Towards Bidirectional Ratcheted Key Exchange

CRYPTO 2018

2018-08-20

Information Security Group

Royal Holloway, University of London

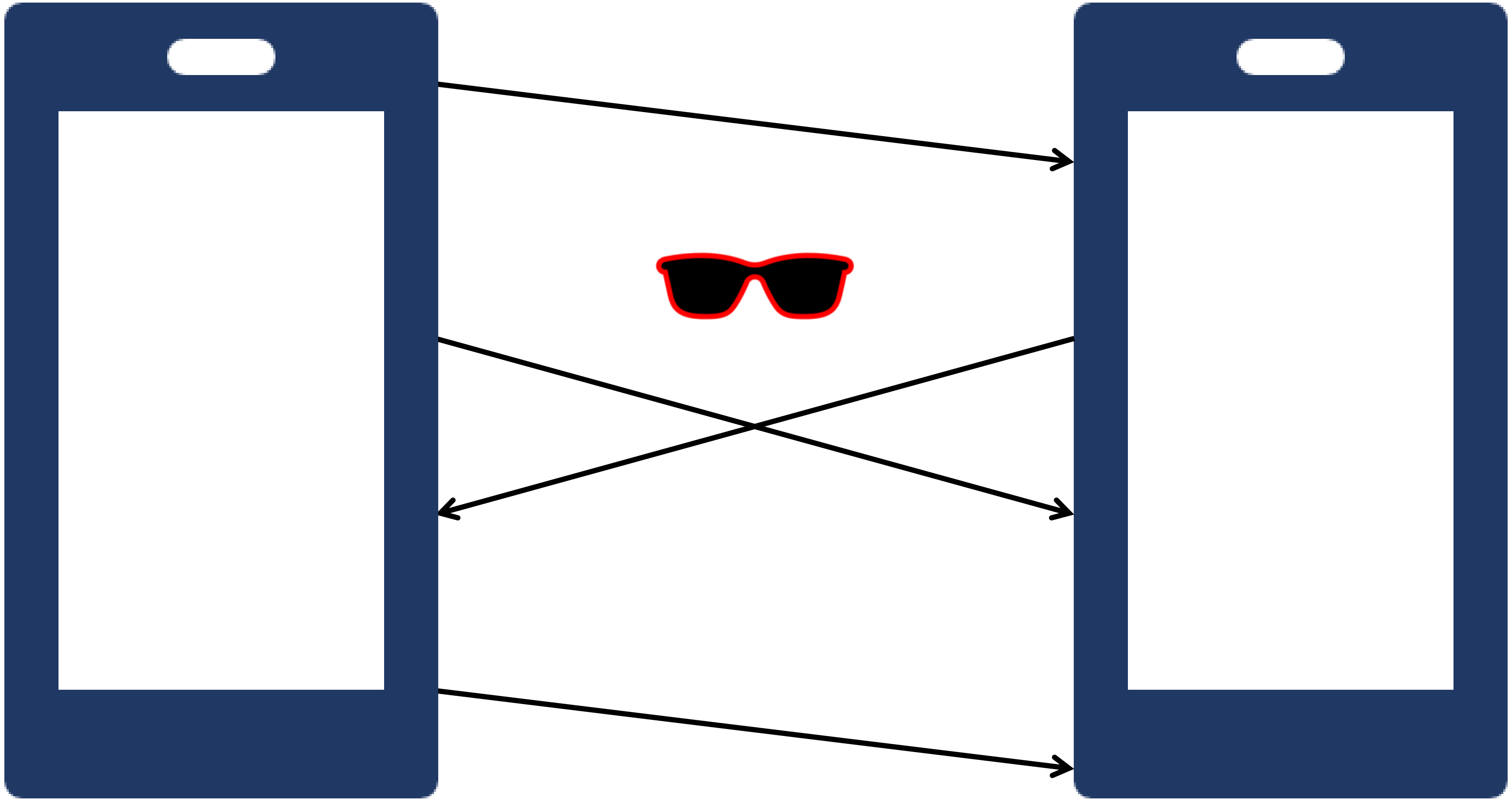
Bertram Poettering

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

Paul Rösler

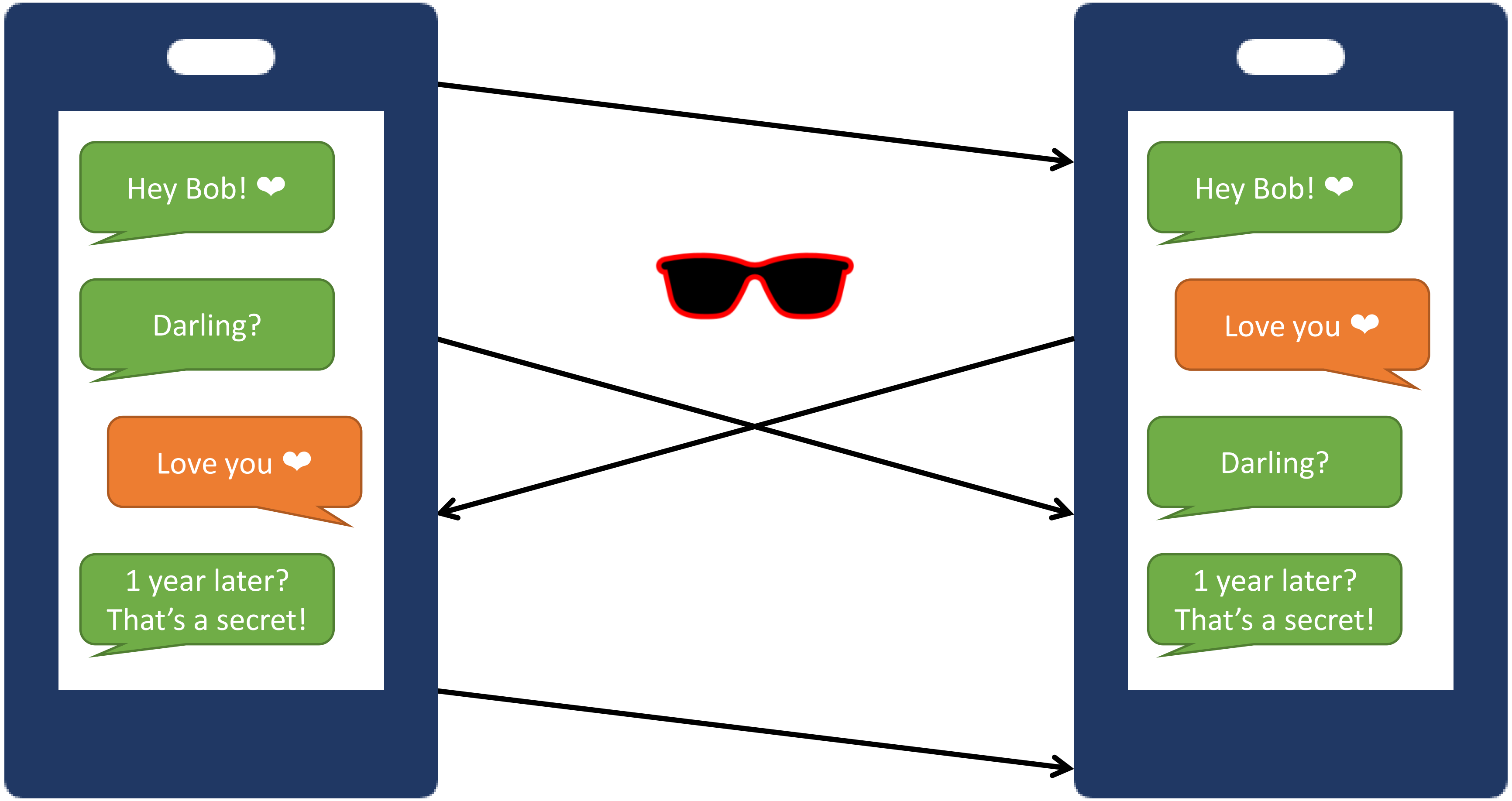
Introduction

- Alice and Bob communicate
- Active adversary



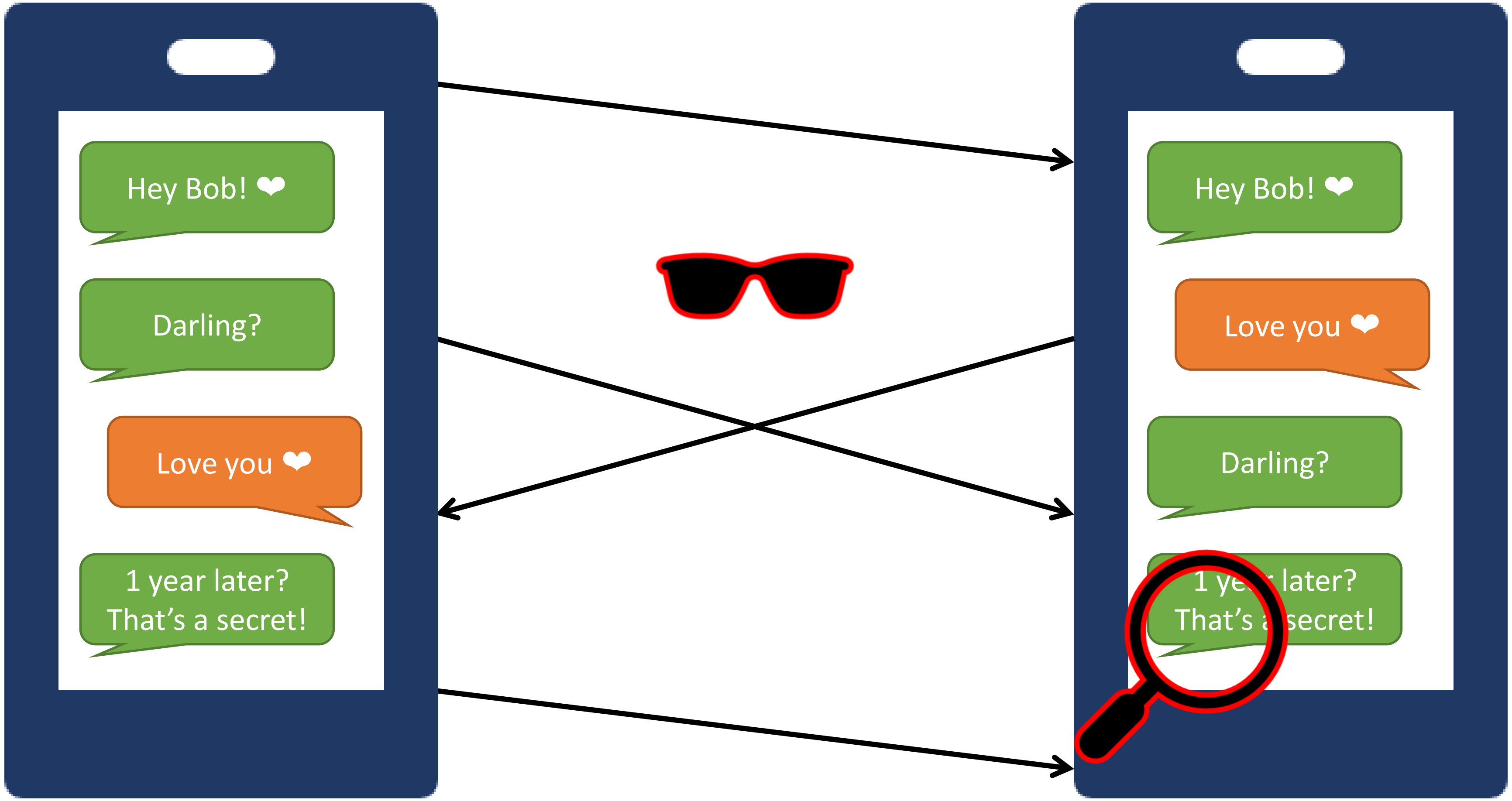
Introduction

- Alice and Bob communicate
- Active adversary


Hey Bob! ❤
Darling?
Love you ❤
1 year later? That's a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That's a secret!

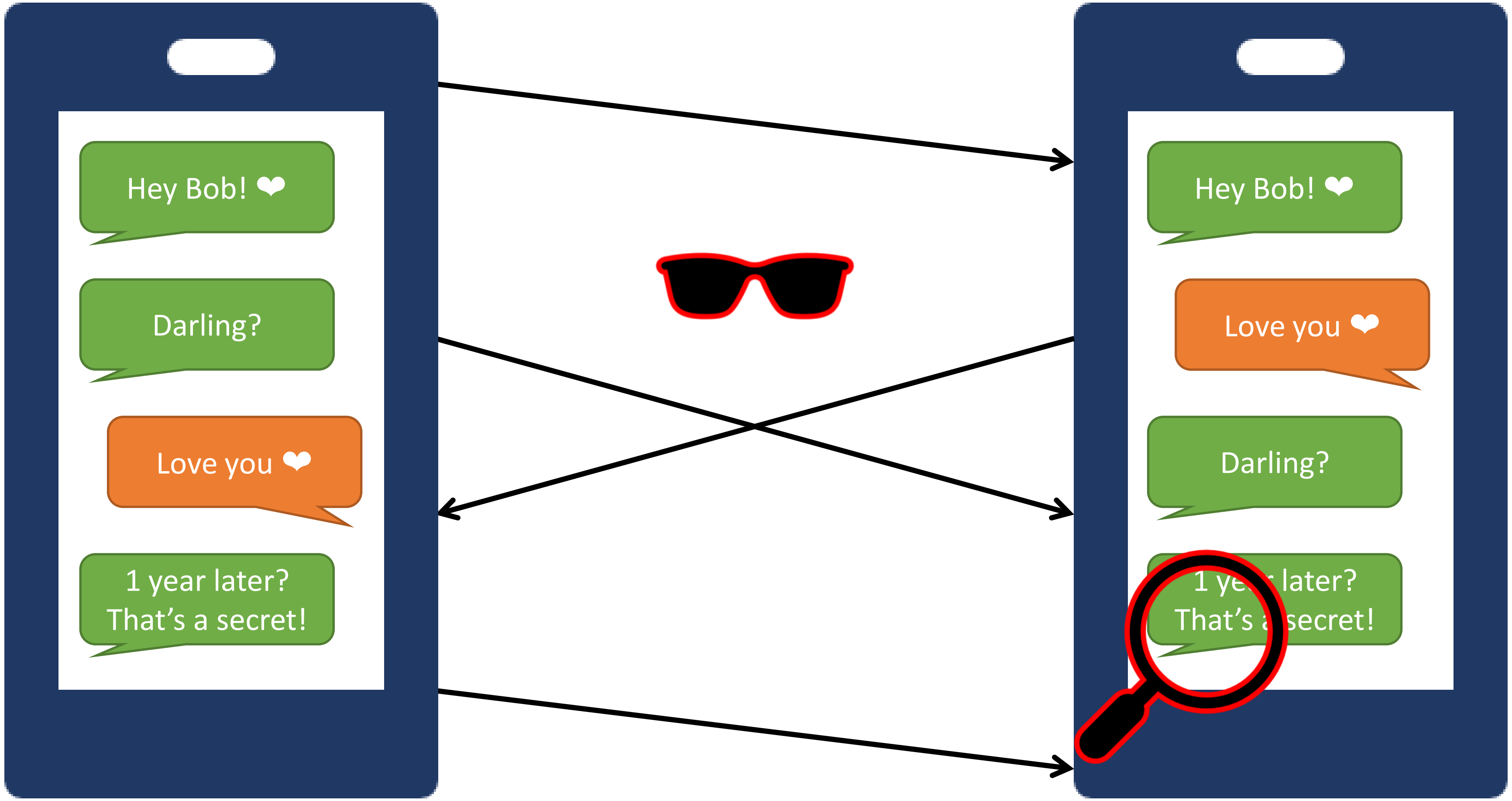
Introduction

- Alice and Bob communicate
- Active adversary
- Long term communication
  - Local (full) state temporarily exposed


Hey Bob! ❤
Darling?
Love you ❤
1 year later? That’s a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That’s a secret!

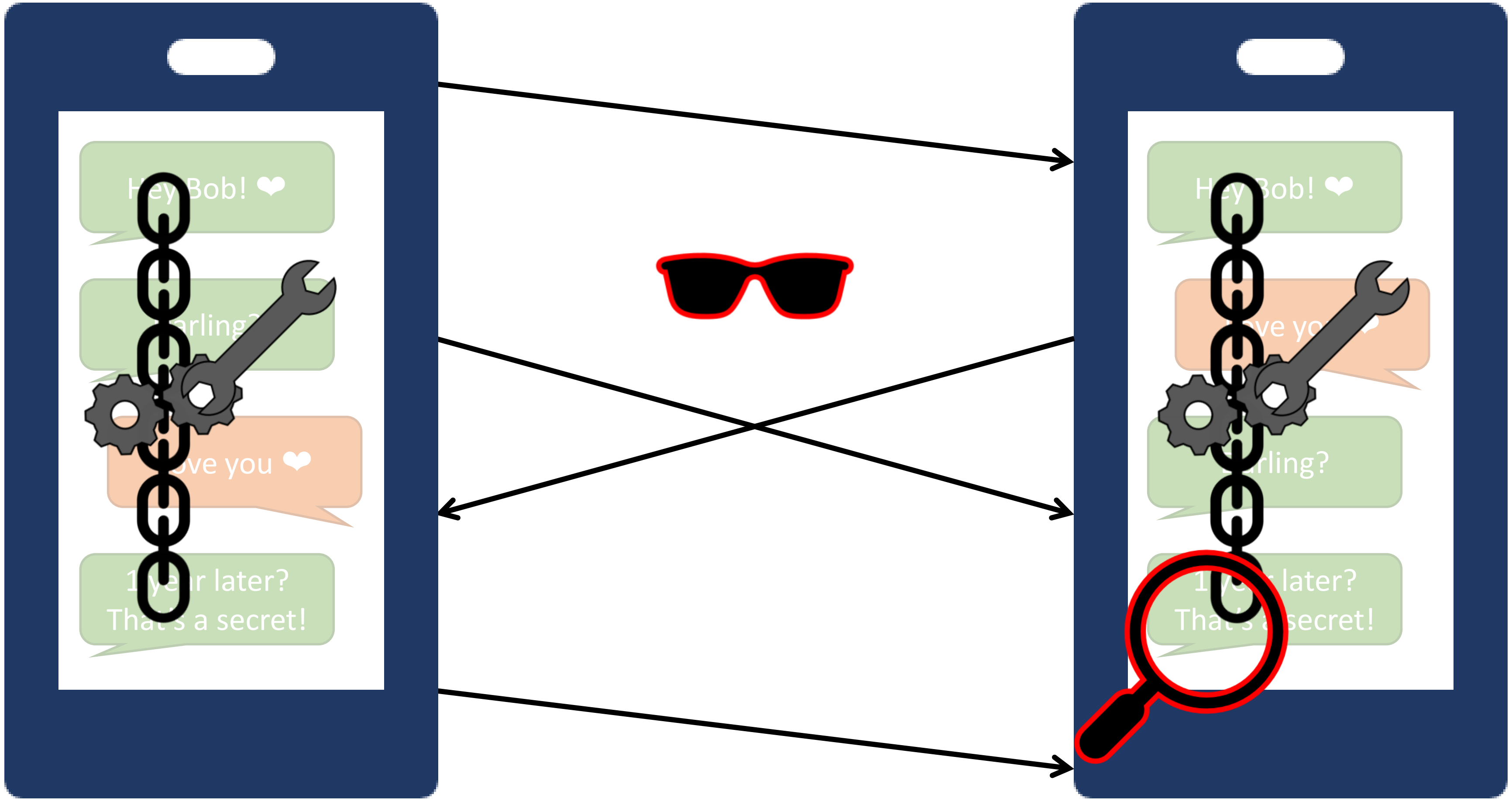
Introduction

- Alice and Bob communicate
- Active adversary
- Long term communication
    - Local (full) state temporarily exposed
- Practical protocols w/o precise security definition
    - E.g., Signal


Hey Bob! ❤
Darling?
Love you ❤
1 year later? That's a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That's a secret!

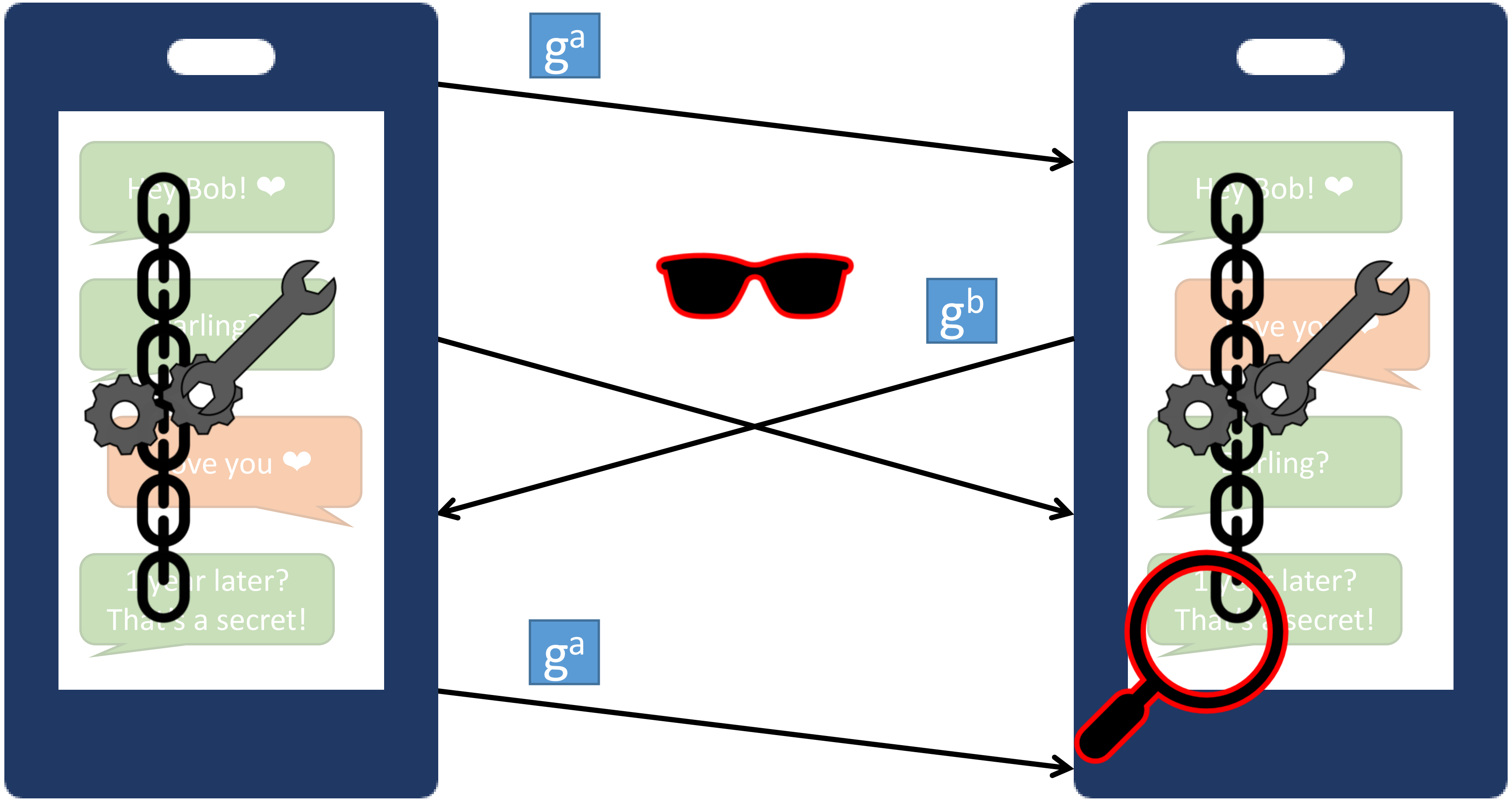
What is Ratcheting?

- Alice and Bob communicate
- Active adversary
- Long term communication
    - Local (full) state temporarily exposed
- Practical protocols w/o precise security definition
    - E.g., Signal


Hey Bob! ❤
Darling?
Love you ❤
1 year later? That's a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That's a secret!

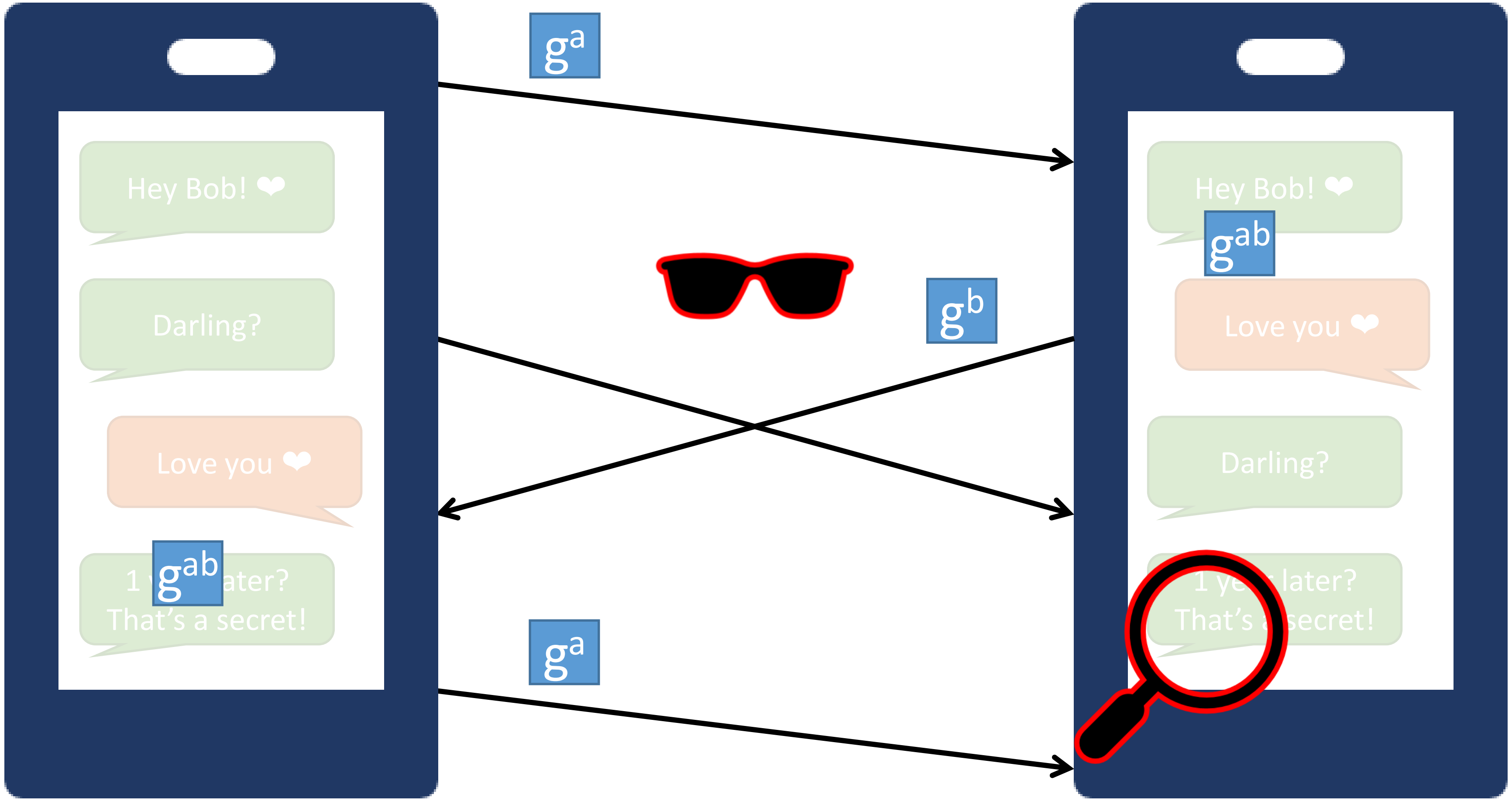
What is Ratcheting?

- Alice and Bob communicate
- Active adversary
- Long term communication
  - Local (full) state temporarily exposed
- Practical protocols w/o precise security definition
  - E.g., Signal



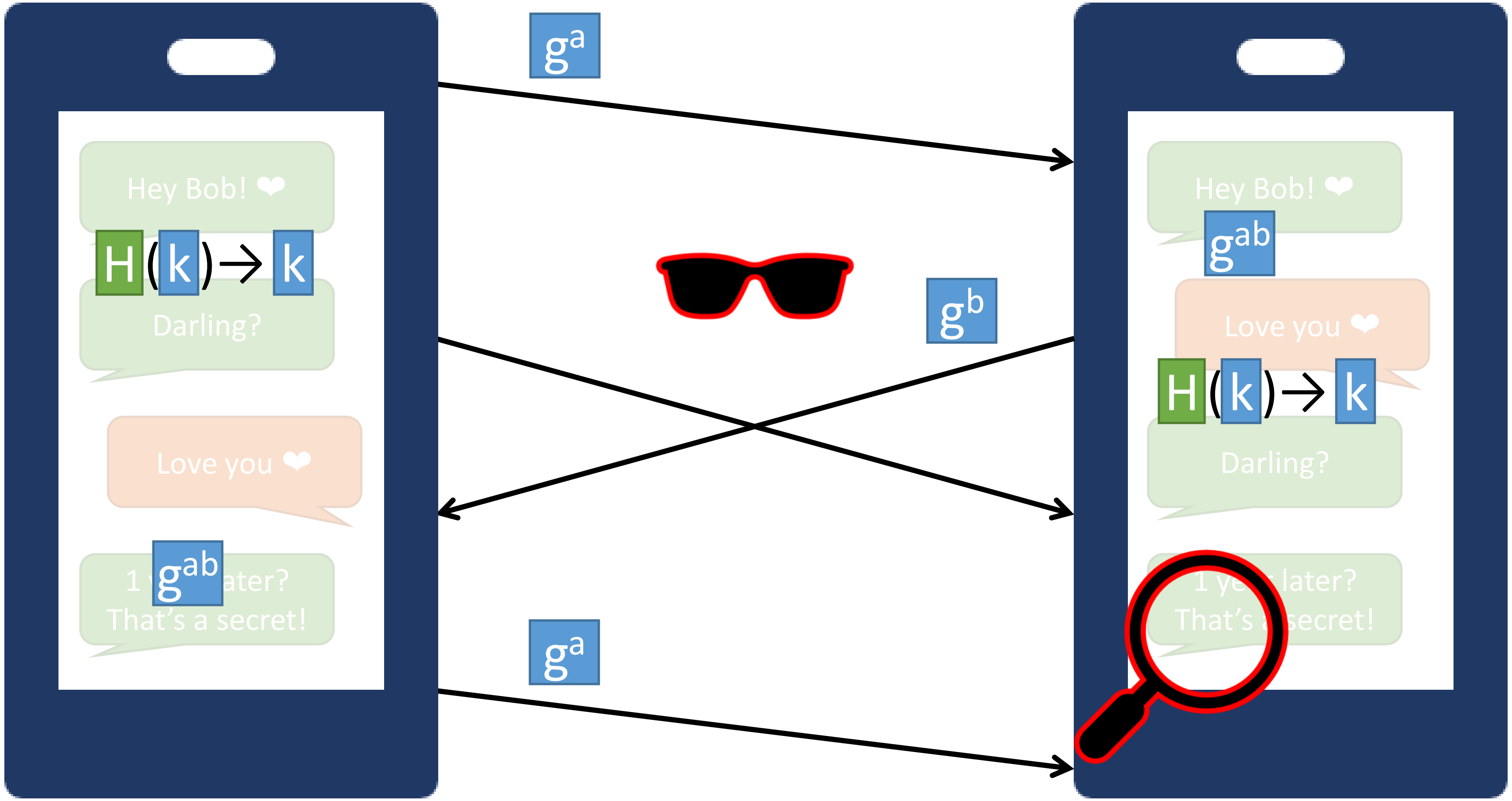
What is Ratcheting?

