Security Protocol Design and Symbolic Analysis: Hybrid Protocols, Derived Adversary Models, and Refined Equational Theories

Defense of **Dhekra Mahmoud**

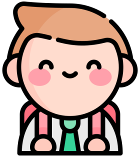
LIMOS, University of Clermont-Auvergne

Under the supervision of **Pascal Lafourcade** and **Jannik Dreier**

June 11, 2025

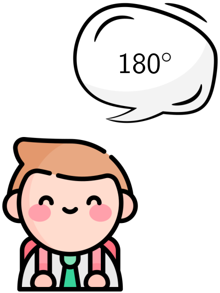
A geometry question

What is the **sum** of the angles in a triangle?



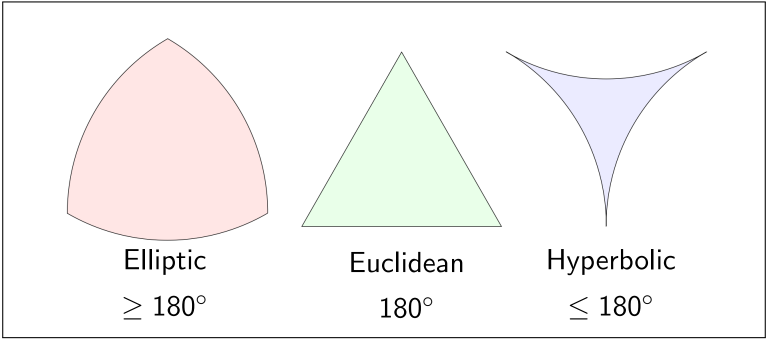
A geometry question

What is the **sum** of the angles in a triangle?

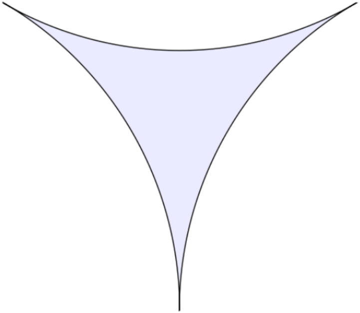


A Geometry Question

What is the **sum** of the angles in a triangle?



A Geometry Question

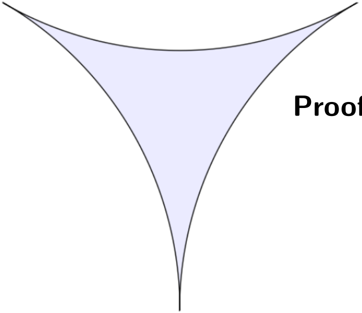


Thales' theorem ✗

Pythagorean theorem ✗

Trigonometric formulas ✗

A Geometry Question



Proofs are model-dependent!

Thales' theorem ✗

Pythagorean theorem ✗

Trigonometric formulas ✗

A Geometry Question

- **Hyperbolic** Geometry can be approximated by **Euclidean** Geometry!
- The **approximation** is effective on "short" distances
- "Simple" models can be very "efficient"!

A Geometry Question

- **Hyperbolic** Geometry can be approximated by **Euclidean** Geometry!
- The **approximation** is effective on "short" distances
- "Simple" models can be very "efficient"!

Paul Valéry

"What is simple is always false. What is not, is unusable."

Analogy with Security Protocols

- Security Protocol's proofs are **model-dependent**
- Attacks on **proven** protocols reveal model **gaps**, not proof flaws

Cryptography (Quic Introduction)

Symmetric encryption

- A **secret** key sk, an **encryption** algorithm senc, a **decryption** algorithm sdec
- $\text{sdec}(\text{senc}(m, sk), sk) = m$

Public key encryption

- A **secret** key sk, a public key pk, an **encryption** algorithm aenc, a **decryption** algorithm adec
- $\text{adec}(\text{aenc}(m, pk), sk) = m$

Cryptography (Quic Introduction)

Symmetric encryption

- A **secret** key sk, an **encryption** algorithm senc, a **decryption** algorithm sdec
- $\text{sdec}(\text{senc}(m, sk), sk) = m$

Public key encryption

- A **secret** key sk, a public key pk, an **encryption** algorithm aenc, a **decryption** algorithm adec
- $\text{adec}(\text{aenc}(m, pk), sk) = m$

ElGamal 1984: $pk = g^{sk}$, $\text{aenc}(m, pk) = (g^r, m \cdot pk^r)$

Cryptography: Introduction

Diffie-Hellman Key Exchange

- From **public** keys $pk_1 = g^x$ and $pk_2 = g^y$, a **shared secret** key $sk = g^{xy}$ derived

