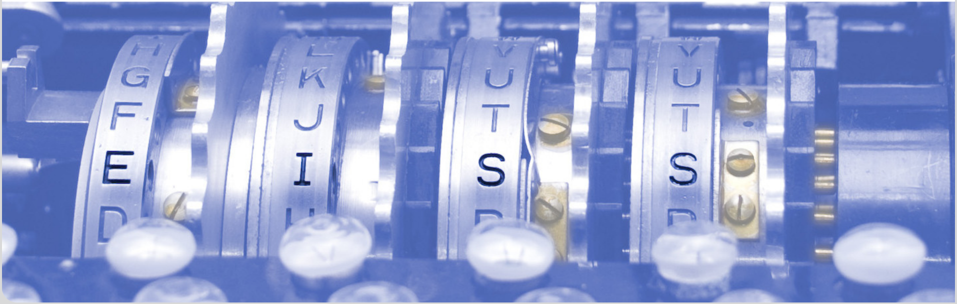
Karlsruhe Institute of Technology

Universally Composable Firewall Architectures using Trusted Hardware

16.10.2014

Dirk Achenbach, Jörn Müller-Quade, Jochen Rill

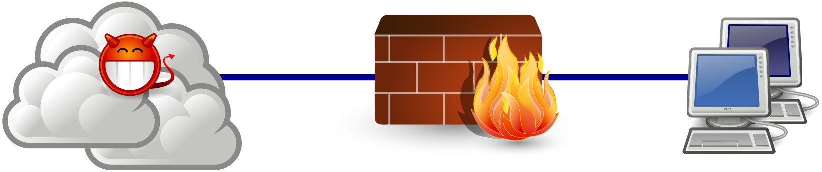
KARLSRUHE INSTITUTE OF TECHNOLOGY



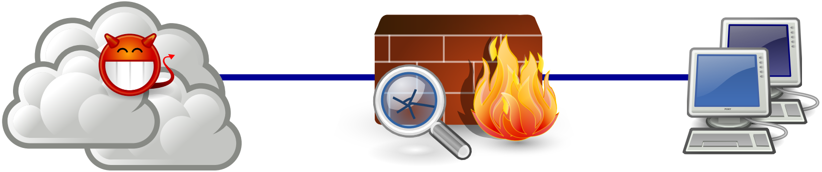
Outline

1. Malicious Firewalls
   - Concatenation of Packet Filters
   - Actively Malicious Firewalls
   - Trusted Hardware
   - Quorum Decisions
2. Analysis in the UC Framework
   - The Universal Composability Framework
   - Proving the Security of Our Approach

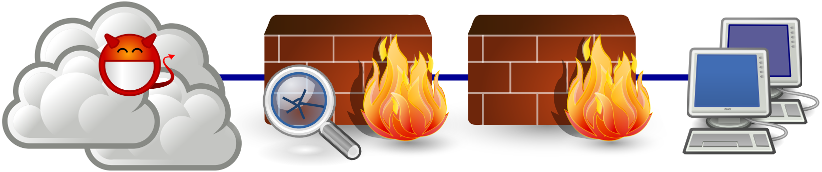
Firewalls



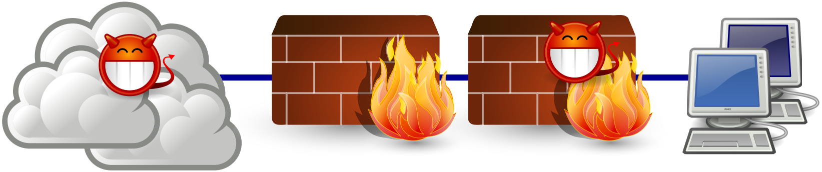
Are Firewalls Really Secure?



Just Use Two!