- Alice and Bob communicate
- Active adversary
- Long term communication
  - Local (full) state temporarily exposed
- Practical protocols w/o precise security definition
  - E.g., Signal



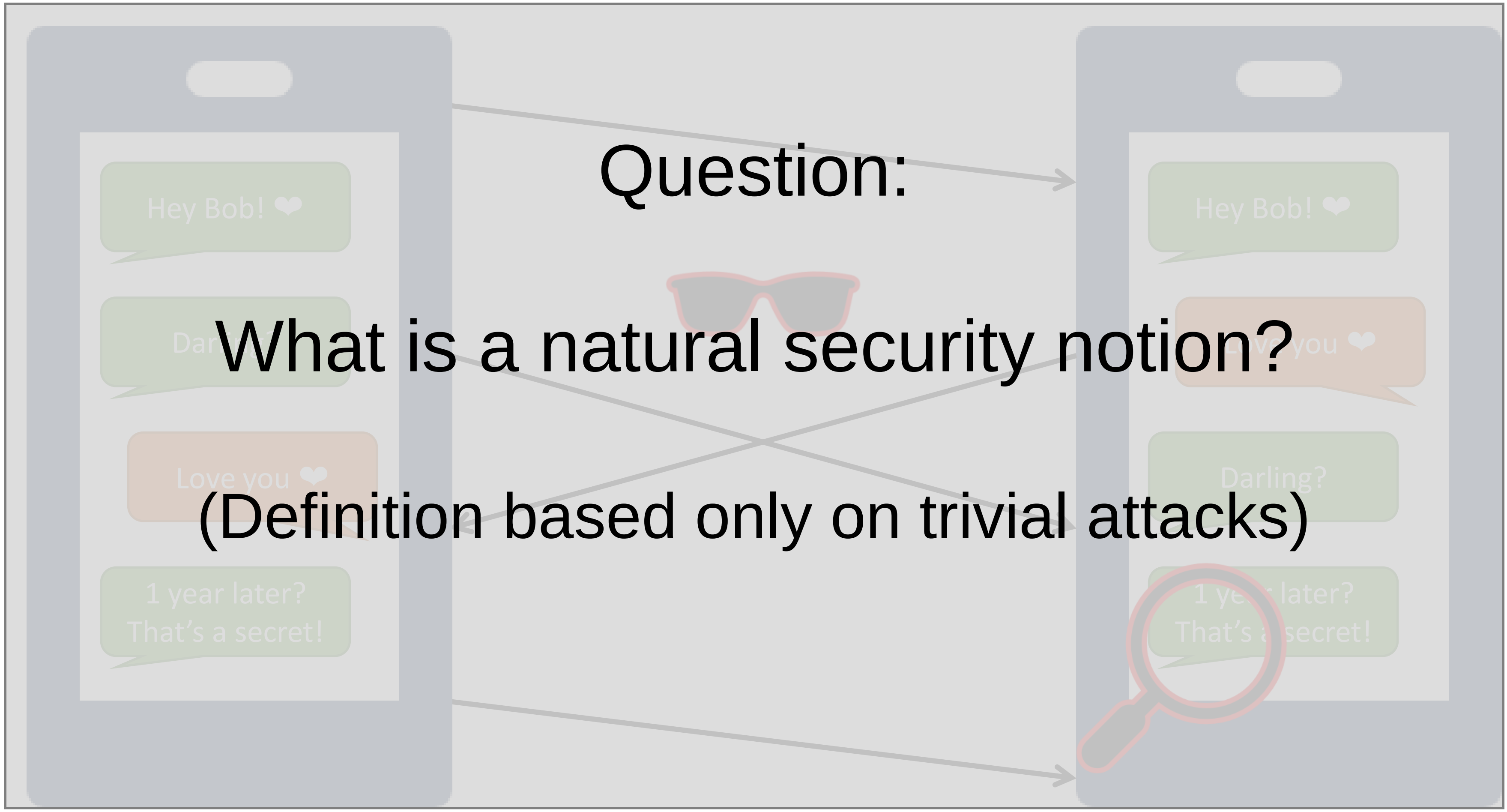
What is Ratcheting?

- Alice and Bob communicate
- Active adversary
- Long term communication
  - Local (full) state temporarily exposed
- Practical protocols w/o precise security definition
  - E.g., Signal



Natural Security Notion for Ratcheting?

- Alice and Bob communicate
- Active adversary
- Long term communication
    - Local (full) state temporarily exposed


Hey Bob! ❤
Darling?
Love you ❤
1 year later? That's a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That's a secret!

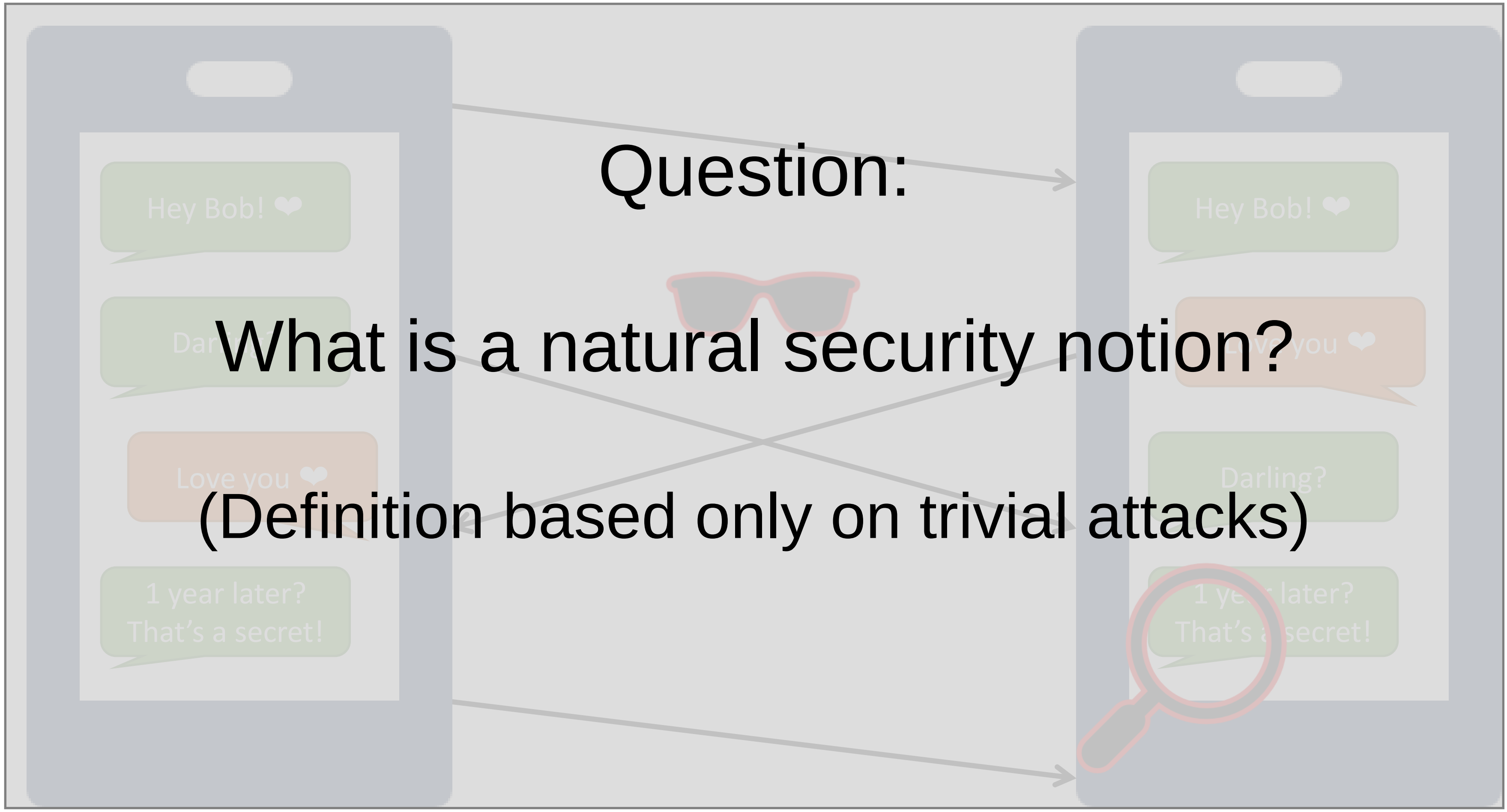
Question:

What is a natural security notion?

(Definition based only on trivial attacks)

Natural Security Notion for Ratcheting?

- Natural security notion
  - Definition based only on trivial attacks
  - Bellare et al. on unidirectional communication C'17
    - Bob cannot be exposed



Hey Bob! ❤
Darling?
Love you ❤
1 year later? That's a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That's a secret!

Question:

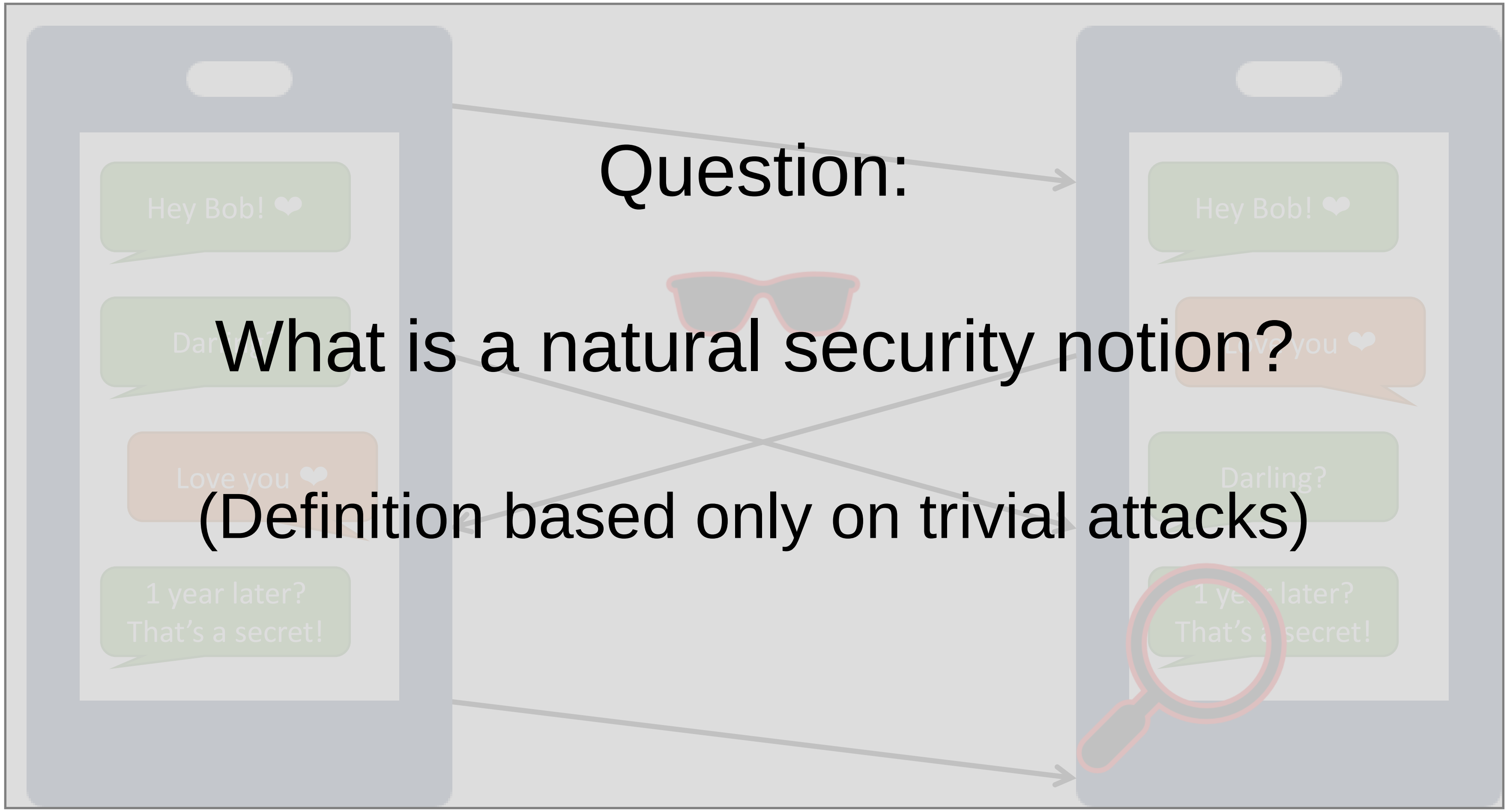
What is a natural security notion?

(Definition based only on trivial attacks)

RUHR
UNIVERSITÄT
BOCHUM
RUB

Natural Security Notion for Ratcheting?

- Natural security notion
  - Definition based only on trivial attacks
  - Bellare et al. on unidirectional communication C'17
    - Bob cannot be exposed



Question:

What is a natural security notion?

(Definition based only on trivial attacks)

Our models require and constructions provide *full* security under:

- Asynchronous communication
- Exposure of both parties

Agenda

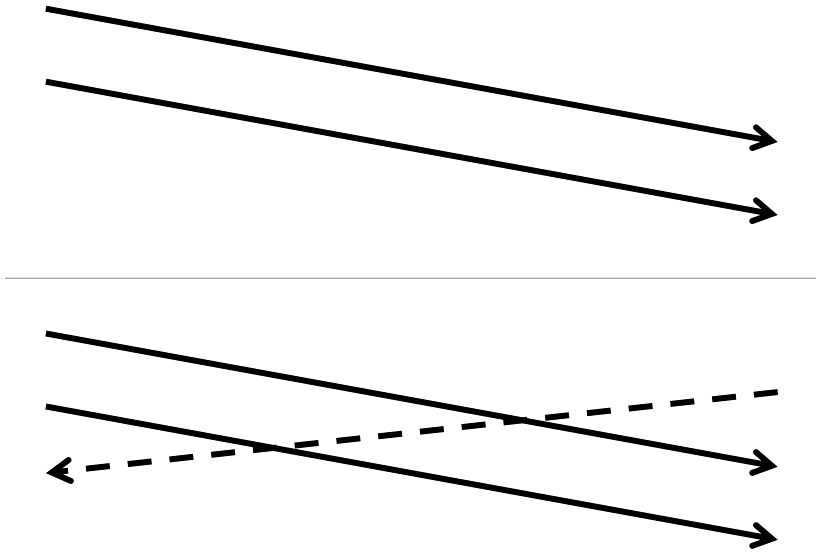
1. **The Primitive Ratcheted Key Exchange**

2. General Adversary Model

3. Unidirectional Ratcheting
   → Model and Construction

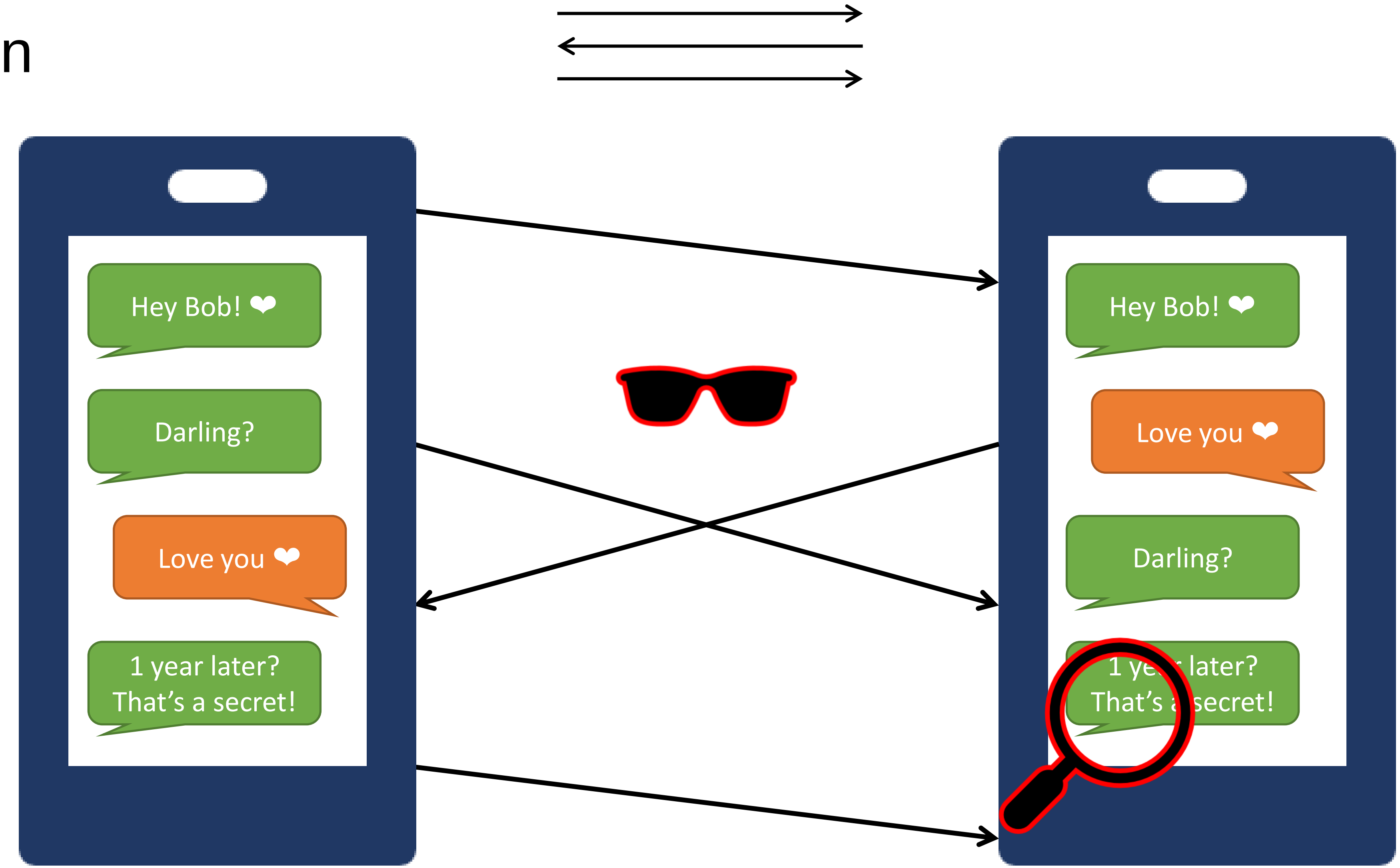
4. Sesquidirectional Ratcheting
   → Model and Construction

5. Results



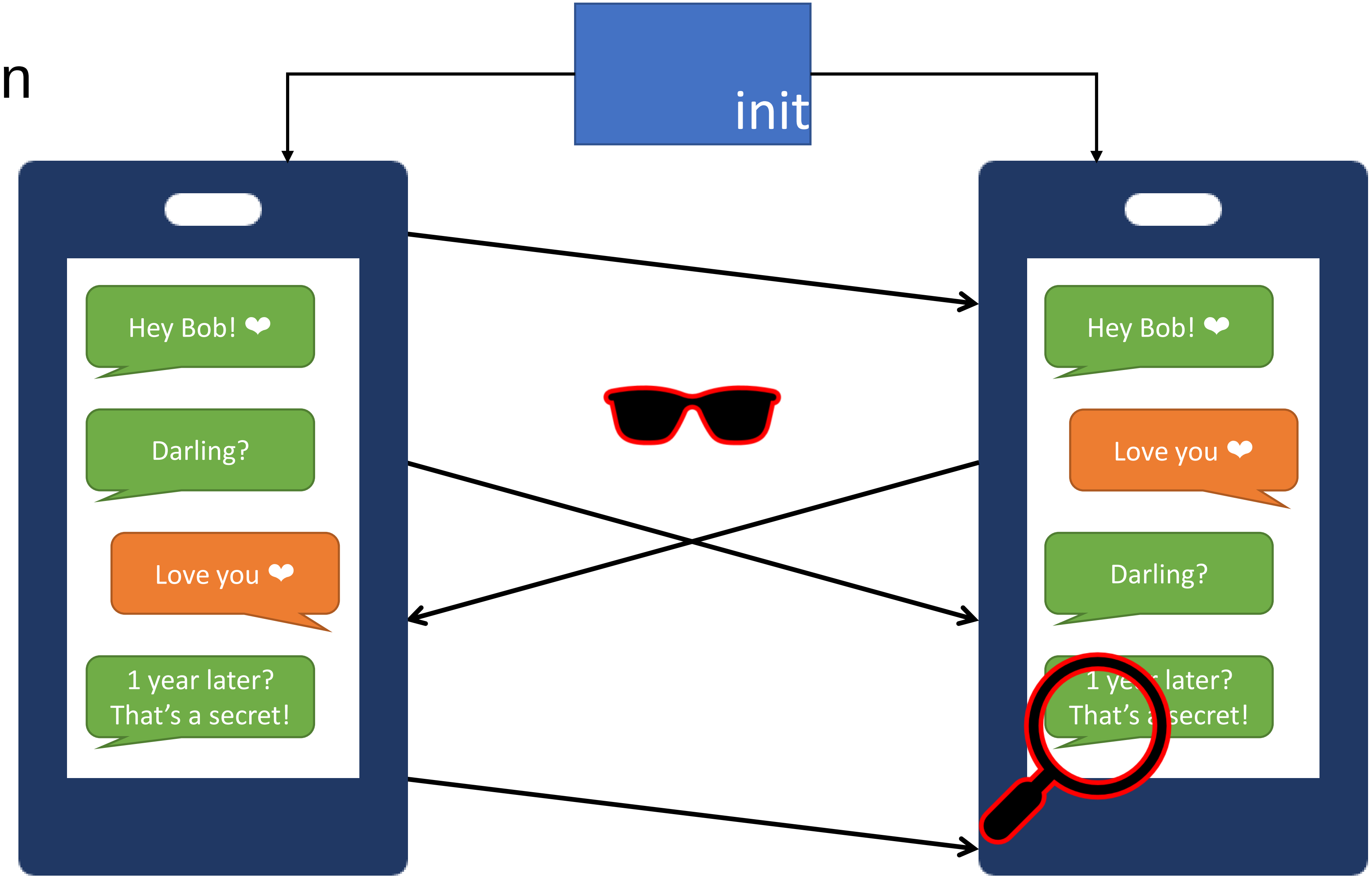
Natural Security Notion for Ratcheting?

- Natural security notion
  - Definition based only on trivial attacks
- Syntax:
  - Initialization


Hey Bob! ❤
Darling?
Love you ❤
1 year later? That’s a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That’s a secret!

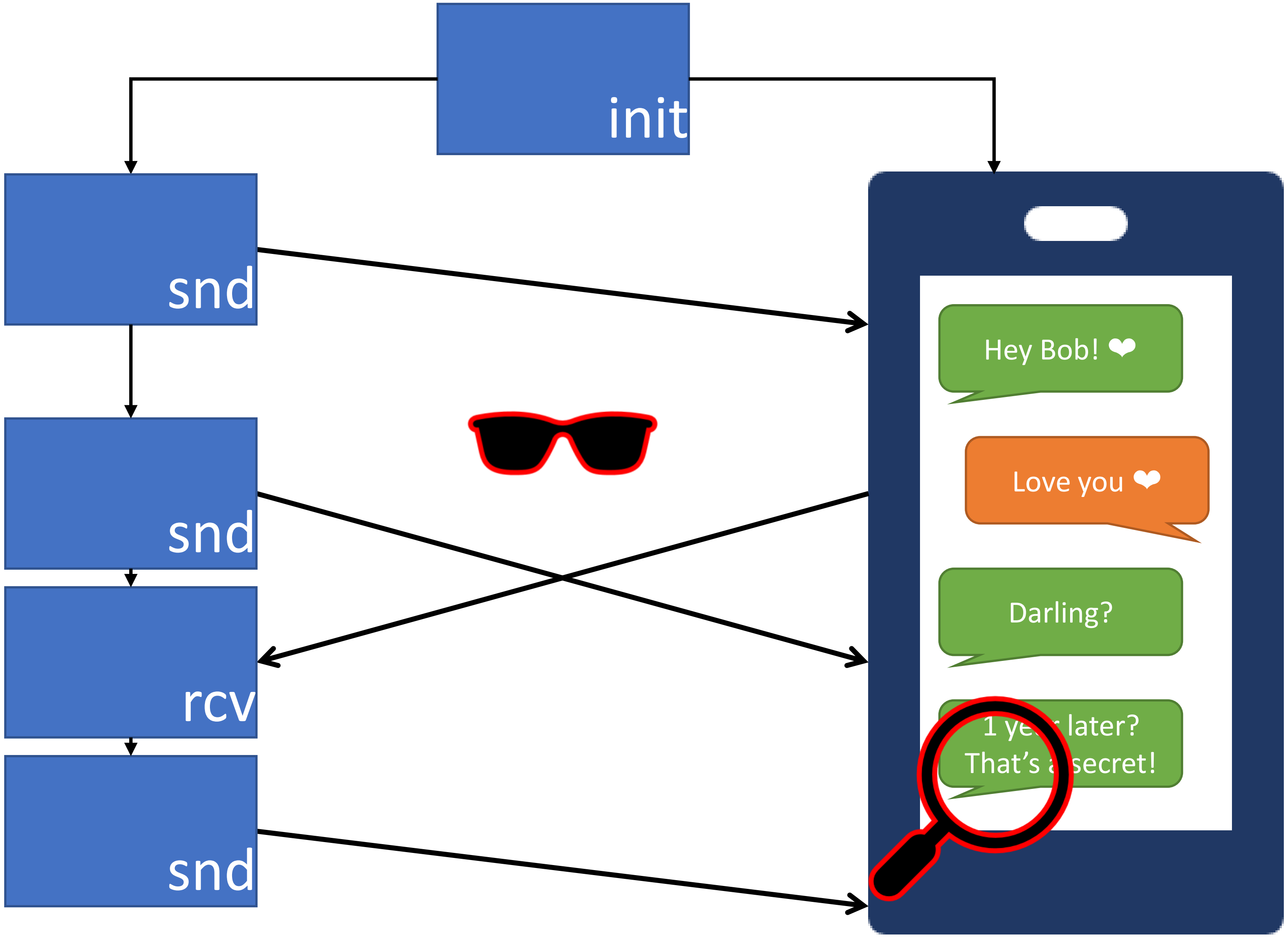
Natural Security Notion for Ratcheting?

- Natural security notion
  - Definition based only on trivial attacks
- Syntax:
  - Initialization


init
Hey Bob! ❤
Darling?
Love you ❤
1 year later? That's a secret!
Hey Bob! ❤
Love you ❤
Darling?
1 year later? That's a secret!

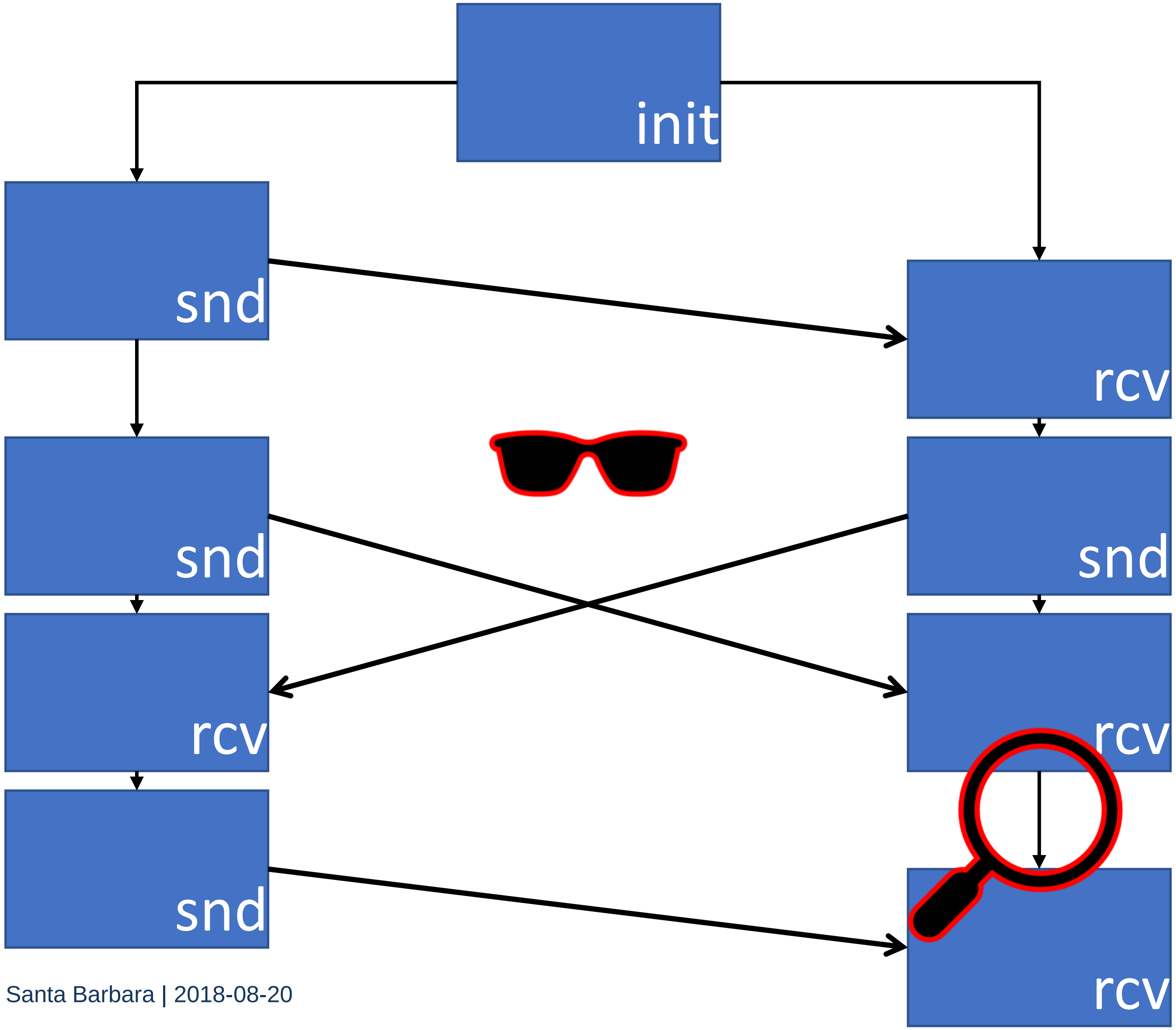
Natural Security Notion for Ratcheting?

- Natural security notion
  - Definition based only on trivial attacks
- Syntax:
  - Initialization
  - Sending & receiving



Natural Security Notion for Ratcheting?

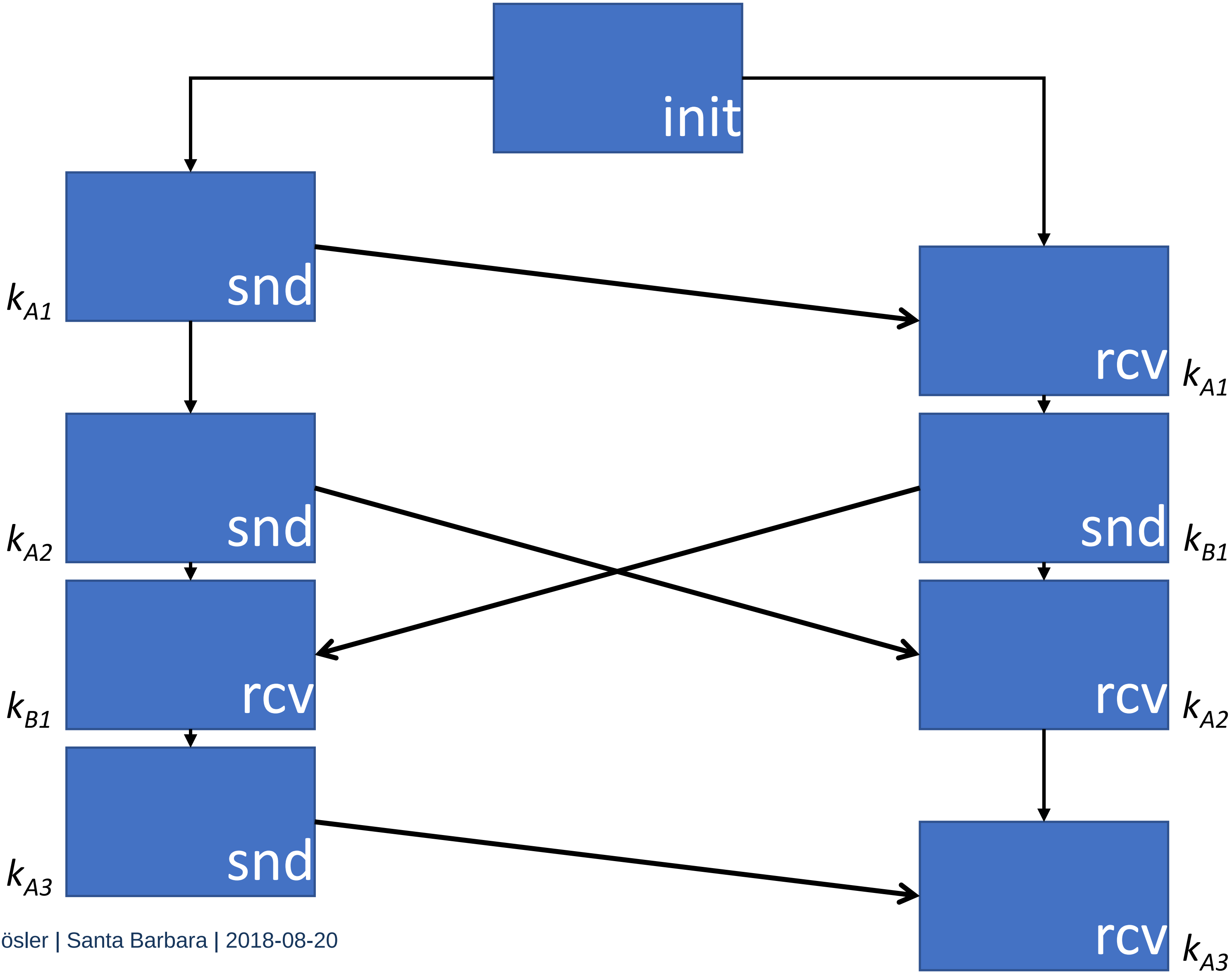
- Natural security notion
  - Definition based only on trivial attacks
- Syntax:
  - Initialization
  - Sending & receiving



Natural Security Notion for Ratcheting?

