A Unified Symbolic Analysis of WireGuard

Pascal Lafourcade\textsuperscript{1, 2} \hspace{1cm} Dhekra Mahmoud\textsuperscript{1, 2} \hspace{1cm} Sylvain Ruhault\textsuperscript{3}

\textsuperscript{1}Université Clermont Auvergne,\hspace{1cm}\textsuperscript{2}Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes,\hspace{1cm}\textsuperscript{3}Agence Nationale de la Sécurité des Systèmes d'Information

March 3, 2024
The **WireGuard** protocol

**Without cookie**

\[
\begin{align*}
G, u, U &= g^u, x, X = g^x, ts, psk \\
G, v, V &= g^v, y, Y = g^y, psk
\end{align*}
\]

- **U**
- **V**

\[
\begin{align*}
[1]0^3||s_i||X||\{U\}||\{ts\}||mac_1^i||0^{16} \\
[2]0^3||s_r||s_i||Y||\{\emptyset\}||mac_1^r||0^{16} \\
[3]0^3||s_r||0||\{\text{pad}(P_0)\} \\
\end{align*}
\]

- **3**
- **0**
- **si**
- **sr**
- **0**
- **pad(P\_0)**

**New model**

- \( u, U = g^u, v, V = g^v \) → static keys, \( x, X = g^x, y, Y = g^y \) → ephemeral keys, \( psk \) → pre-shared key
- \( ts \) timestamp, \( s_i, s_r \) → session identifiers, \( i_∗, i_k \) → counters, \( P_∗ \) → plaintexts
- \( \{\cdot\} \) → encryption
- \( \rho \) → nonce, \( \tau \) → cookie
Current symbolic analyses

Symbolic

- 2018: J. A. Donenfeld and K. Milner, “Formal verification of the WireGuard protocol” *WireGuard*

Threats

- Static private key reveal / set
- Ephemeral private key reveal / set
- PSK reveal / set
- Static key distribution corruption

Security Properties

- Message agreement
- Key secrecy (incl. PFS)
- Anonymity
Symbolic analysis of WireGuard (TAMARIN)


\[ G, u, U = g^u, V, x, X = g^x, \text{psk} \]
\[ G, v, V = g^v, U, y, Y = g^y, \text{psk} \]

Threats
- Static private key reveal ✓ / set \( X \)
- Ephemeral private key reveal ✓ / set \( X \)
- PSK reveal ✓ / set \( X \)
- Static key distribution corruption ✗

Security Properties
- Message agreement ✓
- Key secrecy ✓ (PFS ✗)
- Anonymity ✓
Our target threat model for WireGuard

Threats

- Static private key reveal ✓ / set ✓
- Ephemeral private key reveal ✓ / set ✓
- PSK reveal ✓ / set ✓
- Static reveal ✓ / set ✓
- Static key distribution corruption ✓
- New! Pre-computation reveal ✓ / set ✓

Pre-computation?

- Static-static key:
  - Initiator \( V^u = g^{uv} \)
  - Responder \( U^v = g^{uv} \)

before session begins, hence WireGuard maintains it.

Compromise of \( g^{uv} \) is weaker than compromise of \( u \) or \( v \):

- \( u \land g^v \rightarrow g^{uv} \)
- however \( g^v \land g^{uv} \nRightarrow u \)
Our symbolic models of WireGuard (TAMARIN, PROVERIF, SAPIC⁺)

**Without cookie**

\[
\begin{align*}
G, u, U & = g^u, x, X = g^x, ts, psk \\
G, v, V & = g^v, y, Y = g^y, psk \\
U & \\
V & \\
[1|0^3|s_i|X||\{U||\{ts\}||mac_i]|0^{16}] & \\
[2|0^3|s_i|Y||\{\emptyset\}||mac_i]|0^{16} & \\
[3|0^3|s_i|0||\{pad(P_{ki})\}] & \\
[3|0^3|s_i|i_k||\{pad(P_{ki})\}] & \\
[3|0^3|s_i|r_k||\{pad(P_{ri})\}] & \\
\end{align*}
\]