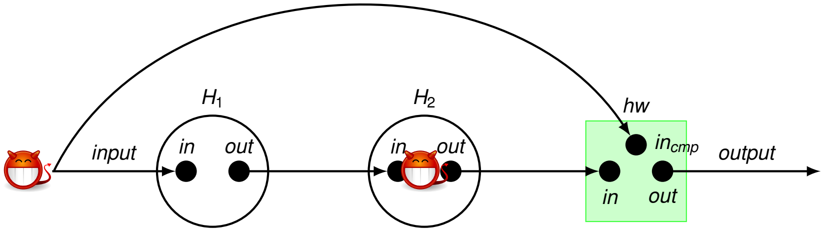
This Doesn't Work



Trusted Hardware

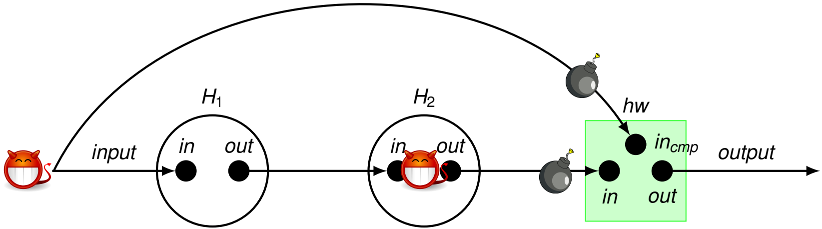
Our idea:

- Use a piece of trusted hardware
- Very simple functionality
- Not programmable, maybe even sealed
- Checks if “what goes in also comes out”

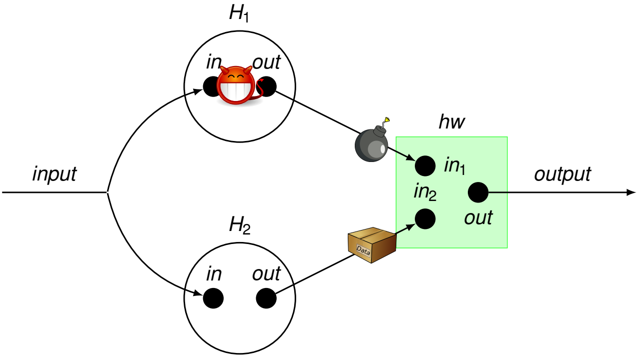


Trusted Hardware

This doesn't work either: The compromised firewall could send “evil packets” with clever timing:



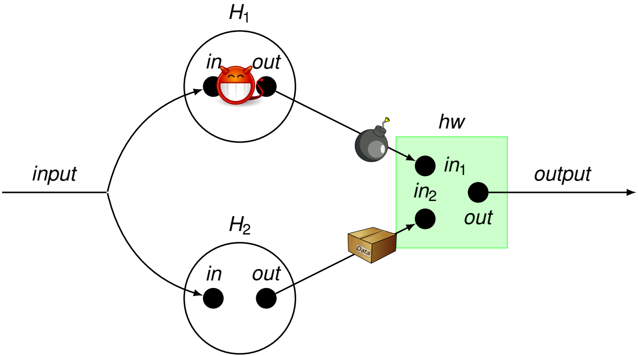
What about this approach?



Research Challenge

Rigorously analyse the security of this approach.

What about this approach?



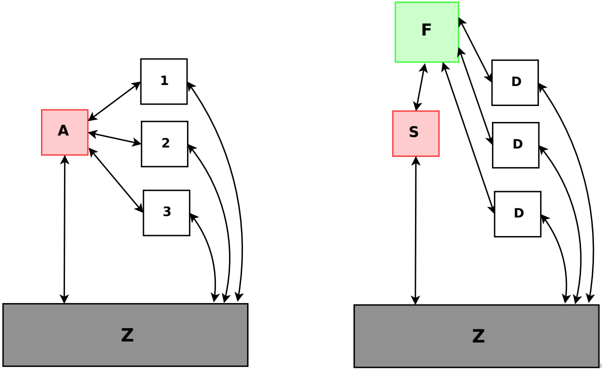
Research Challenge

Rigorously analyse the security of this approach.