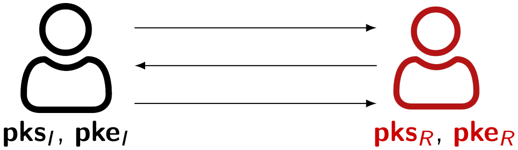
Massage Authentication Code (MAC)

- A **secret** key k, a message m, and an algorithm $MAC(m, k)$

The WireGuard Protocol (Donenfeld 2017)

- A Virtual Private Network (VPN)
- Integrated into the Linux Kernel
- Diffie-Hellman key exchange
- Public Static keys ($\mathbf{pks}_I$, $\mathbf{pks}_R$)
- Ephemeral keys ($\mathbf{pke}_I$, $\mathbf{pke}_R$)

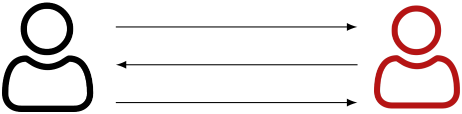
WireGuard
FAST, MODERN, SECURE VPN TUNNEL



The WireGuard Protocol

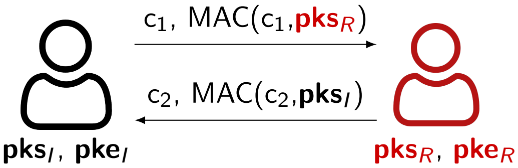
Security Properties

- Secrecy of session keys
- Mutual authentication
- Identity Hiding (Anonymity)

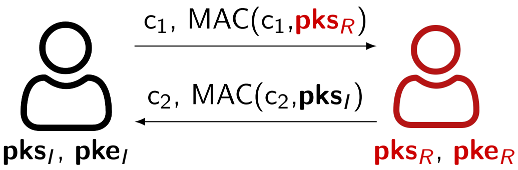


$\mathbf{pks}_I$, $\mathbf{pke}_I$ $\mathbf{pks}_R$, $\mathbf{pke}_R$

First Messages of WireGuard



Does the Protocol guarentee Anonymity?

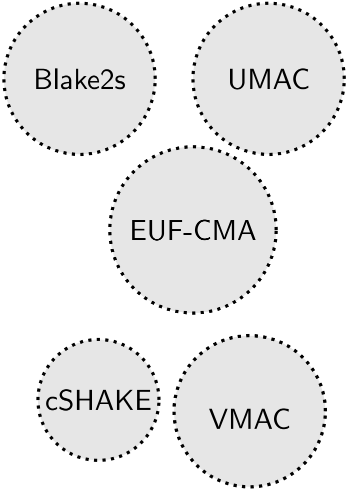


Intruder

- Intercept exchanged messages
- Know **pks**$_R$ and c_1
- Compute MAC(c_1,**pks**$_R$)

Attack on Anonymity

- Independent from the used MAC
- Independent from the used cryptographic assumptions

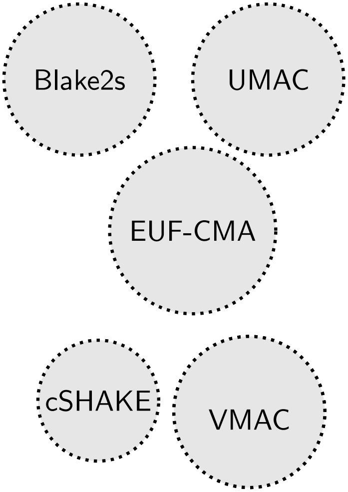


Attack on Anonymity

- Independent from the used MAC
- Independent from the used cryptographic assumptions

Symbolic Model

- The MAC as an abstracted function with an arity 2
- Attacker intercept, delay and inject messages



Public Key Encryption in the Symbolic Model

- **Public key** $\mathrm{pk}(sk)$
- **Encryption** $\mathrm{aenc}(m, \mathrm{pk}(sk), r)$
- **Decryption** $\mathrm{adec}(c, sk)$
- **Correctness** $\mathrm{adec}(\mathrm{aenc}(m, pk(sk), r), sk) = m$ (equational theory)

ELGamal $(g^r, m \cdot pk^r)$ (exponentiation is fully abstracted away)

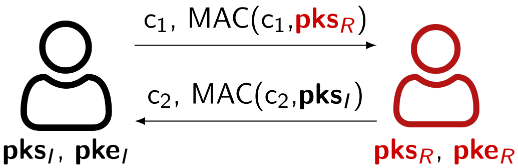
Refined Equational Theories and Application to Protocols using Mix-Nets