**With cookie**

\[
\begin{align*}
G, u, U & = g^u, x, X = g^x, ts, psk \\
G, v, V & = g^v, y, Y = g^y, psk \\
U & \\
V & \\
[1|0^3|s_i|X||\{U||\{ts\}||mac_i]|0^{16}] & \\
[2|0^3|s_i|Y||\{\emptyset\}||mac_i]|0^{16} & \\
[3|0^3|s_i|0||\{pad(P_{ki})\}] & \\
[3|0^3|s_i|i_k||\{pad(P_{ki})\}] & \\
[3|0^3|s_i|r_k||\{pad(P_{ri})\}] & \\
\end{align*}
\]

**Threats**

- Static private key reveal ✓ / set ✓
- Ephemeral private key reveal ✓ / set ✓
- PSK reveal ✓ / set ✓
- Static key distribution corruption ✓
- New! Pre-computation reveal ✓ / set ✓

**Security Properties**

- Message agreement ✓
- Key secrecy ✓ (PFS ✓)
- Anonymity ✓
Our results: necessary and sufficient conditions

- $D_u, D_v$: adversary corrupts public keys distribution
- $R_u, R_v, R_x, R_y, R_s, R_c$: adversary gets private keys ($u, v, x, y$), psk ($s$) or pre-comp. value ($c$)
- $R_u^*, R_v^*, R_s^*$: adversary gets private keys ($u, v$), psk ($s$) or pre-comp. value ($c$) after protocol execution (for PFS)

Results

- agreement of RecHello and TransData (R to I) messages hold **unless**
  $$(D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x)$$
- agreement of TransData (I to R) messages hold **unless**
  $$(D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y)$$
- Key Secrecy from Initiator’s view, including PFS hold **unless**
  $$(D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x) \lor (R_s^* \land R_u^* \land R_x) \lor (R_s^* \land R_v^* \land R_y) \lor (R_c^* \land R_s^* \land R_x \land R_y)$$
- Key Secrecy from Responder’s view, including PFS hold **unless**
  $$(D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y) \lor (R_s^* \land R_u^* \land R_x) \lor (R_s^* \land R_v^* \land R_y) \lor (R_c^* \land R_s^* \land R_x \land R_y)$$
Our results: interpretation

Results

- agreement of RecHello and TransData (R to I) messages hold **unless**
  \((D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x)\)

- agreement of TransData (I to R) messages hold **unless**
  \((D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y)\)

- Key Secrecy from Initiator’s view, including PFS hold **unless**
  \((D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x) \lor (R^*_s \land R^*_u \land R_x) \lor (R^*_s \land R^*_v \land R_y) \lor (R^*_c \land R^*_s \land R_x \land R_y)\)

- Key Secrecy from Responder’s view, including PFS hold **unless**
  \((D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y) \lor (R^*_s \land R^*_u \land R_x) \lor (R^*_s \land R^*_v \land R_y) \lor (R^*_c \land R^*_s \land R_x \land R_y)\)

Key distribution corruption

Agreement and key secrecy hold **unless** adversary:

- compromises \(U\) distribution **AND** gets psk
- **OR** compromises \(V\) distribution **AND** gets psk

⇒ Shall not be eluded!
Our results: interpretation

Results

- agreement of RecHello and TransData (R to I) messages hold unless
  \[(D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x)\]

- agreement of TransData (I to R) messages hold unless
  \[(D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y)\]

- Key Secrecy from Initiator’s view, including PFS hold unless
  \[(D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x) \lor (R^*_s \land R^*_u \land R_x) \lor (R^*_s \land R^*_v \land R_y) \lor (R^*_c \land R^*_s \land R_x \land R_y)\]

- Key Secrecy from Responder’s view, including PFS hold unless
  \[(D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y) \lor (R^*_s \land R^*_u \land R_x) \lor (R^*_s \land R^*_v \land R_y) \lor (R^*_c \land R^*_s \land R_x \land R_y)\]

Pre-shared key

psk compromise is necessary to break all properties.