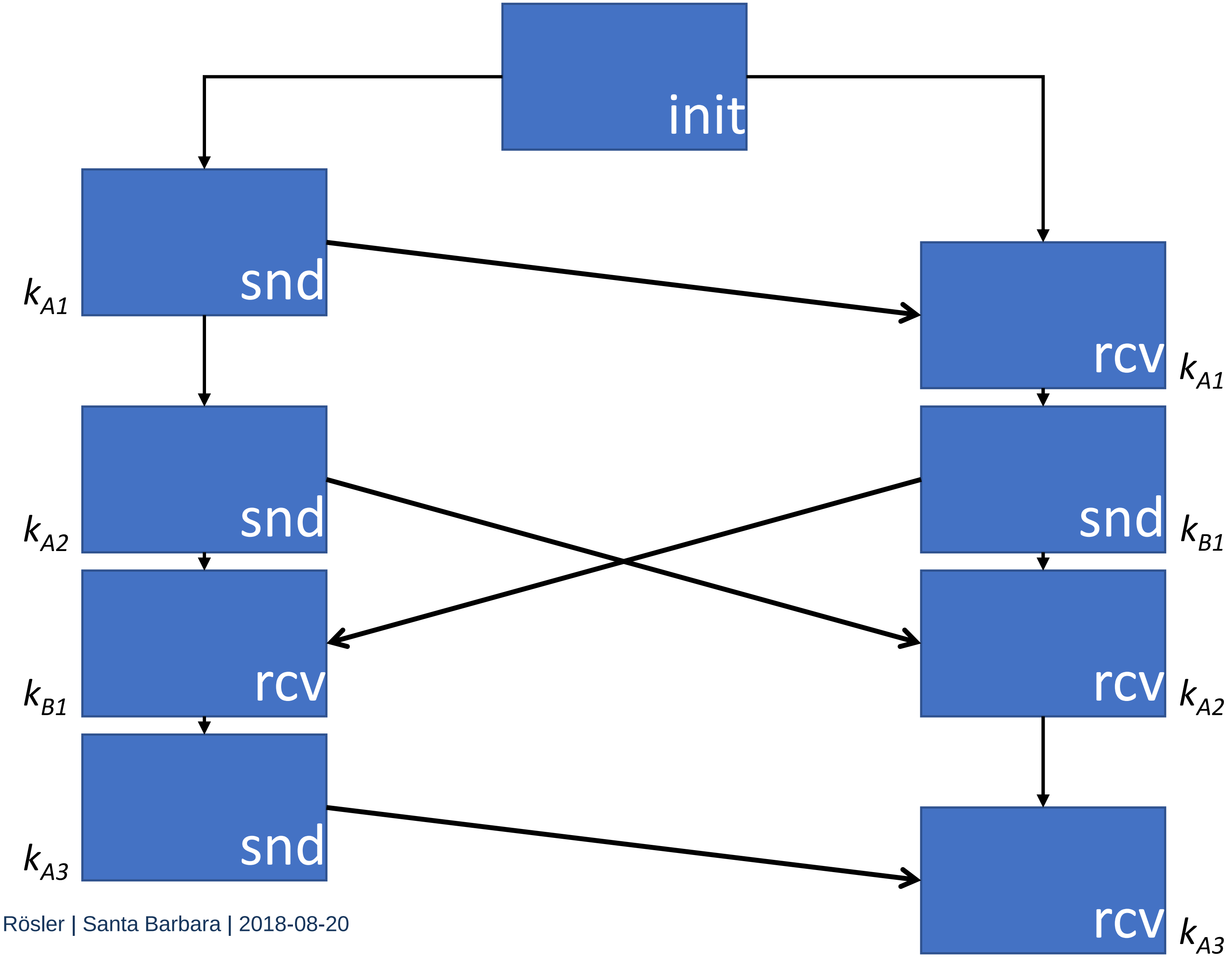
- Natural security notion
  - Definition based only on trivial attacks
- Syntax:
  - Initialization
  - Sending & receiving
  - Key exchange
    - Consecutive establishment of keys in session

  ≠ *Authenticated key exchange*!



Natural Security Notion for Ratcheting?

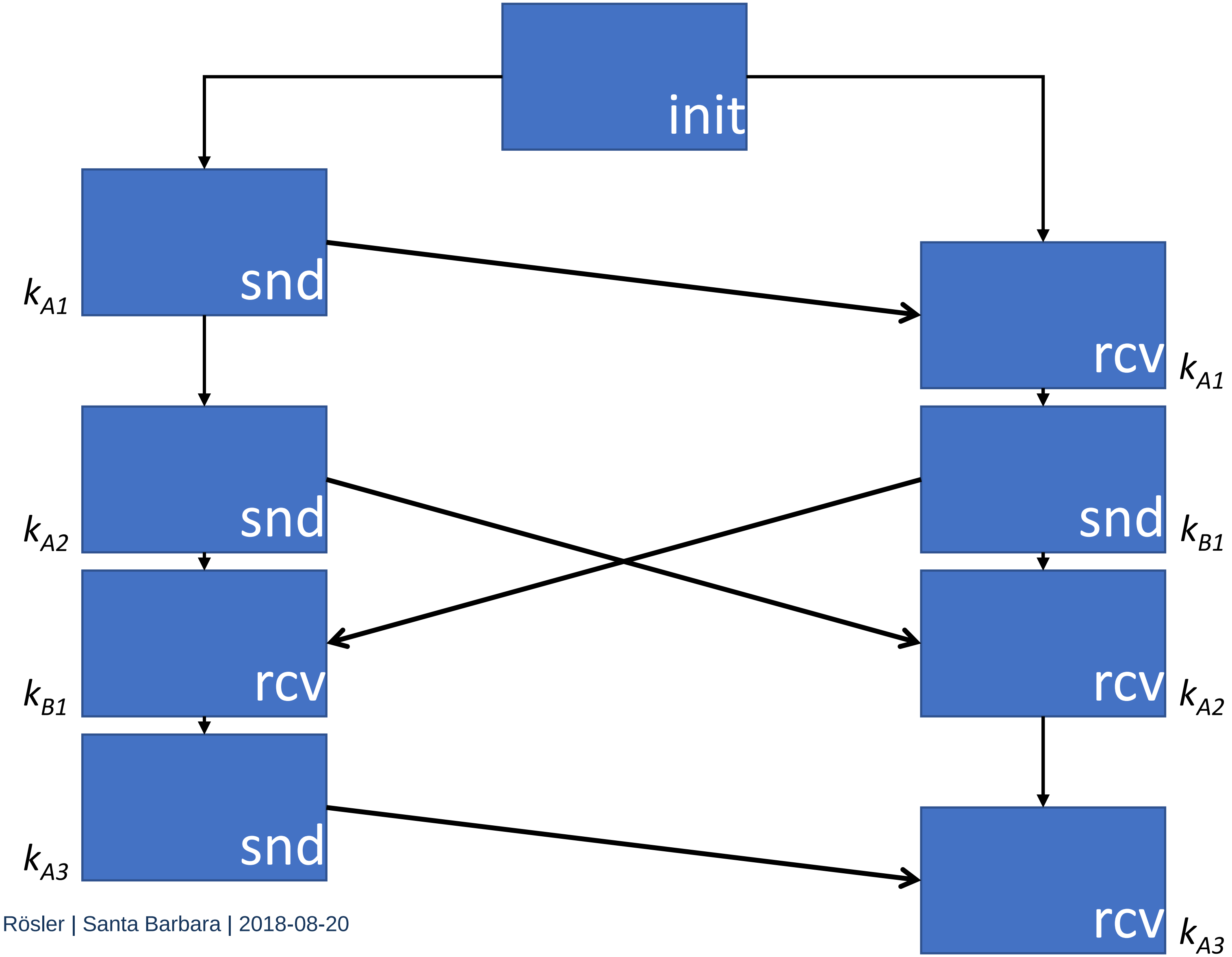
- Natural security notion
  - Definition based only on trivial attacks
- Syntax:
  - Initialization
  - Sending & receiving
  - Key exchange
    - Composition in Bellare et al. C’17



Three Variants of Ratcheting

- Bidirectional ratcheting is complicated

→ Understand its components

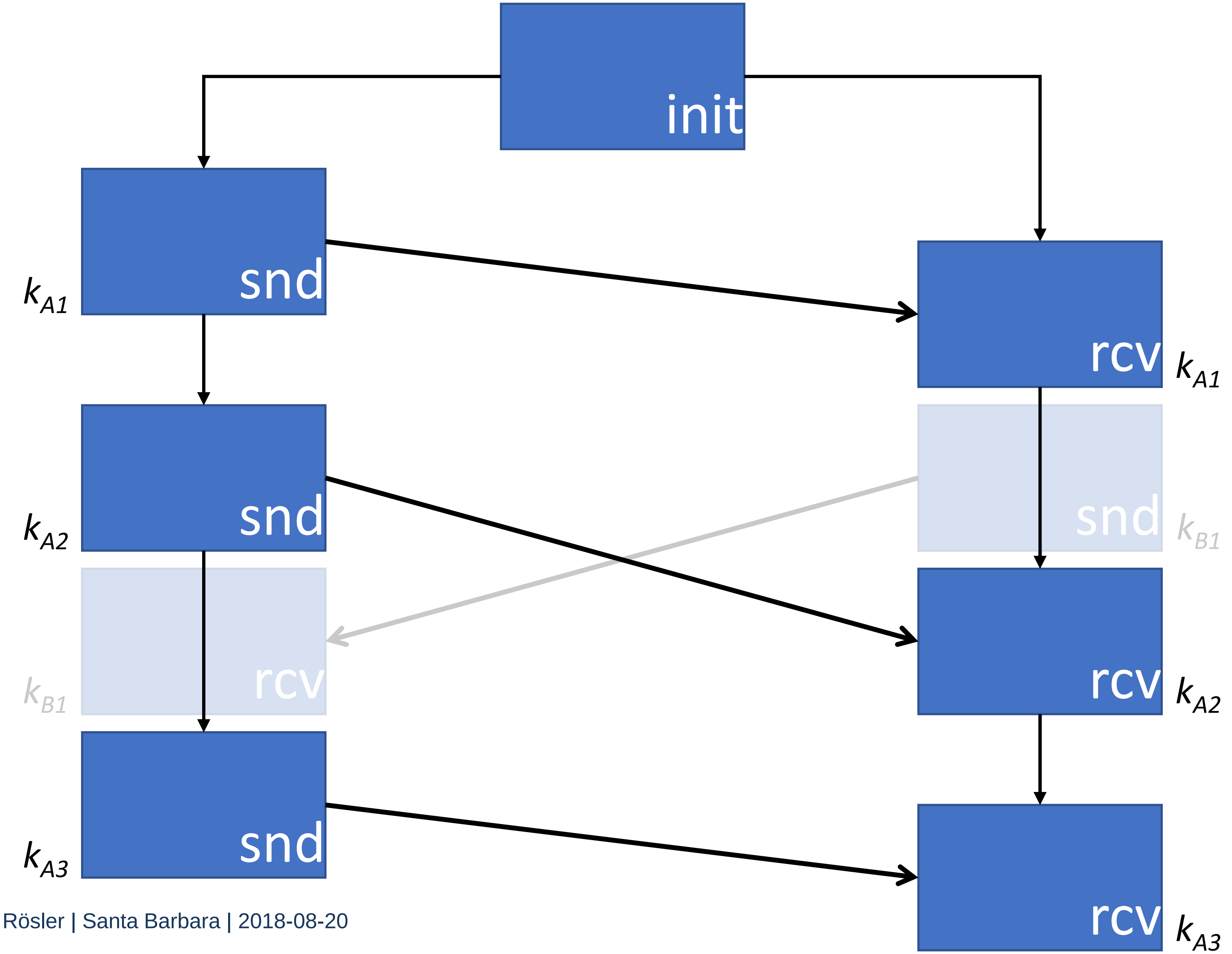


Three Variants of Ratcheting

- Bidirectional ratcheting is complicated

→ Understand its components:

  - Unidirectional key establishment

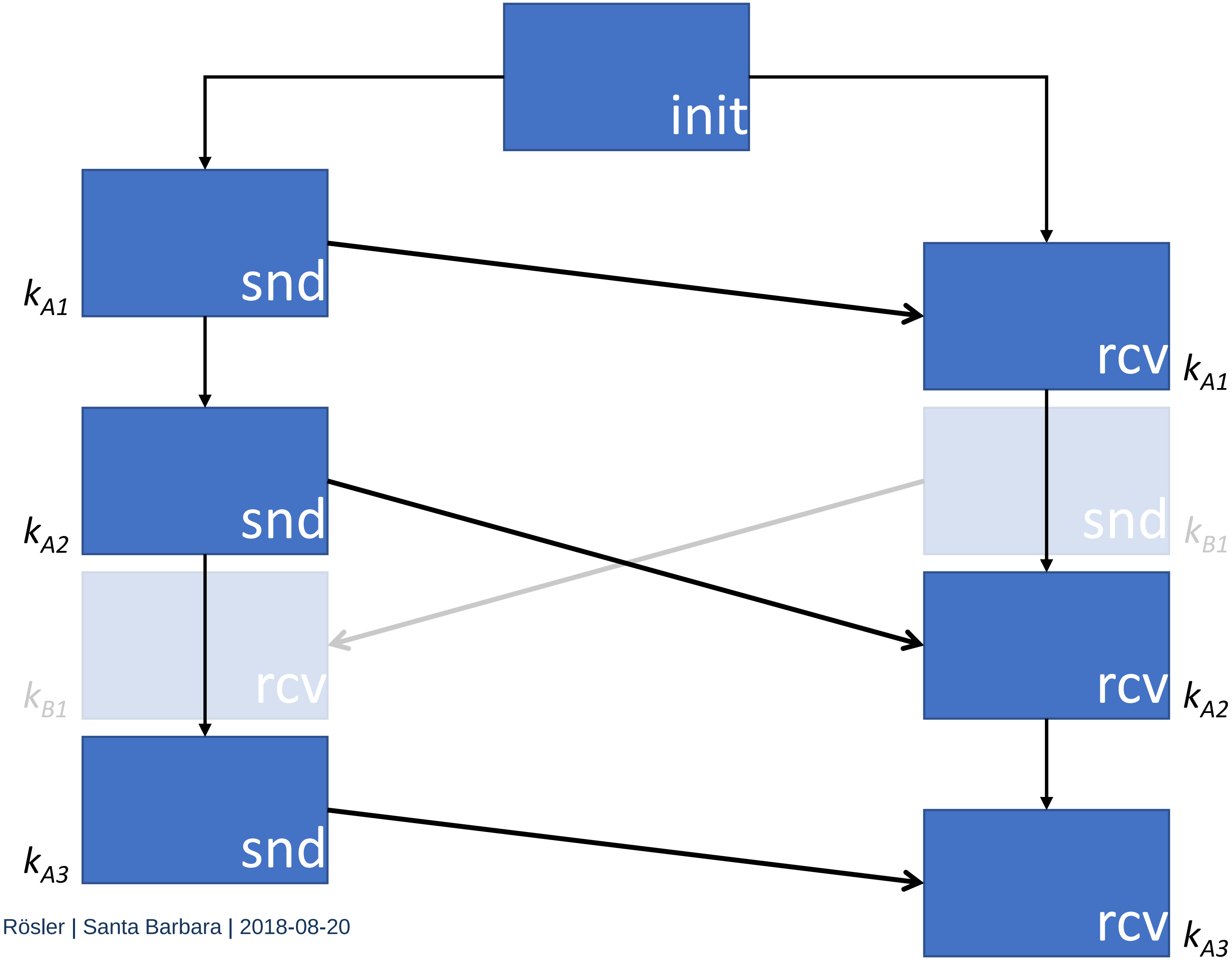


Three Variants of Ratcheting

- Bidirectional ratcheting is complicated

→ Understand its components:

  - Unidirectional key establishment
  - Alice initiates computation of new key
  - Bob does not respond



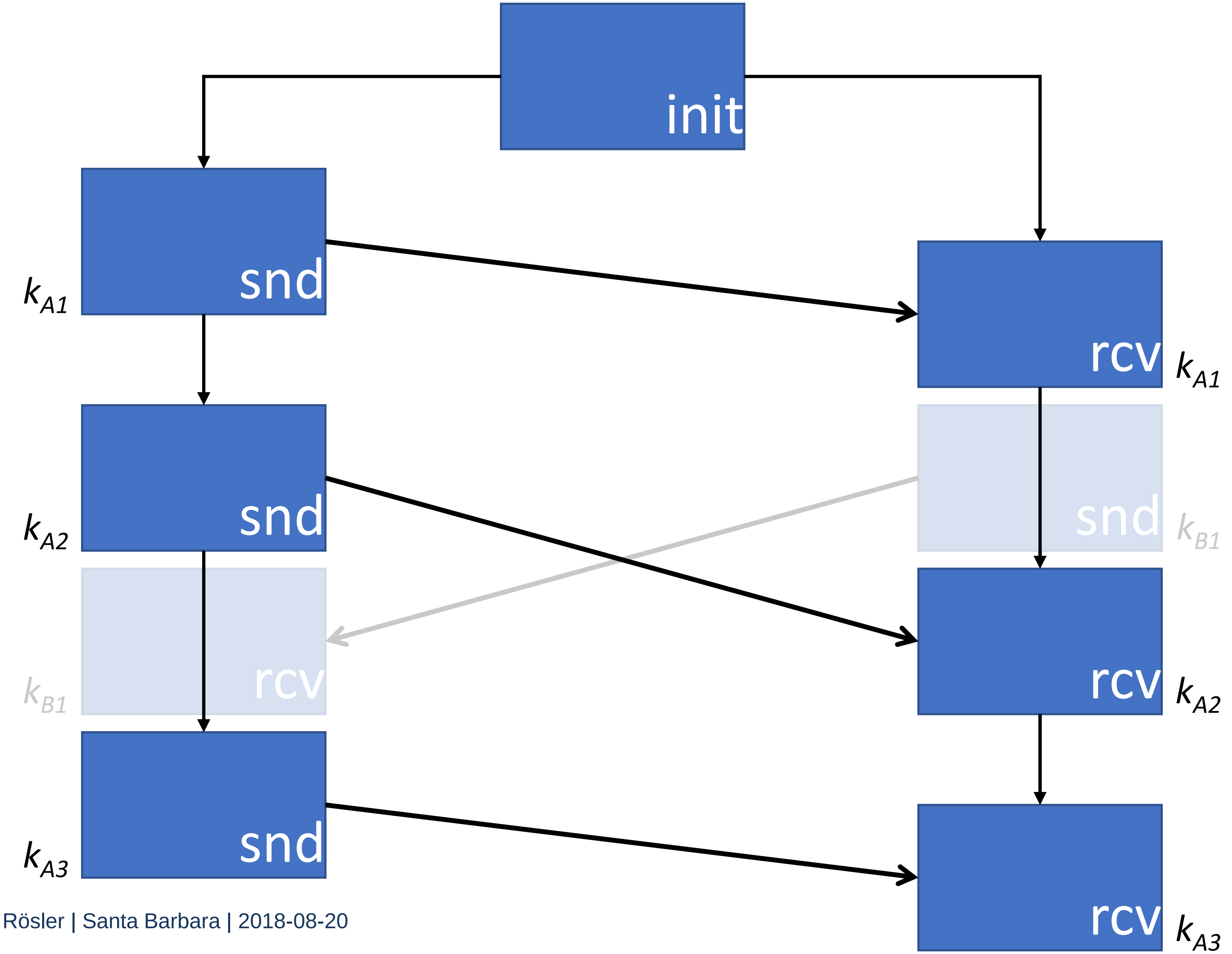
RUHR UNIVERSITÄT BOCHUM RUB

Three Variants of Ratcheting

- Bidirectional ratcheting is complicated

→ Understand its components:

  - Unidirectional ratcheted key exchange (RKE)



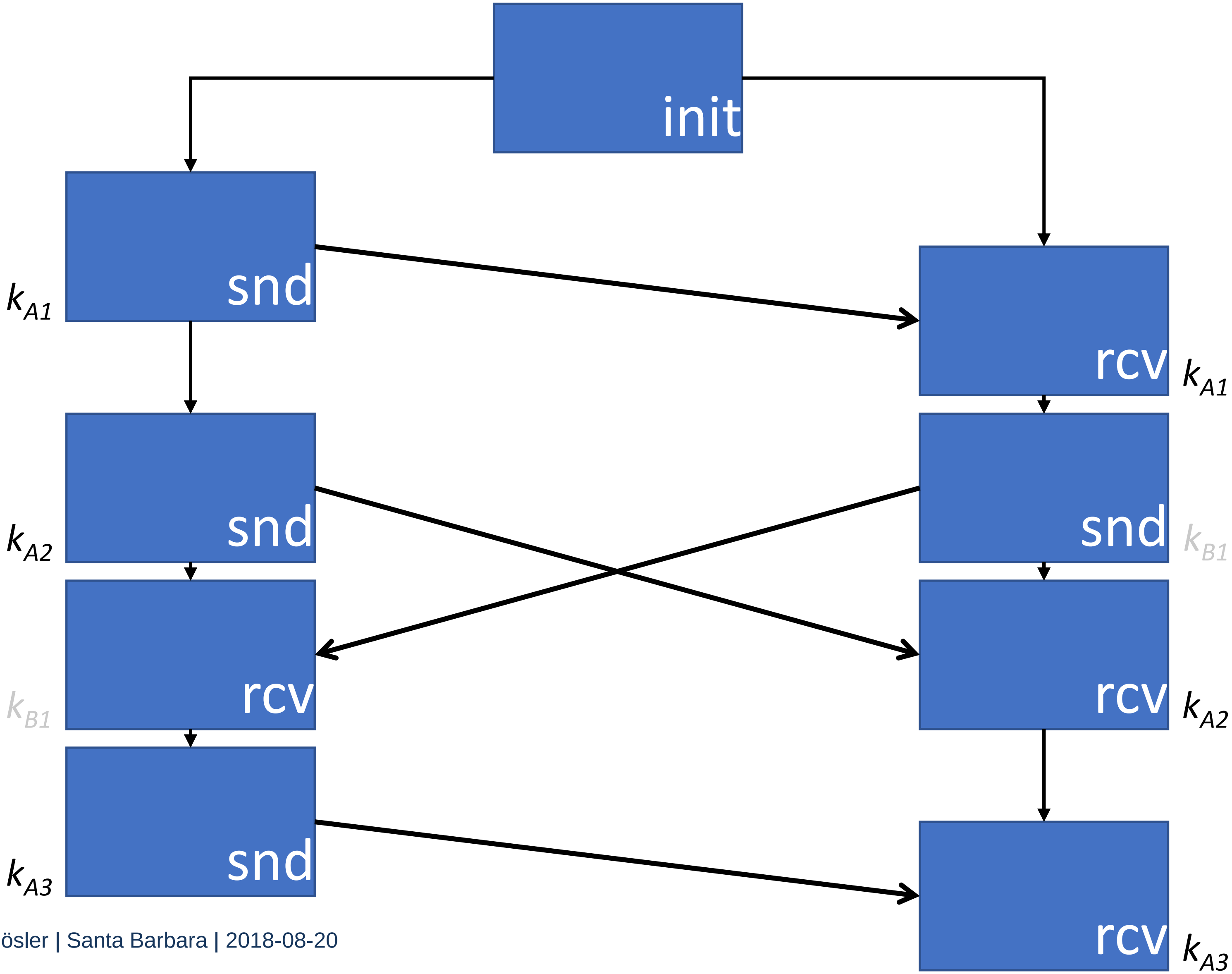
Three Variants of Ratcheting

- Bidirectional ratcheting is complicated

→ Understand its components:

  - Unidirectional RKE
  - Sesquidirectional RKE
  - Bob contributes (but cannot establish keys)
  - Adds security

(sesqui = 1.5)

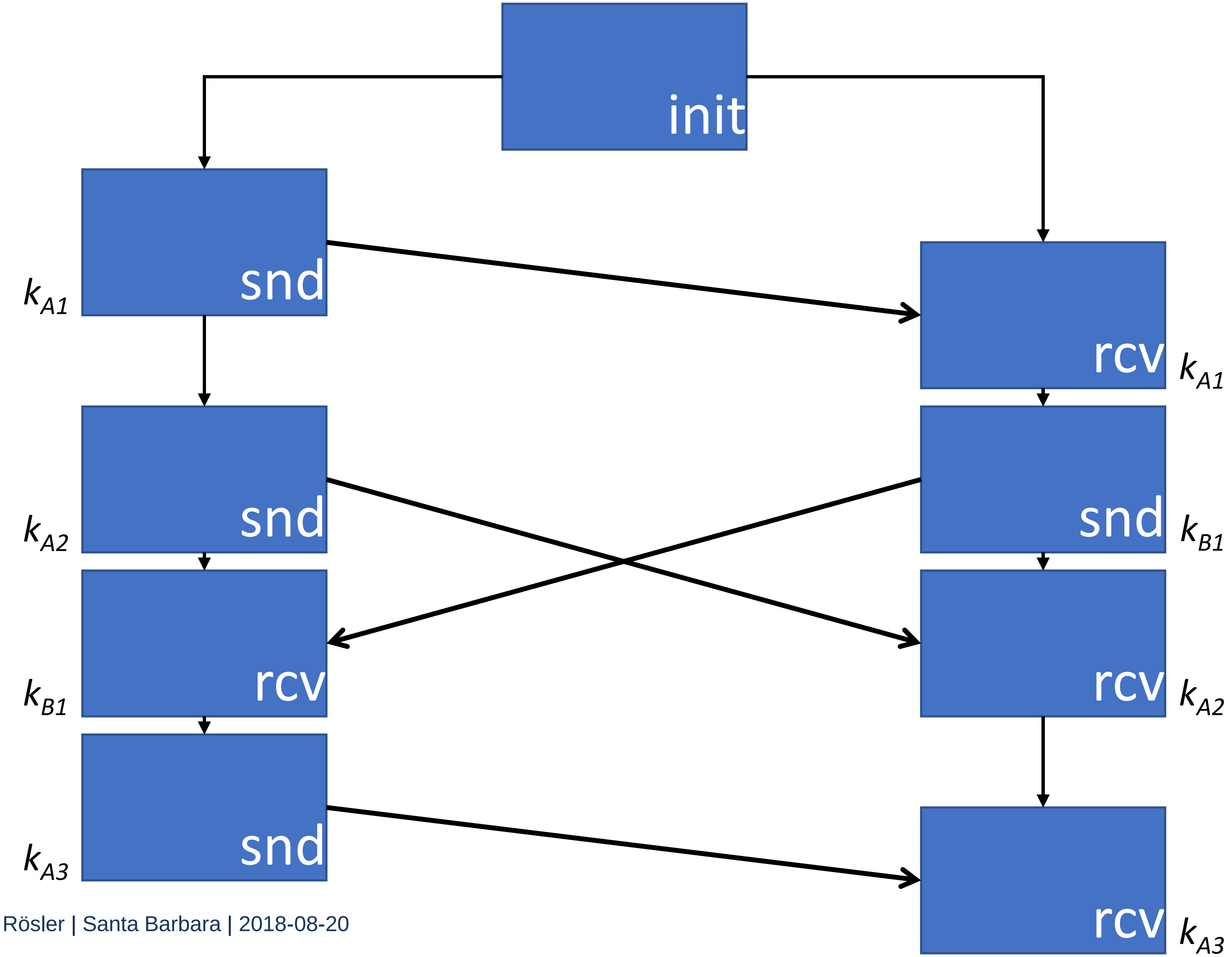


Three Variants of Ratcheting

- Bidirectional ratcheting is complicated

→ Understand its components:

  - Unidirectional RKE
  - Sesquidirectional RKE
  - Symmetric roles

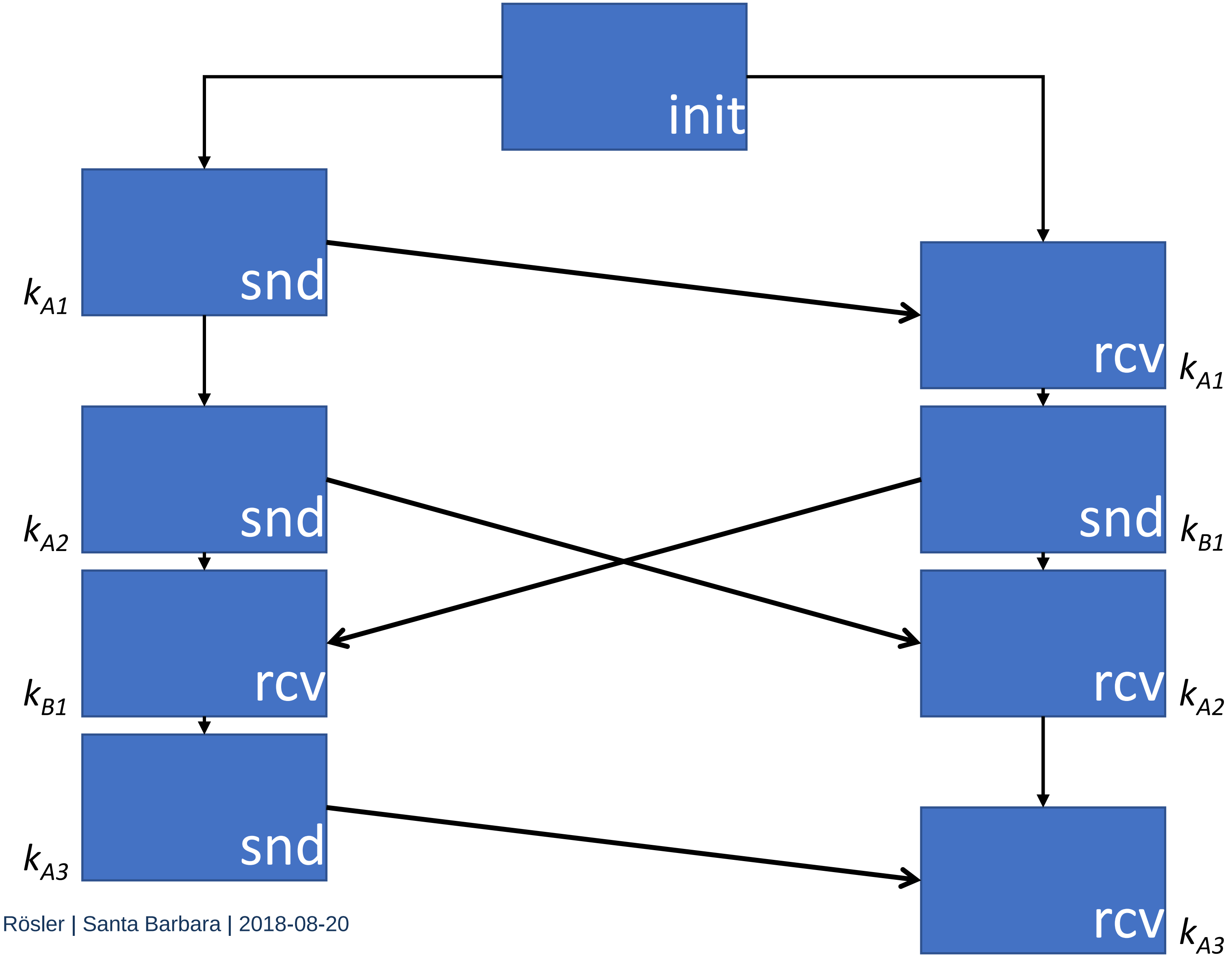


Three Variants of Ratcheting

- Bidirectional ratcheting is complicated

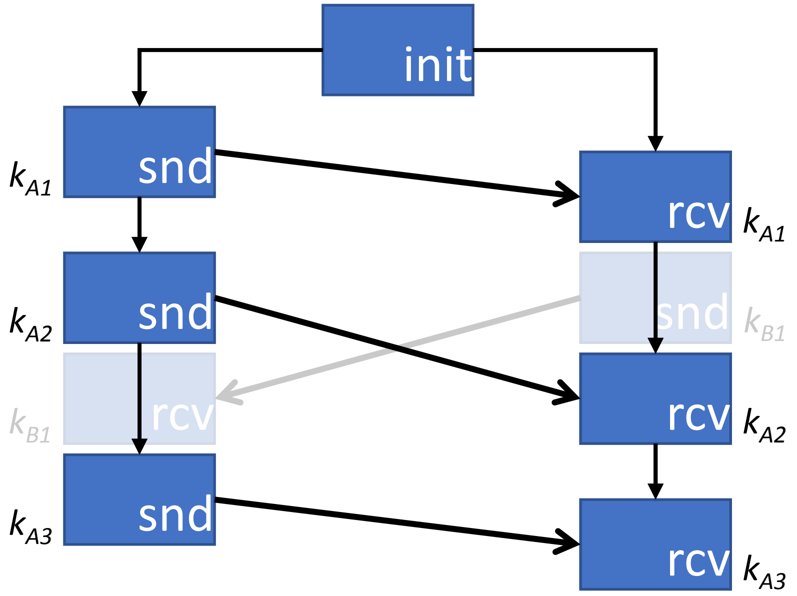
→ Understand its components:

