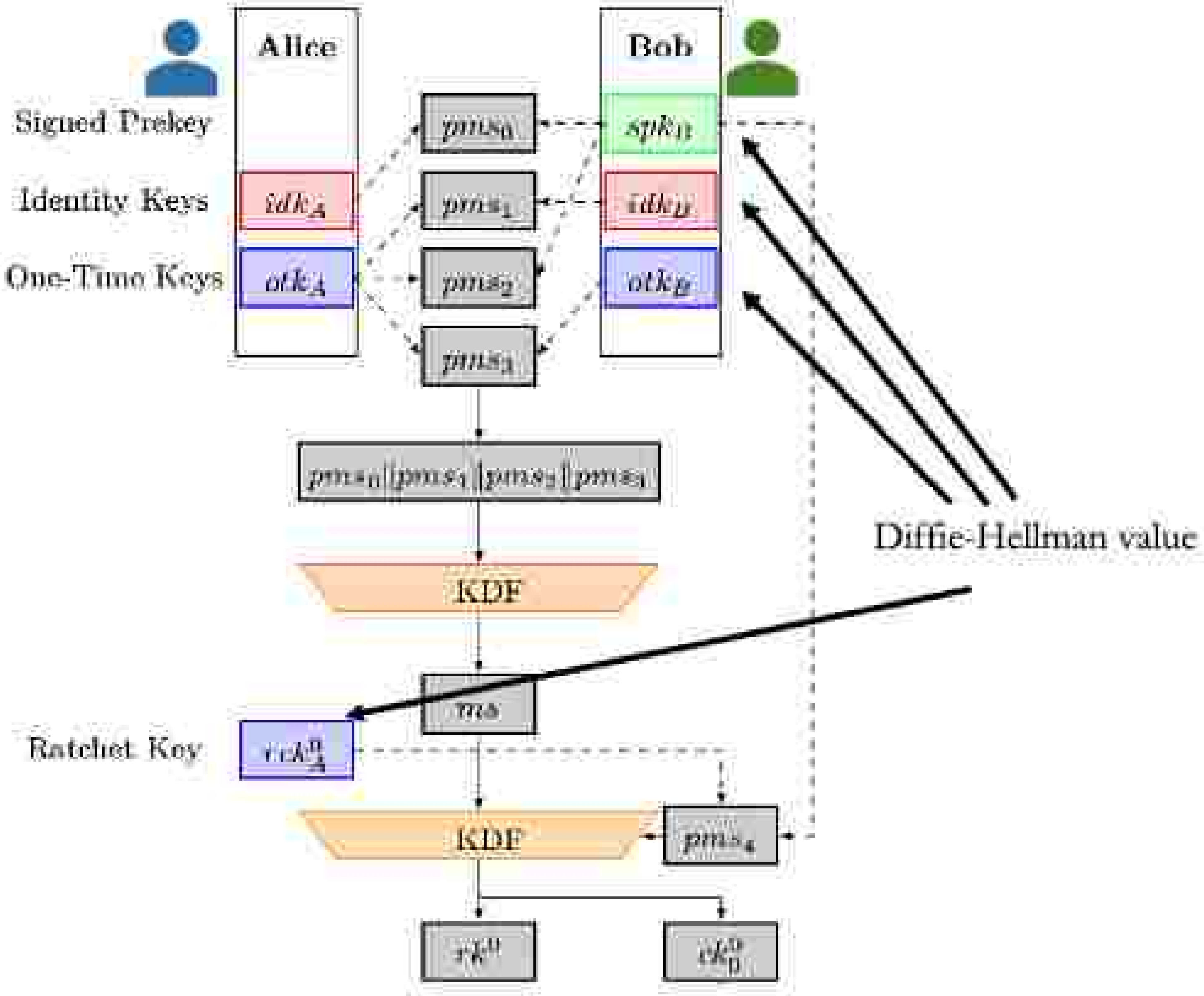
Ratcheting


Alice Ratchet Keys
Root Keys
Bob Ratchet Keys
Chain Keys
Message Keys
rk^{i-1}
rck^i_B
pms
KDF
rk^i
ck^i_0
KDF
ck^i_1
mk^i_1
$E(mk^i_1, msg)$

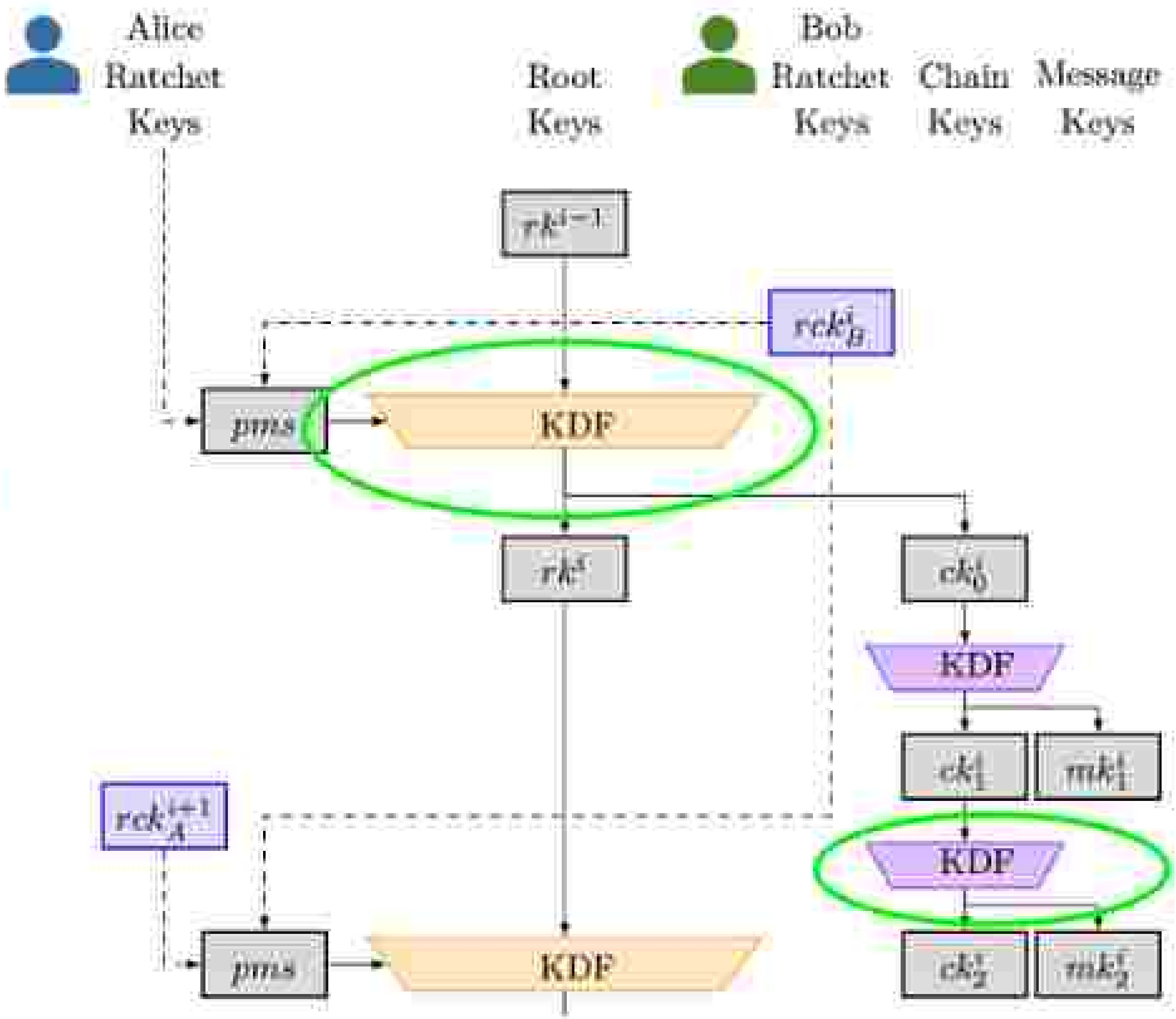
Ratcheting


Alice Ratchet Keys
Root Keys
Bob Ratchet Keys
Chain Keys
Message Keys
rk^{i-1}
rck_B^i
Diffie-Hellman value
pms
KDF
rk^i
ck_0^i
Diffie-Hellman value
KDF
ck_1^i
mk_1^i
rck_A^{i+1}
KDF
pms
KDF
ck_2^i
mk_2^i

Setup (handshake)

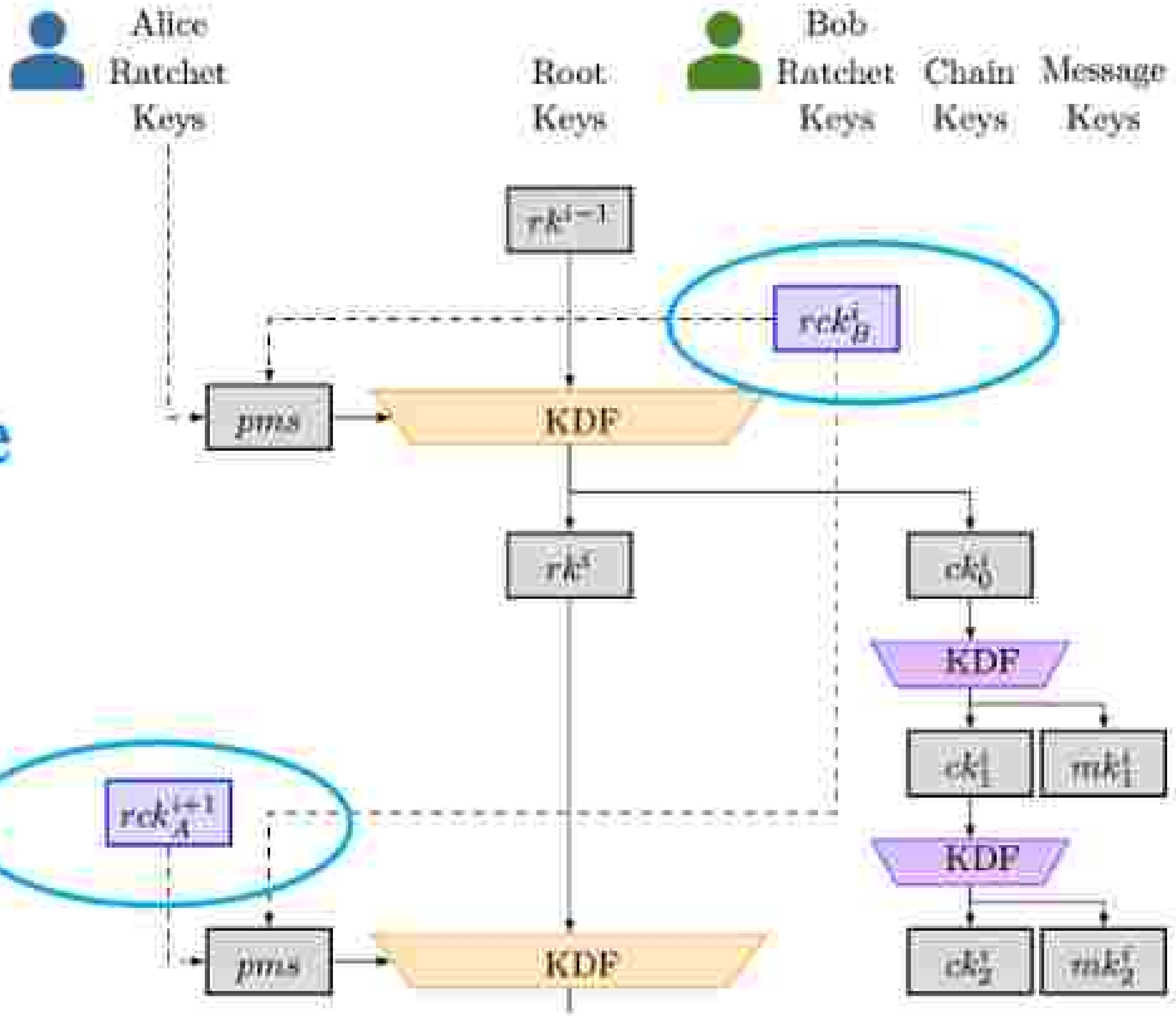


Perfect Forward Secrecy



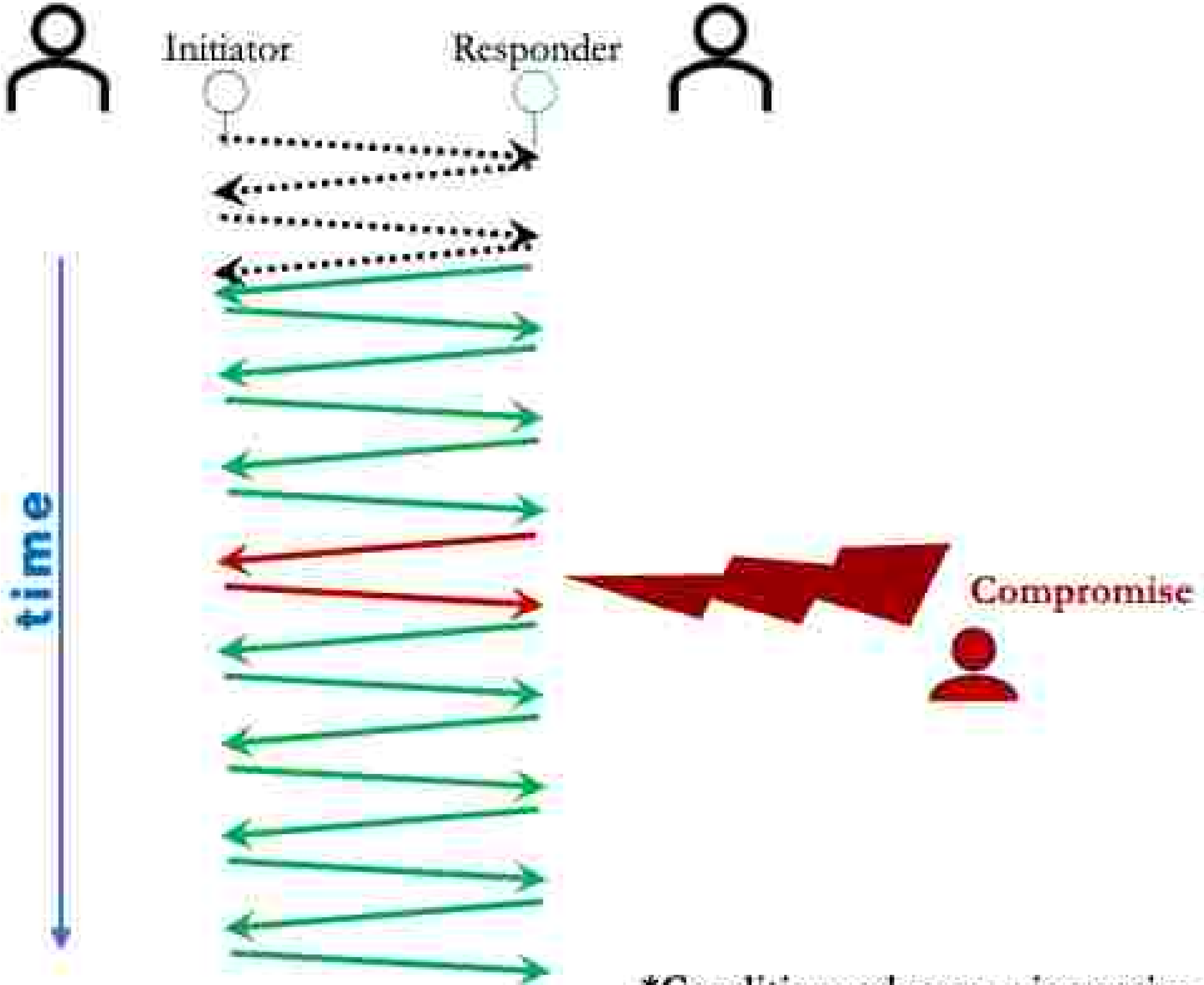
Post-Compromise Security

**Attacker must be passive for an epoch to allow PCS healing*



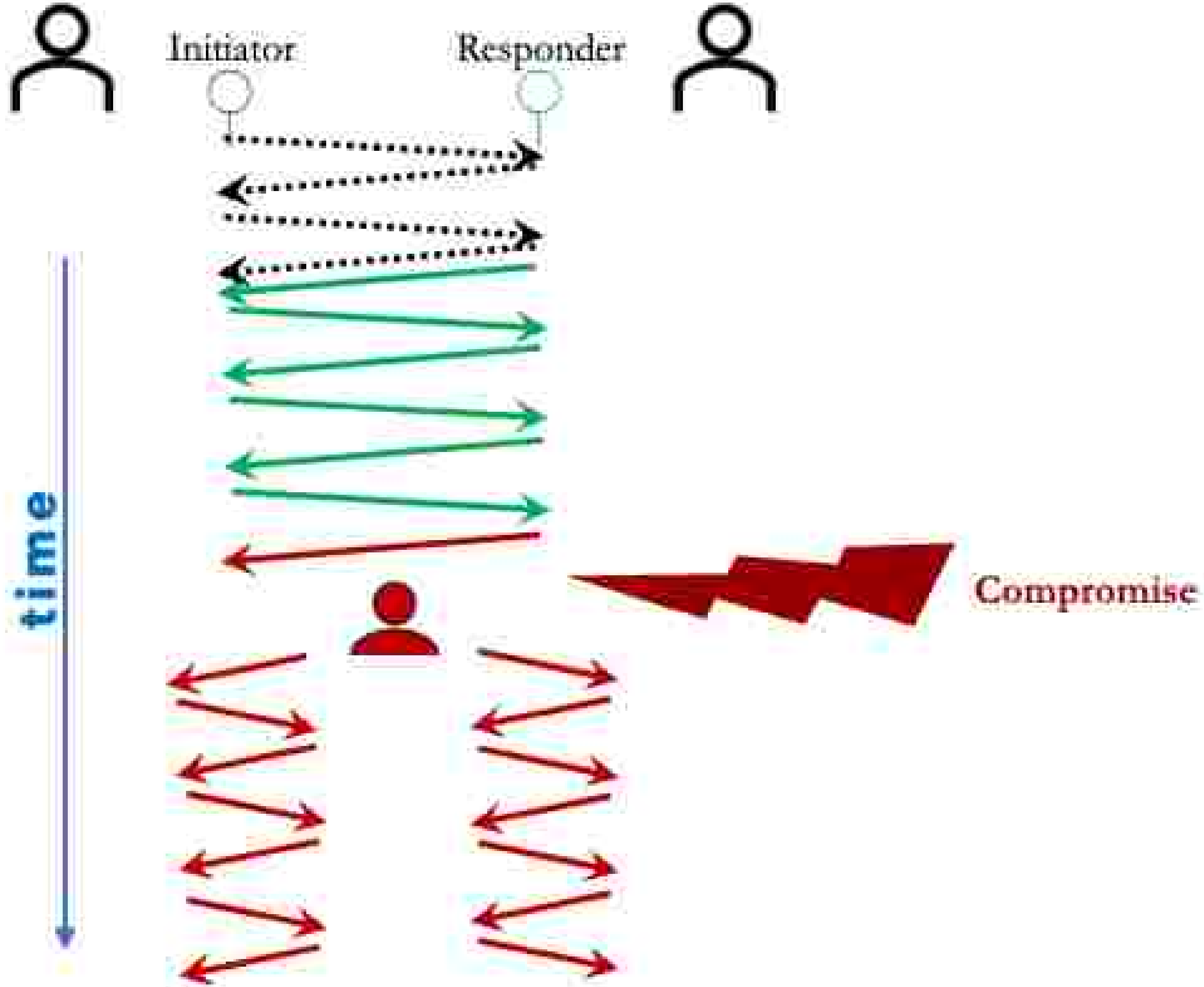
Forward and Post-Compromise Secure End-to-End Messaging

~~Secure End-to-End Messaging~~

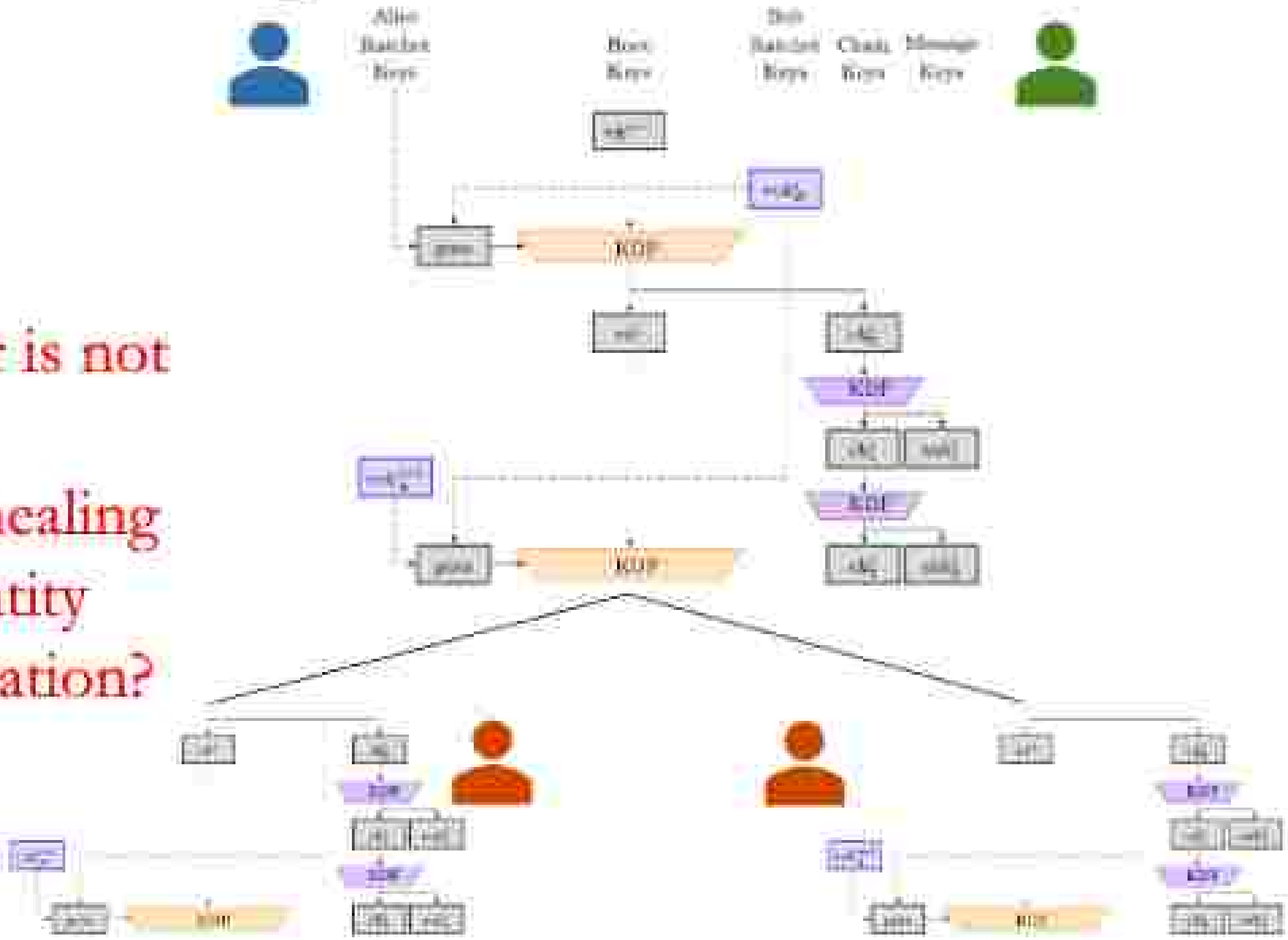


*Condition: adversary is passive for one epoch

Active Attacker is catastrophic to security



Ratcheting – Compromise?

