Subterm-based proof techniques for improving the automation and scope of security protocol analysis

Cas Cremers + Charlie Jacomme + Philip Lukert

∃t. Cispa ⊏ Saarbrücken ⊏ Germany
Subterm-based proof techniques for improving the automation and scope of security protocol analysis.

Cas Cremers + Charlie Jacomme + Philip Lukert
∃t. Cispa ⊏ Saarbrücken ⊏ Germany

Subterms for Tamarin
Subterm-based proof techniques for improving the automation and scope of security protocol analysis

Cas Cremers + Charlie Jacomme + Philip Lukert
St. Cispa ⊏ Saarbrücken ⊏ Germany
Subterm-based proof techniques for improving the automation and scope of security protocol analysis

Cas Cremers \(\sqsubseteq\) Charlie Jacomme \(\sqsubseteq\) Philip Lukert
\(\exists t.\) Cispa \(\sqsubseteq\) Saarbrücken \(\sqsubseteq\) Germany

Protocol Analysis

Data-structures
Subterm-based proof techniques for improving the automation and scope of security protocol analysis

Subterms for Tamarin

Cas Cremers + Charlie Jacomme + Philip Lukert
₹t. Cispa ☛ Saarbrücken ☛ Germany

Protocol Analysis
Data-structures
Subterms
Subterm-based proof techniques for improving the automation and scope of security protocol analysis

Cas Cremers ∙ Charlie Jacomme ∙ Philip Lukert
Dept. Cispa ▫ Saarbrücken ▫ Germany

Protocol Analysis  Data-structures  Subterms  Great Proofs
Protocol Analysis

Verification Tool

it's a Tamarin
Protocol Analysis

TLS 1.3 [1]
5G [2]
WPA2 [3]

Protocol → Verification Tool

[1] Automated Analysis of TLS 1.3 (Cas Cremers, Marko Horvat, Sam Scott, Thyla van der Merwe)
[3] A Formal Analysis of IEEE 802.11’s WPA2 (Cas Cremers, Benjamin Kiesl, Niklas Medinger)
[1] Automated Analysis of TLS 1.3 (Cas Cremers, Marko Horvat, Sam Scott, Thyla van der Merwe)
[3] A Formal Analysis of IEEE 802.11’s WPA2 (Cas Cremers, Benjamin Kiesl, Niklas Medinger)
Protocol Analysis

- TLS 1.3 [1]
- 5G [2]
- WPA2 [3]

[1] Automated Analysis of TLS 1.3 (Cas Cremers, Marko Horvat, Sam Scott, Thyla van der Merwe)
[3] A Formal Analysis of IEEE 802.11’s WPA2 (Cas Cremers, Benjamin Kiesl, Niklas Medinger)
Protocol Analysis

TLS 1.3 [1]  
5G [2]  
WPA2 [3]

Protocol → Verification Tool → Proof  

Property  

Helper Lemmas

[1] Automated Analysis of TLS 1.3 (Cas Cremers, Marko Horvat, Sam Scott, Thyla van der Merwe)  
[3] A Formal Analysis of IEEE 802.11’s WPA2 (Cas Cremers, Benjamin Kiesl, Niklas Medinger)
Protocol Analysis

[1] Automated Analysis of TLS 1.3 (Cas Cremers, Marko Horvat, Sam Scott, Thyla van der Merwe)
[3] A Formal Analysis of IEEE 802.11’s WPA2 (Cas Cremers, Benjamin Kiesl, Niklas Medinger)
Protocol Analysis

- TLS 1.3 [1]
- 5G [2]
- WPA2 [3]

Protocol → Verification Tool → Proof

Property → Attack

Helper Lemmas → Divergence

[1] Automated Analysis of TLS 1.3 (Cas Cremers, Marko Horvat, Sam Scott, Thyla van der Merwe)
[3] A Formal Analysis of IEEE 802.11’s WPA2 (Cas Cremers, Benjamin Kiesl, Niklas Medinger)
Tamarin