1^{st} Contribution

- Proposed refined modeling of several cryprtographic primitives
- Application to protocols using Mix-Nets
- (re)Discover attacks missed in previous symbolic analysis

Transferable, Auditable and Anonymous Ticketing Protocol	ASIACCS 24
Automated Discovery of Subtle Attacks on Protocols Using Mix-Nets	USENIX 24
A Unified Symbolic Analysis of WireGuard	NDSS 24
A Tale of Two Worlds, a Formal Story of WireGuard Hybridization	USENIX 25
Formal Analysis of SDNsec: Payload, Route Integrity and Accountability	ASIACCS 25
Secure and Verifiable Coercion-Resistant Electronic Exam	

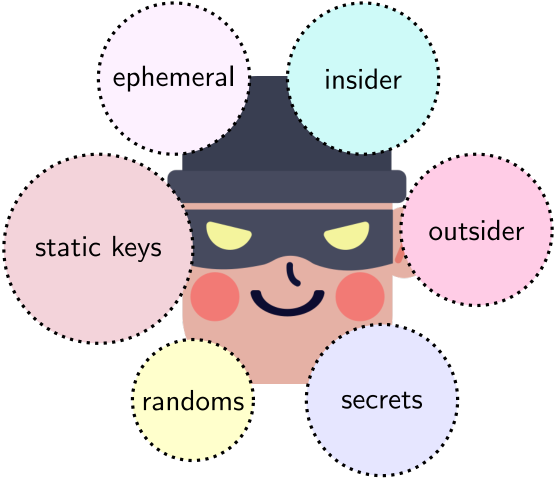
WireGuard's First Messages



Attack on Anonymity

- Public static keys used to compute the MAC
- WireGuard's designer assumed attacker can not access public static keys

Adversary Model



Adversary Model

- Consider all possible compromise cases!
- 5 keys $\implies 2^5 = 32$ possible compromise cases
- 12 keys $\implies 2^{12} = 4096$
- A need for a **methodology**!
- Minimal models to **break** security (offensive models)
- Minimal models to **guarantee** security (defensive models)

Derived Adversary Models: Application to WireGuard, PQ-WireGuard, and Hybrid-WireGuard

2^{nd} Contributions

- Derive all minimal defensive and offensive models
- WireGuard, PQ-WireGuard, PQ-WireGuard*, and Hybrid-WireGuard

Transferable, Auditable and Anonymous Ticketing Protocol	ASIACCS 24
Automated Discovery of Subtle Attacks on Protocols Using Mix-Nets	USENIX 24
A Unified Symbolic Analysis of WireGuard	NDSS 24
A Tale of Two Worlds, a Formal Story of WireGuard Hybridization	USENIX 25
Formal Analysis of SDNsec: Payload, Route Integrity and Accountability	ASIACCS 25
Secure and Verifiable Coercion-Resistant Electronic Exam	

Automated Symbolic Tools

- Symbolic model
- **Input** Protocol model + security property
- **Output** verified, falsified, non-termination, cannot decide

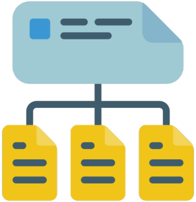
Automated Symbolic Tools

	Tamarin	**ProVerif**	**Deepsec**
Soundness	✓	✓	✓
Completeness	✓*	✗	✓
Unbounded Sessions	✓	✓	✗
Trace Properties	✓	✓	✗
Equivalence Properties	✓	✓	✓

✓* only on trace mode

Sapic$^+$

- **Unifies** the use of ProVerif, Tamarin, and DeepSec
- From 1 model, 3 models
- **Soundness** of models
- Benefits from the **strength** of each tool



An Example of Sapic$^+$ Models

<table>
<tr>
<td>

```
builtins: diffie-hellman

process:
  new x; new y; new z;
  (!out(<('g'^x)^y, ('g'^y)^x, (('g'^z)^x)^y>)
| (!in(<A, B, C>);
  if (not(A = 'g') & not(B = 'g') & not(C = 'g'))
  then event Reach(A, B, C)))

lemma Test:
  exists-trace
  "Ex A B C #i. Reach(A, B, C)@i"
```

