A Comparison and Combination of Theory Generation and Model Checking for Security Protocol Analysis

By Nicholas J. Hopper, Sanjit A. Seshia, Jeannette M. Wing

Presentation by: Heather Goldsby

Overview

• Tools
  – BRUTUS
  – RVChecker

• Protocols
  – Dolev-Yao
  – Tatabayeshi-Matsuzaki-Newman

• Combination

BRUTUS

• Model checker
• State-space analysis
• Covered by
  – Ali Ebnenasir
  – Zhenxiao Yang
RVChecker

• Theory generation
• Based on Belief Logics (BAN)
• Concept:
  – Finite representation of theory
  – Generated by set of rules and assumptions
  – Check for set membership

RVChecker

• Theory Generation Overview
  – Everything is finite, thus theory is finite
  – Starting with assumptions and messages
  – Apply only shrinking rules
  – To test whether formula is part of theory
    • Backwards chaining using growing rules

RVChecker

• RV logic
  – Extension of BAN
    • Explicit interpretation
      – Idealization step
        » Fail to consider other interpretations
        » Hidden assumptions about safety of message
    • Responsibility
      – Account for principal’s irresponsible behavior
RVChecker

• Explicit Interpretation
  – Allows for explicit idealization of the protocol
  – Interpretation must match pattern
  – M matches all or part of a concrete message
  – M’ represents its intended meaning

| P |= M | P |= Q |= M’ | P |= Q |= M’ |
|-------|-------------|-------------|
| P sees M, then P sees M’ | If P believes M, then P believes M’ | If P believes Q says M, then P believes Q says M’ |

• Responsibility
  – Honesty
    • Sending principal believes any interpretation of message they sent
    • Expressed through legit rules
    • Signed – any message encrypted under private or shared key is signed

| P |= Q |= M | Q |= signed(M, M_s, P, Q) |
|-------------|-------------|
| P |= Q |= M’ | Q |= legit(M) |
| If P believes Q said M, then P believes Q said M’ | If Q believes M’ and Q believes the message is signed, then Q believes M’ is legit |

• Responsibility
  – Secrecy
    • Safe for the intruder to see a message
    • Expressed through maysee rules

• P |= Q maysee (X, Y)
  P |= Q maysee X

• If P believes Q maysee (X, Y), then P believes Q maysee X
RVChecker

- Protocol Analysis & Theory Generation
  1. Specification supplied
     - messages, interpretation rules, principals, belief goals for the
       principals, initial assumptions
  2. Theory generation applied to assumptions
  3. For each message $M_i$, receiver $R_i$
     - formula $R_i \vdash M_i$ is added
     - Theory generation is re-applied
  4. Secrecy is checked
  5. Honesty is checked
  6. Each belief goal is checked for set inclusion

Tatabayeshi-Matsuzaki-Newman Protocol

1. $A \rightarrow S : [S.A.B.\{R_a\}_{K_S}]$
   - A sends a message to Server $S$ consisting of $S$, $A$, $B$, and its
     session key $R_a$ encrypted under $K_S$
  2. $S \rightarrow B : [B.S]$
   - Server $S$ sends a message to principal $B$ consisting of $B$, $S$
  3. $B \rightarrow S : [S.B.A.\{R_b\}_{K_S}]$
   - Principal $B$ sends a message to server $S$ consisting of $S$, $B$, $A$, and its
     session key $R_b$ encrypted under $K_S$
  4. $S \rightarrow A : [A.S.B.R_a \oplus R_b]$
   - Server $S$ sends a message to principal $A$ consisting of $A$, $S$, $B$, and the
     exclusive or of the session key of $A$ and the session key of $B$.

Protocol Flaws
- Secrecy
- Authentication
- Simmons

Dolev-Yao Protocol

1. $A \rightarrow B : [A.\{M\}_{K_B}.A]_{K_B.B}$
   - Principal $A$ sends principal $B$ a message consisting of $A$, $M$
     encrypted under key $K_B$, $A$, encrypted under key $K_B$, and $B$
  2. $B \rightarrow A : [B.\{M\}_{K_A}.B]_{K_A.A}$
   - Principal $B$ sends principal $A$ a message consisting of $B$, $M$
     encrypted under key $K_A$, $B$, encrypted under key $K_A$, and $A$

Protocol Flaws
- Secrecy
- Authentication
BRUTUS – Dolev-Yao analysis

- **INITIATOR** =
  - internal ("begin-initiate", b)
  - send \(<a, b, \{M\}_{Kb}, a\>\)
  - receive \(<b, a, \{M\}_{Ka}, b\>\)
  - internal ("end-initiate", b)