The Universal Composability Framework

Karlsruhe Institute of Technology

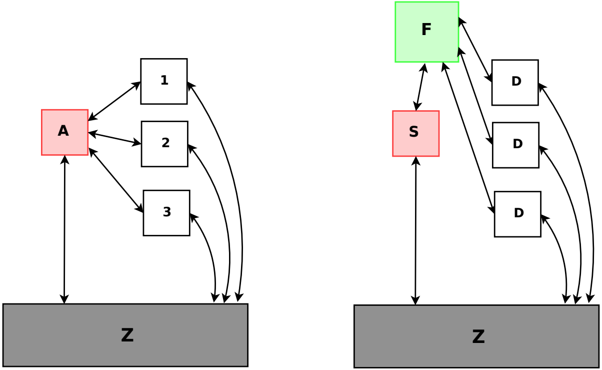
- Formal framework for the security of cryptographic protocols.
- Compare the concrete protocol with an “idealised” version.
- Simulation-based approach.



The Universal Composability Framework

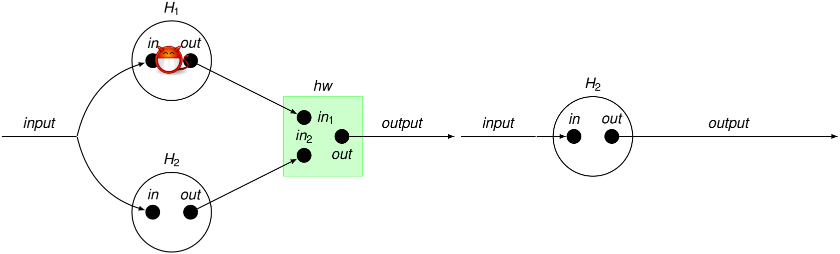
A protocol π securely realises an ideal functionality $\mathcal{F}$ if

$$\forall \mathcal{A} \exists \mathcal{S} \forall \mathcal{Z} : \text{REAL}_{\pi,\mathcal{A},\mathcal{Z}} \approx \text{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$$



The Universal Composability Framework

With this approach we need not specify what a (uncompromised) firewall actually does!

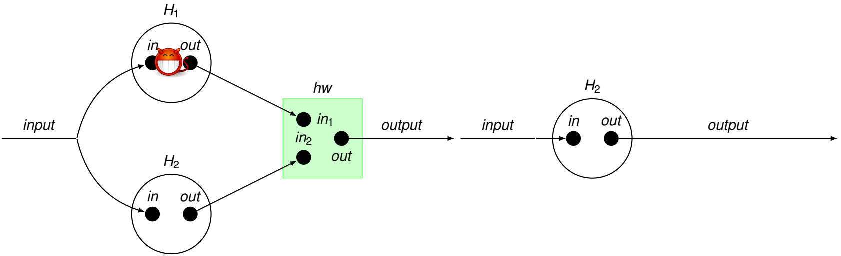


Security Intuition

"As if the compromised firewall was not there."

The Universal Composability Framework

With this approach we need not specify what a (uncompromised) firewall actually does!



Security Intuition

“As if the compromised firewall was not there.“

Caveat Emptor

The Composition Theorem makes it possible to construct secure networks from smaller components:

Theorem (Composition Theorem [1])

Let ρ, ϕ, π be protocols such that ρ uses ϕ as subroutine and π UC-emulates ϕ. Then protocol $\rho^{\phi \rightarrow \pi}$ UC-emulates ρ.

Setup Assumptions

- No non-trivial protocols can be proven secure in the "bare" model [2].
- *Setup assumptions* alleviate this problem: Common Reference Strings [1], Public-Key Infrastructures [3], Tamper-Proof Hardware [4]

Caveat Emptor

The Composition Theorem makes it possible to construct secure networks from smaller components:

Theorem (Composition Theorem [1])

Let ρ, ϕ, π be protocols such that ρ uses ϕ as subroutine and π UC-emulates ϕ. Then protocol $\rho^{\phi \to \pi}$ UC-emulates ρ.