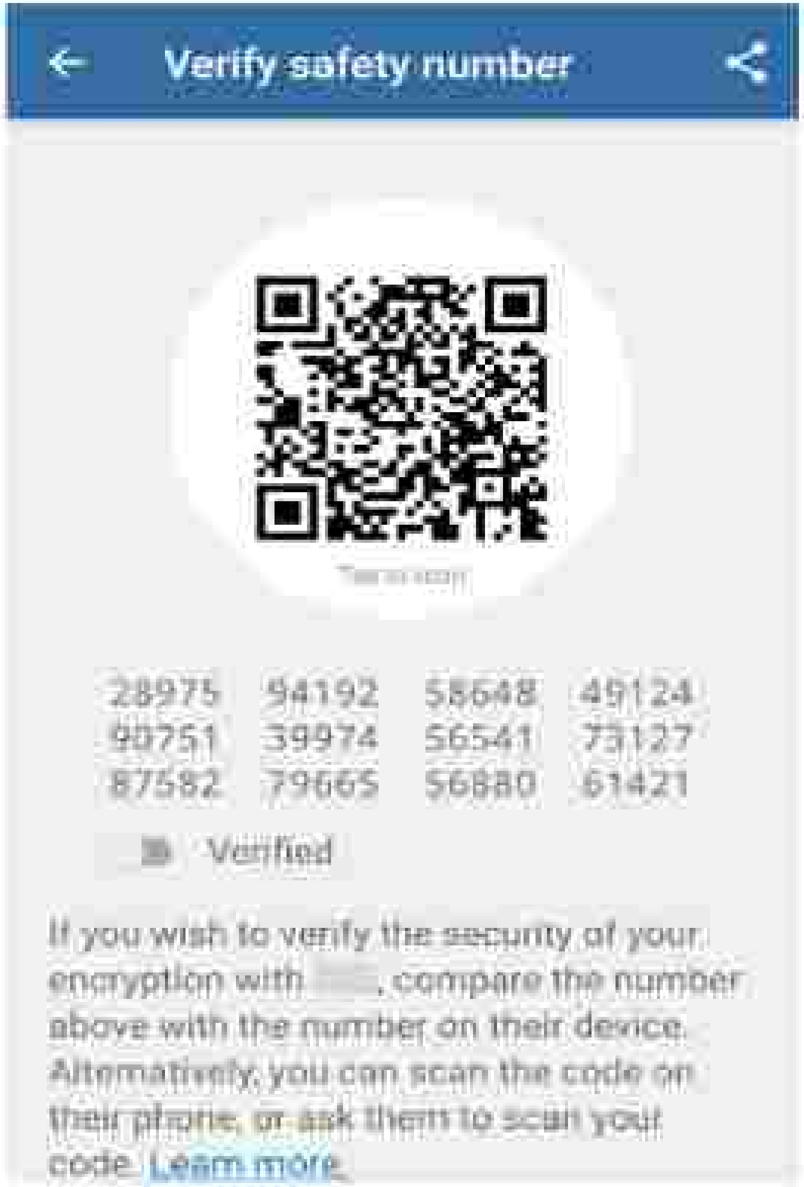


*If attacker is not passive:

- ➢no PCS healing
- ➢breaks entity authentication?

Is that not a break in entity authentication?

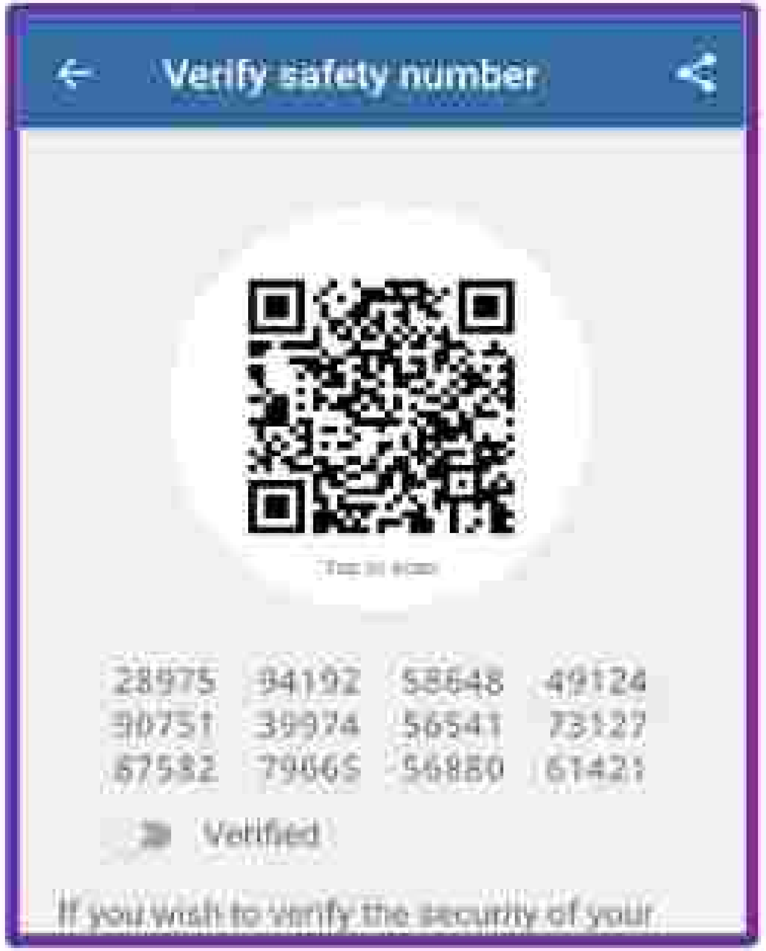
Authentication In Signal



28975 94192 58648 49124
90751 39974 56541 73127
87582 79665 56880 61421

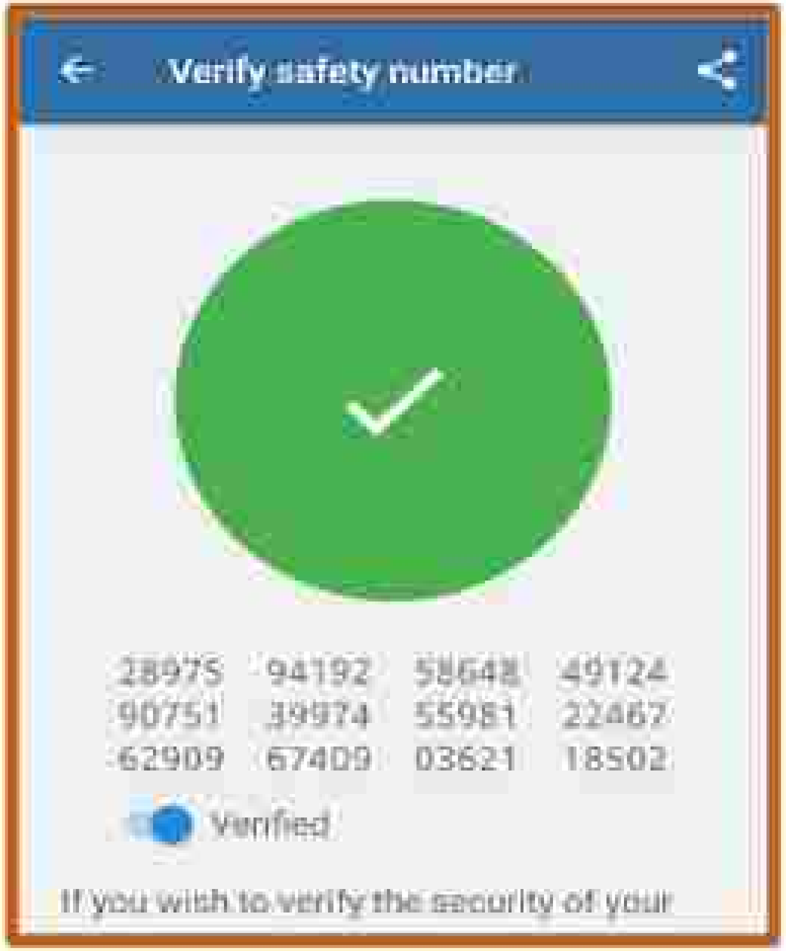
If you wish to verify the security of your encryption with [redacted], compare the number above with the number on their device. Alternatively, you can scan the code on their phone, or ask them to scan your code. Learn more.

One-way QR-code / Numeric Authentication



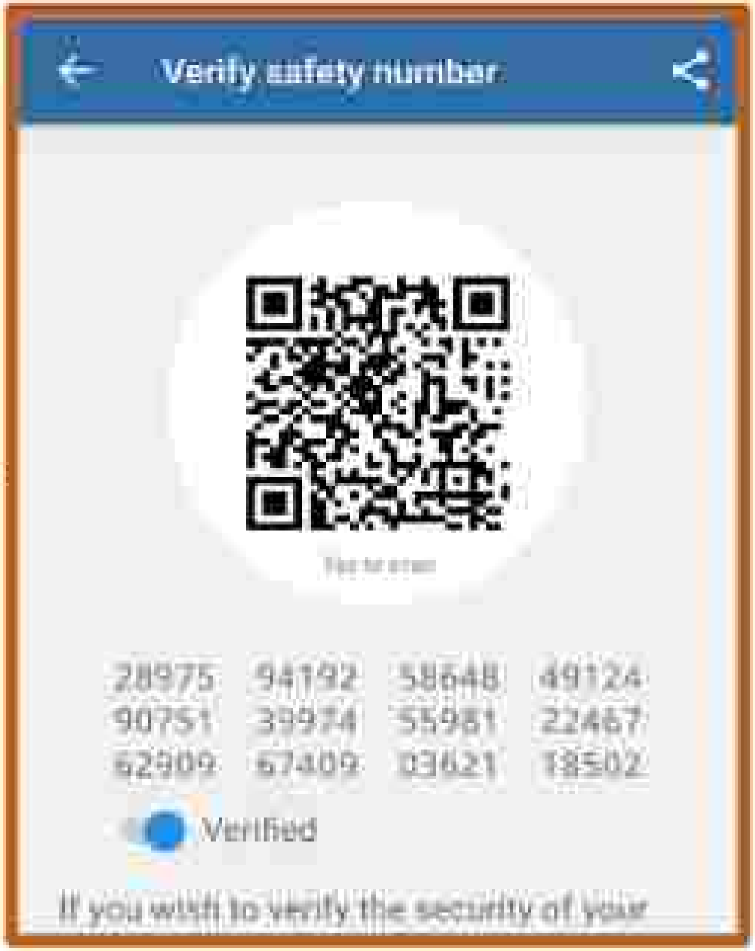
28975 94192 58648 49124
90751 39974 56541 73127
67582 79665 56880 61421

If you wish to verify the security of your



28975 94192 58648 49124
90751 39974 55981 22467
62909 67409 03621 18502

If you wish to verify the security of your

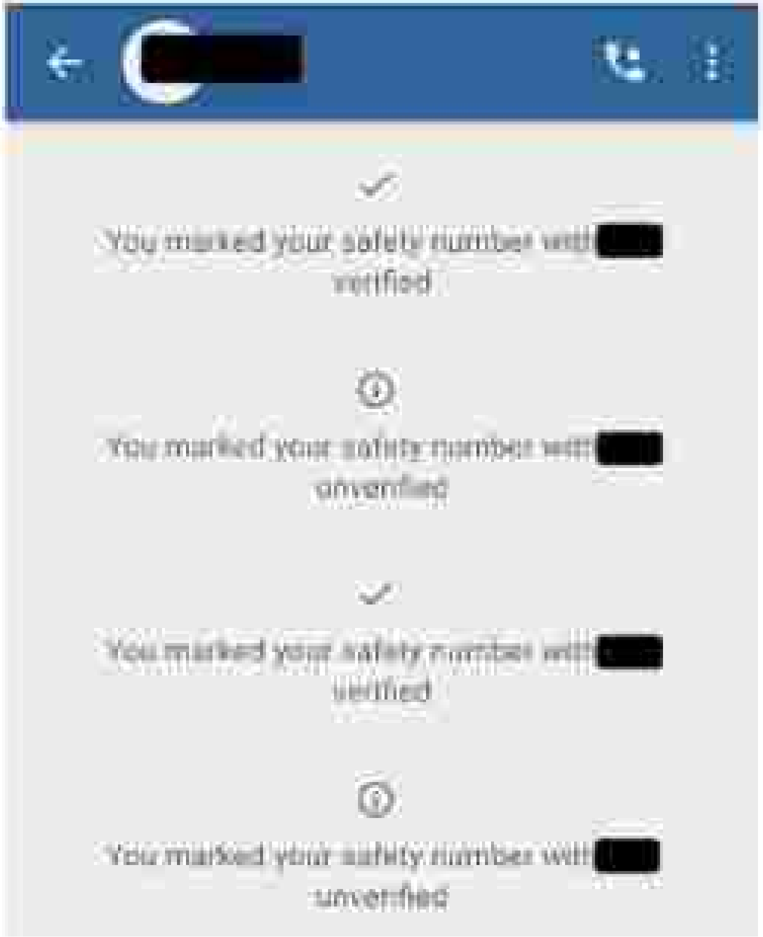


28975 94192 58648 49124
90751 39974 55981 22467
62909 67409 03621 18502

If you wish to verify the security of your

Signal Issues

Static public keys



[redacted]

✓

You marked your safety number with [redacted] verified

ⓘ

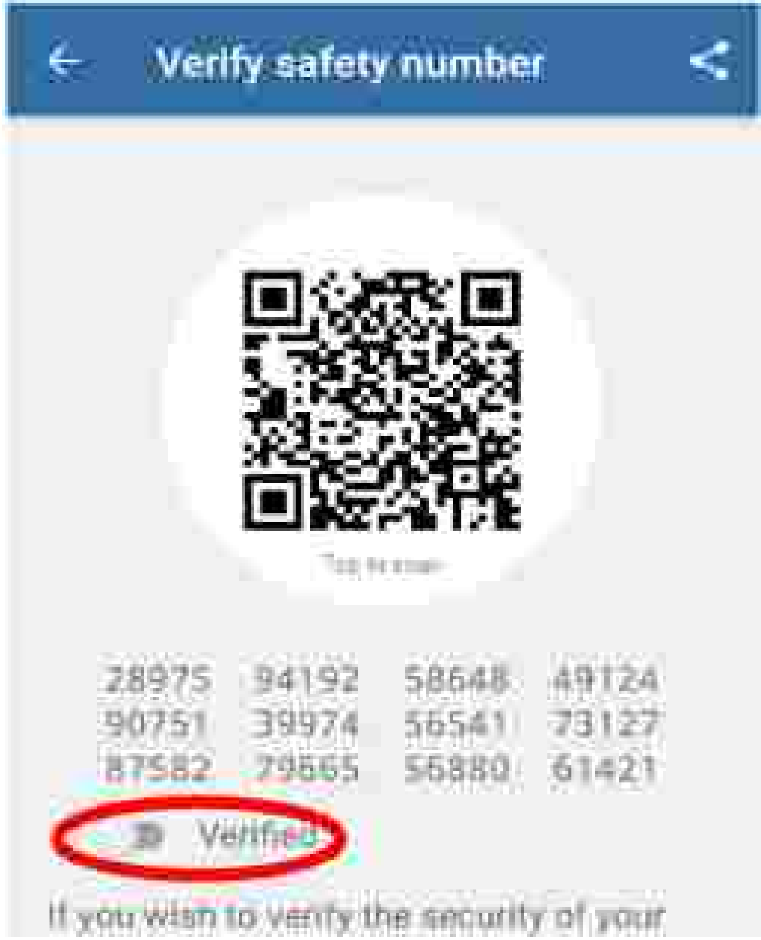
You marked your safety number with [redacted] unverified

✓

You marked your safety number with [redacted] verified

ⓘ

You marked your safety number with [redacted] unverified



Verify safety number

Tap to scan

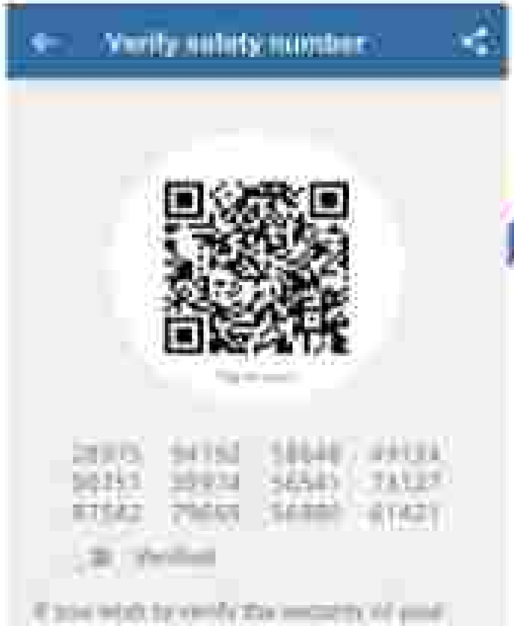
```
28975   94192   58648   49124
90751   39974   56541   73127
87582   79665   56880   61421
```

● Verified

If you wish to verify the security of your

Signal Issues

Static public keys



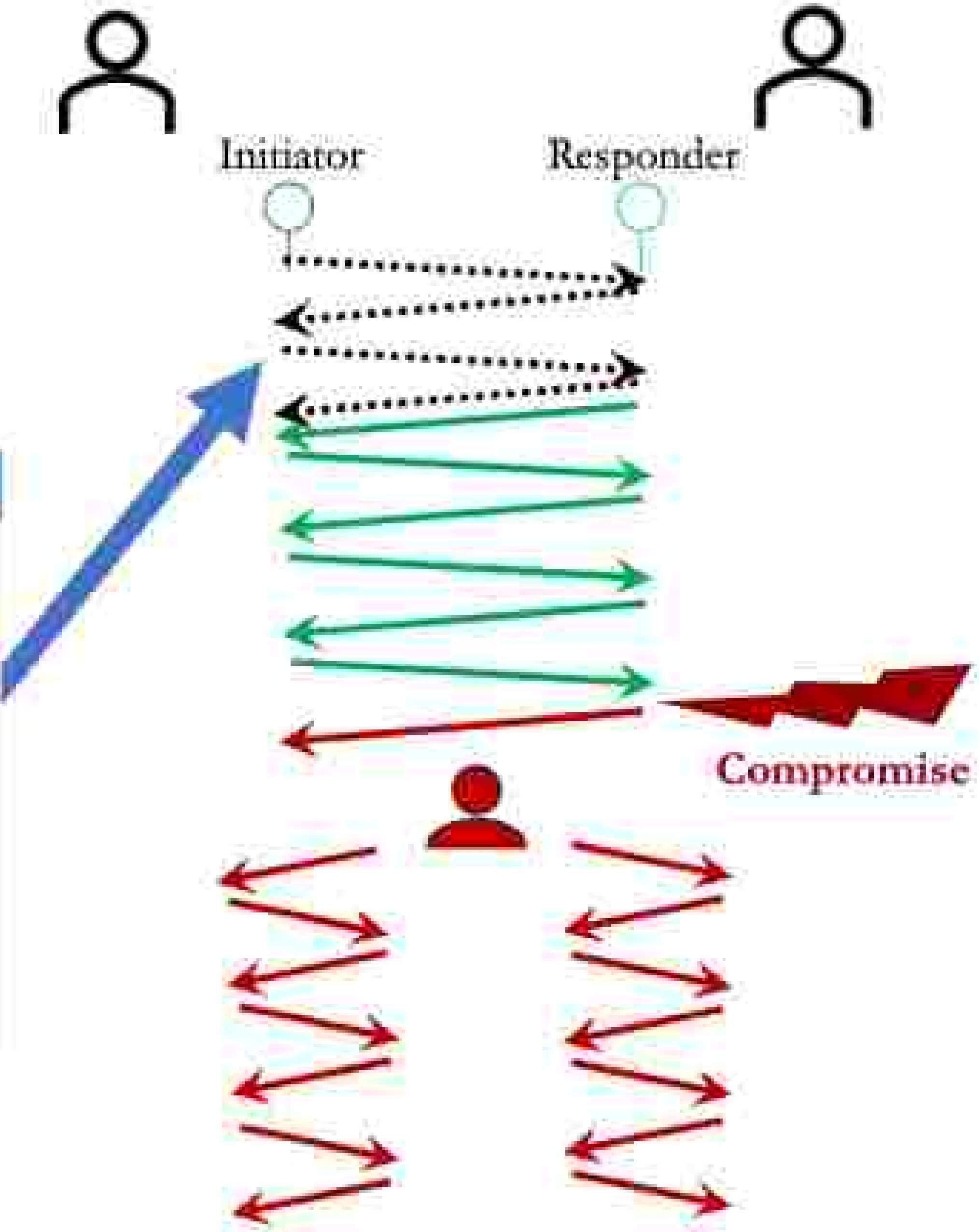
Verify safety number

Tap to scan

28975 94152 58646 49134
98151 35934 56545 73127
81542 79695 56880 61421

Verified

If you wish to verify the security of your

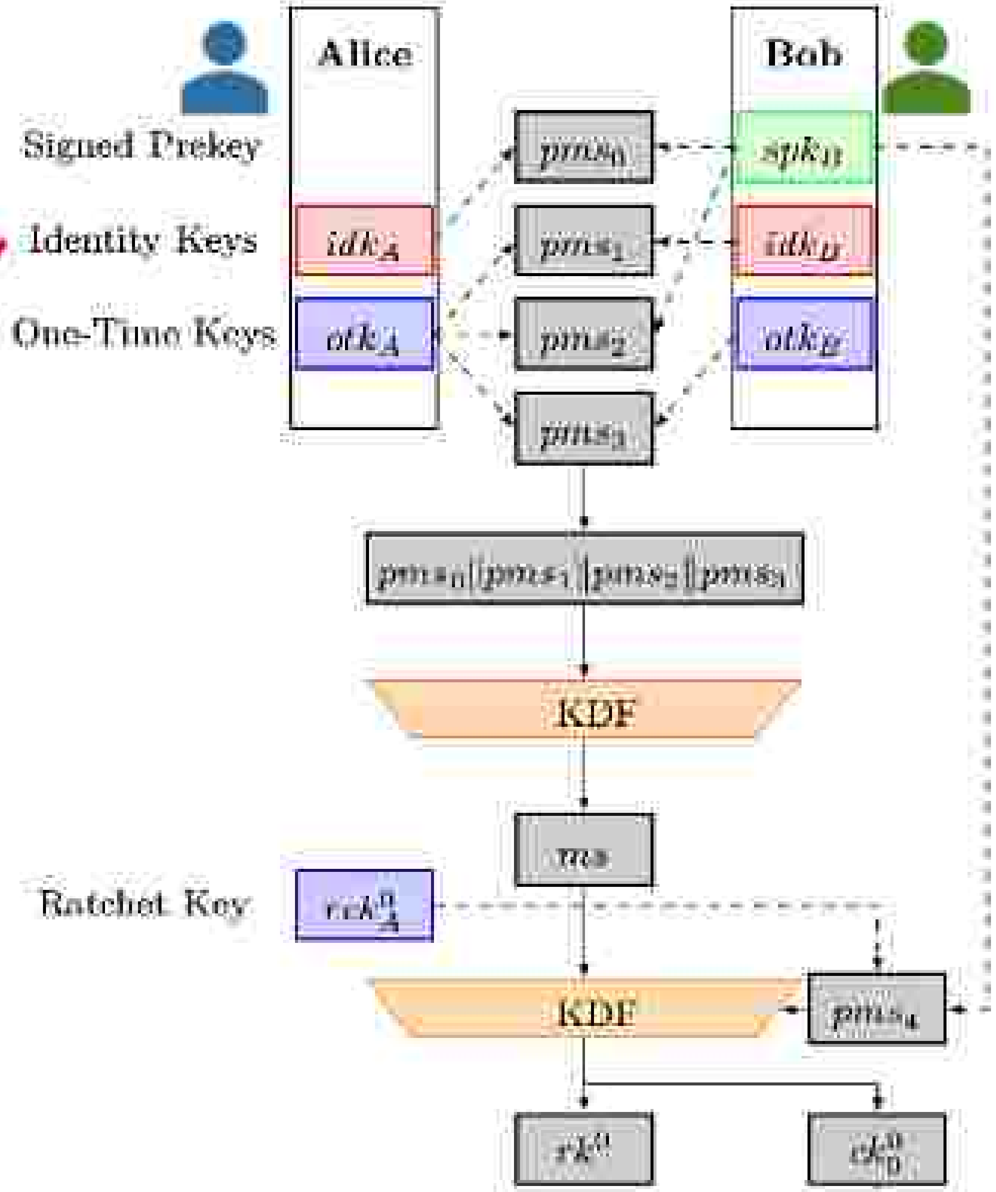


Signal Issues

Static public keys

$\texttt{local\_fprint} = H(0\|\texttt{fvers}\|idpk_A\|\mathrm{ID}_A\|idpk_A)$

$\texttt{remote\_fprint} = H(0\|\texttt{fvers}\|idpk_B\|\mathrm{ID}_B\|idpk_B)$

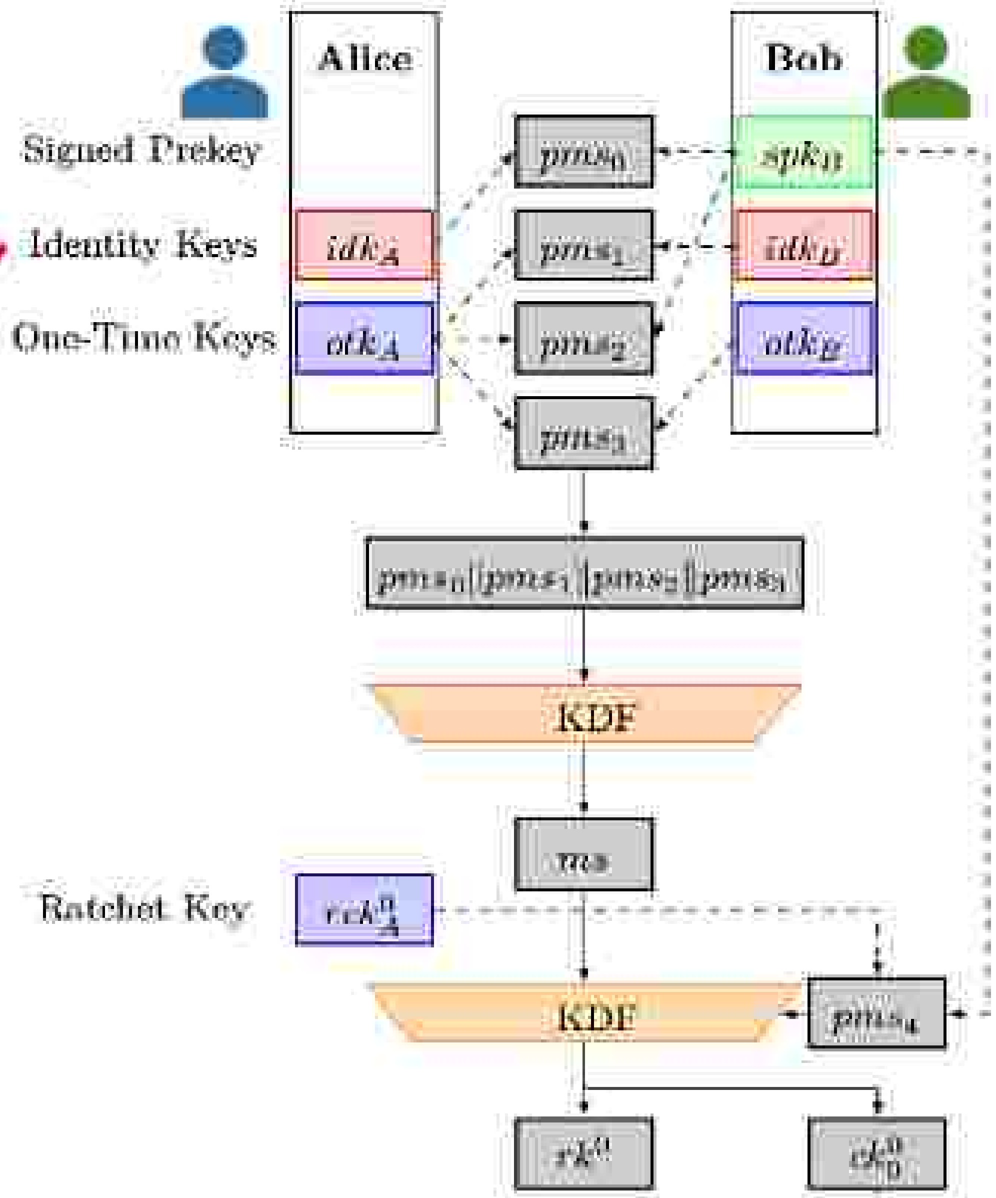


Signal Issues

Static public keys

$$\texttt{local\_fprint} = H(0\|\texttt{fvers}\|idpk_A\|\mathrm{ID}_A\|idpk_A)$$

$$\texttt{remote\_fprint} = H(0\|\texttt{fvers}\|idpk_B\|\mathrm{ID}_B\|idpk_B)$$

