https://rosenpass.eu

https://chaos.social/@rosenpass

MRMCD 2023

2023-09-03

Sichere Kryptografie trotz Quantencomputern: Projektupdate

Emil Engler, Stephan Ajuvo, Karolin Varner

Marei Peischl, Lisa Schmidt, Steffen Vogel, Alice Bowman, Wanja Zaeske, Sven Friedrich, Benjamin Lipp

Funding: NLNet & Prototype Fund

Was passiert im Talk?

- Was bisher geschah
- Zusammenfassung vom EH20 Talk: Was ist Rosenpass
- Was nach dem Easterhegg passiert ist
- Was wir nun vor haben
  - go-rosenpass
  - NetBird
  - Broker-Architektur & Schnittstellen zum Einbinden

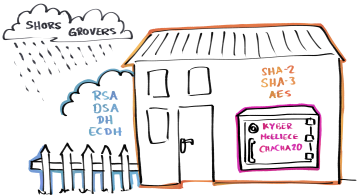
Was bisher geschah

- Seit 2020: Entwicklung der Kryptografie & der Software
- Feb. 2023: Softwarerelease & Whitepaper
- März 2023: NLNet Projekt um Sicherheitsbeweis mit CryptoVerif zu erzeugen
- März 2023: Talk auf dem Real World Post-Quantum Krypto Workshop in Tokyo
- April 2023: Vorstellung/Erklärung auf dem Easterhegg[1]
- Aug. 2023: Release Kandidat 0.2.0 mit FreeBSD unterstützung
- Sep. 2023: Beginn des Prototype Fund 14 Projektes für Isolation in Rosenpass

[1] https://media.ccc.de/search/?q=rosenpass

Warum sind Quantencomputer (k)eine Bedrohung?

- Grovers Algorithmus **schwächt** symmetrische Kryptografie
  - AES, SHA-2, SHA-3, Chacha20
  - Lösung: größere Keys
- Shors Algorithmus **bricht** asymmetrische Kryptografie
  - RSA, DSA, DH, ECDH
  - Lösung: alternative Kryptografie
- Nur auf großen Quantencomputern
  - Die existieren noch nicht
  - Problem: Store now, decrypt later



Quantencomputer überschatten Kryptografieverfahren.

PQ-sichere VPNs: WireGuard + Rosenpass

- Hybride Sicherheit
  - Bricht nur, wenn Rosenpass **und** WireGuard versagen
- Überall nutzbar, wo WireGuard schon läuft
- Ohne Anpassung vom WireGuard Source Code
  - Shared Secret aus Rosenpass = PSK für WireGuard
- Aber:
  - Ein Prozess mehr
  - Handshake alle 2 Minuten

WireGuard mit Rosenpass.

Rosenpass: Sicherheitseigenschaften

WireGuard	PQ WireGuard [3]	Rosenpass
Session-key secrecy	**Post-Quantum Security**	**Non-Interruptability** [5]
…	**Hybrid security**	**Hybrid security** [6]
Identity Hiding	**Non-Interruptability** [4]	
Non-Interruptability [2]		
Post-Quantum Security		

[2] Angenommen der Systemzeit wird Vertraut
[3] Hülsing, Ning, Schwabe, Weber, Zimmermann. “Post-quantum WireGuard”. https://ia.cr/2020/379
[4] Assuming a PSK
[5] Through cookies
[6] Wenn es mit WireGuard benutzt wird

Zum Nachbauen… aus dem Whitepaper:

Initiator Code	Responder Code	Comments

1 — InitHello { sidi, epki, sctr, pidiC, auth } — **2**

Line	Variables ← Action	Variables ← Action	Line	Comment
IH1	ck ← thash("chaining key init", spkr)	ck ← thash("chaining key init", spkr)	IHR1	Initialize the chaining key, and bind to the responder's public key.
IH2	sidi ← random_session_id();			The session ID is used to associate packets with the handshake state.
IH3	eski, epki ← EKEM::keygen();			Generate fresh ephemeral keys, for forward secrecy.
IH4	mix(sidi, epki);	mix(sidi, epki)	IHR4	InitHello includes sidi and epki as part of the protocol transcript, and so we mix them into the chaining key to prevent tampering.
IH5	sctr ← encaps_and_mix<SKEM>(spkr);	decaps_and_mix<SKEM>(sskr, spkr, ct1)	IHR5	Key encapsulation using the responder's public key. Mixes public key, shared secret, and ciphertext into the chaining key, and authenticates the responder.
IH6	pidiC ← encrypt_and_mix(pidi);	spki, psk ← lookup_peer(decrypt_and_mix(pidiC))	IHR6	Tell the responder who the initiator is by transmitting the peer ID.
IH7	mix(spki, psk);	mix(spki, psk);	IHR7	Ensure the responder has the correct view on spki. Mix in the PSK as optional static symmetric key, with epki and spkr serving as nonces.
IH8	auth ← encrypt_and_mix(empty())	decrypt_and_mix(auth)	IHR8	Add a message authentication code to ensure both participants agree on the session state and protocol transcript at this point.

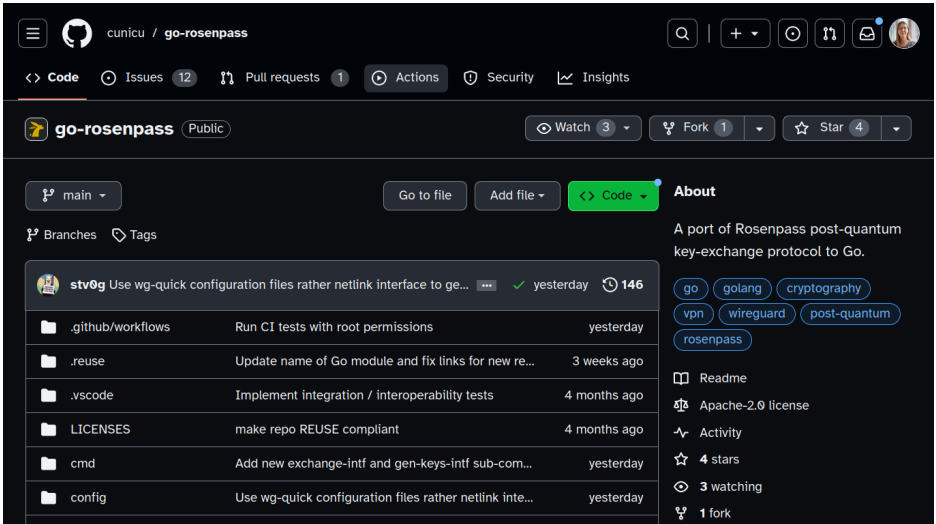
4 — RespHello { sidr, sidi, ecti, scti, biscuit, auth } — **3**

Line	Variables ← Action	Variables ← Action	Line	Comment
RHR1		sidr ← random_session_id()	RHR1	Responder generates a session ID.
RHR2	ck ← lookup_session(sidi);		RHR2	Initiator looks up their session state using the session ID they generated.
RHR3	mix(sidr, sidi);	mix(sidr, sidi);	RHR3	Mix both session IDs as part of the protocol transcript.
RHR4	decaps_and_mix<EKEM>(eski, epki, ecti);	ecti ← encaps_and_mix<EKEM>(epki);	RHR4	Key encapsulation using the ephemeral key, to provide forward secrecy.
RHR5	decaps_and_mix<SKEM>(sski, spki, scti);	scti ← encaps_and_mix<SKEM>(spki);	RHR5	Key encapsulation using the initiator's static key, to authenticate the initiator, and non-forward-secret confidentiality.
RHR6	mix(biscuit)	biscuit ← store_biscuit();	RHR6	The responder transmits their state to the initiator in an encrypted container to avoid having to store state.
RHR7	decrypt_and_mix(auth)	auth ← encrypt_and_mix(empty());	RHR7	Add a message authentication code for the same reason as above.

