Topic 4.3
Applications – Signal and iMessage

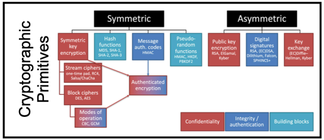
Douglas Stebila

CO 487/687: Applied Cryptography

Fall 2024

UNIVERSITY OF WATERLOO

Applications

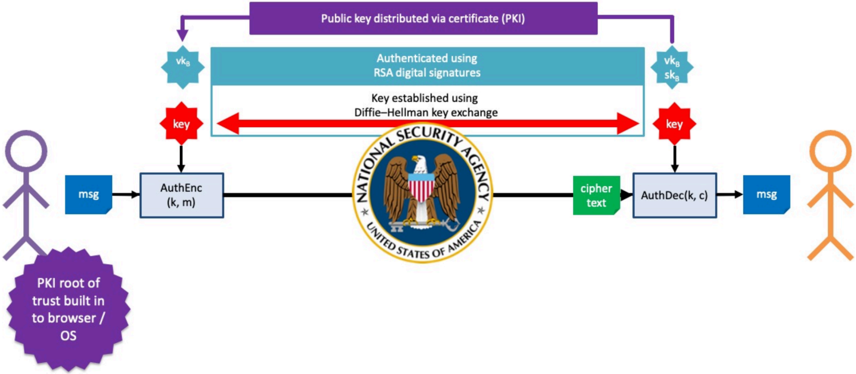


Key management
- PKI
- X.509 certificates

Secure channels
- TLS
- SSH
- Signal

Other applications
- Bitcoin
- Zero knowledge

Secure channels



Outline

Signal

iMessage

Introduction

The Signal Protocol was designed by Moxie Marlinspike and Trevor Perrin.

- Won the 2017 Levchin Prize for contributions to real-world cryptography

It is free, open source, and is used in:

- Signal (free messaging app)
- WhatsApp
- Facebook Messenger (“secret conversations” optional feature)
- Skype (“private conversations” optional feature)
- Google Messages (RCS-based messages)

Interview with Moxie Marlinspike: `https://www.newyorker.com/magazine/2020/10/26/taking-back-our-privacy`

Signal Goals

Participants: Alice, Bob, WhatsApp, ThirdParty (E)

1. Long-lived sessions. Alice and Bob establish a long-lived secure communications session. The session lasts until events such as app reinstall or device change.
2. Asynchronous setting. Alice can send Bob a secure message even if Bob is offline. Messages can be delayed, delivered out of order, or can be lost entirely without problem.
3. Fresh session keys. Each message is encrypted/authenticated with a fresh session key.
    - Encrypt-then-MAC: $c = \text{AES-CBC}_{k_1}(m)$, $t = \text{HMAC}_{h_1}(c)$, where k_1 and h_1 are each 256-bits.

Signal Goals (2)

4. Immediate decryption. Bob can decrypt a ciphertext as soon as he receives it.
5. End-to-end encryption. WhatsApp and E do not possess any of Alice's or Bob's secret keys, nor do they get access to any plaintext.
    - However, WhatsApp (but not E) does get all the metadata, e.g., who sent a message to whom, and when.
6. Forward secrecy. If a party's state is leaked, then none of the previous messages should be compromised (assuming they have been deleted from the state).
7. Post-compromise security. Parties recover from a state compromise (if the attacker remains passive).

Cryptographic Ingredients

1. AES-CBC: 128-bit IV, 256-bit key.
2. HMAC: with SHA256, and a 256-bit key.
3. KDF: A key derivation function (either HMAC or HKDF).
4. Curve25519: See Topic 3.6.
5. Elliptic curve key pairs: (X, x)
   $x \in_R [1, n-1]$ is a secret key
   $X = xP$ is the corresponding public key.
6. ECDH.
7. EdDSA: (an ECDSA-like signature scheme).

Signal Protocol

1. Registration: upload public keys to server
2. Root key establishment: setup initial shared secret between sender and receiver
3. Message transmission: generate encryption keys for messages and update encryption keys using ratchets

Note 1: All of Alice's and Bob's message are sent via WhatsApp.

Note 2: All communication between Alice/Bob and WhatsApp is encrypted/authenticated using a TLS-like protocol.

Registration

1. After Alice has downloaded the WhatsApp app, she sends WhatsApp:
    - ID_A: her identifier (cell phone number)
    - A: her long-term public key
    - U: her medium-term public key
    - $\text{Sign}_A(U)$: her signature on U
    - $S_1, S_2, \ldots, S_\ell$: one-time public keys

    and Alice securely stores her secret keys a, u, $s_1, s_2, \ldots, s_\ell$.