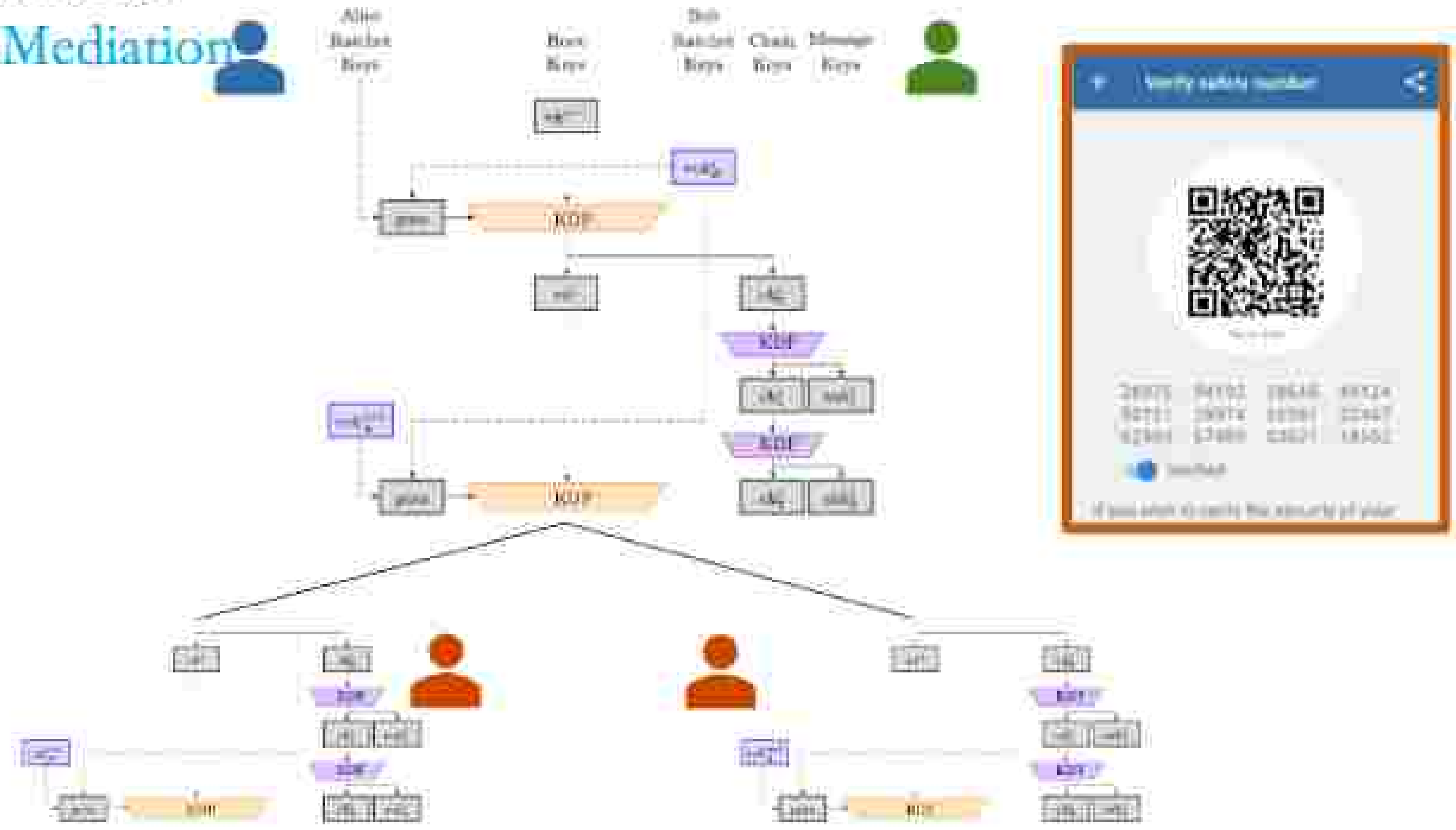


Based on public information only

No link to Signal protocol

Signal Issues

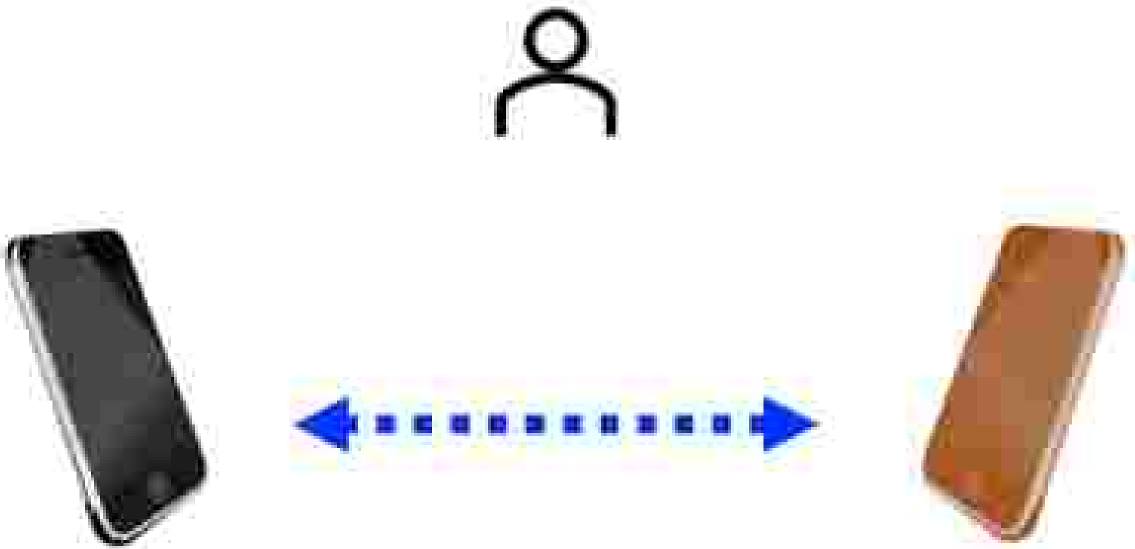
Weak User Mediation



If you wish to verify the security of your

User-to-Device: Real Life is Complex

Weak User Mediation

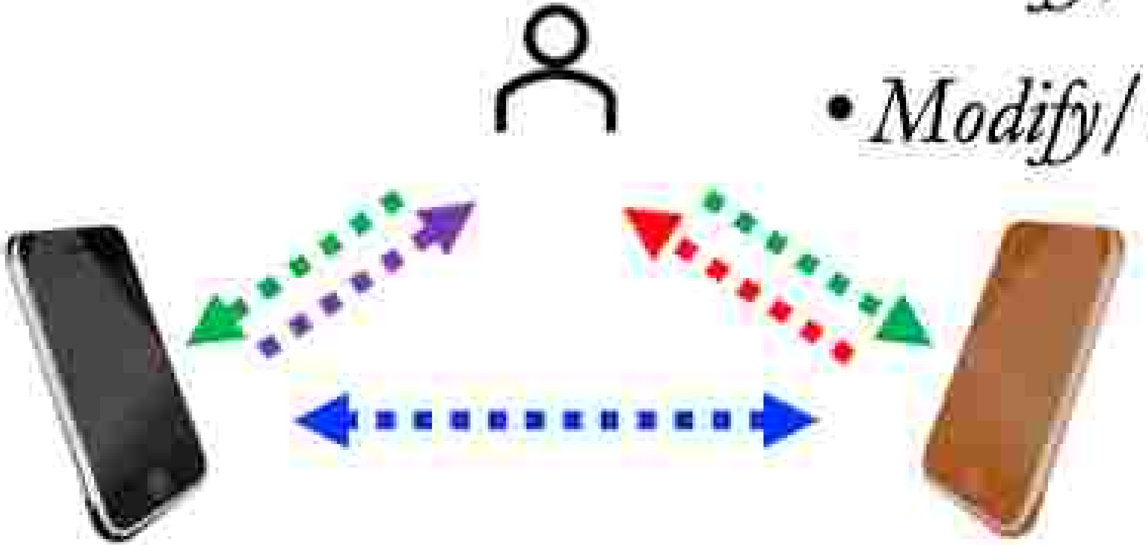


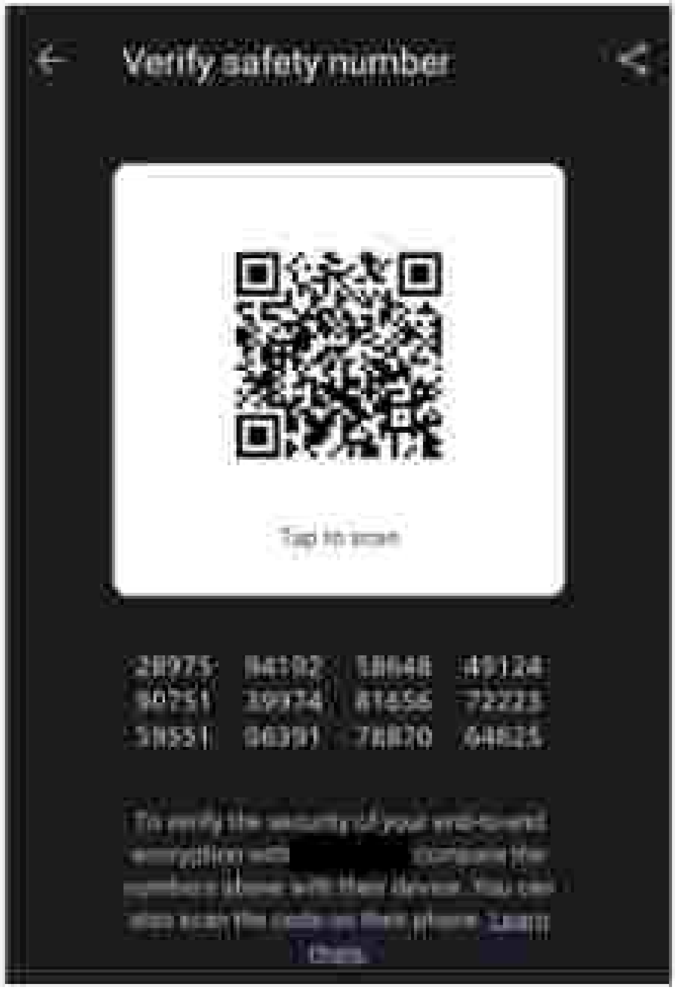
User-to-Device: Real Life is Complex

Weak User Mediation

Adversary allowed: Read, Replay, Delete

- *Modify/ create User-to-Device messages?*
- *Modify/ create Device-to-User messages?*
- *Modify/ create Device-to-Device messages?*

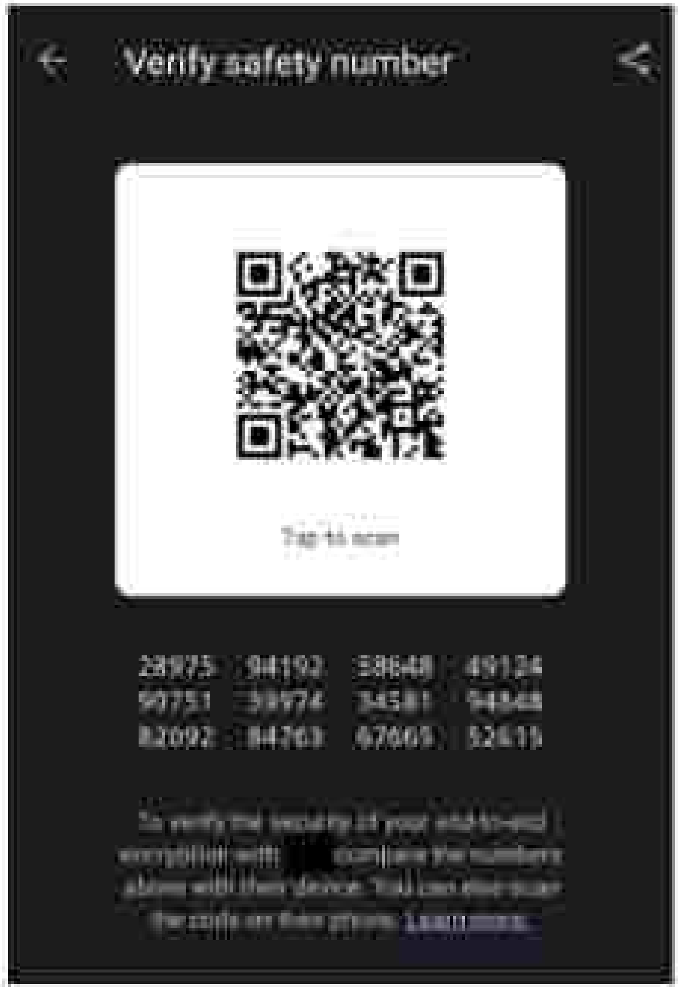




Verify safety number

```
28975   94192   58648   49124
99751   39974   81456   72223
59551   06391   78870   64625
```

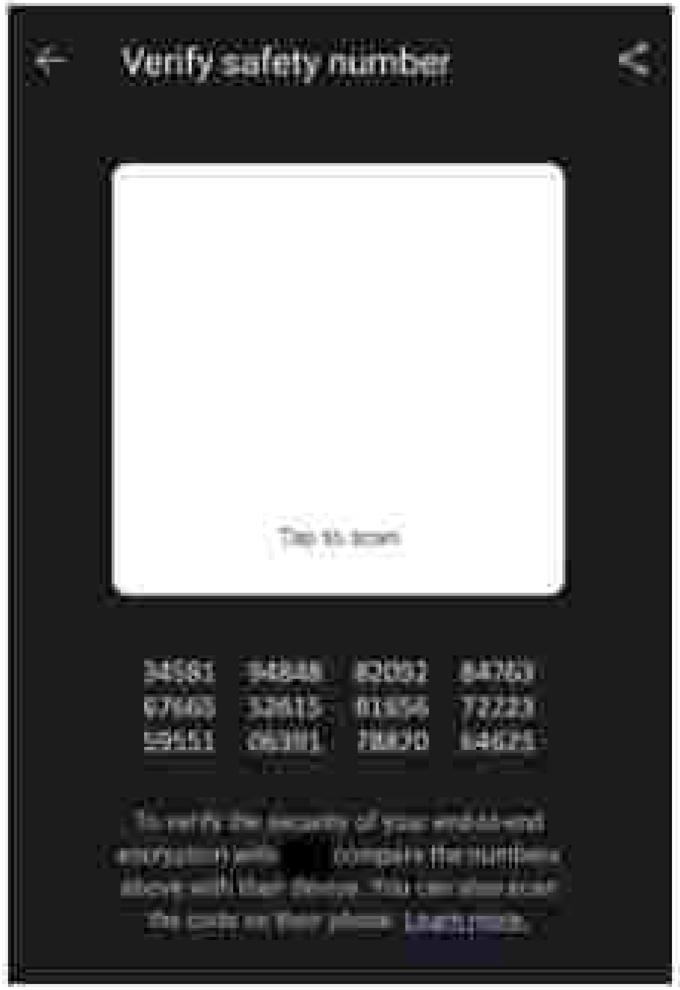
To verify the security of your end-to-end encryption with [redacted] compare the numbers above with their device. You can also scan the code on their phone. Learn more.



Verify safety number

```
28975   94192   58648   49124
99751   39974   34581   94848
82092   84763   67665   52615
```

To verify the security of your end-to-end encryption with [redacted] compare the numbers above with their device. You can also scan the code on their phone. Learn more.



Verify safety number

```
34581   94848   82092   84763
67665   52615   81456   72223
59551   06391   78870   64625
```

To verify the security of your end-to-end encryption with [redacted] compare the numbers above with their device. You can also scan the code on their phone. Learn more.

$$\texttt{local\_fprint} = H(0\|\texttt{fvers}\|idpk_A\|\mathrm{ID}_A\|idpk_A)$$

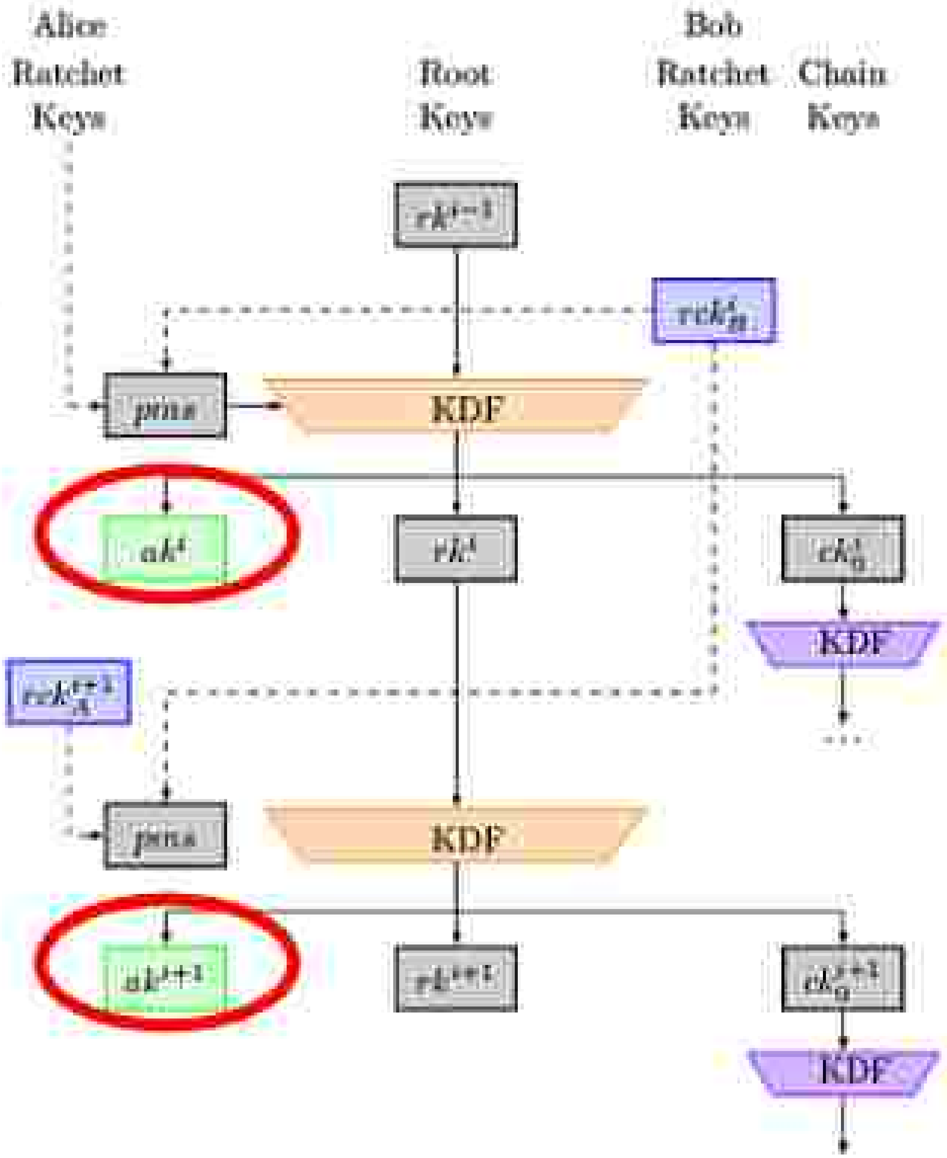
$$\texttt{remote\_fprint} = H(0\|\texttt{fvers}\|idpk_B\|\mathrm{ID}_B\|idpk_B)$$

Fixing authentication:

1) accounting for user interaction

2) detection of active man-in-the-middle attack

Modified Device-to-User Signal Authentication (MoDUSA)



New QR-code computation:

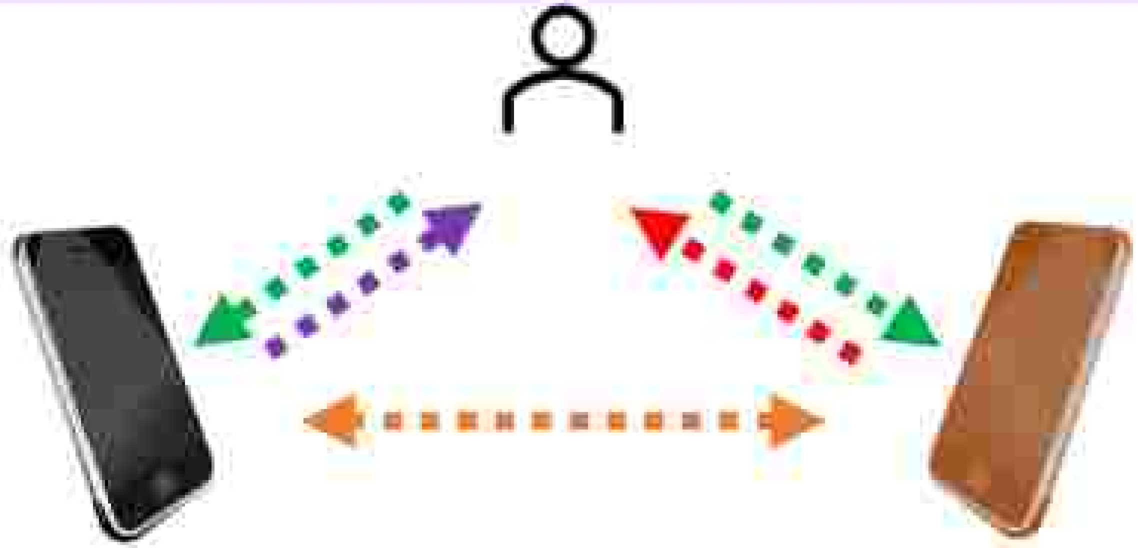
$$\texttt{fprint}^{i-1} = \mathsf{HMAC}(ak^{i-1}, H^{i-1} \| \texttt{fvers} \| role)$$
$$\texttt{fprint}^{i} = \mathsf{HMAC}(ak^{i}, H^{i} \| \texttt{fvers} \| role)$$

*Session specific

*Asynchronicity in computation

User-to-Device: Real Life is Complex

Auth. Initiator I	Auth. Responder I'	CD Without E.	CD with E.	CU Without E.	CU With E.
Display match	Display match	✓	✓	✓	X
Display match	Scan match	✓	✓	X	X
Scan match	Display match	✓	✓	✓	X
Scan match	Scan match	✓	✓	✓	X
Display non-match	Scan non-match	✓	✓	X	X
Scan non-match	Display non-match	✓	✓	✓	✓
Scan non-match	Scan non-match	✓	✓	✓	✓