  - Unidirectional RKE
  - Sesquidirectional RKE
  - Symmetric roles
  - Bidirectional RKE = 2x Sesquid. RKE (extended version)



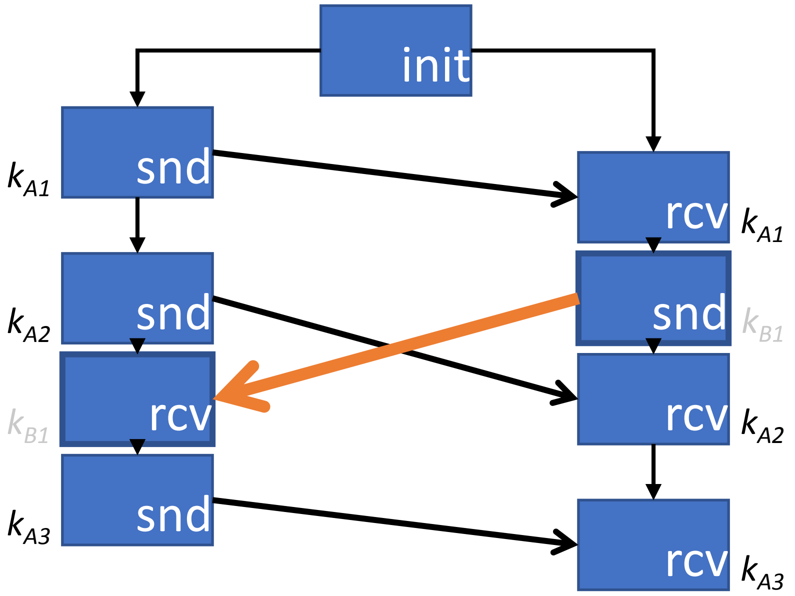
Three Variants of Ratcheting

Unidirectional RKE
(+ Exposure of Bob)



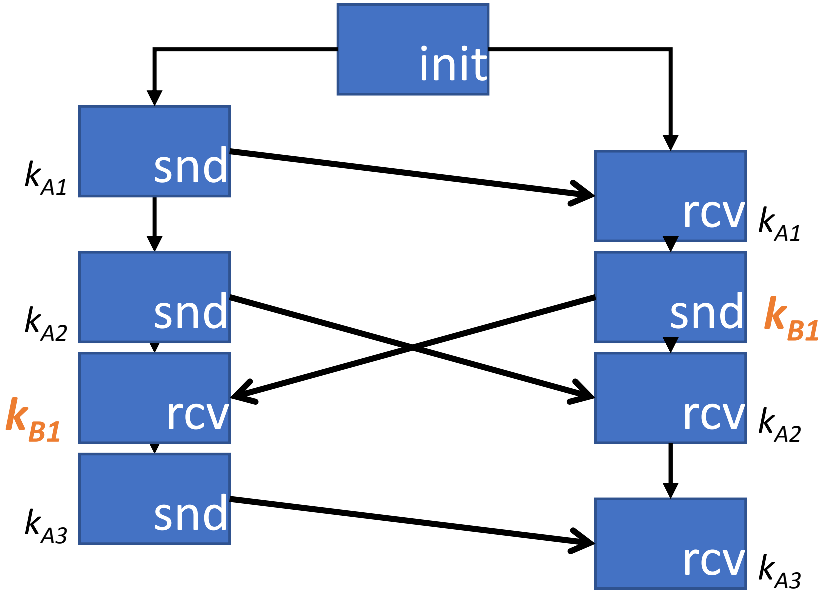
No responses from Bob

Sesquidirectional RKE



Bob’s responses only help to recover

Bidirectional RKE



Symmetric roles

(extended version)

Agenda

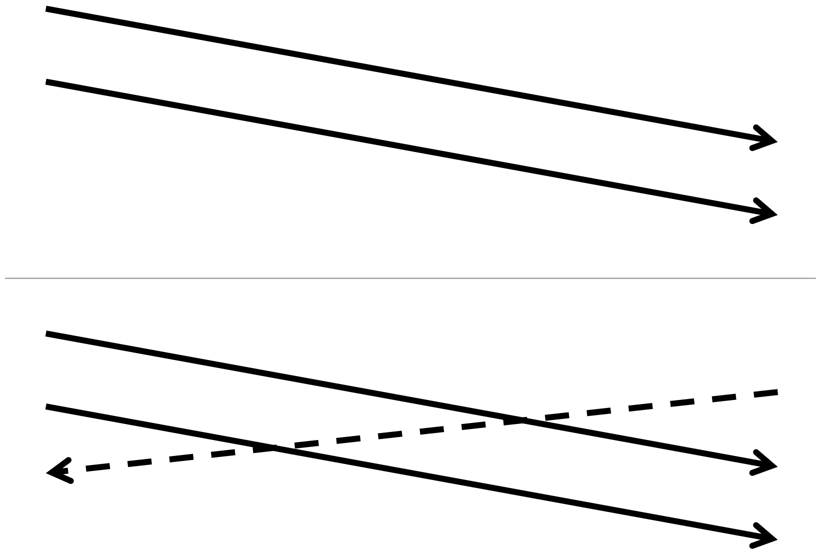
1. The Primitive Ratcheted Key Exchange

2. **General Adversary Model**

3. Unidirectional Ratcheting
   → Model and Construction

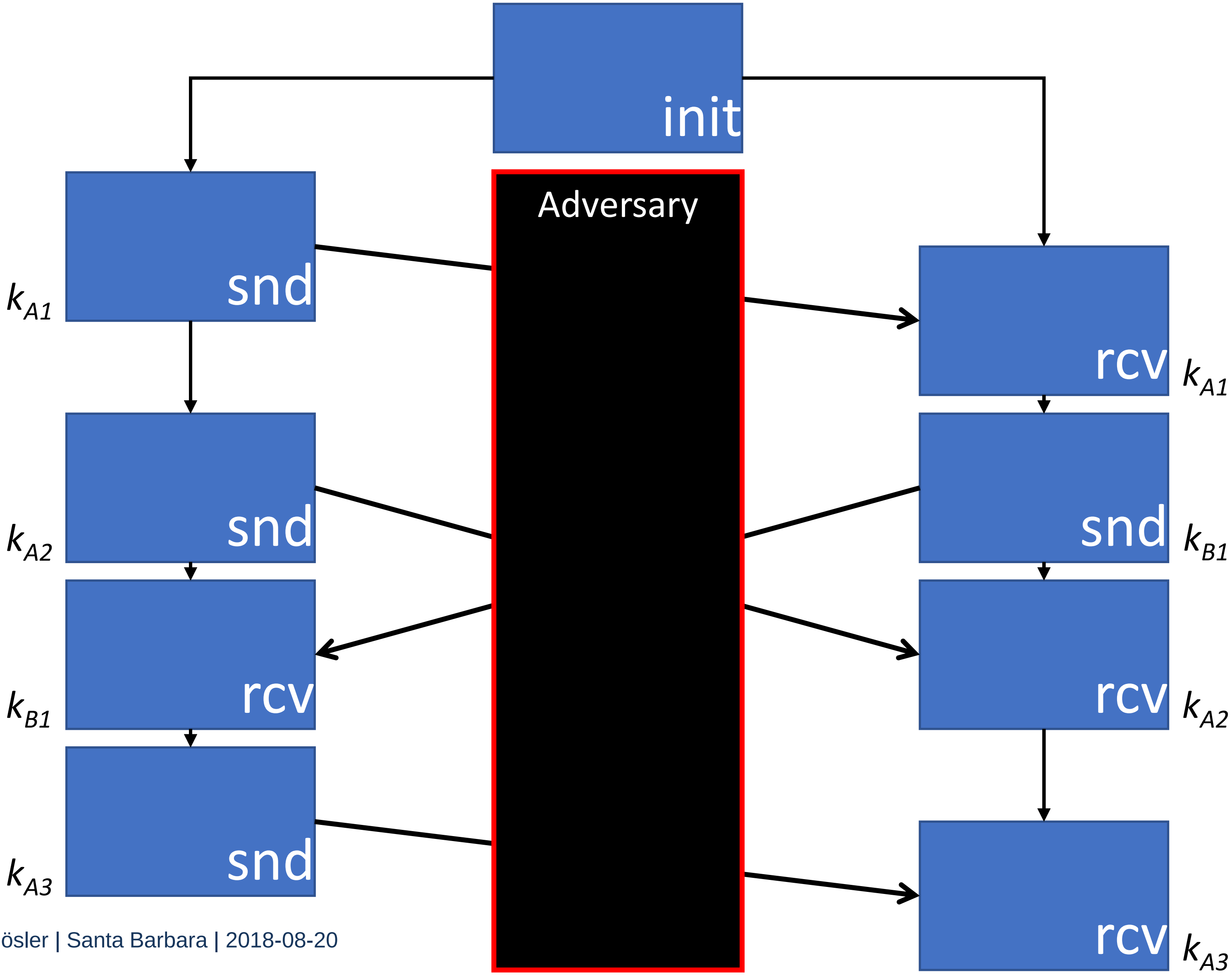
4. Sesquidirectional Ratcheting
   → Model and Construction

5. Results



Modeling Ratcheted Key Exchange

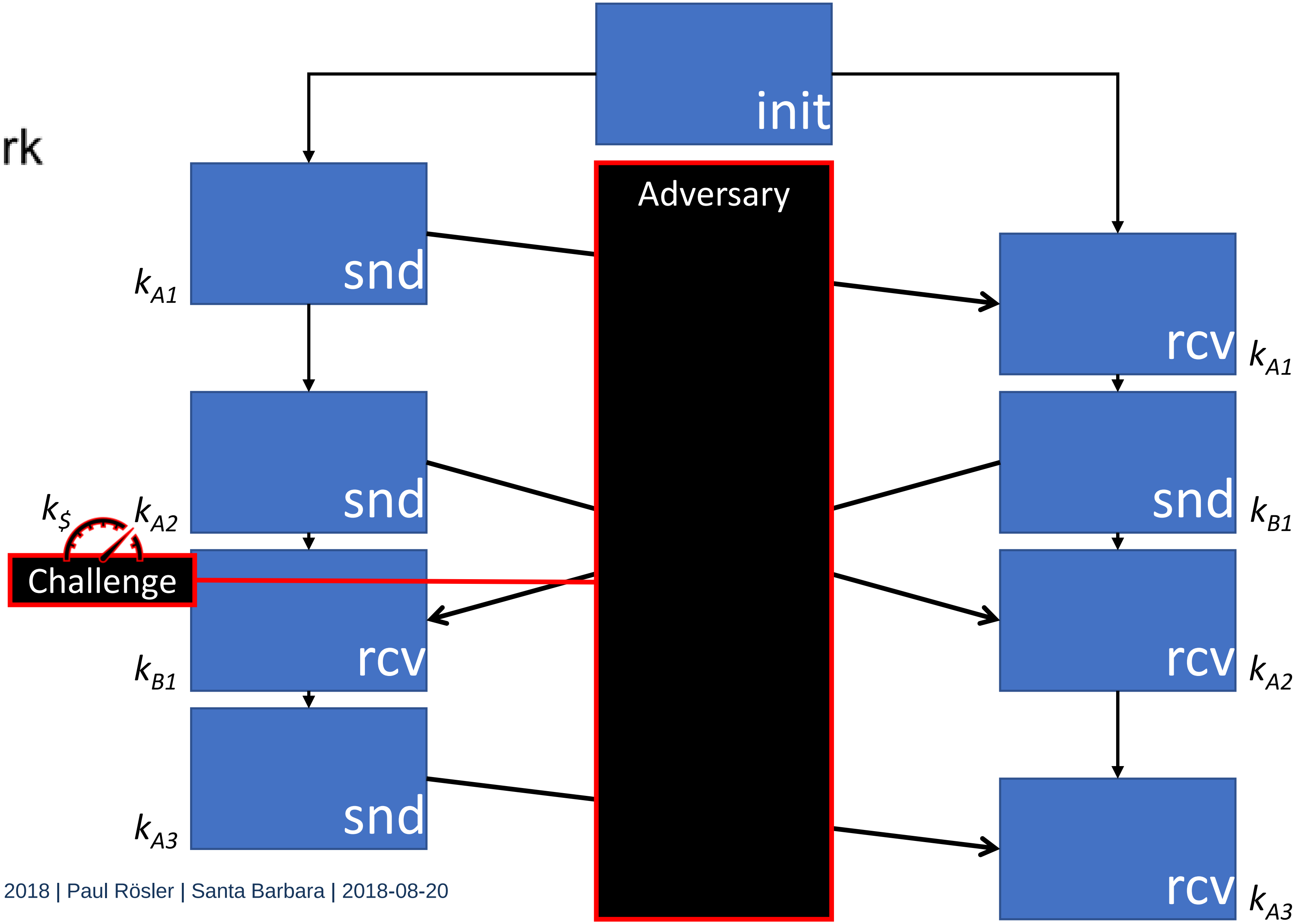
- Active adversary
  - Control whole network traffic



Modeling Ratcheted Key Exchange

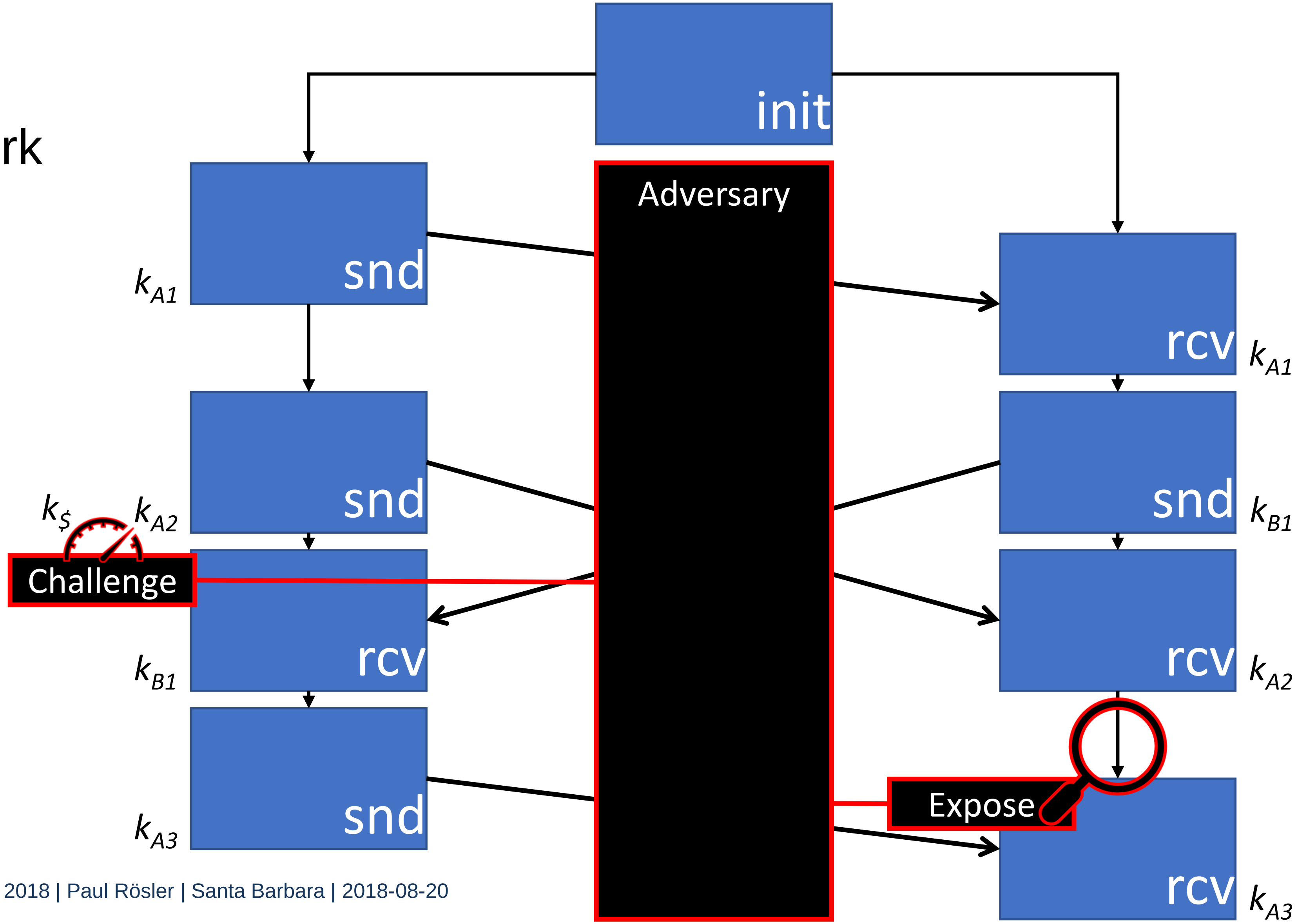
- Active adversary
  - Control whole network traffic
- Analyze key indistinguishability
  - Multi-challenge real or random key

  → Guess bit $b \in \{0,1\}$



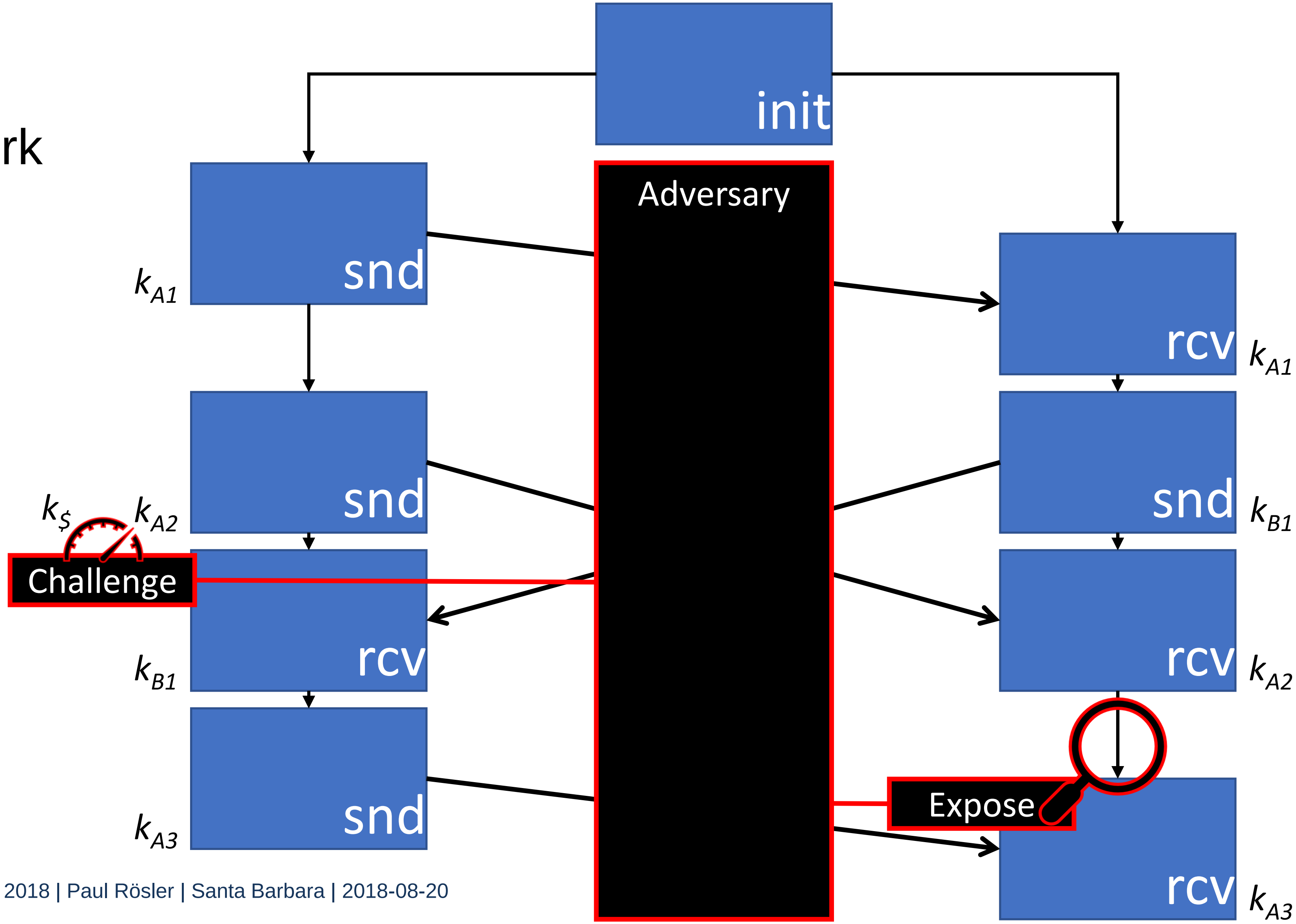
Modeling Ratcheted Key Exchange

- Active adversary
  - Control whole network traffic
- Analyze key indistinguishability
  - Multi-challenge real or random key
- Model exposures of local state



Modeling Ratcheted Key Exchange

- Active adversary
  - Control whole network traffic
- Analyze key indistinguishability
  - Multi-challenge real or random key
- Model exposures of local state
- Single session
- Init abstracted



Agenda

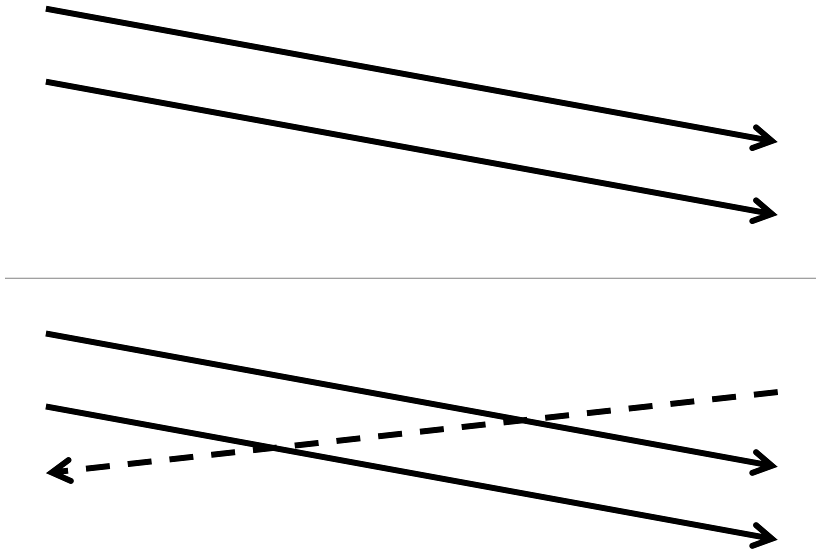
1. The Primitive Ratcheted Key Exchange

2. General Adversary Model

3. **Unidirectional Ratcheting**
   → **Model** and Construction

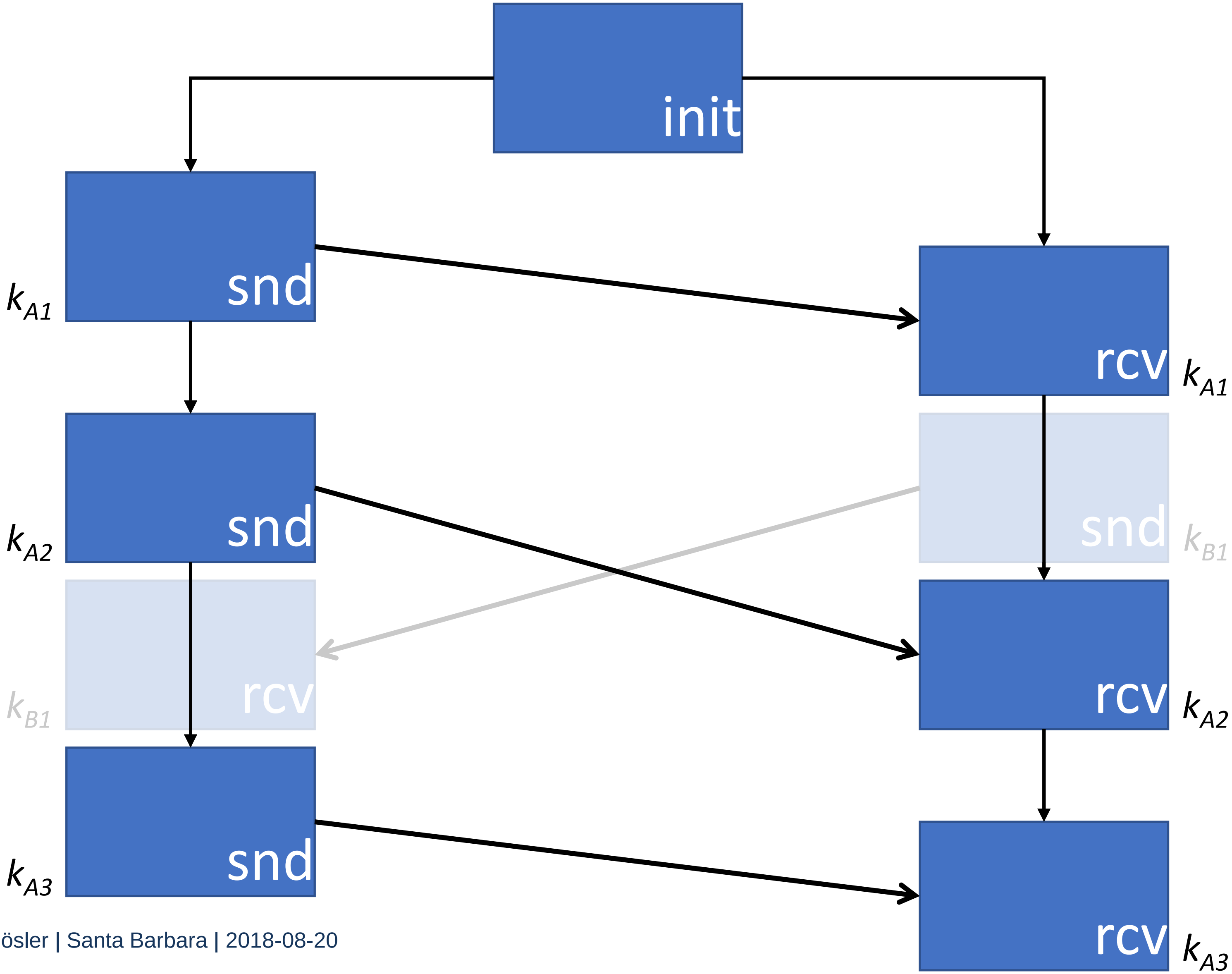
4. Sesquidirectional Ratcheting
   → Model and Construction

5. Results



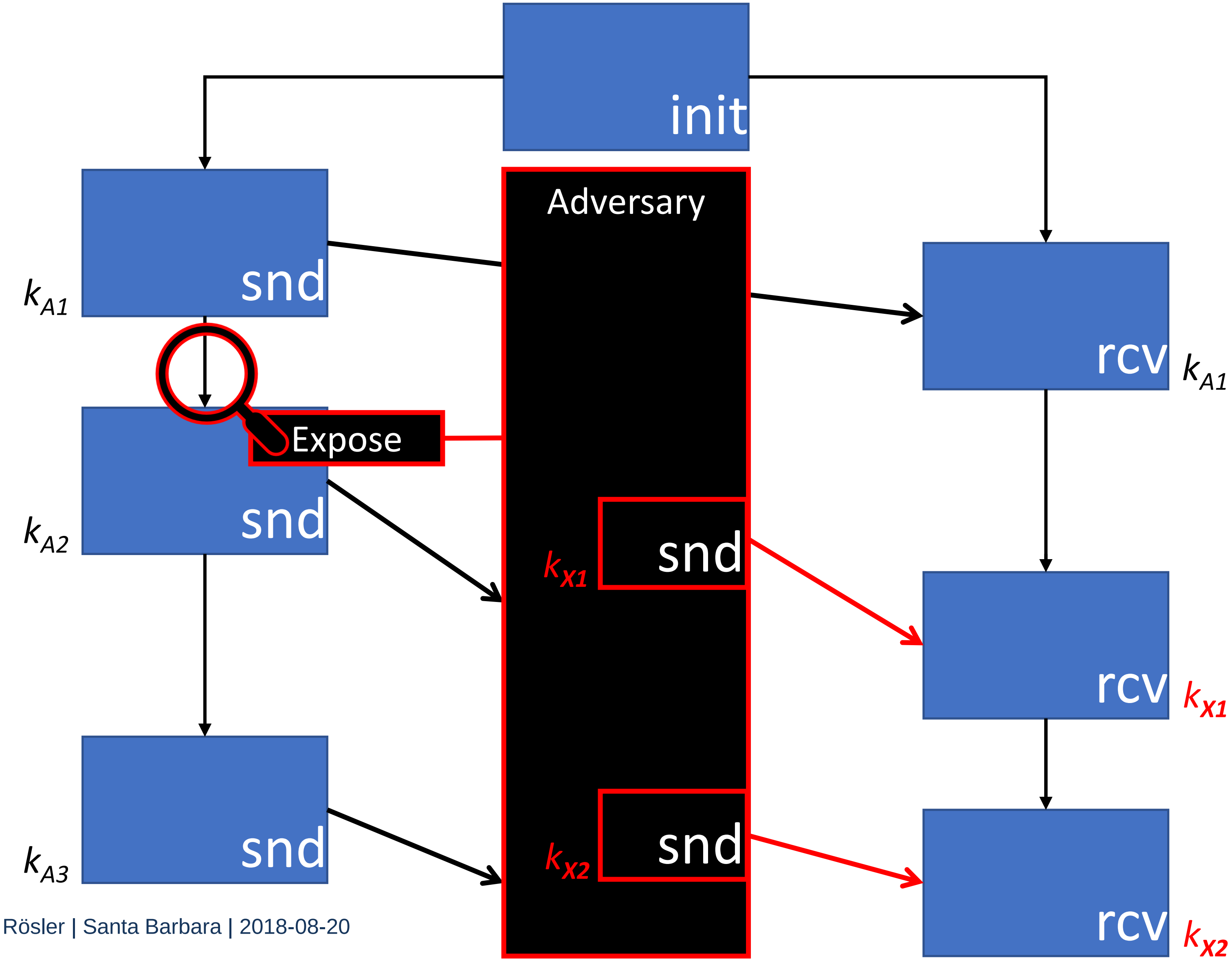
What is Ratcheting?
- Modeling RKE
Construction Intuition
Results

