Definitional Foundations of Ratcheting and their Impact on Practice

Workshop on Secure Messaging – Eurocrypt 2019

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

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

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

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

- Complex syntax definition
Messaging is complex

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


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
• 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 Dowling¹, Paul Rösler², and Jörg Schwenk²

¹ Information Security Group, Royal Holloway, University of London
benjamin.dowling@rhul.ac.uk
² Hoer-Götz 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
Syntax – Taming complexity

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

Bidirectional ratcheted key exchange (BRKE)
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)
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\textsuperscript{1} and Paul Rösler\textsuperscript{2}

\textsuperscript{1} Information Security Group, Royal Holloway, University of London, bertram.poettering@rhul.ac.uk
\textsuperscript{2} Horst-Görtz Institute for IT Security, Chair for Network and Data Security, Ruhr-University Bochum, paul.roesler@rub.de
• 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)
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
Attacker

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

• 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
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
Unpreventable Attacks
• Impersonation
• Expose Bob
⇒ No future Challenge on Bob’s keys
• Remaining keys secure

Preventable Attacks
• Symmetric leakage
• Active attack \(\not\Rightarrow\) independence of states
• No exposure of Bob’s state, … (more in bidirectional setting)
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)
Unpreventable Attacks
- Impersonation
- Expose Bob
\[ \Rightarrow \text{No future Challenge on Bob's keys} \]
- 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

---

- **Checkmark**
  - can expose local states
- **Question mark**
  - should not distinguish real key from random key
- **Set Random**
- **Challenge**
- **Expose**

---

**Diagram Elements**:
- `snd`: Send
- `rcv`: Receive
- `k1`, `k2`, `k3`, `k4`, `k5`: Keys
- `init`: Initialization
- `Expose`: Expose attack
- `Set Random`: Set random
- `Challenge`: Challenge attack
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
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 Sesquidical 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 $\Leftrightarrow$ key-updatable PKC
  • Unidirectional RKE is part of Sesquidirectional RKE, which is part of Bidirectional RKE

• Key-updatable PKC core primitive of strongly secure messaging
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
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