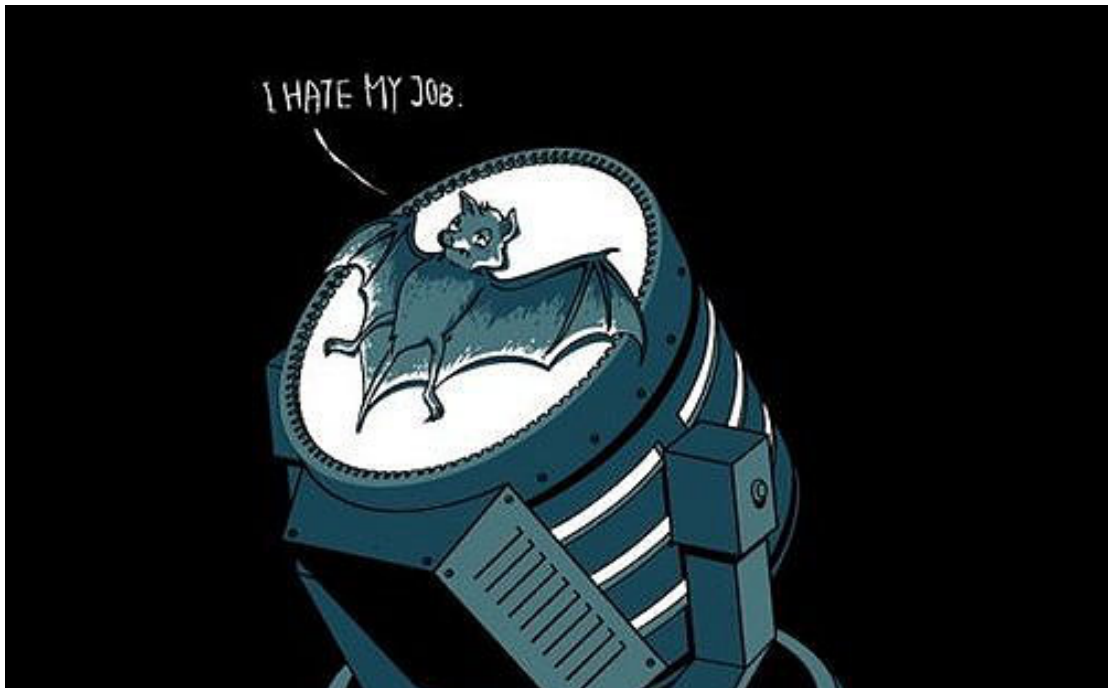
Implementing Signal

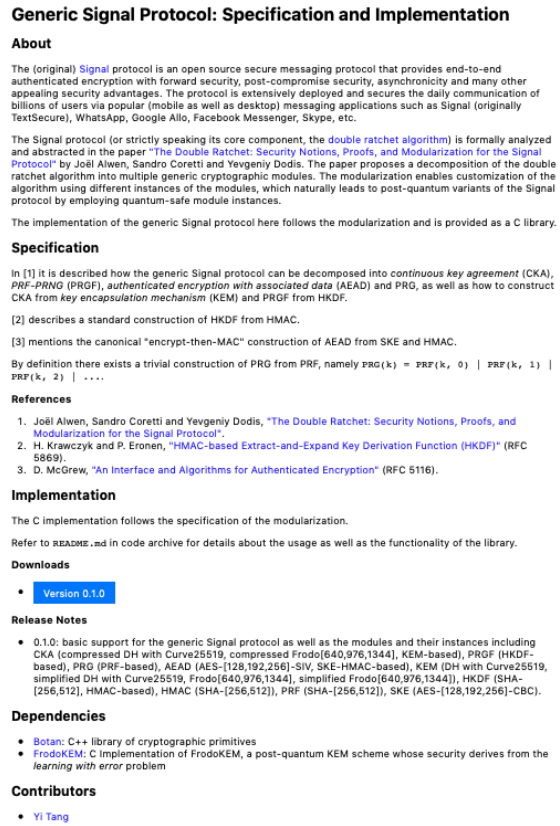
Yi Tang, Yevgeniy Dodis
New York University

August 20, 2019



Project Homepage

URL: `https://cims.nyu.edu/~yt1433/signal.html`.