Modeling Unidirectional RKE



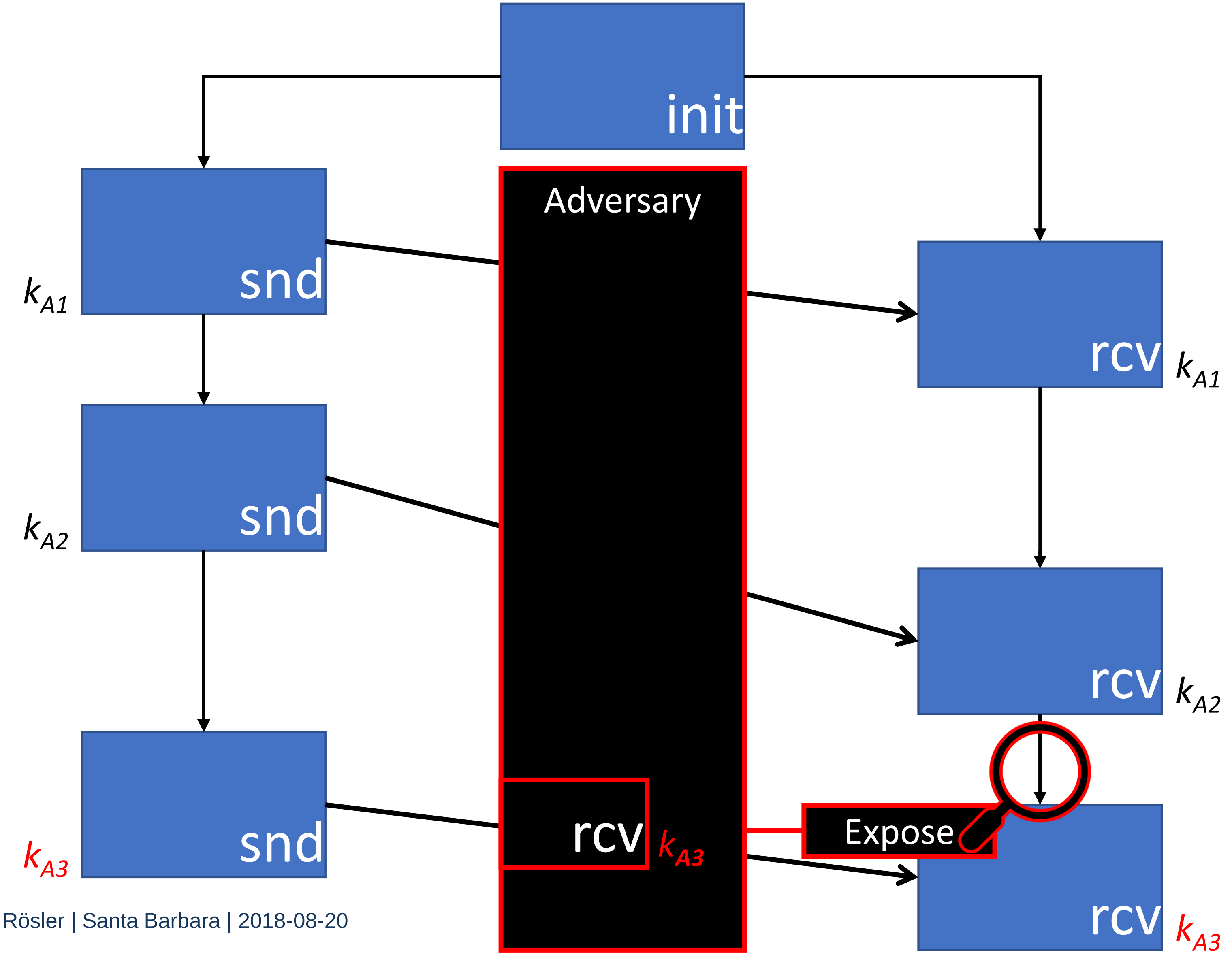
Modeling Unidirectional RKE

- Impersonation
  ⇒ No future Challenge on Bob



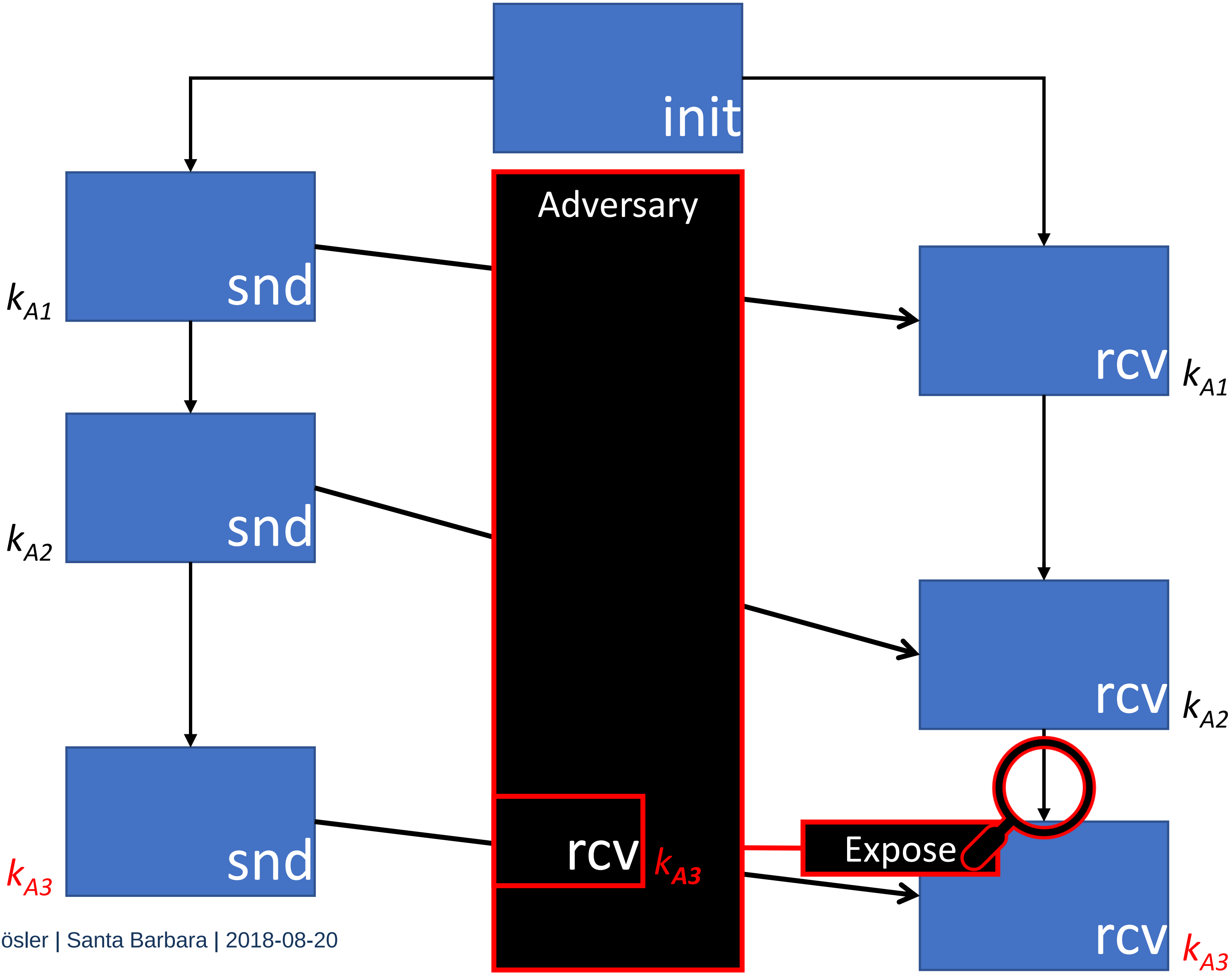
Modeling Unidirectional RKE

- Impersonation
  ⇒ No future Challenge on Bob
- Expose Bob
  → Allowed in our model



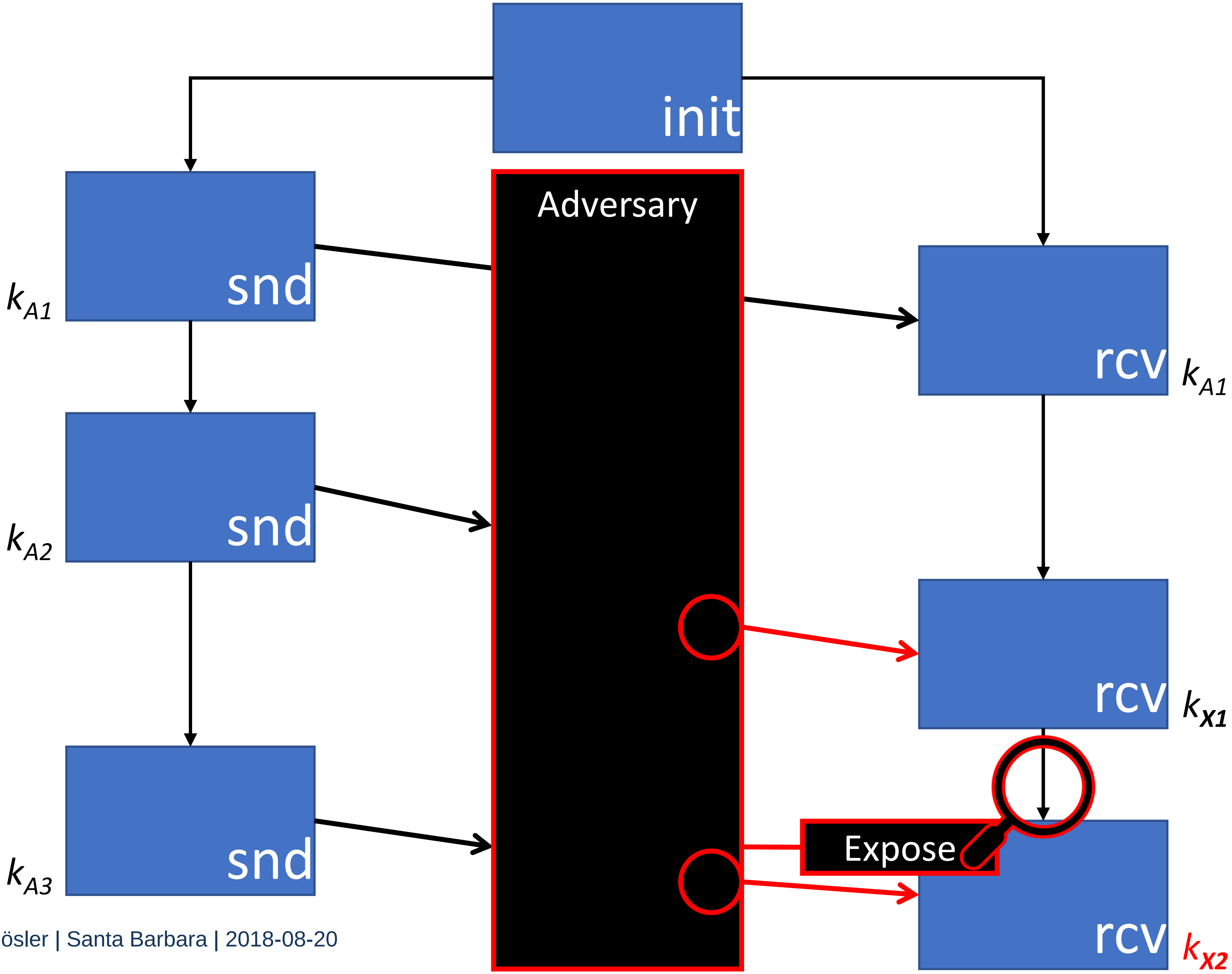
Modeling Unidirectional RKE

- Impersonation
  ⇒ No future Challenge on Bob
- Expose Bob
  ⇒ No future Challenge



Modeling Unidirectional RKE

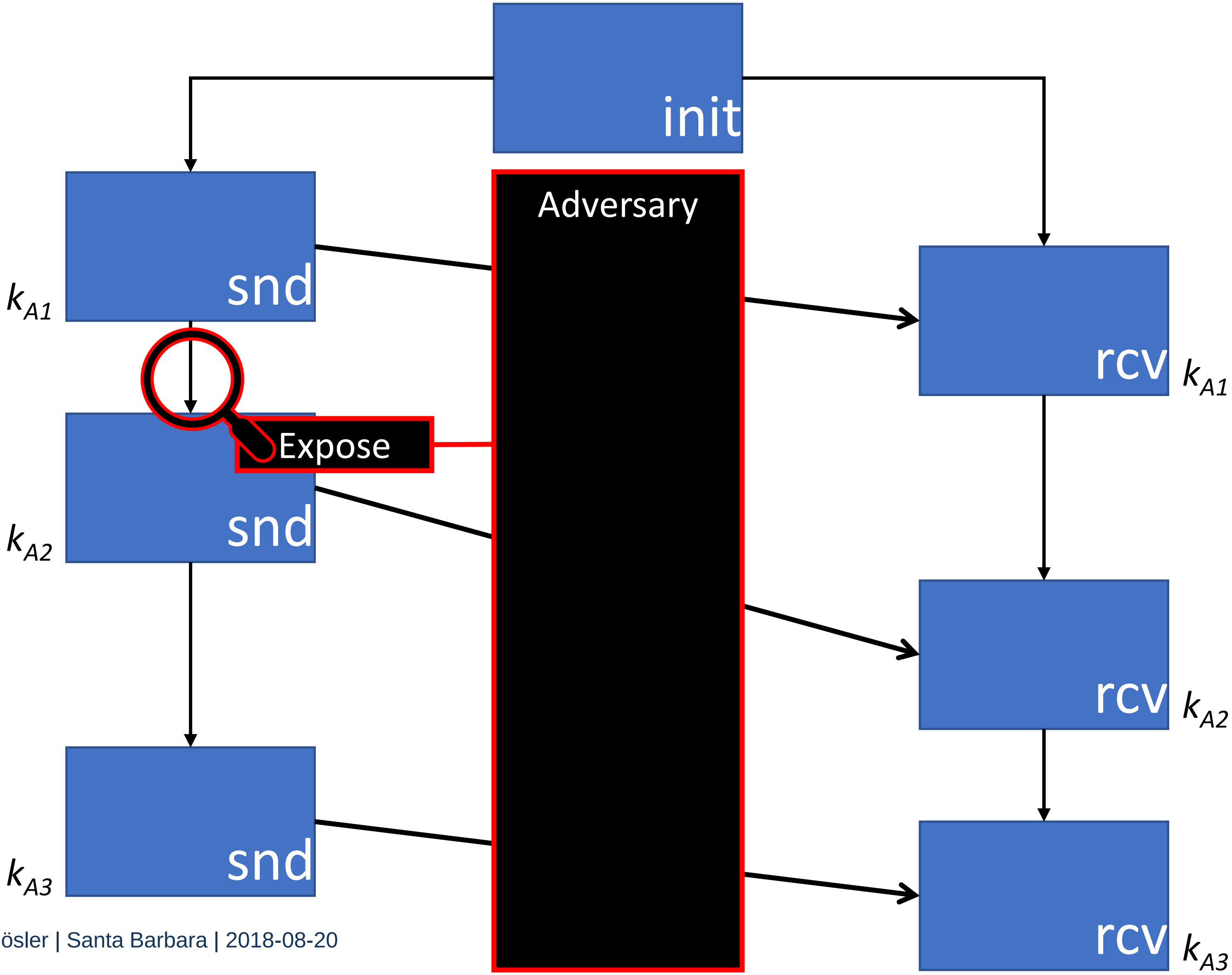
- Impersonation
  ⇒ No future Challenge on Bob
- Expose Bob
  ⇒ No future Challenge **if synchronous** (= if no previous active attack)



Modeling Unidirectional RKE

- Impersonation
  ⇒ No future Challenge on Bob
- Expose Bob
  ⇒ No future Challenge if synchronous

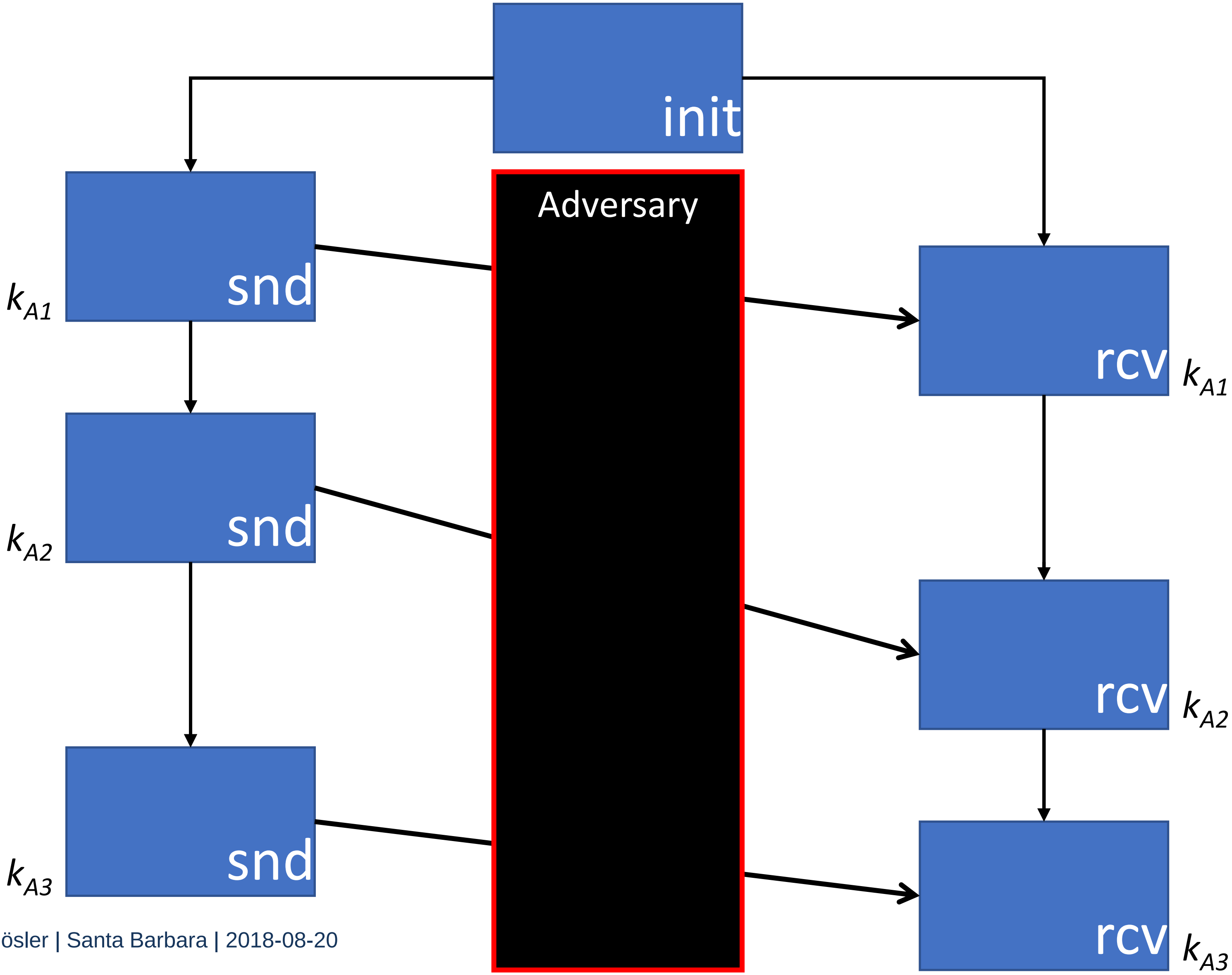
⇒ Exposure of Alice (solely) “okay”



Modeling Unidirectional RKE

- Impersonation
  ⇒ No future Challenge on Bob
- **Expose Bob**
  ⇒ No future Challenge if synchronous

⇒ Exposure of Alice (solely) “okay”



Agenda

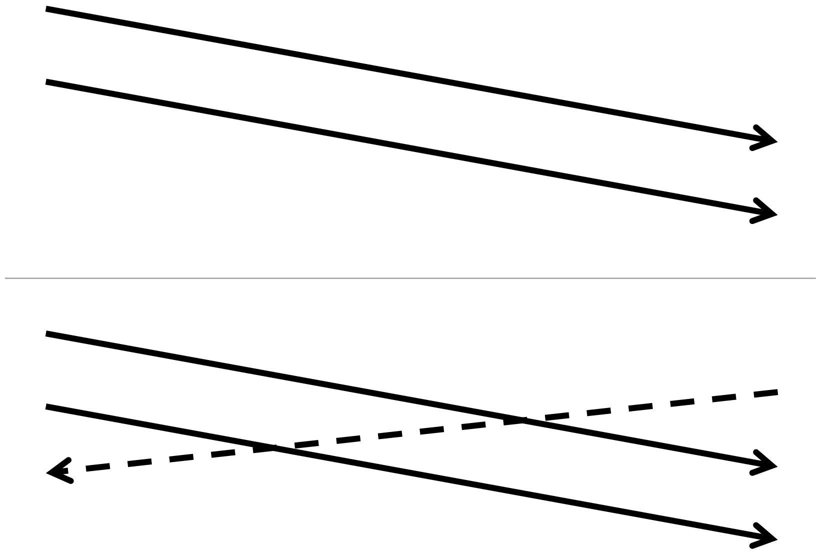
1. The Primitive Ratcheted Key Exchange

2. General Adversary Model

3. **Unidirectional Ratcheting**
   → Model and **Construction**

4. Sesquidirectional Ratcheting
   → Model and Construction

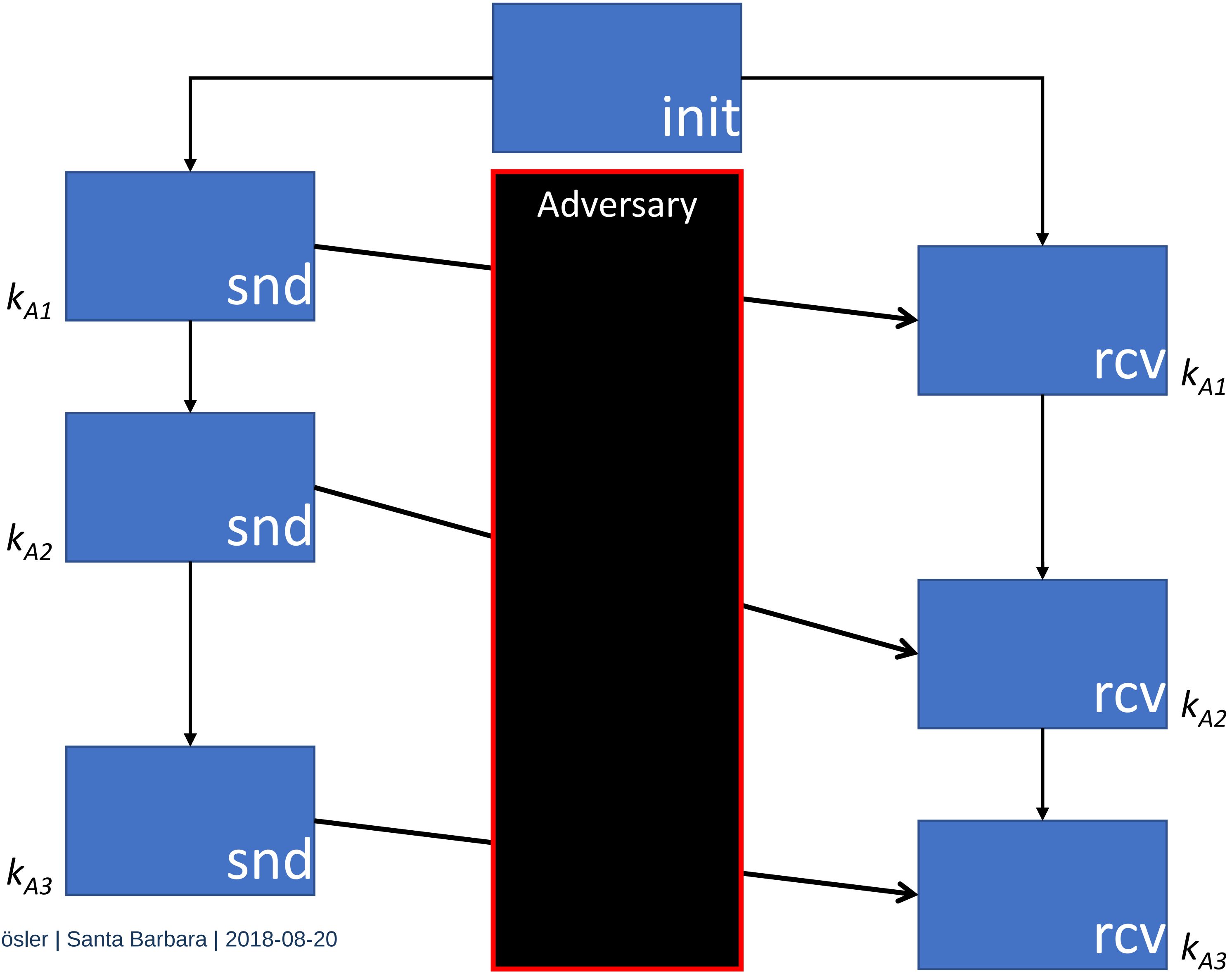
5. Results



Constructing Unidirectional RKE

- **Expose Alice okay**

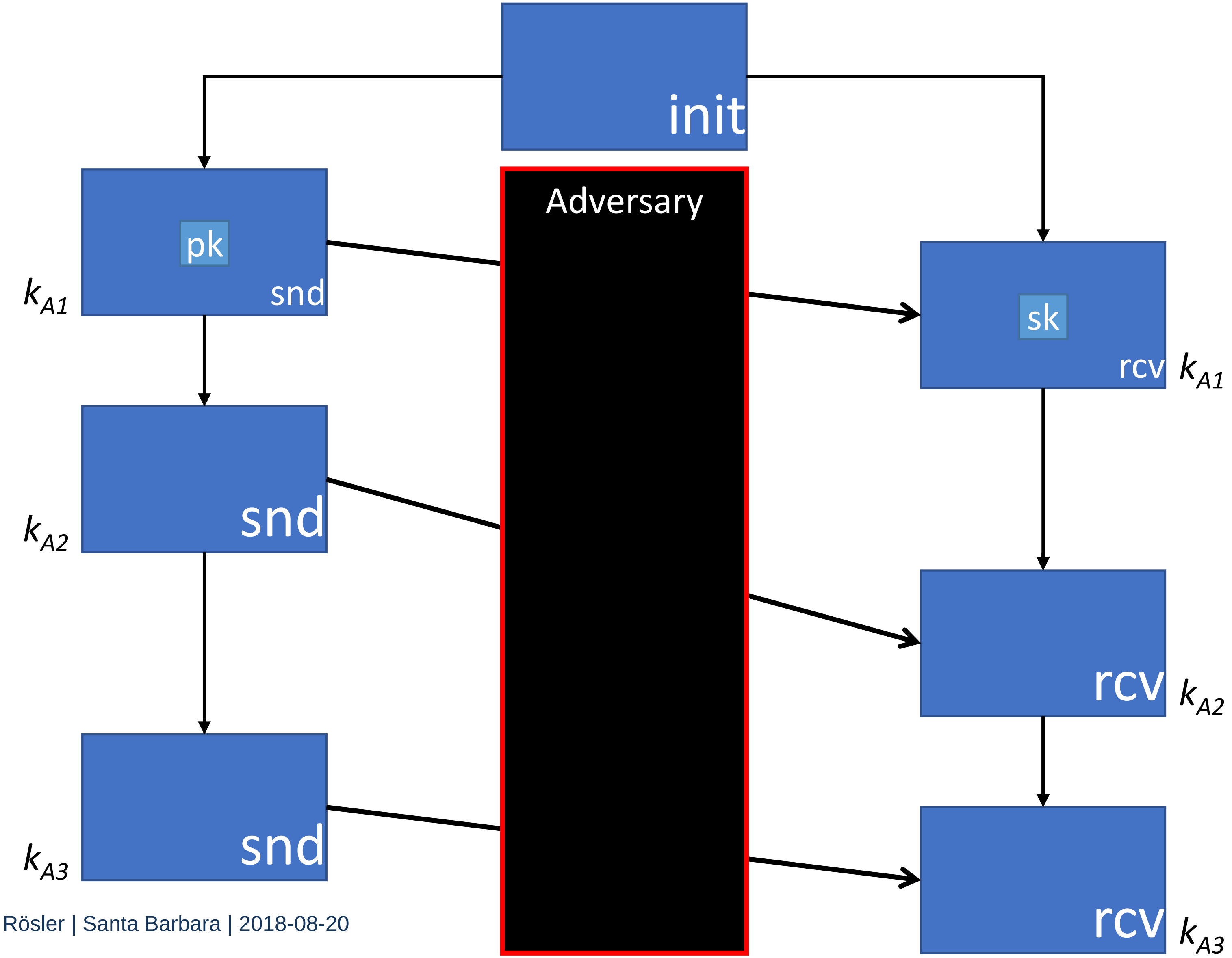
- **Expose Bob**
  ⇒ No future Challenge if synchronous