</td>
<td>

```
builtins: diffie-hellman

process:
  new ~x; new ~y; new ~z;
  (!out(<('g'^~x)^~y, ('g'^~y)^~x, (('g'^~z)^~x)^~y>)
| (!in(<A, B, C>);
  if (not(A = 'g') & not(B = 'g') & not(C = 'g'))
  then event Reach(A, B, C)))

lemma Test:
  exists-trace
  "Ex A B C #i. Reach(A, B, C)@i"
```

</td>
</tr>
<tr>
<td>4 Tamarin rules</td>
<td>4 Tamarin rules</td>
</tr>
<tr>
<td>Timeout after 2 hours!</td>
<td>

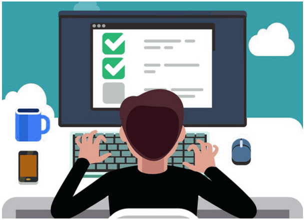
```
processing time:  0.64s
Test (exists-trace):  verified (3 steps)
```

</td>
</tr>
</table>

The Remark! Protocol

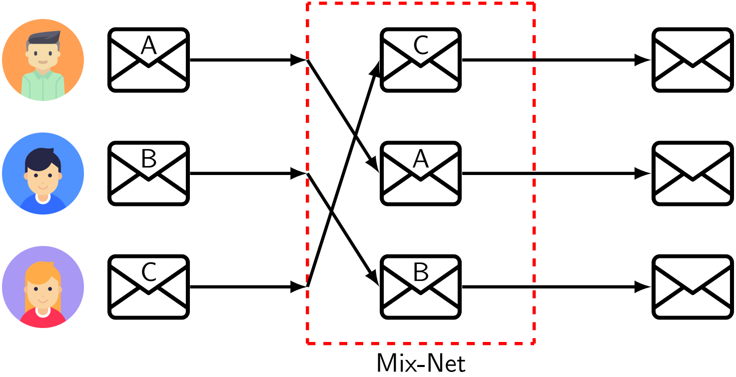
Remark! (Giustolisi et al. 2014)

- An **e-exam** protocol
- **Anonymity** of the candidates during examination (impartiality)
- **Anonymity** of the examiners (avoid coercion)
- Based on **Exponentiation-Mixnet**!



Mix-Networks

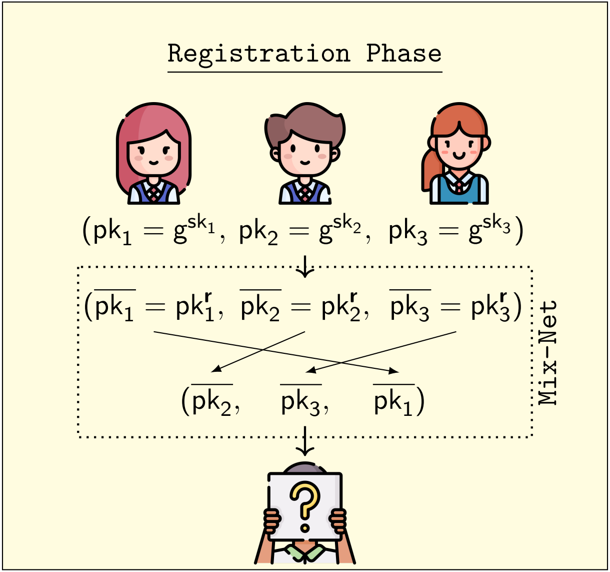
- Mix-Networks were introduced by Chaum in 1981.
- **Purpose:** Hiding the **correspondence** between its input and output!



Exponentiation Mix-Nets (Haenni et al. 2011)

- **Input:** List of ElGamal public keys
- **Output:** List of **anonymized** ElGamal public keys
- Anonymized keys used by the candidates to sign answers (Remark! Protocol)

Exponentiation Mix-Net in Remark!



Formal Analysis using ProVerif

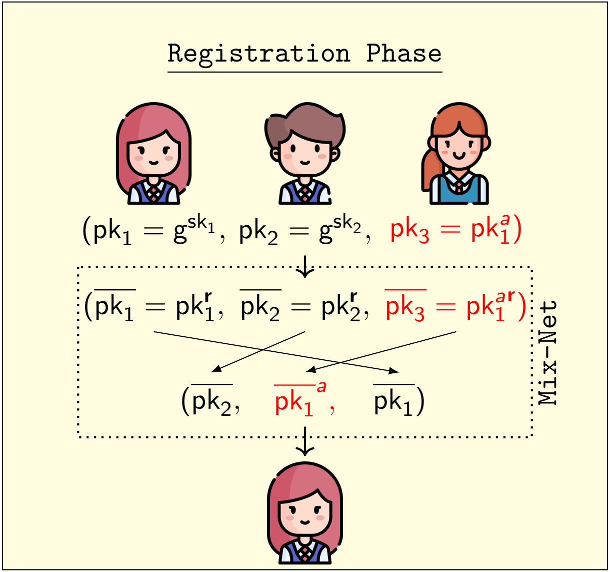
Formal Analysis of Remark! Protocol (Dreier et al. 2014)

