arrows.meta >=Latex[round]
dateplot
commentEnv
cryptoverif morekeywords=collision, const, crypto, define, defined, do, else, end, equation,
equiv, event,
event$_a bort, expand, find, forall, foreach, fun, get, implementation, in, if, inj, insert, length, let, l$
$< -, < -R,, sensitive = true, morecomment = [s](**), morestring =$
$[b]", cvoutputmorekeywords = , otherkeywords = , sensitive = true, morecomment = [s](**), mo$

Rosenpass

Securing & Deploying Post-Quantum WireGuard

Karolin Varner, with Benjamin Lipp, Wanja Zaeske, Lisa Schmidt

https://rosenpass.eu/whitepaper.pdf

November 23, 2023

Structure of the talk

- Problem statement: Post-quantum WireGuard
- Post-quantum WireGuard[1]: How to build an interactive key exchange from KEMs
- Attack we found: State Disruption Attacks
- Real-World Concerns
- Biscuits as a defense against State Disruption Attacks

[1] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber, and Philip R. Zimmermann. "Post-quantum WireGuard". In: 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021. Full version: https://eprint.iacr.org/2020/379

What needs to be done to deploy Post-Quantum WireGuard

- Updating the WireGuard protocol to support post-quantum security
- Updating the (post quantum) WireGuard protocol to be secure against state disruption attacks
- Reference implementation of the Rosenpass protocol in Rust
- A way to create hybrid post-quantum secure WireGuard VPNs
- Stand-alone key exchange app
- A Sci-Comm project teaching people about post-quantum security

WireGuard[3]

- VPN protocol in the linux kernel
- Based on Noise IKpsk1 from the Noise Protocol Framework[2]
- Small, fast, open source crypto

[2] Trevor Perrin. The Noise Protocol Framework. 2016. url: http://noiseprotocol.org/noise.pdf

[3] Jason A. Donenfeld. "WireGuard: Next Generation Kernel Network Tunnel". In: 24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017. Whitepaper: https: //www.wireguard.com/papers/wireguard.pdf.

WireGuard/Noise IKpsk security properties

Session-key secrecy

Forward-secrecy

Mutual authentication

Session-key Uniqueness

Identity Hiding

(DoS Mitigation – First packet is authenticated[4])

[4] Based on the unrealistic assumption of a monotonic counter – We found a practical attack

Security of Rosenpass

WireGuard

- Session-key secrecy
- Forward-secrecy
- Mutual authentication
- Session-key Uniqueness
- Identity Hiding
- (DoS Mitigation)

Post-Quantum WireGuard

- Identity Hiding [a]
- DoS Mitigation [b]

[a] Based on a Identity Hiding/ANON-CCA security of McEliece; unclear whether that holds.

[b] PQWG provides DoS mitigation under the assumption of a secret PSK, which quite frankly is cheating.

Rosenpass

- DoS Mitigation
- Hybrid Post-Quantum security[a]

[a] In deployments using WireGuard + Rosenpass; Rosenpass on its own provides post-quantum security.

Building post-quantum WireGuard: NIKE vs KEM

NIKE:

$(\mathtt{sk}_1, \mathtt{pk}_1) \leftarrow \mathtt{NIKE.KeyGen}$

$(\mathtt{sk}_2, \mathtt{pk}_2) \leftarrow \mathtt{NIKE.KeyGen}$

$\mathtt{NIKE.SharedKey}(\mathtt{sk}_1, \mathtt{pk}_2) = \mathtt{NIKE.SharedKey}(\mathtt{sk}_2, \mathtt{pk}_1)$

