State-Separating Proofs
A Reduction Methodology for Real-World Protocols

Chris Brzuska
Antoine Delignat-Lavaud
Konrad Kohbrok
Markulf Kohlweiss
I sometimes suffer.
And the goal of this paper is to ease my suffering.
Key exchange researchers sometimes suffer.
And the goal of this paper is to ease our suffering.
Once upon a time...
...I was a PhD student and I worked on key exchange.

At night, when I wrote proofs, I suffered.

- a) tons of work
- b) many *seemingly* simple steps
- c) not human-verifiable.
...one of the worst proofs was to prove that Bellare-Rogaway secure key exchange protocols are composable.

Key Exchange

outputs symmetric keys that look random.

Application Protocol

Is secure with random keys
What I was really concerned about: The theorem seems so simple, why is the proof so hard?

Tried in
CCS 2011 paper, my thesis, Stephen Williams‘s thesis...

Key Exchange

Outputs symmetric keys that look random.

Application Protocol

Is secure with random keys
What is the difficulty?

Key Exchange

- Outputs symmetric keys that look random.

Application Protocol

- Is secure with random keys

...one of the worst proofs was to prove that Bellare-Rogaway security key exchange protocols are composable.
Key Exchange

Outputs symmetric keys that look random.

Application Protocol

Is secure with random keys
State is passed from one game to another, *defining* the composition is already annoying.

Key Exchange

*Outputs symmetric keys that look random.*

Application Protocol

*Is secure with random keys*
State is passed from one game to another, *defining* the composition is already annoying.

Key Exchange

Outputs symmetric keys that look random.

Multi-session & multi-instance

Application Protocol

Is secure with random keys
Key Exchange

Application Protocol

Reduction

Replace real keys with random keys
Key Exchange

Reduction

Application Protocol

Reduce to security of the application
Replace real keys with random keys.

Administrate lists in the reduction, pass on keys to the right sessions etc.
Key Exchange

Application Protocol

Reduction

Reduce to security of the application

Administrate lists in the reduction, pass on keys to the right sessions etc..
Idea 1: Move shared state to an external algorithm ("package").
Idea 2: Specify rules to compose packages

Key Exchange

Application Protocol

Key Package
Idea 2: Specify rules to compose packages

Key Exchange

Application Protocol

Key Package
The reduction is defined *automatically* and *precisely*.

How? The main focus of the paper is package composition.
Bracket operator for parallel composition

Key Exchange

Application Protocol

Circle operator for packages that interact

Key Package
Hybrid

Idea 3: Specify algebraic rules

- $A^1[1]$
- $A^1[i-1]$
- $A^0[i]$
- $A^0[i+1]$
- $A^0[n]$
Hybrid

Idea 3: Specify algebraic rules

A^{1}[1]
A^{1}[i-1]
ID
A^{0}[i+1]
A^{0}[n]
A^{0}[i]
Hybrid

Idea 4: Automatic precise description of reductions

Reduction

\[ A^1[1] \]

\[ A^1[i-1] \]

ID

\[ A^0[i+1] \]

\[ A^0[n] \]

\[ A^0[i] \]
Hybrid

Reduction

A^1[1]
A^1[i-1]
ID
A^0[i+1]
A^0[n]

Idea 4: Automatic precise description of reductions
Hybrid

$A^1[1]$  
$A^1[i-1]$  
ID  
$A^0[i+1]$  
$A^0[n]$  

$A^1[i]$
Hybrid

<table>
<thead>
<tr>
<th>$A^1[1]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^1[i-1]$</td>
</tr>
<tr>
<td>$A^1[i]$</td>
</tr>
<tr>
<td>$A^0[i+1]$</td>
</tr>
<tr>
<td>$A^0[n]$</td>
</tr>
</tbody>
</table>
Motivation: Simple steps are

a) little work
b) precise
c) human-verifiable
Usability

Mike Rosulek uses some ideas and especially similar notation in the draft of his book “The Joy of Cryptography“ which he has used in his undergraduate classes for years.
Our Hope:

a) Use method to prove TLS 1.3
b) Use method for meta-reductions
c) Make key exchange papers readable again
d) Suffer less, understand more 😊
Request/Suggestion

If you suffer from writing seemingly simple proof steps or from making them accessible to readers, check whether our notation can help.

If you have the same struggle in teaching, check whether Mike’s book can help you.
Selection of
Acknowledgements & Inspirations

• Universal composability, Ran Canetti
• Random systems, abstract crypto, constructive crypto, Ueli Maurer & Renato Renner
• miTLS, Microsoft Research & Inria Paris
• Pi-Calculus

We put existing ideas together with a focus on proofs and real-life protocol proofs in mind.
State-Separating Proofs
A Reduction Methodology for Real-World Protocols

Chris Brzuska, Antoine Delignat-Lavaud,
Konrad Kohbrok, Markulf Kohlweiss
Now on ePrint:

State-Separating Proofs
A Reduction Methodology for Real-World Protocols

Chris Brzuska, Antoine Delignat-Lavaud,
Konrad Kohbrok, Markulf Kohlweiss

Maybe, it can help you, too 😊