Generic Signal Protocol: Specification and Implementation

About

The (original) Signal protocol is an open source secure messaging protocol that provides end-to-end authenticated encryption with forward security, post-compromise security, asynchronicity and many other appealing security advantages. The protocol is extensively deployed and secures the daily communication of billions of users via popular (mobile as well as desktop) messaging applications such as Signal (originally TextSecure), WhatsApp, Google Allo, Facebook Messenger, Skype, etc.

The Signal protocol (or strictly speaking its core component, the double ratchet algorithm) is formally analyzed and abstracted in the paper "The Double Ratchet: Security Notions, Proofs, and Modularization for the Signal Protocol" by Joël Alwen, Sandro Coretti and Yevgeniy Dodis. The paper proposes a decomposition of the double ratchet algorithm into multiple generic cryptographic modules. The modularization enables customization of the algorithm using different instances of the modules, which naturally leads to post-quantum variants of the Signal protocol by employing quantum-safe module instances.

The implementation of the generic Signal protocol here follows the modularization and is provided as a C library.

Specification

In [1] it is described how the generic Signal protocol can be decomposed into *continuous key agreement* (CKA), *PRF-PRNG* (PRGF), *authenticated encryption with associated data* (AEAD) and PRG, as well as how to construct CKA from *key encapsulation mechanism* (KEM) and PRGF from HKDF.

[2] describes a standard construction of HKDF from HMAC.

[3] mentions the canonical "encrypt-then-MAC" construction of AEAD from SKE and HMAC.

By definition there exists a trivial construction of PRG from PRF, namely `PRG(k) = PRF(k, 0) | PRF(k, 1) | PRF(k, 2) | ...`.

References

1. Joël Alwen, Sandro Coretti and Yevgeniy Dodis, "The Double Ratchet: Security Notions, Proofs, and Modularization for the Signal Protocol".
2. H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)" (RFC 5869).
3. D. McGrew, "An Interface and Algorithms for Authenticated Encryption" (RFC 5116).

Implementation

The C implementation follows the specification of the modularization.

Refer to `README.md` in code archive for details about the usage as well as the functionality of the library.

Downloads

- Version 0.1.0

Release Notes

- 0.1.0: basic support for the generic Signal protocol as well as the modules and their instances including CKA (compressed DH with Curve25519, compressed Frodo[640,976,1344], KEM-based), PRGF (HKDF-based), PRG (PRF-based), AEAD (AES-[128,192,256]-SIV, SKE-HMAC-based), KEM (DH with Curve25519, simplified DH with Curve25519, Frodo[640,976,1344], simplified Frodo[640,976,1344]), HKDF (SHA-[256,512], HMAC-based), HMAC (SHA-[256,512]), PRF (SHA-[256,512]), SKE (AES-[128,192,256]-CBC).

Dependencies

- Botan: C++ library of cryptographic primitives
- FrodoKEM: C implementation of FrodoKEM, a post-quantum KEM scheme whose security derives from the *learning with error* problem

Contributors

- Yi Tang

Figure: Screenshot of project homepage.

Table of Contents

Signal Protocol

The *Signal protocol* is an open source secure messaging protocol that provides end-to-end authenticated encryption. The protocol is extensively deployed among popular messaging applications e.g.

- Signal (originally TextSecure),
- WhatsApp,
- Google Allo,
- Facebook Messenger,
- Skype, etc.

Generalization of Signal Protocol

- [ACD19] decomposes the *double ratchet algorithm* (core of Signal) into generic cryptographic modules.
- Customize Signal by using different module instances.
- Create *post-quantum* variants of Signal by employing quantum-safe module instances!

Modularization of Signal Protocol

Signal Scheme

Init-A (k)
- id $\leftarrow$ A
- $(k_{\mathsf{root}}, k_{\mathsf{CKA}}) \leftarrow k$
- $\sigma_{\mathsf{root}} \leftarrow$ P-Init(k_{root})
- $(\sigma_{\mathsf{root}}, w_{\mathsf{R}}) \leftarrow$ P-Up$(\sigma_{\mathsf{root}}, \lambda)$
- $\gamma \leftarrow$ CKA-Init-A(k_{CKA})
- $T_{\mathsf{cur}} \leftarrow \lambda$
- $\ell_{\mathsf{prv}} \leftarrow 0$
- $t_{\mathsf{cur}}, i_{\mathsf{S}}, i_{\mathsf{R}} \leftarrow 0$
- $\mathcal{D}[\cdot] \leftarrow \lambda$

skip (t, u)
- **while** $i_{\mathsf{R}} < u$
  - i_{R} ++
  - $(w_{\mathsf{R}}, K) \leftarrow G(w_{\mathsf{R}})$
  - $\mathcal{D}[t, i_{\mathsf{R}}] \leftarrow K$