Setup Assumptions

- No non-trivial protocols can be proven secure in the “bare” model [2].
- *Setup assumptions* alleviate this problem: Common Reference Strings [1], Public-Key Infrastructures [3], Tamper-Proof Hardware [4]

Our Setup Assumption: A Trusted Packet Comparator

An idealised description of trusted hardware hw

Keep a local cache realised as an unordered list.

Upon receiving packet p on interface i:

- If there is another input interface $j \neq i$, and a corresponding entry (j, q) with $p \equiv q$ in the cache:
  - Remove (j, q) from the cache,
  - output p.
- Otherwise, store (i, p) in the cache.

This is a much simpler functionality than that of a firewall!

Our Setup Assumption: A Trusted Packet Comparator

An idealised description of trusted hardware hw

Keep a local cache realised as an unordered list.

Upon receiving packet p on interface i:

- If there is another input interface $j \neq i$, and a corresponding entry (j, q) with $p \equiv q$ in the cache:
  - Remove (j, q) from the cache,
  - output p.
- Otherwise, store (i, p) in the cache.

This is a much simpler functionality than that of a firewall!

Security of Two Firewalls

The ideal functionality of two firewalls $\mathcal{F}_{\text{ideal}}$

- Upon receiving (input, p):
  - Ask the adversary if p should be delivered. If yes, let fw_k be the non-corrupted party; calculate $F_{\mathrm{fw}_k}(p, \mathrm{in}, s) = (p', i', s')$. Write p' to the output tape of hw, if $p' \neq \bot$ and $i' \neq \bot$. Else, do nothing. Save the new internal state s'.

This is not an absolute guarantee! We state what the adversary's capabilities "ideally" should be.

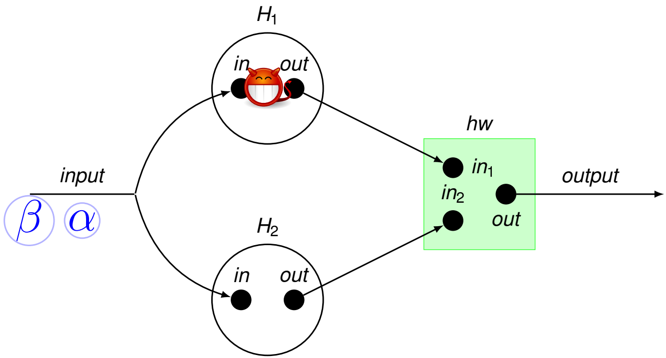
Security of Two Firewalls

The ideal functionality of two firewalls $\mathcal{F}_{\text{ideal}}$

- Upon receiving (input, p):
  - Ask the adversary if p should be delivered. If yes, let fw_k be the non-corrupted party; calculate $F_{\mathrm{fw}_k}(p, \mathrm{in}, s) = (p', i', s')$. Write p' to the output tape of hw, if $p' \neq \perp$ and $i' \neq \perp$. Else, do nothing. Save the new internal state s'.

This is not an absolute guarantee! We state what the adversary's capabilities "ideally" should be.

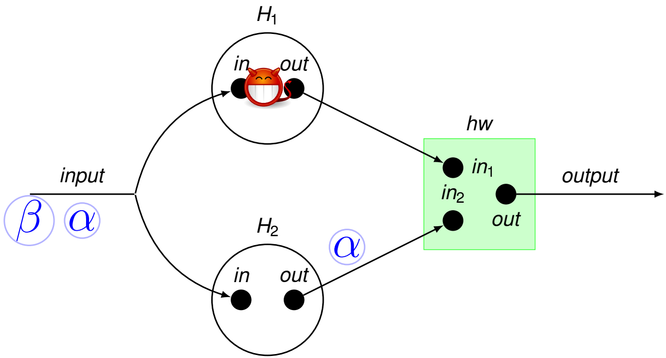
No!



No!

The adversary can *re-order packets* at will!