Constructing Unidirectional RKE

- **Expose Alice okay**
  → Public key crypto

- **Expose Bob**
  ⇒ No future Challenge if synchronous



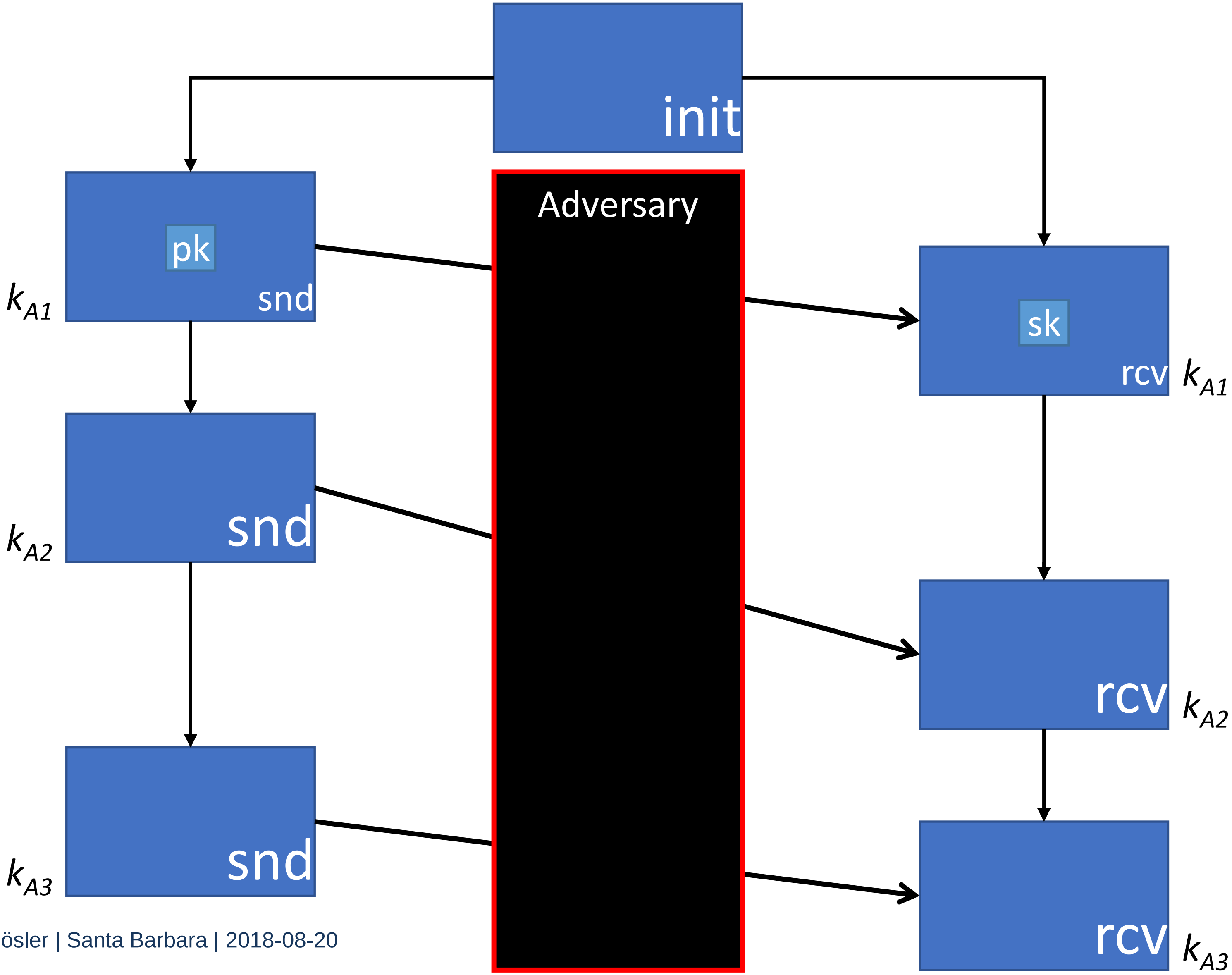
RUHR UNIVERSITÄT BOCHUM RUB

Constructing Unidirectional RKE

- **Expose Alice okay**
  → KEM:

enc (pk) →$_\$$ c k   dec (sk c) →$_\$$ k

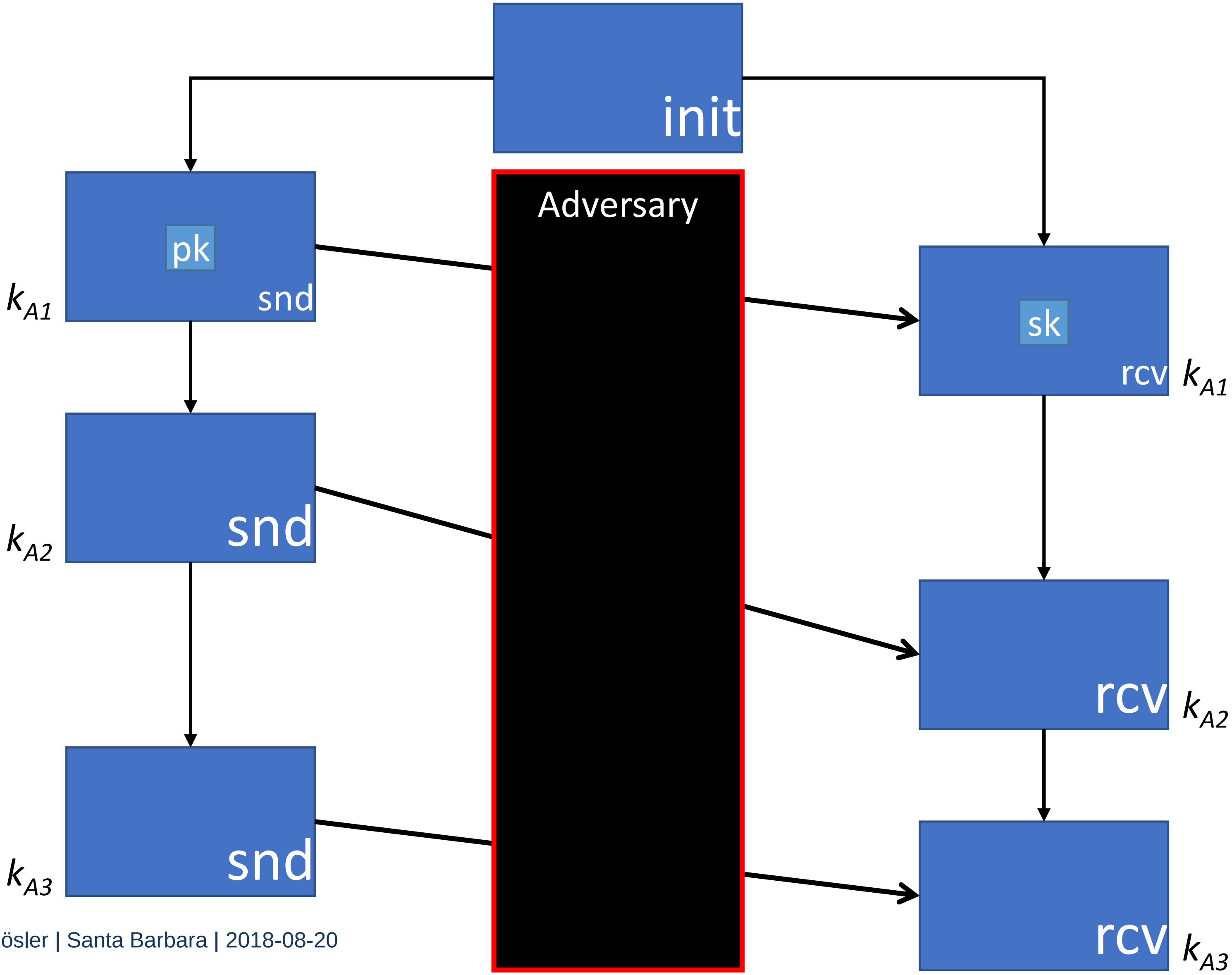
- **Expose Bob**
  ⇒ No future Challenge if synchronous



Constructing Unidirectional RKE

- **Expose Alice okay**
  → KEM:
  enc(pk) →$_\$$ c k dec(sk c) →$_\$$ k

- **Expose Bob**
  ⇒ No future Challenge if synchronous

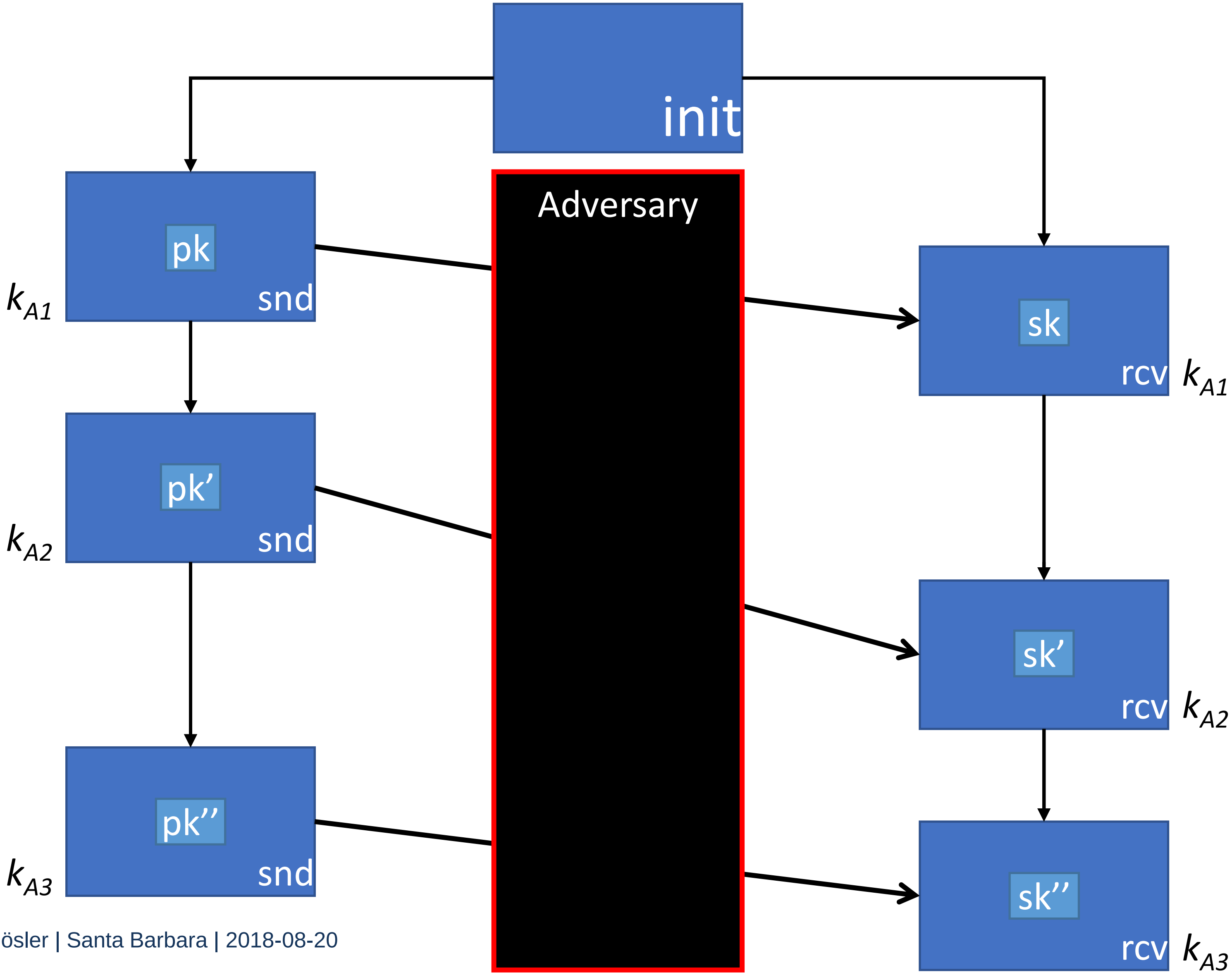


Constructing Unidirectional RKE

- **Expose Alice okay**
  → KEM:

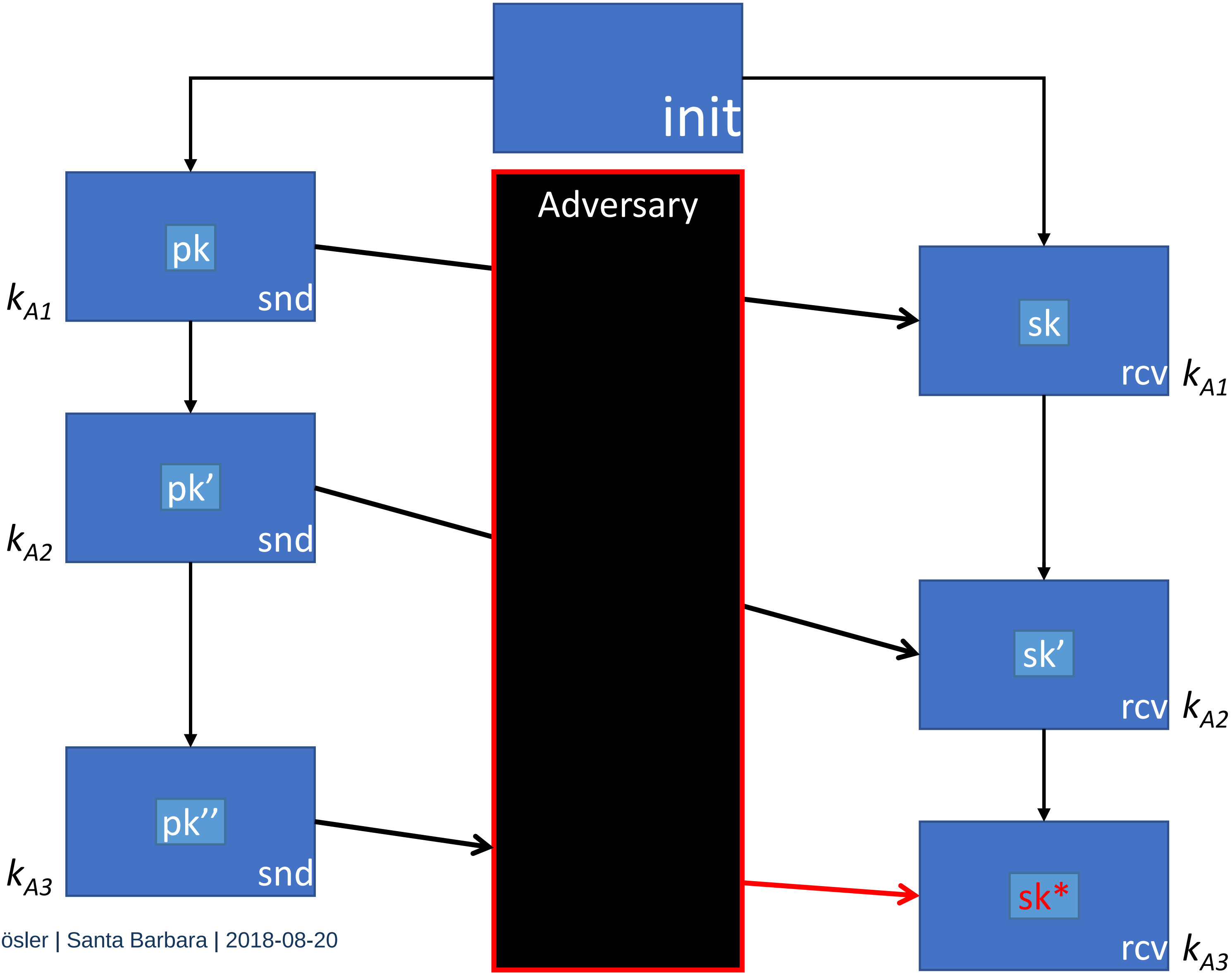
  enc (pk) →$_\$$ c k   dec (sk c) →$_\$$ k

- **Expose Bob**
  ⇒ No future Challenge if synchronous
  → Forward secrecy of Bob’s state



Constructing Unidirectional RKE

- **Expose Alice okay**
  → KEM:
  enc(pk)→$_\$$ c k   dec(sk c)→$_\$$ k
- **Expose Bob**
  ⇒ No future Challenge if synchronous
  → Forward secrecy of Bob's state
  → Divergence of states



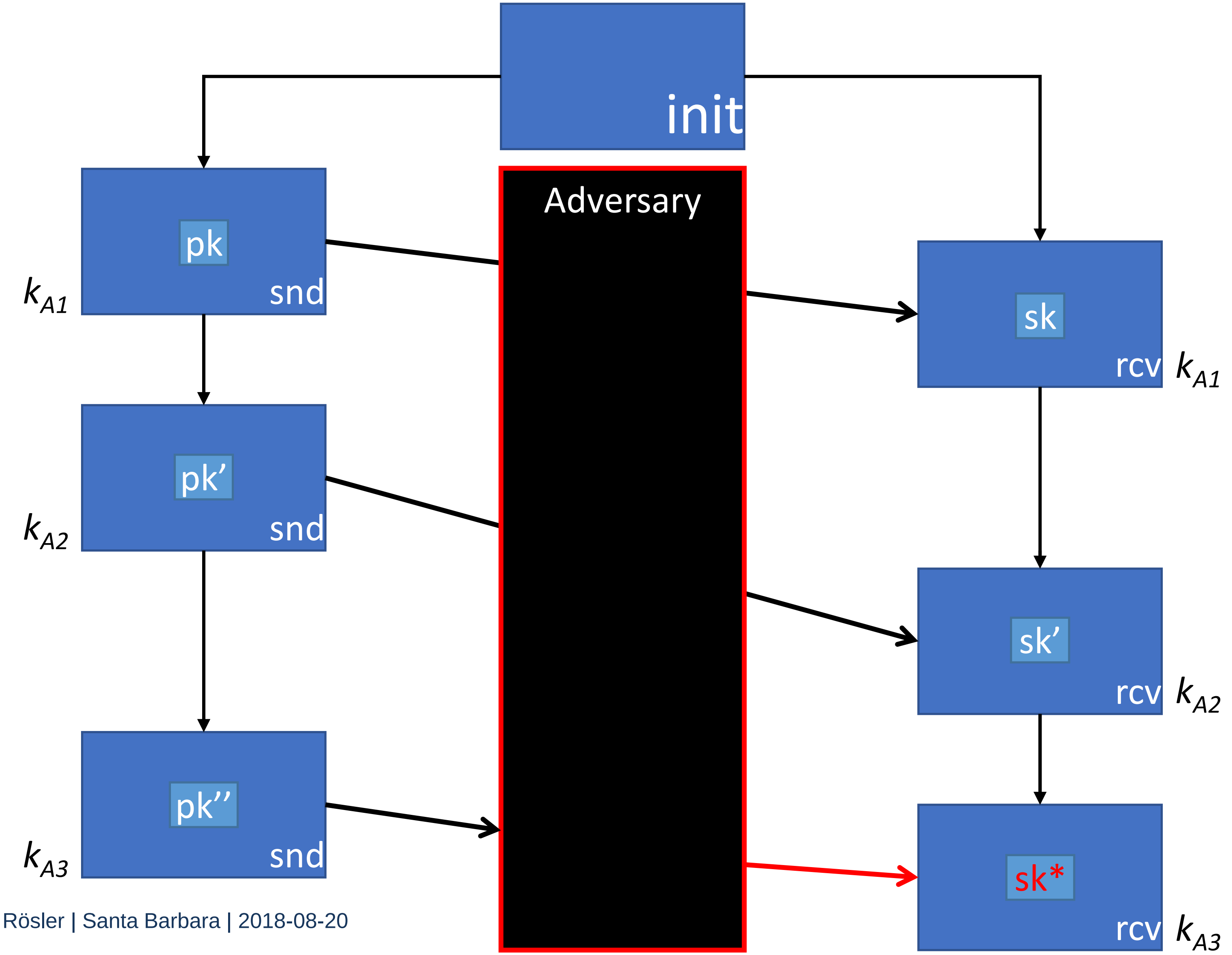
Constructing Unidirectional RKE

- Expose Alice okay
  → KEM:

  $\mathrm{enc}(pk) \rightarrow_{\$} c\ k$ $\quad$ $\mathrm{dec}(sk\ c) \rightarrow_{\$} k$

- **Expose Bob**
  ⇒ No future Challenge if synchronous
  → Forward secrecy of Bob's state
  → Divergence of states
  → Random oracle:

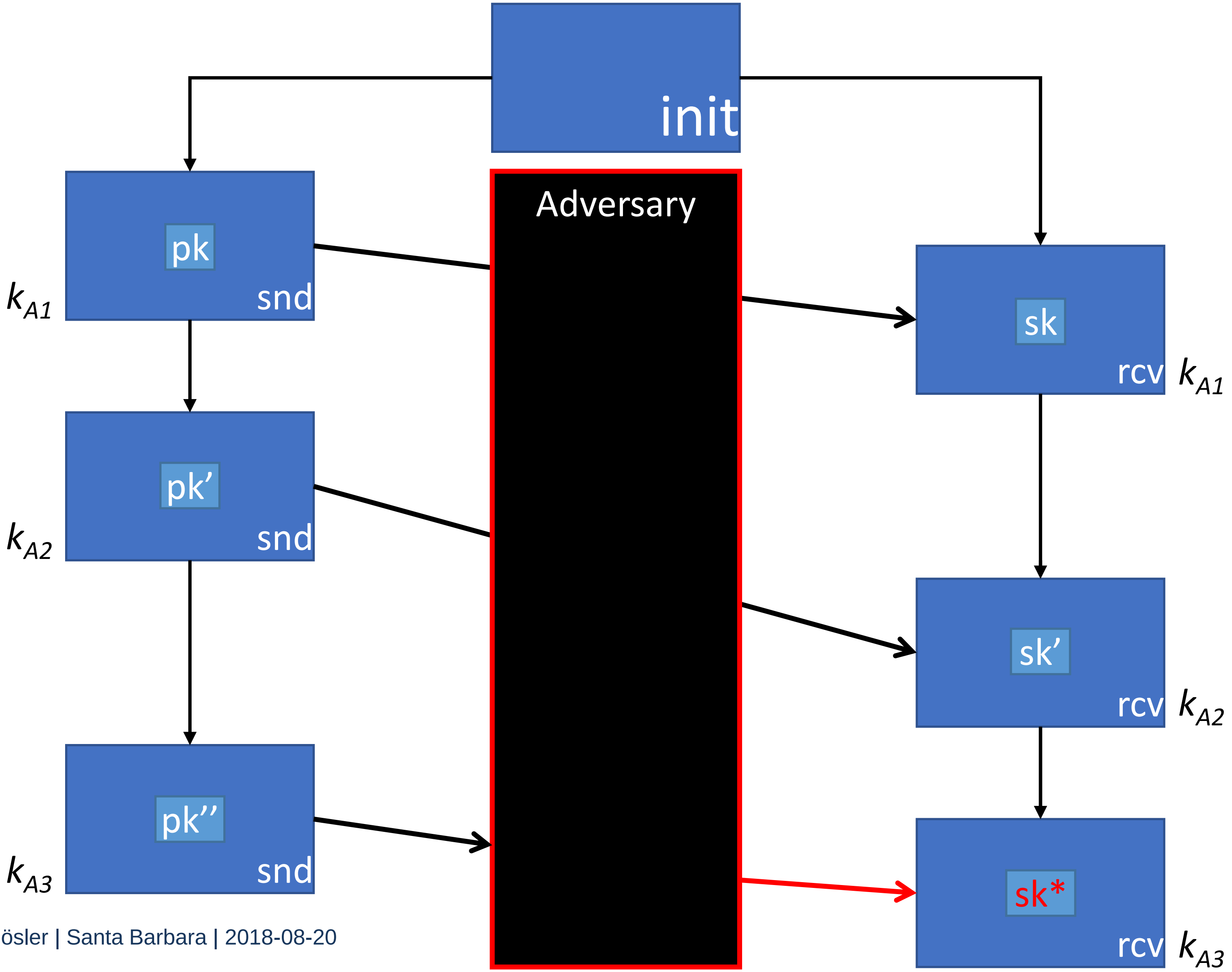
  $H(c\ k) \rightarrow k_{Xn}\ sk$



Constructing Unidirectional RKE

- **Expose Alice okay**
  → KEM:
  enc(pk)→$_\$$ c k dec(sk c)→$_\$$ k

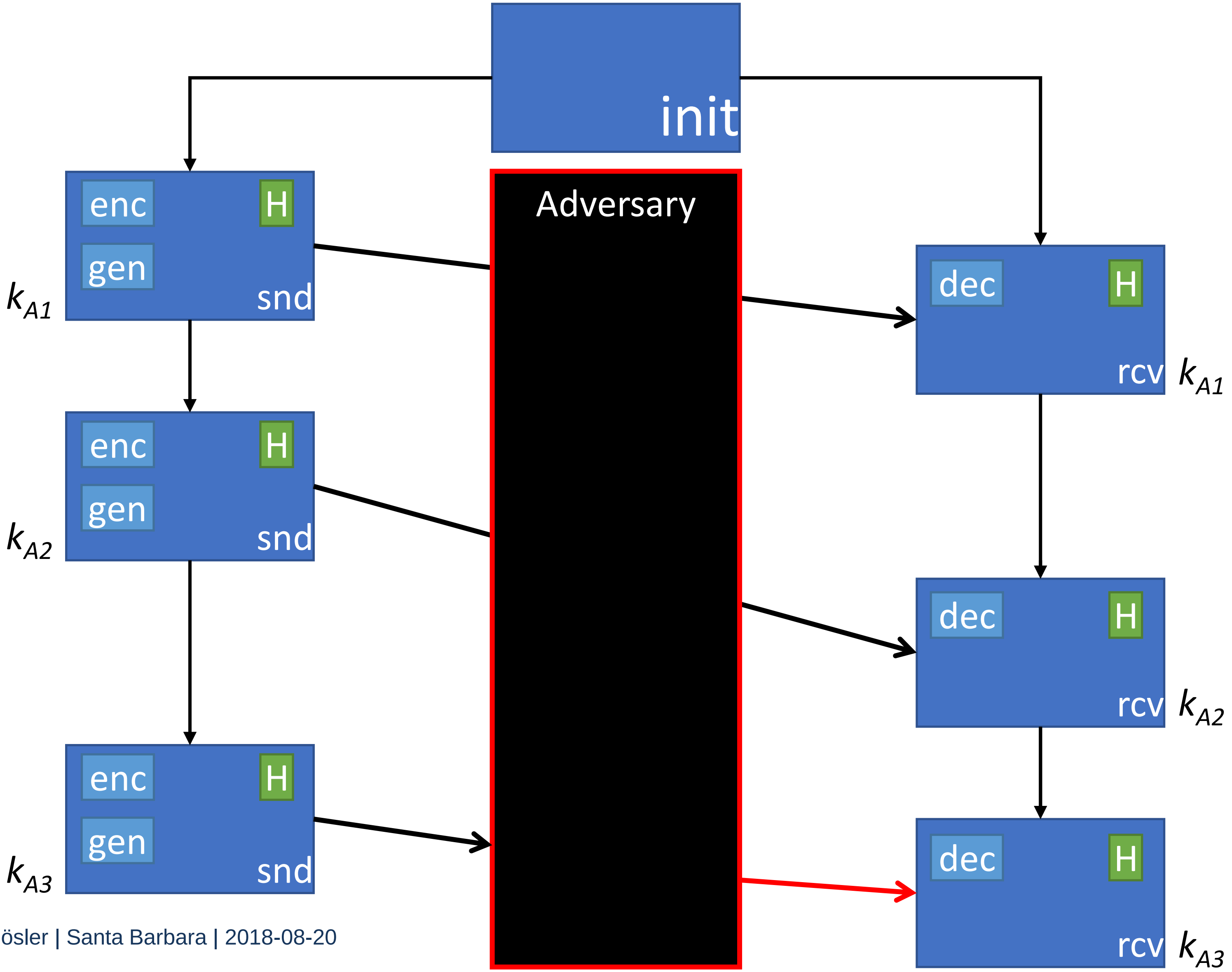
- **Expose Bob**
  ⇒ No future Challenge if synchronous
  → Forward secrecy of Bob's state
  → Divergence of states
  → Random oracle:
  H(c k)→ k_{Xn} sk
  gen(sk)→pk



Constructing Unidirectional RKE

- **Expose Alice okay**
  → KEM:
  enc(pk)$\rightarrow_{\$}$ c k dec(sk c)$\rightarrow_{\$}$ k

- **Expose Bob**
  ⇒ No future Challenge if synchronous
  → Forward secrecy of Bob's state
  → Divergence of states
  → Random oracle:
  H(c k)→ k_{Xn} sk
  gen(sk)→pk



Agenda

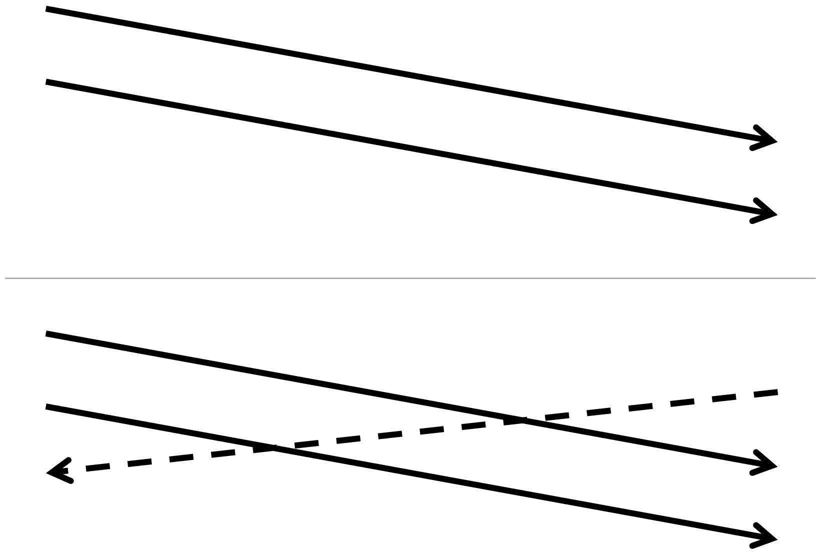
1. The Primitive Ratcheted Key Exchange

2. General Adversary Model

3. Unidirectional Ratcheting
   → Model and Construction

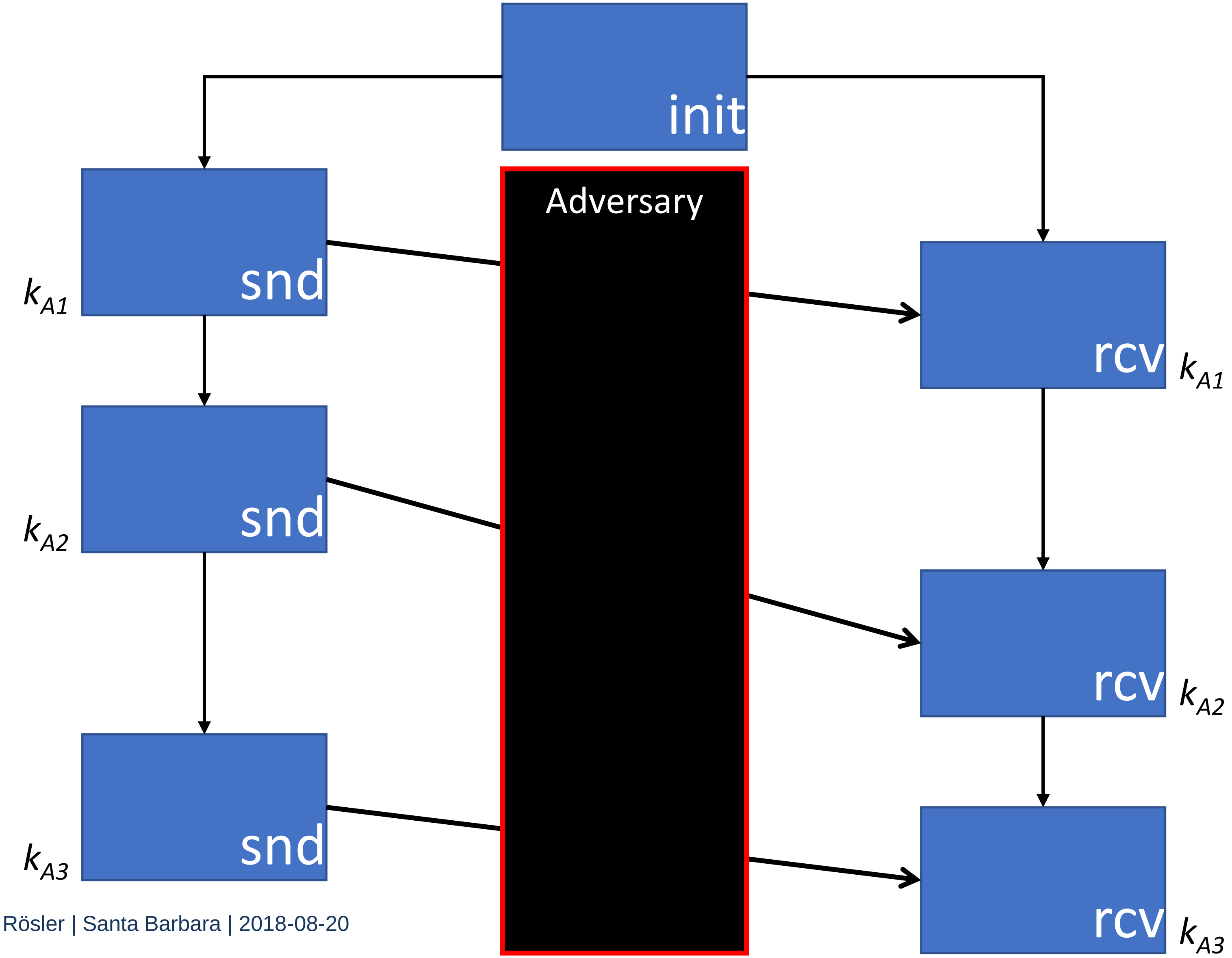
4. **Sesquidirectional Ratcheting**
   → **Model** and Construction