Send-A (m)
- **if** t_{cur} *is even*
  - t_{cur} ++
  - $\ell_{\mathsf{prv}} \leftarrow i_{\mathsf{S}}$
  - $i_{\mathsf{S}} \leftarrow 0$
  - $(\gamma, T_{\mathsf{cur}}, I) \leftarrow\!\!\$$ CKA-S(γ)
  - $(\sigma_{\mathsf{root}}, w_{\mathsf{S}}) \leftarrow$ P-Up$(\sigma_{\mathsf{root}}, I)$
- i_{S} ++
- $(w_{\mathsf{S}}, K) \leftarrow G(w_{\mathsf{S}})$
- $h \leftarrow (t_{\mathsf{cur}}, i_{\mathsf{S}}, T_{\mathsf{cur}}, \ell_{\mathsf{prv}})$
- $e \leftarrow$ Enc(K, h, m)
- **return** (h, e)

try-skipped (t, i)
- $K \leftarrow \mathcal{D}[t, i]$
- $\mathcal{D}[t, i] \leftarrow \perp$
- **return** K

Rcv-A (c)
- $(h, e) \leftarrow c$
- $(t, i, T, \ell) \leftarrow h$
- **req** t even and $t \leq t_{\mathsf{cur}} + 1$
- **if** $t = t_{\mathsf{cur}} + 1$
  - skip$(t-2, \ell)$
  - t_{cur} ++, $i_{\mathsf{R}} \leftarrow 0$
  - $(\gamma, I) \leftarrow$ CKA-R(γ, T)
  - $(\sigma_{\mathsf{root}}, w_{\mathsf{R}}) \leftarrow$ P-Up$(\sigma_{\mathsf{root}}, I)$
- $K \leftarrow$ try-skipped(t, i)
- **if** $K = \perp$
  - skip$(t, i-1)$
  - i_{R} ++
  - $(w_{\mathsf{R}}, K) \leftarrow G(w_{\mathsf{R}})$
- $m \leftarrow$ Dec(K, h, e)
- **if** $m = \perp$
  - **error**
- **return** (t, i, m)

Figure: The Modularization of Signal protocol proposed by [ACD19], involving "CKA", "P", "G" and ("Enc", "Dec"). (Figure 9 in [ACD19].)

Modularization of Signal Protocol (cont.)

The Signal protocol is decomposed into

- "CKA": *Continuous Key Agreement* (CKA),
- "P": *PRF-PRNG* (PRGF),
- "G": Pseudo-Random Generator (PRG),
- ("Enc", "Dec"): *Authenticated Encryption with Associated Data* (AEAD).

Constructions of Miscellaneous Modules

- CKA from *Key Encapsulation Mechanism* (KEM) ([ACD19]),
- PRGF from Hash-based Key Derivation Function (HKDF) ([ACD19]);
- HKDF from Hash-based Message Authentication Code (HMAC) (e.g. see [KE10]),
- AEAD from Secret/Symmetric Key Encryption (SKE) along with HMAC (e.g. see [McG08]),
- PRG from PRF (trivial according to definitions).

Hierarchical Overview of Implementation

Bottom-up:

- Abundant module instances of CKA, PRGF, PRG, AEAD (and also KEM, HKDF, HMAC, PRF, SKE)
  - where sub-hierarchy exists according to the specification
  - e.g. CKAs can be generically constructed from KEMs
- High-level Signal protocol interface, providing e.g. `send` and `recv` functionalities
- Top-level testing shell, hiding back all Signal-related details

All components are implemented in C.

Examples of Module Instances

Example 1: reconstructing the original Signal Protocol

- CKA: (compressed) Diffie–Hellman with Curve25519
- PRGF: [HKDF-based]
  - HKDF: SHA-256
- PRG: [PRF-based]
  - PRF: SHA-256
- AEAD: AES-128-SIV

Examples of Module Instances (cont.)

Example 2: *post-quantum* variant of the Signal Protocol

- CKA: [KEM-based]
  - KEM: Frodo640[1]
- PRGF: [HKDF-based]
  - HKDF: SHA-256
- PRG: [PRF-based]
  - PRF: SHA-256
- AEAD: AES-128-SIV

[1] FrodoKEM, a post-quantum KEM scheme whose security derives from the *learning with error* problem

Examples of Module Instances (cont.)

