Definitional Foundations of Ratcheting and their Impact on Practice

Workshop on Secure Messaging – Eurocrypt 2019

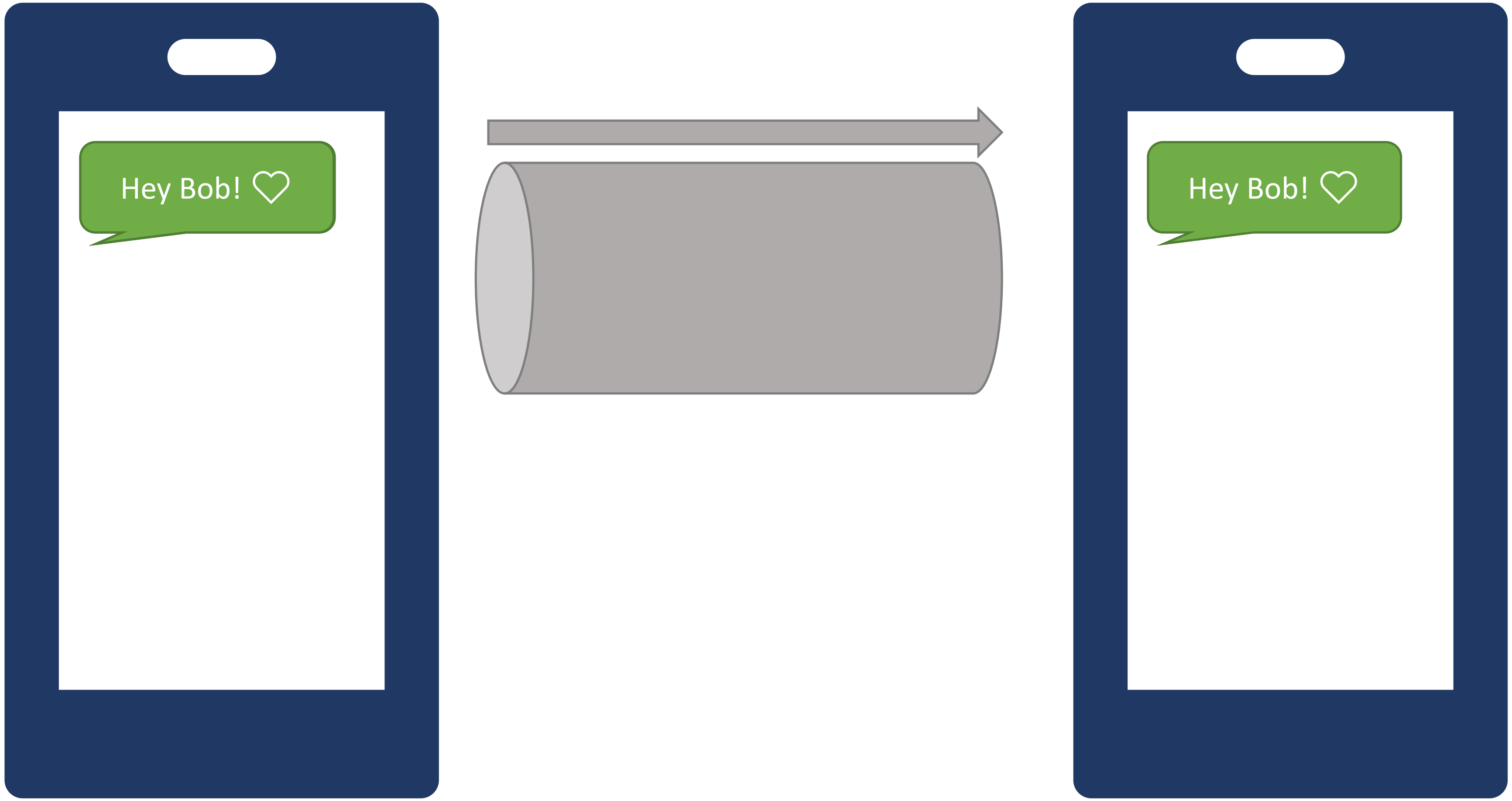
2019-05-18

Horst Görtz Institute for IT Security
Chair for Network and Data Security
Ruhr University Bochum
Paul Rösler

RUHR UNIVERSITÄT BOCHUM RUB

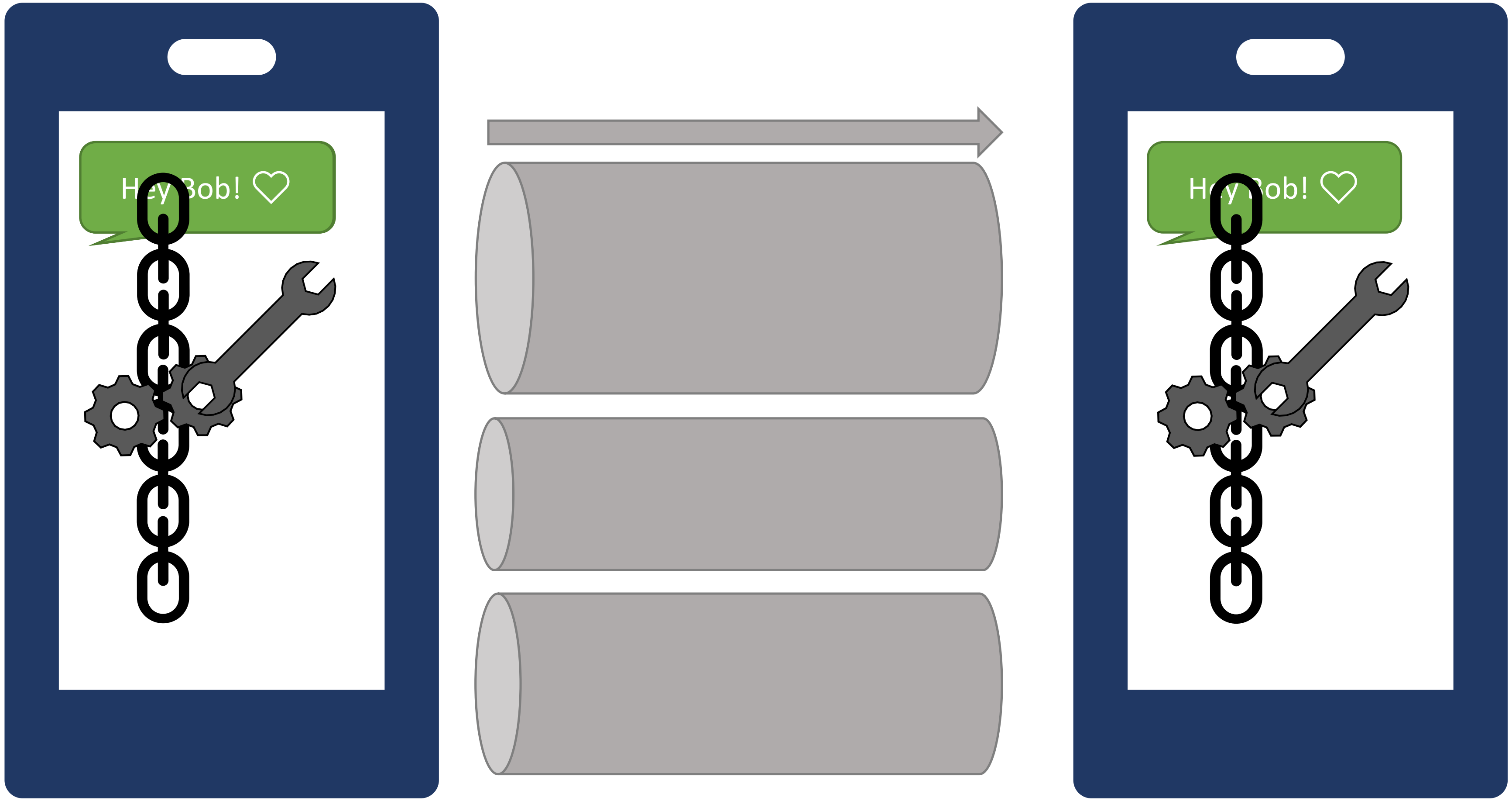
Messaging is complex

- (Asynchronous) session initialization
- “Secure” channel



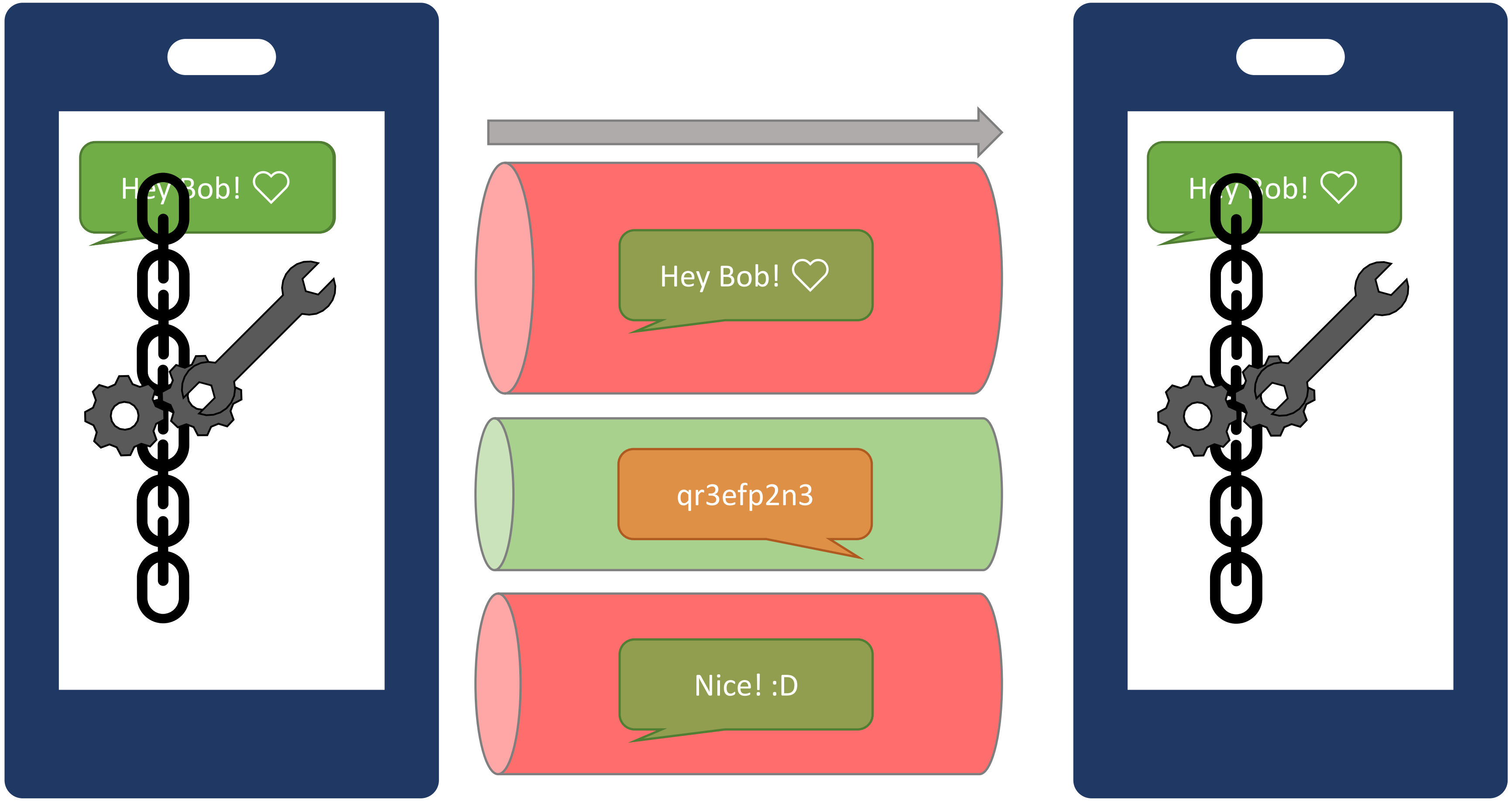
Messaging is complex

- (Asynchronous) session initialization
- “Secure” channel
- Strong security



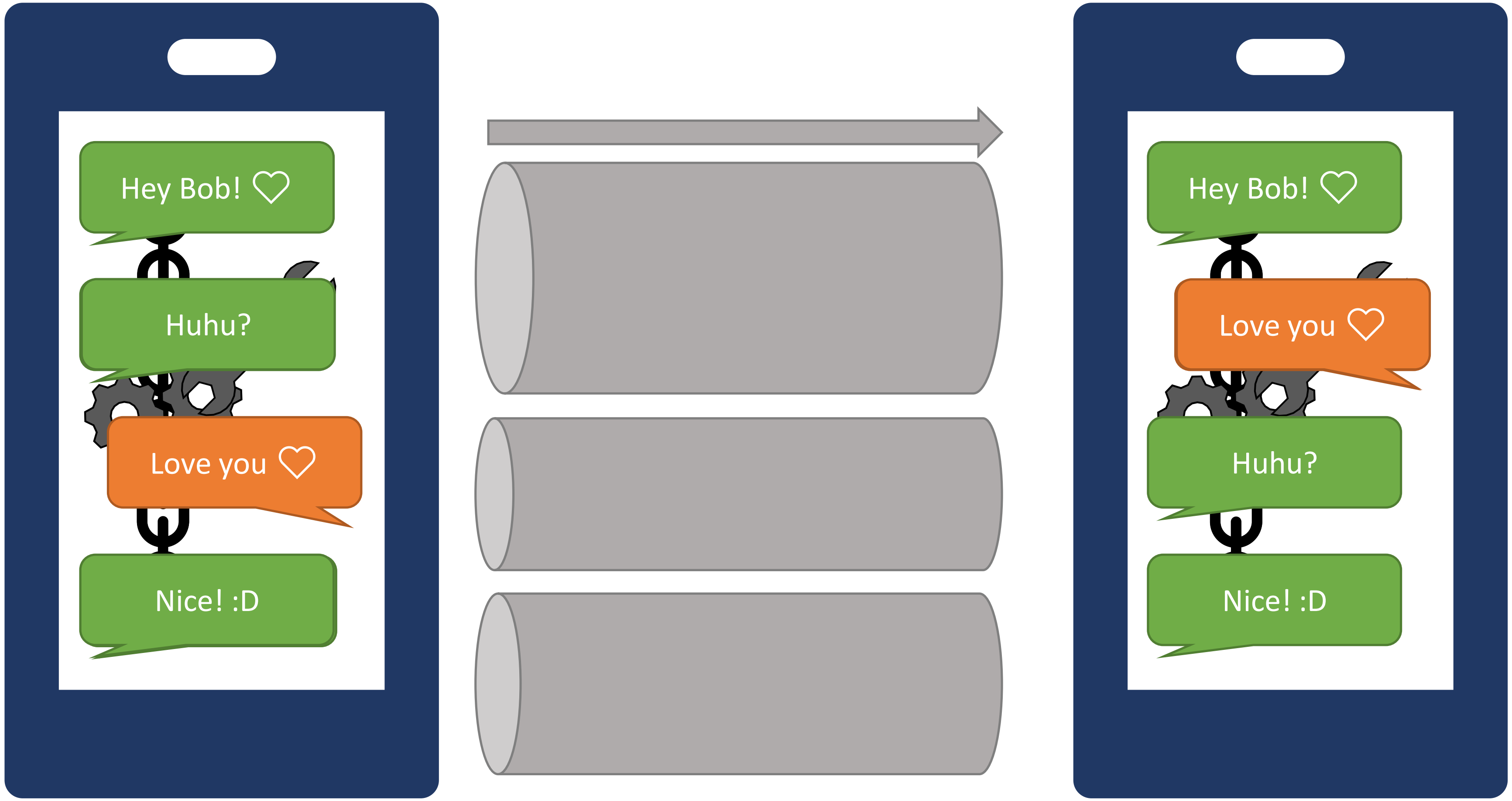
Messaging is complex

- (Asynchronous) session initialization
- “Secure” channel
- Strong security



Messaging is complex

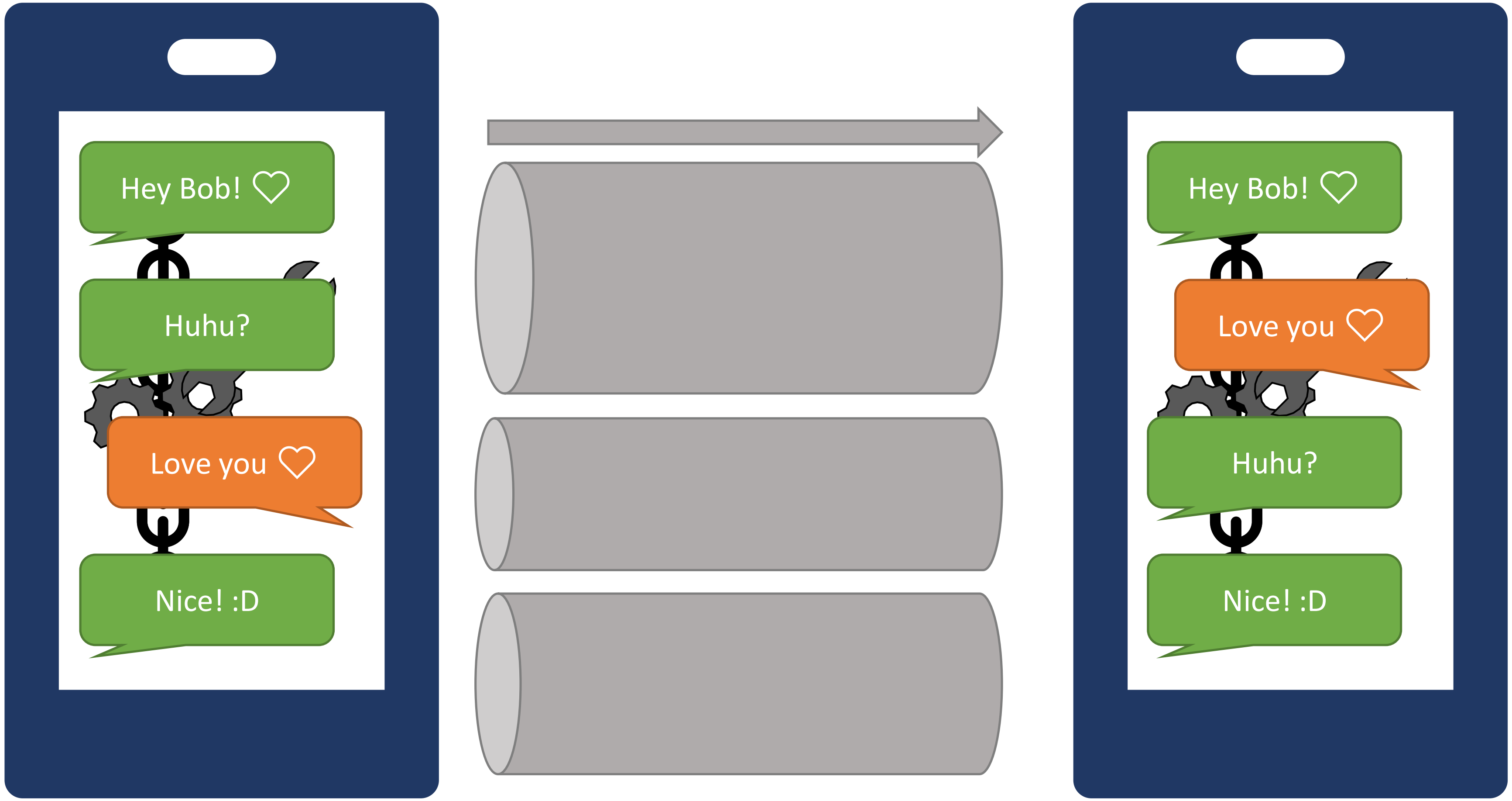
- (Asynchronous) session initialization
- “Secure” channel
- Strong security
- Concurrent communication



RUHR UNIVERSITÄT BOCHUM RUB

Messaging is complex

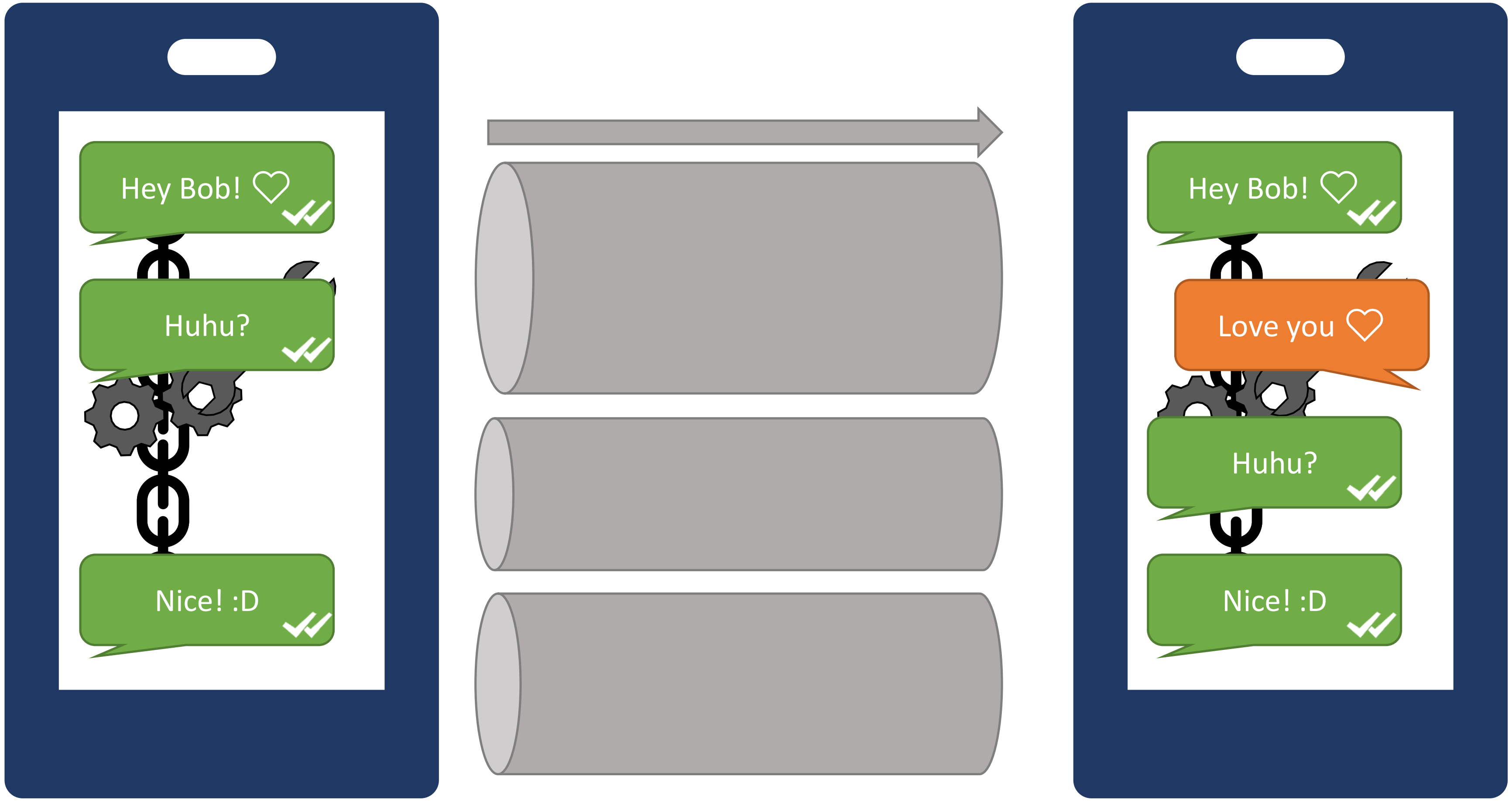
- (Asynchronous) session initialization
- “Secure” channel
- Strong security
- Concurrent communication
- Unreliable network