5. Results



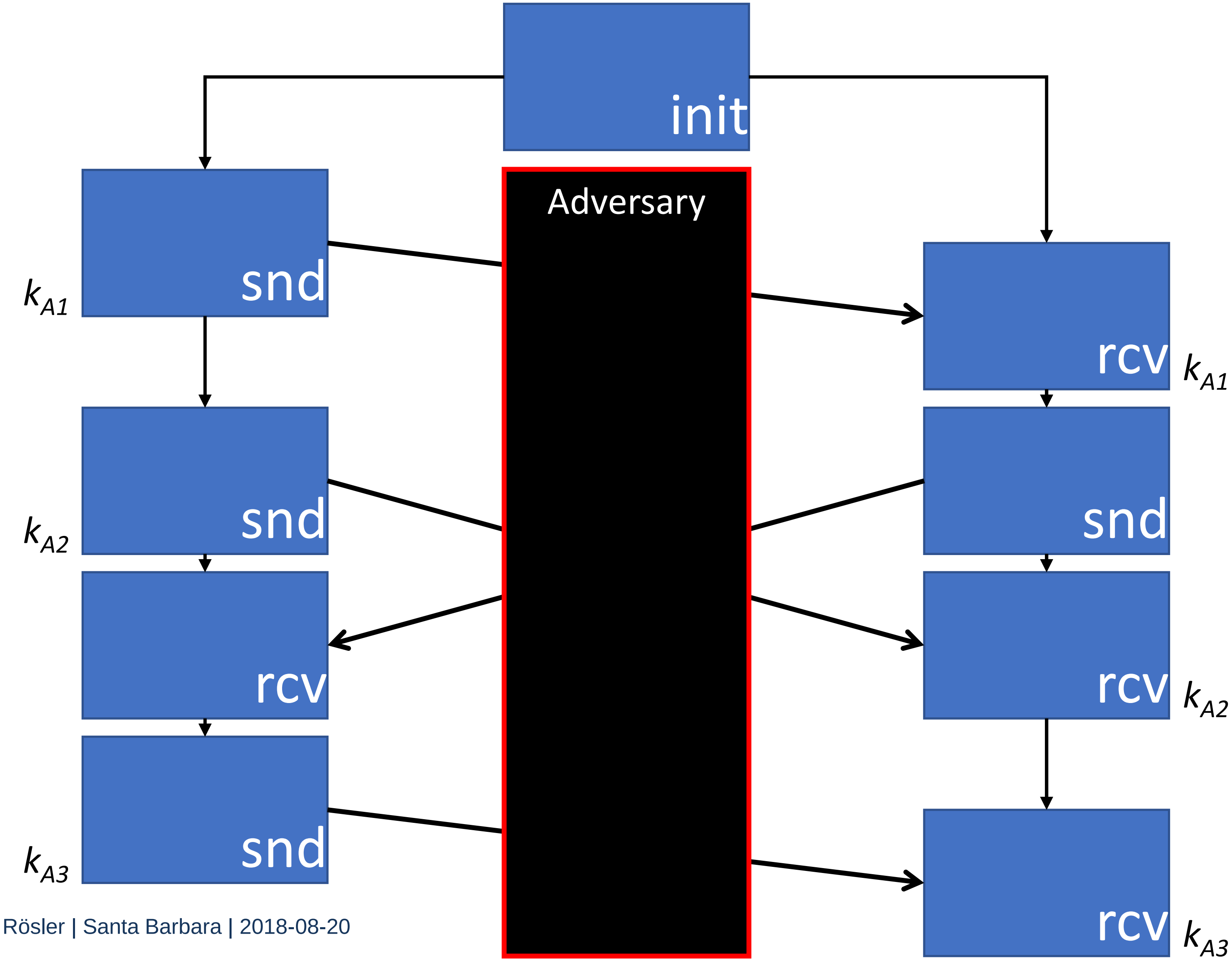
Modeling Unidirectional RKE

- Impersonation A → B
  ⇒ No future Challenge on Bob
- Expose Bob
  ⇒ No future Challenge if synchronous



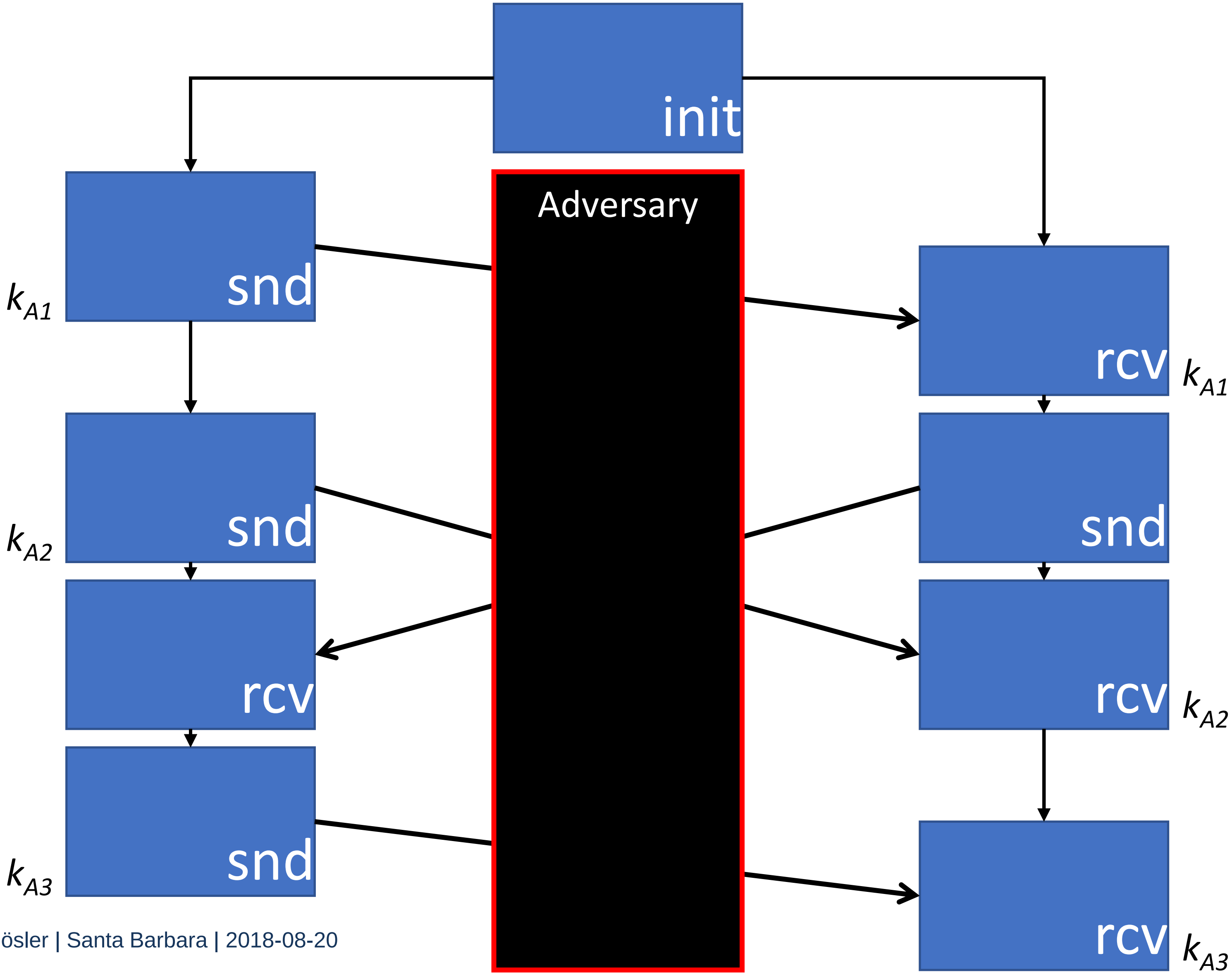
Modeling Sesquidirectional RKE

- Impersonation A → B
  ⇒ No future Challenge on Bob
- Expose Bob
  ⇒ No future Challenge if synchronous



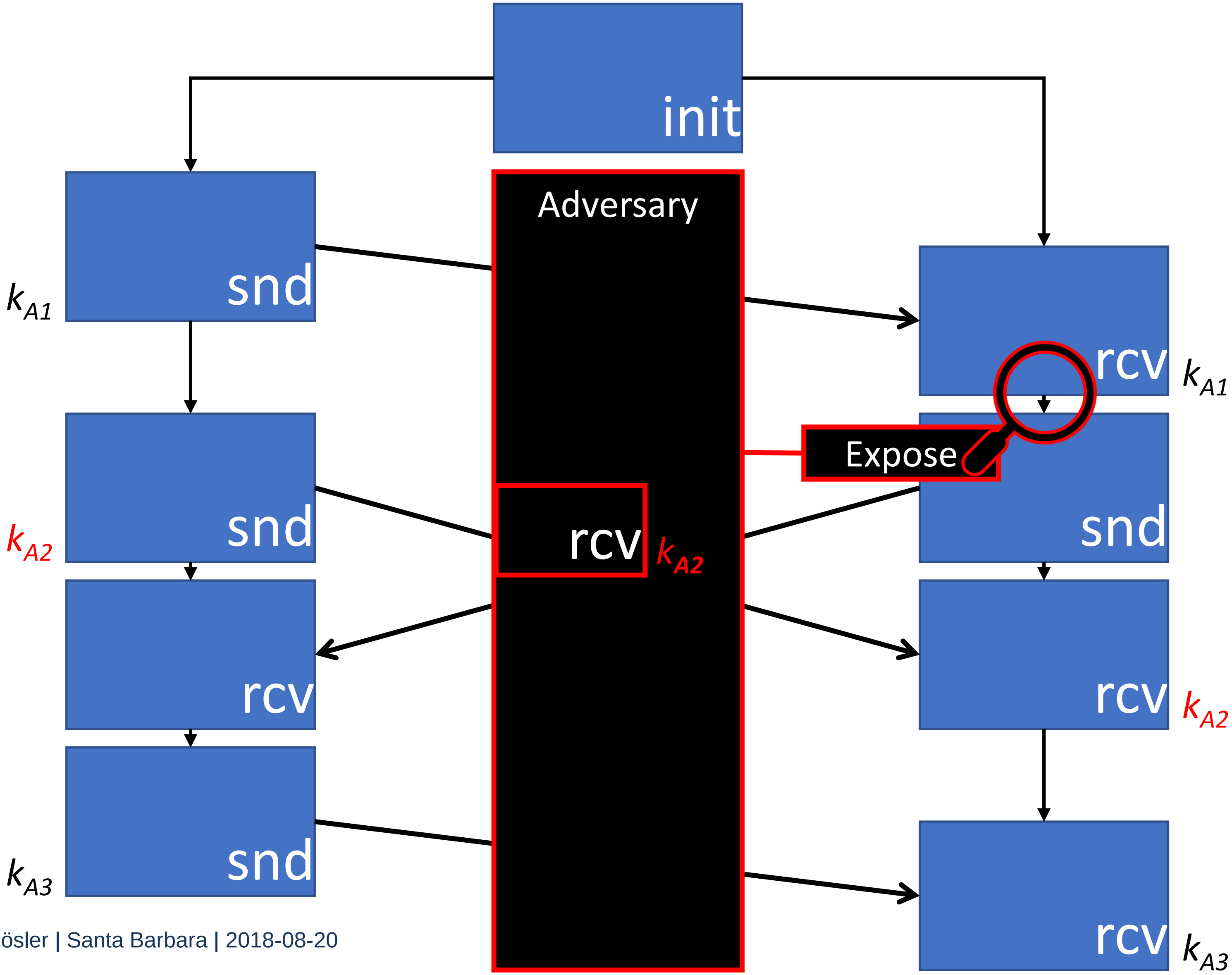
Modeling Sesquidirectional RKE

- Impersonation A → B
  ⇒ No future Challenge on Bob
- Impersonation B → A
  ⇒ No future Challenge on Alice
- Expose Bob
  ⇒ No future Challenge if synchronous



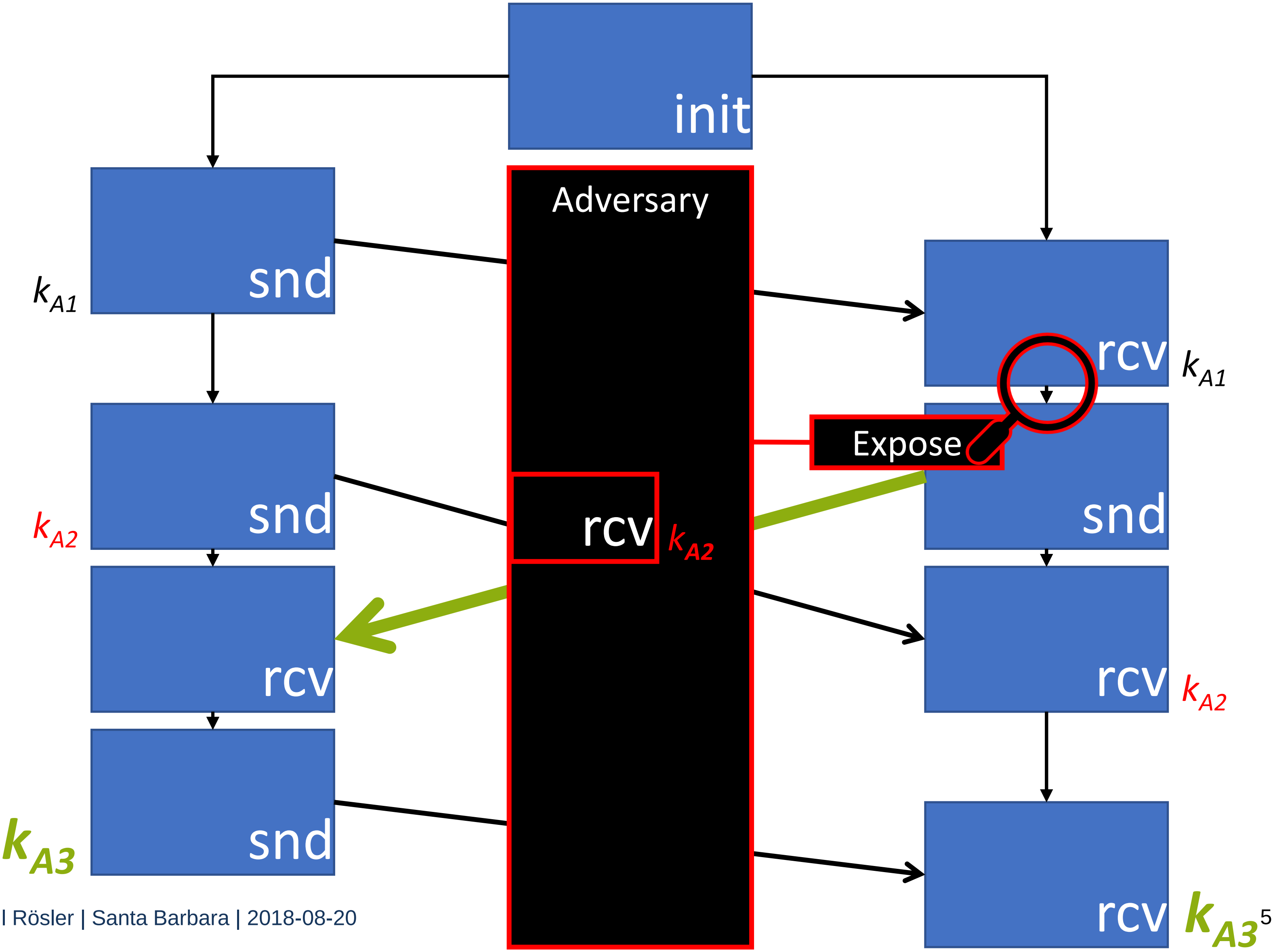
Modeling Sesquidirectional RKE

- Impersonation A → B
  ⇒ No future Challenge on Bob
- Impersonation B → A
  ⇒ No future Challenge on Alice
- Expose Bob
  ⇒ No future Challenge if synchronous **until Bob recovered**



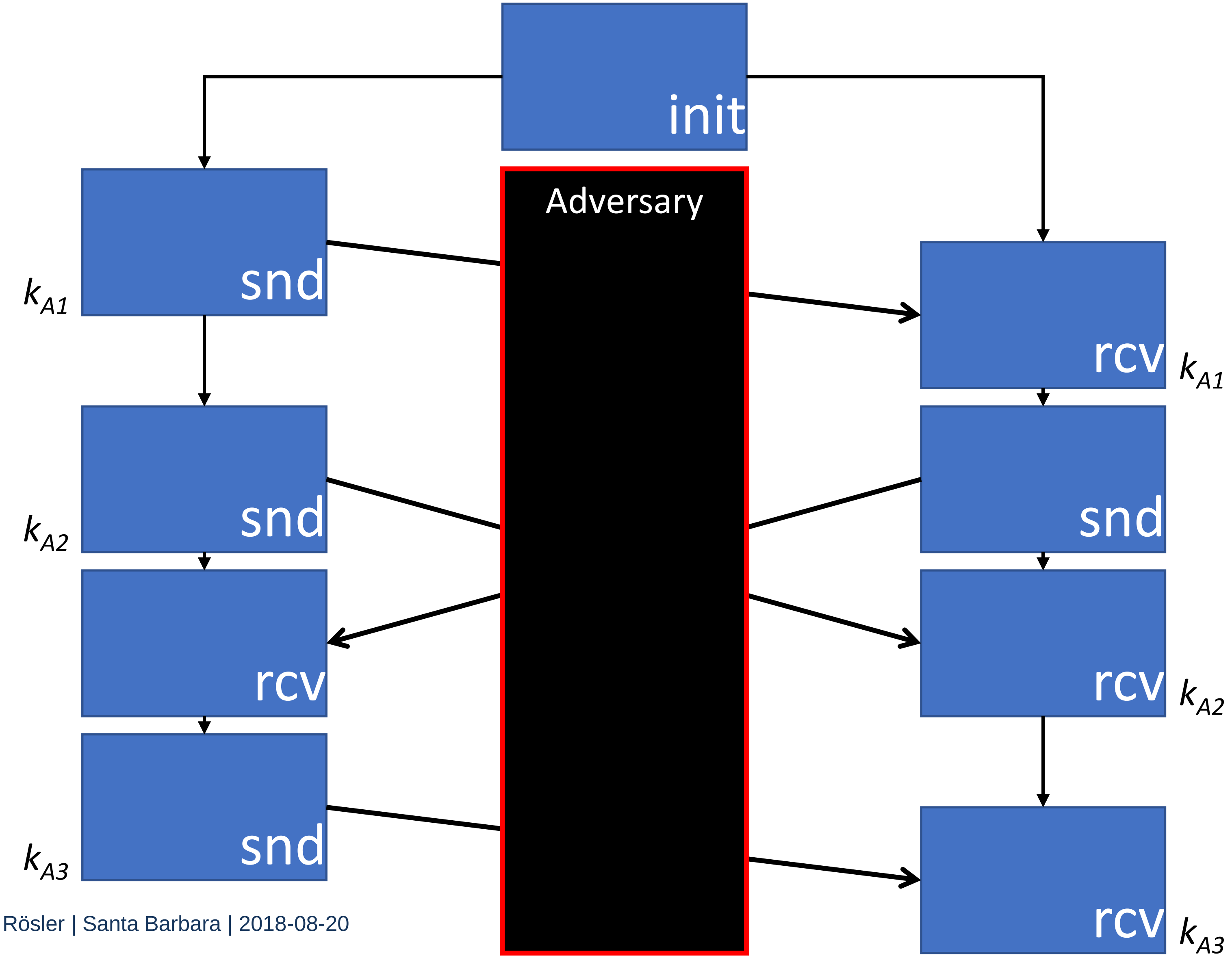
Modeling Sesquidirectional RKE

- Impersonation A → B
  ⇒ No future Challenge on Bob
- Impersonation B → A
  ⇒ No future Challenge on Alice
- Expose Bob
  ⇒ No future Challenge if synchronous **until Bob recovered**



Modeling Sesquidirectional RKE

- Impersonation A → B
  ⇒ No future Challenge on Bob
- Impersonation B → A
  ⇒ No future Challenge on Alice
- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered



Agenda

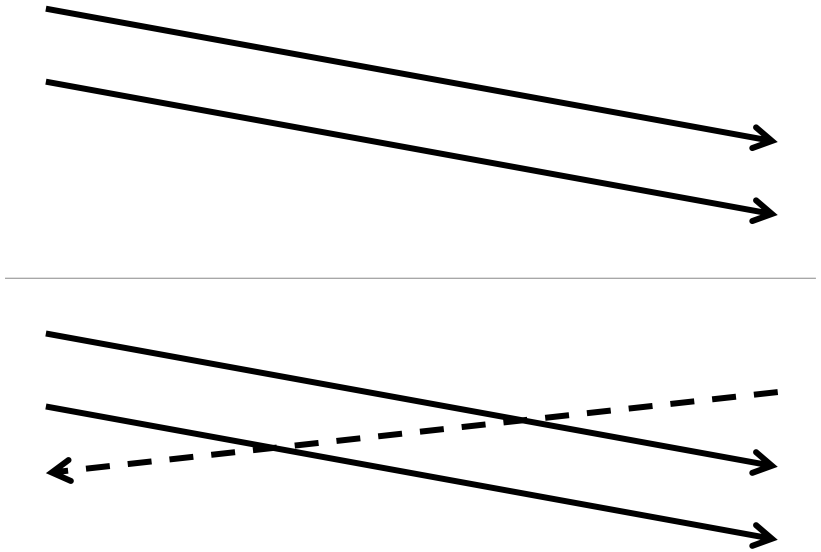
1. The Primitive Ratcheted Key Exchange

2. General Adversary Model

3. Unidirectional Ratcheting
   → Model and Construction

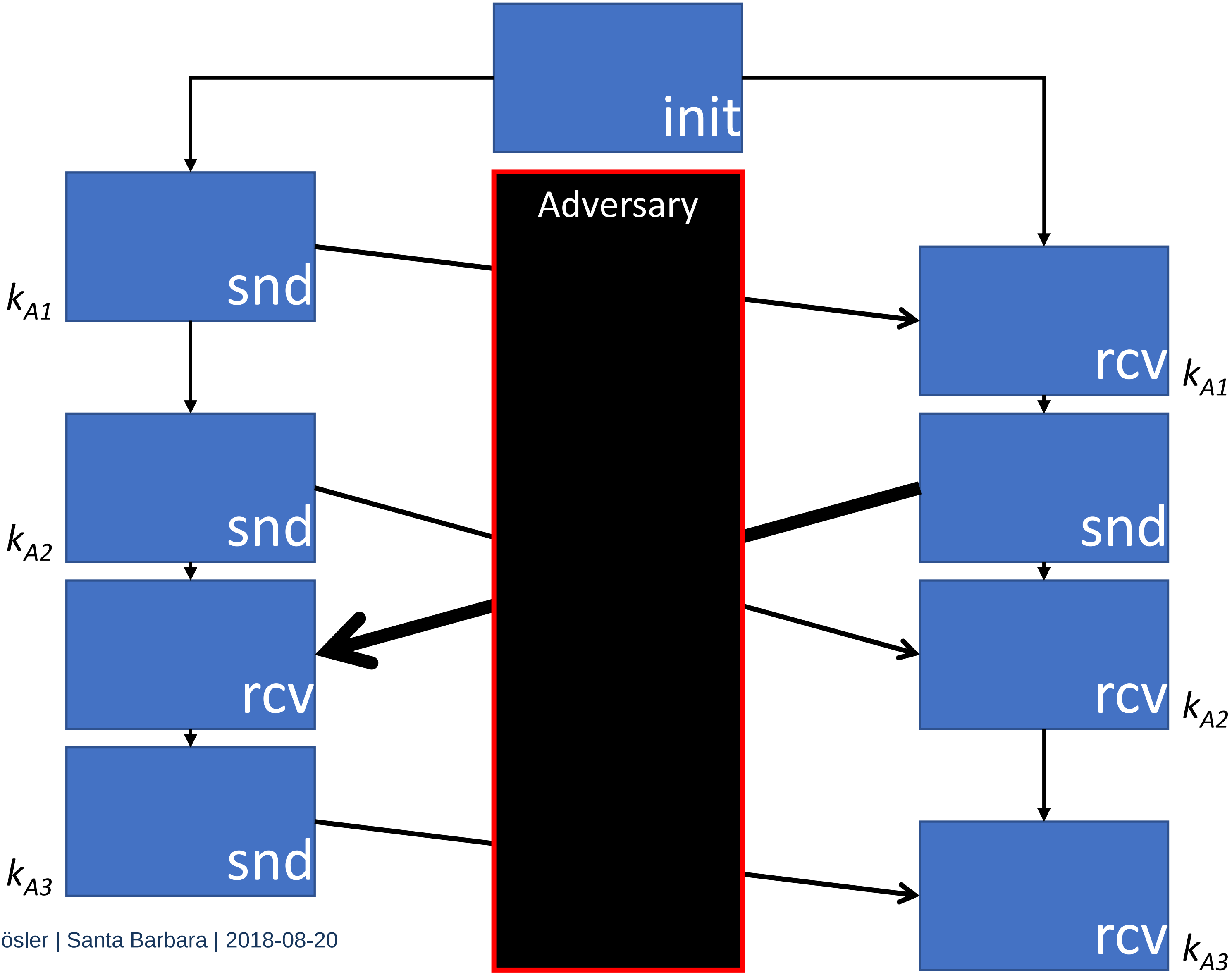
4. **Sesquidirectional Ratcheting**
   → Model and **Construction**

5. Results



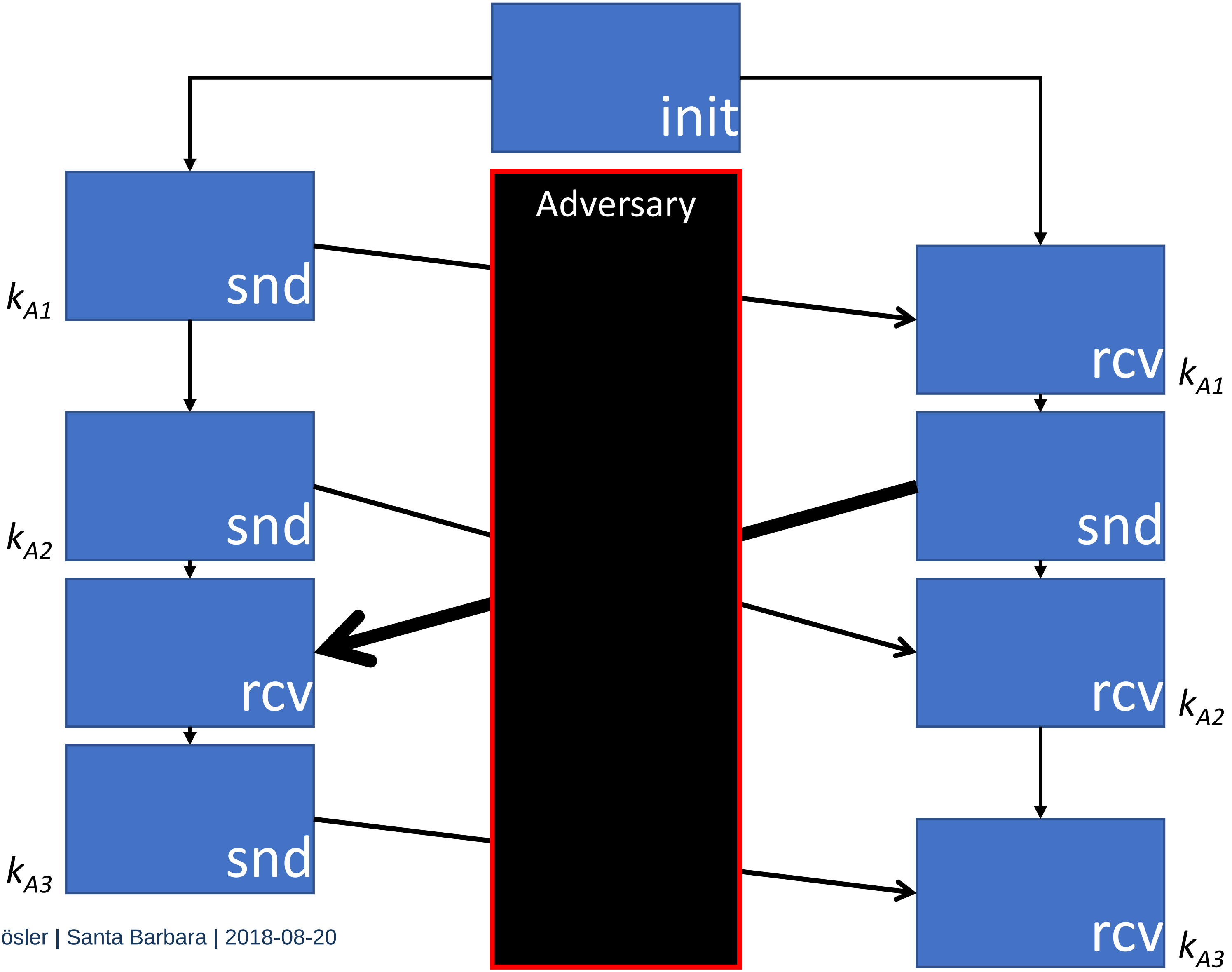
Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge
  if synchronous
  until Bob recovered



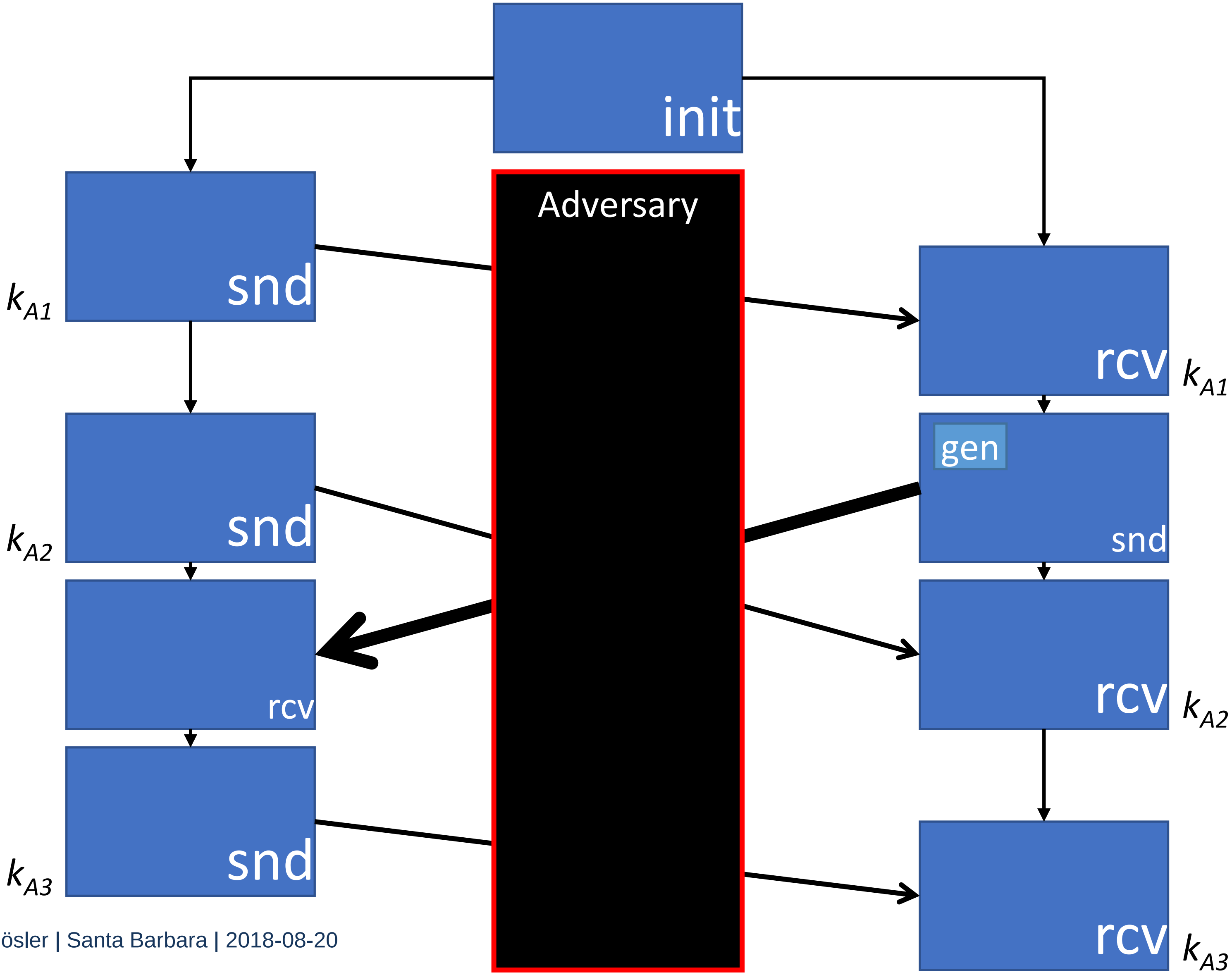
Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob's state