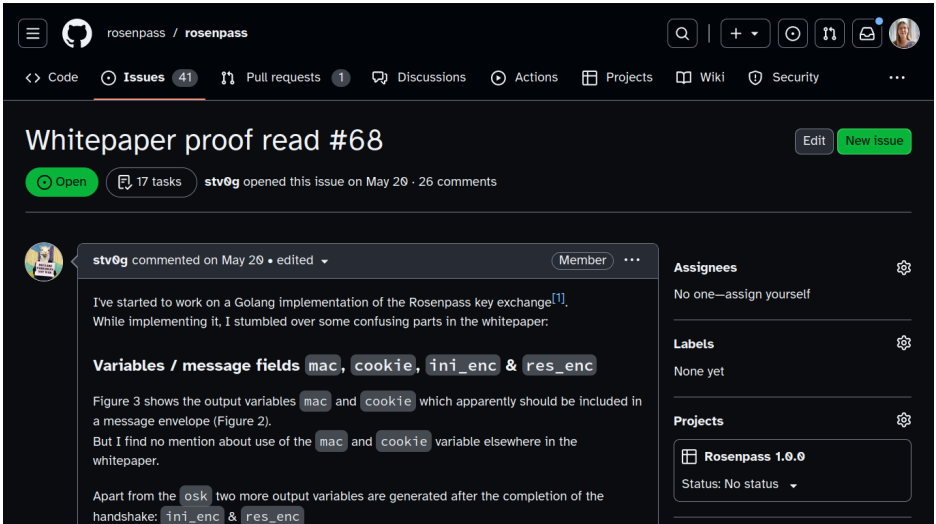
5 — InitConf { sidi, sidr, biscuit, auth } — **6**

Line	Variables ← Action	Variables ← Action	Line	Comment
ICI1		biscuit_no ← load_biscuit(biscuit);	ICR1	Responder loads their biscuit. This restores the state from after RHR6.
ICI2		encrypt_and_mix(empty());	ICR2	Responder recomputes RHR7, since this step was performed after biscuit encoding.
ICI3	mix(sidi, sidr);	mix(sidi, sidr);	ICR3	Mix both session IDs as part of the protocol transcript.
ICI4	auth ← encrypt_and_mix(empty);	decrypt_and_mix(auth);	ICR4	Message authentication code for the same reason as above, which in particular ensures that both participants agree on the final chaining key.
ICI5		assert(biscuit_no > biscuit_used);	ICR5	Biscuit replay detection.
ICI6		biscuit_used ← biscuit_no;	ICR6	Biscuit replay detection.
ICI7	enter_live();	enter_live();	ICR7	Derive the transmission keys, and the output shared key for use as WireGuard's PSK.

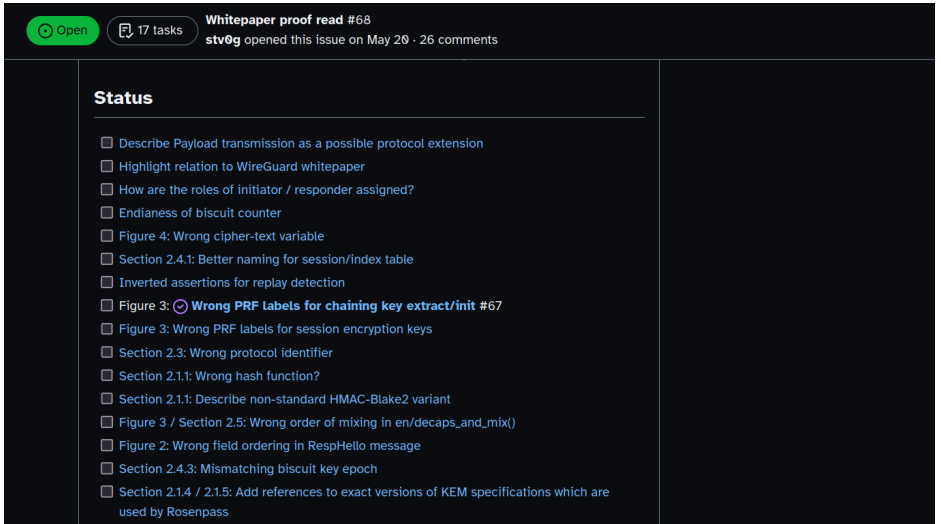
Zum Nachbauen… go-rosenpass – Steffen Vogel FTW