RUHR UNIVERSITÄT BOCHUM RUB

Messaging is complex

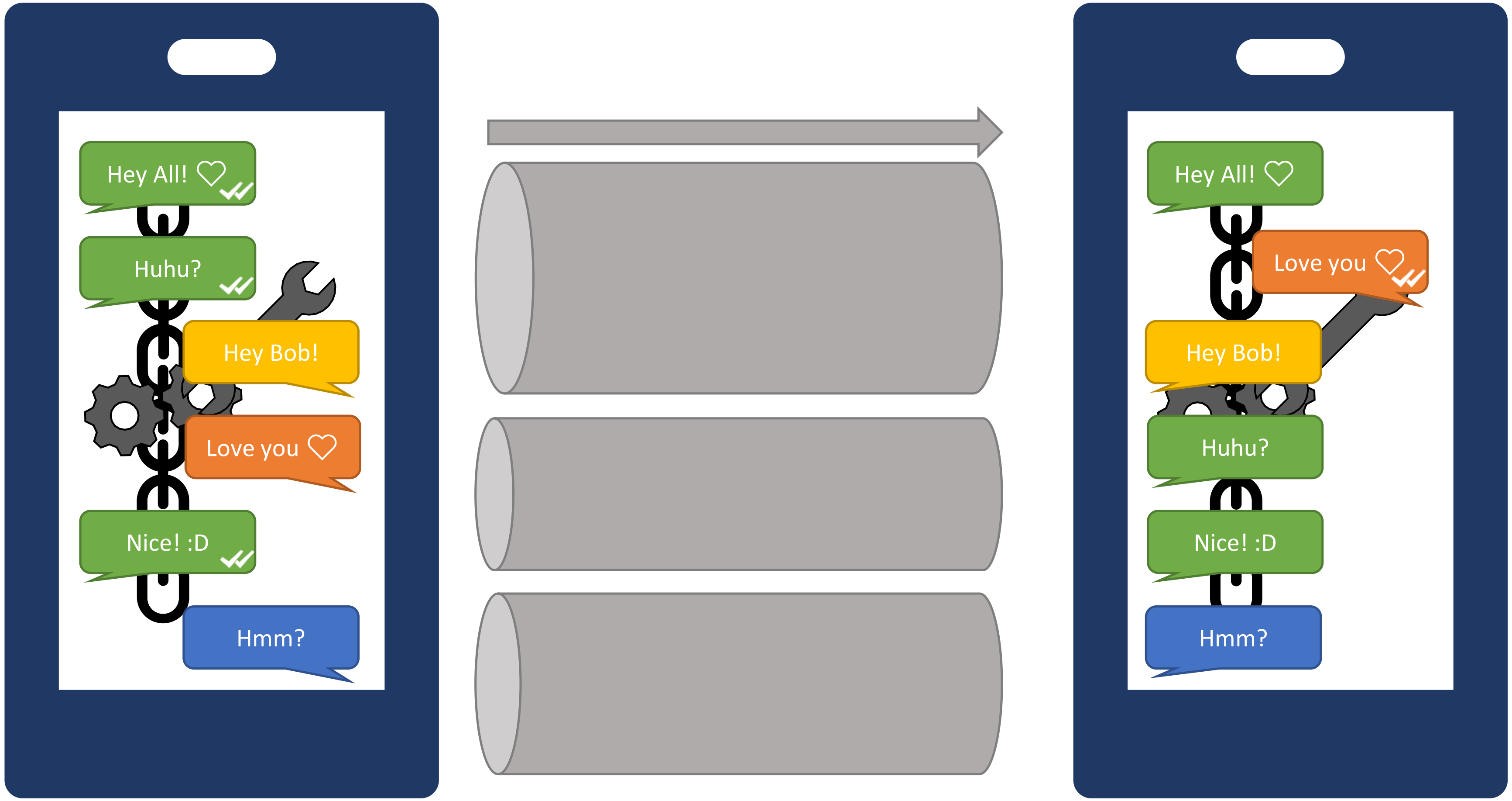
- (Asynchronous) session initialization
- "Secure" channel
- Strong security
- Concurrent communication
- Unreliable network
- Explicit reliability



RUHR UNIVERSITÄT BOCHUM

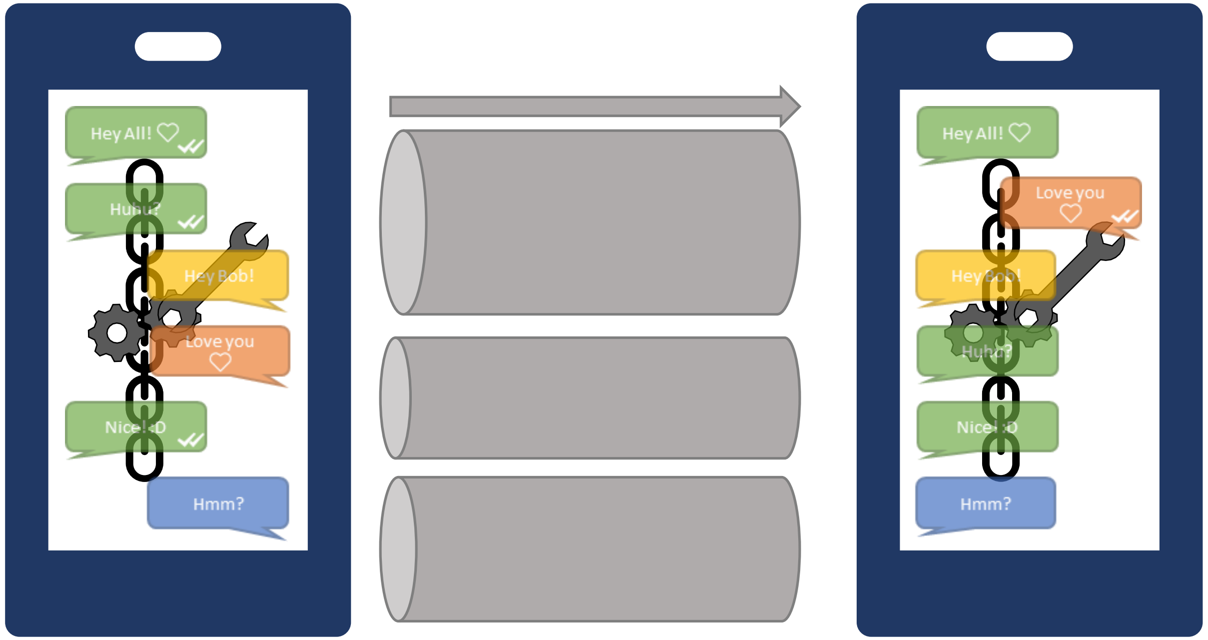
Messaging is complex

- (Asynchronous) session initialization
- “Secure” channel
- Strong security
- Concurrent communication
- Unreliable network
- Explicit reliability
- Group communication



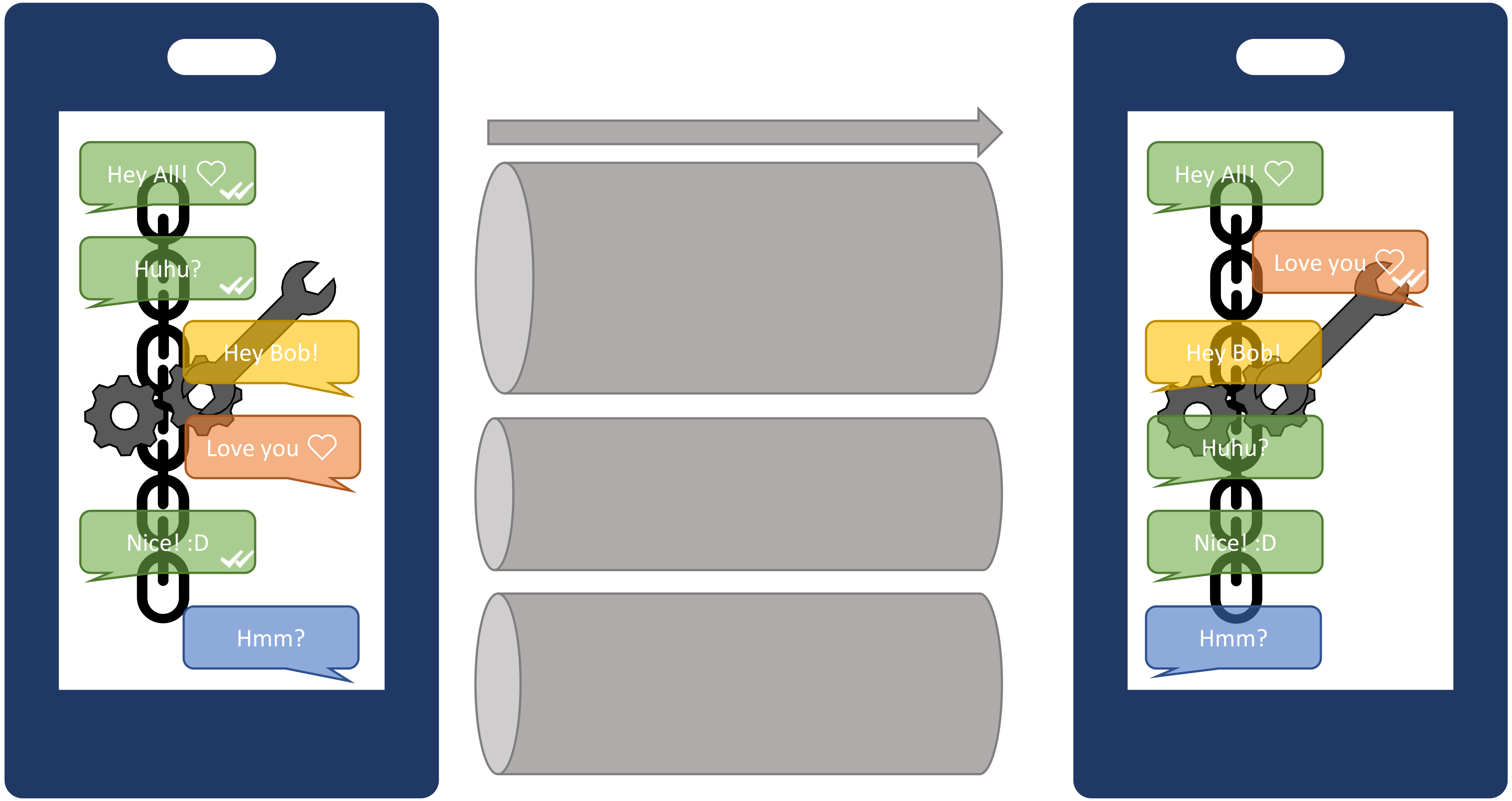
Agenda

- **Messaging is complex**
  ⇒ Comprehensible science helps
- Finding a Syntax
- Understanding Attackers
- Defining Security
- Core Primitive of Ratcheting (of strongly secure Messaging)



Messaging is complex

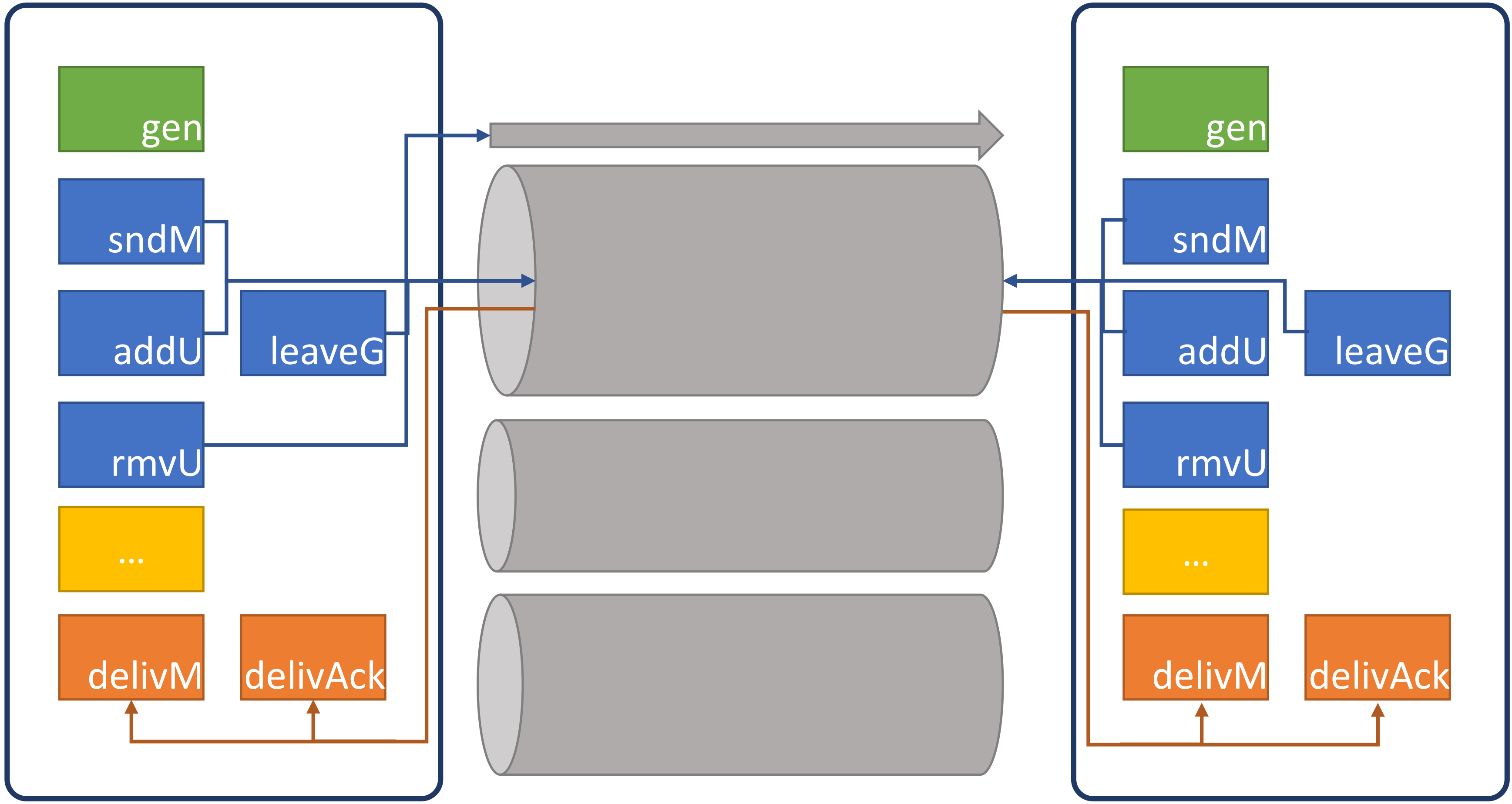
- (Asynchronous) session initialization
- “Secure” channel
- Strong security
- Concurrent communication
- Unreliable network
- Explicit reliability
- Group communication



RUHR UNIVERSITÄT BOCHUM RUB

Messaging is complex

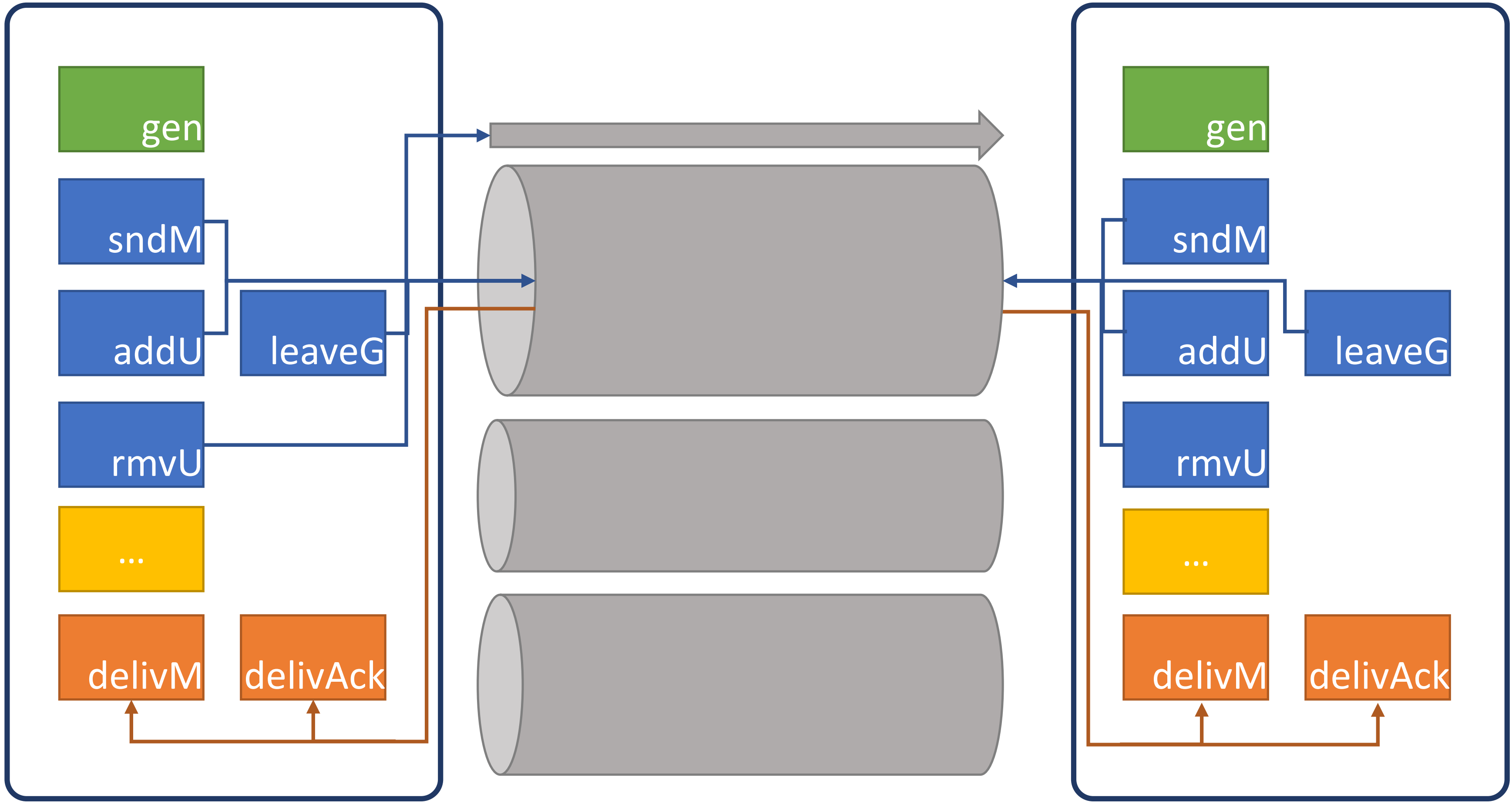
- (Asynchronous) session initialization
- “Secure” channel
- Strong security
- Concurrent communication
- Unreliable network
- Explicit reliability
- Group communication