- Analysis using ProVerif
- Candidates' anonymity ✓
- Examiners' anonymity ✓

ElGamal: $\mathrm{dec}(\mathrm{enc}(m, \mathrm{pub}(pk(k), rce), r), \mathrm{priv}(k, exp(rce))) = m$

(abstract exponentiation)

Attack on Exponentiation MIX-NET (Amin *et al.* 2022)



Attack on Exponentiation MIX-NET (Amin *et al.* 2022)

- Attack found **manually**
- ZKPS as a fix: proving possession of the secret key
- Can't this attack be found with a **symbolic tool**?

Refined Equational Theories

Primitive	Equation
Exponentiation	$\mathsf{exp}(\mathsf{exp}(g, x), y) = \mathsf{exp}(\mathsf{exp}(g, y), x)$ $\mathsf{exp}(\mathsf{exp}(\mathsf{exp}(g, x), y), z) = \mathsf{exp}(\mathsf{exp}(\mathsf{exp}(g, x), z), y)$
ElGamal Encryption	$\mathsf{dec}(\mathsf{enc}(m, X, \mathsf{exp}(X, s), r), X, s) = m$
ElGamal Signature	$\mathsf{checksign}(\mathsf{sign}(m, X, s), X, \mathsf{exp}(X, s)) = m$
Strong ZKP	$\mathsf{ck}(\mathsf{szkp}(A, g, x), g, \mathsf{exp}(g, x), \mathsf{Hash}(g, \mathsf{exp}(g, x), A)) = \mathsf{true}$
Weak ZKP	$\mathsf{ck}(\mathsf{wzkp}(A, X, x), X, \mathsf{exp}(X, x), \mathsf{Hash}(A)) = \mathsf{true}$

Applications

Protocol	ZKP	Property	Result	Time
Remark! e-exam (Giustolisi *et al.* 2024)	without	Anonymous Marking	✗	3 m 16 s
		Anonymous Examiner	✗	4 m 19 s
	weak	Anonymous Marking	✗	9 m 35 s
		Anonymous Examiner	✗	9 m 23 s
	strong	Anonymous Marking	✓	11 s
		Anonymous Examiner	✓	7 s
Haenni e-voting (Haenni *et al.* 2011)	without	Vote Privacy	✗	4 m 35 s
	weak		✗	9 m 35 s
	strong		✓	14 s
Crypto Santa (Y.A. Ryan 2015)	weak	Anonymous Shuffling	✗	4 m 6 s
	strong		✓	9 s

The Needham-Schroeder Public key Protocol

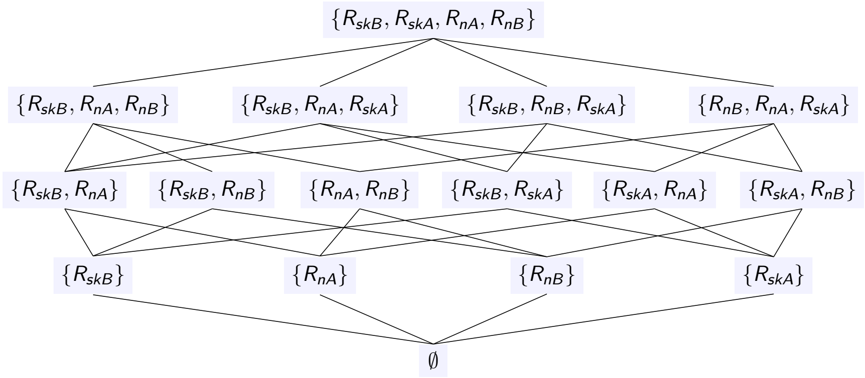
$$\begin{array}{lcll} A & \rightarrow & B & : \quad \mathrm{enc}((nA, \mathrm{pk}(skA)), \mathrm{pk}(skB)) \\ B & \rightarrow & A & : \quad \mathrm{enc}((nA, nB), \mathrm{pk}(skA)) \\ A & \rightarrow & B & : \quad \mathrm{enc}(nB, \mathrm{pk}(skB)) \end{array}$$