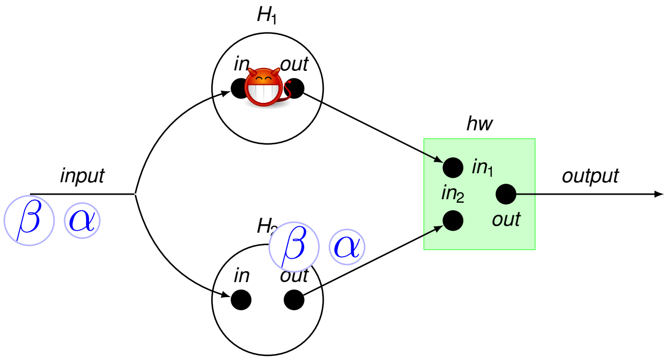
No!



No!

The adversary can *re-order packets* at will!

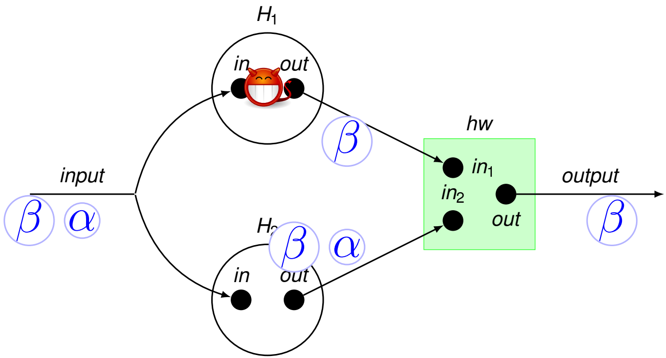
No!



No!

The adversary can *re-order packets* at will!

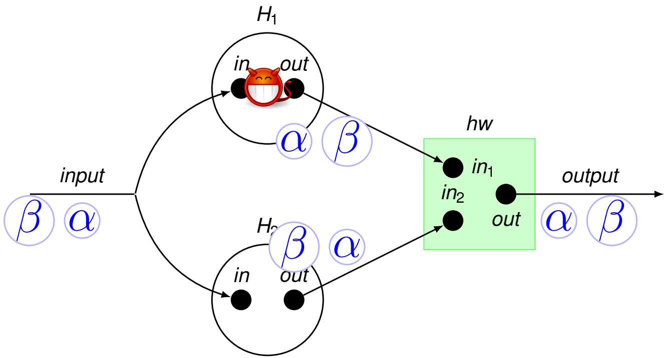
No!



No!

The adversary can *re-order packets* at will!

No!



No!

The adversary can *re-order packets* at will!

The ideal functionality of the two-firewall approach

The ideal functionality of two firewalls with packet reordering $\mathcal{F}_{\text{ideal}_2}$

- Upon receiving (input, p): Let w.l.o.g fw_1 be the non-corrupted party; calculate $F_{\mathrm{fw}_1}(p, \mathrm{in}, s) = (p', i', s')$. If $p' \neq \bot$ and $i' \neq \bot$, save p' in an indexed memory structure m at the next free index. Save new internal state s'. Give p to the adversary.
- Upon receiving $(\text{deliver}, j)$ from the adversary: If $m[j]$ contains a valid packet, write $(\mathrm{out}, m[j])$ to the output tape of hw and clear $m[j]$; else do nothing.

This *explicitly* models the adversary's ability to schedule packets!