- **RESPONDER** =
  - receive \(<a, b, \{M\}_{Ka}, a\>\)
  - internal ("begin-respond", a)
  - send \(<b, a, \{M\}_{Kb}, b\>\)
  - internal ("end-respond", a)

BRUTUS – Dolev-Yao analysis

- **Secrecy**
  - Check property: \(\neg I \text{ Knows M}\)
  - BRUTUS catches error (property false)
  - B listens for 2 sessions
  - A initiates one session
  - A Intruder I B
    - Start(B, I, K(A, K(b, M)))
    - Send(B, I, K(B, K(A, K(b, M))))
    - Send(I, B, K(I, K(A, K(b, M))))
    - Send(I, B, K(I, K(b, M)))
    -Recv(B, I, K(I, K(b, M)))
    -Recv(B, I, K(I, K(b, M)))
    -Knows(M)

BRUTUS – Dolev-Yao analysis

- **Authentication**
  - Check properties:
    1. If A finishes initiating a message to B, B responded to this message at an earlier point in time
    2. If B finishes responding to a message from A, A initiated sending this message at an earlier point in time
  - BRUTUS catches error (property false)
  - More than 1 session
  - A Intruder I B
    - Start(A, I, K(A, K(b, M)))
    - Send(B, I, K(b, K(A, M)))
    - Receives(B, A, K(A, K(b, M)))
    - A thinks it is communicating with B while really it is communicating with I
    - B thinks it is communicating with I (it really is communicating with I)
RVChecker Dolev-Yao analysis

• Authentication
  – Desired belief goals
    - A ⊨ B ⊨ (A ⊨ M)
      - A believes B says that it is A who says M
    - B ⊨ A ⊨ M
      - B believes it is A who says M
  – RVChecker finds - beliefs are not in the protocol
  – Fixable by encryption

RVChecker Dolev-Yao analysis

• Secrecy
  – Flaw
    - comes from double encryption
    - Lack of authentication (shown on previous slide)
  – RVChecker does not find flaw

Comparison

<table>
<thead>
<tr>
<th>System</th>
<th>Dolev-Yao Auth</th>
<th>Dolev-Yao Secrecy</th>
<th>TMN Auth/Secrecy</th>
<th>TMN Simmons/Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>RVChecker</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>BRUTUS</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
### Comparison

<table>
<thead>
<tr>
<th>BRUTUS</th>
<th>RVChecker</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Fully automated</td>
<td>- Fully automated</td>
</tr>
<tr>
<td>- State-space explosion</td>
<td>- High level</td>
</tr>
<tr>
<td>- Complicated counterexample</td>
<td>- Assumptions = attacks?</td>
</tr>
<tr>
<td>- Number of protocol sessions needed</td>
<td>- Conservative approach to security</td>
</tr>
<tr>
<td>- Considers all principals to be honest</td>
<td></td>
</tr>
</tbody>
</table>

How much previous knowledge of protocols is necessary?

### Combination

**Assumptions to Counterexamples**
- RVChecker identifies assumptions
- Search for counterexamples using BRUTUS
  - For assumption held by A
    - If A believes something about its own behavior
      » Intruder plays role of A
      » Model extra sessions of other principals
    - If A believes something about principal B
      » Intruder plays role of B
      » Model extra sessions of A

**Counterexamples to Assumptions**
- BRUTUS identifies counter examples
- RVChecker then isolates assumptions
  - Model entire counter example trace as a protocol
  - Add assumptions until goal properties satisfied
  - Find assumptions about I playing the role of Q
    - Indicate assumptions made about Q
    - Discovers how I subverted the protocol
Combination – Dolev-Yao

A
Init(b)
Send(A,B,Kb(A,Kb(M)))

Intruder I

B
Send(B,I,Kb(I,Kb(M)))
Recv(I,B,Kb(I,Kb(M)))
Send(I,B,Kb(I,Kb(M)))
Recv(I,B,Kb(I,Kb(M)))
Send(I,B,Kb(I,Kb(M)))
Recv(I,B,Kb(I,Kb(M)))
knows(M)

Recv(B,A,Ka(B,Ka(M)))

Combination Dolev-Yao

• Counterexamples to Assumptions
  – New assumptions necessary
    • I |¬ I may see M
    • I |¬ B may see M
    • B |¬ I may see M
  – Shows B must believe it is ok for the Intruder to see message M

References

RV logic – Legit Rules

- \( P \vdash Q \Rightarrow M \quad P \vdash Q \Rightarrow M' \)
- If \( P \) believes \( Q \) says \( M \), then \( P \) believes \( Q \) says \( M' \)

- \( P \vdash M \quad P \vdash M' \)
- If \( P \) sees \( M \), then \( P \) sees \( M' \)