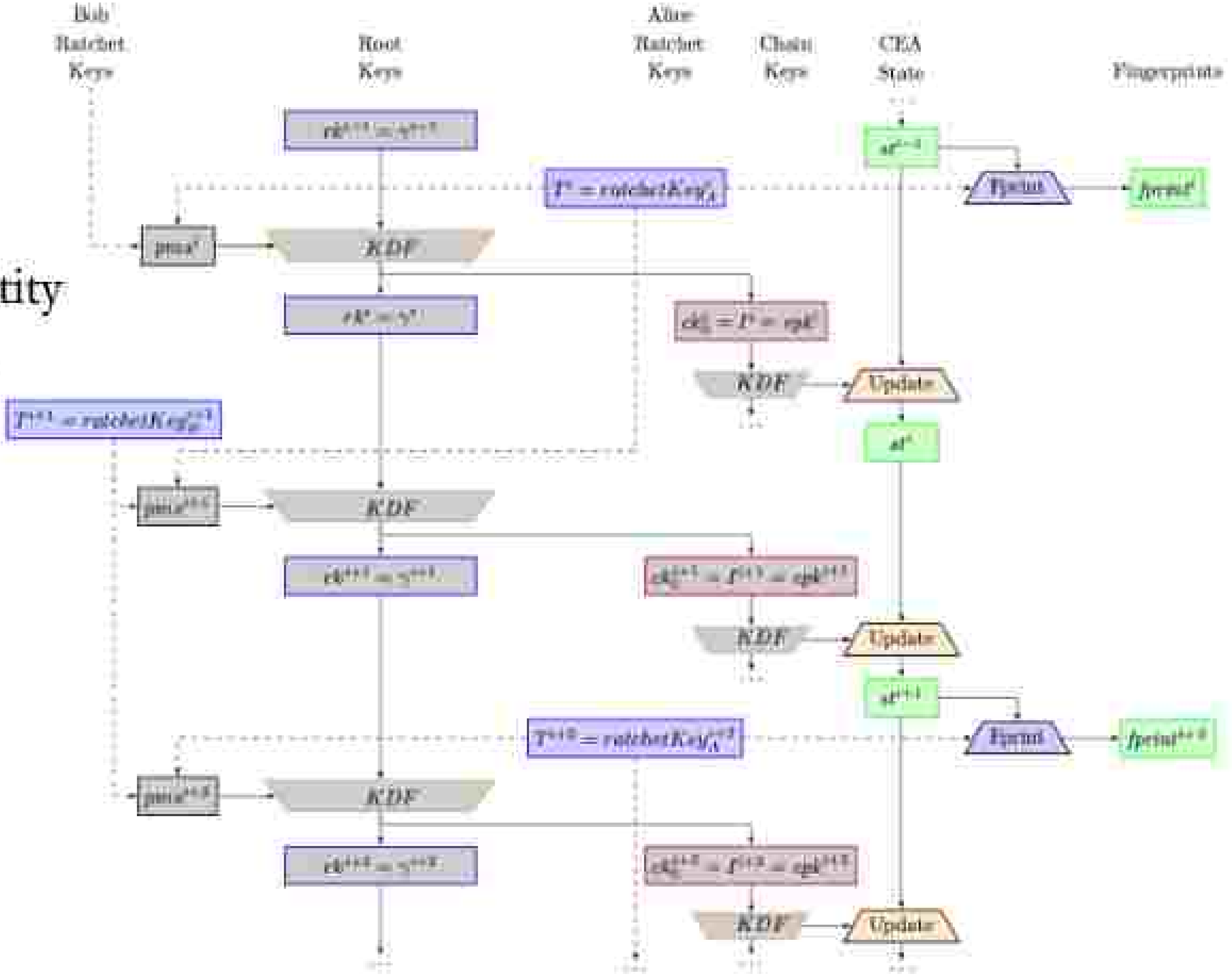


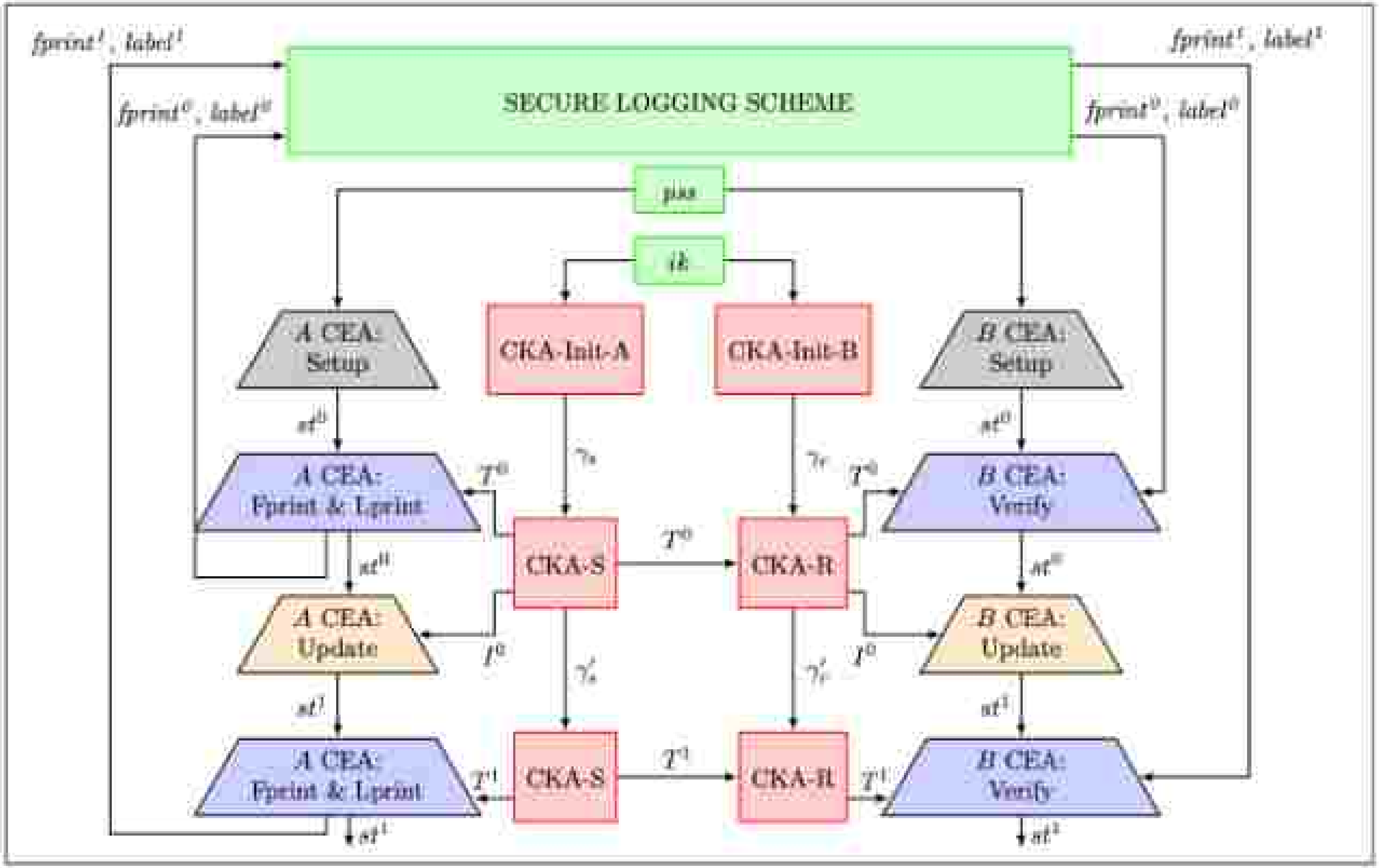
Great… but I never compared QR codes to begin with.

Great… but I never compared QR codes to begin with.

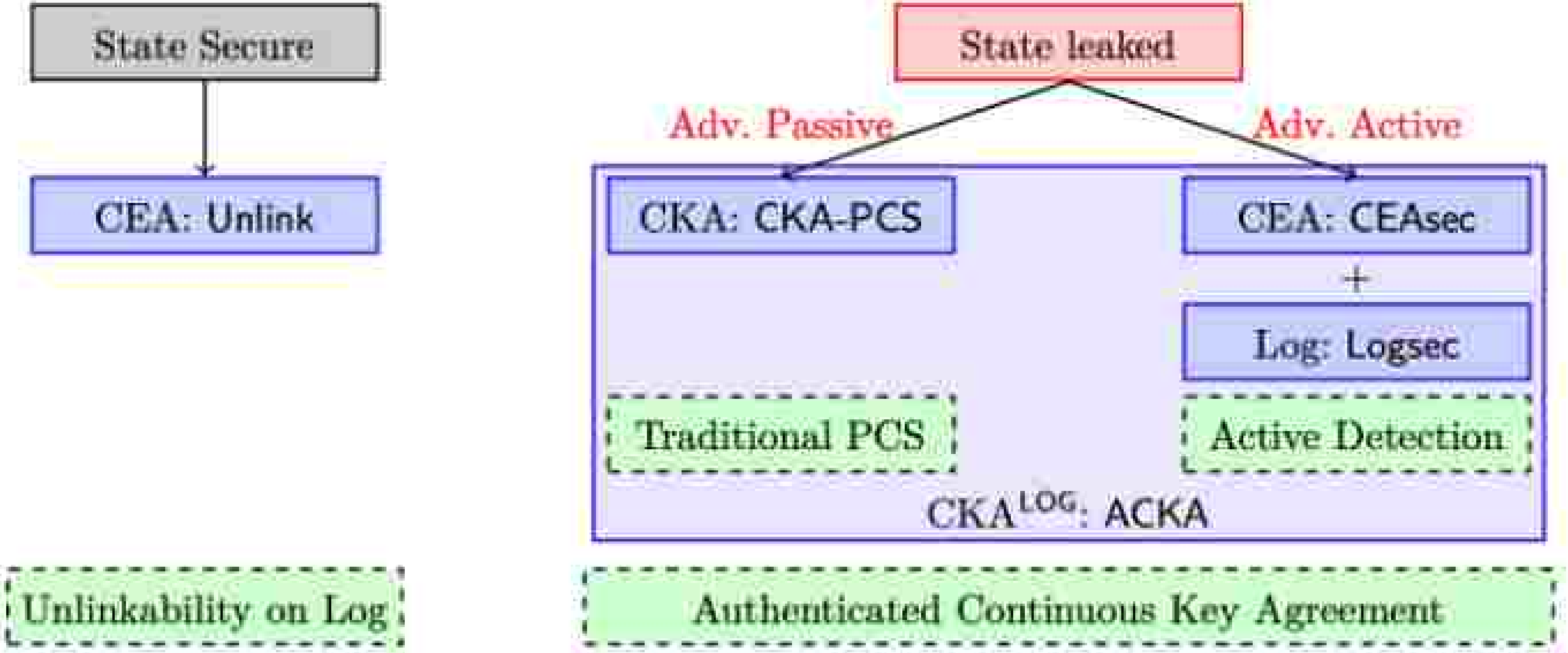
Can we automate ratcheted authentication to get man-in-the-middle detection without relying on human users?

CEA: Continuous Entity Authentication



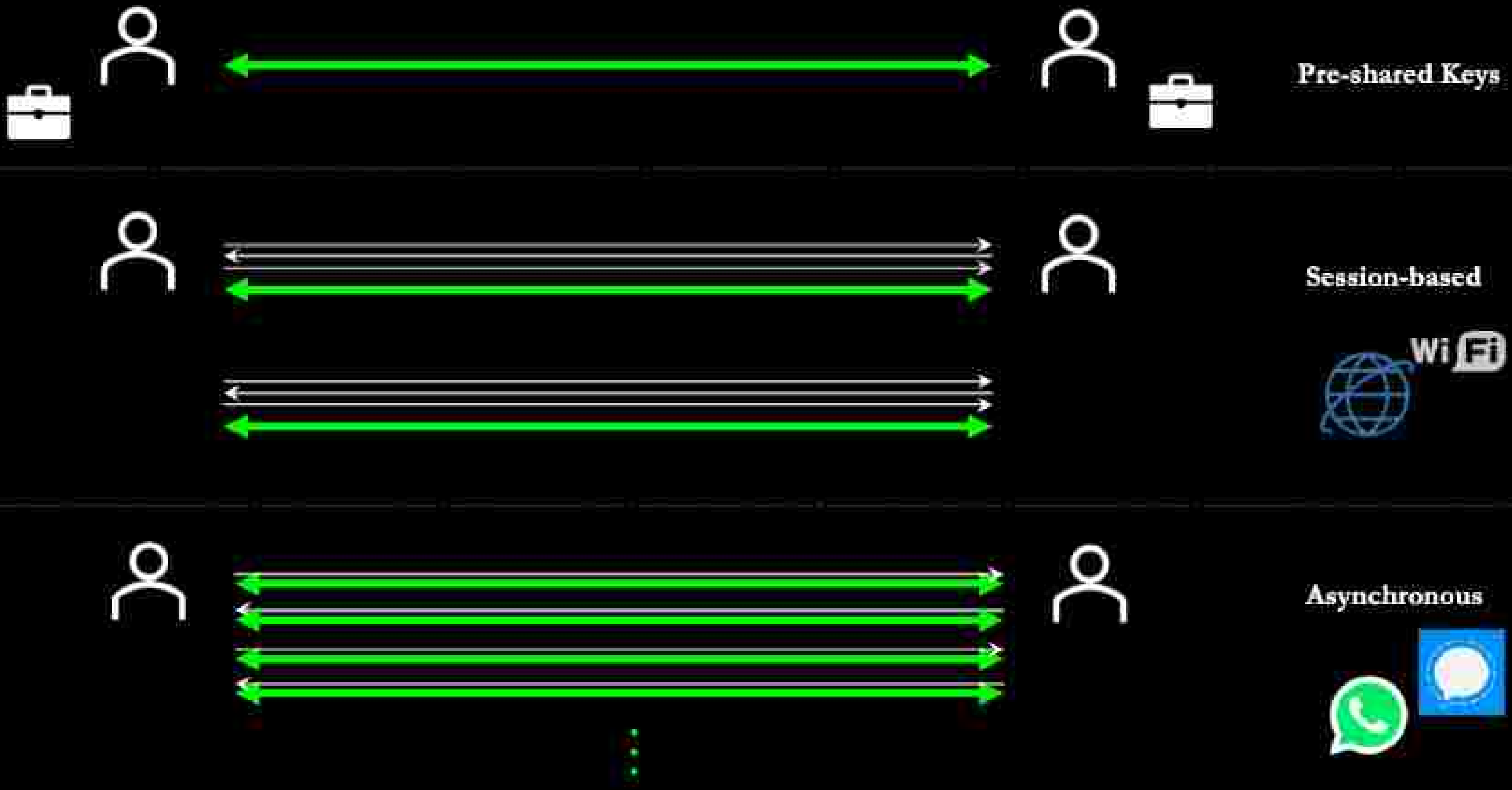

$fprint^1$, $label^1$
$fprint^0$, $label^0$
SECURE LOGGING SCHEME
$fprint^1$, $label^1$
$fprint^0$, $label^0$
pks
ik
A CEA: Setup
CKA-Init-A
CKA-Init-B
B CEA: Setup
st^0
γ_s
γ_r
st^0
A CEA: Fprint & Lprint
T^0
T^0
B CEA: Verify
CKA-S
T^0
CKA-R
st^0
st^0
A CEA: Update
I^0
γ'_s
γ'_r
I^0
B CEA: Update
st^1
st^1
A CEA: Fprint & Lprint
T^1
CKA-S
T^1
CKA-R
T^1
B CEA: Verify
st^1
st^1

ACKA: Authenticated Continuous Key Agreement



Forward and Post-Compromise End-to-End Messaging with Man-in-the-Middle Detection

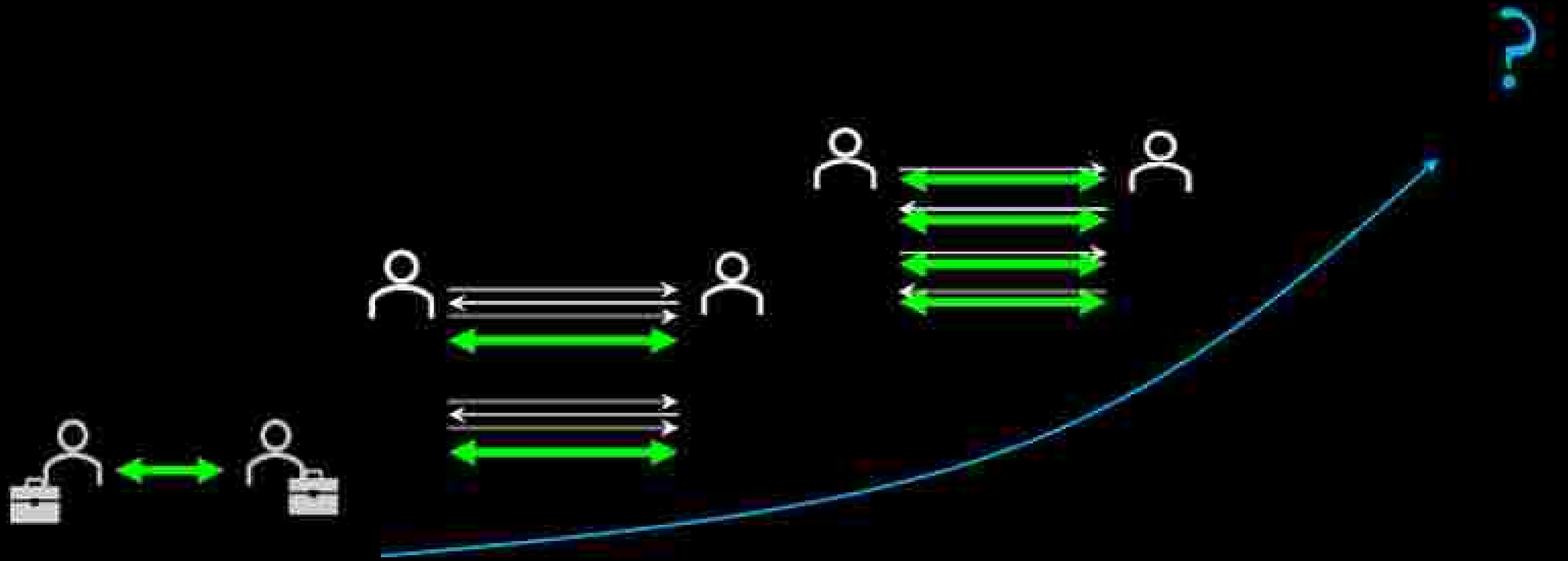
~~Forward and Post-Compromise Secure End-to-End Messaging~~