⇒ Shall be mandatory (and not optional)!
Our results: interpretation

Results

- agreement of RecHello and TransData (R to I) messages hold unless:
  \[(D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x)\]

- agreement of TransData (I to R) messages hold unless:
  \[(D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y)\]

- Key Secrecy from Initiator’s view, including PFS hold unless:
  \[(D_v \land R_s) \lor (R_s \land R_v) \lor (R_c \land R_s \land R_x) \lor (R_s \land R_u \land R_x) \lor (R_s^* \land R_u^* \land R_x) \lor (R_s^* \land R_v^* \land R_y) \lor (R_c^* \land R_s^* \land R_x \land R_y)\]

- Key Secrecy from Responder’s view, including PFS hold unless:
  \[(D_u \land R_s) \lor (R_s \land R_u) \lor (R_c \land R_s \land R_y) \lor (R_s \land R_v \land R_y) \lor (R_s^* \land R_u^* \land R_x) \lor (R_s^* \land R_v^* \land R_y) \lor (R_c^* \land R_s^* \land R_x \land R_y)\]

Pre-computation

In some cases, \(R_c\) has same impact as \(R_u\) or \(R_v\), although weaker.

\[\Rightarrow\text{Shall be removed!}\]
Anonymity

Claim: Wireguard guarantees Identity Hiding

(Identity hiding proven in 2018 model with TAMARIN)

\[ G, u, U = g^u, V_1, V_2, x, X = g^x, ts, psk \]
\[ G, v^*, V^* = g^{v^*}, U, y, Y = g^y, psk \]
\[ [1|0^3|s_i|X||\{U\}||\{ts\}||\text{mac}_i^i||0^{16}] \]

- InitHello message is \([1|0^3|s_i|X||\{U\}||\{ts\}||\text{mac}_i^i||0^{16}]\)
- \(\text{mac}_i^i = \text{mac}(H(V), [1|\cdots||\{ts\}]), \) where \(V\) is public \(\implies\) Responder’s Identity can leak!
Anonymity

Claim: Wireguard guarantees Identity Hiding

(Identity hiding proven in 2018 model with TAMARIN)

\[ G, u, U_s = g^u, V, x, X = g^x, ts, psk \]
\[ G, v, V = g^v, U_1, U_2, y, Y = g^y, psk \]
\[
[1||0^3||s_r||\{U\}||\{ts\}||\text{mac}_1||0^{16}] \\
[2||0^3||s_r||s_i||Y||\{\emptyset\}||\text{mac}_1||0^{16}] \\
\text{mac}(H(U_1), [2||\cdots||\{\emptyset\}) \overset{?}{=} \text{mac}_1^f \\
\text{mac}(H(U_2), [2||\cdots||\{\emptyset\}) \overset{?}{=} \text{mac}_1^f \\
\]

However issue is the same for RecHello message! (explained in “A mechanised cryptographic proof of the WireGuard VPN protocol”)

- RecHello message is \([2||0^3||s_r||s_i||Y||\{\emptyset\}||\text{mac}_1^f||0^{16}]\)
- \(\text{mac}_1^f = \text{mac}(H(U), [2||\cdots||\{\emptyset\}]), \) where \(U\) is public \(\Rightarrow\) Initiator's Identity can leak!
Anonymity

Claim: Wireguard guarantees Identity Hiding

(Identity hiding proven in 2018 model with TAMARIN)

⇒ Reality: WireGuard does **not** provide anonymity at all (key compromise is not necessary) ...
Anonymity

Claim: Wireguard guarantees Identity Hiding

(Identity hiding proven in 2018 model with TAMARIN)

⇝ Reality: WireGuard does not provide anonymity at all (key compromise is not necessary) ...

Proposed fixes

▶ Remove mac (i.e. use IKpsk2)
▶ Change mac computation :
  ▶ $mac' = \text{mac}(H(U||g^uv), [2||\cdots||\emptyset])$
  ▶ $mac_1 = \text{mac}(H(U||psk), [2||\cdots||\emptyset])$

⇒ With these fixes anonymity is verified with PROVERIF
Conclusion

Currently WireGuard ensures:
- Agreement
- Key secrecy and PFS

Recommendations for end users:
- Use pre-shared key
- Care about static key distribution
- Do not rely on WireGuard for anonymity

Recommendations for stakeholders:
- Remove pre-computation
- Fix anonymity
Conclusion

- Currently WireGuard ensures:
  - Agreement
  - Key secrecy and PFS

- Recommendations for end users:
  - Use pre-shared key
  - Care about static key distribution
  - Do not rely on WireGuard for anonymity