- \( Q \vdash M' \quad Q \vdash \text{signed}(M, P, Q) \)
- If \( Q \) believes \( M' \) and \( Q \) believes the message is signed, then \( Q \) believes \( M \)

- \( Q \vdash M' \quad Q \vdash \text{legit}(M) \)
- If \( Q \) believes \( M' \), then \( Q \) believes \( M \) is legit

RV logic – maysee rules

- \( P \vdash Q \Rightarrow R \quad P \vdash \text{maysee} K \)
- If \( P \) sees \( K \), the shared key of \( Q \) and \( R \), then \( P \) believes \( Q \) maysee \( K \)

- \( P \vdash \text{maysee} Y \quad P \vdash \text{maysee} X \)
- If \( P \) sees \( Q \) maysee, then \( P \) believes \( Q \) maysee \( Y \)

- \( P \vdash \text{maysee}(X, Y) \quad P \vdash \text{maysee} X \)
- If \( P \) believes \( Q \) maysee \( (X, Y) \), then \( P \) believes \( Q \) maysee \( X \)

- \( P \vdash \text{maysee} X \quad P \vdash \text{maysee} \{X\} \quad K \)
- If \( P \) believes \( Q \) maysee \( X \) and \( P \) believes \( Q \) maysee \( X \) encrypted under key \( K \), then \( P \) believes \( Z \) maysee \( X \) encrypted under key \( K \)
RVChecker

- **Finite Model**
  - Finite number of entities
  - Parties communicating
  - Messages exchanged
  - Types of messages
  - Encryption and decryption keys
  - Finite number of rules of inference
  - Grow in a controlled manner

- **Inference Rules**
  - S-rules (shrinking rules)
    - Conclusion same size or smaller than premises
    - Each variable in conclusion occurs in premises
    - E.g.
  - G-rules (growing rules)
    - Conclusion larger than each premise
    - Each variable in conclusion occurs in premises
    - E.g.
  - Rewrites (same size)
    - A pair of formulas $f_1, f_2$
    - Any occurrence of $f_1$ can be replaced with $f_2$
    - $f_1, f_2$ are the same size and have the same variables
    - E.g.

- **Theory Generation**
  - Modeled as directed a cyclical graph
  - Roots: assumptions and messages
  - 1. Apply S-rules to generate new formulas
    - Can apply G or R rules to generate premises for S-rules
  - 2. Show completeness
    - Backwards chaining with G-Rules & R-rules
BRUTUS - TMN analysis

- INITIATOR =
  internal ("begin-initiate", B)
  send <A,S,B,(R_a)K_s>
  receive <S,A,b,(R_b)K_s>
  internal ("end-initiate", b)

- RESPONDER =
  receive <S,B,a>
  internal/"begin-respond", a)
  send <B,S,a,(R_b)K_s>
  internal ("end-respond", a)

- SERVER =
  receive <A,S,B,(R_a)K_s>
  internal ("begin-initiate", b)
  send <B,S,a,(R_b)K_s>
  receive <b,S,a,(R_a)K_s>
  internal ("begin-respond", a)
  send <S,b,(R_b)K_s>
  initiate ("end-respond", a)

NOTE: The authentication property is not shown for this protocol because it is the same as the Dolev-Yao protocol. BRUTUS catches the error.

---

BRUTUS – TMN analysis

- Secrecy
  - Check properties
    - ¬(I Knows N_B)
      - I cannot read messages passed between A and B
    - ¬(I Knows N_B) ∧ ¬(I Knows N_A)
      - I cannot masquerade as S while communicating with A
  - BRUTUS catches error (both properties false)
    - 2 sessions

---

BRUTUS – TMN analysis

- Intruder IS B
  - Init(B)
  - Send(A,S,B,(R_a)K_s)
  - recv(A,S,B,(R_a)K_s)
  - send(B,A)
  - know(B)
  - recv(B,I,A,(R_b)K_i)
  - know(B)
  - send(U,I,A,(R_i)K_i)
  - recv(U,S,A,(R_i)K_i)
  - send(U,A,I)
  - recv(S,I,A,(R_i)K_i)
  - send(S,A,I)
  - recv(S,I,A,(R_i)K_i)
  - send(S,I,A,(R_i)K_i)
  - recv(S,I,A,(R_i)K_i)
  - send(S,I,A,(R_i)K_i)
  - recv(S,I,A,(R_i)K_i)
RVChecker – TMN analysis

- Authentication
- Desired belief goals
  - $A \models A \leftarrow Rb \rightarrow B$
  - $B \models A \leftarrow Rb \rightarrow B$
  - $A \models B \models A \leftarrow Rb \rightarrow B$
  - $B \models A \models A \leftarrow Rb \rightarrow B$
- RVChecker finds - beliefs are not in the protocol