Example 3: *improved* post-quantum variant of the Signal Protocol

- CKA: (compressed) Frodo640
- PRGF: [HKDF-based]
  - HKDF: SHA-256
- PRG: [PRF-based]
  - PRF: SHA-256
- AEAD: AES-128-SIV

Time Benchmarking

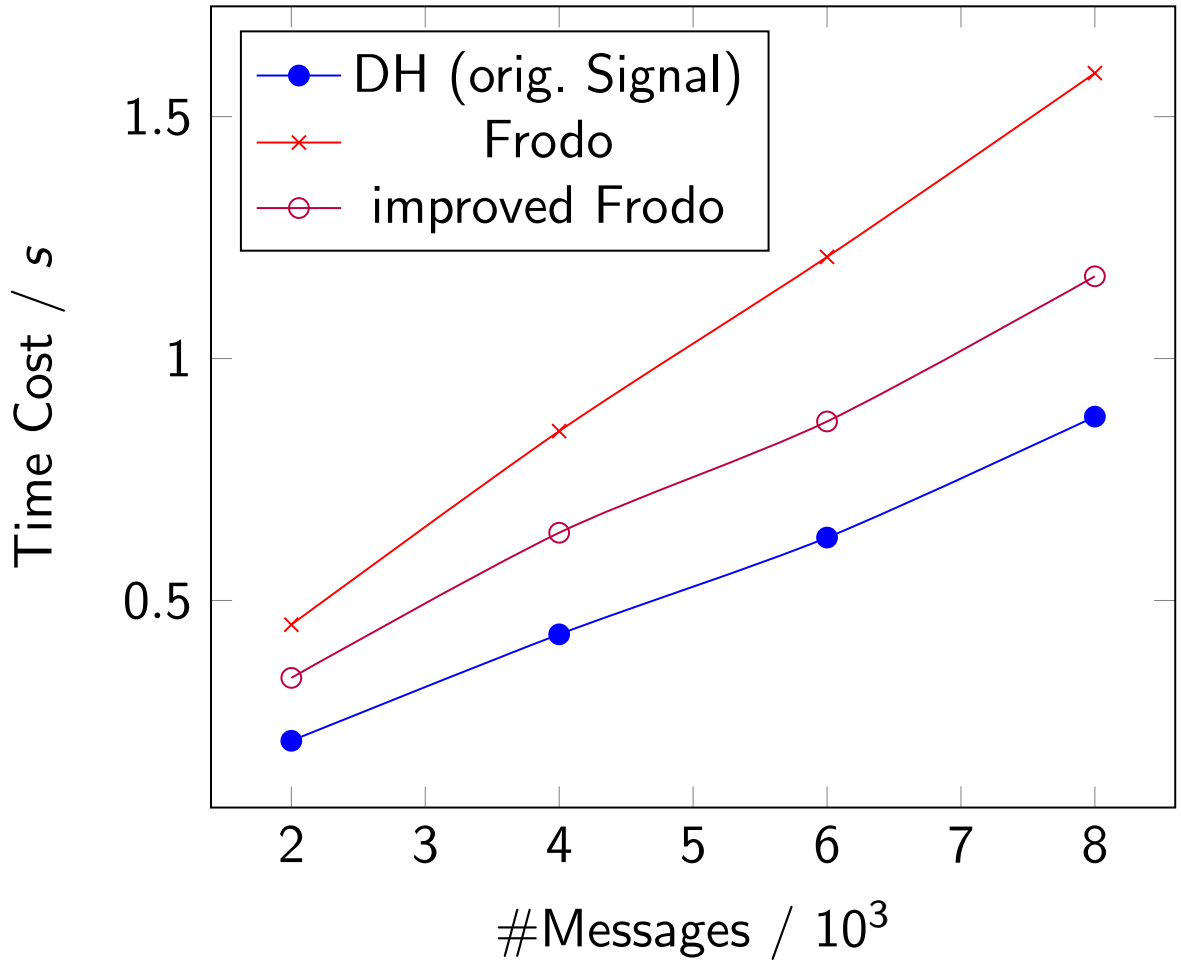


Figure: Comparison of time performance of example Signal variants under random asynchronous messaging test.