[ ] --[ Start ]--> [ Out("ping"), State() ]

required state facts

action

produced state facts

ping

pong
[ ] --[ Start ]--> [ Out("ping"), State() ]
[ In("ping") ] --[ Answer ]--> [ Out("pong") ]
Tamarin

```
[ ]  --[ Start ]-->  [ Out("ping"), State() ]
[ In("ping") ]  --[ Answer ]-->  [ Out("pong") ]
[ In("pong"), State() ]  --[ Finish ]-->  [ ]
```
Tamarin

required state facts

produced state facts

action

[ ] --> [ Start ] --> [ Out("ping"), State() ]
[ In("ping") ] --> [ Answer ] --> [ Out("pong") ]
[ In("pong"), State() ] --> [ Finish ] --> [ ]
Tamarin

Start → Answer → Finish
Start → Answer → Finish

Start → Start → Answer → Start → Finish
Tamarin

Start → Answer → Finish

Start → Start → Answer → Start → Finish

∀ Finish ⇒ ∃ Answer
∀ Finish → ∃ Answer

Start → Answer → Finish

Start → Start → Answer → Start → Finish
Symbolic Attacker

\[ \forall \text{ Finish} \Rightarrow \exists \text{ Answer} \]

Start → Answer → Finish

Start → Start → Answer → Start → Finish
Symbolic Attacker
• observe all Out(...)
Symbolic Attacker

- observe all Out(...)
- controls all In(...)

\[ \forall \text{Finish} \implies \exists \text{Answer} \]
Symbolic Attacker
- observe all Out(...)
- controls all In(...)
- drop messages

∀ Finish ⇒ ∃ Answer

Start → Answer → Finish

Start → Start → Answer → Start → Finish
Symbolic Attacker
- observe all `Out(...)`
- controls all `In(...)`
- drop messages
- send a "pong"

∀ Finish ⇒ ∃ Answer

Start → Answer → Finish

Start → Start → Answer → Start → Finish
Symbolic Attacker
- observe all Out(...)
- controls all In(...)
- drop messages
- send a "pong"

∀ Finish ⇒ ∃ Answer

Start → Answer → Finish

Start → Start → Answer → Start → Finish

attacker sends "pong"
Symbolic Attacker
- observe all Out(...)
- controls all In(...)
  - drop messages
  - send a "pong"

∀ Finish ⇒ ∃ Answer

Start → Answer → Finish
Start → Start → Answer → Start → Finish

attacker sends "pong"
(Unbounded) Data Structures
(Unbounded) Data Structures

• Trees
(Unbounded) Data Structures

- Trees
  - TreeKEM, Merkle-Trees, ...
(Unbounded) Data Structures

- Trees
  - TreeKEM, Merkle-Trees, ...
  - prove invariant over all sub-trees
(Unbounded) Data Structures

- Trees
  - TreeKEM, Merkle-Trees, ...
  - prove invariant over all sub-trees
- Chains
Trees
  • TreeKEM, Merkle-Trees, ...
  • prove invariant over all sub-trees

Chains
  • Hash-Chains, Blockchains, ...
(Unbounded) Data Structures

- Trees
  - TreeKEM, Merkle-Trees, ...
  - prove invariant over all sub-trees
- Chains
  - Hash-Chains, Blockchains, ...
  - blockchain: is block X in the chain?
(Unbounded) Data Structures

- Trees
  - TreeKEM, Merkle-Trees, ...
  - prove invariant over all sub-trees
- Chains
  - Hash-Chains, Blockchains, ...
  - blockchain: is block X in the chain?
- Counters
(Unbounded) Data Structures

- **Trees**
  - TreeKEM, Merkle-Trees, ...
  - prove invariant over all sub-trees
- **Chains**
  - Hash-Chains, Blockchains, ...
  - blockchain: is block X in the chain?
- **Counters**
  - WPA-2, 5G, ...
(Unbounded) Data Structures

• Trees
  • TreeKEM, Merkle-Trees, ...
  • prove invariant over all sub-trees

• Chains
  • Hash-Chains, Blockchains, ...
  • blockchain: is block X in the chain?