RUHR UNIVERSITÄT BOCHUM RUB

Messaging is complex

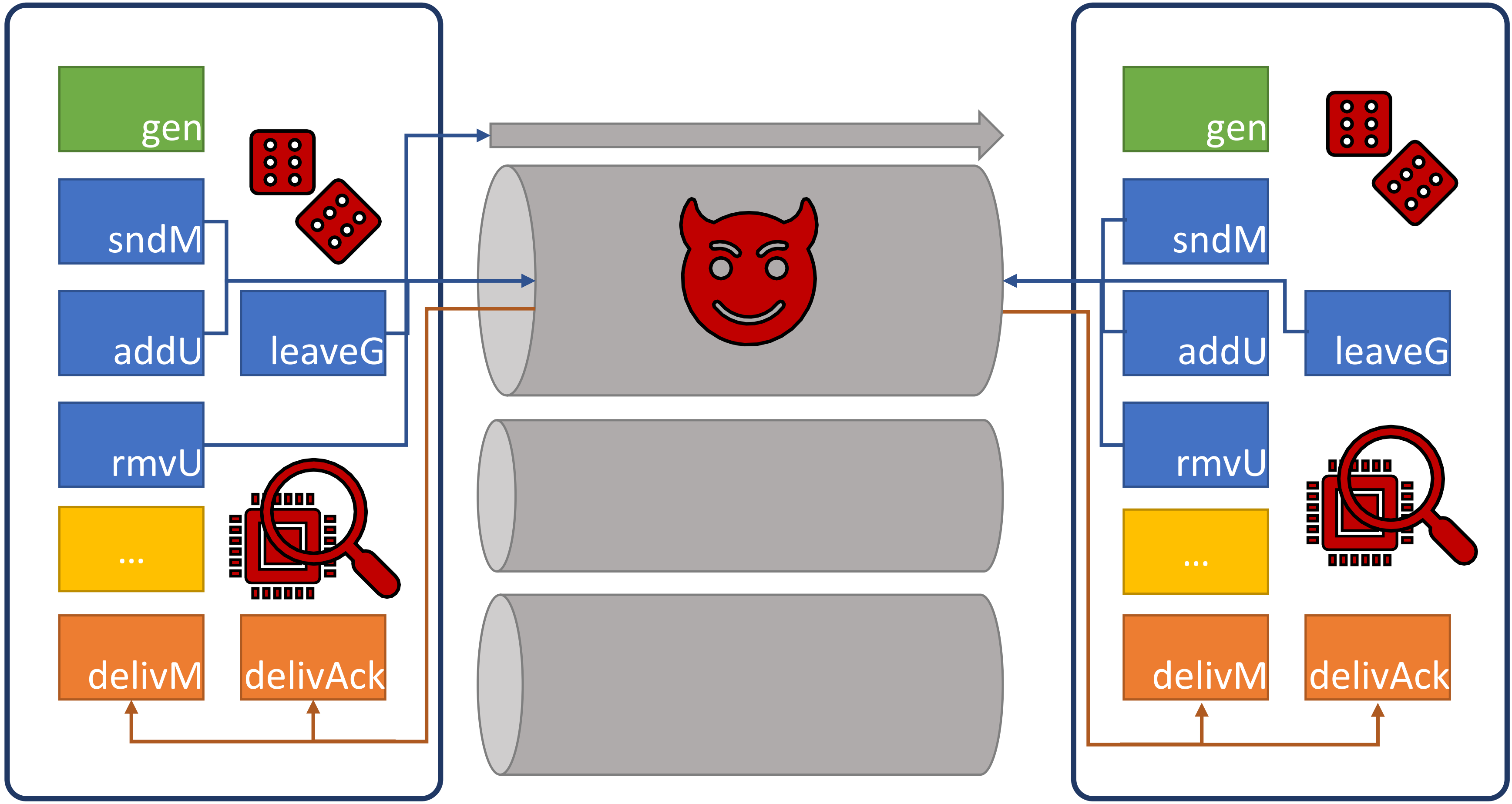
- Complex syntax definition



RUHR UNIVERSITÄT BOCHUM RUB

Messaging is complex

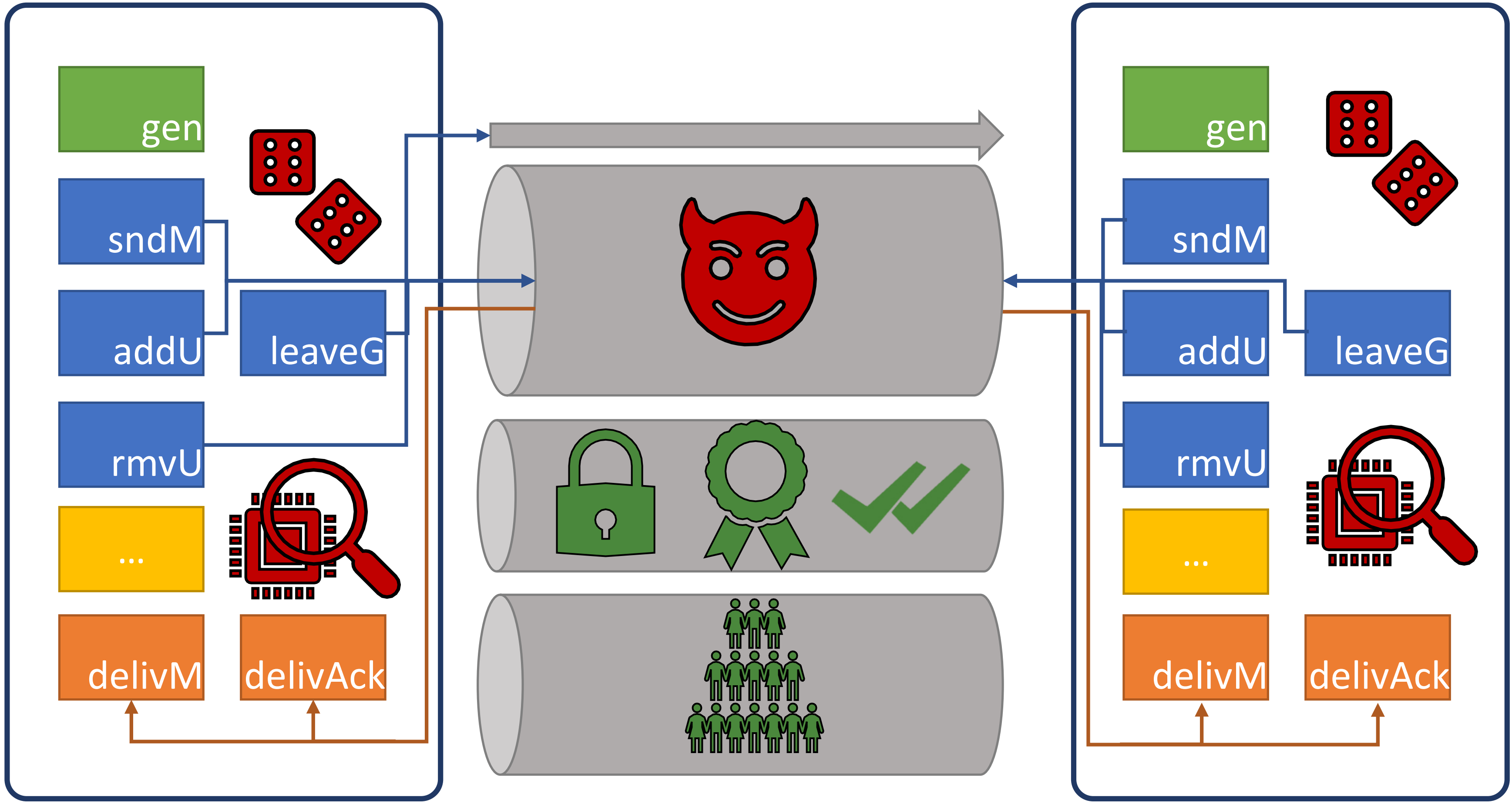
- Complex syntax definition
- Strong attacker
  - Active MitM
  - Exposure of device's secrets
  - Execution's random coins might be weak
  - …



RUHR UNIVERSITÄT BOCHUM RUB

Messaging is complex

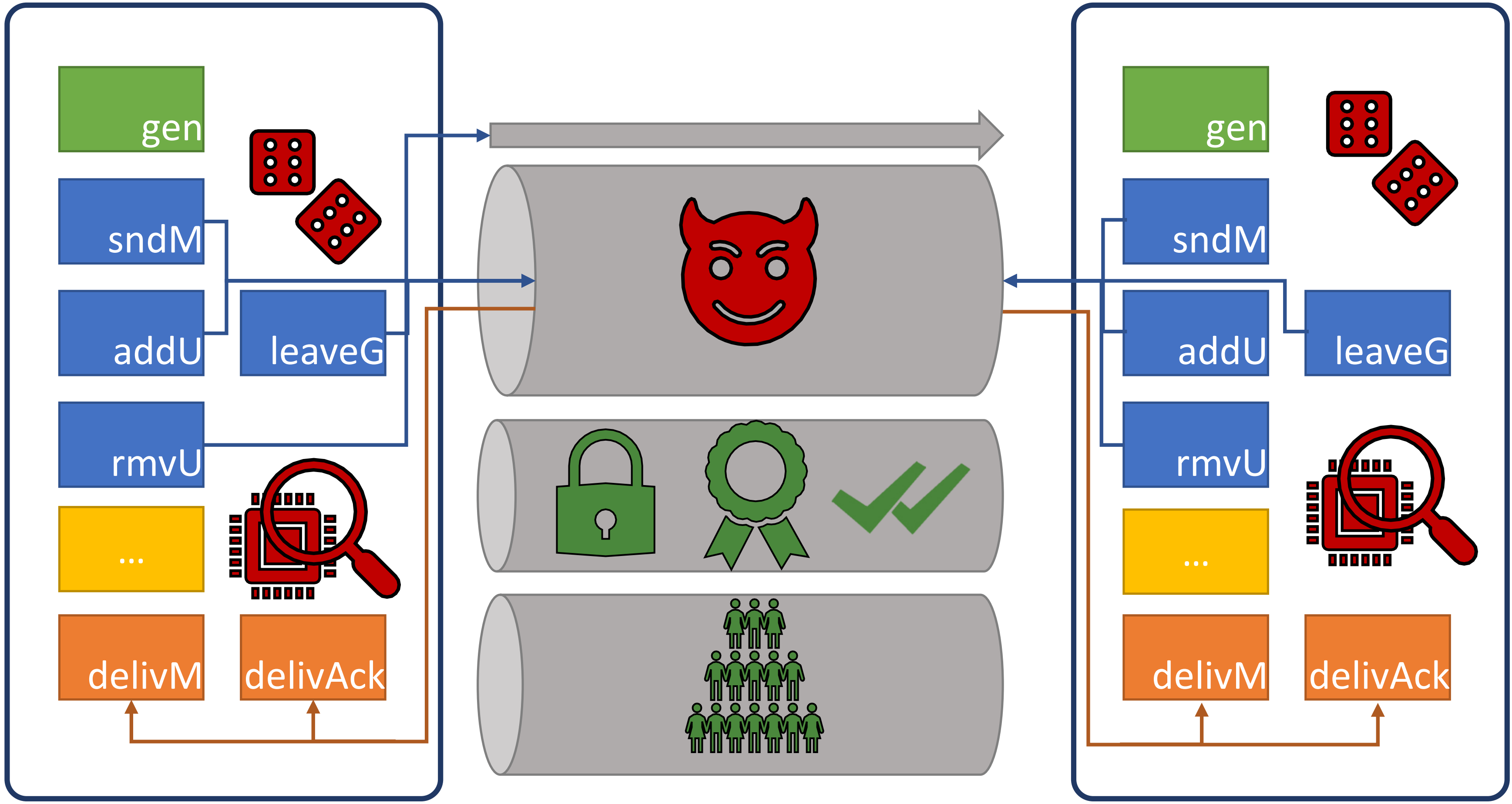
- Complex syntax definition
- Strong attacker
- Multiple security properties
  - Confidentiality
  - Authenticity
  - Reliable acks
  - Secure group management



Messaging is complex

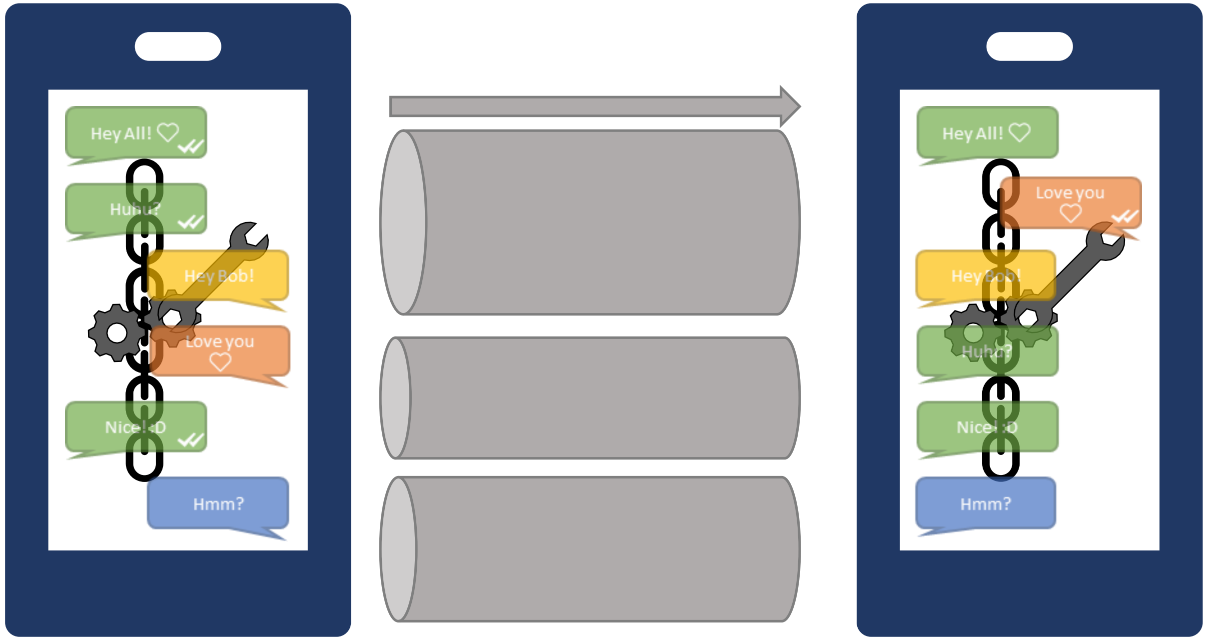
- Complex syntax definition
- Strong attacker
- Multiple security properties

⇒ Single model to analyze security?



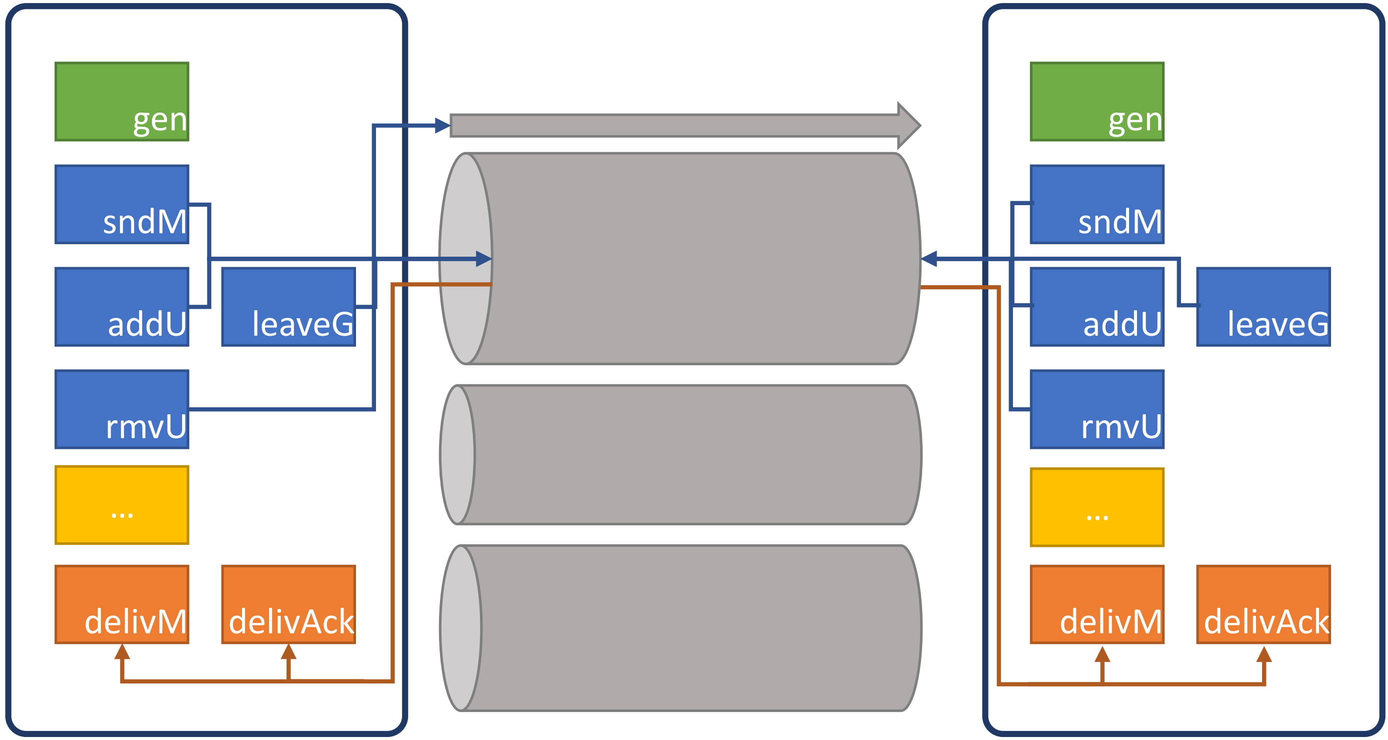
Agenda

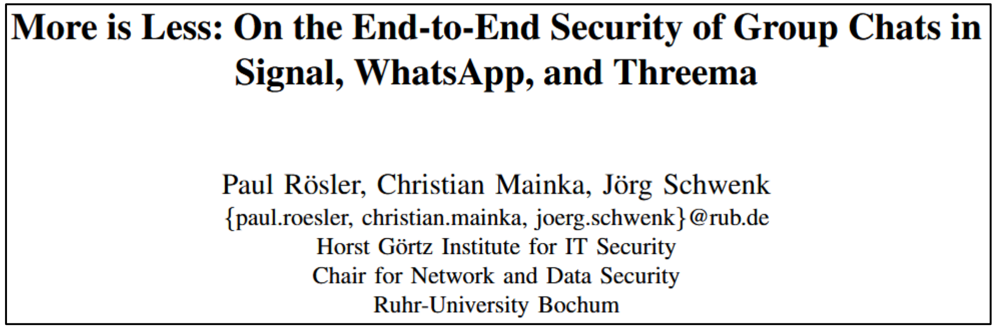
- Messaging is complex
- **Finding a Syntax**
- Understanding Attackers
- Defining Security
- Core Primitive of RKE



Syntax – Taming complexity

- Messenger with
  - Two-party channels
  - Delivery notifications
  - Group channels
  - Group management



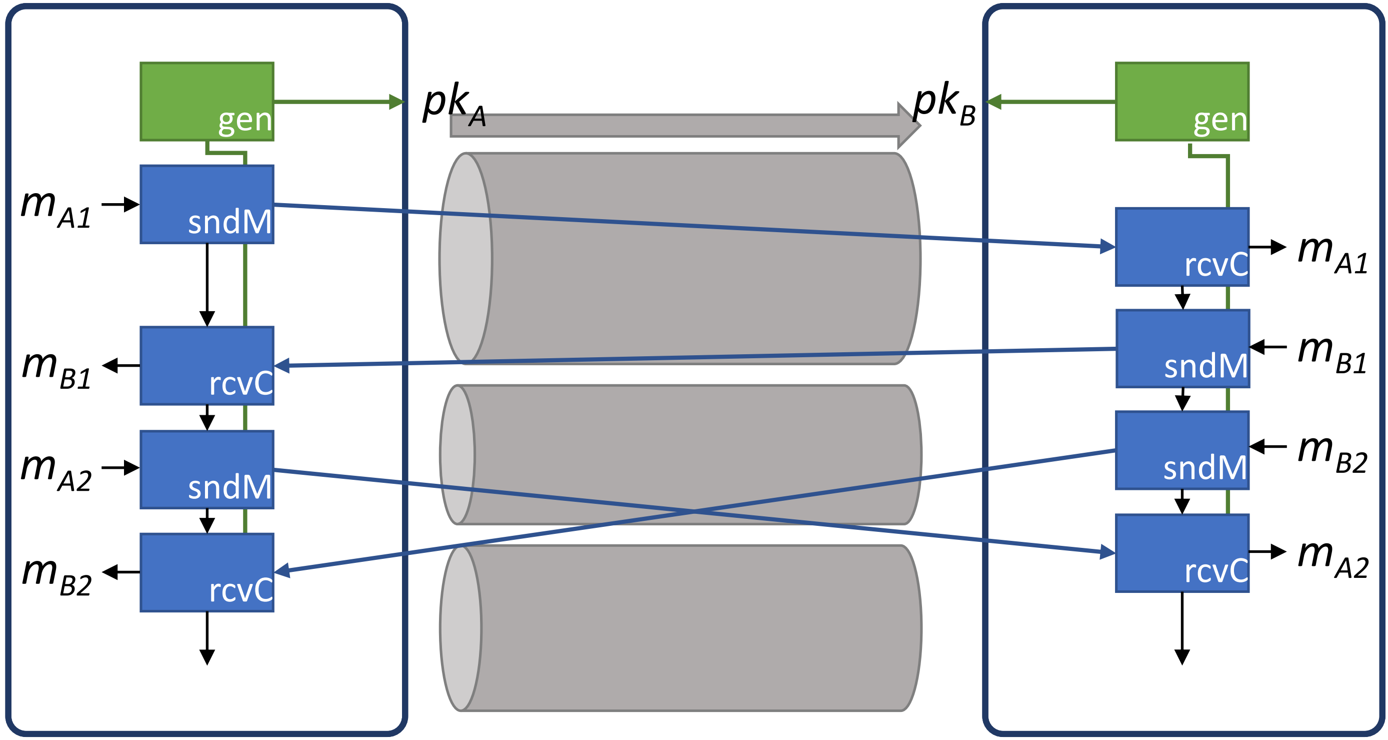

More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema

Paul Rösler, Christian Mainka, Jörg Schwenk
{paul.roesler, christian.mainka, joerg.schwenk}@rub.de
Horst Görtz Institute for IT Security
Chair for Network and Data Security
Ruhr-University Bochum

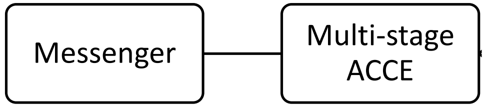
Messenger

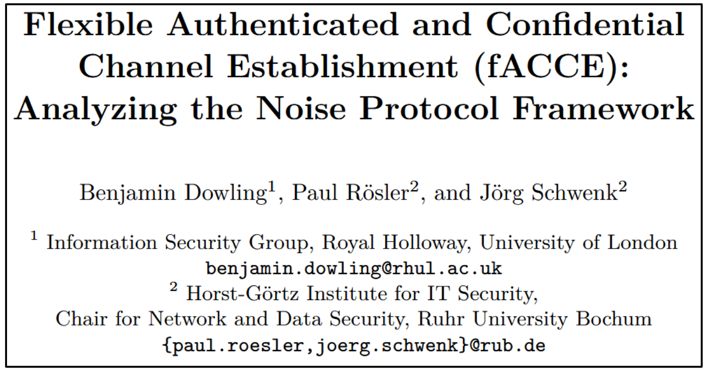
Syntax – Taming complexity

- Remove:
  1. Delivery notifications
  2. Group channels
  3. Group management



Two-party channel establishment
("Multi-stage ACCE")



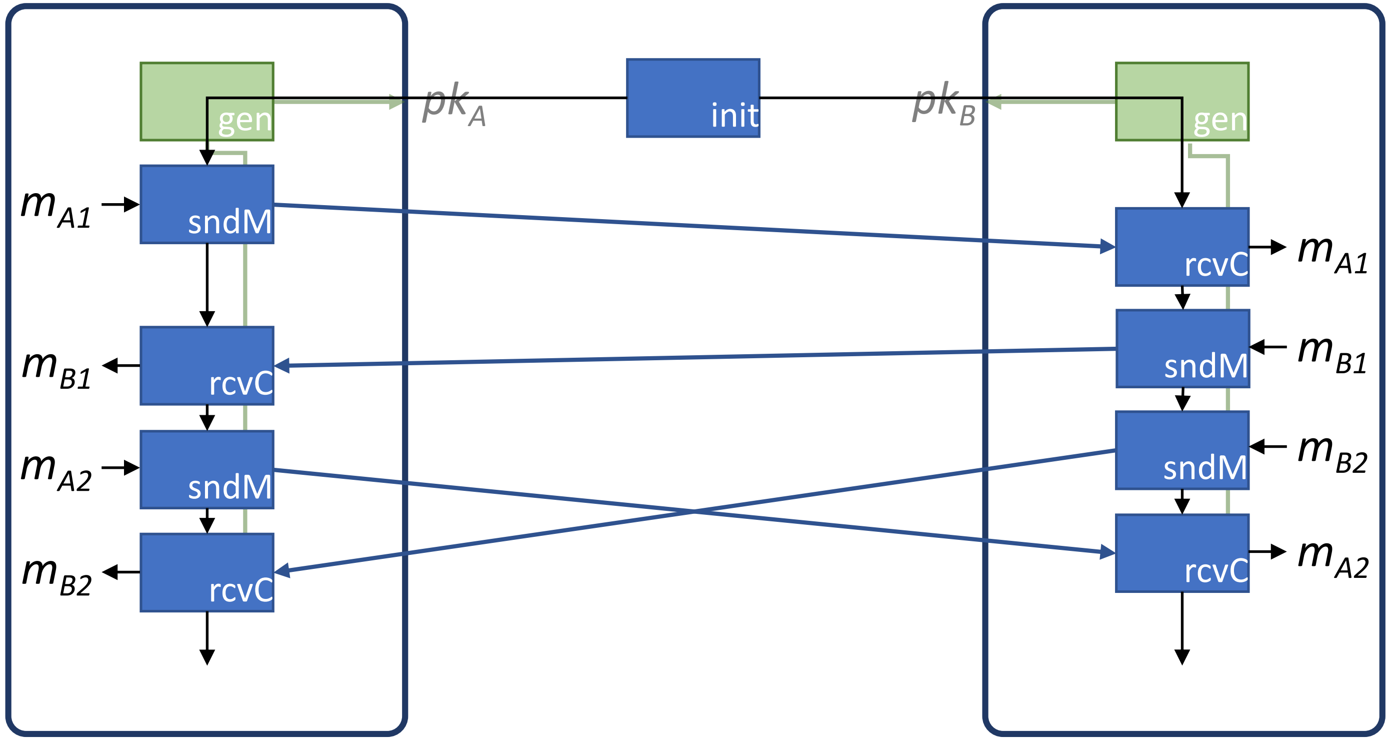

Flexible Authenticated and Confidential Channel Establishment (fACCE): Analyzing the Noise Protocol Framework