The Needham-Schroeder Public key Protocol

$$\begin{array}{lllll} \text{A} & \to & \text{B} & : & \mathsf{enc}((nA, \mathsf{pk}(skA)), \mathsf{pk}(skB)) \\ \text{B} & \to & \text{A} & : & \mathsf{enc}((nA, nB), \mathsf{pk}(skA)) \\ \text{A} & \to & \text{B} & : & \mathsf{enc}(nB, \mathsf{pk}(skB)) \end{array}$$

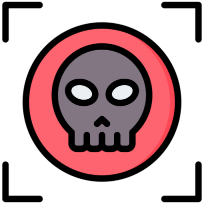
- 2 keys (skA , skB)
- 2 nonces (nA , nB)

Lattice of adversary models ordered by set inclusion for the Needham-Schroeder protocol

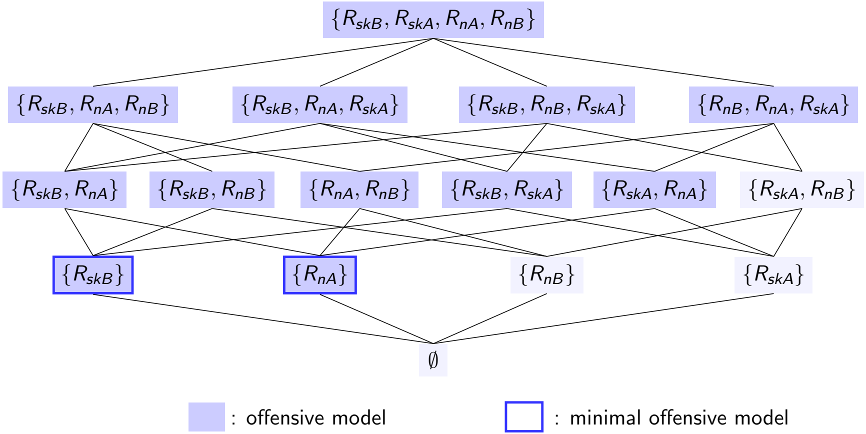


Offensive Adversary Model

- If compromise *skB* and *nB*, then the agreement ✗
- If compromise *nB*, then the agreement ✓
- If compromise *skB*, then the agreement ✗
  (**minimal offensive model**)



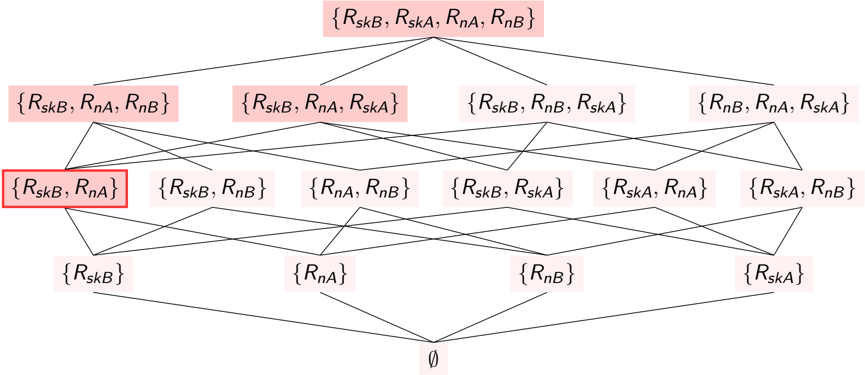
Lattice of adversary models ordered by set inclusion for the Needham-Schroeder protocol



Defensive Model

- If *skB* and *nA* not compromised, then the agreement ✓
- *skB* and *nA* is a **minimal defensive model**

Finding minimal defensive models



: defensive model   : minimal defensive model

Security Formula

Definition (Security Formula)

Given a protocol model $\mathcal{P}$, a set of atomic capabilities Γ and a security property φ, a *security formula* is the logical disjunctions of all the minimal offensive adversary models $\mathcal{D}_{i\mathcal{P},\Gamma,\varphi}$, defined as:

$$\mathcal{O}_{1\mathcal{P},\Gamma,\varphi} \vee \ldots \vee \mathcal{O}_{k\mathcal{P},\Gamma,\varphi}$$

where k is the number of all minimal defensive models.

Security Formulas from offensive models

Theorem

The disjunction of all non-empty minimal offensive models yield a security formula:

$$\bigvee_{j=1}^{k} \mathcal{O}_{j\mathcal{P},\Gamma,\varphi} = \bigwedge_{i=1}^{k'} \mathcal{D}_{i\mathcal{P},\Gamma,\varphi}$$

where k and k' are the number of all minimal non-empty offensive adversary models and all non-empty minimal defensive models respectively.