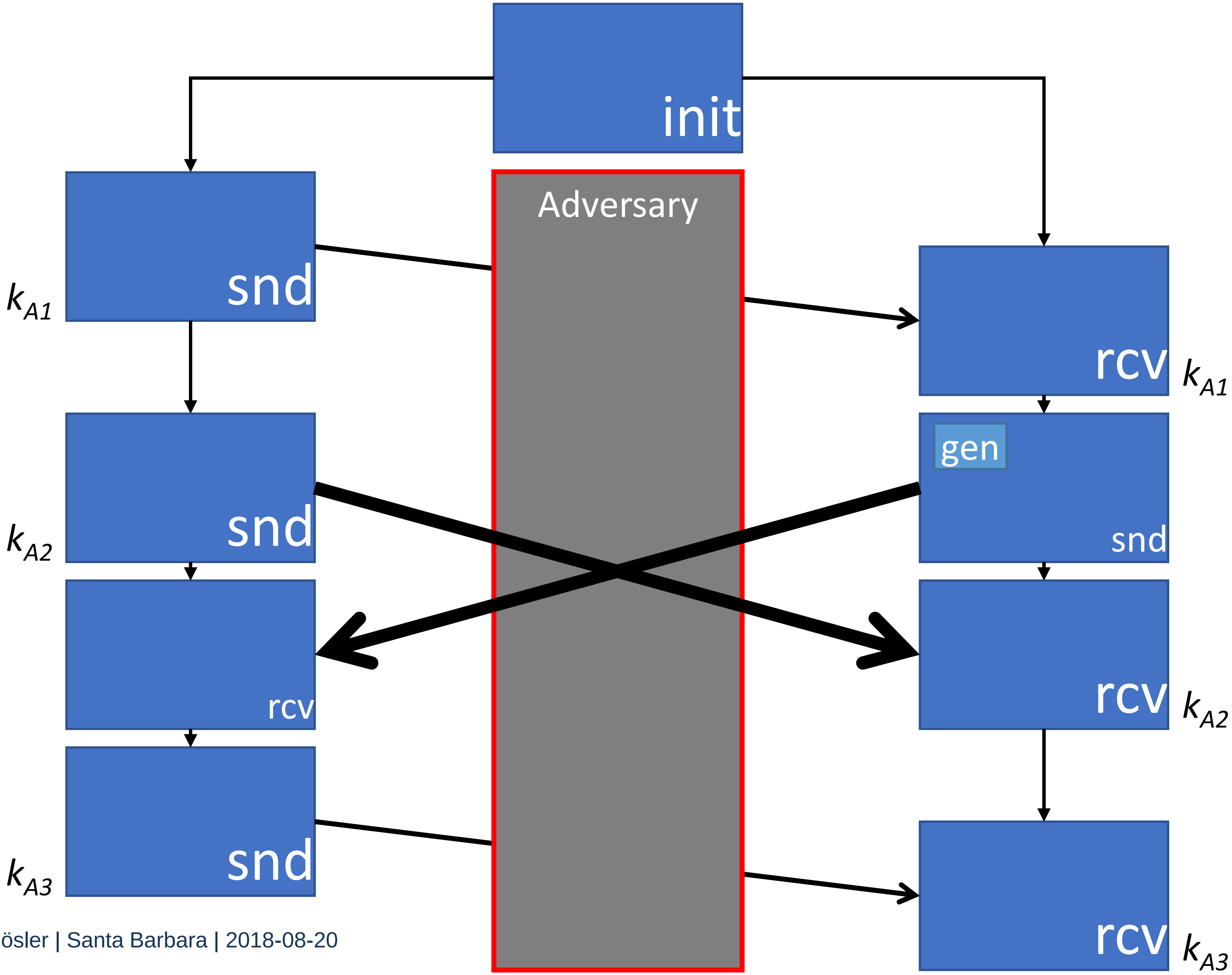
Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob’s state
  → Send new pk



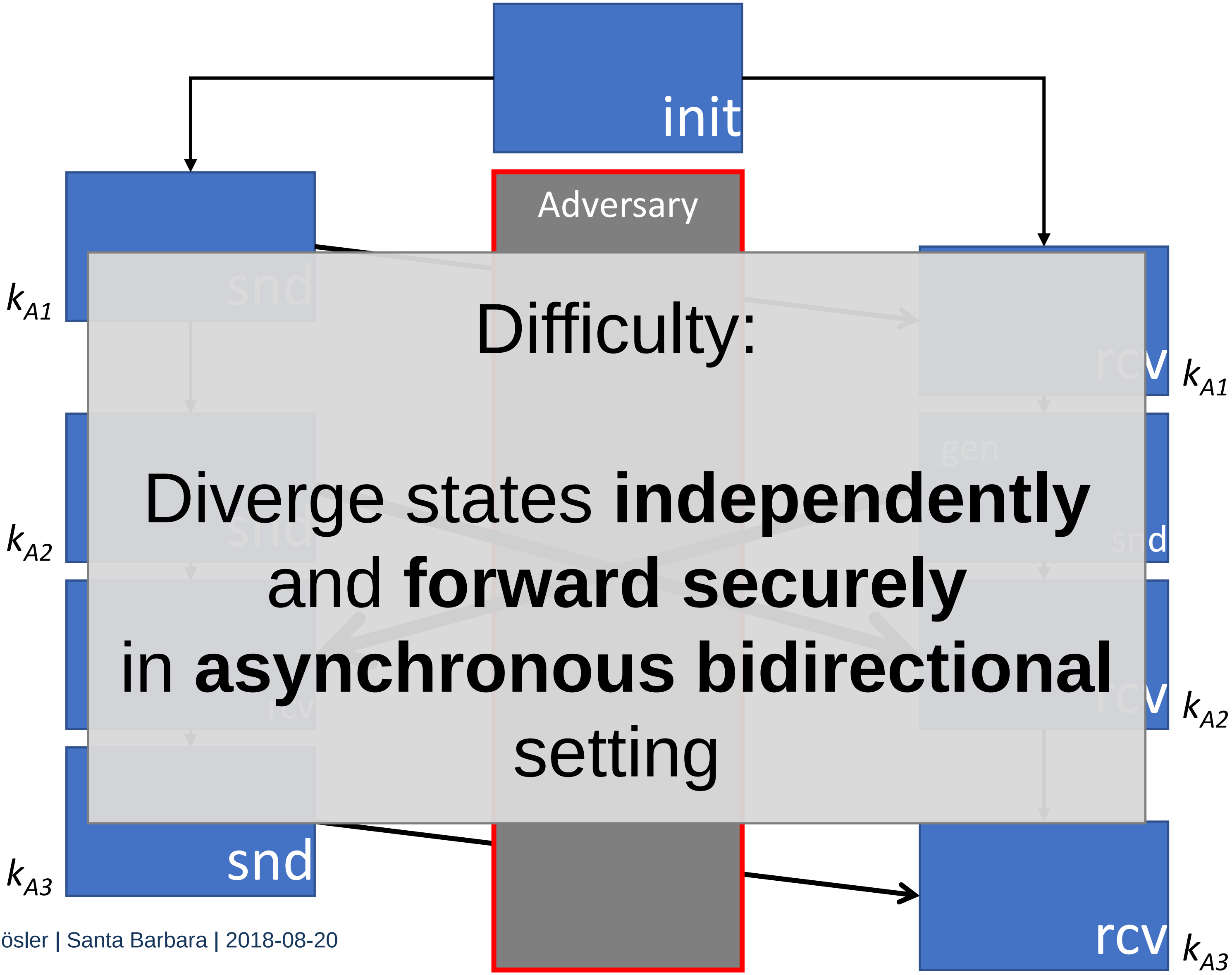
Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob’s state
  → Send new pk
  → Divergence of states



Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob's state
  → Send new pk
  → Divergence of states

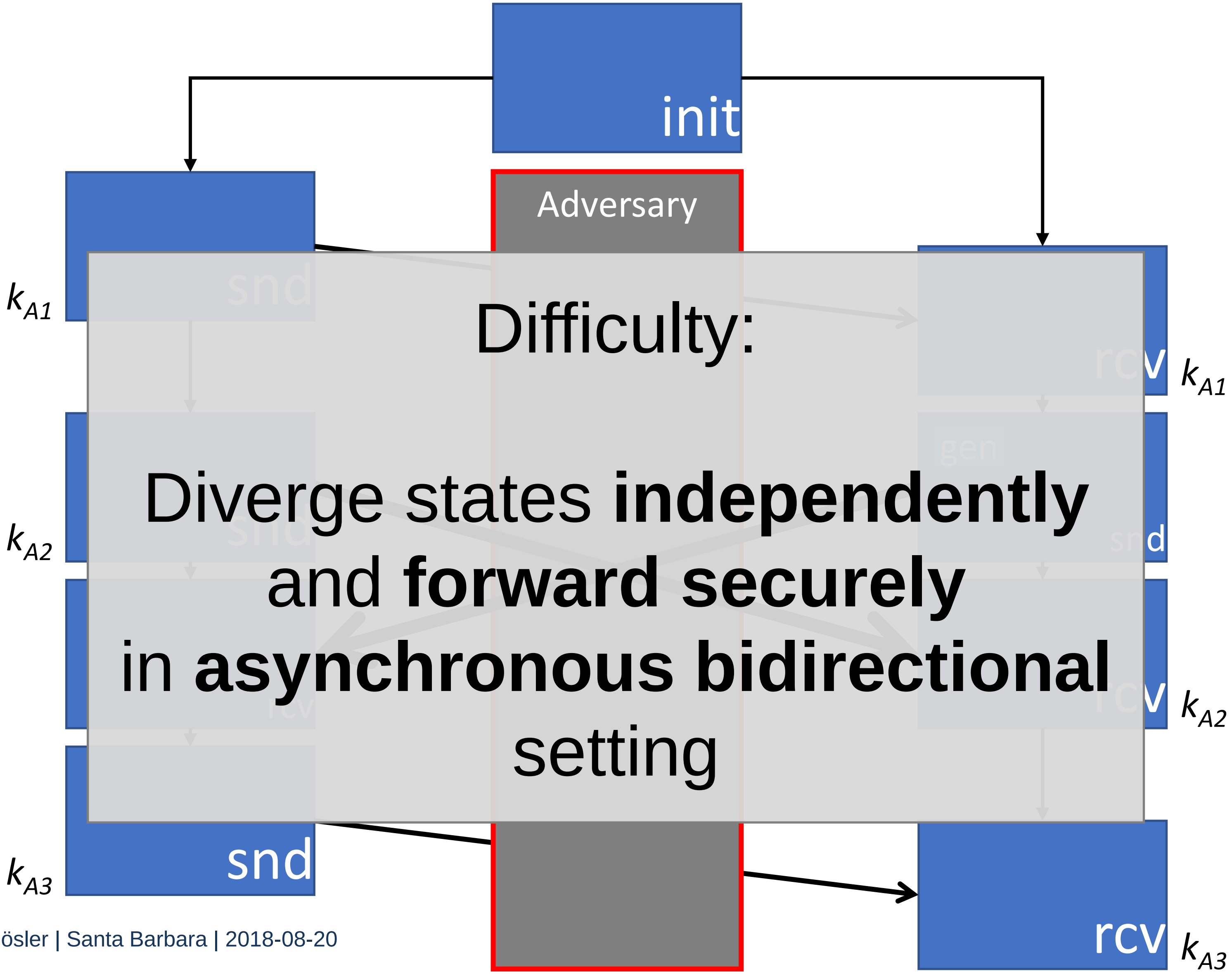


Difficulty:

Diverge states **independently** and **forward securely** in **asynchronous bidirectional** setting

Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob’s state
    → Send new pk
  → Divergence of states
    → Update key pair



Difficulty:

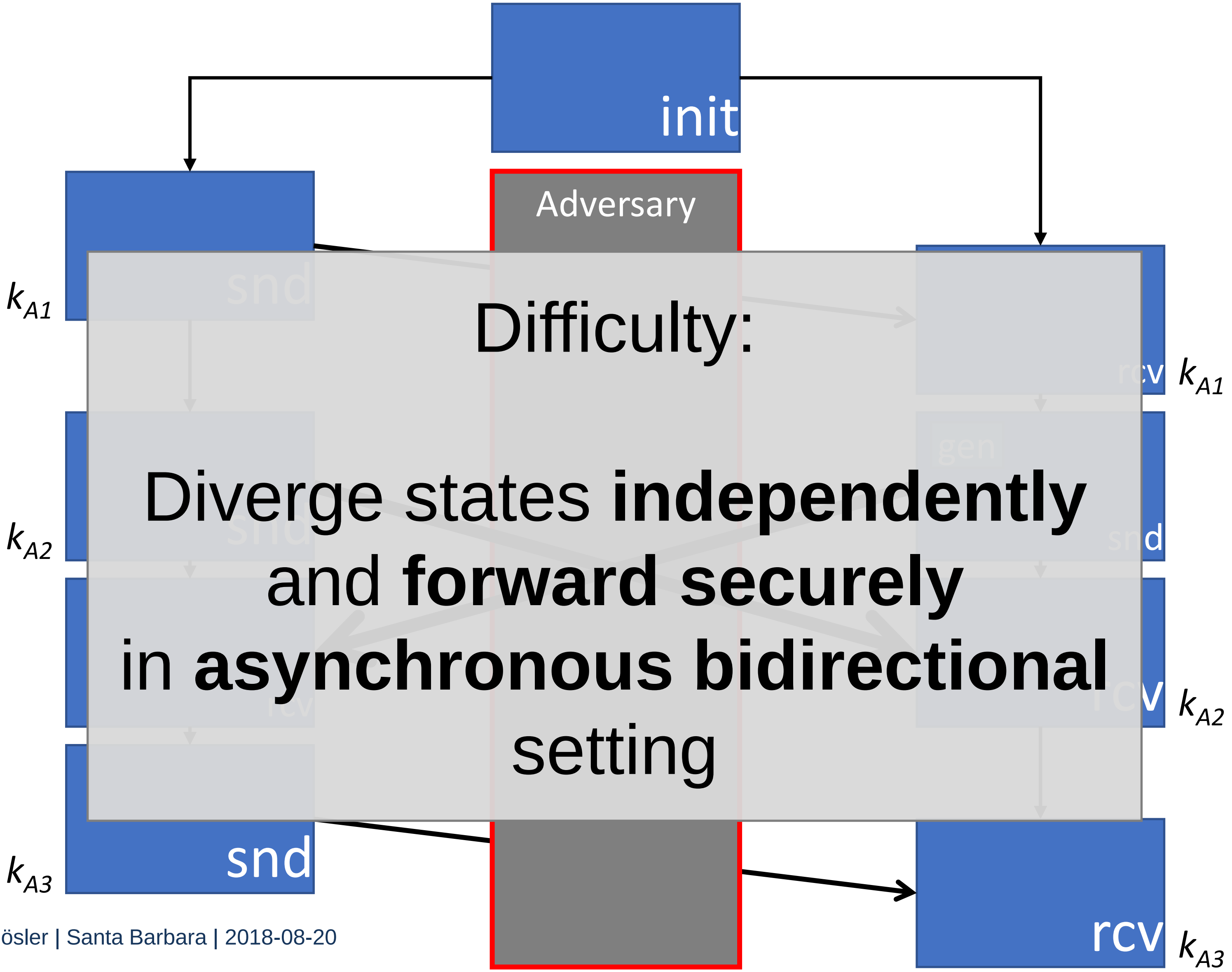
Diverge states **independently** and **forward securely** in **asynchronous bidirectional** setting

Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob's state
  → Send new pk
  → Divergence of states
  → Update key pair

up (sk, *T*)→sk

up (pk, *T*)→pk



Difficulty:

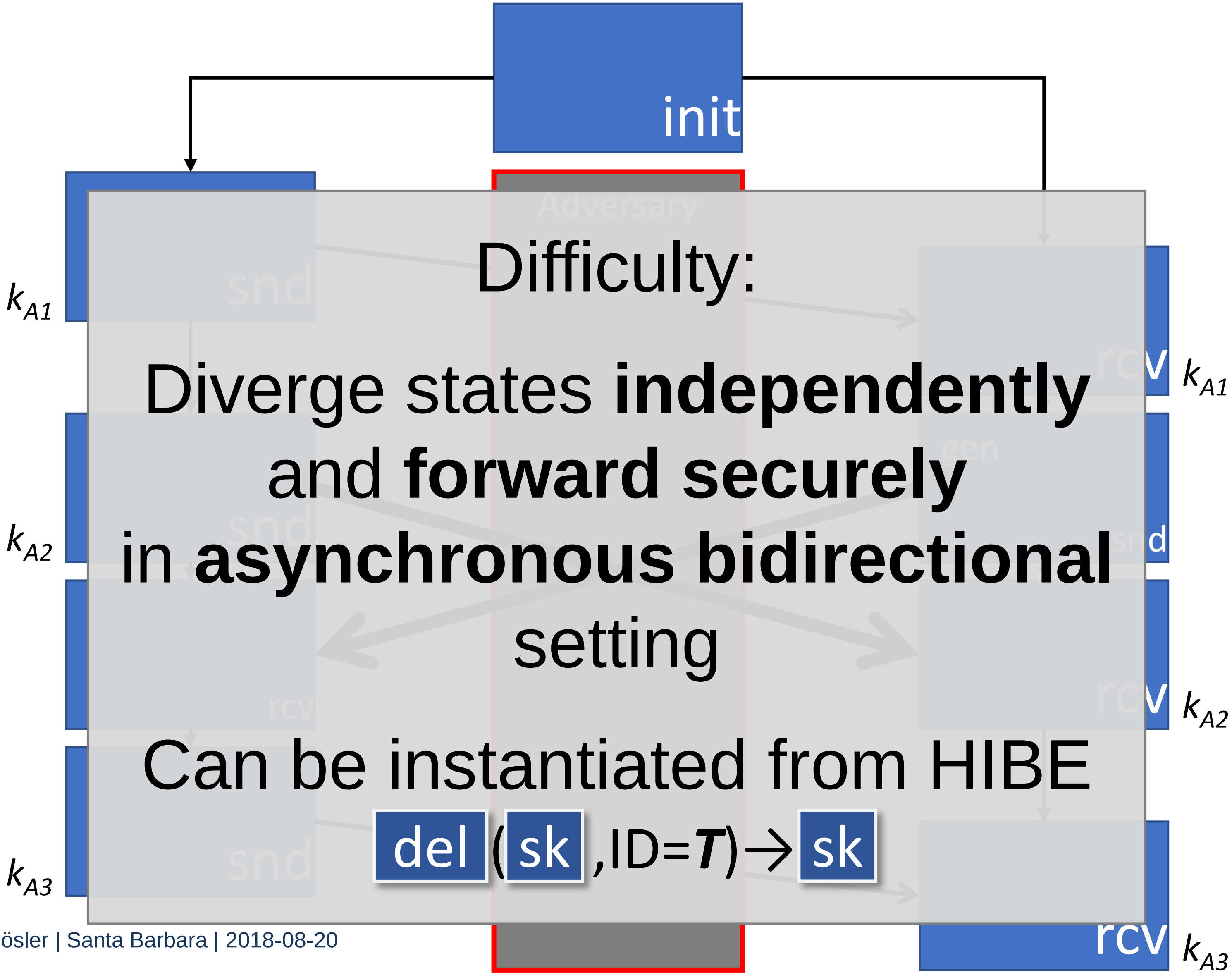
Diverge states **independently** and **forward securely** in **asynchronous bidirectional** setting

Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob’s state
    → Send new pk
  → Divergence of states
    → Update key pair

up(sk, T)→sk

up(pk, T)→pk



Difficulty:

Diverge states **independently** and **forward securely** in **asynchronous bidirectional** setting

Can be instantiated from HIBE

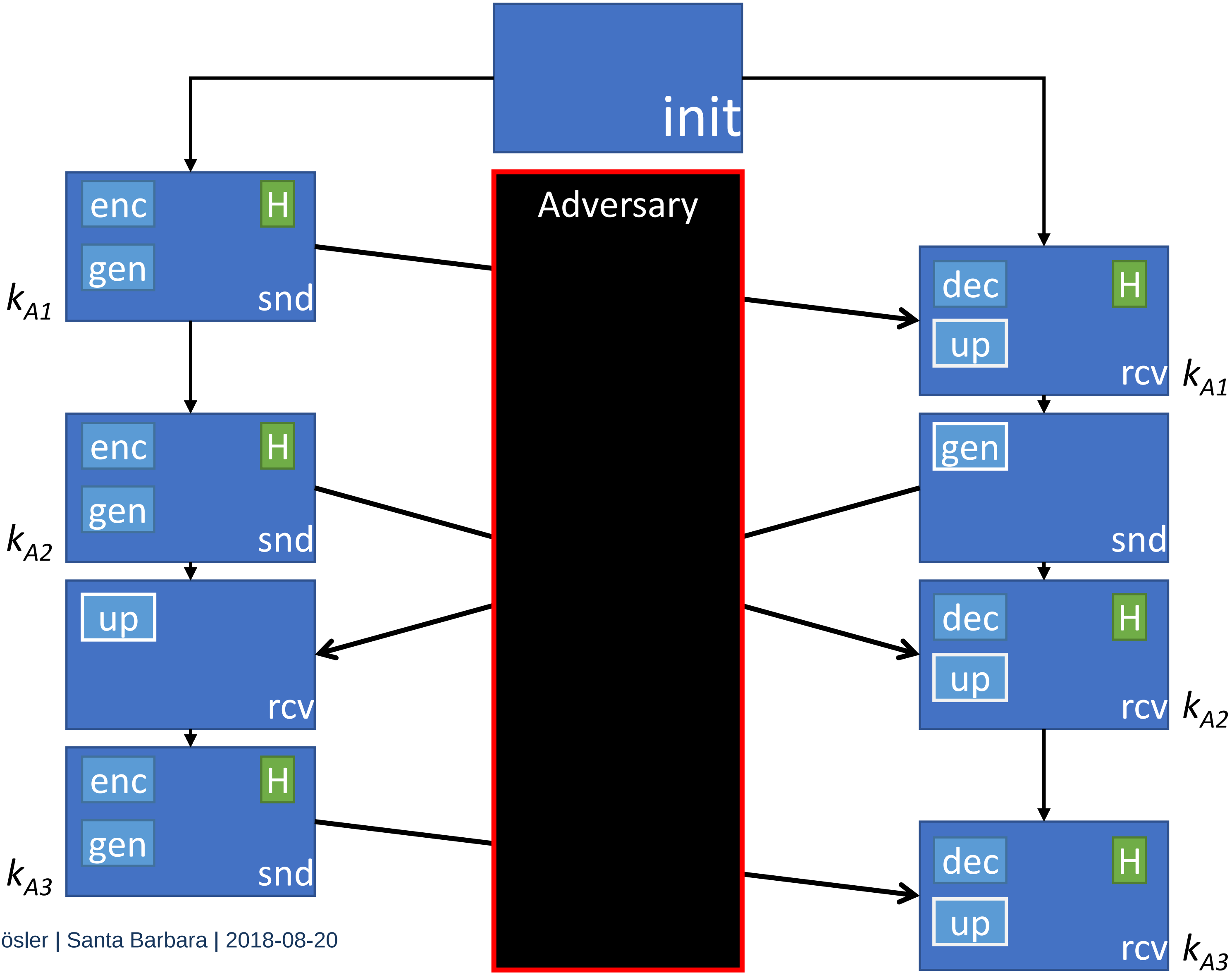
del(sk,ID=T)→sk

Constructing Sesquidirectional RKE

- **Expose Bob**
  ⇒ No future Challenge if synchronous until Bob recovered
  → Forward secrecy and recovery of Bob's state
  → Send new pk
  → Divergence of states
  → Update key pair

up(sk, T)→sk

up(pk, T)→pk

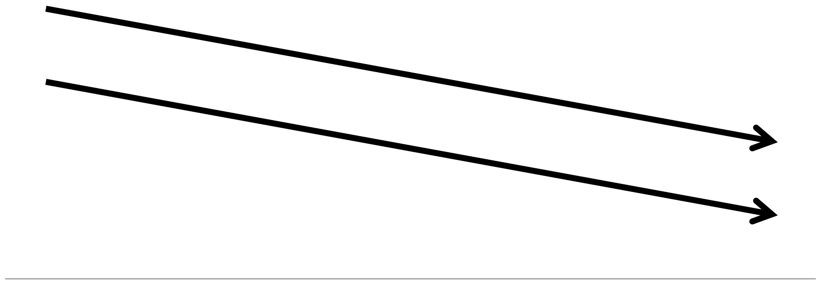


Agenda

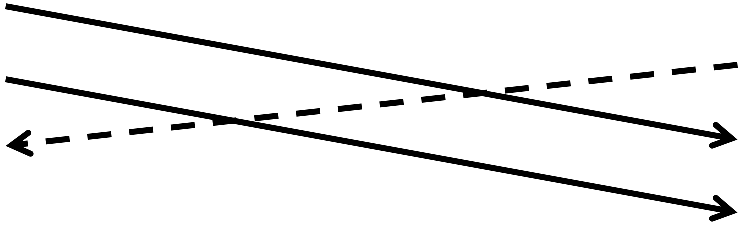
1. The Primitive Ratcheted Key Exchange

2. General Adversary Model

3. Unidirectional Ratcheting
   → Model and Construction



4. Sesquidirectional Ratcheting
   → Model and Construction



5. **Results**

Results

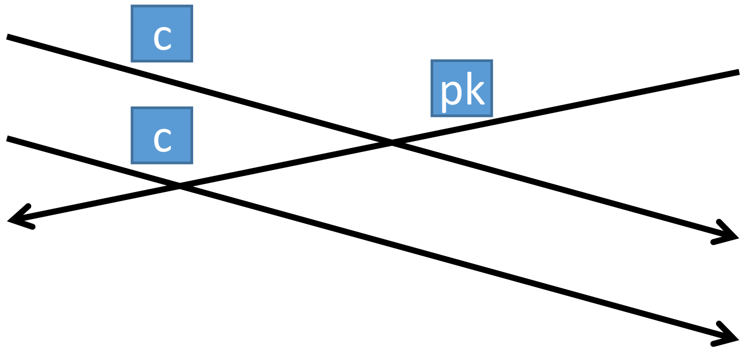
- Unidirectional RKE
    - KEM + ROM (+ MAC)

ia.cr/2018/296 (ext. version) @roeslpa

Results

- Unidirectional RKE
  - KEM + ROM (+ MAC)
- Sesquidirectional RKE
  - Key updatable KEM (+ signatures)
  - # up (sk *T*) = #*crossing* ciphertexts

    → Depth of HIBE practically bounded



ia.cr/2018/296 (ext. version) @roeslpa

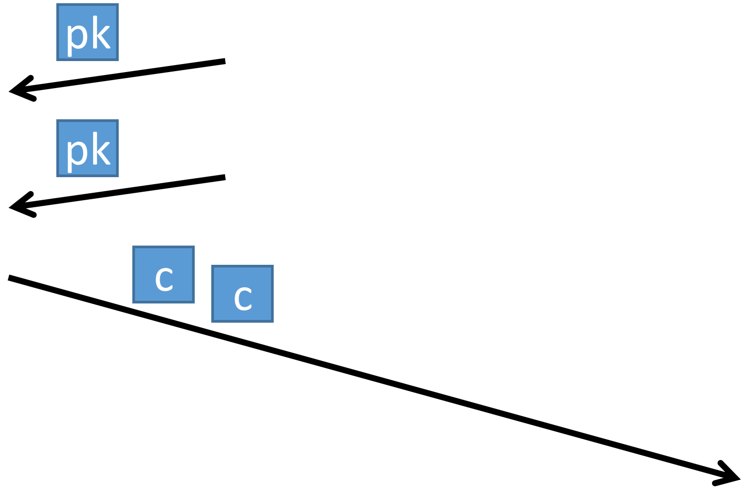
Results

- Unidirectional RKE
  - KEM + ROM (+ MAC)
- Sesquidirectional RKE
  - Key updatable KEM (+ signatures)
  - # up (sk ***T***) = #*crossing* ciphertexts

    → Depth of HIBE practically bounded
  - Multi encapsulation

    → Bounded in ping-pong pattern

    → Alternative: key updatable signatures



ia.cr/2018/296 (ext. version) @roeslpa

Results

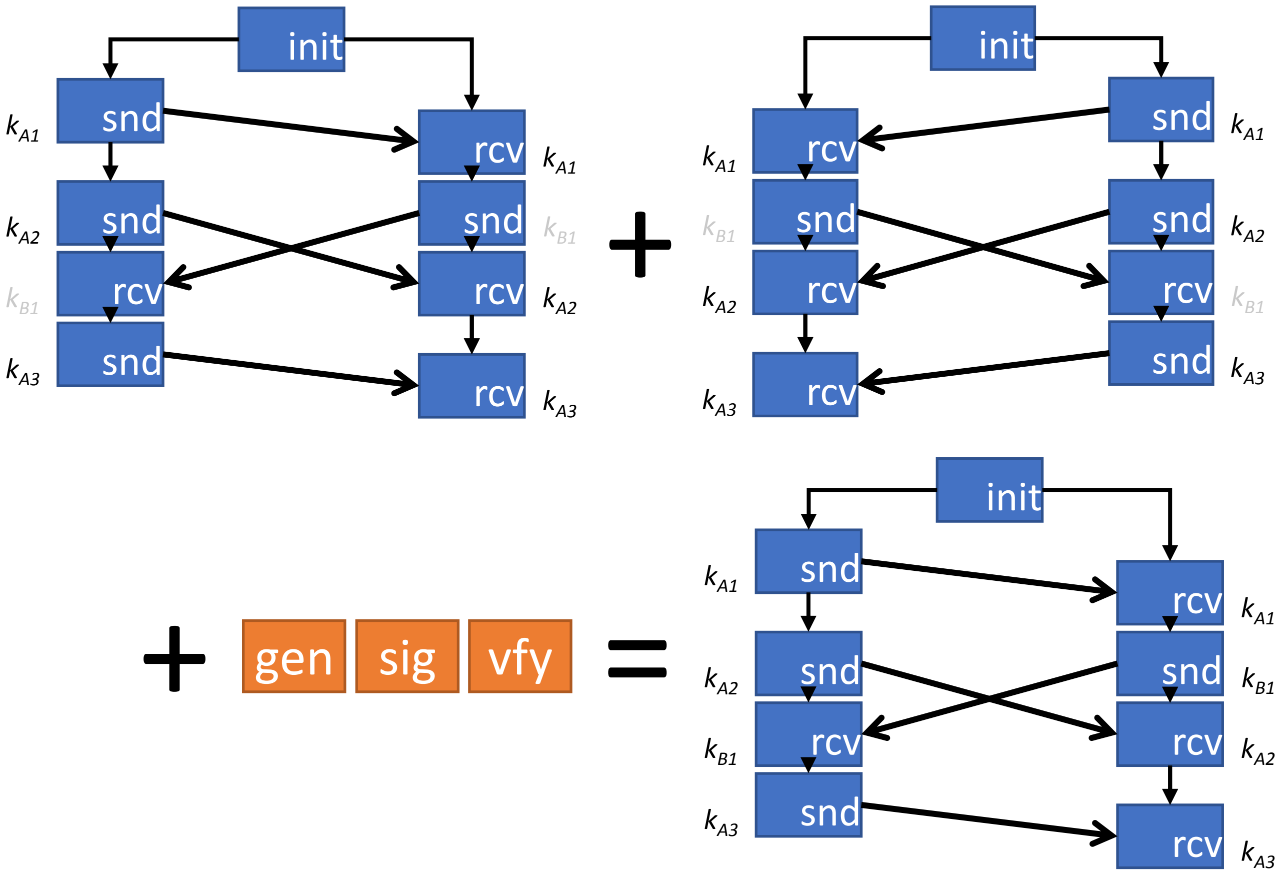
- Unidirectional RKE
  - KEM + ROM (+ MAC)
- Sesquidirectional RKE
  - Key updatable KEM (+ signatures)
  - # up (sk T) = #crossing ciphertexts

    → Depth of HIBE practically bounded
  - Multi encapsulation

    → Bounded in ping-pong pattern

    → Alternative: key updatable signatures
- BRKE = 2x SRKE + OT signatures

  → Build SRKE, BRKE too complex!



ia.cr/2018/296 (ext. version) @roeslpa