Benjamin Dowling[1], Paul Rösler[2], and Jörg Schwenk[2]

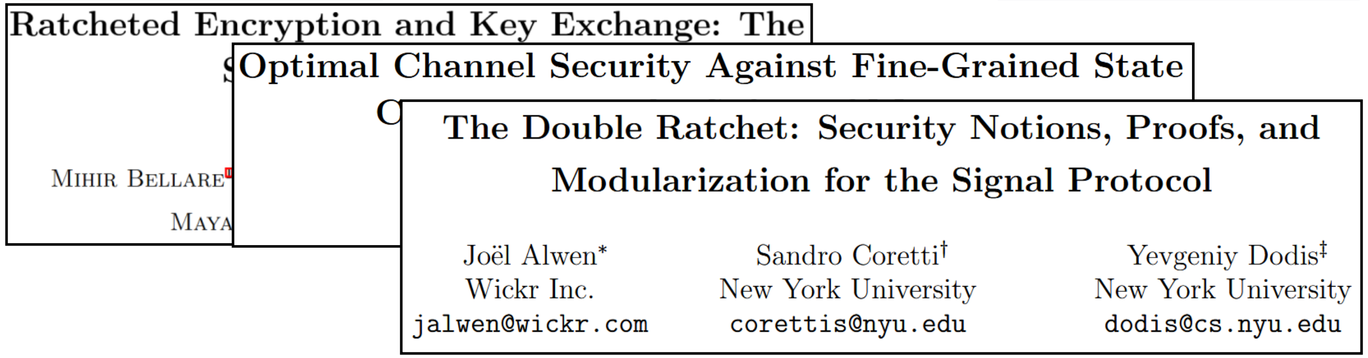
[1] Information Security Group, Royal Holloway, University of London
benjamin.dowling@rhul.ac.uk
[2] Horst-Görtz Institute for IT Security,
Chair for Network and Data Security, Ruhr University Bochum
{paul.roesler,joerg.schwenk}@rub.de

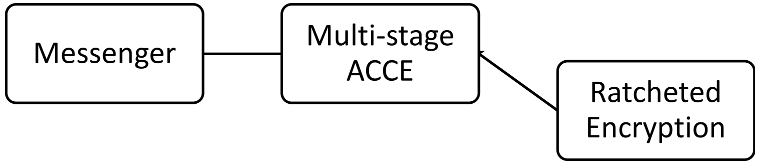
Syntax – Taming complexity

- Remove:
  1. Delivery notifications
  2. Group channels
  3. Group management
  4. Channel establishment



Ratcheted encryption

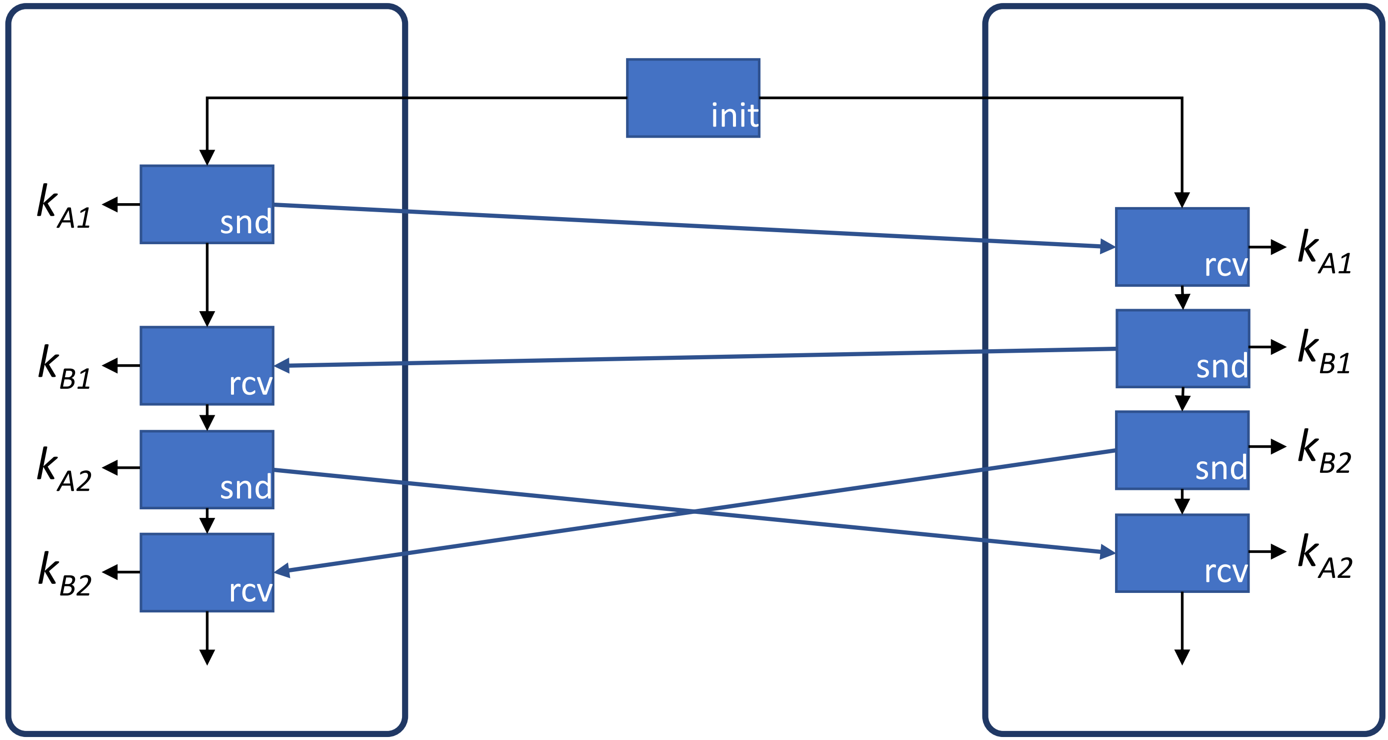

Ratcheted Encryption and Key Exchange: The
Optimal Channel Security Against Fine-Grained State
The Double Ratchet: Security Notions, Proofs, and Modularization for the Signal Protocol
Mihir Bellare
Maya
Joël Alwen* Wickr Inc. jalwen@wickr.com
Sandro Coretti† New York University corettis@nyu.edu
Yevgeniy Dodis‡ New York University dodis@cs.nyu.edu



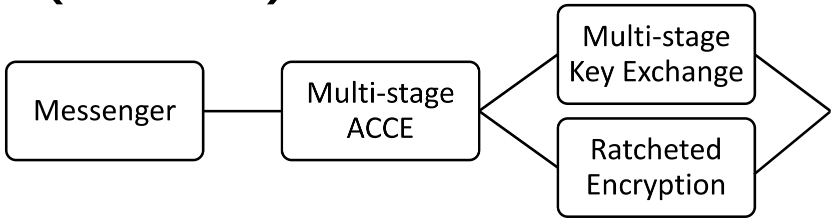
RUHR UNIVERSITÄT BOCHUM RUB

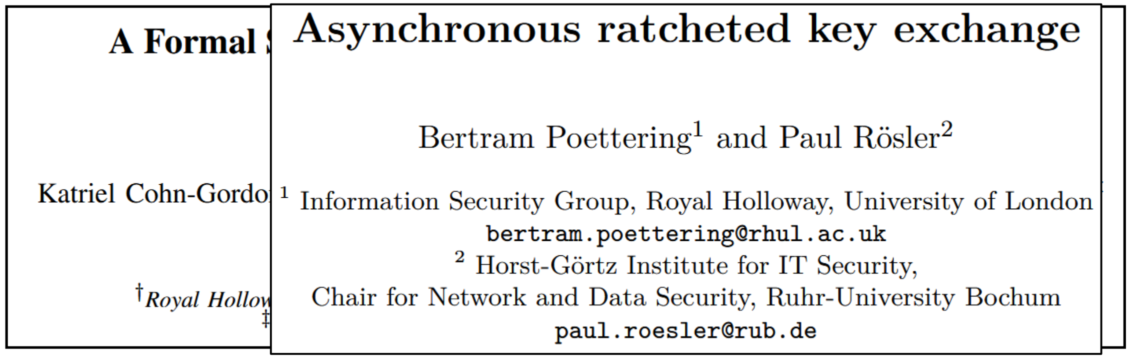
Syntax – Taming complexity

- Remove:
  1. Delivery notifications
  2. Group channels
  3. Group management
  4. Channel establishment
  5. Symmetric encryption



Bidirectional ratcheted key exchange (BRKE)




A Formal
Katriel Cohn-Gordo
†Royal Hollow

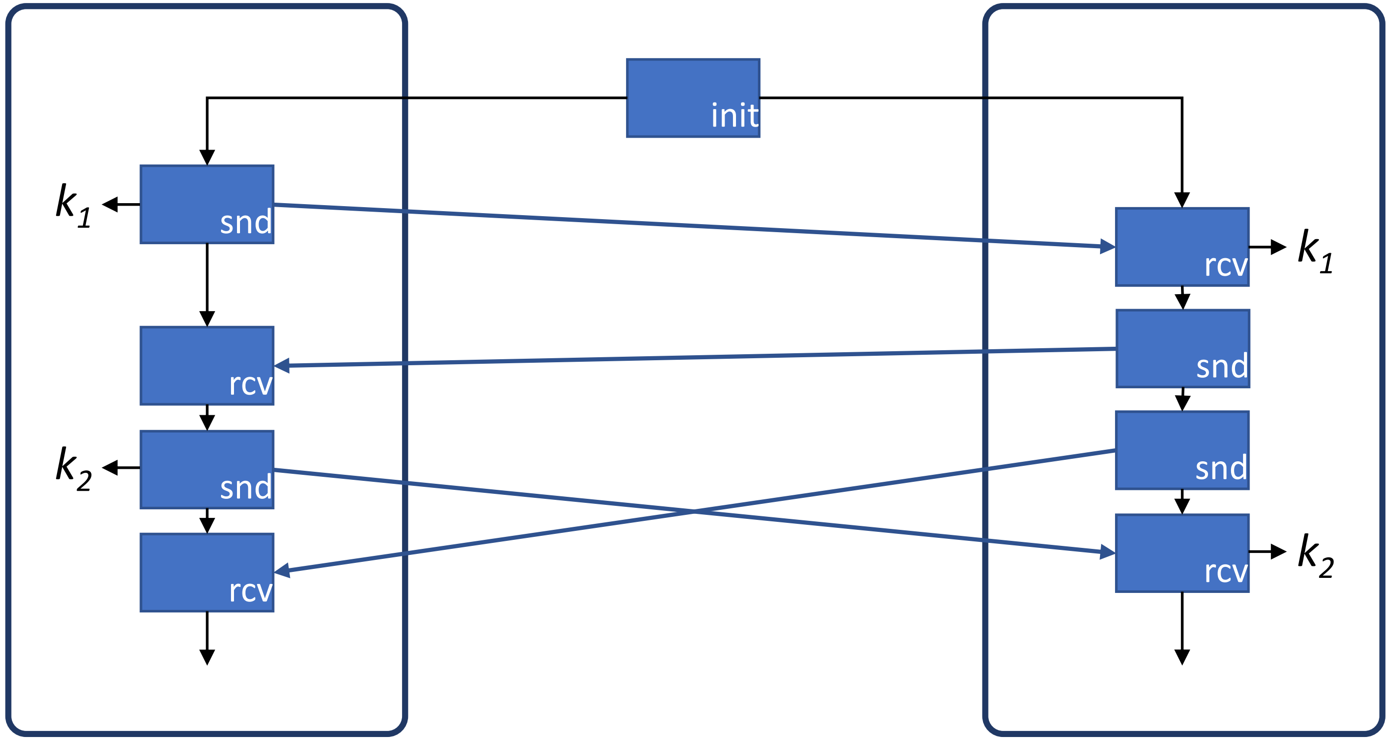
Asynchronous ratcheted key exchange

Bertram Poettering[1] and Paul Rösler[2]

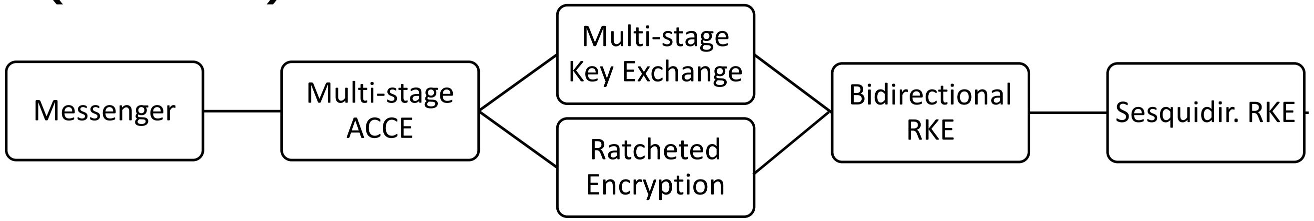
[1] Information Security Group, Royal Holloway, University of London
bertram.poettering@rhul.ac.uk
[2] Horst-Görtz Institute for IT Security,
Chair for Network and Data Security, Ruhr-University Bochum
paul.roesler@rub.de

Syntax – Taming complexity

- Remove:
  1. Delivery notifications
  2. Group channels
  3. Group management
  4. Channel establishment
  5. Symmetric encryption
  6. Key establishment B-to-A



Sesquidirectional ratcheted key exchange (SRKE)



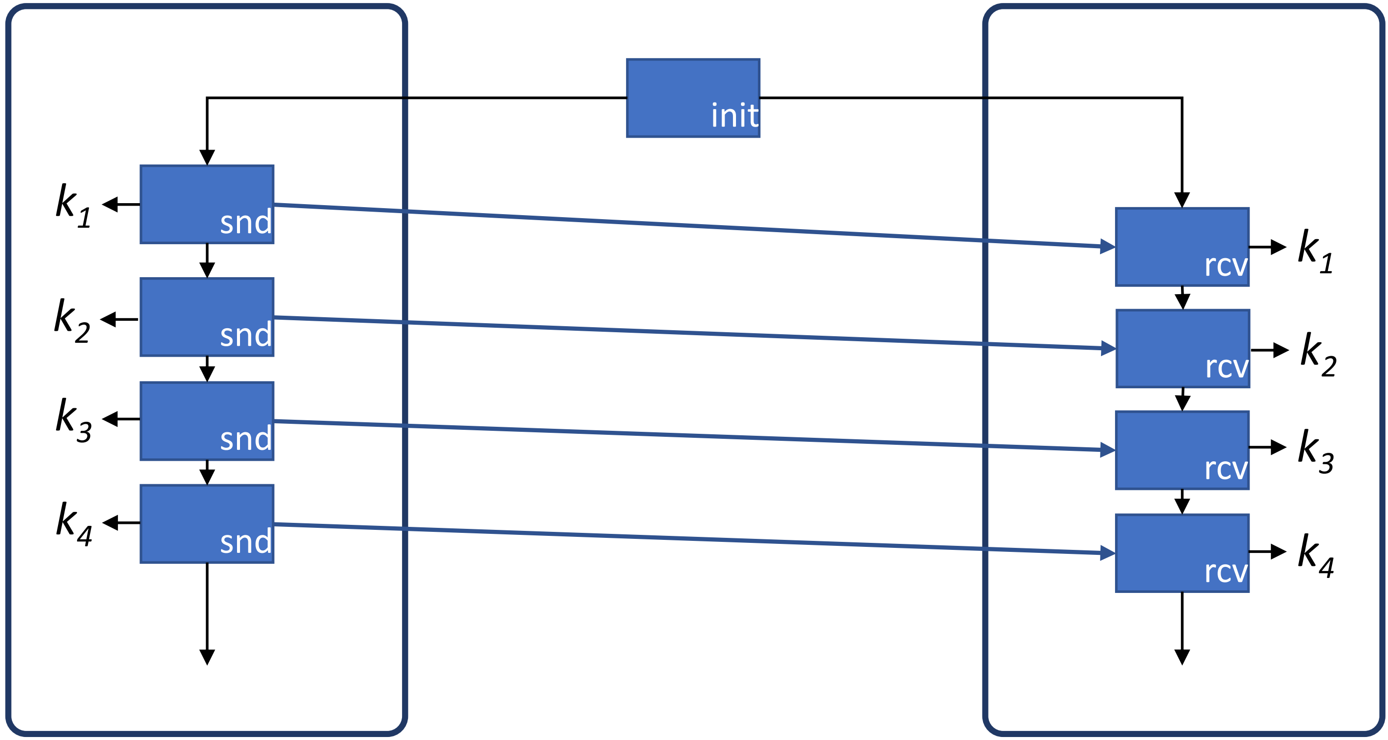
Asynchronous ratcheted key exchange

Bertram Poettering[1] and Paul Rösler[2]

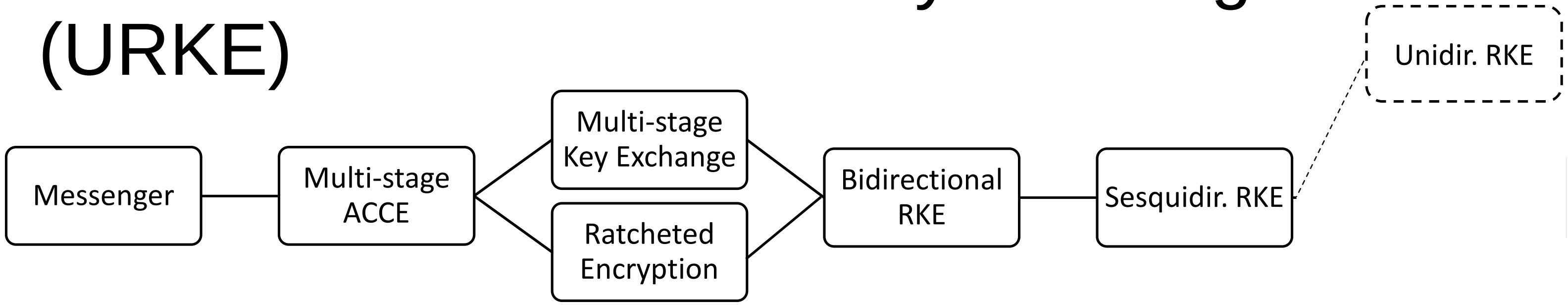
[1] Information Security Group, Royal Holloway, University of London
bertram.poettering@rhul.ac.uk
[2] Horst-Görtz Institute for IT Security,
Chair for Network and Data Security, Ruhr-University Bochum
paul.roesler@rub.de

Syntax – Taming complexity

- Remove:
  1. Delivery notifications
  2. Group channels
  3. Group management
  4. Channel establishment
  5. Symmetric encryption
  6. Key establishment B-to-A
  7. B-to-A communication



Unidirectional ratcheted key exchange (URKE)



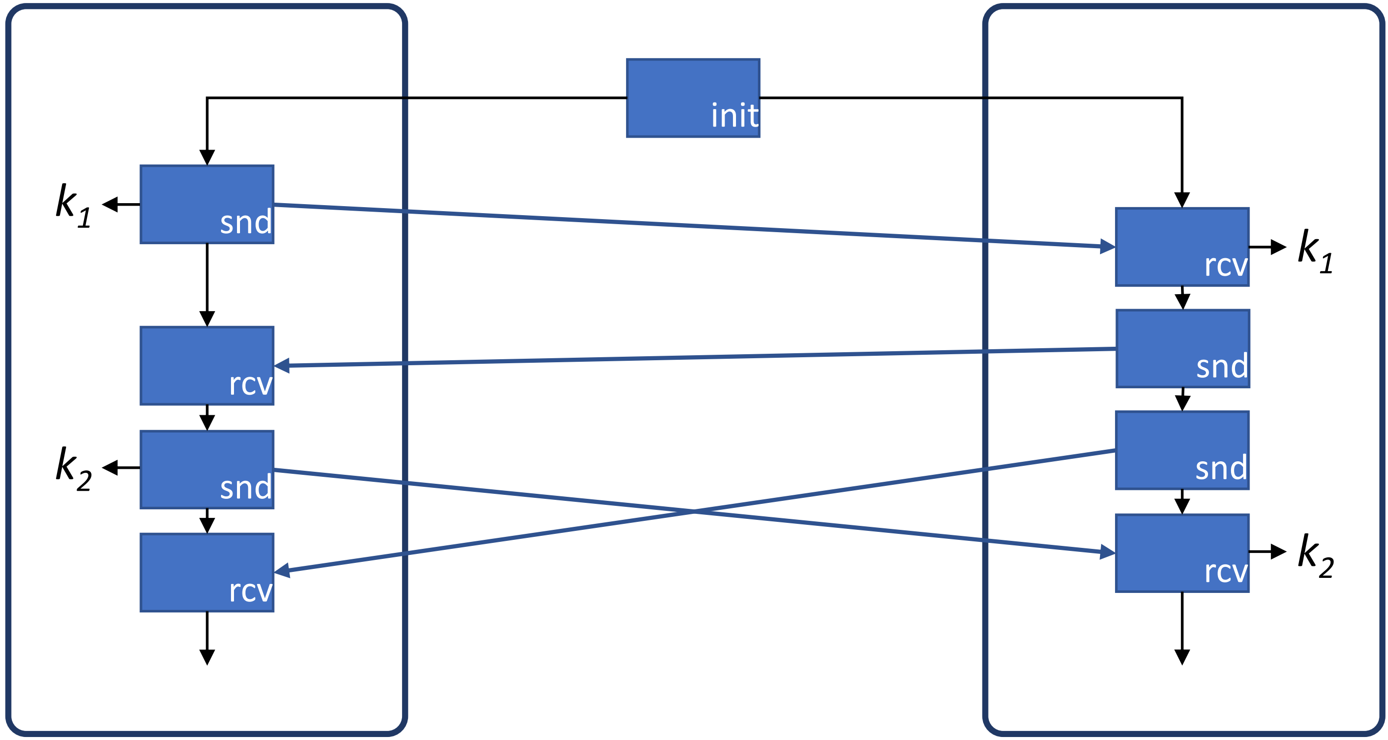

Asynchronous ratcheted key exchange

Bertram Poettering[1] and Paul Rösler[2]

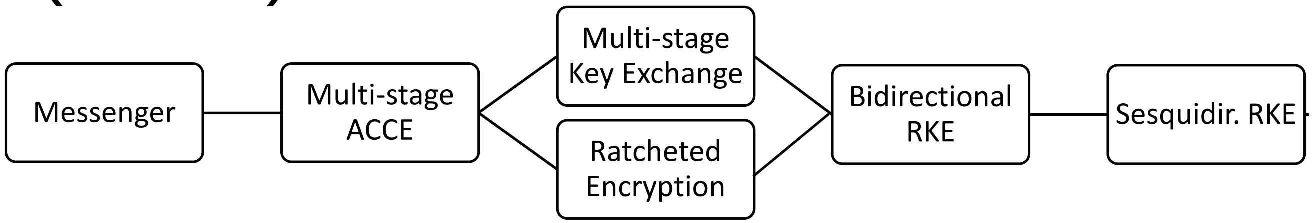
[1] Information Security Group, Royal Holloway, University of London
bertram.poettering@rhul.ac.uk
[2] Horst-Görtz Institute for IT Security,
Chair for Network and Data Security, Ruhr-University Bochum
paul.roesler@rub.de

