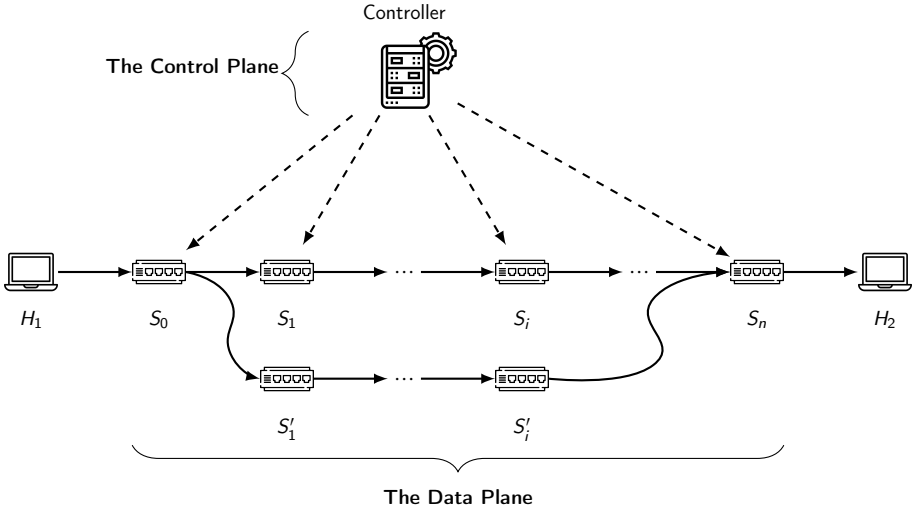
Formal Analysis of SDNsec: Attacks and Corrections for Payload, Route Integrity and Accountability

Ayoub Ben Hassen Pascal Lafourcade Dhekra Mahmoud Maxime Puys

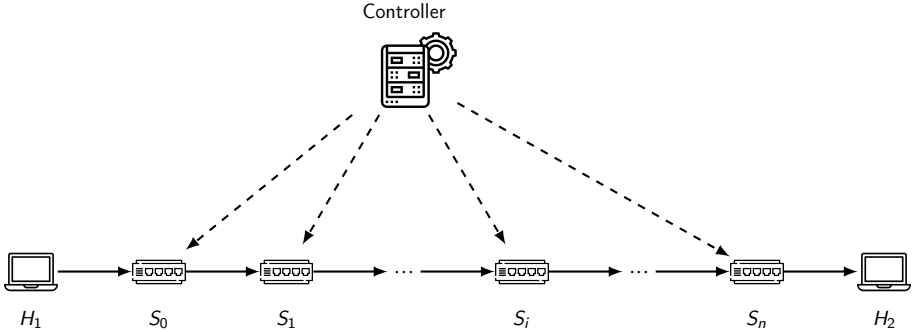
Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Etienne, LIMOS