- Recommendations for stakeholders:
  - Remove pre-computation
  - Fix anonymity

- Complete model of WireGuard
- **Fix** for anonymity property
- Precise threat model, including initial key distribution and **pre-computations**
- Necessary and sufficient conditions
- Process with SAPIC\(^+\), PROVERIF, TAMARIN
Conclusion

- Currently WireGuard ensures:
  - Agreement
  - Key secrecy and PFS

- Recommendations for end users:
  - Use pre-shared key
  - Care about static key distribution
  - Do not rely on WireGuard for anonymity

- Recommendations for stakeholders:
  - Remove pre-computation
  - Fix anonymity

- Complete model of WireGuard
- Fix for anonymity property
- Precise threat model, including initial key distribution and pre-computations
- Necessary and sufficient conditions
- Process with SAPIC$^+$, PROVERIF, TAMARIN

- Thanks for your attention!
- Do you have questions?
Computationnal analysis of **WireGuard** (manual)

2018: B. Dowling *et al.*, “A cryptographic analysis of the WireGuard protocol”

$$G, u, U = g^u, V, x, X = g^x, psk$$

$$G, v, V = g^v, U, y, Y = g^y, psk$$

---

**Threats**

- Static private key reveal ✓ / set ✗
- Ephemeral private key reveal ✓ / set ✗
- PSK reveal ✓ / set ✗
- Static key distribution corruption ✗

**Security Properties**

- Message agreement ✓
- Key secrecy ✓ (PFS ✗)
- Anonymity ✗

**Verified Combinations**

- ✗
Computationnnaal analysis of WireGuard (CRYPTOVERIF)

2019: B. Lipp et al., “A mechanised cryptographic proof of the WireGuard VPN protocol”

\[ \mathbb{G}, u, U = g^u, x, X = g^x, \text{psk} \quad \mathbb{G}, v, V = g^v, y, Y = g^y, \text{psk} \]

Threats

- Static private key reveal ✓ / set ✓
- Ephemeral private key reveal ✓ / set ✗
- PSK reveal ✓ / set ✓
- Static key distribution corruption ✓

Security Properties

- Message agreement ✓
- Key secrecy ✓ (PFS ✓)
- Anonymity ✗

Verified Combinations

✗
Symbolic analysis of **IKpsk2 (PROVERIF)**


\[
\begin{align*}
G, u, U &= g^u, V, x, X = g^x, \text{psk} \\
G, v, V &= g^v, U, y, Y = g^y, \text{psk}
\end{align*}
\]

**Threats**
- Static private key reveal ✓ / set ✗
- Ephemeral private key reveal ✗ / set ✗
- PSK reveal ✓ / set ✗
- Static key distribution corruption ✗

**Security Properties**
- Message agreement ✓
- Key secrecy ✓ (PFS ✓)
- Anonymity ✗

**Verified Combinations**
- ✗
Symbolic analysis of **IKpsk2 (TAMARIN)**


\[
\begin{align*}
G, u, U = g^u, x, X = g^x, ts, psk & \quad \quad \quad G, v, V = g^v, y, Y = g^y, psk
\end{align*}
\]

**Threats**
- Static private key reveal ✓ / set ✓
- Ephemeral private key reveal ✓ / set ✓
- PSK reveal ✓ / set ✓
- Static key distribution corruption ✓

**Security Properties**
- Message agreement ✓
- Key secrecy ✓ (PFS ✓)
- Anonymity ✓

**Verified Combinations**
- ✓
Benchmarks

With a dedicated 256 cores server

- Evaluation of agreement and secrecy properties (PROVERIF, TAMARIN, SAPIC\(^+\)) : 9 hours
- Evaluation of fix for anonymity, based on \(g^{uv}\) (PROVERIF) : 12 hours
- Evaluation of fix for anonymity, based on psk (PROVERIF) : 2 hours
Combinations

**With pre-computation**

Adversary can

- get $u, v, x, y, \text{psk}, g^{uv}$ before / after protocol execution
- set $u, v, x, y, \text{psk}, g^{uv}$ for Initiator and $g^{uv}$ for Responder
- compromise $U$ and $V$ distribution
- and combine $(2^6+6+7+2 = 2^{21} = 2097152$ combinations per property)!