Syntax – Taming complexity

- Remove:
  1. Delivery notifications
  2. Group channels
  3. Group management
  4. Channel establishment
  5. Symmetric encryption
  6. Key establishment B-to-A



Sesquidirectional ratcheted key exchange (SRKE)



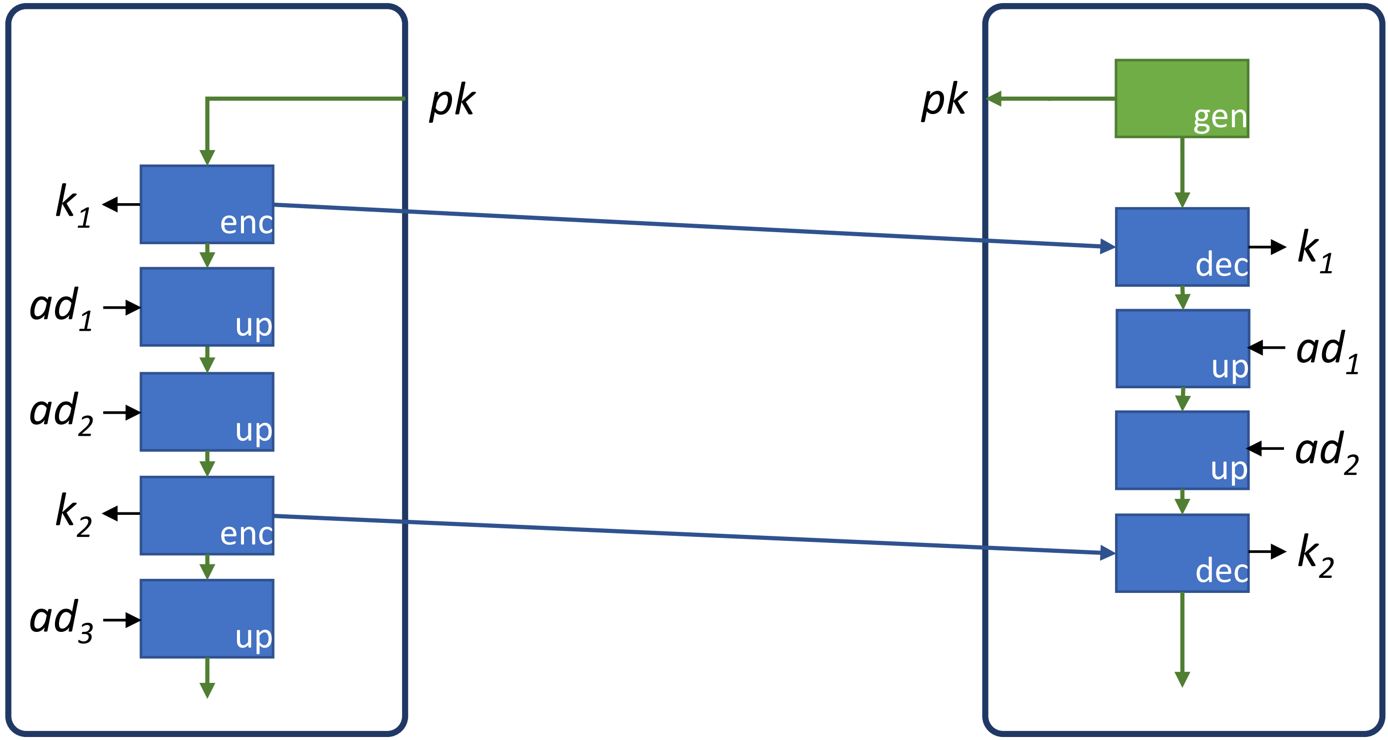
Asynchronous ratcheted key exchange

Bertram Poettering[1] and Paul Rösler[2]

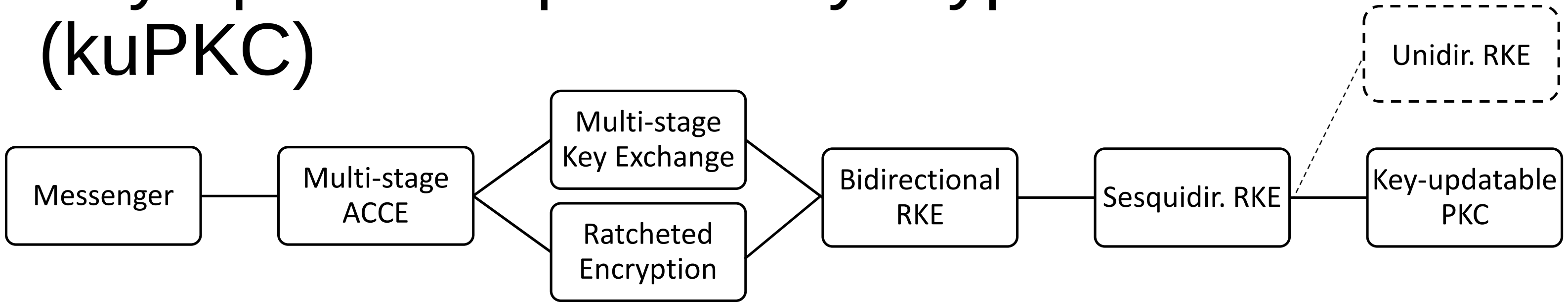
[1] Information Security Group, Royal Holloway, University of London
bertram.poettering@rhul.ac.uk
[2] Horst-Görtz Institute for IT Security,
Chair for Network and Data Security, Ruhr-University Bochum
paul.roesler@rub.de

Syntax – Taming complexity

- Remove:
  1. Delivery notifications
  2. Group channels
  3. Group management
  4. Channel establishment
  5. Symmetric encryption
  6. Key establishment B-to-A
  7. Interaction



Key-updatable public key crypto (kuPKC)



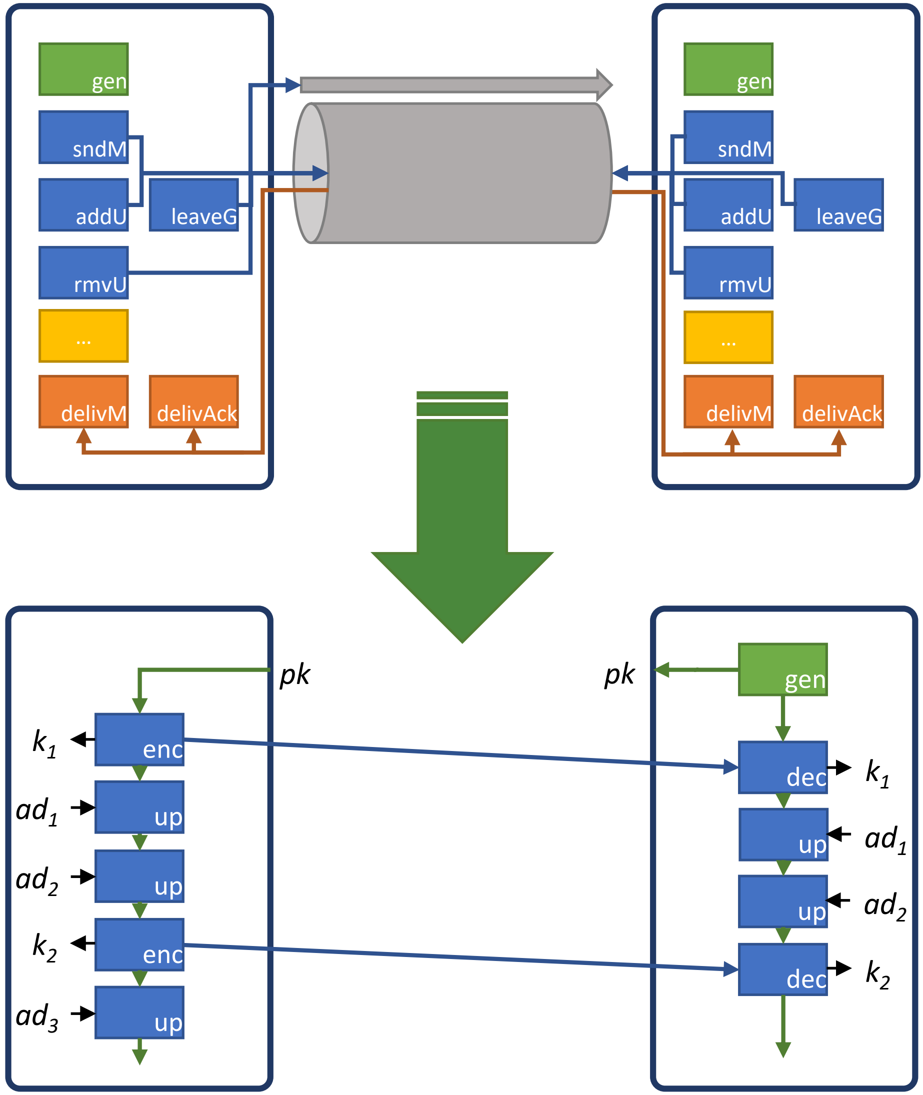
Asynchronous ratcheted key exchange

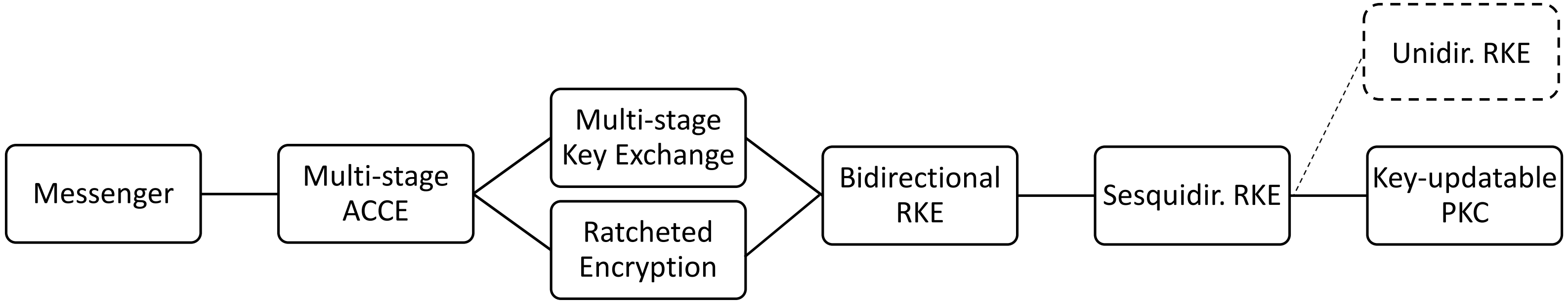
Bertram Poettering[1] and Paul Rösler[2]

[1] Information Security Group, Royal Holloway, University of London
bertram.poettering@rhul.ac.uk
[2] Horst-Görtz Institute for IT Security,
Chair for Network and Data Security, Ruhr-University Bochum
paul.roesler@rub.de

Syntax – Taming complexity

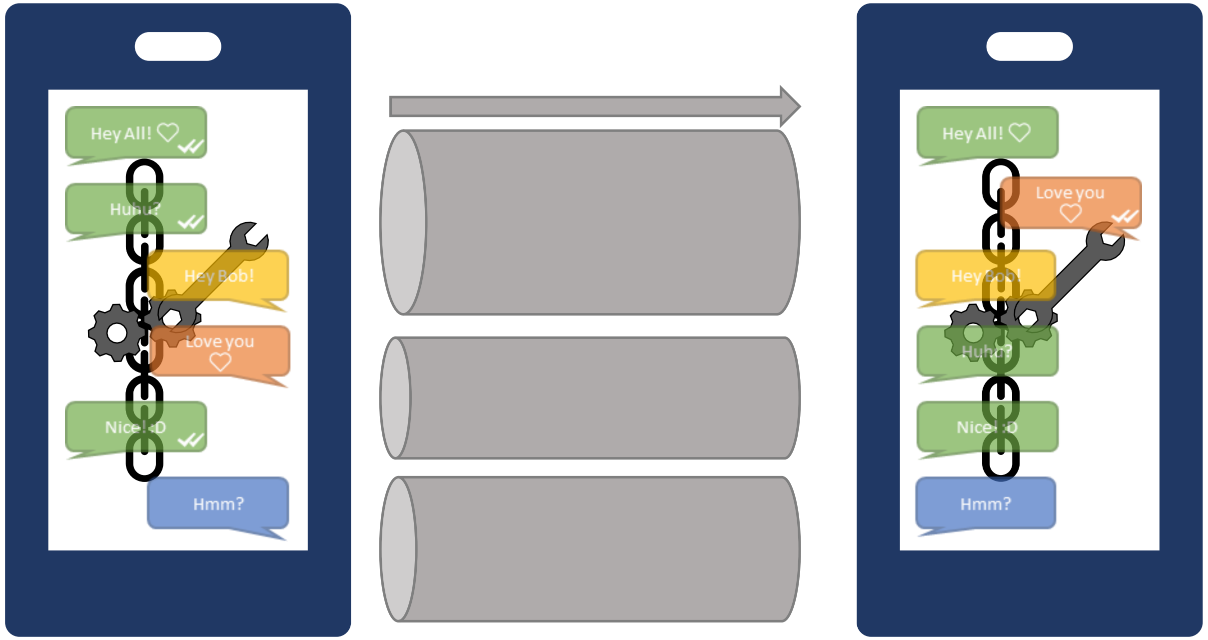
- Valid approach to reduce complexity by using compositions?
  - Less secure, less efficient than ad-hoc solutions
  - Usual approach in cryptography
    - Not an argument
  - Helps to understand components
  - Helps to exclude independent building blocks
- TODO: We need clear & useful interfaces





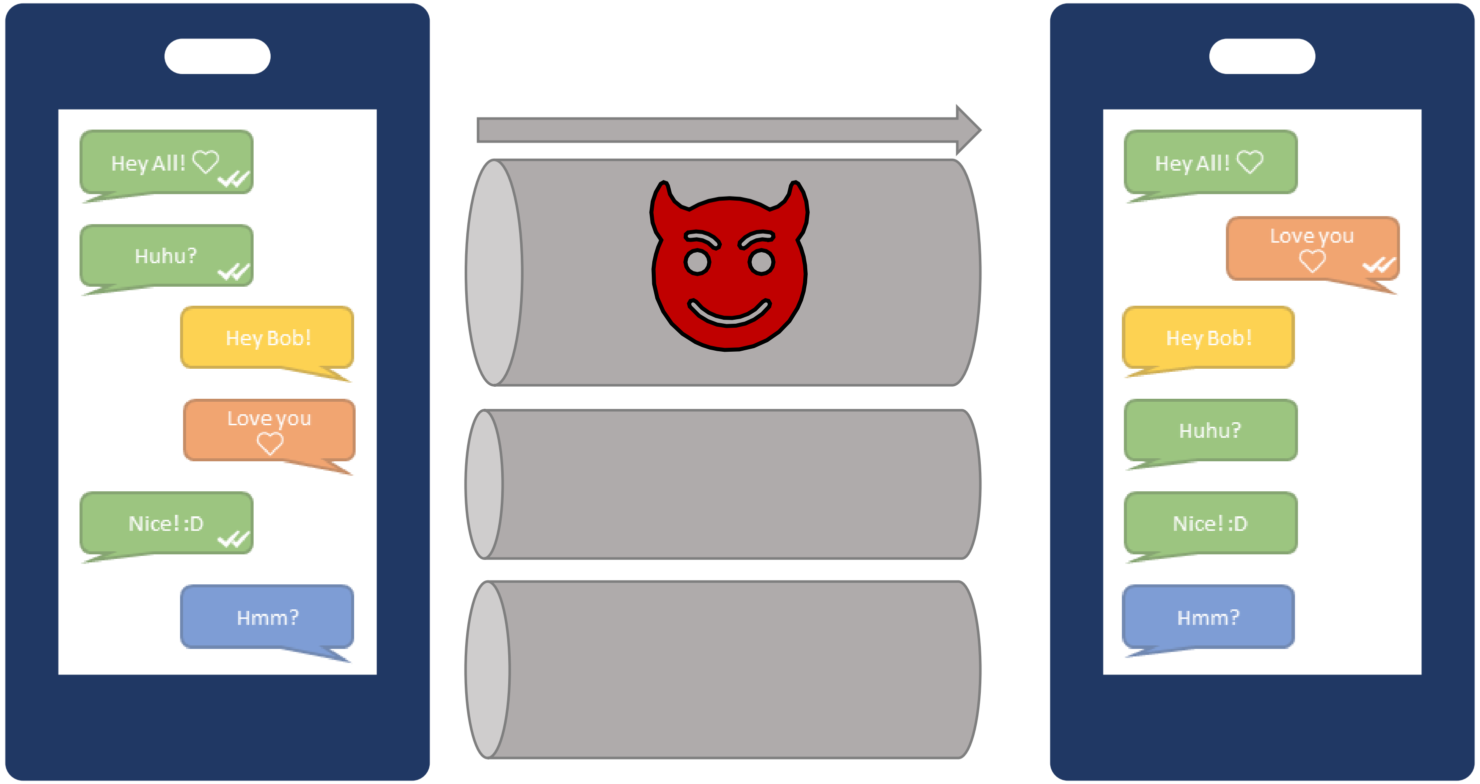
Agenda

- Messaging is complex
- Finding a Syntax
- **Understanding Attackers**
- Defining Security
- Core Primitive of RKE



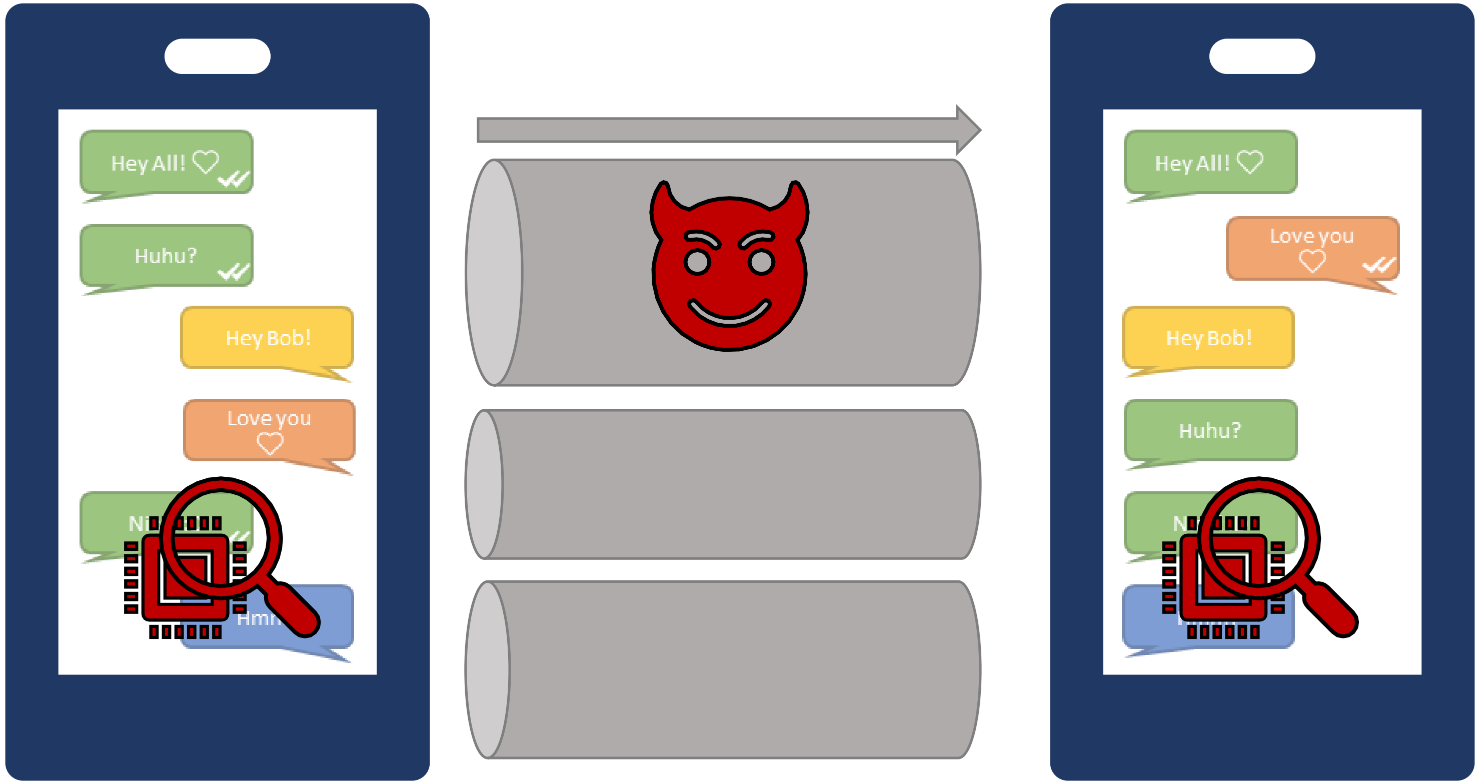
Attacker

- Active attacker on network
  - No trust in infrastructure
  - Becoming instance on network (path) is easy
- Manipulation of all traffic