Security Formulas

- Protocol model
- Security property
- Attacker's capabilities

Security Formulas: secrecy of the session key from the Initiator's point of view

Protocol	**Security Formula**
WireGuard	$psk \wedge (s_r^c \vee e_i^c) \wedge (s_r^c \vee s_i^c \vee dh_{s_i s_r})$
PQ-WireGuard	$psk \wedge (s_r^{pq} \vee r_i) \wedge (s_r^{pq} \vee \sigma_i)$
PQ-WireGuard*	$psk \wedge (s_r^{pq} \vee r_i)$
Hybrid-WireGuard	$psk \wedge (s_r^c \vee e_i^c) \wedge (s_r^c \vee s_i^c \vee dh_{s_i s_r})$ $\bigwedge$ $psk \wedge (s_r^{pq} \vee r_i)$

Initiator's Anonymity with ProVerif (Hybrid-WireGuard)

Adversary Model	Result	Time
psk	✗	1m15s
$Sic \wedge Siq$	✗	6m25s
$Sic \wedge Rr$	✗	11m47s
$Src \wedge Srq$	✗	3m22s
$Src \wedge Ri$	✗	6m46s
$Eic \wedge Srq$	✗	3m40s
$Eic \wedge Ri$	✗	4m26s
$Erc \wedge Siq$	✗	5m12s
$Erc \wedge Rr$	✗	7m59s
$Sic \wedge Src \wedge Eic \wedge Erc \wedge Eiq \wedge Re$	✓	9m20s
$Sic \wedge Erc \wedge Srq \wedge Eiq \wedge Ri \wedge Re$	✓	9m19s
$Src \wedge Eic \wedge Siq \wedge Eiq \wedge Rr \wedge Re$	✓	9m09s

Agreement Properties with Tamarin (Hybrid-WireGuard)

	Security Formula
Agreement on InitHello	$psk \wedge (dhsisr \vee Sic \vee Src)$
Agreement on Rechello	$psk \wedge (Srq \vee Ri) \wedge (Src \vee Eic) \wedge (dhsisr \vee Sic \vee Src)$
Agreement on Confirm	$psk \wedge (Siq \vee Rr) \wedge (Sic \vee Erc) \wedge (dhsisr \vee Sic \vee Src)$

Lemma	**Heuristic(p)**	**Heuristic(s)**	**Tactic(s)**	**Oracle(s)**
Agreement on InitHello	299	152	26	22
Agreement on Rechello	696	236	✗	54
Agreement on Confirm	∞	∞	✗	90

∞: timeout after 5 hours ✗: unable to find tactic

Sapic$^+$: Experience feedback and lessons learned

- Outputs' placement matters!

<table>
<tr>
<td>

```
process:
new kemltkI;out(pk(kemltkI));
new kemltkR;out(pk(kemltkR));
new ldhI;out('g'^ldhI);
new ldhR;out('g'^ldhR)
```

</td>
<td>

```
process:
new kemltkI;
new kemltkR;
new ldhI;
new ldhR;
out(<pk(kemltkI), 'g'^ldhI, pk(kemltkR), 'g'^ldhR>)
```

</td>
</tr>
<tr>
<td>4 Tamarin rules!</td>
<td>1 Tamarin rule!</td>
</tr>
</table>

Sapic$^+$: Experience feedback and lessons learned

- How to model private channel matters!

<pre>functions: chp/0[private]
 	
process:
  new skI;
  (out(chp, skI) | in(chp, x))</pre>
	<pre>process:
	  new chp;
	  new skI;
	  (out(chp, skI) | in(chp, x))</pre>

8 Tamarin rules!	**3 Tamarin rules!**

Summary of Contributions

Transferable, Auditable and Anonymous Ticketing Protocol	ASIACCS 24
Automated Discovery of Subtle Attacks on Protocols Using Mix-Nets	USENIX 24
A Unified Symbolic Analysis of WireGuard	NDSS 24
A Tale of Two Worlds, a Formal Story of WireGuard Hybridization	USENIX 25
Formal Analysis of SDNsec: Payload, Route Integrity and Accountability	ASIACCS 25
Secure and Verifiable Coercion-Resistant Electronic Exam	