• Counters
  • WPA-2, 5G, ...
  • is $a < b$?
(Unbounded) Data Structures

- Trees
  - TreeKEM, Merkle-Trees, ...
  - prove invariant over all sub-trees
- Chains
  - Hash-Chains, Blockchains, ...
  - blockchain: is block X in the chain?
- Counters
  - WPA-2, 5G, ...
  - is $a < b$ ?
→ Divergence
(Unbounded) Data Structures

• Trees
  • TreeKEM, Merkle-Trees, ...
  • prove invariant over all sub-trees
• Chains
  • Hash-Chains, Blockchains, ...
  • blockchain: is block X in the chain?
• Counters
  • WPA-2, 5G, ...
  • is \( a < b \) ?
→ Divergence

Solution: Subterms
Our Contribution
Our Contribution

Modeling
Our Contribution

Modeling

- Subterm-Predicate "□"
- we can now state $x \sqsubset h(h(x))$
Our Contribution

Modeling

- Subterm-Predicate "□"
  - we can now state $x □ h(h(x))$

- New Tamarin-Result-Type
  - "we don't know"
  - does $x □ x ⊕ y$ hold?
Our Contribution

Modeling

- Subterm-Predicate "□"
  - we can now state $x \sqsubseteq h(h(x))$

- New Tamarin-Result-Type
  - "we don't know"
  - does $x \sqsubseteq x \oplus y$ hold?

- Natural Numbers
  - adding a "+"-operator
Our Contribution

**Modeling**

- Subterm-Predicate "\(\sqsubseteq\)"
  - we can now state \(x \sqsubseteq h(h(x))\)

- New Tamarin-Result-Type
  - "we don't know"
  - does \(x \sqsubseteq x \oplus y\) hold?

- Natural Numbers
  - adding a "\(+\)"-operator

**Proof Techniques**

- "under the hood"
  - Algorithm for Numbers
  - Monotonicity
  - Fresh Ordering
Our Contribution

Modeling

- Subterm-Predicate "\(\sqsubseteq\)"
  - we can now state \(x \sqsubseteq h(h(x))\)
- New Tamarin-Result-Type
  - "we don't know"
  - does \(x \sqsubseteq x \oplus y\) hold?
- Natural Numbers
  - adding a "+"-operator

Proof Techniques

"under the hood"
- Algorithm for Numbers
- Monotonicity
- Fresh Ordering

Case Studies

- New Proofs
- Application to Old Proofs
(small) Numbers
(small) Numbers

- well studied: associative and commutative (AC) operator $\#$

\[
\begin{align*}
(a \# b) \# c &= a \# (b \# c) \\
a \# b &= b \# a
\end{align*}
\]
(small) Numbers

• well studied: associative and commutative (AC) operator $\oplus$
• used as multiset: $a \oplus b \oplus b = \{a, b, b\}$

$(a \oplus b) \oplus c = a \oplus (b \oplus c)$

$a \oplus b = b \oplus a$
(small) Numbers

• well studied: associative and commutative (AC) operator $\oplus$
• used as multiset: $a \oplus b \oplus b = \{a, b, b\}$
• used for counting: one $\oplus$ one $\oplus$ one = 3
(small) Numbers

- well studied: associative and commutative (AC) operator $\dagger$
- used as multiset: $a \dagger b \dagger b = \{a, b, b\}$
- used for counting: $\text{one} \dagger \text{one} \dagger \text{one} = 3$
- our improvement:
  - type system, dedicated operator $+$

$$(a \dagger b) \dagger c = a \dagger (b \dagger c)$$

$$a \dagger b = b \dagger a$$
(small) Numbers

- well studied: associative and commutative (AC) operator \( \dagger \)
- used as multiset: \( a \dagger b \dagger b = \{a, b, b\} \)
- used for counting: one \( \dagger \) one \( \dagger \) one = 3
- our improvement:
  - type system, dedicated operator +
  - comperator: \( a < b \iff \exists x. a + x = b \)
(small) Numbers

• well studied: associative and commutative (AC) operator \( \oplus \)
• used as multiset: \( a \oplus b \oplus b = \{a, b, b\} \)
• used for counting: one \( \oplus \) one \( \oplus \) one = 3
• our improvement:
  • type system, dedicated operator +
  • comperator: \( a < b \iff \exists x. a + x = b \)
  • dedicated algorithm: \( a < b < a+2 \Rightarrow b = a+1 \)
Dedicated Proof Techniques