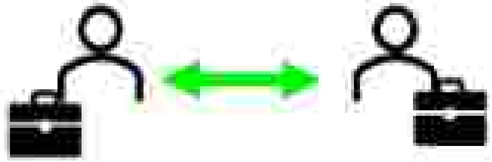


Pre-shared Keys

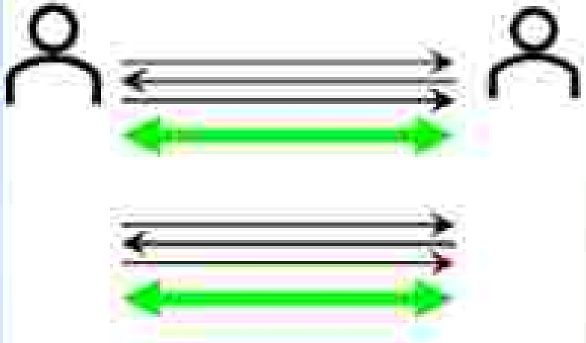
Session-based

Asynchronous

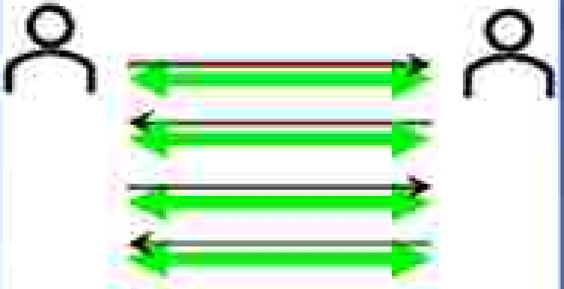




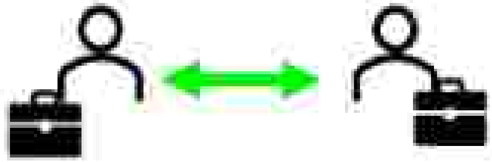
Pre-shared Keys



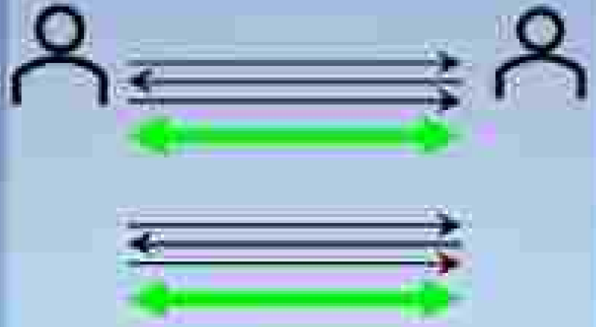
Session-based



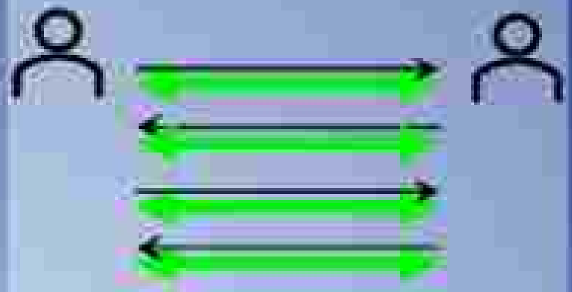
Asynchronous



Pre-shared Keys

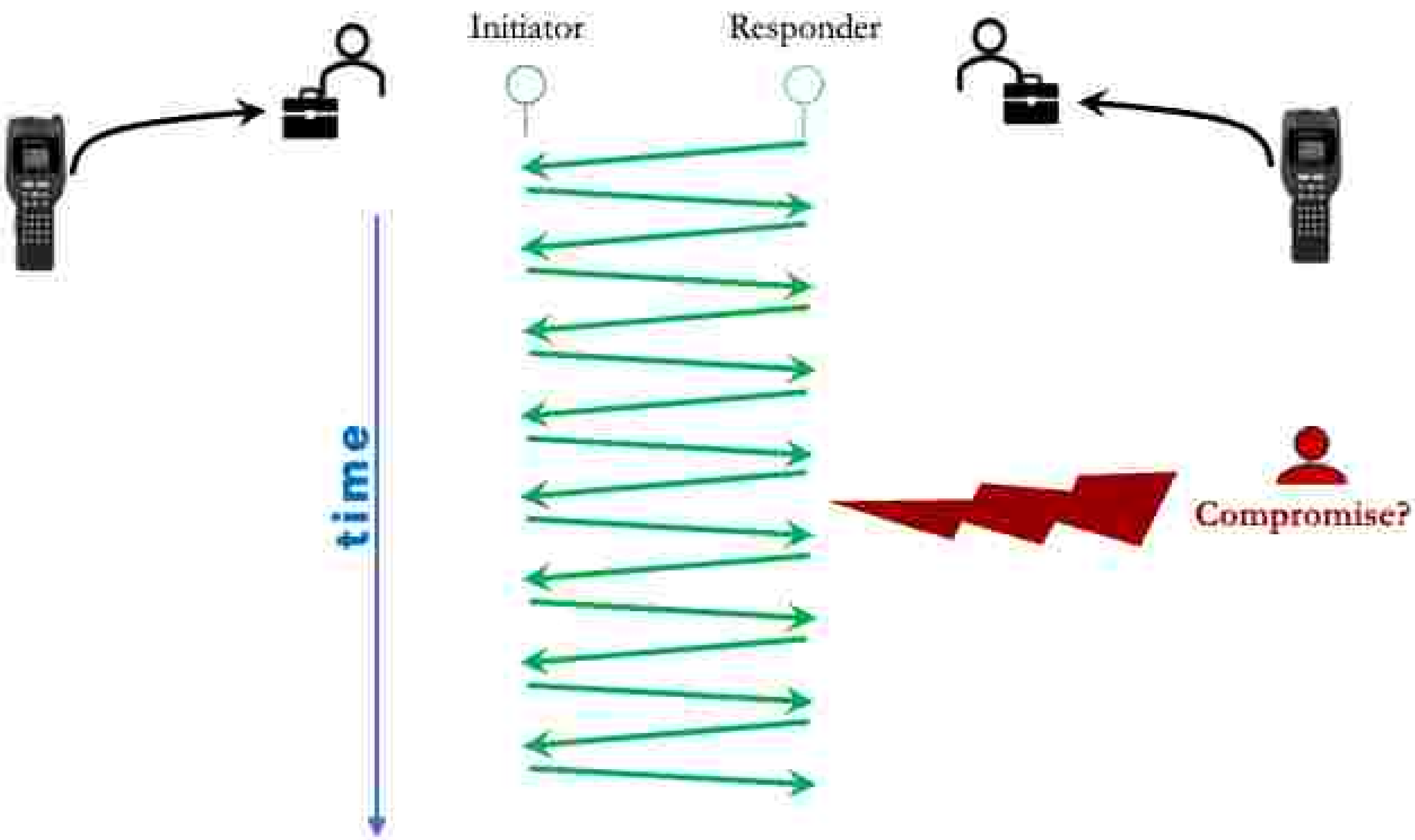


Session-based

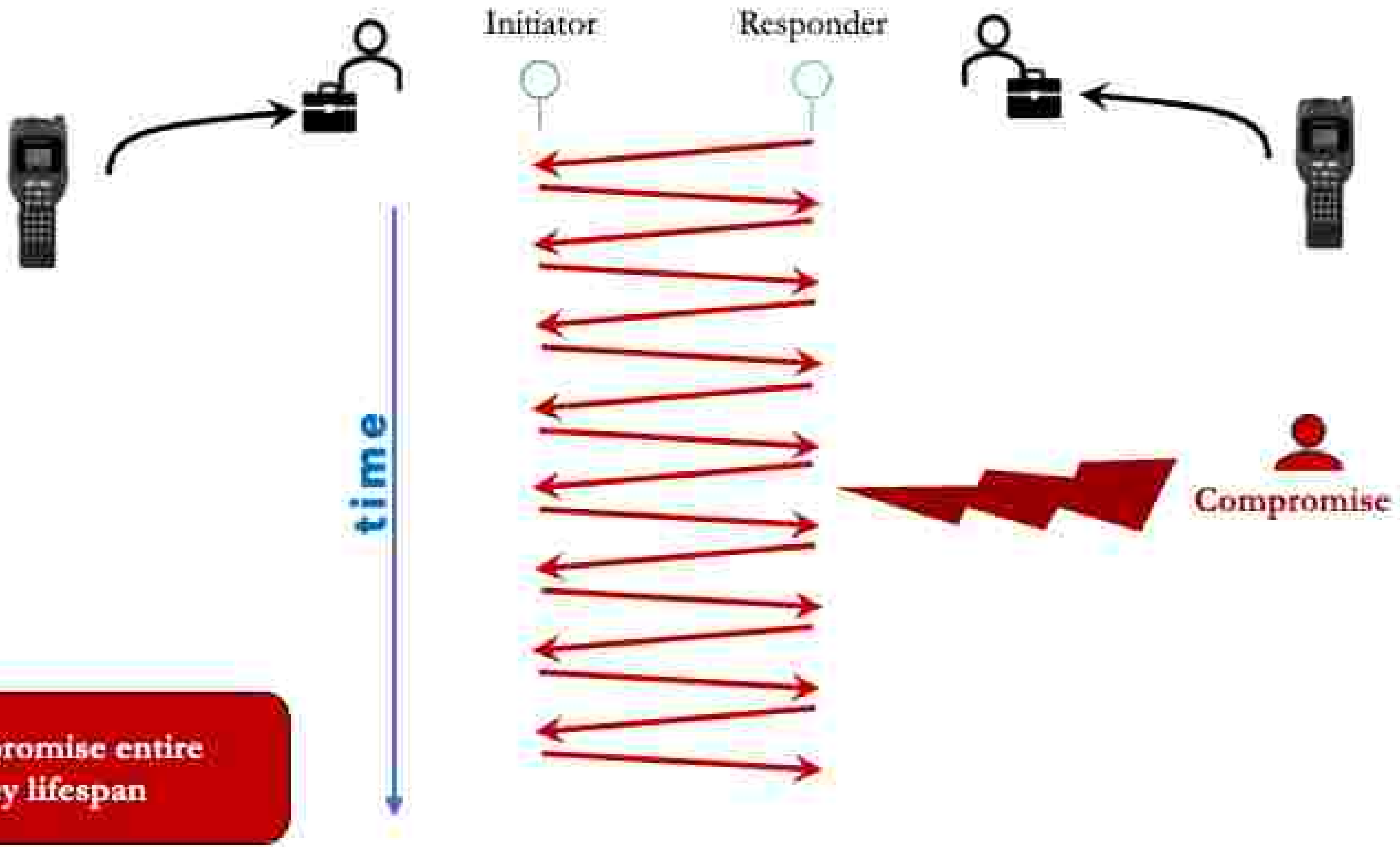


Asynchronous

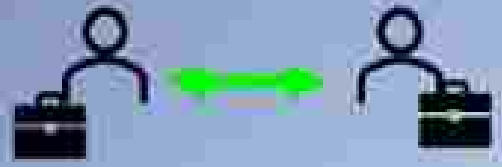
Attack risk



Attack risk



Compromise entire key lifespan



Pre-shared Key

Jamming

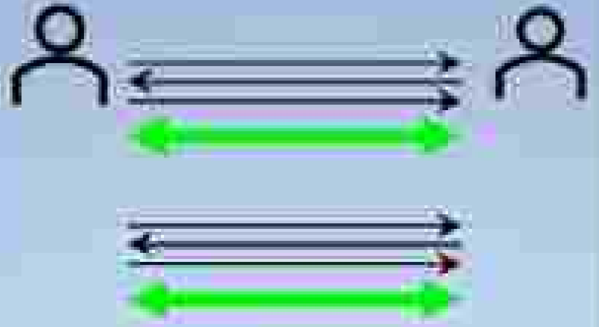
Traceability

Attack risk

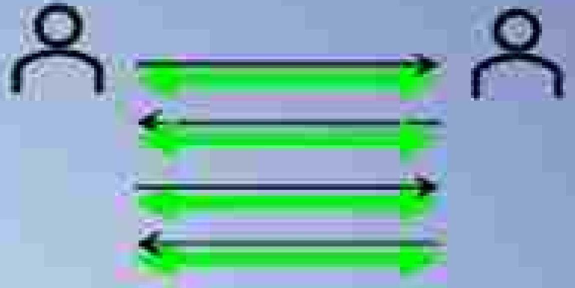
Interoperability

Manual overhead

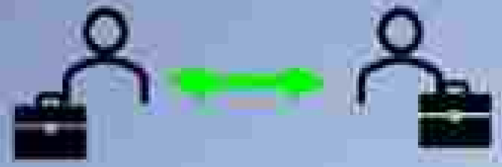
Scalability



Session-based



Asynchronous



Pre-shared Key

Jamming

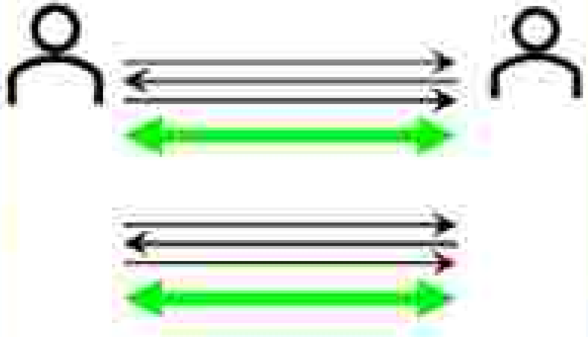
Traceability

Attack risk

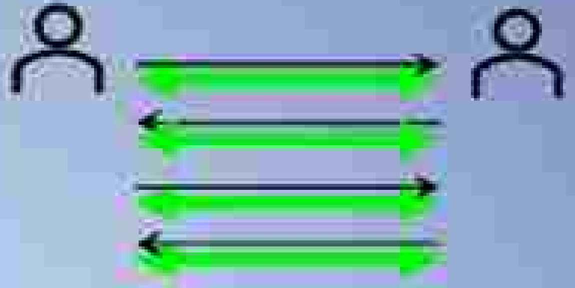
Interoperability

Manual overhead

Scalability

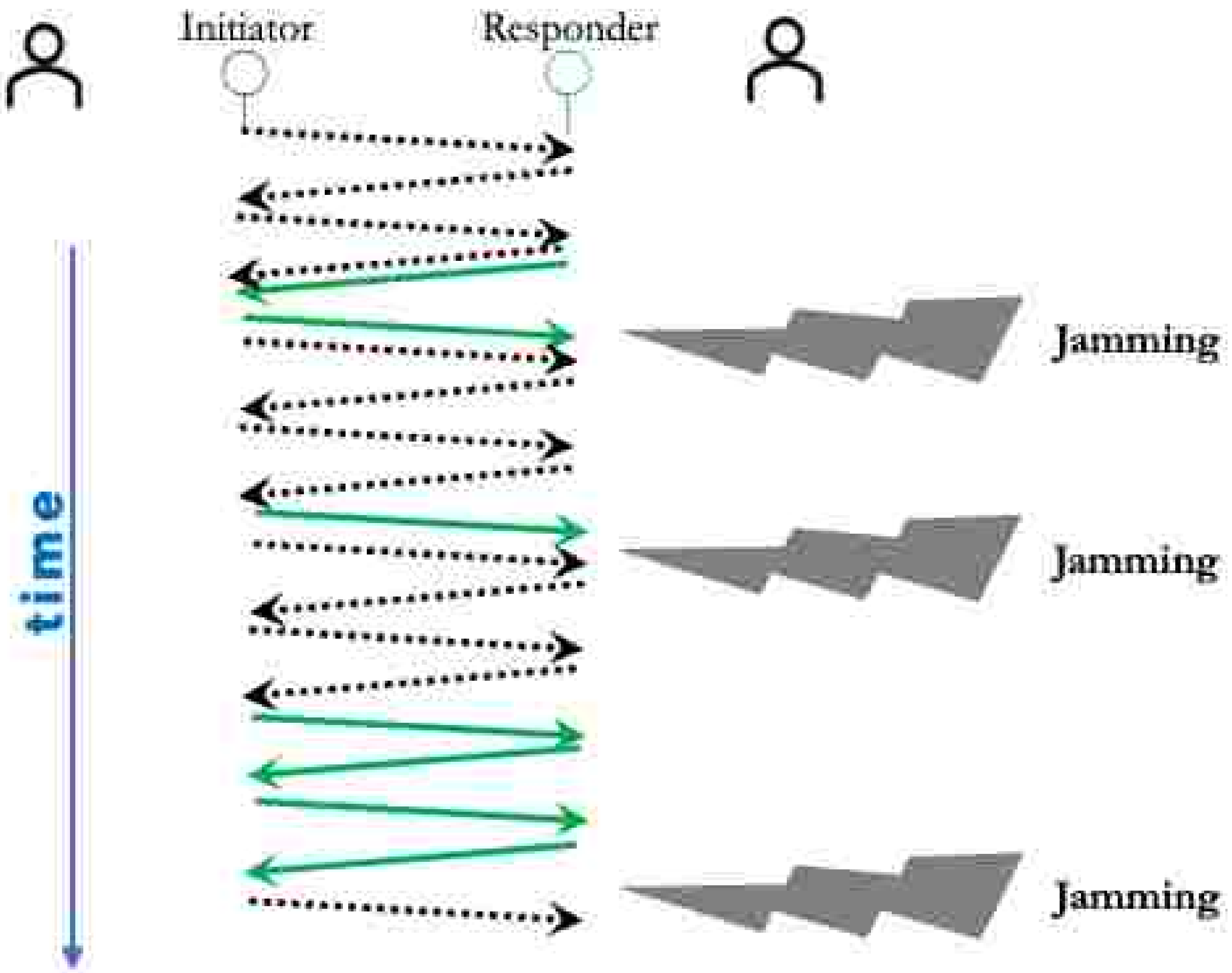


Session-based



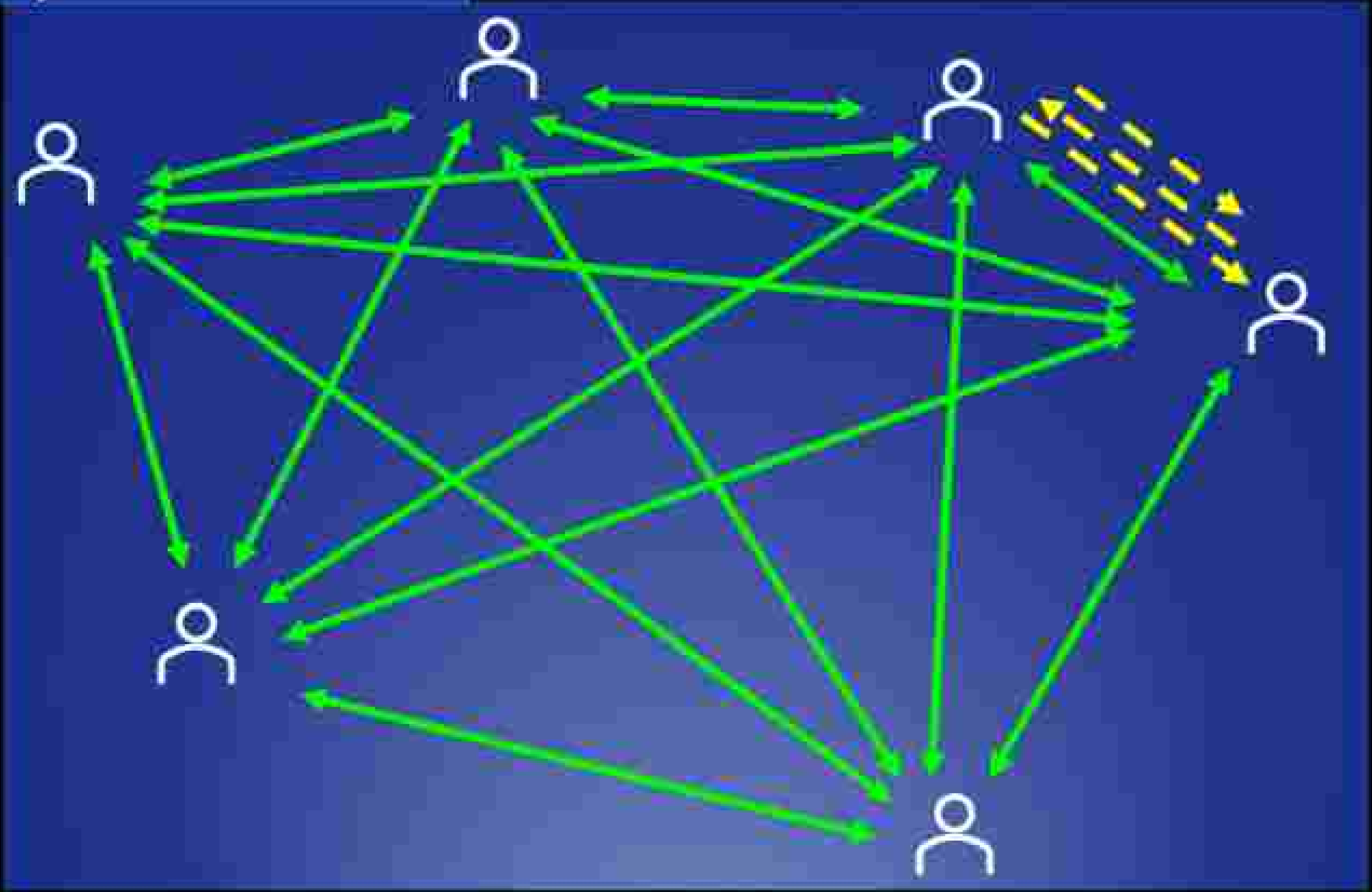
Asynchronous

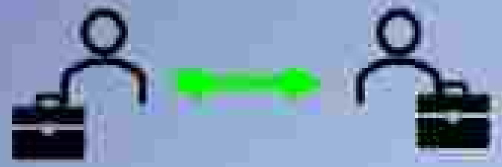
Jamming



Added delays under jamming

Scalability





Pre-shared Key

Jamming

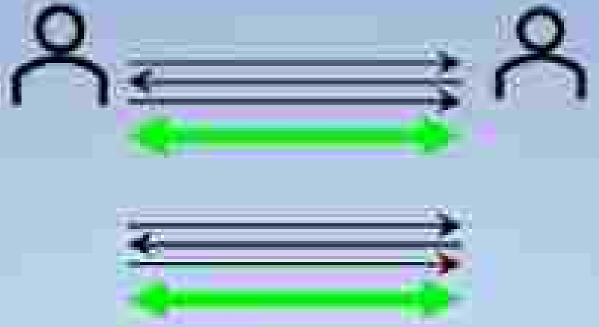
Traceability

Attack risk

Interoperability

Manual overhead

Scalability



Session-based

Jamming

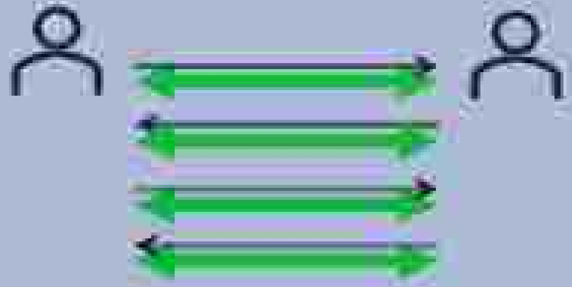
Traceability

Attack risks (also server access)

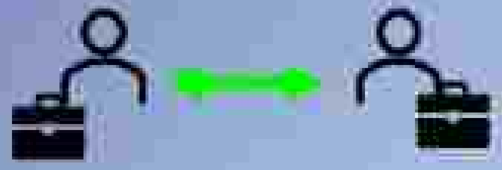
Interoperability

Manual overhead

Scalability



Asynchronous



Pre-shared Key

Jamming

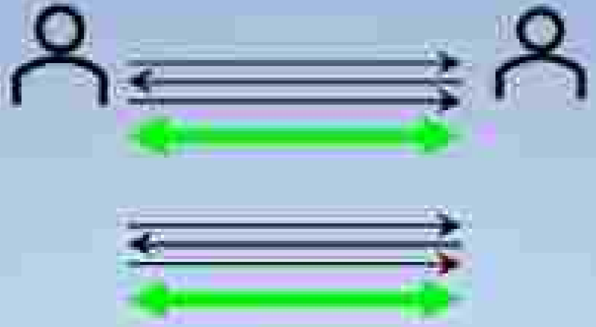
Traceability

Attack risk

Interoperability

Manual overhead

Scalability



Session-based

Jamming

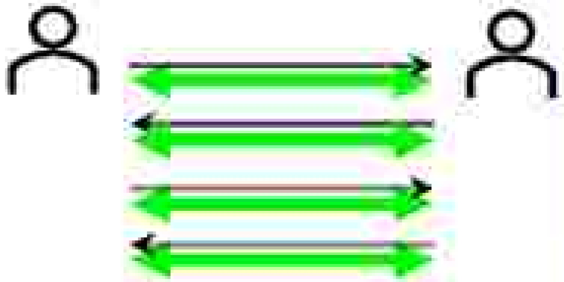
Traceability

Attack risks (also server access)

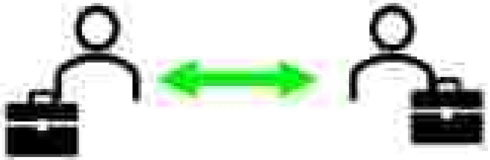
Interoperability

Manual overhead

Scalability



Asynchronous



Keyfill devices

Pre-shared Key

Jamming

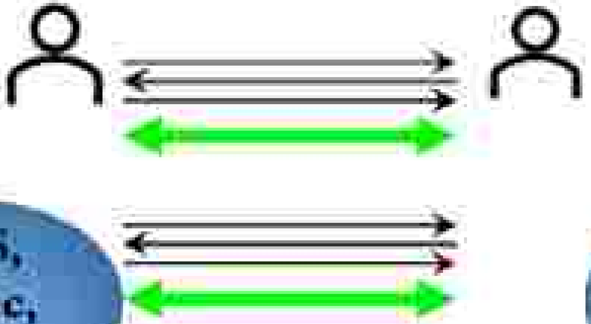
Traceability

Attack risk

Interoperability

Manual overhead

Scalability



TLS, IPSec, etc.

Session-based

Jamming

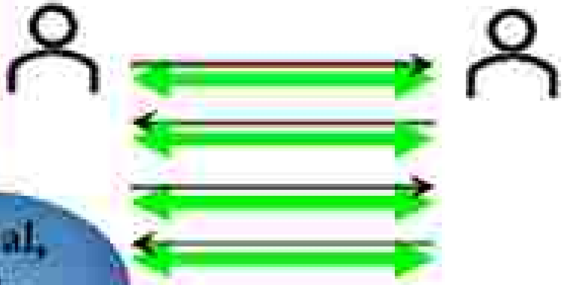
Traceability

Attack risks (also server access)

Interoperability

Manual overhead

Scalability



Signal, Wickr, etc.