Zum Nachbauen… go-rosenpass – Steffen Vogel FTW



Zum Nachbauen… go-rosenpass – Steffen Vogel FTW



Open · 17 tasks

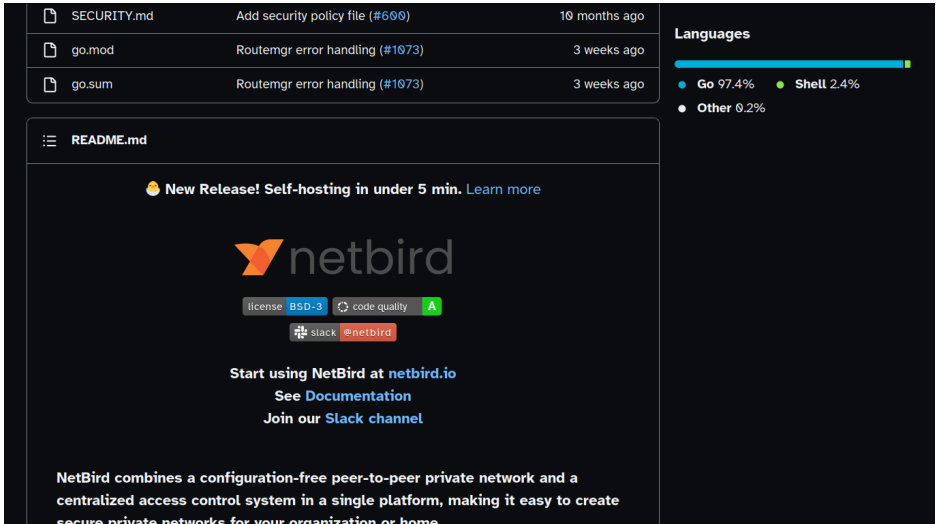
Whitepaper proof read #68

stv0g opened this issue on May 20 · 26 comments

Status

- [ ] Describe Payload transmission as a possible protocol extension
- [ ] Highlight relation to WireGuard whitepaper
- [ ] How are the roles of initiator / responder assigned?
- [ ] Endianess of biscuit counter
- [ ] Figure 4: Wrong cipher-text variable
- [ ] Section 2.4.1: Better naming for session/index table
- [ ] Inverted assertions for replay detection
- [ ] Figure 3: **Wrong PRF labels for chaining key extract/init** #67
- [ ] Figure 3: Wrong PRF labels for session encryption keys
- [ ] Section 2.3: Wrong protocol identifier
- [ ] Section 2.1.1: Wrong hash function?
- [ ] Section 2.1.1: Describe non-standard HMAC-Blake2 variant
- [ ] Figure 3 / Section 2.5: Wrong order of mixing in en/decaps_and_mix()
- [ ] Figure 2: Wrong field ordering in RespHello message
- [ ] Section 2.4.3: Mismatching biscuit key epoch
- [ ] Section 2.1.4 / 2.1.5: Add references to exact versions of KEM specifications which are used by Rosenpass

Zum Integrieren... NetBird



SECURITY.md	Add security policy file (#600)	10 months ago
go.mod	Routemgr error handling (#1073)	3 weeks ago
go.sum	Routemgr error handling (#1073)	3 weeks ago