AsiaCCS'2025
August 29th, 2025

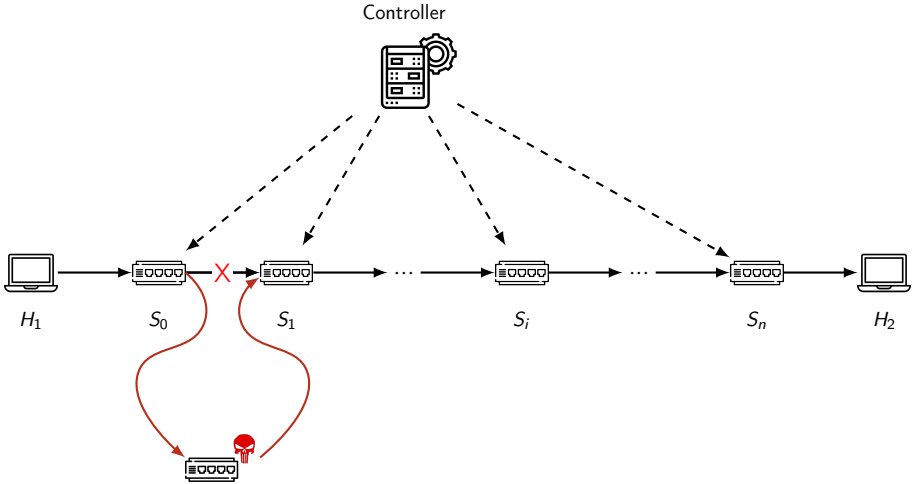
SDN Networks and Routing



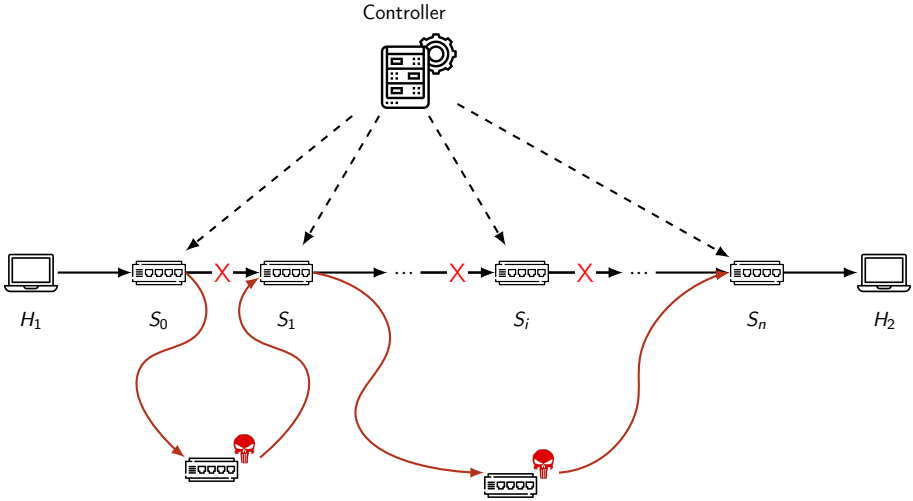
Attacks against SDN Routing



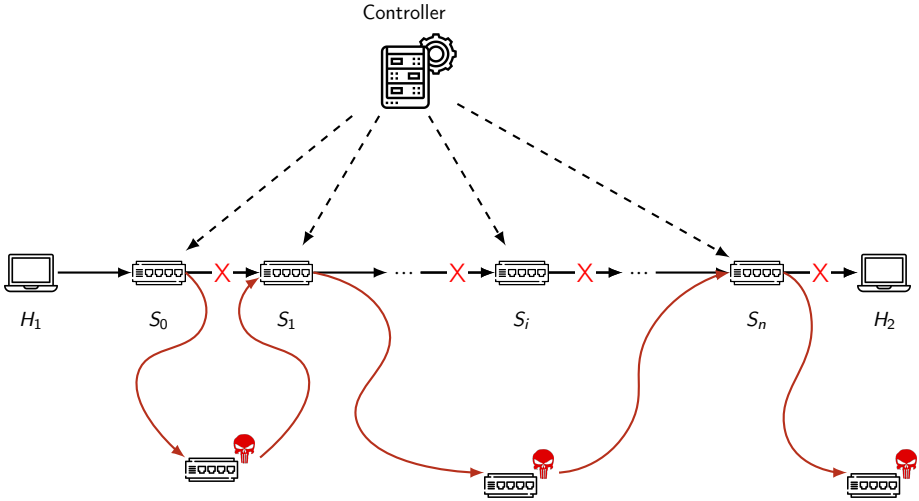
Attacks against SDN Routing



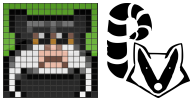
Attacks against SDN Routing



Attacks against SDN Routing



Formal Verification of Cryptographic Protocols



Crucial to **verify** that protocols guarentee security properties!

Numerous tools exist (e.g.: Tamarin [MSCB13] or ProVerif [Bla01]):

- **Formally** verify the protocol **in presence of attacker** (Dolev-Yao [DY81]).
- Check secrecy, authentication, observational equivalence, and other trace properties.

Research Question

How can we **model and verify SDN security protocols** to check if they guarantee **route integrity**, **payload integrity**, and **accountability**?