The Decisional Diffie-Hellman (**DDH**) assumption in ProVerif

Equational Theory	$(g^a, g^b, g^{ab}) \approx_l (g^a, g^b, g^c)$
$(g^x)^y = (g^y)^x$	true
$(g^x)^y = (g^y)^x$ $((g^x)^y)^z = ((g^x)^z)^y$	cannot be proved

$$\text{diff } [(g^a, g^b, g^{ab}), (g^a, g^b, g^c)\ ]$$
$$\text{diff } [(g^a, g^b, (g^{abx})^y), (g^a, g^b, (g^{cx})^y)\ ]$$
$$\text{diff } [(g^a, g^b, (g^{aby})^x), (g^a, g^b, (g^{cy})^x)\ ]$$

ElGamal public key encryption

Equation	**Strength**	**Weaknesses**
$\mathrm{dec}(\mathrm{enc}(m, X, X^s, r), X, s) = m$	More precise	Cannot decrypt knowing only r
		Cannot be used in TAMARIN

ElGamal public key encryption

Equation	Strength	Weaknesses
$\mathrm{dec}(\mathrm{enc}(m, X, X^s, r), X, s) = m$	More precise	Cannot decrypt knowing only r
		Cannot be used in TAMARIN

Model	Strength
$(\mathrm{g}^r, \mathrm{senc}(m, (g^x)^r))$	More precise
	Can be used in TAMARIN and PROVERIF
	Can decrypt knowing only r

Key Encapsulation Mechanism

- Public key encryption $\mathrm{aenc}(ss, \mathrm{pk})$
- Ciphertexts **bind** to keys
- Ciphertexts **bind** to shared secrets

Analyze PQ-WireGuard and Hybrid-WireGuard with different binding assumptions.

Stateless vs stateful protocols

For WireGuard, PQ-WireGuard and Hybrid-WireGuard

- Keys are never updated
- State disruption attacks not modeled

Re-analyze considering stateful models

Thank you for your attention!

Thank you for your attention!

Standard Equational Theories

Primitive	**Equation**
Exponentiation	$\mathrm{exp}(\mathrm{exp}(g, x), y) = \mathrm{exp}(\mathrm{exp}(g, y), x)$
ElGamal Encryption	$\mathrm{dec}(\mathrm{enc}(m, \mathrm{pk}(sk), r), sk) = m$
Digital Signature	$\mathrm{checksign}(\mathrm{sign}(m, sk), \mathrm{pk}(sk)) = m$

- $\overline{\mathrm{pk}_1} = \mathrm{g}^{\mathrm{sk}_1\,\mathbf{r}} = \mathrm{g}^{\mathbf{r}\,\mathrm{sk}_1}$
- $\mathrm{pk}_1^{a\,\mathbf{r}} = \mathrm{g}^{\mathrm{sk}_1\,a\,\mathbf{r}} = \mathrm{g}^{a\,\mathrm{sk}_1\,\mathbf{r}} \neq \mathrm{g}^{\mathrm{sk}_1\,\mathbf{r}\,a} = \overline{\mathrm{pk}_1}^{a}$
- $\mathrm{exp}(g, x) \neq \mathrm{pk}(x)$

Sapic$^+$: Experience feedback and lessons learned

- How to express conditionals matters!

`let main(kemltkI, kemltkR, kemltkC) =`
`if (kemltkI = kemltkR) then`
`if (kemltkI = kemltkC) then`
`if (kemltkR = kemltkC) then`
`  (`
`   out(<pk(kemltkI), pk(kemltkR), pk(kemltkC)>)`
`  )`
`process:`
`  new kemltkI;`
`  main(kemltkI, kemltkI, kemltkI)`
	`let main(kemltkI, kemltkR, kemltkC) =`
	`if (kemltkI = kemltkR) & (kemltkI = kemltkC) & (kemltkR = kemltkC) then`
	`  (`
	`   out(<pk(kemltkI), pk(kemltkR), pk(kemltkC)>)`
	`  )`
	`process:`
	`  new kemltkI;`
	`  main(kemltkI, kemltkI, kemltkI)`

4 Tamarin rules!	**1 Tamarin rule!**
DeepSec ✓	**DeepSec ✗**

Sapic^{+}: Experience feedback and lessons learned

- The more events, the more rules!

<pre>process:
    event Dummy1();
    event Dummy2();
    event Dummy3()</pre>
	<pre>process:
	    event Dummy1();
	    event Dummy2();
	    event Dummy3();
	    event Dummy4();
	    event Dummy5()</pre>

3 Tamarin rules!	**5 Tamarin rules!**