README.md

New Release! Self-hosting in under 5 min. Learn more

netbird

license BSD-3 | code quality A

slack @netbird

Start using NetBird at netbird.io
See Documentation
Join our Slack channel

NetBird combines a configuration-free peer-to-peer private network and a centralized access control system in a single platform, making it easy to create secure private networks for your organization or home

Languages

Go 97.4% Shell 2.4%
Other 0.2%

Zum Integrieren... NetBird

README.md

Together with CISPA Helmholtz Center for Information Security NetBird brings the security best practices and simplicity to private networking.

Testimonials

We use open-source technologies like WireGuard®, Pion ICE (WebRTC), Coturn, and Rosenpass. We very much appreciate the work these guys are doing and we'd greatly appreciate if you could support them in any way (e.g. giving a star or a contribution).

Legal

WireGuard and the *WireGuard* logo are registered trademarks of Jason A. Donenfeld.

Zum Integrieren... NetBird

- Netbird: Einfaches user interface
- Rosenpass: Hochsichere PQ-Crypto
- go-rosenpass: Um Plattformen zu unterstützen auf denen die Rust Variante schwer zu integrieren ist
  - Android
  - iOS
  - Windows
  - ...

Zum Integrieren... Prototypefund 14 Projekt

- Schnittstelle zwischen Komponenten
- Kommunikation über Unix-Sockets
- Spezielle Serialisierungsbibliothek für Schlüsseldaten[7]
- Broker-Pattern für Rosenpass– Jede Komponente in einem eigenen Prozess
- Mikro-VMs um wirklich hohe sicherheit zu haben
- Minimale Privilegien; Sandboxing

[7] Externes Memory-Management

Rosenpass Roadmap

- Sicherheitsbeweis
- Formelle Verifikation der Implementierung
- Einfachere Benutzbarkeit
- Kryptografie + Safety Forschung:
  - Kryptografie in der Avionik
  - Decryption Despite Error

Rosenpass Strukturpläne

- Translationsforschung: Schnittstelle zwischen Industrie und Wissenschaft
- Mit mehreren Integratoren arbeiten; Open-Source R&D-Abteilung
- Antihierarchisches Arbeiten
- Karo hätte gerne mal wieder Freizeit

Idee: Kryptografie als Notar Erklären

- Problem: Kryptografie wird als schwarze Magie verstanden
- Problem: Krude vorschläge zur Verwaltungsautomatisierung
- Problem: Und Strafverfolgung
- Problem: Krypto wird auf Verschlüsselung reduziert
- Problem: Kaum jemand weiß was moderne Verfahren tun
  - Elliptic-Curve Pairings
  - Multi-Party Computation
  - Homomorpe Verschlüsselung
  - Datenbanken mit anonymem Zugriff
  - Anonyme Kommunikation

Quelle: White Rabbit aus Alice in Wonderland – CC0

Häschen sind die besseren Menschen.

Idee: Kryptografie als Notar Erklären

- Krypto: Einsatzfähig für viele Prozesse in denen Information Übertragen wird
- Datenschutz: Anonyme, Nutzergesteuerte Prozesse
- Idee: Metapher von Kryptografie als Notar
  - Notare werden Bestraft wenn sie Dinge Zusichern die sie nicht können
  - Oder wenn sie gegen Regeln verstoßen
  - Spezieller schutz vor dem Recht
  - Kryptografie: Ähnlich, nur Matematisch, statt mit Staatsgewalt

Quelle: White Rabbit aus Alice in Wonderland – CCO

Häßchen sind die besseren Menschen.

rosenpass.eu | @rosenpass@chaos.social | info@rosenpass.eu
Emil Engler @engler@chaos.social | Stephan Ajuvo @ajuvo@chaos.social | Karolin Varner @kora@chaos.social