The ideal functionality of the two-firewall approach

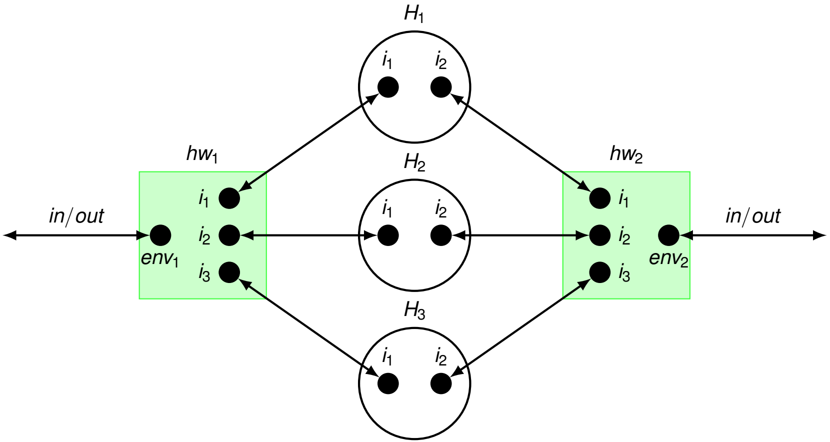
Karlsruhe Institute of Technology

The ideal functionality of two firewalls with packet reordering $\mathcal{F}_{\text{ideal}_2}$

- Upon receiving (input, p): Let w.l.o.g fw_1 be the non-corrupted party; calculate $F_{\mathrm{fw}_1}(p, \mathrm{in}, s) = (p', i', s')$. If $p' \neq \bot$ and $i' \neq \bot$, save p' in an indexed memory structure m at the next free index. Save new internal state s'. Give p to the adversary.
- Upon receiving $(\text{deliver}, j)$ from the adversary: If $m[j]$ contains a valid packet, write $(\mathrm{out}, m[j])$ to the output tape of hw and clear $m[j]$; else do nothing.

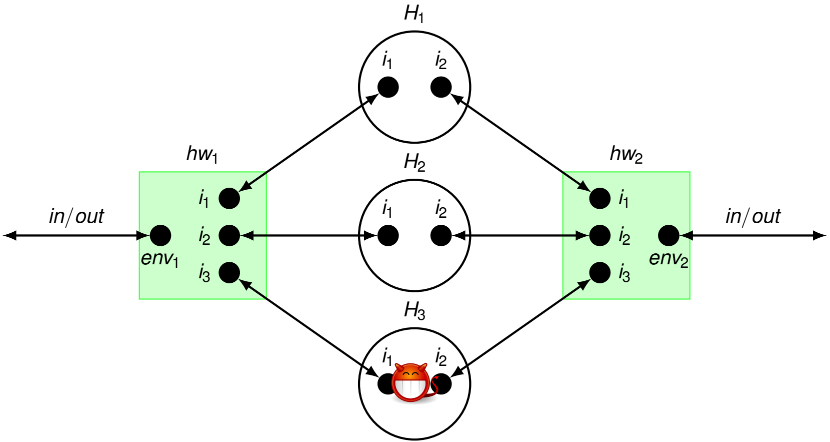
This *explicitly* models the adversary's ability to schedule packets!

What About Availability?

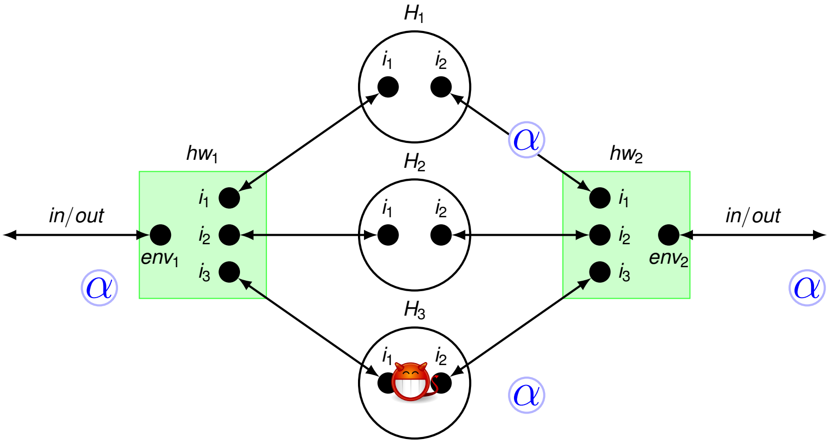


Is this as secure as the 2-firewall approach?