Space Benchmarking

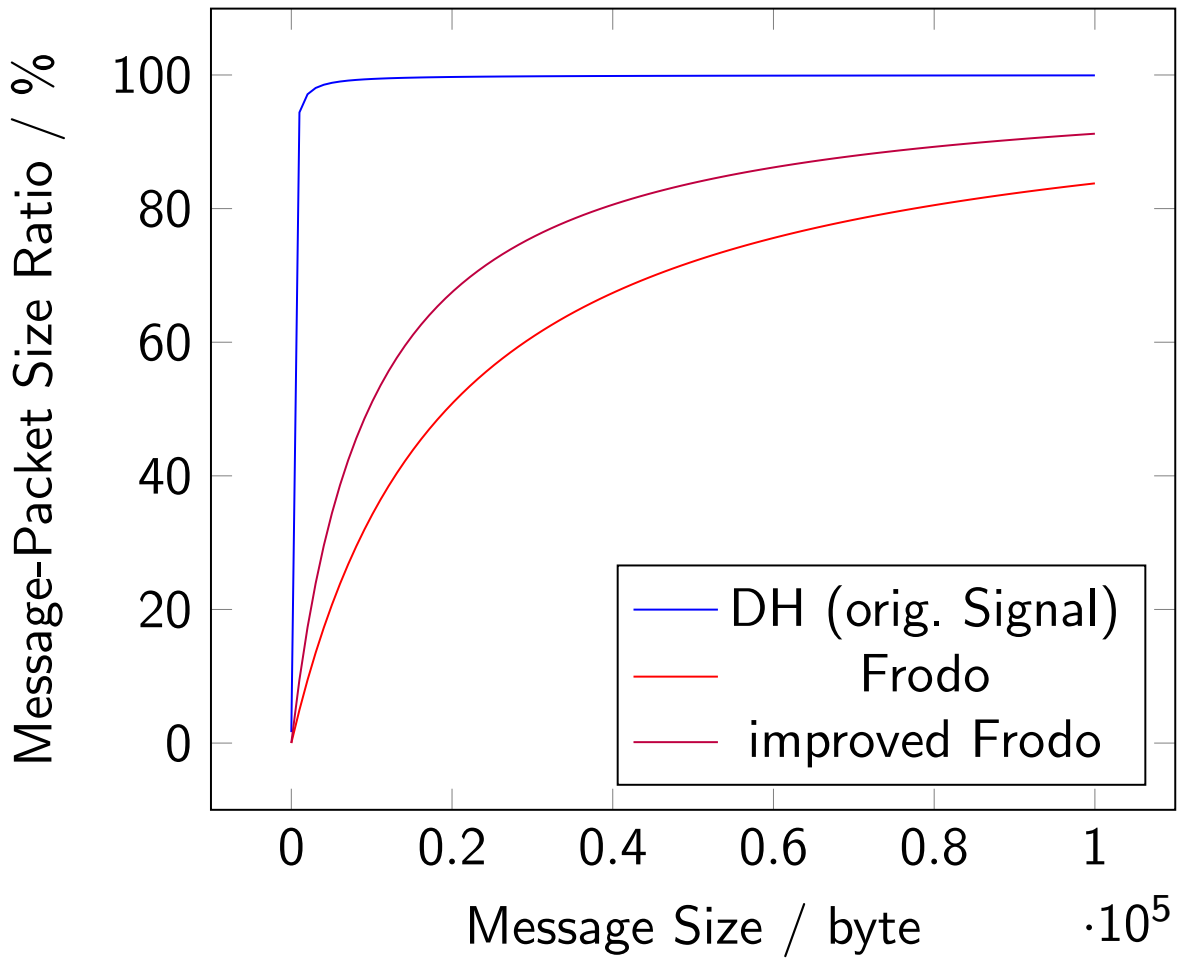


Figure: Comparison of message-packet size ratio of example Signal variants.

References

[ACD19] Joël Alwen, Sandro Coretti and Yevgeniy Dodis, "The Double Ratchet: Security Notions, Proofs, and Modularization for the Signal Protocol", `https://cims.nyu.edu/~dodis/ps/signal.pdf`.

[KE10] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)" (RFC 5869), `https://tools.ietf.org/html/rfc5869`.

[McG08] D. McGrew, "An Interface and Algorithms for Authenticated Encryption" (RFC 5116), `https://tools.ietf.org/html/rfc5116`.


I'm just here
for the signal