Asynchronous

Jamming

Traceability

Attack risks*

Interoperability

Manual overhead

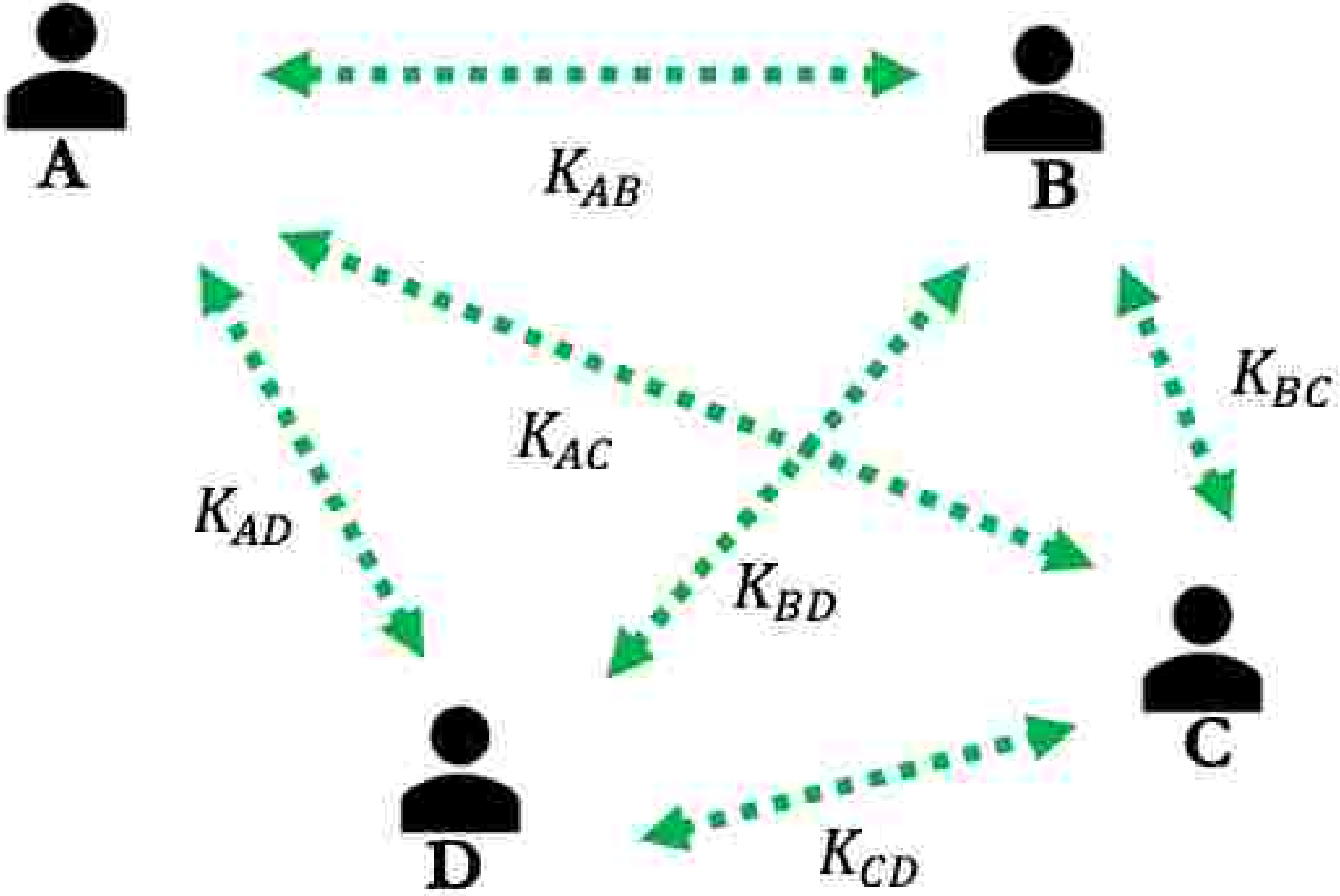
Scalability

* With ACKA

Scalability

Scalability

Pairwise Signal

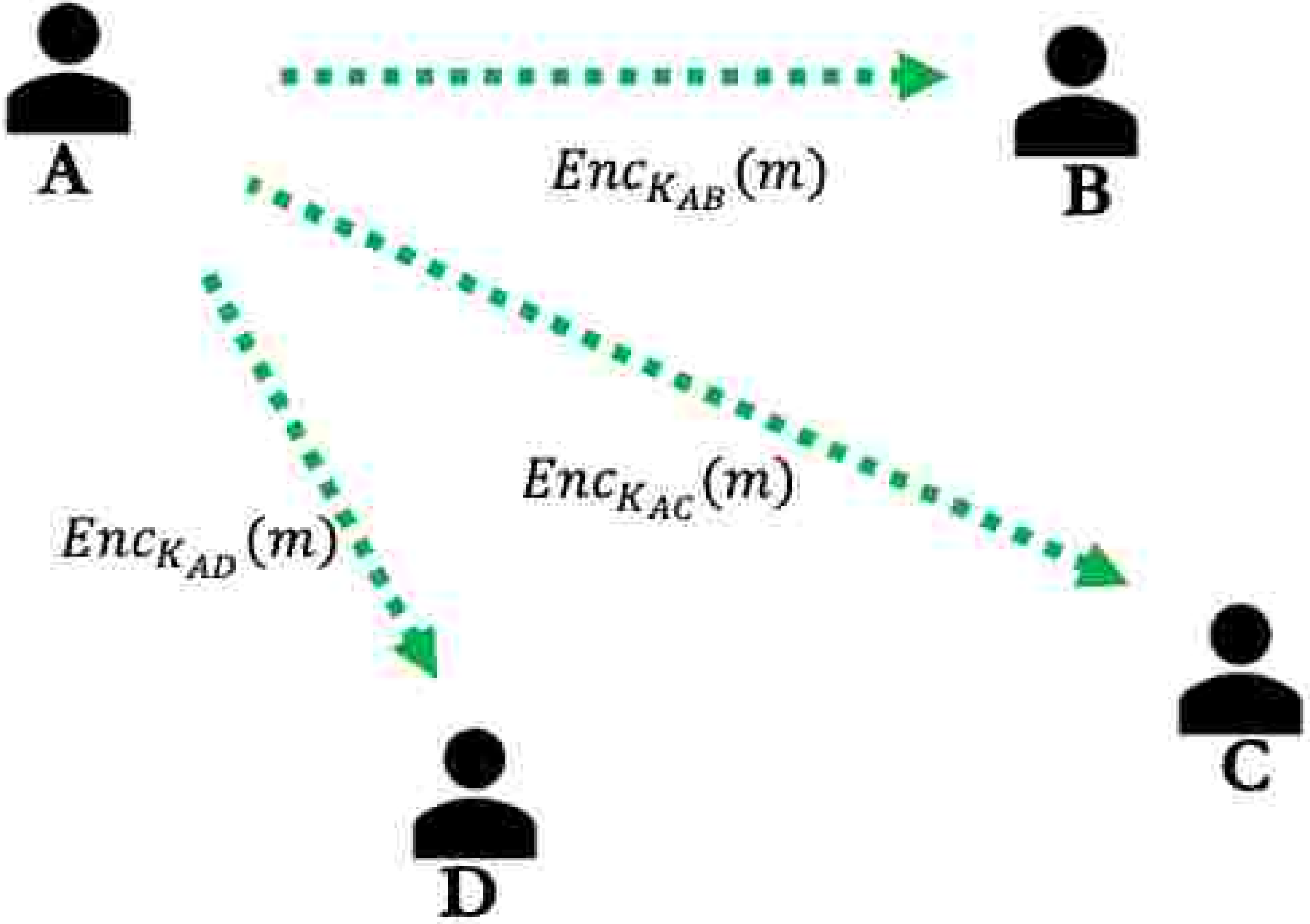


Scalability

Pairwise Signal

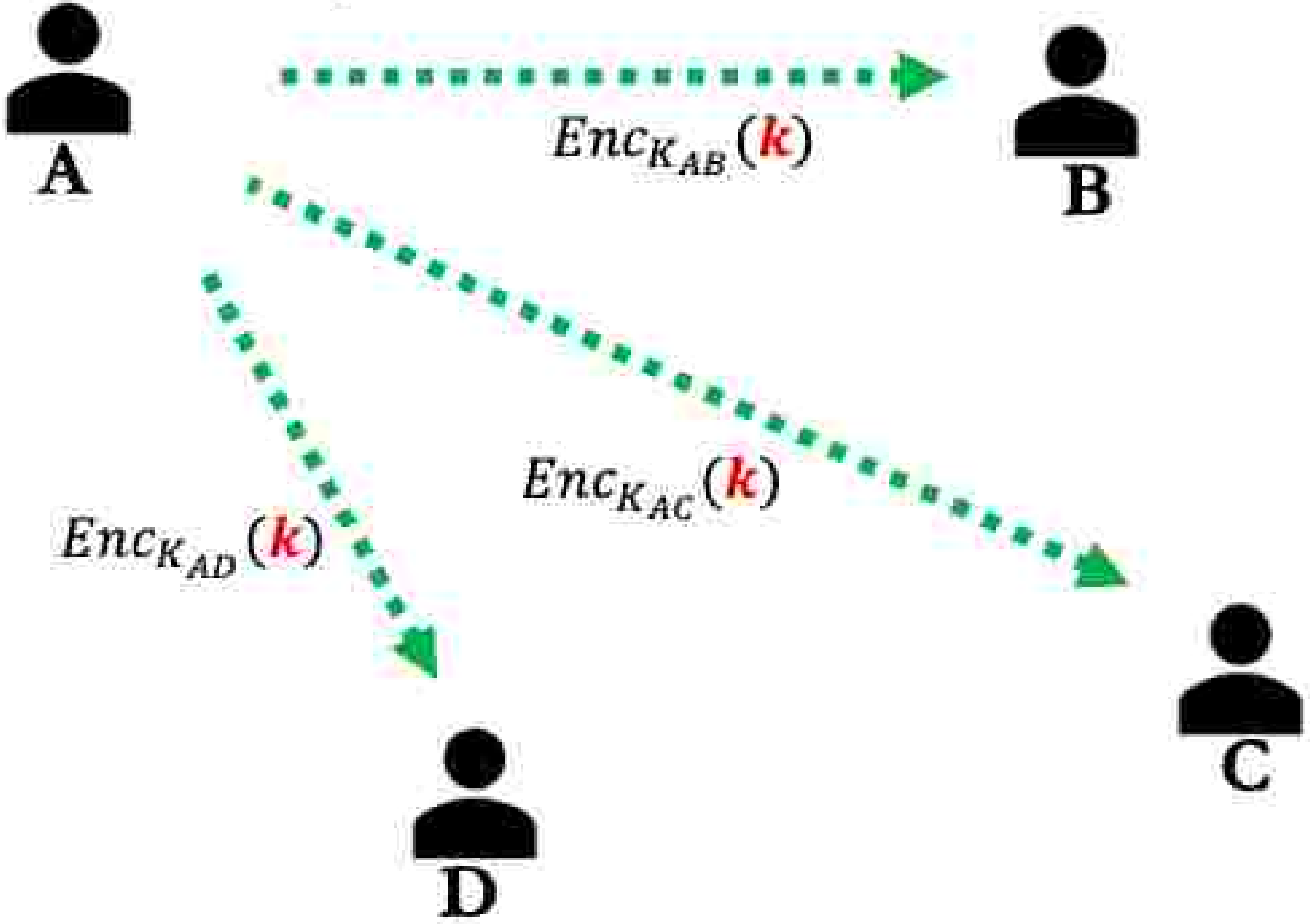
Message: m

Overhead!



Solution attempt: Sender Keys

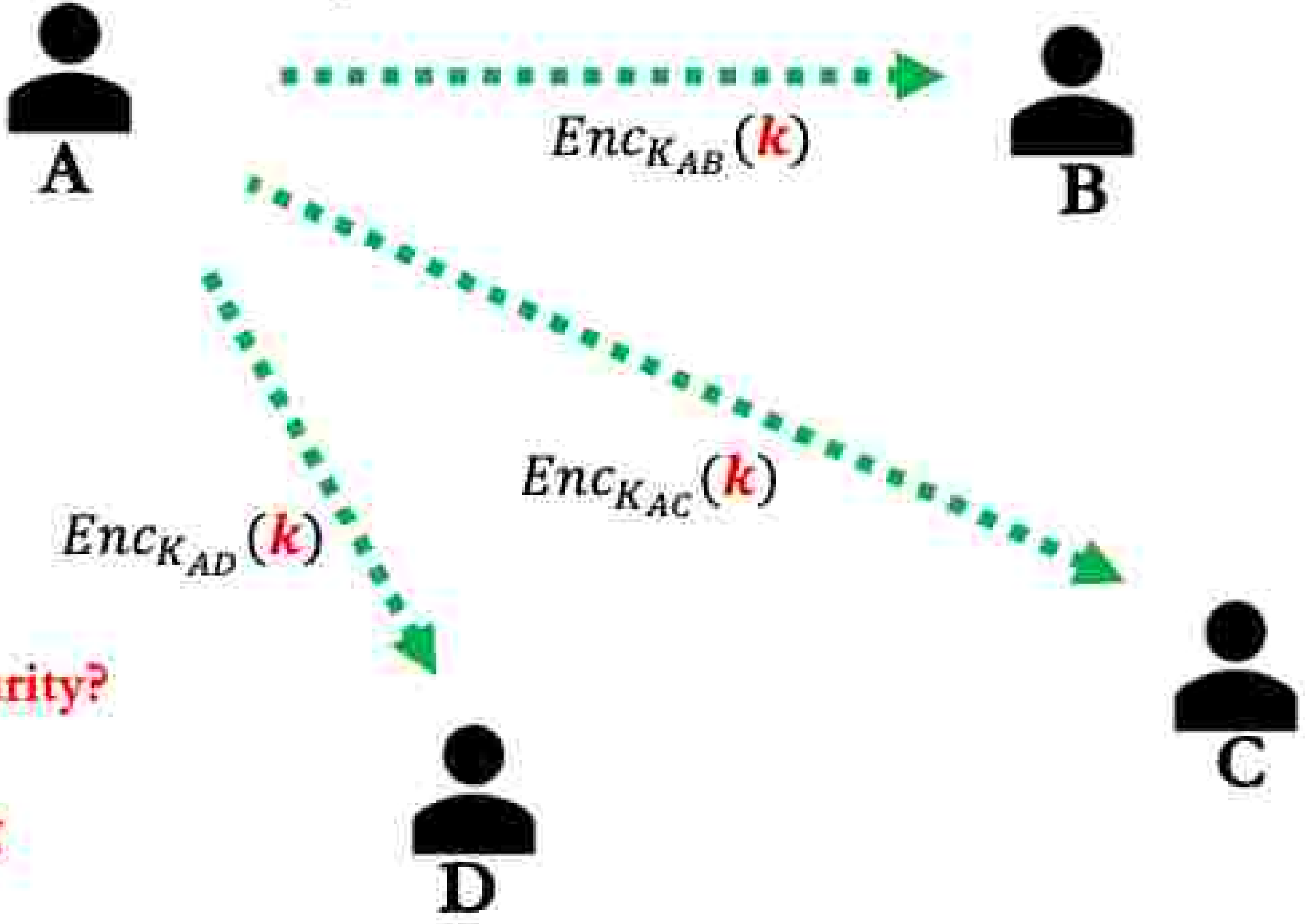
Message: m


A
B
C
D
$Enc_{K_{AB}}(\boldsymbol{k})$
$Enc_{K_{AC}}(\boldsymbol{k})$
$Enc_{K_{AD}}(\boldsymbol{k})$

Reducing overhead…

Solution attempt: Sender Keys

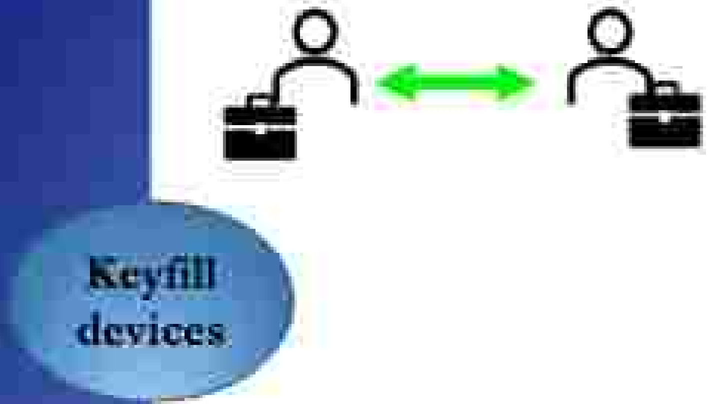
Message: m


A
B
C
D
$Enc_{K_{AB}}(\boldsymbol{k})$
$Enc_{K_{AC}}(\boldsymbol{k})$
$Enc_{K_{AD}}(\boldsymbol{k})$

Reducing overhead…

What about Forward Security?

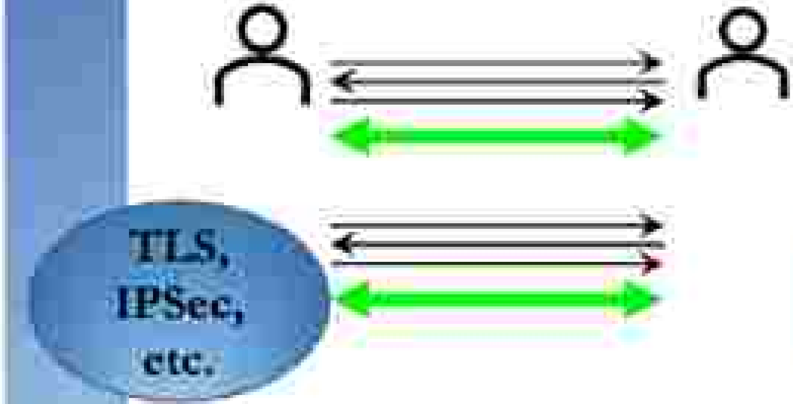
What about mis-behaving users?



Keyfill devices

Pre-shared Key

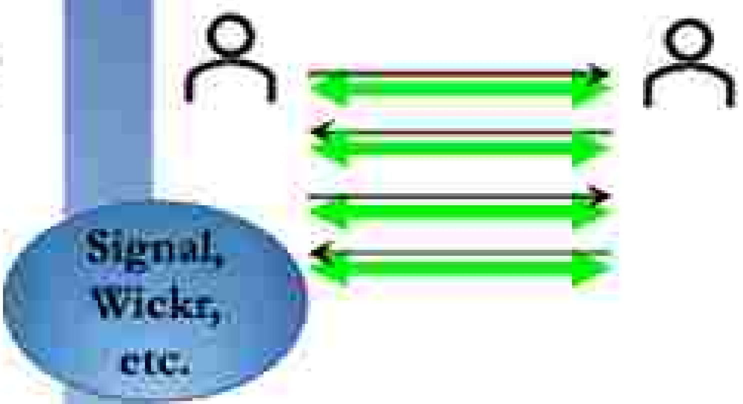
- Jamming
- Traceability
- Attack risk
- Interoperability
- Manual overhead
- Scalability



TLS, IPSec, etc.

Session-based

- Jamming
- Traceability
- Attack risks
- Interoperability
- Manual overhead
- Scalability

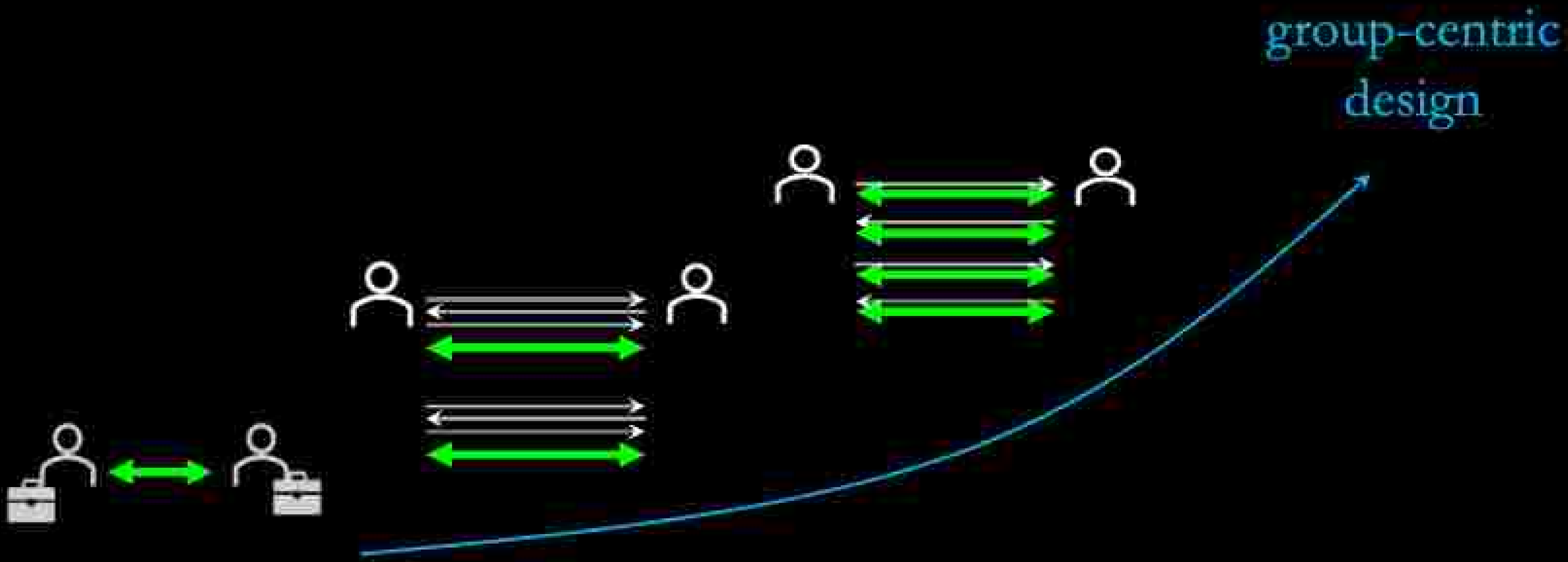


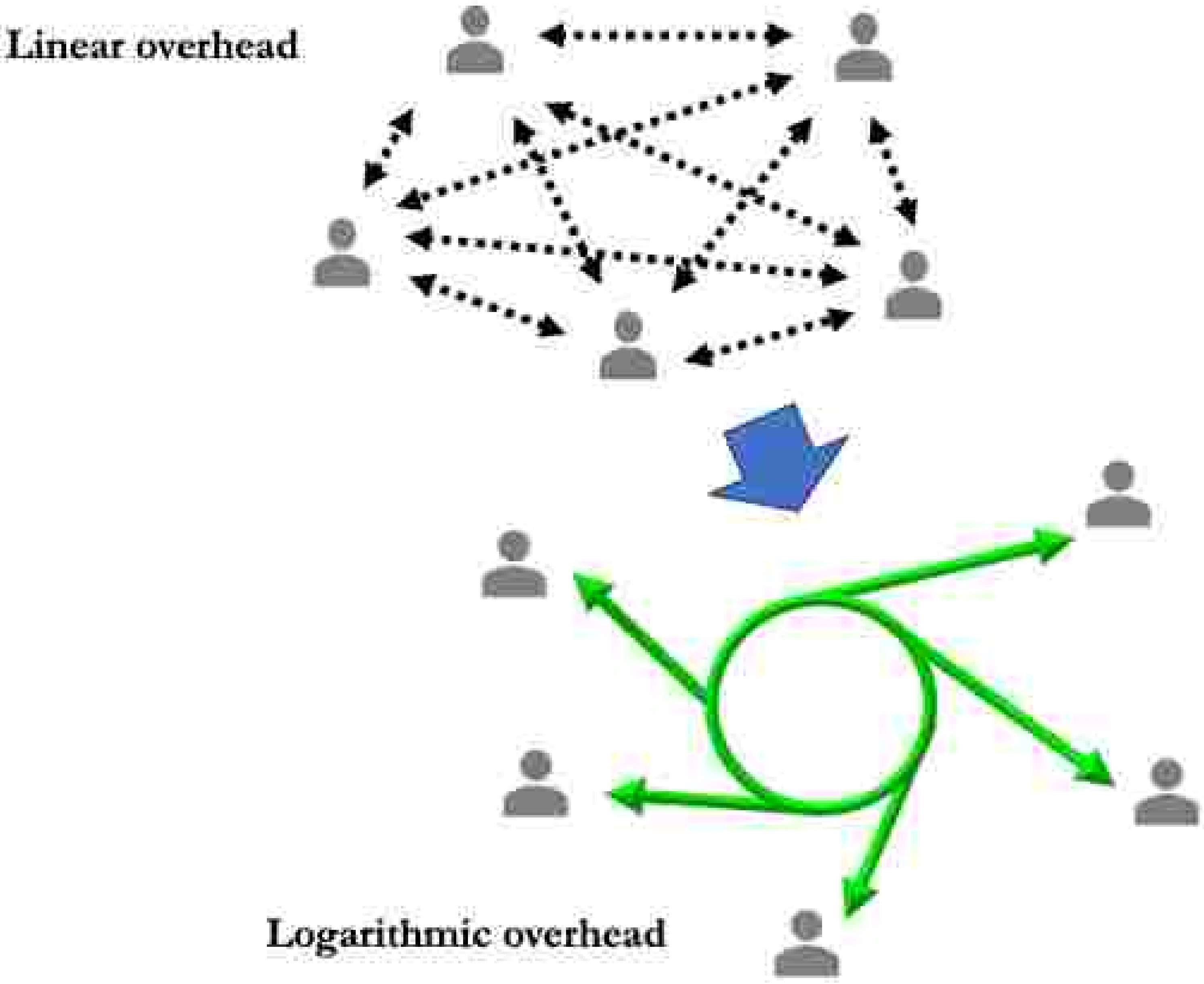
Signal, Wickr, etc.

Asynchronous

- Jamming
- Traceability
- Attack risks*
- Interoperability
- Manual overhead
- Scalability

* With ACKA


group-centric
design



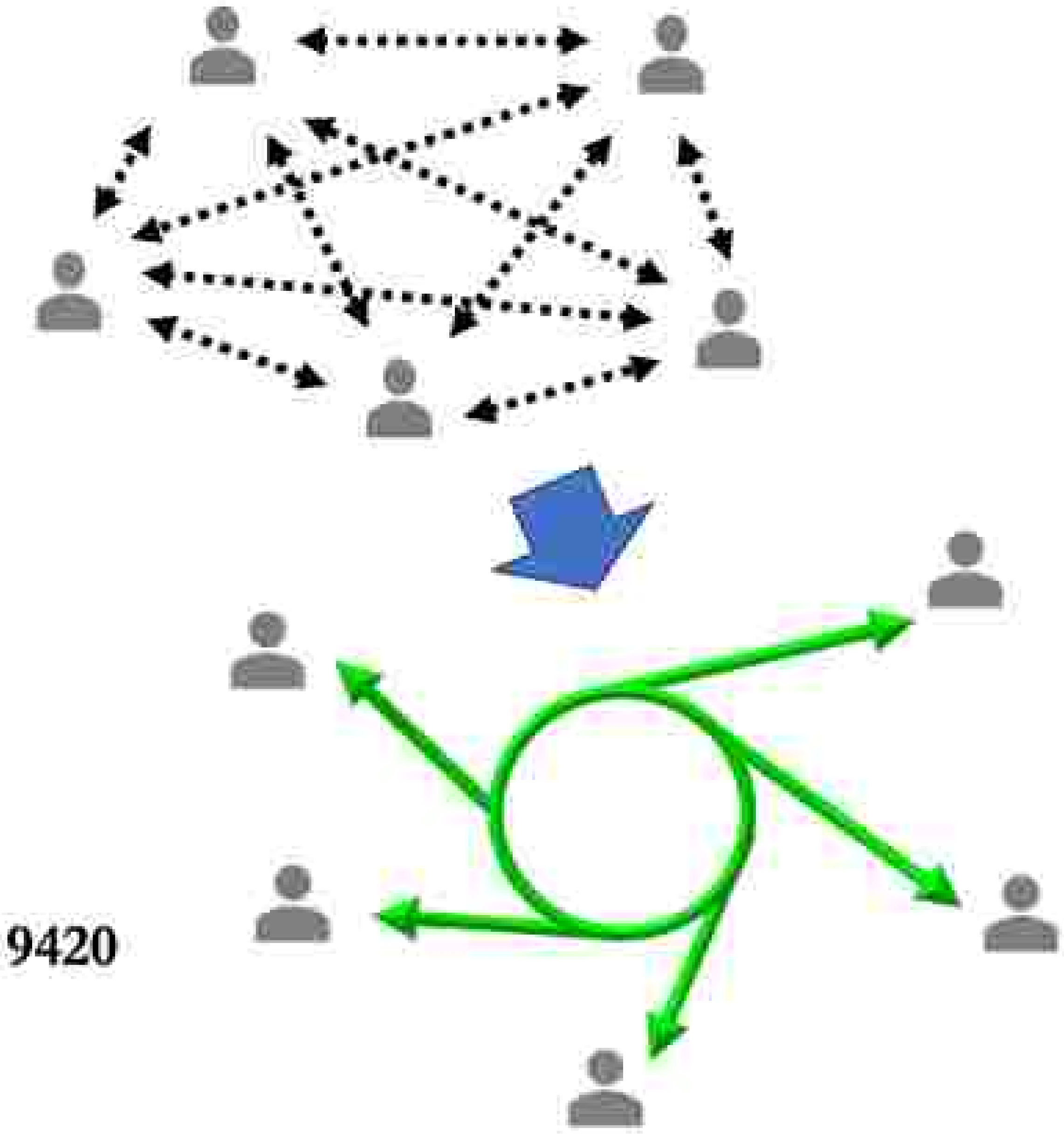
Linear overhead

Logarithmic overhead

IETF

Messaging Layer Security (MLS)