Monotonicity

\[ F(\neg k) \]
\[ F(h(\neg k)) \]
\[ F(h(h(\neg k))) \]
\[ \ldots \]
Monotonicity

- \( F(s)@i, \quad F(t)@j \)
Monotonicity

- $F(s)@i$, $F(t)@j$
- $s \subset t \Rightarrow i < j$
Monotonicity

- $F(s)@i, F(t)@j$
- $s \sqsubseteq t \Rightarrow i < j$
- $s = t \Rightarrow i = j$
Monotonicity

- \( F(s)@i, \ F(t)@j \)
- \( s \sqsubseteq t \Rightarrow i < j \)
- \( s = t \Rightarrow i = j \)
- \( i \neq j \Rightarrow s \neq t \)
Monotonicity

- $F(s)@i, F(t)@j$
- $s \subseteq t \Rightarrow i < j$
- $s = t \Rightarrow i = j$
- $i \neq j \Rightarrow s \neq t$
- some more ...
Monotonicity

- $F(s)@i, F(t)@j$
  - $s \sqsubseteq t \Rightarrow i < j$
  - $s = t \Rightarrow i = j$
  - $i \neq j \Rightarrow s \neq t$
  - some more ...

- 10x speed-up of WPA-2 proof
Monotonicity

- \( F(\mathbf{s})@i, \ F(\mathbf{t})@j \)
  - \( s \sqsubset t \Rightarrow i < j \)
  - \( s = t \Rightarrow i = j \)
  - \( i \neq j \Rightarrow s \neq t \)
  - some more ...

- 10x speed-up of WPA-2 proof

Fresh Order

- \( F(h(h(\sim k))) \)
- \( \text{Using}(h(\sim k)) \)
- \( Fr(\sim k) \)
Monotonicity

- $F(s)@i, F(t)@j$
- $s \subseteq t \Rightarrow i < j$
- $s = t \Rightarrow i = j$
- $i \neq j \Rightarrow s \neq t$
- some more ...

- 10x speed-up of WPA-2 proof

Fresh Order

- time-ordering
  $Fr(\sim k) < Using(\sim k)$

- $Fr(\sim k)$
  $Using(h(\sim k))$
### Monotonicity

- \( F(s)@i, F(t)@j \)
  - \( s \sqsubset t \Rightarrow i < j \)
  - \( s = t \Rightarrow i = j \)
  - \( i \neq j \Rightarrow s \neq t \)
  - some more ...

- 10x speed-up of WPA-2 proof

### Fresh Order

- time-ordering \( Fr(~k) < Using(~k) \)
  - great with \( \sqsubset \)
Monotonicity

- \( F(s)@i, \ F(t)@j \)
- \( s \subseteq t \Rightarrow i < j \)
- \( s = t \Rightarrow i = j \)
- \( i \neq j \Rightarrow s \neq t \)
- some more ...

- 10x speed-up of WPA-2 proof

Fresh Order

- time-ordering \( Fr(~k) < Using(~k) \)
- great with \( \subseteq \)

- 30x speed-up of CH'07 RFID proof
Applied to Existing Models

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Runtime</th>
<th>Helper-Lemmas</th>
<th>Why is it faster?</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPA-2</td>
<td>1:20h → 7min</td>
<td>74 → 73</td>
<td>monotonicity (of counters)</td>
</tr>
<tr>
<td>5G</td>
<td>8min → 2min</td>
<td>7 → 6</td>
<td>our number system</td>
</tr>
<tr>
<td>YubiKey</td>
<td>20s → 1s</td>
<td>4 → 3</td>
<td>our number system</td>
</tr>
<tr>
<td>PKCS#11</td>
<td>1min → 10s</td>
<td>4 → 0</td>
<td>each single improvement</td>
</tr>
<tr>
<td>CH'07 RFID</td>
<td>50min → 2min</td>
<td>0 → 0</td>
<td>fresh order</td>
</tr>
</tbody>
</table>
# Applied to Existing Models