Related Works

Solution	Cryptography	Misrouting Detection	Payload Integrity
VeriFlow [KZZ+12]	X	X	X
Avant-Guard [SYG13]	X	X	X
FortNox [YFT+12]	X	X	X
Sphinx [DPMM15]	X	X	X
FlowMon [KF15]	X	X	X
WedgeTail [SKJ17]	X	✓	X
FOCES [ZXY+20]	X	✓	X
WhiteRabbit [SKOY19]	X	✓	X
REV [ZWZL20]	✓	✓	X
SDNsec [SPL+16]	✓	✓	X

✓: **Property claimed** X: **Property absent**

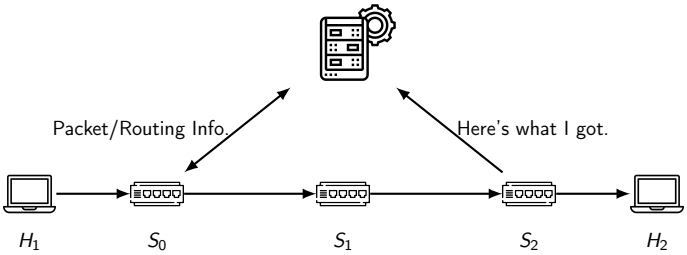
Modeling SDN Protocols

- 1x Controller
- 1x Ingress switch
- Nx Core switches
- 1x Egress switch
- (N+2)x Private channels between controller and each switch
- 1x Source host
- 1x Destination host

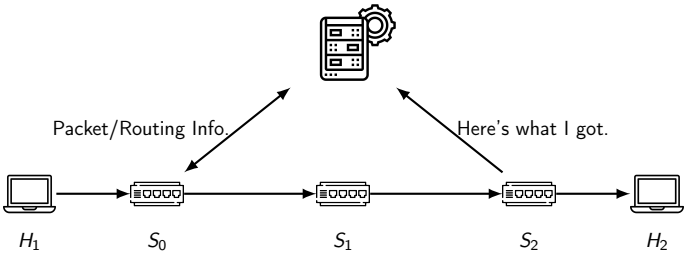
⇒ Attacker completely controls the network and can freely choose the topology **but cannot attack** between source host and ingress switch (resp. destination host and egress switch).

⇒ Controller chooses the genuine route and sends it to the switches according to the protocol.

Modeling Security Properties for SDN Protocols

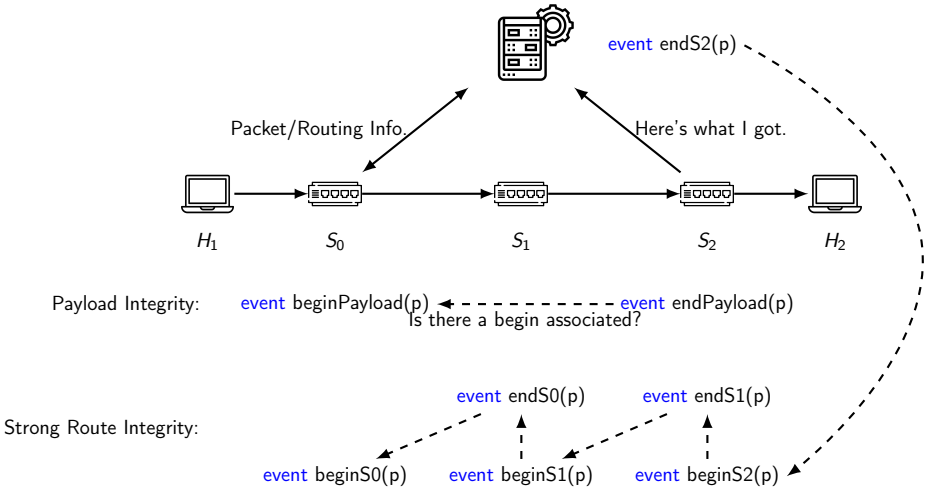


Modeling Security Properties for SDN Protocols

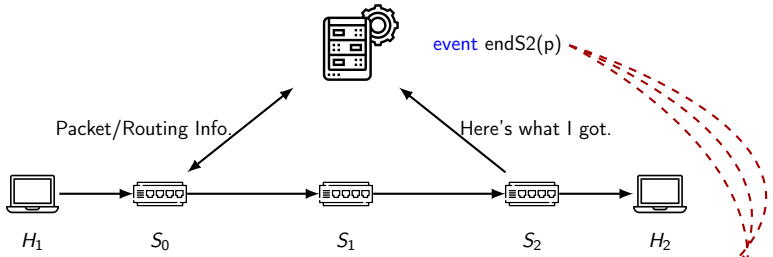


Payload Integrity: event beginPayload(p) ◄- - - - - - - - - - - - event endPayload(p)
Is there a begin associated?