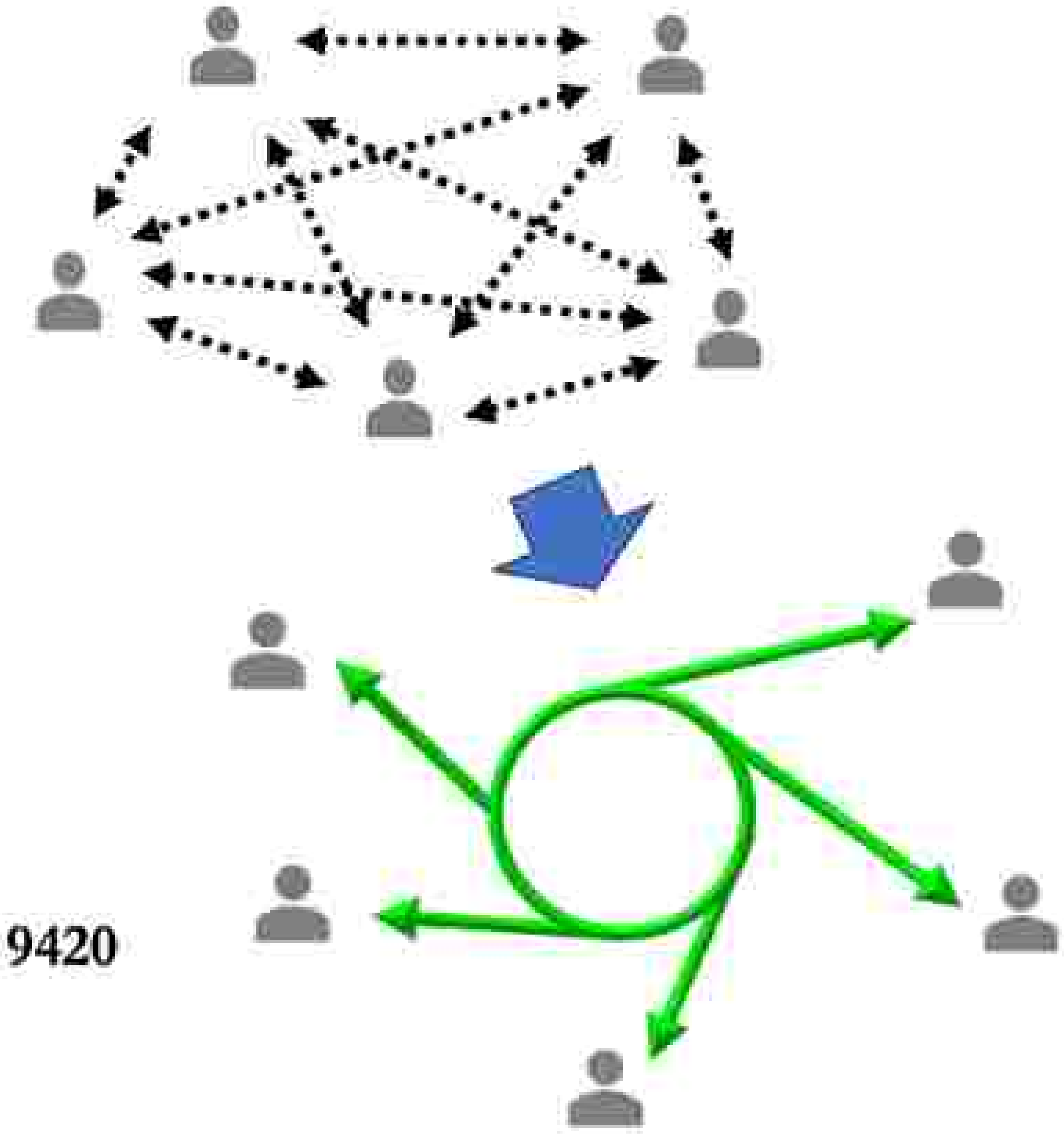
*International Standard: IETF RFC 9420



webex by CISCO RingCentral

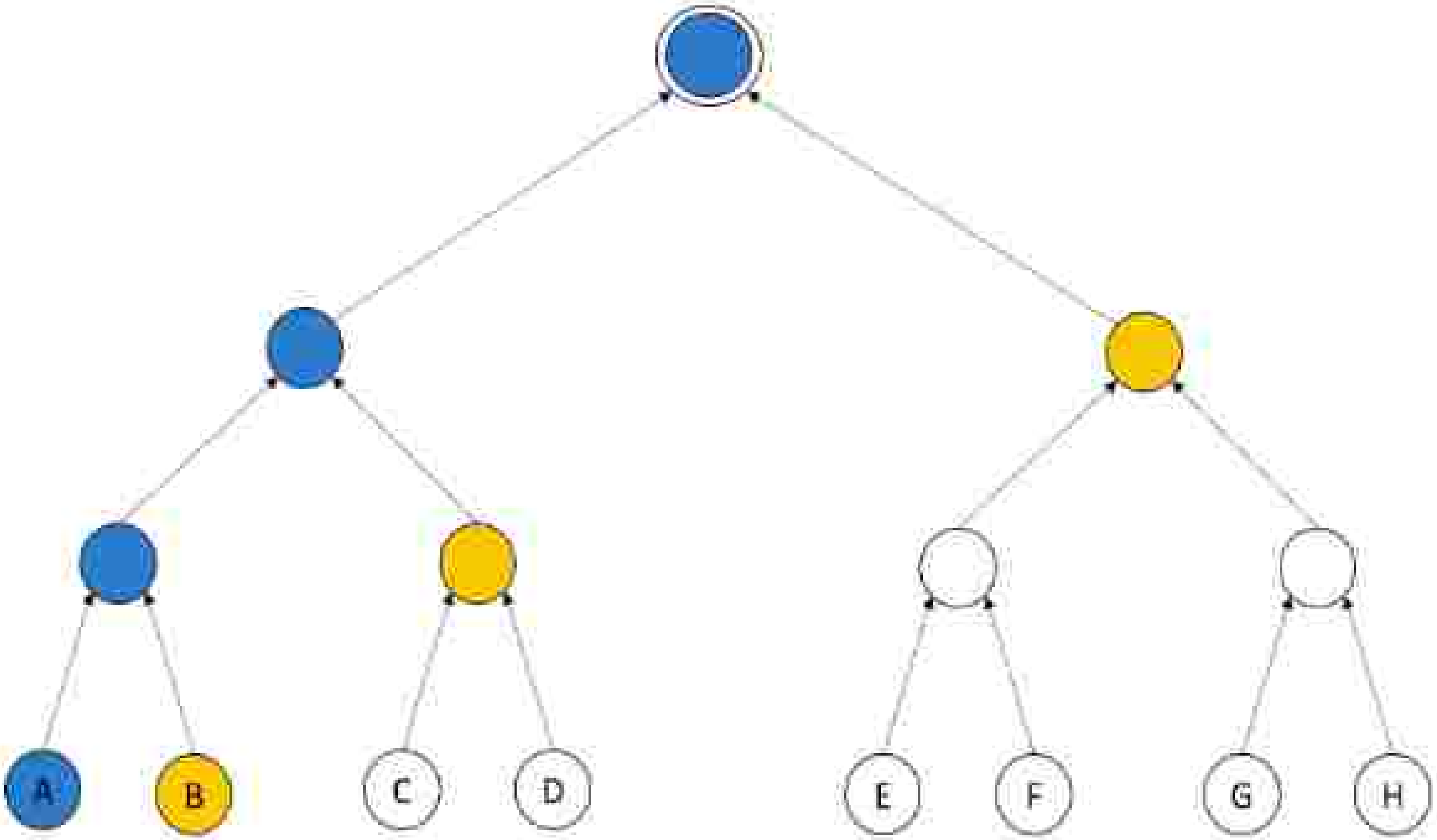
aws wickr Ph

IETF



Messaging Layer Security (MLS)