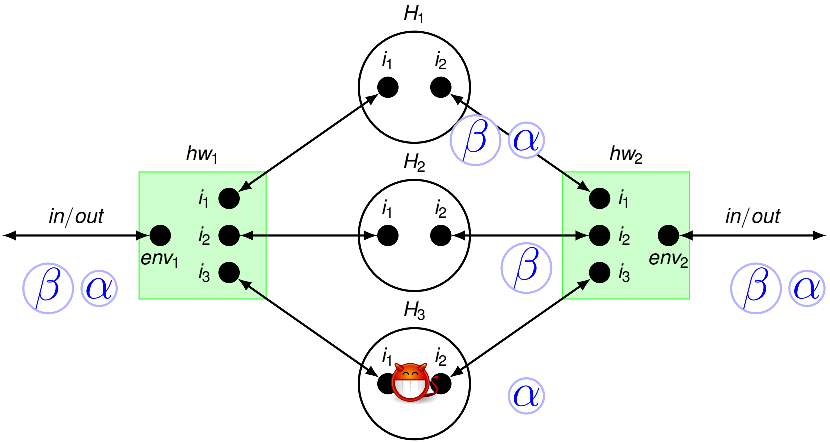
Packet Duplication Attack



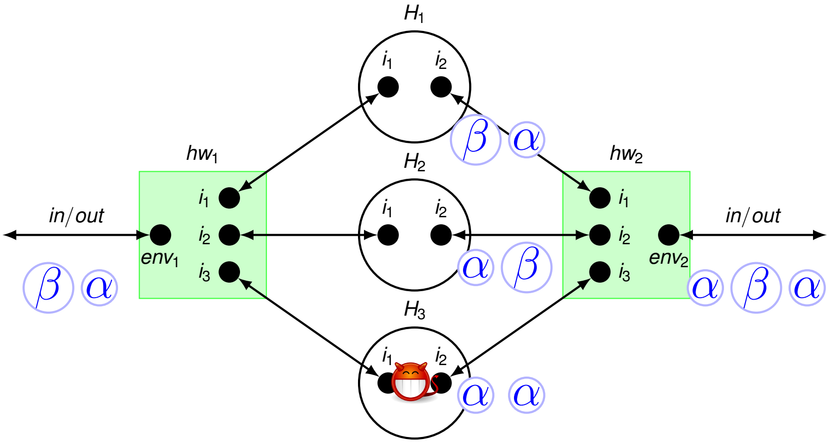
Packet Duplication Attack



Packet Duplication Attack



Packet Duplication Attack



Fix: "Packet Accounting"

Keep a local cache for each incoming interface realised as an unordered list. Upon receiving packet p on interface i:

- Check if the cache of interface i contains an entry $-q$ with $p \equiv q$. If so, delete $-q$ and halt.
- Check if there exists an interface $j \neq i$ with an entry q with $p \equiv q$ in the cache of that interface:
  - Remove q from the cache,
  - output p,
  - add an entry $-p$ to the cache of all other interfaces k with $k \neq i$ and $k \neq j$.
- Otherwise, store p in the cache of interface i.

Conclusion

Karlsruhe Institute of Technology

- We investigated the idea of actively compromised firewalls.
- Goal: Combine several candidate implementations into one secure firewall.
- Serial concatenation does not work, even with trusted hardware.
- The quorum does work.
- Future Work: Model availability in UC, Bounded Queues.

References

- R. Canetti, "Universally composable security: a new paradigm for cryptographic protocols," in *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, oct. 2001.
- R. Canetti and M. Fischlin, "Universally composable commitments," in *Advances in Cryptology–Crypto 2001*. Springer, 2001, pp. 19–40.
- B. Barak, R. Canetti, J. B. Nielsen, and R. Pass, "Universally composable protocols with relaxed set-up assumptions," in *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*. IEEE, 2004, pp. 186–195.
- J. Katz, "Universally composable multi-party computation using tamper-proof hardware," in *Advances in Cryptology – EUROCRYPT 2007*, ser. Lecture Notes in Computer Science, M. Naor, Ed. Springer Berlin Heidelberg, 2007, vol. 4515, pp. 115–128. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72540-4_7