Modeling Security Properties for SDN Protocols



Modeling Security Properties for SDN Protocols



Payload Integrity: event beginPayload(p) ← event endPayload(p)

Is there a begin associated?

event endS0(p) event endS1(p)

Weak Route Integrity:

event beginS0(p) event beginS1(p) event beginS2(p)

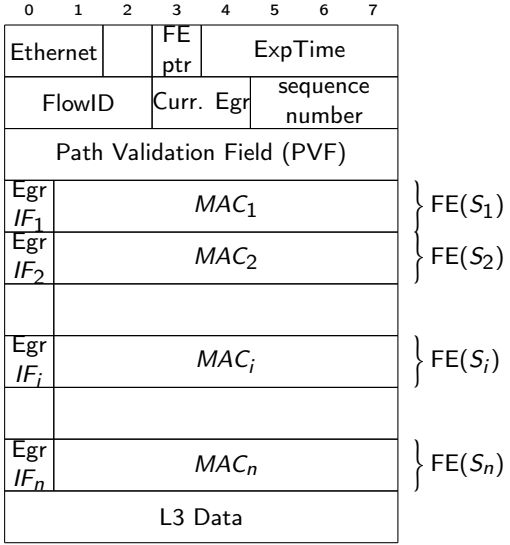
SDNsec [SPL+16]

Preemprive check by each switch:

$$B = FlowID \parallel ExpTime$$
$$FE(S_i) = egr(S_i) \parallel MAC(S_i)$$
$$MAC(S_i) = MAC_{K_i}(egr(S_i) \parallel FE(S_{i-1}) \parallel B)$$

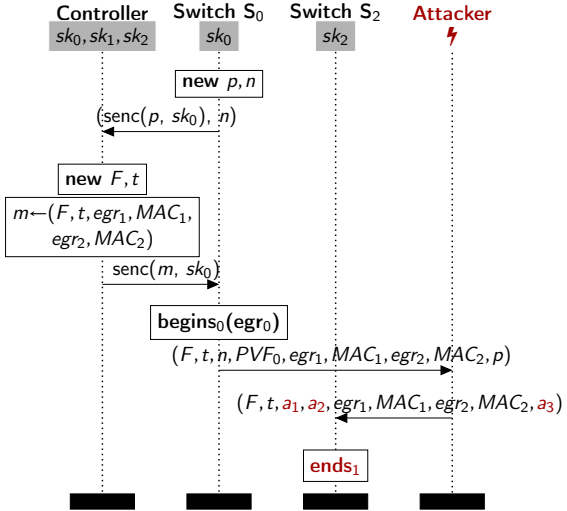
Retro-active check by the controller:

$$C = FlowID \parallel SeqNo$$
$$PVF(S_0) = MAC_{K0}(C)$$
$$PVF(S_i) = MAC_{K_i}(PVF(S_{i-1}) \parallel C)$$



An Attack on Strong Route Integrity against SDNsec

Retrospectively a poor candidate as extremely unsecure:



Proposed Correction and Results

$$B = FlowID \parallel ExpTime$$
$$FE(S_i) = egr(S_i) \parallel MAC(S_i)$$
$$MAC(S_i) = MAC_{K_i}(egr(S_i) \parallel FE(S_{i-1}) \parallel$$
$$B \parallel \mathrm{H}(\mathrm{p} \parallel \mathrm{PVF}(\mathrm{S_{i-1}}) \parallel \mathrm{SeqNo_{i-1}}))$$

	Payload Integrity	Route Integrity				Accountability	
		Local RI	Trans. RI	Weak RI	Strong RI	Soundness	Completeness
SDNsec [SPL+16]	UNSAFE	SAFE	UNSAFE	UNSAFE	UNSAFE	SAFE	UNSAFE
SDNsec⋆	SAFE	SAFE	SAFE	SAFE	SAFE	SAFE	SAFE

Conclusion

- Formal analysis of the SDNsec protocol, focusing on three key security properties: payload integrity, route integrity, and accountability.
- Implementation with RYU [RYU14] and Mininet [GNN+84].