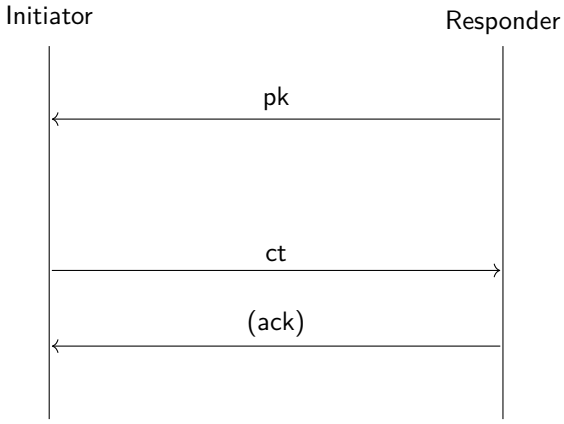
KEM:

$(\mathtt{sk}, \mathtt{pk}) \leftarrow \mathtt{KEM.KeyGen}$

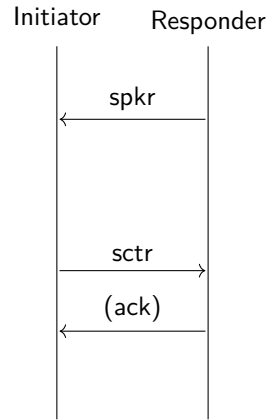
$(\mathtt{shk}, \mathtt{ct}) \leftarrow \mathtt{KEM.Encaps}(\mathtt{pk})$

$\mathtt{shk} = \mathtt{KEM.Decaps}(\mathtt{sk}, \mathtt{ct})$

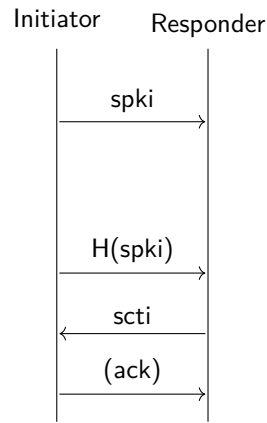
Minimal key exchange using KEMs



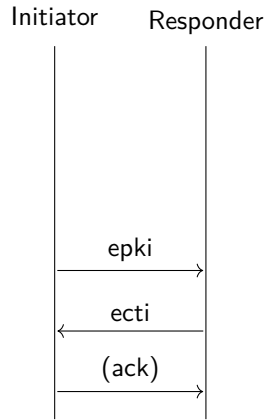
Three encapsulations: Achieving mutual authentication & forward secrecy