Attacker

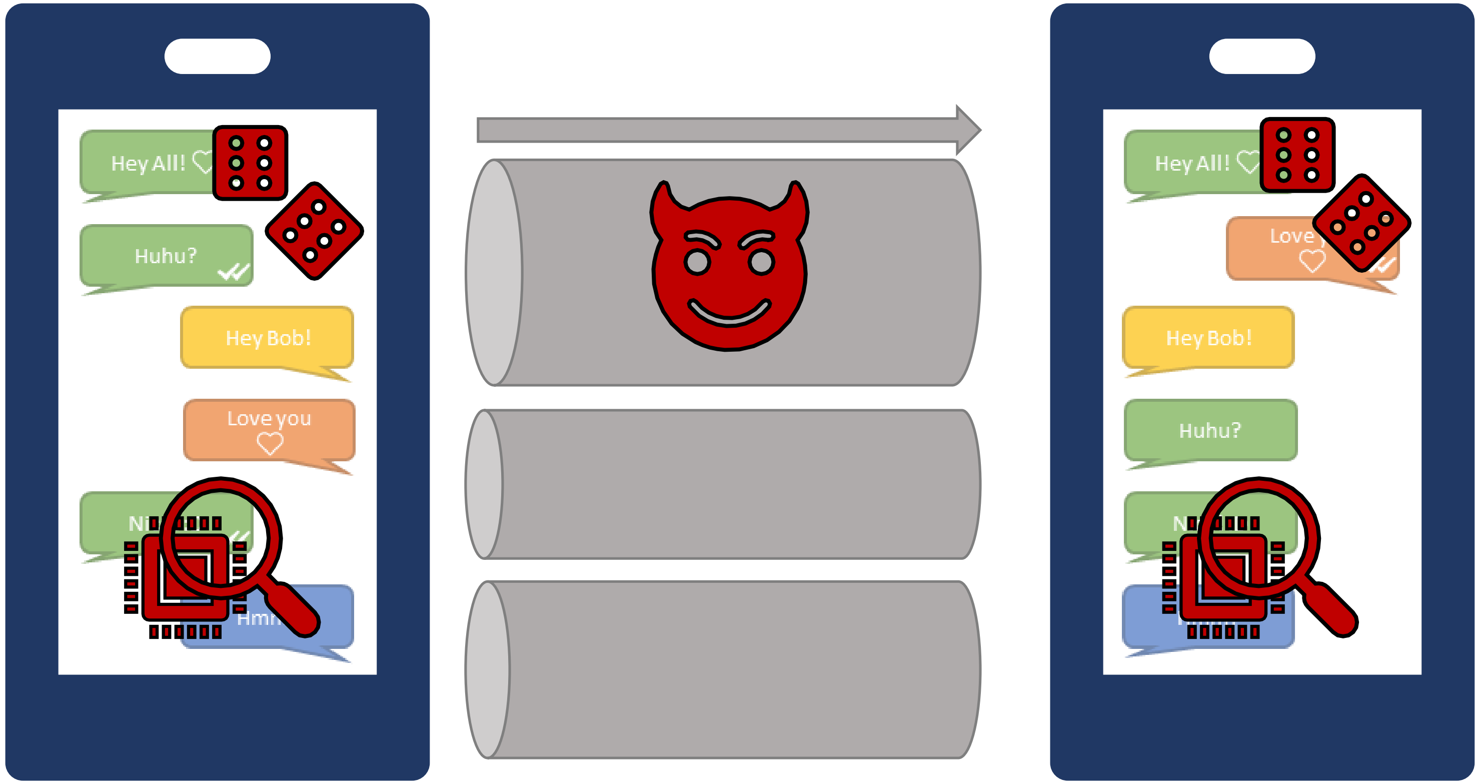
- Leakage of stored secrets
  - Mobile devices are easily accessible
  - Sessions take long time
- Exposure of local session state



RUHR UNIVERSITÄT BOCHUM RUB

Attacker

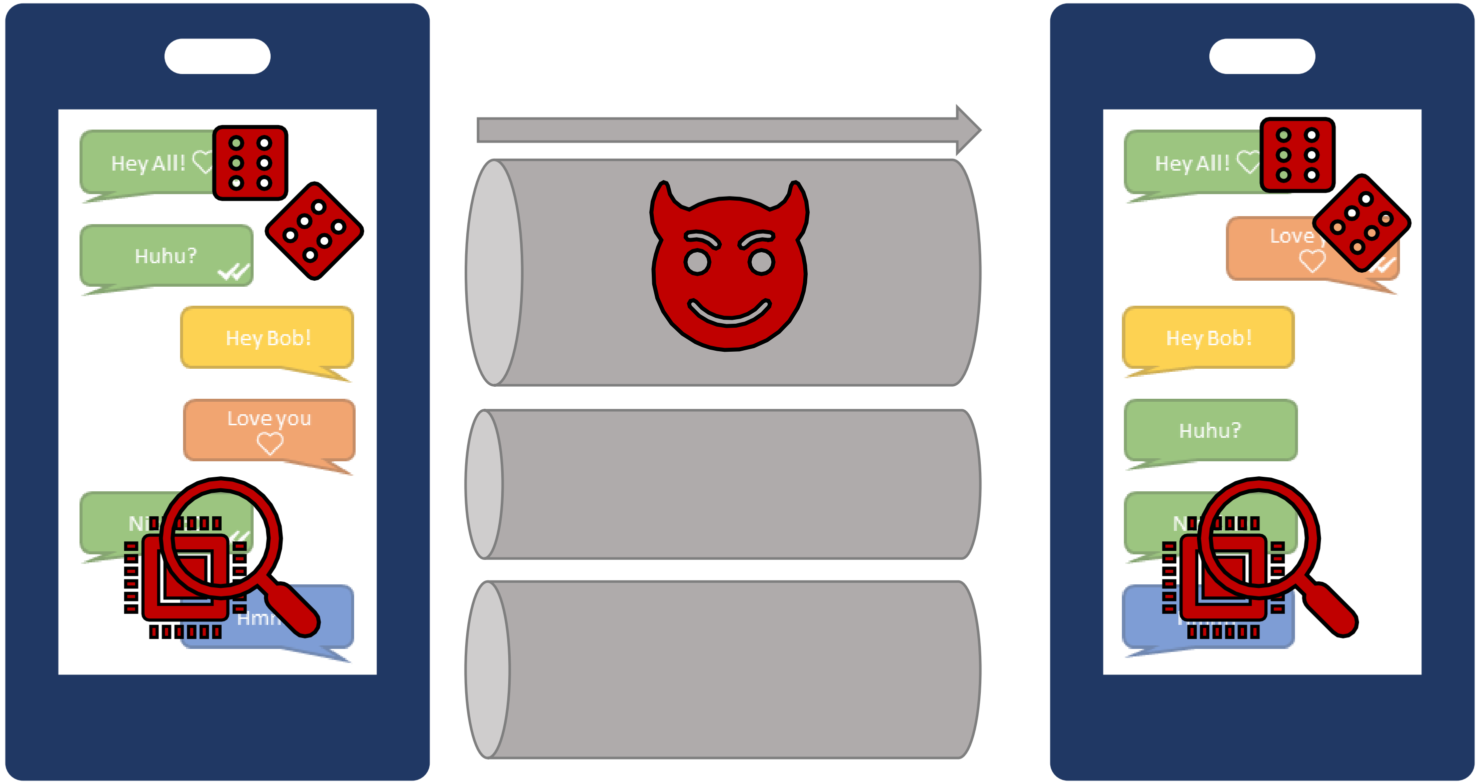
- Attacks against executions' randomness
  - Entropy low
  - Ba(d/ckdoored) randomness generator
- Reveal of random coins
  - Known (but good) random-ness?
- Manipulation of randomness
  - All bad distributions



RUHR UNIVERSITÄT BOCHUM RUB

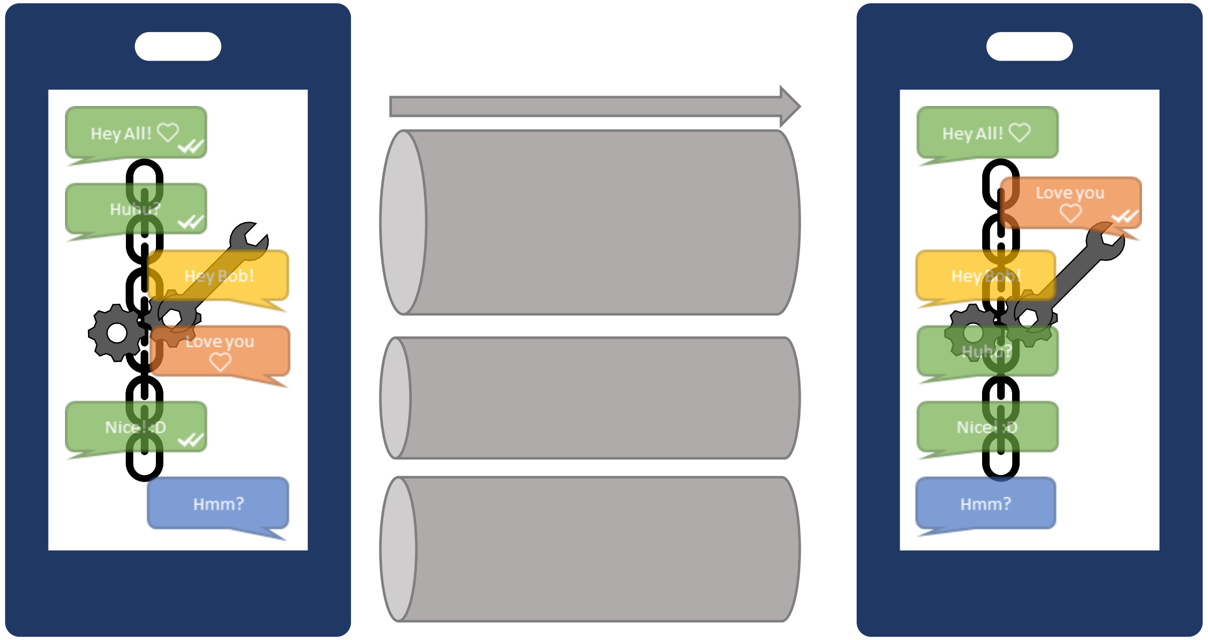
Attacker

- Many more attacker scenarios…
  - Attacker against key distribution
  - Attackers in attacked group
  - Leakage during computation
  - Attacker in implementation



Agenda

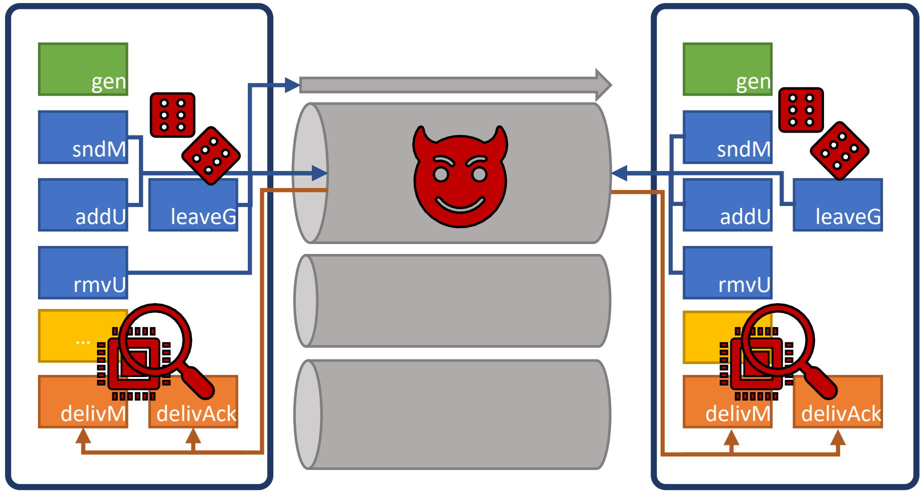
- Messaging is complex
- Finding a Syntax
- Understanding Attackers
- **Defining Security**
- Core Primitive of RKE

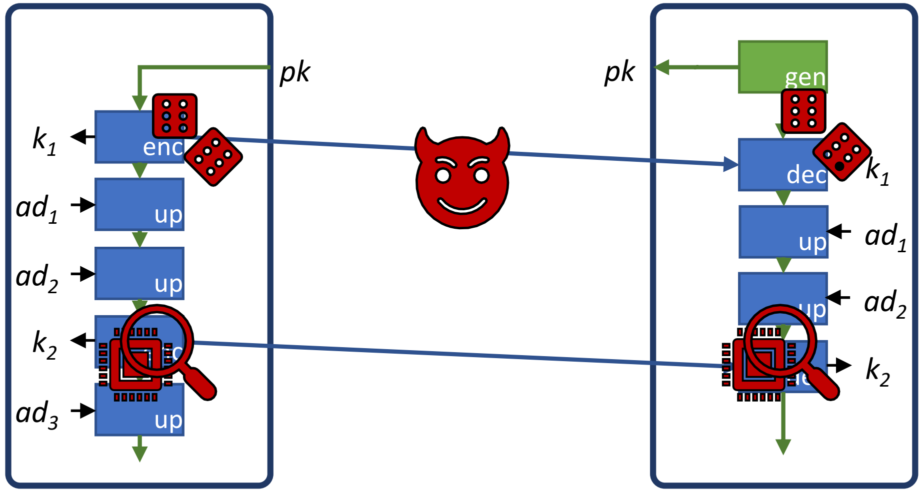


RUHR UNIVERSITÄT BOCHUM RUB

Security definition

- Many security properties, depend on:
  - Syntax
  - Correctness (i.e., no inconsistencies)
  - Functionality (i.e., [honest] execution guarantees)
    - Hard for abstract interactive protocols
  - Semantic (ambiguous)
- Multiple levels of properties:
  - Strongest security
  - Intuitive security (ambiguous)
  - Efficiently instantiable security (ambiguous)

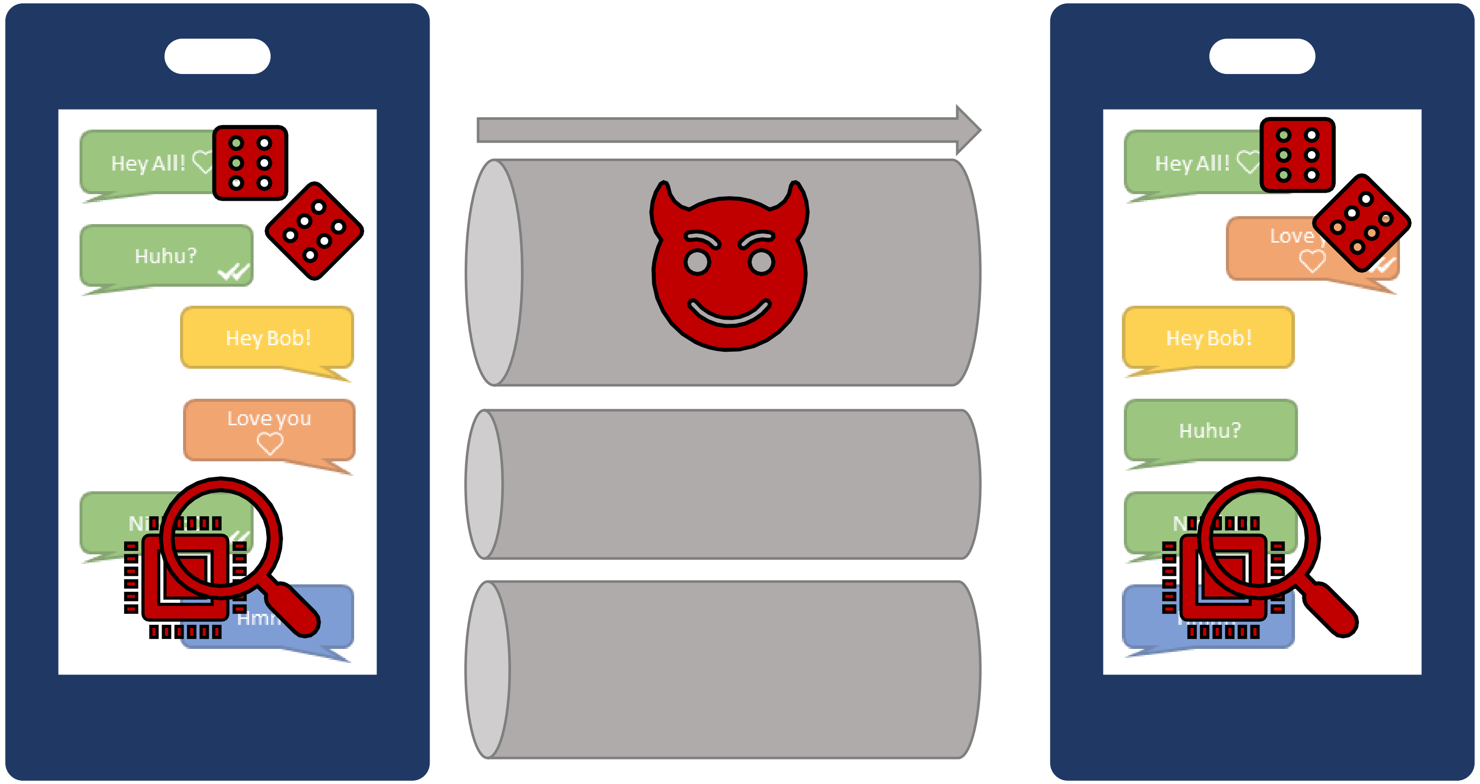




(Strongest) Security definition

RUHR UNIVERSITÄT BOCHUM RUB

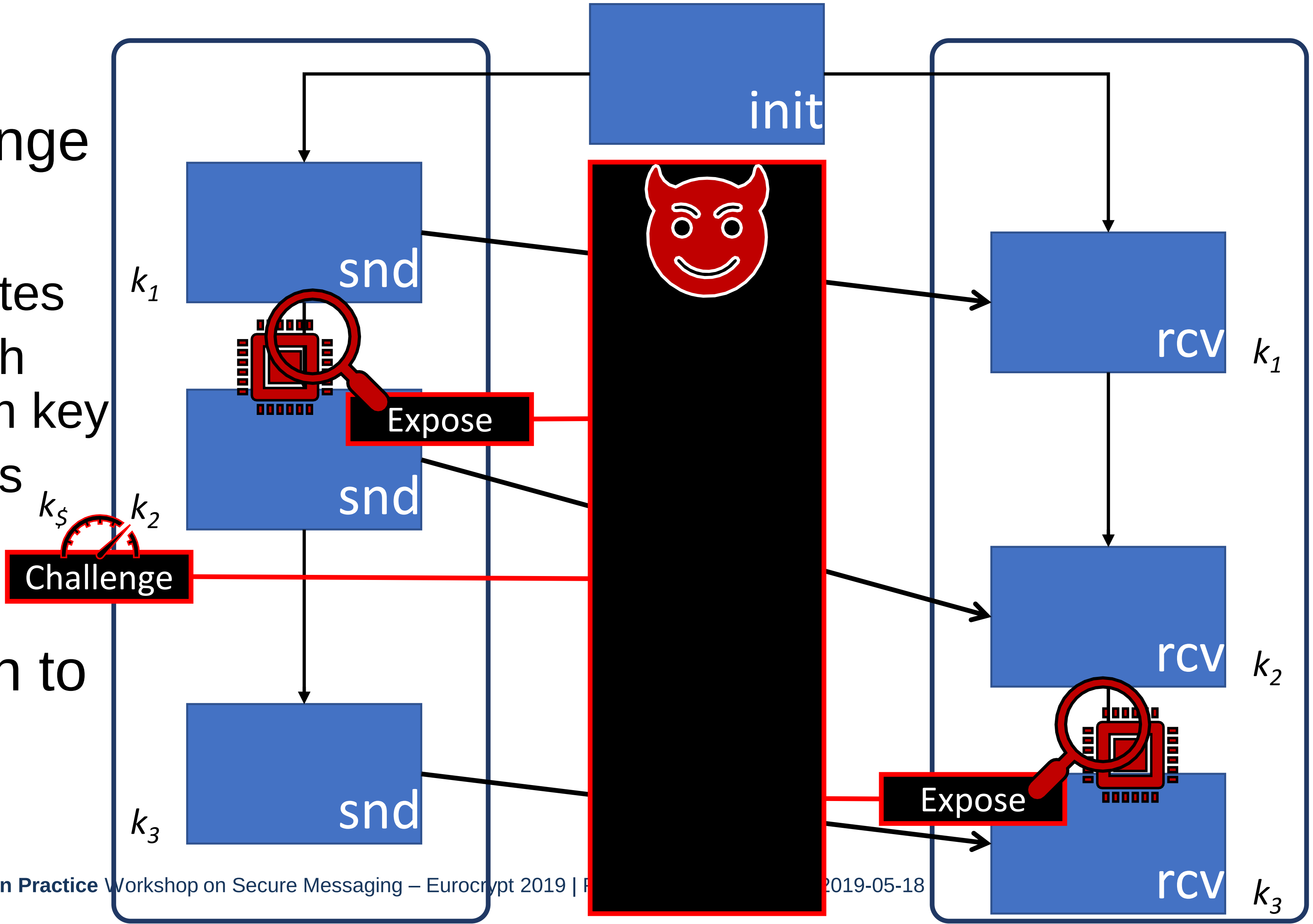
- Allow attacker full (defined) power
- Define security property as:
  Event that attacker should not trigger
- Forbid ways that directly trigger that event (unpreventable attacks)



- Example: simplified ratcheted key exchange variant

Security of Unidirectional RKE

- Restricted variant of ratcheted key exchange
- Attacker
  - can expose local states
  - should not distinguish real key from random key
  - (exclude randomness for now)
- Which keys are unpreventably known to attacker?