2. Similarly, Bob sends WhatsApp:
    ID_B, B, V, $\text{Sign}_B(V)$, T_1, $T_2, \ldots, T_\ell$.

Root Key Establishment (X3DH handshake)

Alice (initiator) wishes to connect with Bob (responder).

1. Alice $\rightarrow$ WhatsApp: request to create session with Bob.
2. WhatsApp $\rightarrow$ Alice: B, V, $\text{Sign}_B(V)$, T_1 (and deletes T_1).
3. Alice does the following:
   3.1 Verify $(V, \text{Sign}_B(V))$ using B.
   3.2 Select an ephemeral key pair (X_1, x_1).
   3.3 Compute root key $\text{root}_0 = \text{KDF}(aV, x_1B, x_1V, x_1T_1)$.
   (root_0 has bitlength 256 bits).

Note: Given A and X_1, Bob can compute

$$\text{root}_0 = \text{KDF}(vA, bX_1, vX_1, t_1X_1).$$

X3DH handshake

	initiator	responder	intended use
signed prekey	U	V	medium-term, reused across sessions
identity key	A	B	long-term, bound to identity
one-time prekey	X_1	T_1	unique to each session, never reused

Diffie–Hellman shared secrets used in the X3DH root key establishment:

$$\text{root}_0 = \text{KDF}(DH(A, V)), DH(X_1, B), DH(X_1, V), DH(X_1, T_1)).$$

The dashed line is optional: it is omitted from the session key derivation if T_1 is not sent.

Note the asymmetry: when Alice initiates a session with Bob, her signed prekey is not used at all.

Making Signal post-quantum: PQXDH handshake

	initiator	responder	intended use
signed prekey	U	V	medium-term, reused across sessions
identity key	A	B	long-term, bound to identity
one-time prekey	X_1	T_1	unique to each session, never reused
PQ one-time prekey	$PQCT$	$PQPK$	unique to each session, never reused

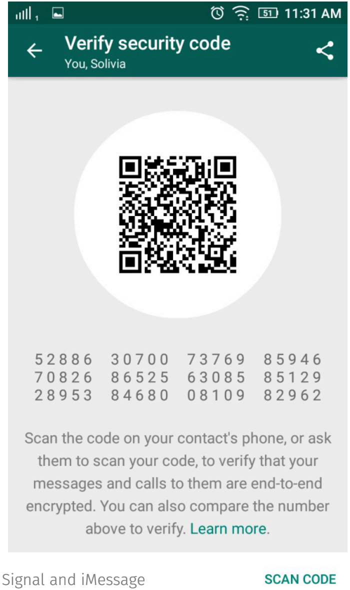
Shared secrets used in the PQXDH root key establishment:

$$\text{root}_0 = \text{KDF}(DH(A, V)), DH(X_1, B), DH(X_1, V), DH(X_1, T_1), PQSS).$$

Verifying Long-Term Public Keys

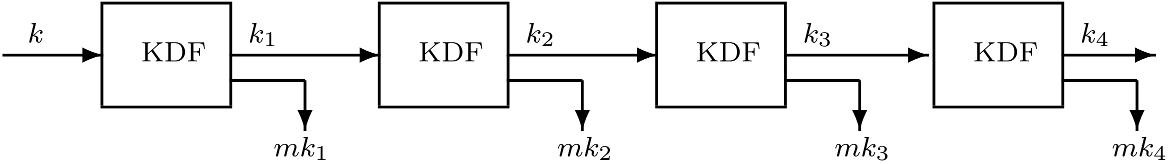
QR codes and 60-digit numbers encode identifiers and long-term public keys; (Alice, A) and (Bob, B).

Alice and Bob should verify these prior to sending each other messages.



Forward Secrecy (“symmetric ratchet”)

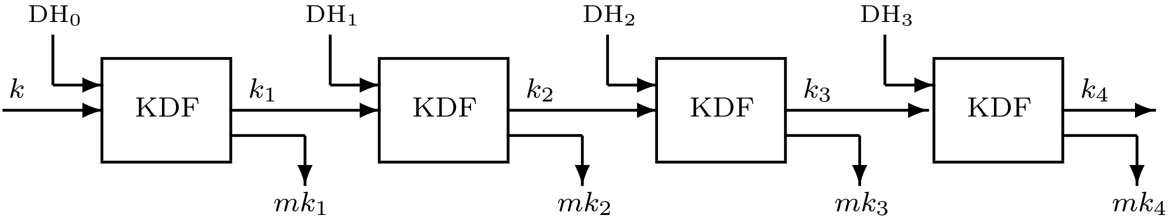
Suppose that Alice and Bob share a secret key k. They can ratchet k and derive message encryption keys $mk_1, mk_2, mk_3 \ldots$ as follows:



- Keys are deleted as soon as they are no longer needed.
- For example, k is deleted as soon as k_1 and mk_1 are computed. Also, mk_1 is deleted as soon as it is used to encrypt (or decrypt) a message.
- Suppose that E learns k_2 and mk_2 (by gaining access to Alice’s device). Then E can compute $k_3, mk_3, k_4, mk_4, \ldots$. But E cannot compute mk_1. Thus, ciphertext that was generated using mk_1 cannot be decrypted by E.