- Formal modeling on SDN protocols,
- Formal definitions of these security properties,
- Future work: Verify other SDN security protocols!

Thanks for your attention!

References I

- Bruno Blanchet, *An efficient cryptographic protocol verifier based on prolog rules*, Proceedings. 14th IEEE Computer Security Foundations Workshop, 2001., CSFW '01, IEEE, IEEE Computer Society, 2001, pp. 82–96.
- Mohan Dhawan, Rishabh Poddar, Kshiteej Mahajan, and Vijay Mann, *Sphinx: Detecting security attacks in software-defined networks*, 2015.
- D. Dolev and A. C. Yao, *On the security of public key protocols*, Proceedings of the 22Nd Annual Symposium on Foundations of Computer Science (Washington, DC, USA), vol. 29, SFCS '81, no. 2, IEEE Computer Society, 1981, pp. 350–357.
- Marco Longhi Gelati, Giovanni Neri, Pierantonio Natali, Richard C. S. Morling, Gerald D. Cain, and Eugenio Faldella, *MININET: A local area network for real-time instrumentation applications*, Comput. Networks **8** (1984), 107–131.

References II

- Andrzej Kamisinski and Carol Fung, *Flowmon: Detecting malicious switches in software-defined networks*, Proceedings of the 2015 ACM Workshop on Automated Decision Making for Active Cyber Defense, ACM, 2015.
- Ahmed Khurshid, Xuan Zou, Wenxuan Zhou, Matthew Caesar, and P. Brighten Godfrey, *Veriflow: Verifying network-wide invariants in real time*, ACM SIGCOMM Computer Communication Review **42** (2012), 467–472.
- Simon Meier, Benedikt Schmidt, Cas Cremers, and David Basin, *The TAMARIN prover for the symbolic analysis of security protocols*, International Conference on Computer Aided Verification (Natasha Sharygina and Helmut Veith, eds.), Lecture Notes in Computer Science, vol. 8044, Springer, Springer, 2013, pp. 696–701.
- RYU, *Ryu sdn framework - english edition*, Release 1.0, RYU project team, 2014.

References III

- Arash Shaghaghi, Mohamed Ali Kaafar, and Sanjay Jha, *Wedgetail: An intrusion prevention system for the data plane of software defined networks*, Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ACM, 2017.
- Takahiro Shimizu, Naoya Kitagawa, Kohta Ohshima, and Nariyoshi Yamai, *Whiterabbit: Scalable software-defined network data-plane verification method through time scheduling*.
- Takayuki Sasaki, Christos Pappas, Taeho Lee, Torsten Hoefler, and Adrian Perrig, *Sdnsec: Forwarding accountability for the sdn data plane*, 2016 25th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2016, Available: NEC Corporation and ETH Zurich, pp. 1–10.
- P. P. S. Shin, V. Yegneswaran, and G. Gu, *Avant-guard: Scalable and vigilant switch flow management in software-defined networks*, CProceedings of the 2013 ACM SIGSAC conference on Computer & communications security (2013), 413–424.

References IV

V. Yegneswaran, M. Fong, M. Thottan, P. Porras, S. Shin, and G. Gu, *A security enforcement kernel for openflow networks*, ACM, 2012.

Peng Zhang, Hui Wu, Dan Zhang, and Qi Li, *Verifying rule enforcement in software defined networks with rev*, IEEE/ACM Transactions on Networking (2020).

Peng Zhang, Shimin Xu, Zuoru Yang, Hao Li, Qi Li, Huanzhao Wang, and Chengchen Hu, *Foces: Detecting forwarding anomalies in software defined networks*.