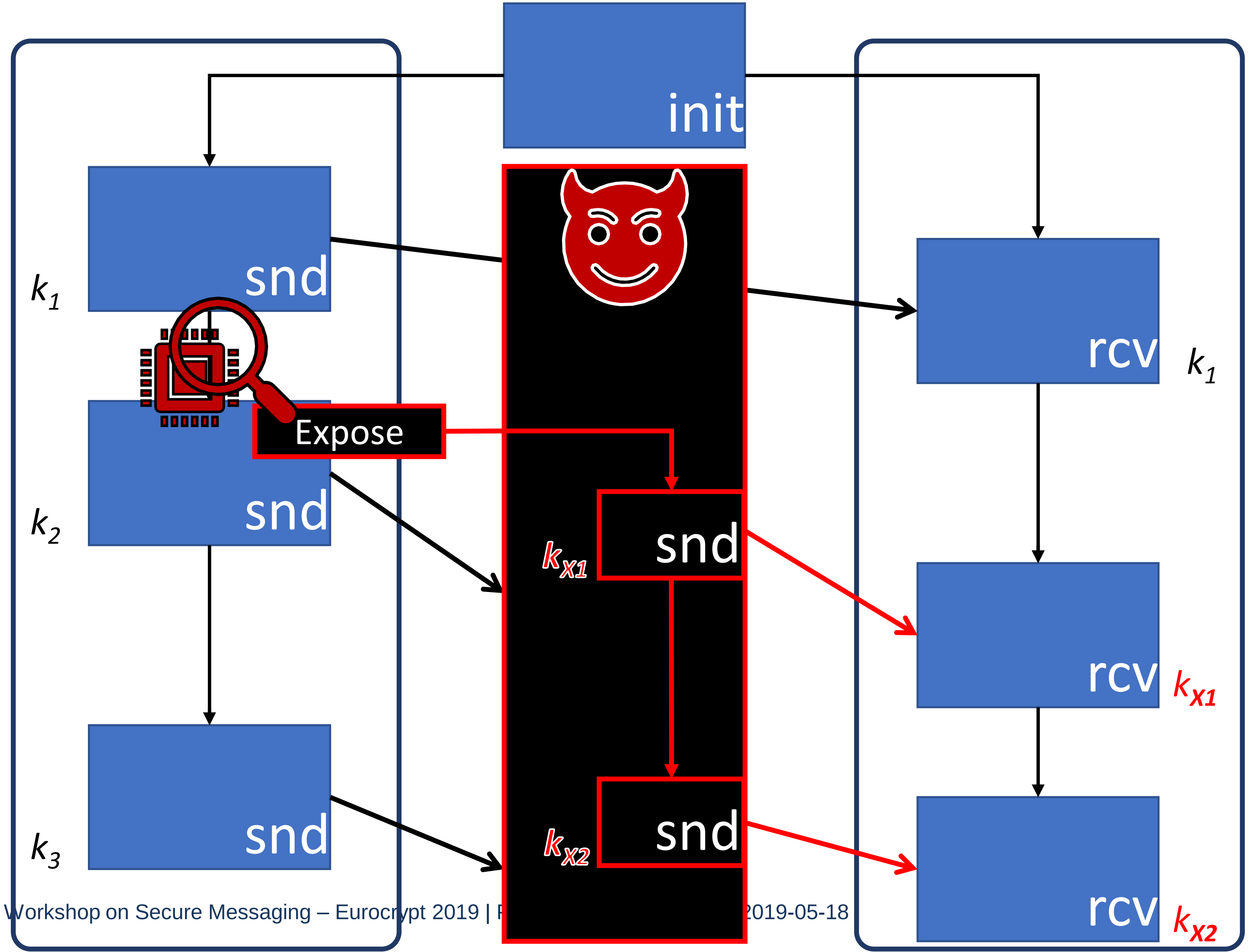
Security of Unidirectional RKE

Unpreventable Attacks

1. Exposure of Alice's state
2. Use state to forge ciphertext to Bob

⇒ Adversary knows key

- Impersonation
  ⇒ No future Challenge on Bob's keys



Security of Unidirectional RKE

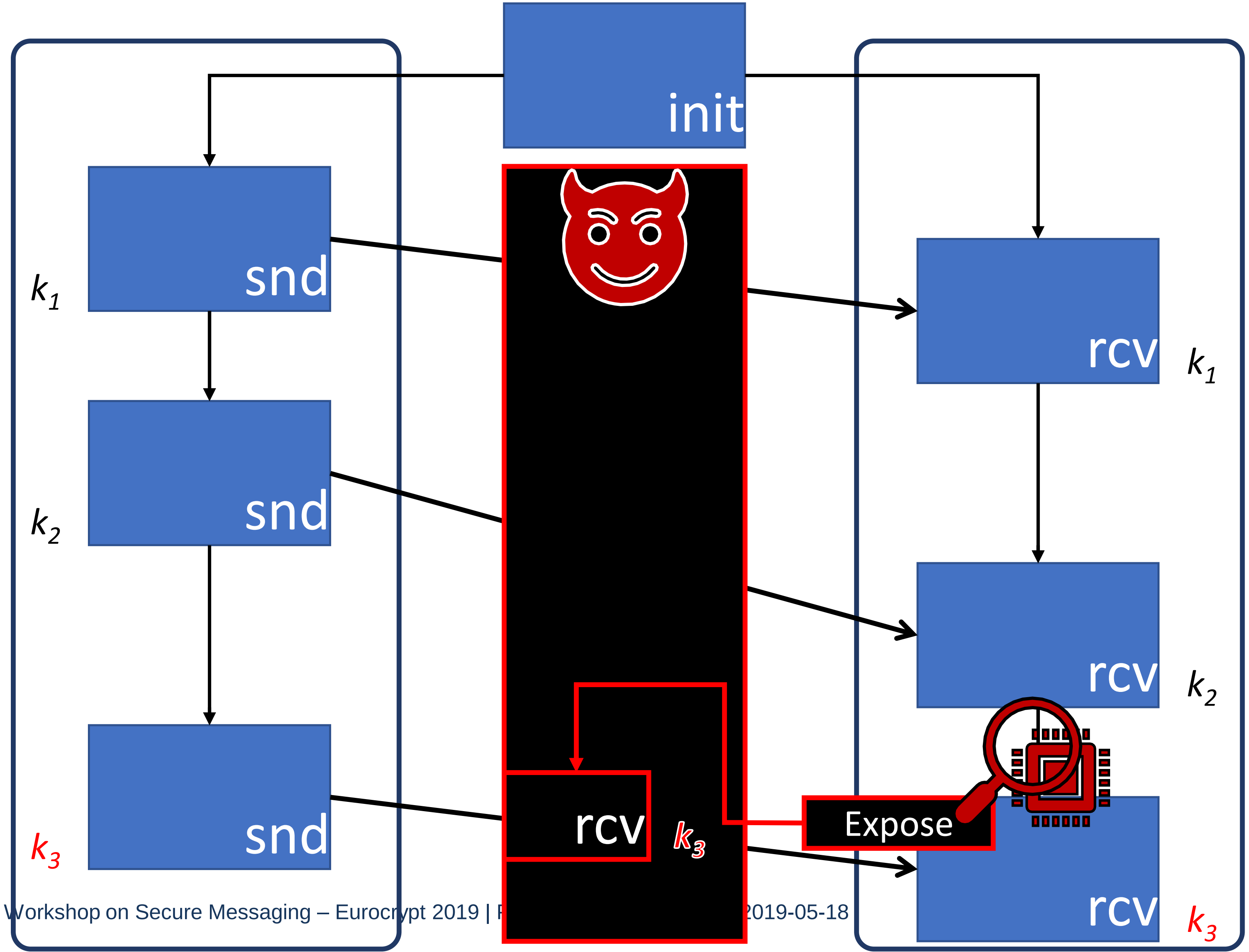
Unpreventable Attacks

- Impersonation
  ⇒ No future Challenge on Bob's keys

1. Expose Bob's state
2. Use state to receive ciphertext to Bob

⇒ Adversary knows key

- Expose Bob
  ⇒ No future Challenge on Bob's keys



Security of Unidirectional RKE

Unpreventable Attacks

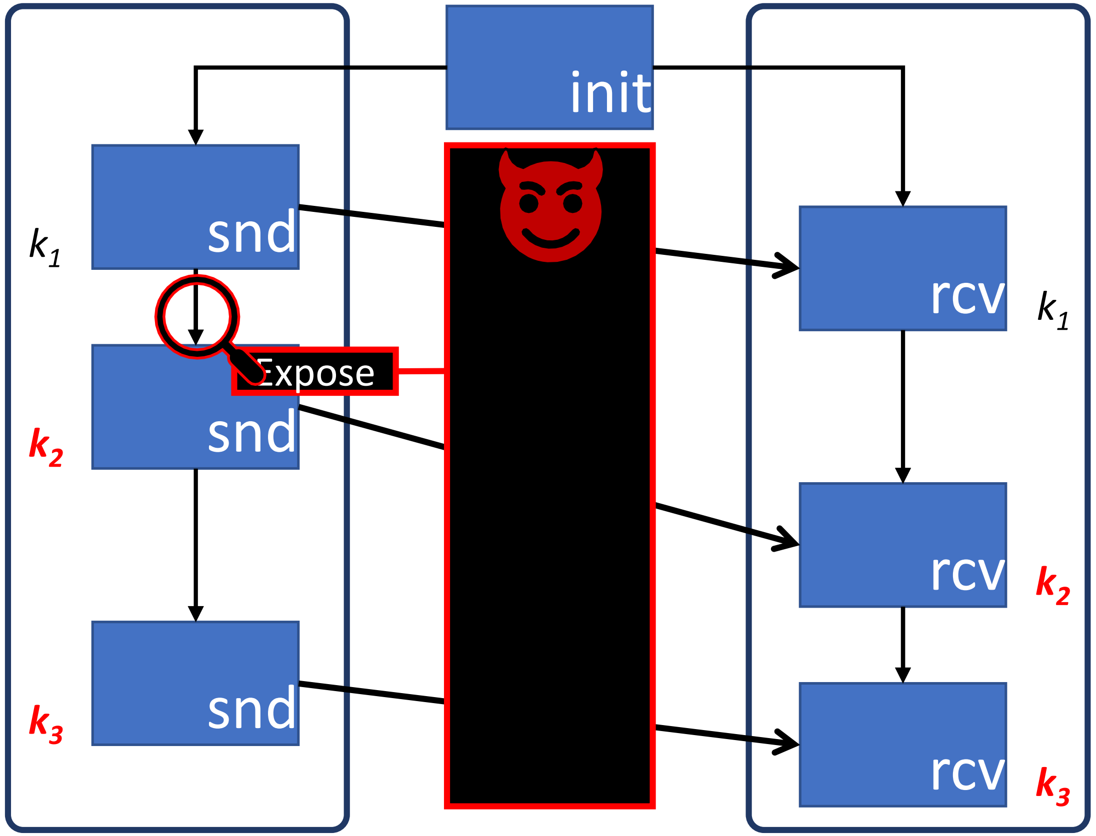
- Impersonation
- Expose Bob

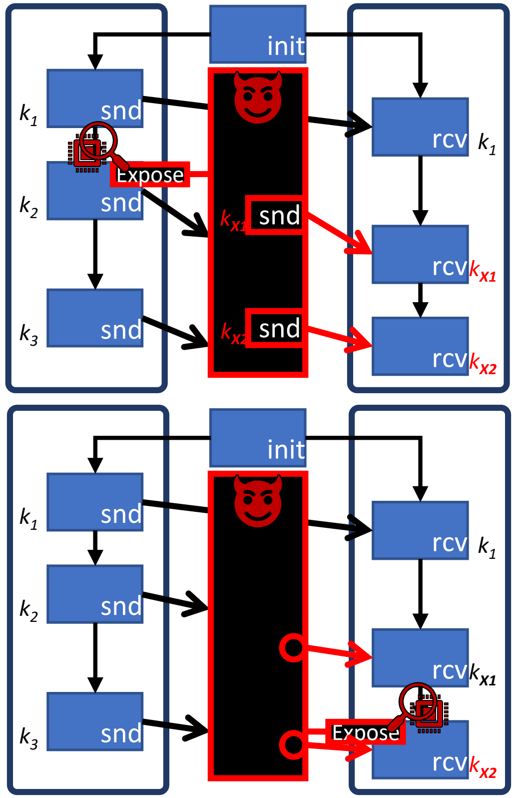
⇒ No future Challenge on Bob's keys

- Remaining keys secure

Preventable Attacks

- Symmetric leakage





RUHR UNIVERSITÄT BOCHUM RUB

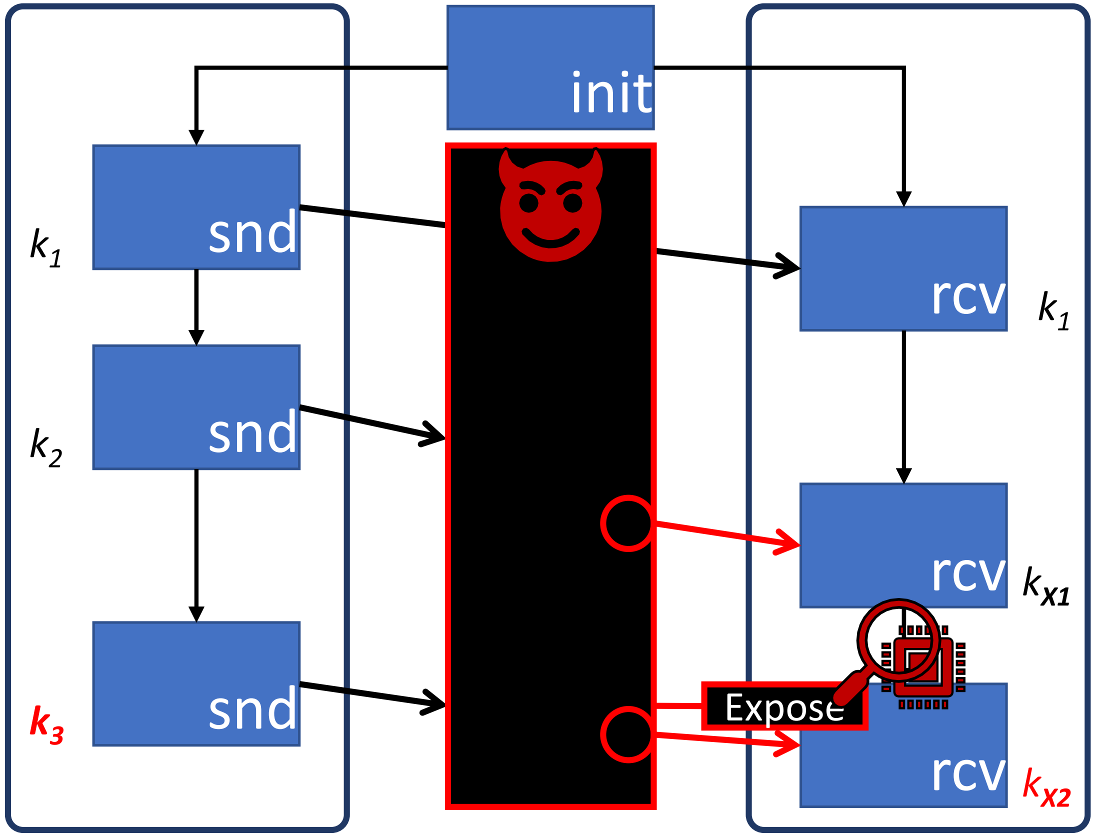
Security of Unidirectional RKE

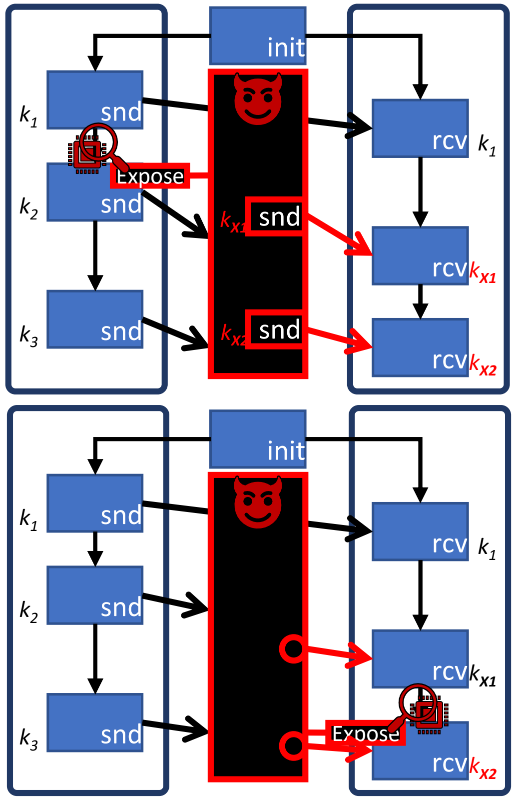
Unpreventable Attacks

- Impersonation
- Expose Bob

⇒ No future Challenge on Bob's keys

- Remaining keys secure





Preventable Attacks

- Symmetric leakage
- Active attack ⇏ independence of states
- No exposure of Bob's state, … (more in bidirectional setting)

RUHR UNIVERSITÄT BOCHUM RUB

Security of Unidirectional RKE

Unpreventable Attacks

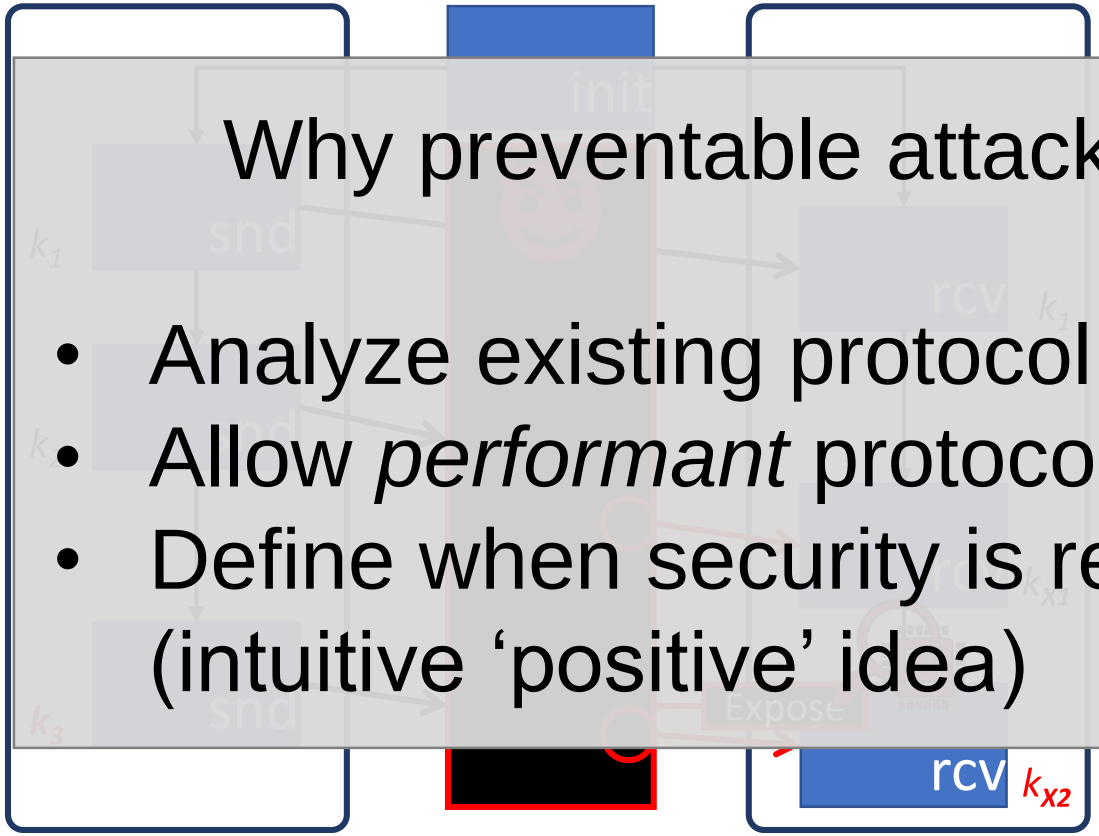
- Impersonation
- Expose Bob

⇒ No future Challenge on Bob’s keys

- Remaining keys secure

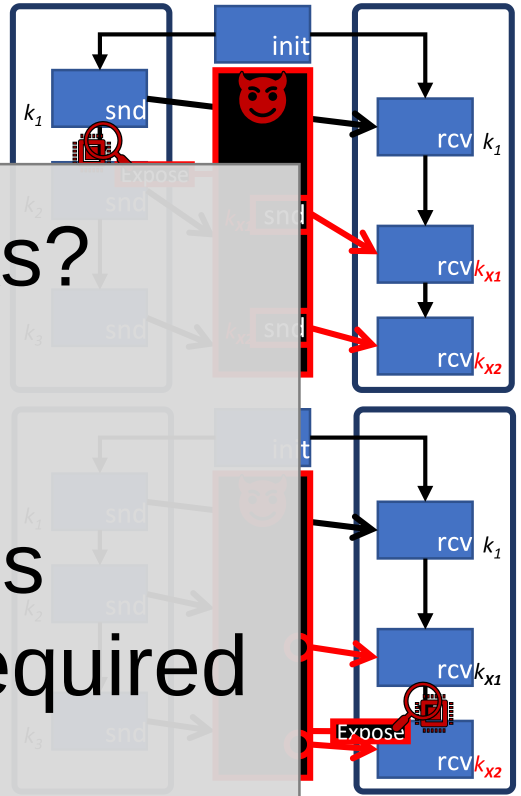
Preventable Attacks

- Symmetric leakage
- Active attack ⇏ independence of states
- No exposure of Bob’s state, … (more in bidirectional setting)



Why preventable attacks?

- Analyze existing protocol
- Allow *performant* protocols
- Define when security is required (intuitive ‘positive’ idea)