Responder Auth

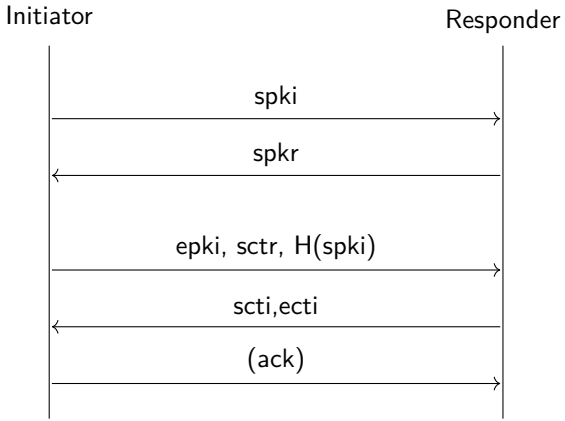


Initiator Auth



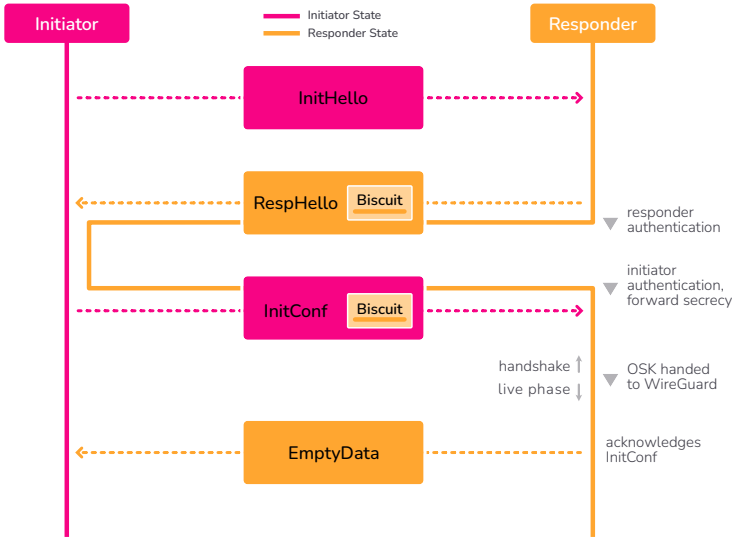
Forward secrecy

Combining the three encapsulations in one protocol

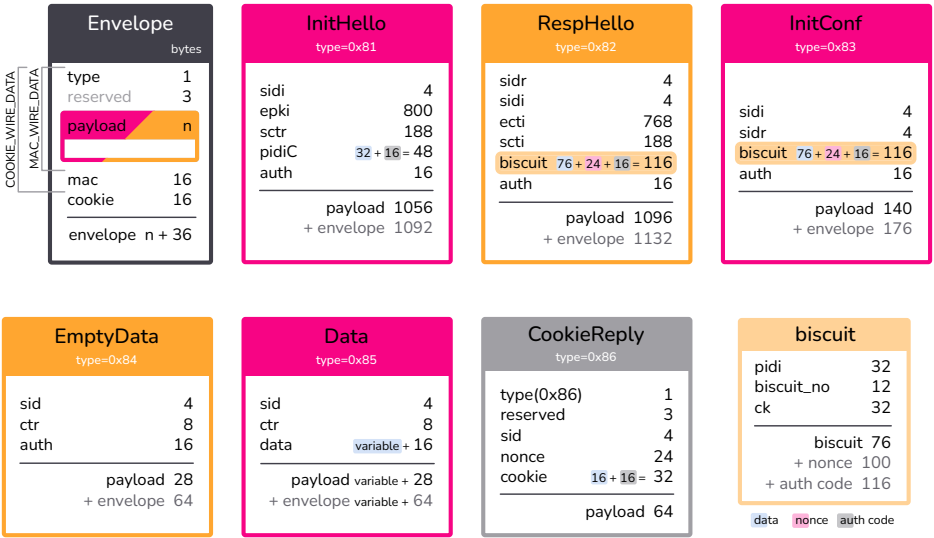


Note that the initiator is not authenticated until they send ‘(ack)‘.

In Rosenpasss specifically



In Rosenpasss specifically



State Disruption Attacks

- Use the fact that the initiator is not authenticated until their last message
- Send faux initiations, overwriting – and thus erasing – the responder's handshake state
- Erasing the state aborts protocol execution
- PQWG argues: The first package is authenticated using the PSK, therefor sending faux initiations works
- Attacker could replay a legitimate message, but...

State Disruption Attacks on authenticated initial package

- In Classic WireGuard the initial message (InitHello) is authenticated through static-static Diffie-Hellman
- Replay protection uses monotonic counter
- WireGuard stores the time of the last initiator t_i
- When WireGuard receives legitimate initiaton with timestamp t, it stores that time $t_i \leftarrow t$a
- All InitHello messages with a stale timestamp ($t \leq t_i$) get rejected

CVE-2021-46873 – Attacking WireGuard through NTP

- The replay protection in classic WireGuard assumes a monotonic counter
- But the system time is attacker controlled because NTP is insecure
- This generates a kill packet that can be used to render WireGuard keys useless
- Attack is possible in the real world!