Post-Compromise Security ("asymmetric ratchet")

In order to achieve post-compromise security, a fresh ECDH shared secret established by Alice and Bob is use each time the KDF is applied.



- Here, $\mathrm{DH}_i = \mathrm{ECDH}(X_i, Y_i)$, where X_i is contributed by Alice, and Y_i is contributed by Bob.
- Suppose that E learns k_2 and mk_2. Then E cannot compute k_3, mk_3 unless she also learns x_2 (or y_2).

Message Transmission

Alice maintains three key chains:

1. A root key chain (used to seed the other two chains).
2. A sending key chain (to generate message sending keys).
3. A receiving key chain (to generate message receiving keys).

Bob also maintains three key chains:

1. A root key chain (the same one as Alice's).
2. A receiving key chain (the same as Alice's sending chain).
3. A sending key chain (the same as Alice's receiving chain).

Message Transmission (2)

Consider the following example:

1. Alice $\rightarrow$ Bob: M_{11}^1, M_{11}^2, M_{11}^3, M_{11}^4.
   (Alice's first sending chain of 4 messages)
2. Alice $\leftarrow$ Bob: M_{12}^1, M_{12}^2.
   (Alice's first receiving chain of 2 messages)
3. Alice $\rightarrow$ Bob: M_{22}^1.
   (Alice's second sending chain of 1 messages)
4. Alice $\leftarrow$ Bob: M_{23}^1, M_{23}^2, M_{23}^3
   (Alice's second receiving chain of 3 messages)

Message Transmission (3)

Notation:

- sk = chaining key for sending key chain.
- rk = chaining key for receiving key chain.
- msk = message sending key.
- mrk = message receiving key.

Alice sending M_{ii}^j: $C_{ii}^j = \text{AuthEnc}_{msk_{ii}^j}((A, B, X_i, j, L_{i-1}), M_{ii}^j)$,
where L_{i-1} is the length of Alice's $(i-1)$th sending chain.

Here,

$$\text{AuthEnc}_k(T, M) = \text{AES-CBC}_{IV,k_1}(M), \text{HMAC}_{k_2}(\text{AES-CBC}_{IV,k_1}(M), T), T$$

where $k = (IV, k_1, k_2)$ with $IV \in \{0,1\}^{128}$, $k_1, k_2 \in \{0,1\}^{256}$.

Outline

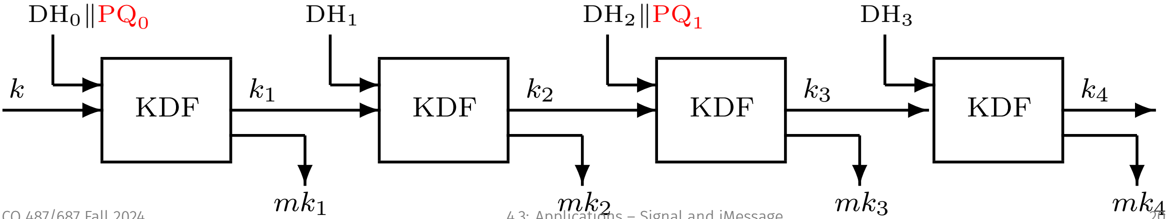
Signal

iMessage

iMessage

Similar high-level structure to Signal, but with some differences:

- **Long-term keys**: signature keys, not Diffie–Hellman keys
- **Root key establishment**:
  - Diffie–Hellman key exchange using one-time prekeys
  - post-quantum (ML-KEM) key exchange using one-time prekeys
- **Symmetric ratchet**: similar
- **Asymmetric ratchet**: post-quantum (ML-KEM) key exchange periodically (but not every message)



References

Signal and WhatsApp:

- Signal technical information: X3DH, PQXDH, Double Ratchet
- Signal Private Group system for group messaging
- Interview with Signal co-inventor Moxie Marlinspike
- Paper on Signal encryption by D. Stebila
- Talk on Signal PQXDH
- WhatsApp Encryption Overview

Apple iMessage PQ3:

- Apple iMessage PQ3 blog post and talk video
- Paper on iMessage PQ3 by D. Stebila

IETF Messaging Layer Security