RUHR UNIVERSITÄT BOCHUM RUB

Security of Unidirectional RKE

Unpreventable Attacks

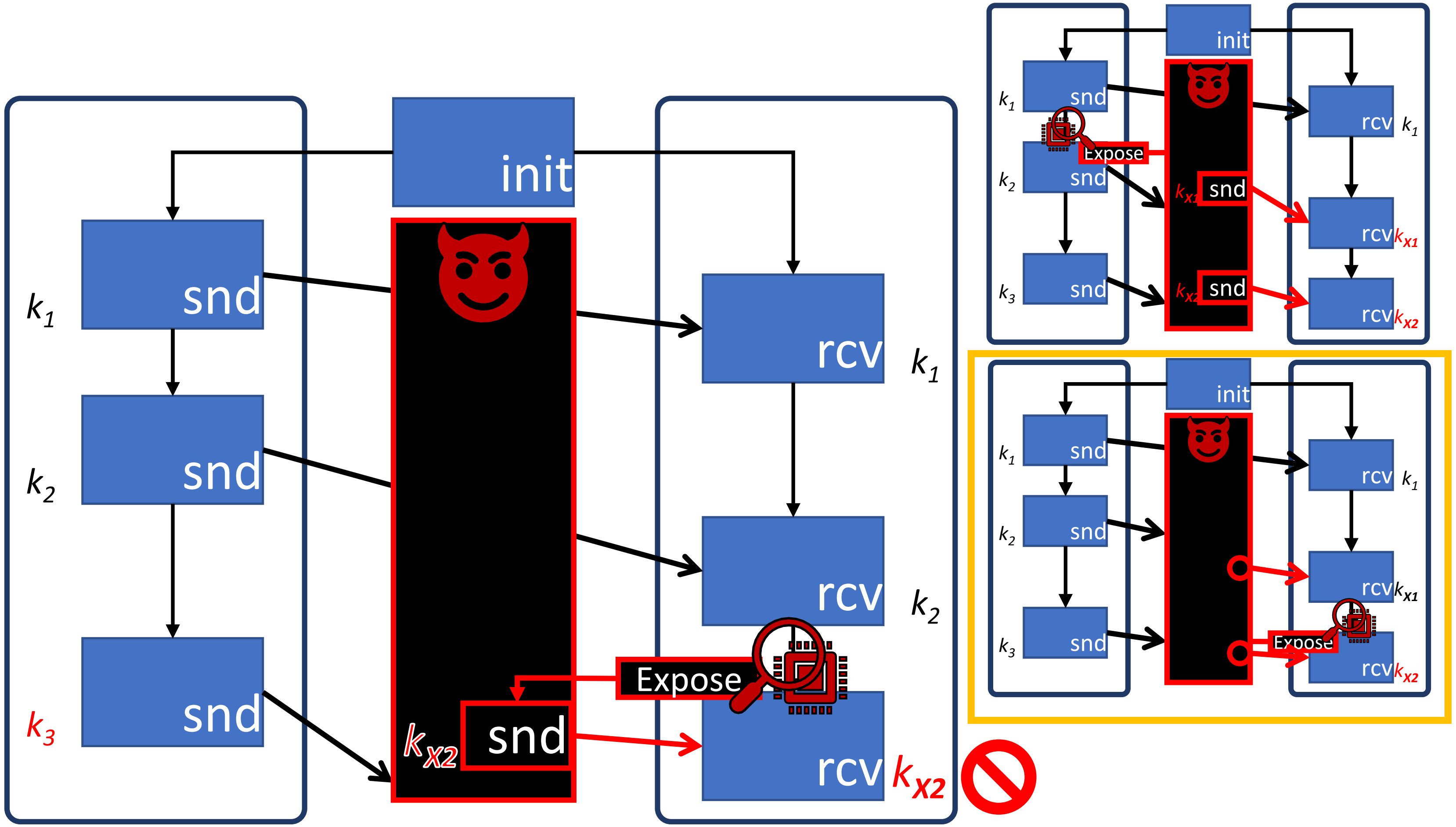
- Impersonation
- Expose Bob

⇒ No future Challenge on Bob's keys

- Remaining keys secure

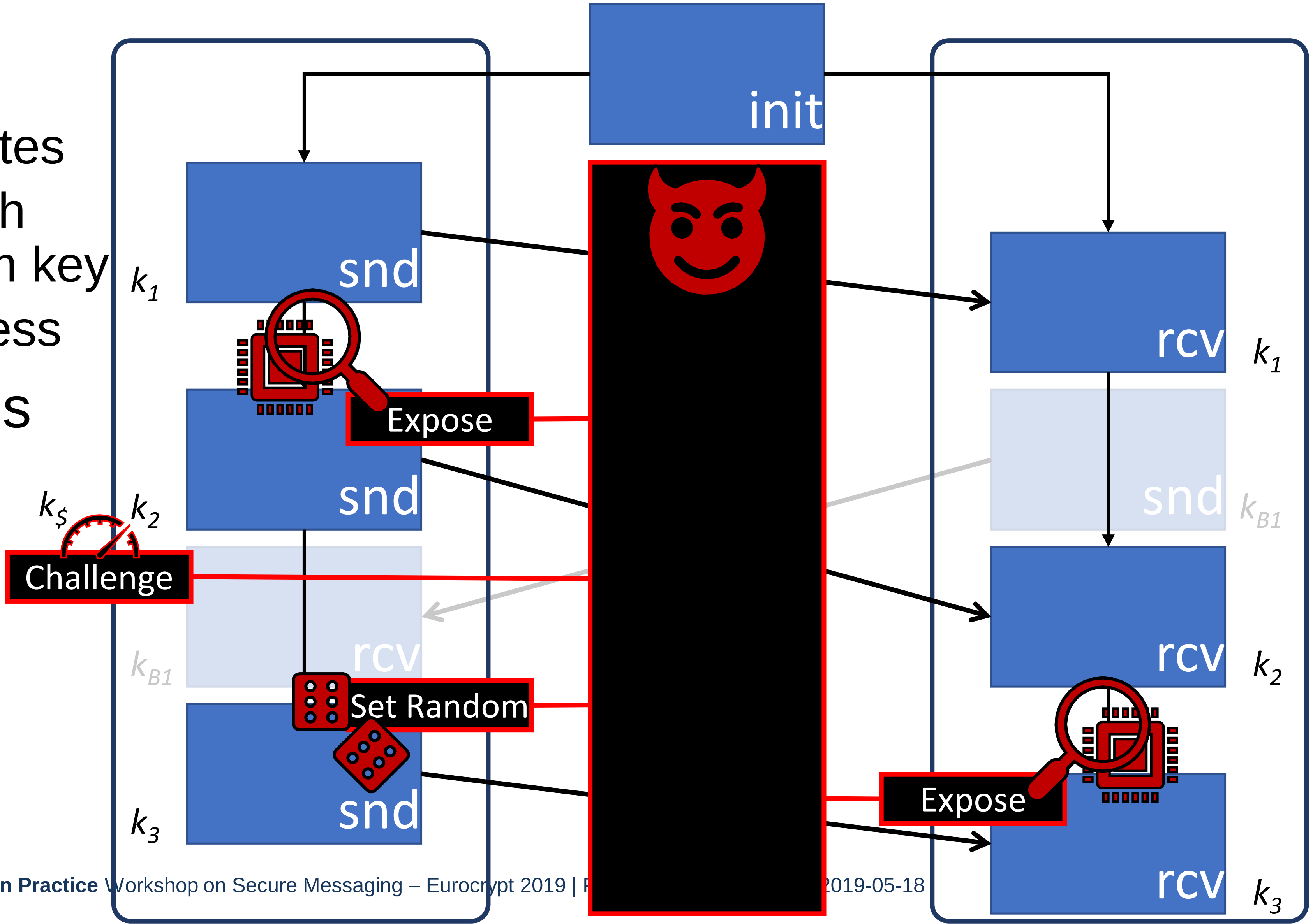
Further properties

- Explicit authentication
  - No self-impersonation (authenticating keys?)
  - TODO: build compilers/extensions (e.g., sign ciphertexts)



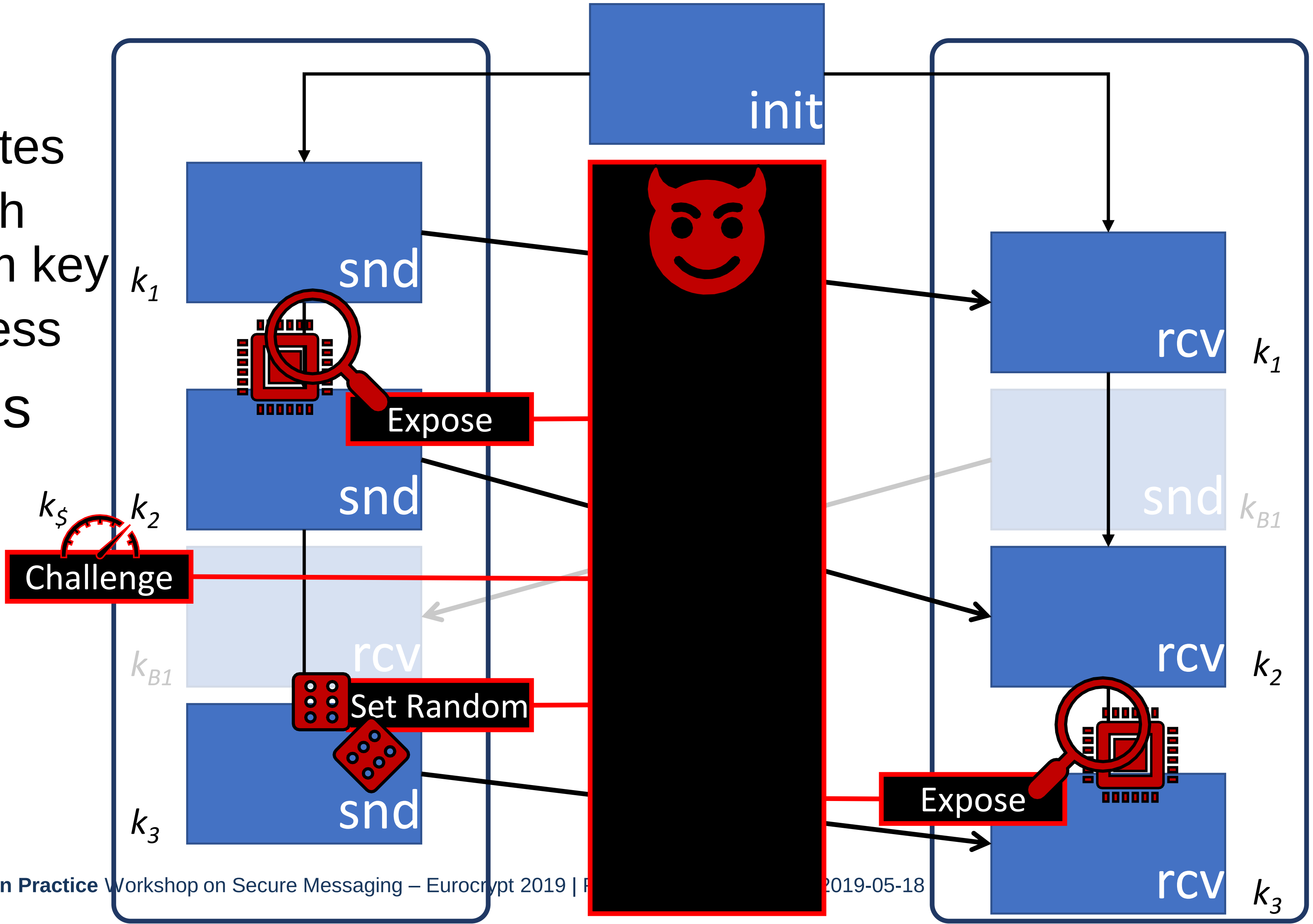
Security of Unidirectional RKE

- Attacker
  - ✓ can expose local states
  - ✓ should not distinguish real key from random key
  - ? can attack randomness
- Multiple constructions via public key crypto
  - Sufficient
  - Necessary



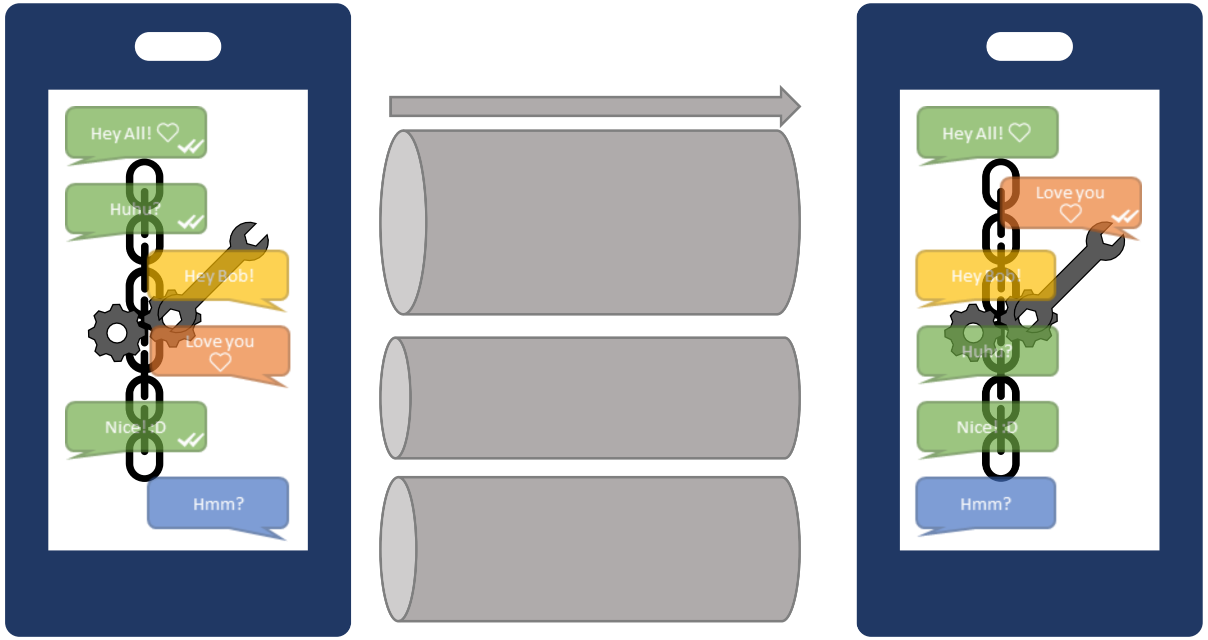
Security of Unidirectional RKE

- Attacker
  - ✓ can expose local states
  - ✓ should not distinguish real key from random key
  - ? can attack randomness
- Multiple constructions via public key crypto
  - ~~Sufficient~~
  - Necessary



Agenda

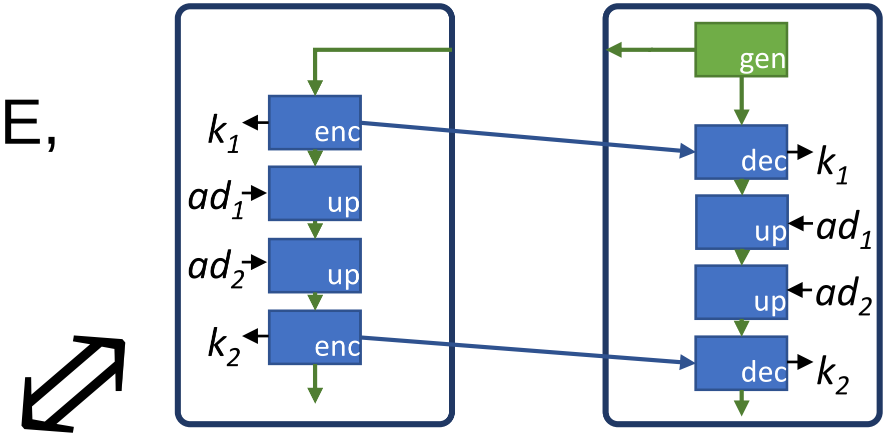
- Messaging is complex
- Finding a Syntax
- Understanding Attackers
- Defining Security
- **Core Primitive of RKE**

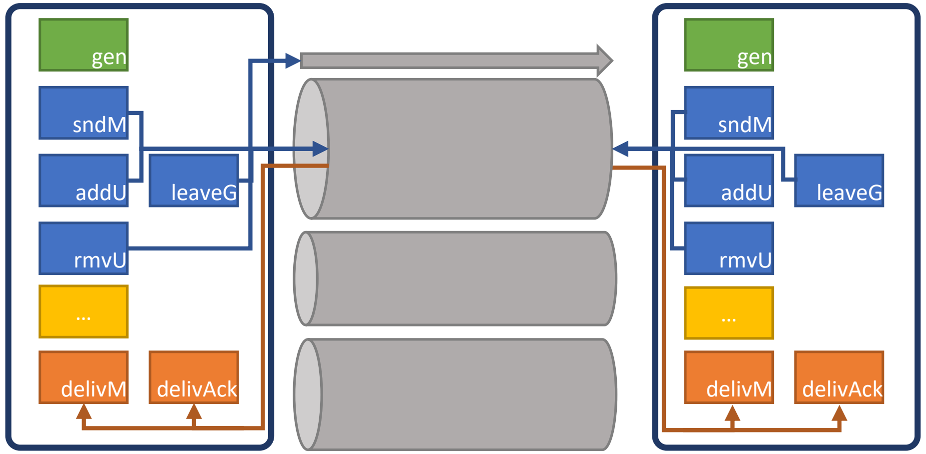


RUHR UNIVERSITÄT BOCHUM RUB

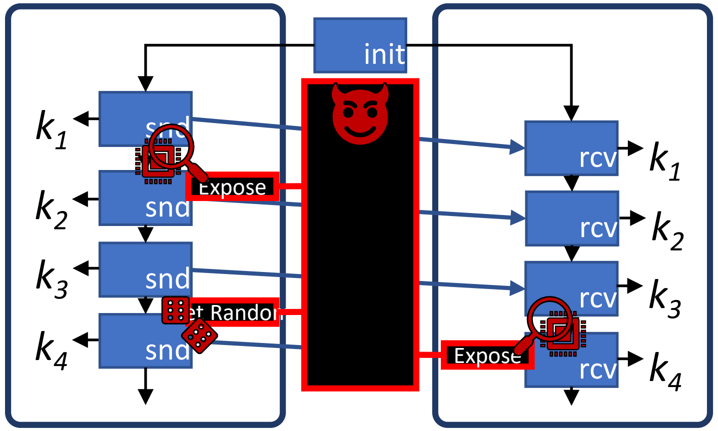
Implications of security definition

- Unpublished work (w/ Serge Vaudenay & Fatih Balli)
  - If randomness is revealed, Unidirectional RKE ⇔ key-updatable PKC
  - Unidirectional RKE is part of Sesquidectional RKE, which is part of Bidirectional RKE
- Key-updatable PKC core primitive of strongly secure messaging



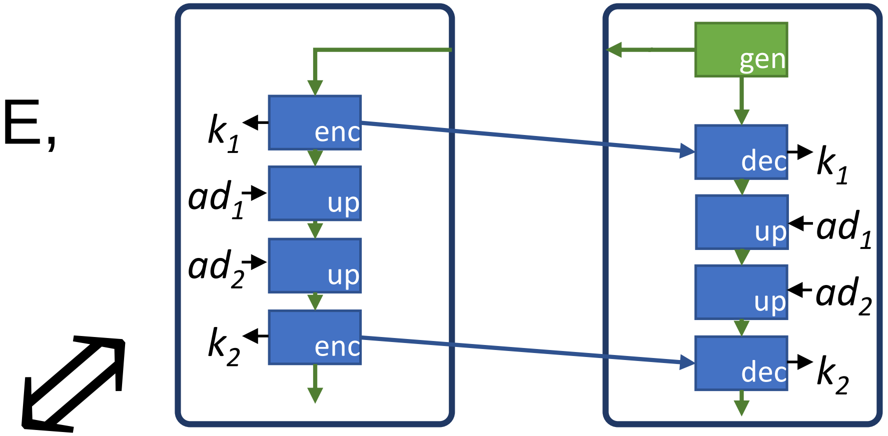


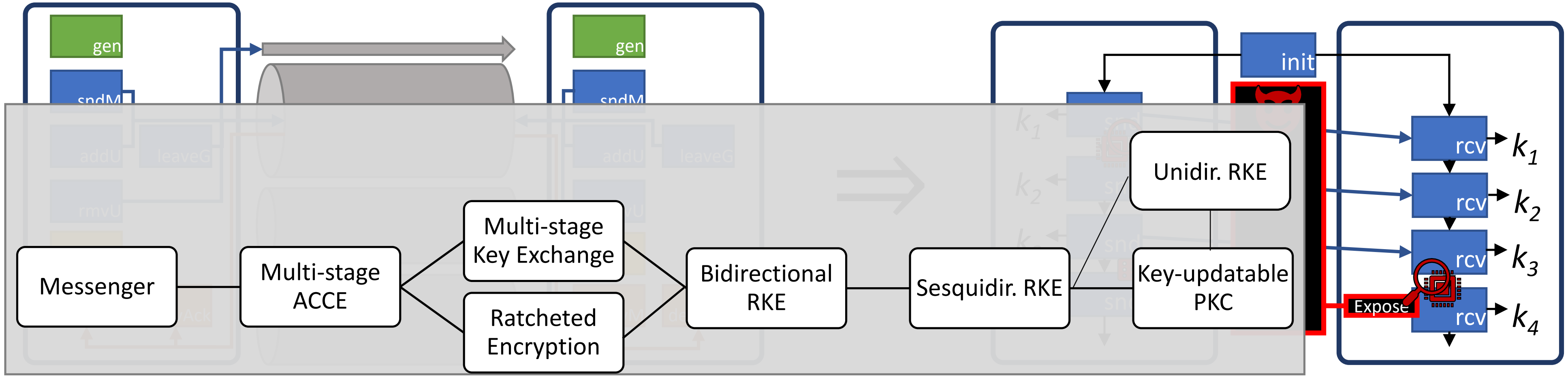
⇒



Implications of security definition

- Ongoing work (w/ Serge Vaudenay & Fatih Balli)
  - If randomness is revealed,
    Unidirectional RKE ⇔ key-updatable PKC
  - Unidirectional RKE is part of Sesquidectional RKE, which is part of Bidirectional RKE
- Key-updatable PKC core primitive of strongly secure messaging

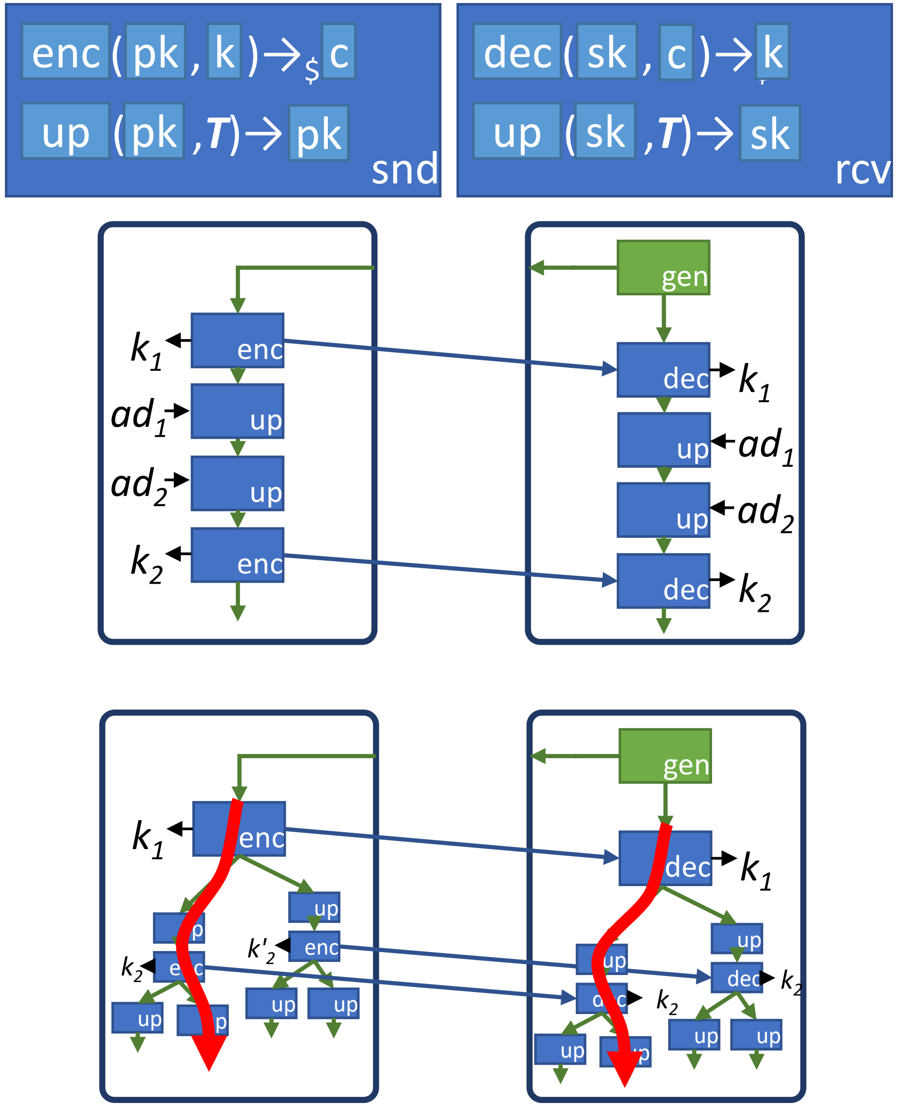




RUHR UNIVERSITÄT BOCHUM RUB

Implications of security definition

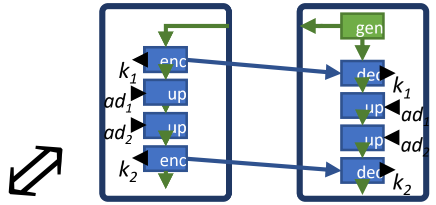
- Most previous ratcheting schemes with PKC
  - Security definitions not via trivial attacks
  - Attacker not able to attack randomness
- ‘Optimal’ ratcheting security only via (expensive) key-updatable PKC
- Idea of key-updatable PKC : update pk and sk independently and forward securely
- Based on (expensive) HIBE
  - Not full HIBE, only path on ‘identity tree’
  - TODO: enhance performance with this restriction

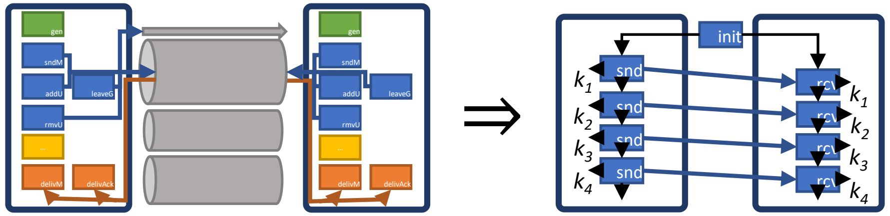


RUHR UNIVERSITÄT BOCHUM RUB

Summary

- Signal is secure enough for most applications
- Research should understand ratcheting
  - Abstractly approach syntax, attackers, security definition
  - Find relations
    - Among notions of ratcheting
    - Towards related primitives
  - Necessary to overcome ambiguities
- TODOs:
  - Define security before designing protocols
  - More efficient key-updatable PKC
  - Compositions up to messaging (avoid ad-hoc solutions)
  - Implement your protocols
    - Marco Smeets implemented (theoretically) strongly secure RKE





@roeslpa

github.com/
RUB-NDS/RKE