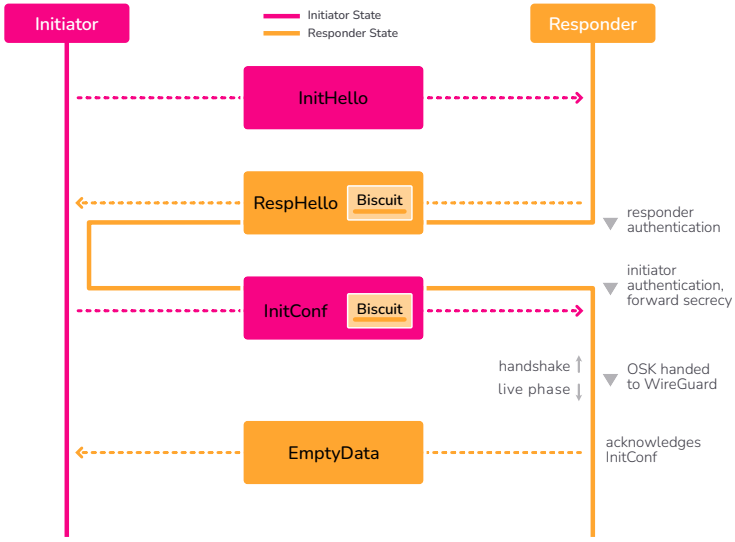
State disruption in Post-Quantum WireGuard

- This mechanism needs an authenticated InitHello message
- Post-Quantum WireGuard relies on the psk to provide InitHello authentication
- PQWG sets $\texttt{psk} = H(\texttt{spki} \oplus \texttt{spkr})$ to achieve a secret psk.a
- Relying on private public keys is absurd

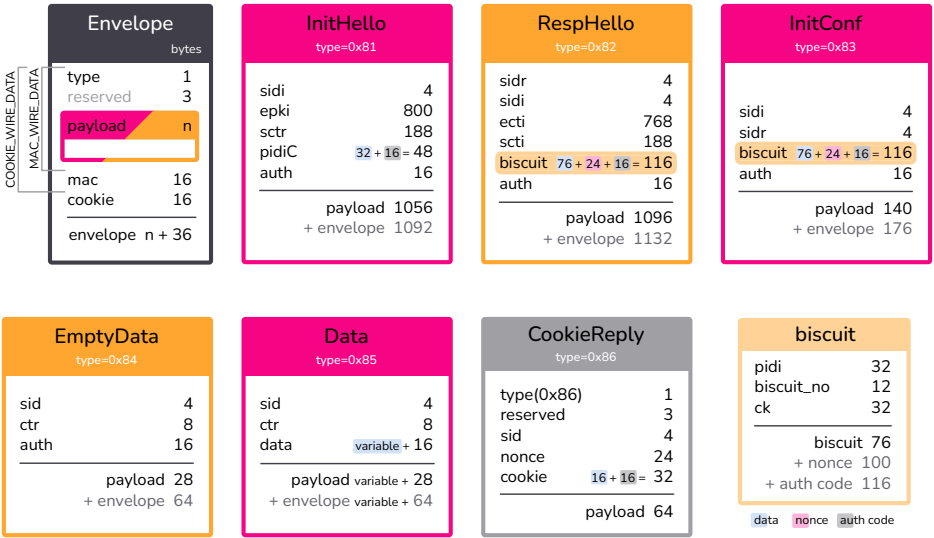
$\Rightarrow$ With InitHello effectively unauthenticated, attacker can just generate their own kill packet

Solution: Store the responder state in a biscuit (cookie), so there is no state to override.

Biscuits in the protocol flow



Biscuits in the messages



Biscuits

- Assumptions such as a monotonic counter are perilous in the real world
- Giving the adversary access to state is dangerous
- In noise protocols the handshake state is very small (32-64 bytes)
- Sending the state to the protocol peer is a viable course of action!
- Formalization of State Disruption Attacks covers many attacks of this style

Security proof of rosenpass

- CryptoVerif in progress (Benjamin Lipp)
- Really fast symbolic analysis using ProVerif

Deployment

- Rust implementation in userspace
- Integrates with WireGuard through the PSK feature to provide Hybrid security

Final statements

- Post-quantum crypto can be deployed now
- There are real complexities in protocol design
- DoS-Resistance needs formalization work
- Availability needs love and attention from cryptographers
- Try it out! https://rosenpass.eu/