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Runtime</th>
<th>Helper-Lemmas</th>
<th>Why is it faster?</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPA-2</td>
<td>1:20h → 7min</td>
<td>74 → 73</td>
<td>monotonicity (of counters)</td>
</tr>
<tr>
<td>5G</td>
<td>8min → 2min</td>
<td>7 → 6</td>
<td>our number system</td>
</tr>
<tr>
<td>YubiKey</td>
<td>20s → 1s</td>
<td>4 → 3</td>
<td>our number system</td>
</tr>
<tr>
<td>PKCS#11</td>
<td>1min → 10s</td>
<td>4 → 0</td>
<td>each single improvement</td>
</tr>
<tr>
<td>CH'07 RFID</td>
<td>50min → 2min</td>
<td>0 → 0</td>
<td>fresh order</td>
</tr>
</tbody>
</table>
# Applied to Existing Models

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Runtime</th>
<th>Helper-Lemmas</th>
<th>Why is it faster?</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPA-2</td>
<td>1:20h → 7min</td>
<td>74 → 73</td>
<td>monotonicity (of counters)</td>
</tr>
<tr>
<td>5G</td>
<td>8min → 2min</td>
<td>7 → 6</td>
<td>our number system</td>
</tr>
<tr>
<td>YubiKey</td>
<td>20s → 1s</td>
<td>4 → 3</td>
<td>our number system</td>
</tr>
<tr>
<td>PKCS#11</td>
<td>1min → 10s</td>
<td>4 → 0</td>
<td>each single improvement</td>
</tr>
<tr>
<td>CH'07 RFID</td>
<td>50min → 2min</td>
<td>0 → 0</td>
<td>fresh order</td>
</tr>
</tbody>
</table>
## Applied to Existing Models

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Runtime</th>
<th>Helper-Lemmas</th>
<th>Why is it faster?</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPA-2</td>
<td>1:20h → 7min</td>
<td>74 → 73</td>
<td>monotonicity (of counters)</td>
</tr>
<tr>
<td>5G</td>
<td>8min → 2min</td>
<td>7 → 6</td>
<td>our number system</td>
</tr>
<tr>
<td>YubiKey</td>
<td>20s → 1s</td>
<td>4 → 3</td>
<td>our number system</td>
</tr>
<tr>
<td>PKCS#11</td>
<td>1min → 10s</td>
<td>4 → 0</td>
<td>each single improvement</td>
</tr>
<tr>
<td>CH'07 RFID</td>
<td>50min → 2min</td>
<td>0 → 0</td>
<td>fresh order</td>
</tr>
</tbody>
</table>
## Applied to Existing Models

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Runtime</th>
<th>Helper-Lemmas</th>
<th>Why is it faster?</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPA-2</td>
<td>1:20h → 7min</td>
<td>74 → 73</td>
<td>monotonicity (of counters)</td>
</tr>
<tr>
<td>5G</td>
<td>8min → 2min</td>
<td>7 → 6</td>
<td>our number system</td>
</tr>
<tr>
<td>YubiKey</td>
<td>20s → 1s</td>
<td>4 → 3</td>
<td>our number system</td>
</tr>
<tr>
<td>PKCS#11</td>
<td>1min → 10s</td>
<td>4 → 0</td>
<td>each single improvement</td>
</tr>
<tr>
<td>CH'07 RFID</td>
<td>50min → 2min</td>
<td>0 → 0</td>
<td>fresh order</td>
</tr>
</tbody>
</table>
Our Proofs

- TreeKEM
- distributed tree
- forward-secrecy
Our Proofs

- TreeKEM
  - distributed tree
  - forward-secrecy
- S/Key
  - hash-chain
  - authentication

https://www.ftsafe.com/products/OTP/Single_Button_OTP
Our Proofs

- TreeKEM
  - distributed tree
  - forward-secrecy
- S/Key
  - hash-chain
  - authentication
- Tesla Scheme 2
  - hash-chain like S/Key
  - authentication, secrecy
  - prev. example of Tamarins limits

https://www.ftsafe.com/products/OTP/Single_Button_OTP
Paper Summary
Paper Summary

Tamarin
Paper Summary

Tamarin

Complex Protocols
Paper Summary

- Tamarin
- Subterms + Proof Techniques
- Divergence
- Complex Protocols
Paper Summary

Complex Protocols

Subterms + Proof Techniques

Divergence

Tamarin

eprint.iacr.org/2022/1130