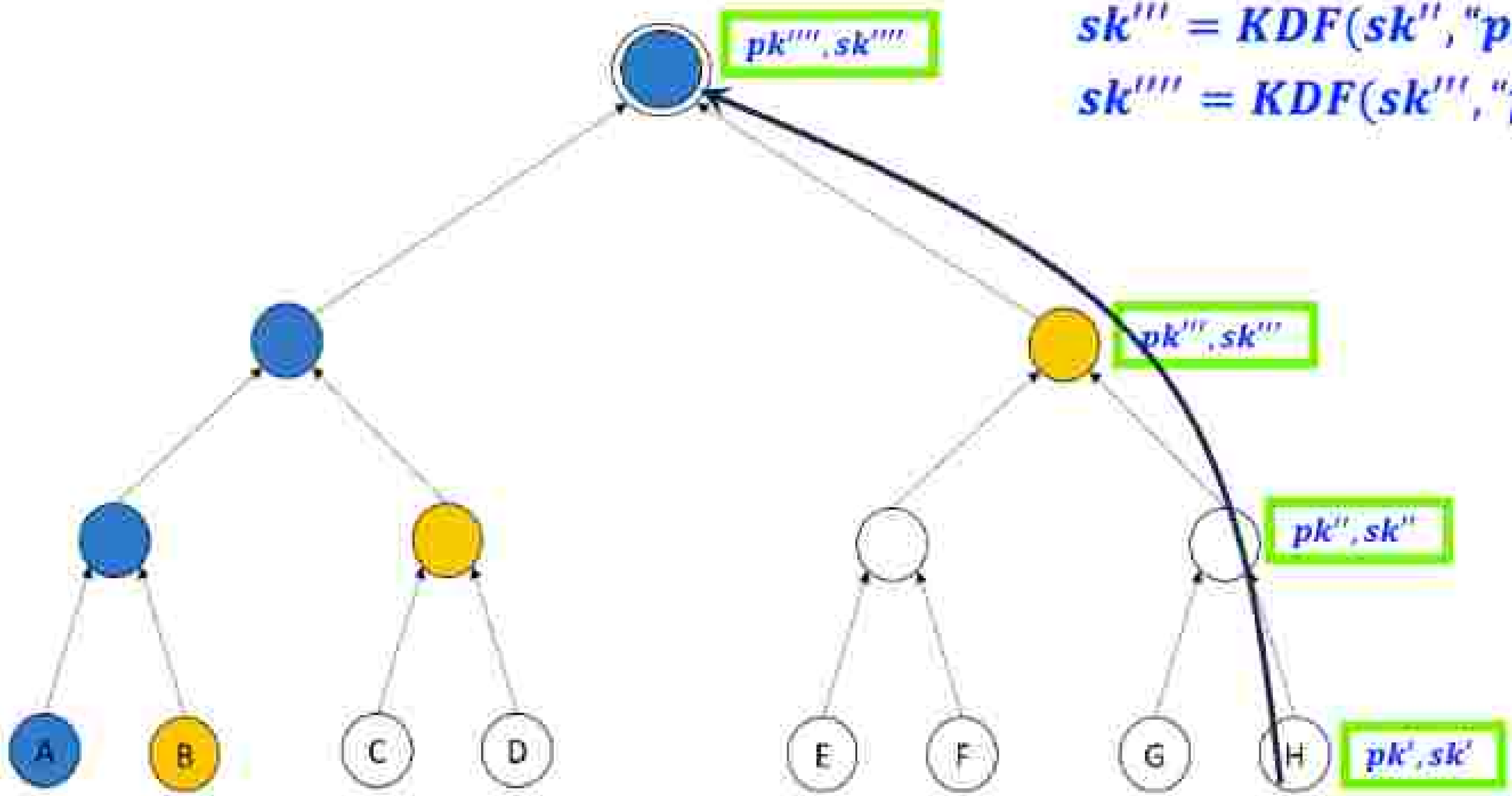
*International Standard: IETF RFC 9420

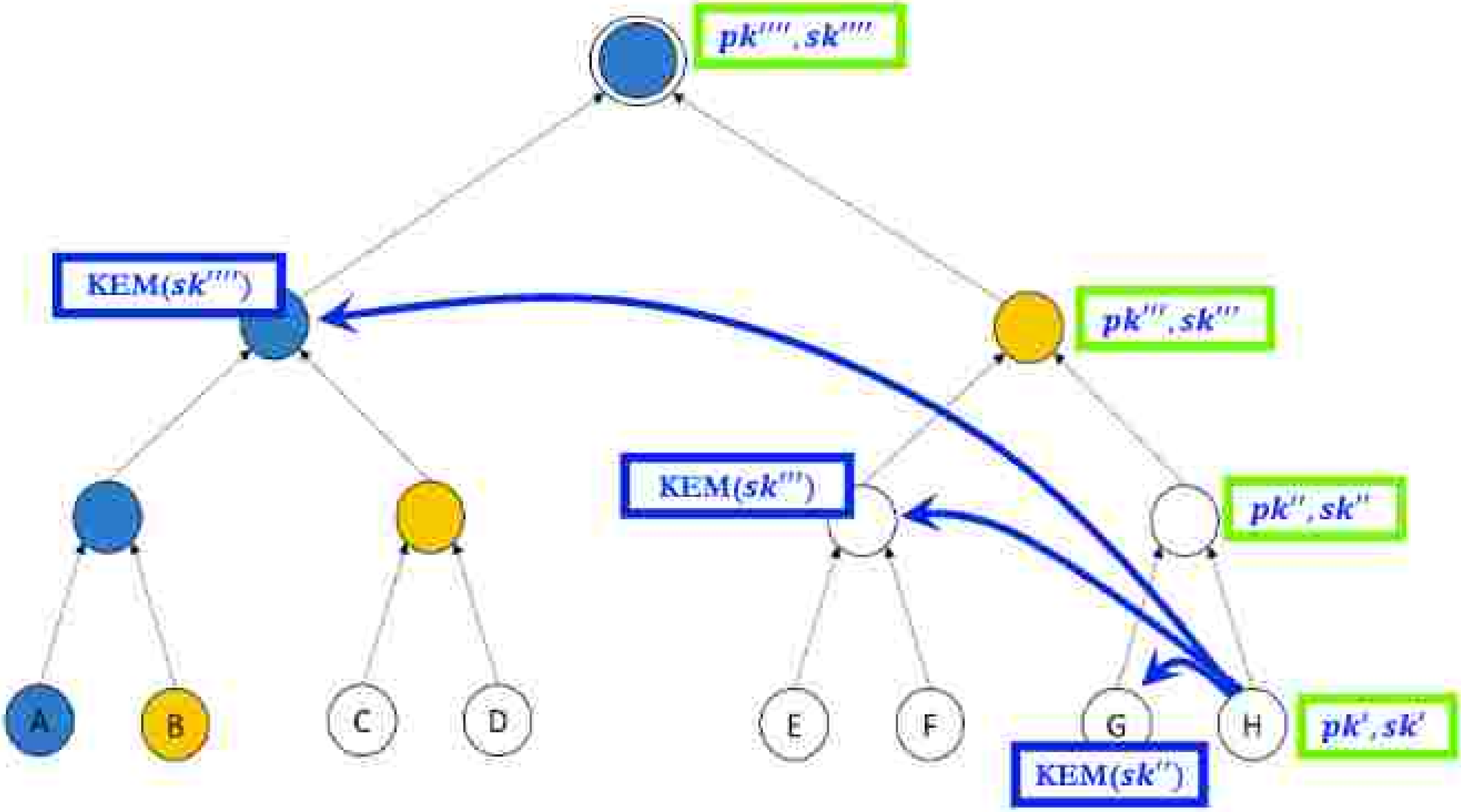

A
B
C
D
E
F
G
H

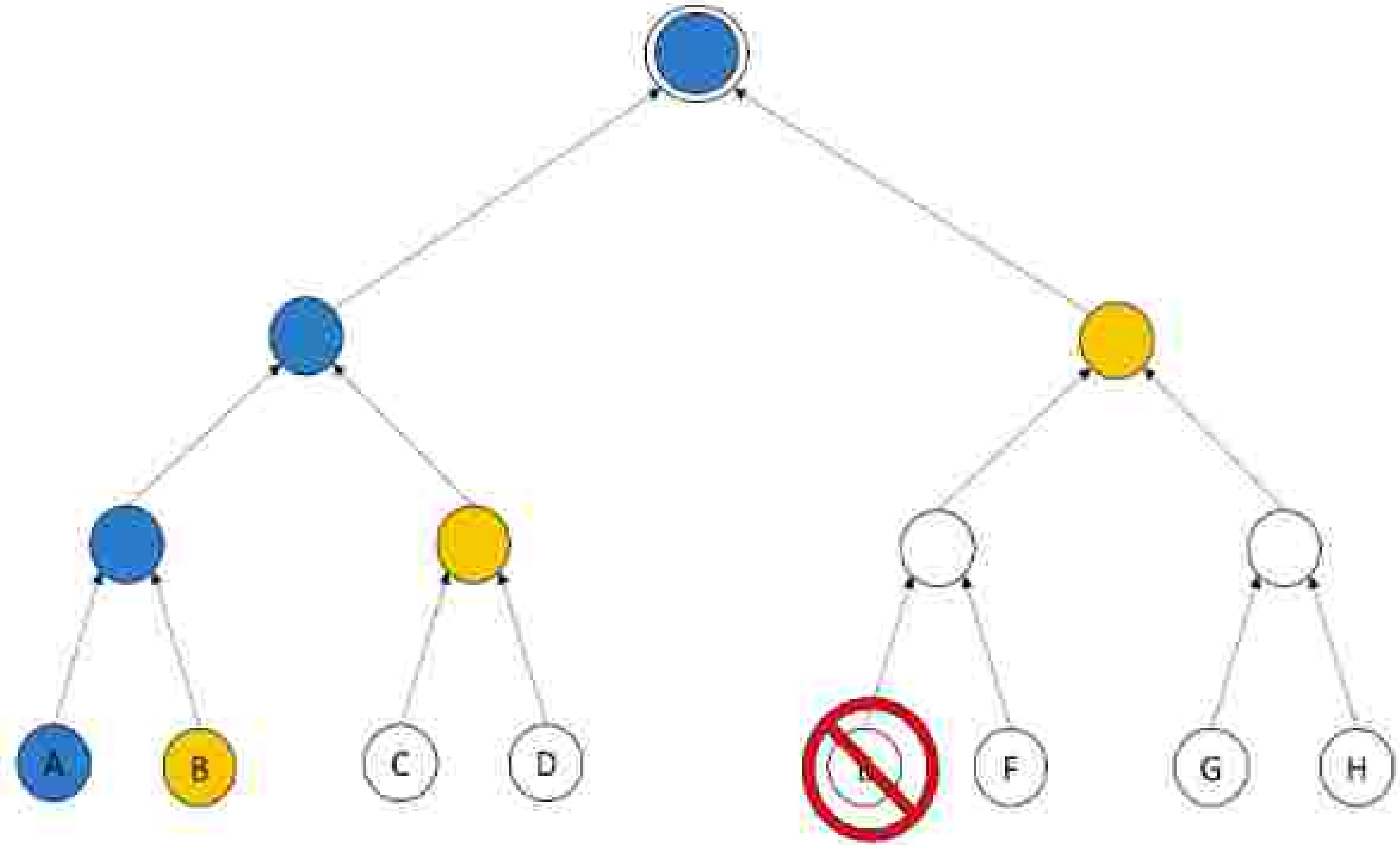
$$sk'' = KDF(sk', \text{"parent"})$$

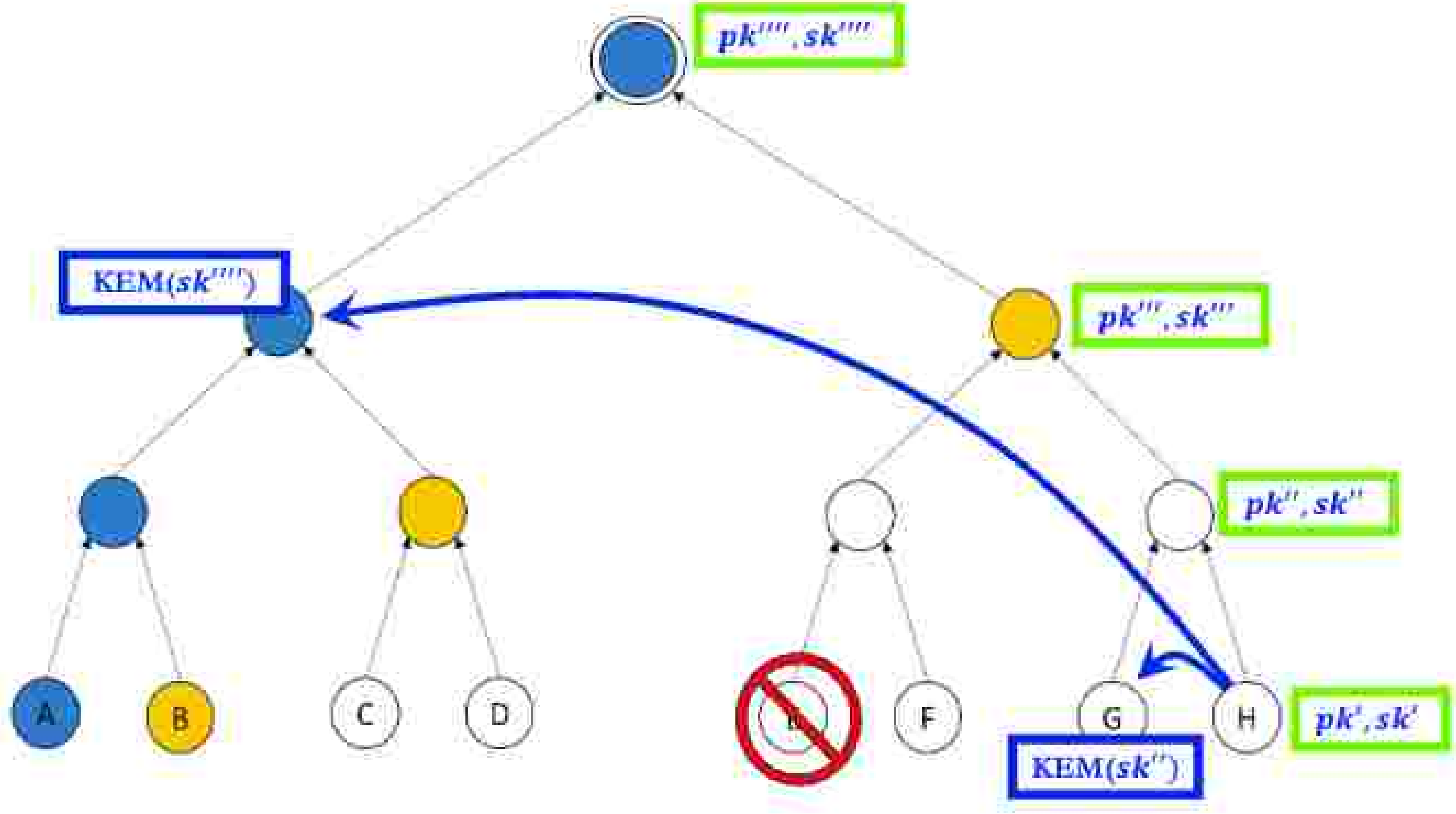
$$sk''' = KDF(sk'', \text{"parent"})$$

$$sk'''' = KDF(sk''', \text{"parent"})$$

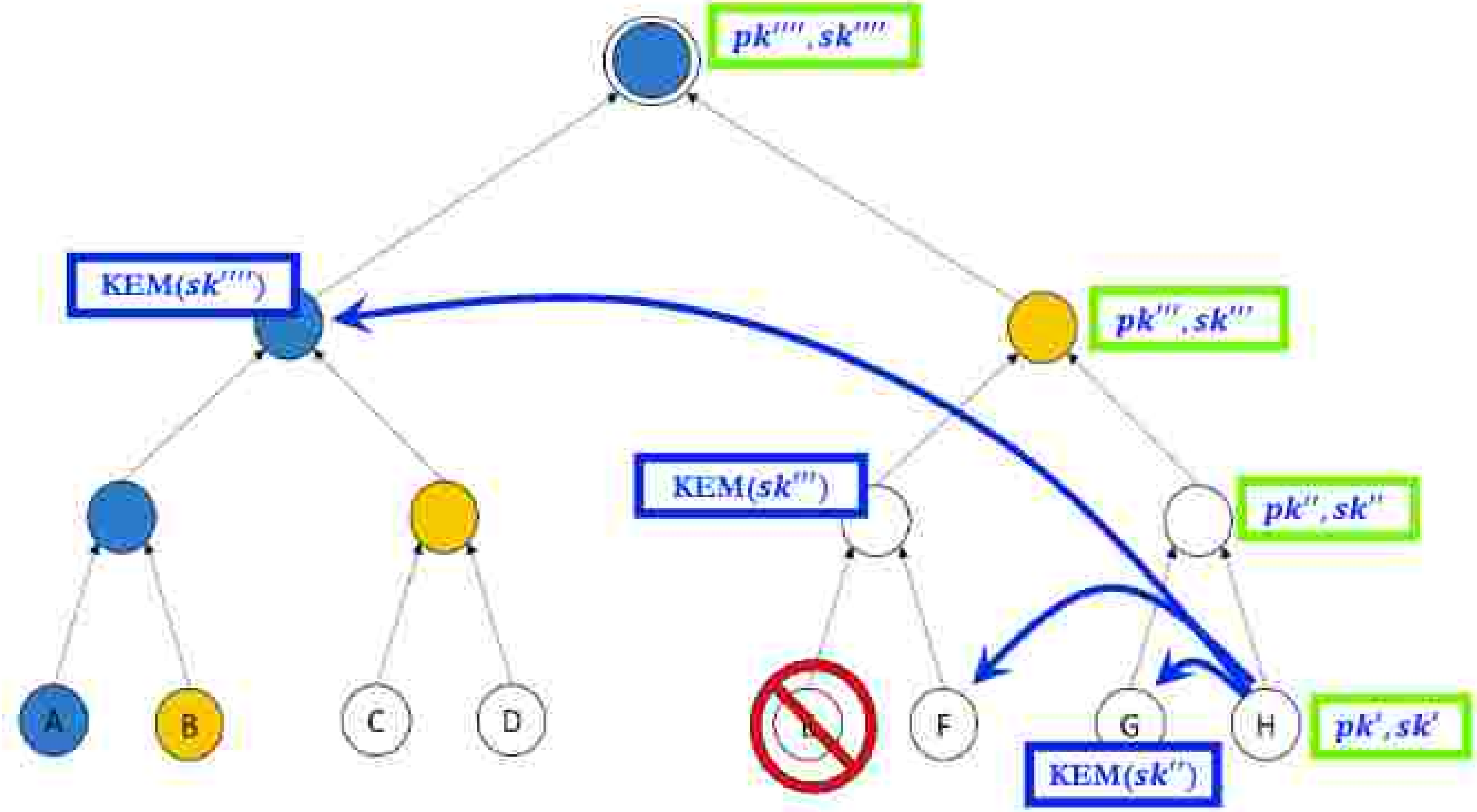


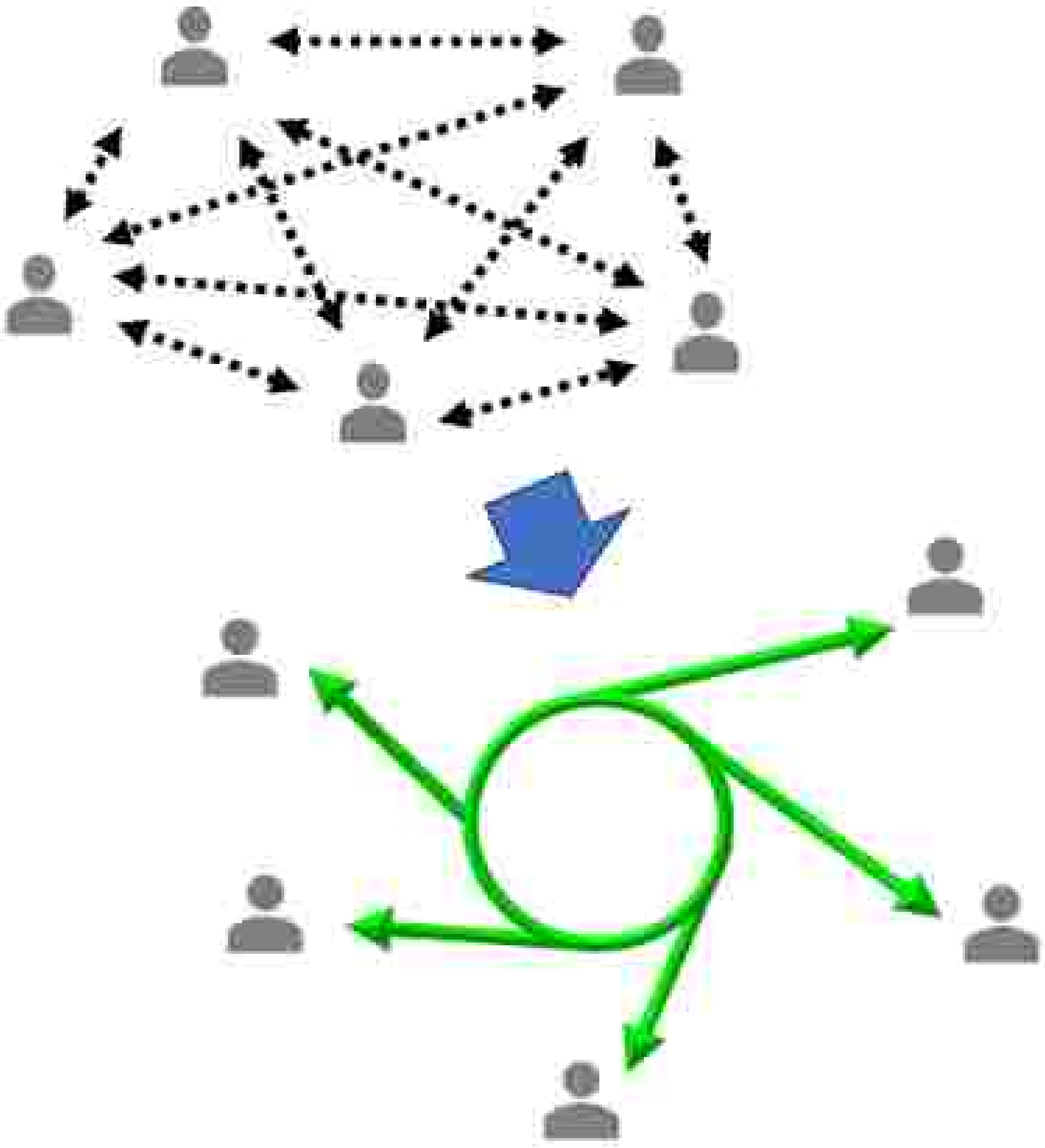

pk'''', sk''''
KEM(sk'''')
pk''', sk'''
KEM(sk''')
pk'', sk''
A
B
C
D
E
F
G
H
pk', sk'
KEM(sk'')


A
B
C
D
F
G
H



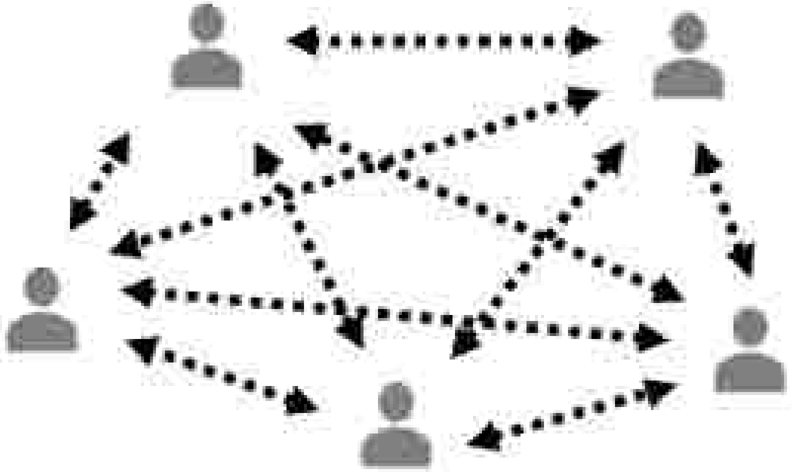
pk'''', sk''''
KEM(sk'''')
pk''', sk'''
pk'', sk''
A
B
C
D
E
F
G
H
pk', sk'
KEM(sk'')


pk'''', sk''''
KEM(sk'''')
pk''', sk'''
KEM(sk''')
pk'', sk''
A
B
C
D
E
F
G
H
pk', sk'
KEM(sk'')



Message Layer Security (MLS)

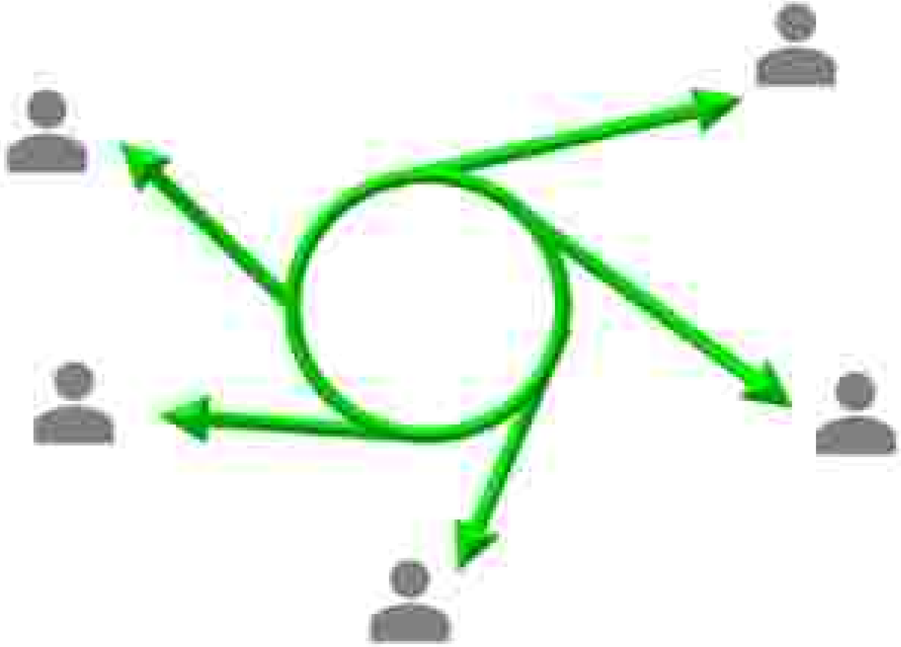
- Add group members
- Remove/eject group members
- Key evolution
- Create new groups
- Subgroup branching
- Post-quantum compatible



Multi-device = groups of pairs

↑

Design for pairs

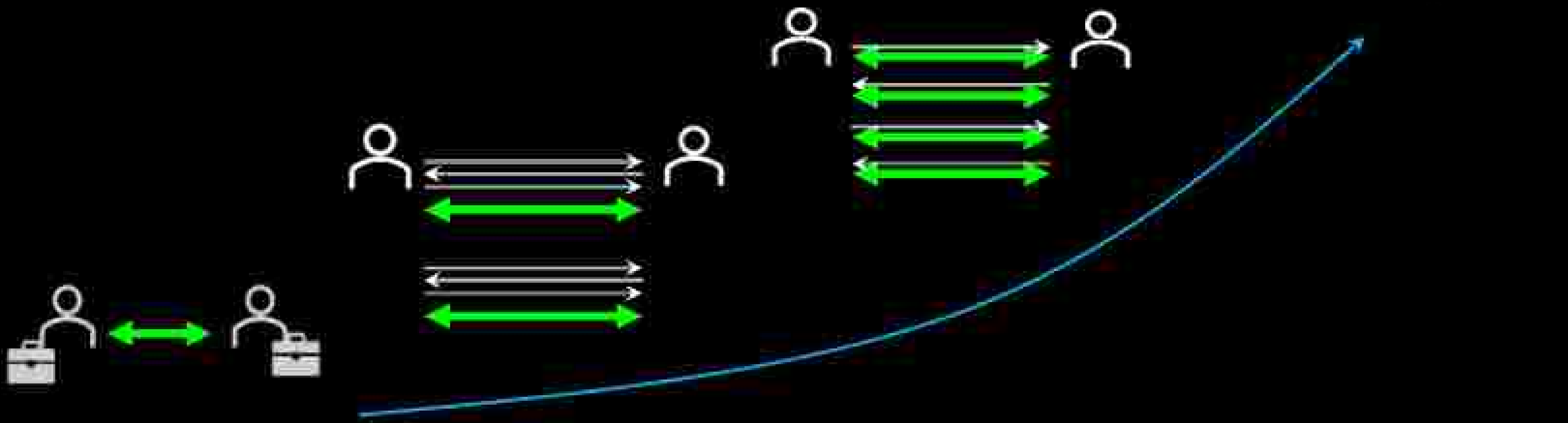


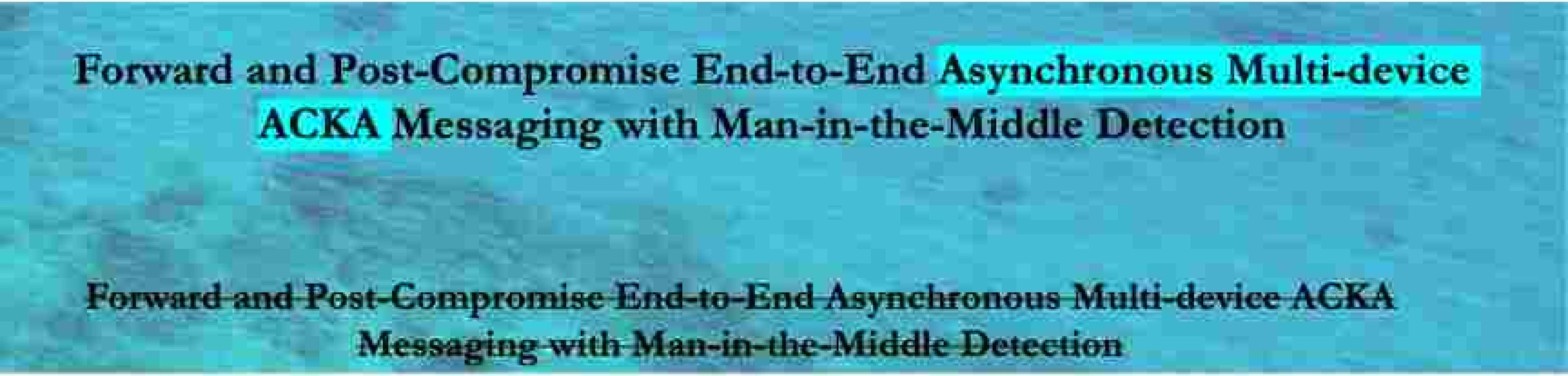
Design for multi-device

↓

Works for groups of size 2

Scalability to groups
Asynchronicity for relays / retrieval / delays
ACKA for continuous authentication





Forward and Post-Compromise End-to-End Asynchronous Multi-device ACKA Messaging with Man-in-the-Middle Detection

~~Forward and Post-Compromise End-to-End Asynchronous Multi-device ACKA Messaging with Man-in-the-Middle Detection~~

Have we covered "security"?

- Deniability / unlinkability
- Guardianship for offline Post-Compromise Security
- Signature key ratcheting for impersonation protection in future groups

Deniability: an MLS design story 😉

Application message deniability:

It is not possible to prove authorship of a given message M.

- Assuming the adversary is not a conversation partner (group external)
- Assuming that the adversary is a conversation partner
- Assuming that the adversary is the distribution service
- Assuming that the adversary is the authentication service

Ciphertext deniability:

It is not possible to prove authorship of a given ciphertext C.

- Assuming the adversary is not a conversation partner (group external)
- Assuming that the adversary is a conversation partner
- Assuming that the adversary is the distribution service
- Assuming that the adversary is the authentication service

Key deniability:

It is not possible to prove ownership of a given key K (regardless of messages sent).

- Assuming the adversary is not a conversation partner (group external)
- Assuming that the adversary is a conversation partner
- Assuming that the adversary is the distribution service
- Assuming that the adversary is the authentication service

Non-application message deniability:

It is not possible to prove authorship of a given non-application message M.

- Assuming the adversary is not a conversation partner (group external)
- Assuming that the adversary is a conversation partner
- Assuming that the adversary is the distribution service
- Assuming that the adversary is the authentication service

Conversation membership unlinkability:

It is not possible to prove membership in a given conversation.

- Assuming the adversary is not a conversation partner (group external)
- Assuming that the adversary is a conversation partner
- Assuming that the adversary is the distribution service
- Assuming that the adversary is the authentication service

Ciphertext unlinkability:

If in possession and proof of authorship of a ciphertext C1, it is not possible to prove authorship of another ciphertext C2.

- Assuming the adversary is not a conversation partner (group external)
- Assuming that the adversary is a conversation partner
- Assuming that the adversary is the distribution service
- Assuming that the adversary is the authentication service

Each of the possibilities can be considered under **online** or **offline** deniability….

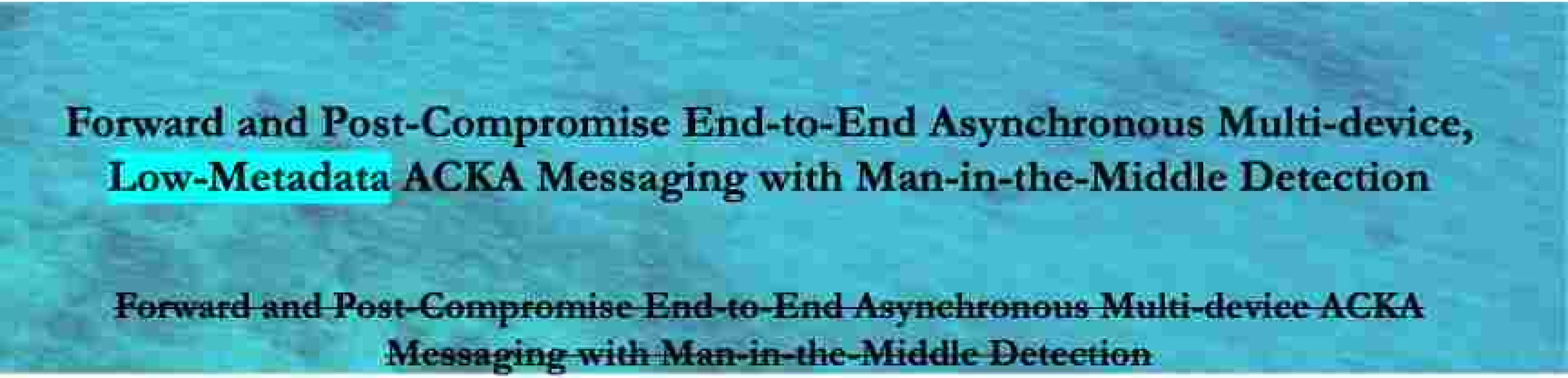
So those are 48 options to start with. 😃

What deniability/privacy guarantees do people want?

- Activists (courts? framing?)
- "Normal" end users (false accusations? misinterpretations?)
- Governments (untraceability?)
- Cryptographic researchers (cool new algorithms and protocols?)

(OTR is over 15yrs old already!)

Metadata is dangerous

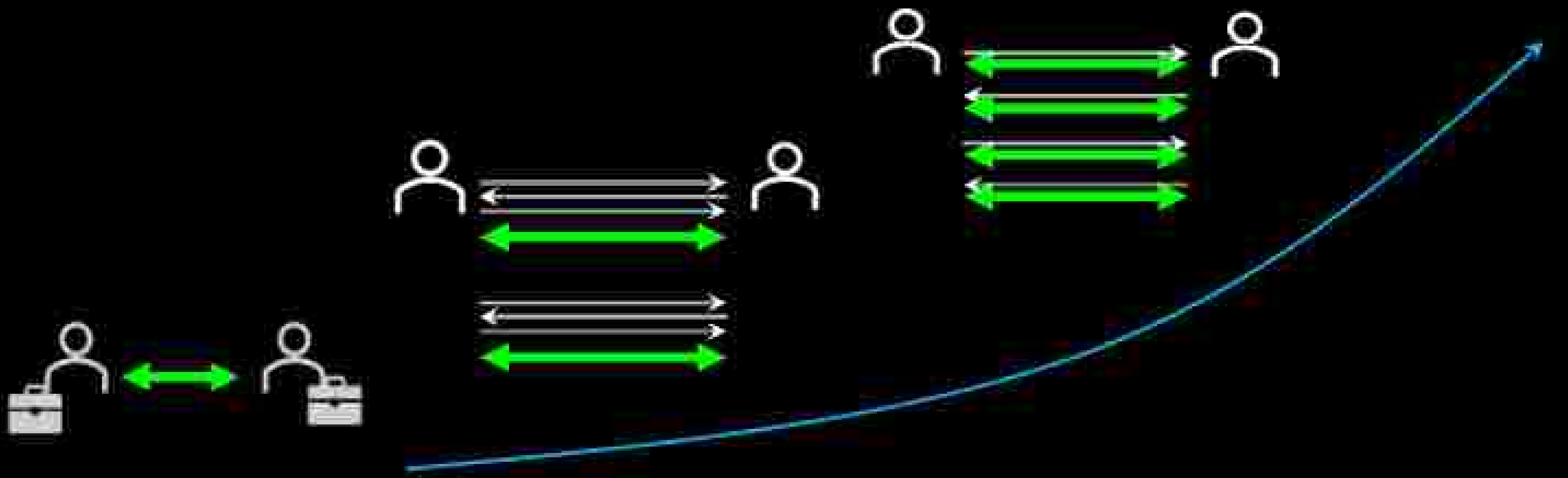


Forward and Post-Compromise End-to-End Asynchronous Multi-device, Low-Metadata ACKA Messaging with Man-in-the-Middle Detection

~~Forward and Post-Compromise End-to-End Asynchronous Multi-device ACKA Messaging with Man-in-the-Middle Detection~~

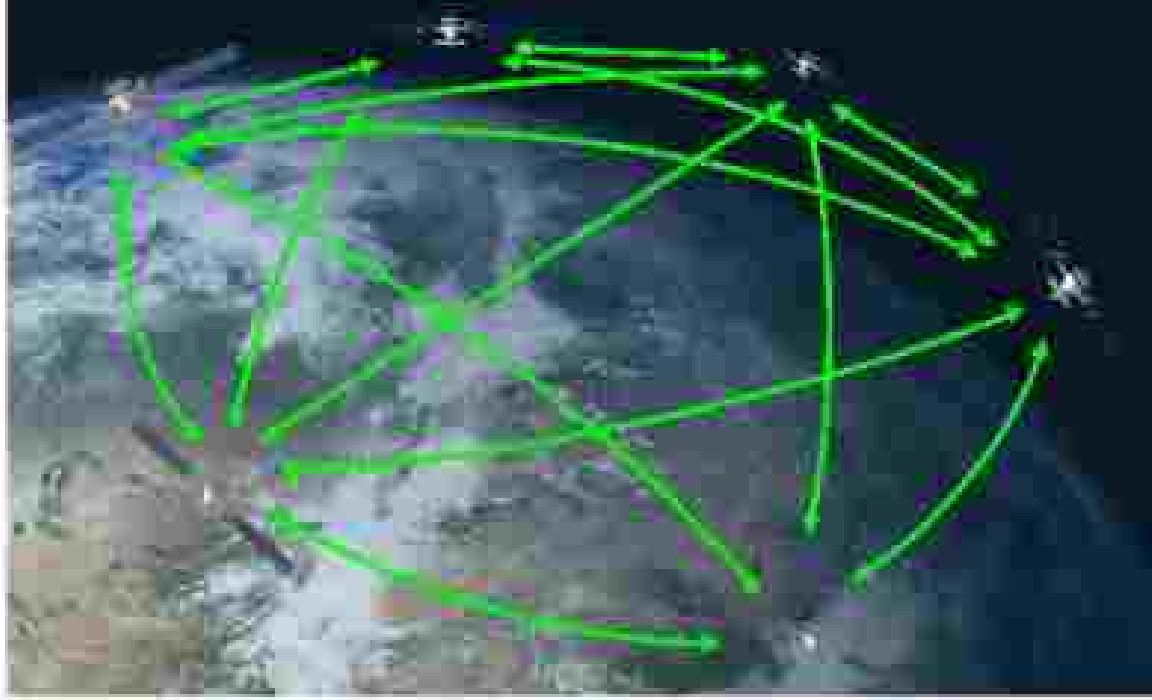
Forward and Post-Compromise End-to-End Asynchronous Multi-device, Low-Metadata ACKA Messaging with Man-in-the-Middle Detection

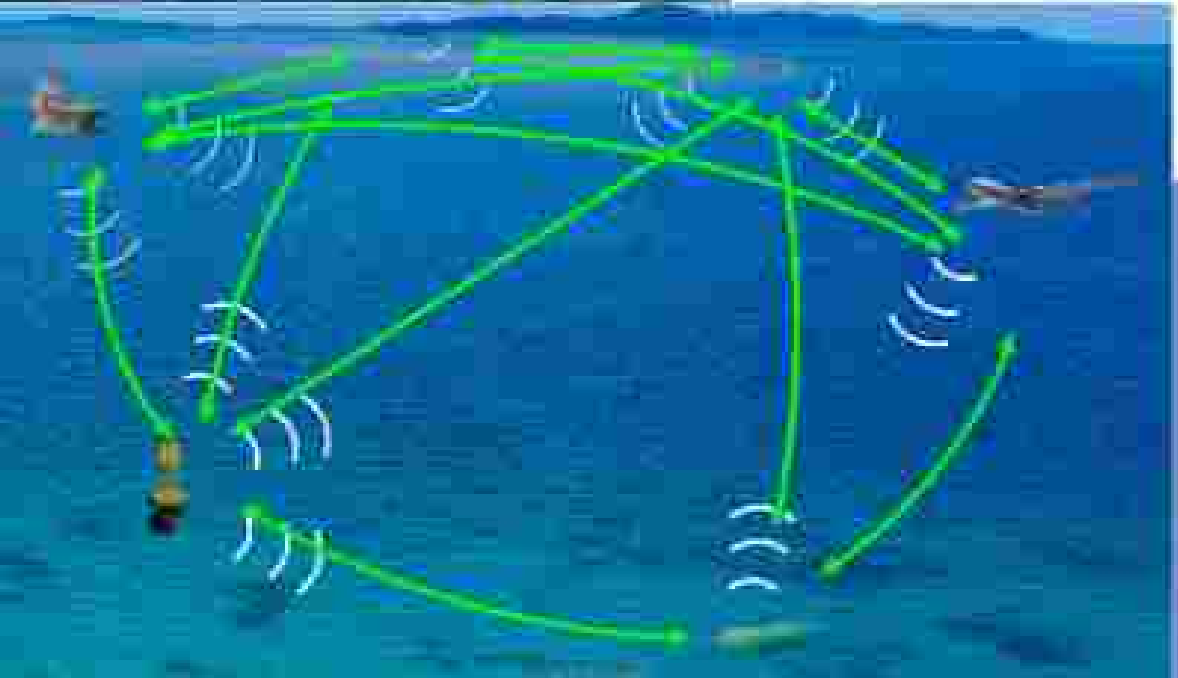
Forward and Post-Compromise End-to-End Asynchronous Multi-device, Low-Metadata ACKA [redacted] with Man-in-the-Middle Detection



group-centric
design

Space Systems





Unmanned Systems

Summary:

Attacks and subversion methods are continuously changing → security is a moving target

Cryptography should meet that challenge but can also be applied in unanticipated ways

Britta Hale
britta.hale@nps.edu