**Signal Protocol Summary**

A very brief overview of the Signal Protocol.

(disclaimer: I am not a crypto expert, just a guy who has read up a bit on this stuff.)

**What is E2E encryption?**

* a variant of SP used by all standard E2E encrypted messangers
* why is it important?

**How do lesser encryption paradigms fall short?**

* symmetric key exchange
  + how to share key?
* asymmetric key exchange
  + slow, fixed keys
* pretty good privacy (PGP)
  + if single key is broken, entire chat history decryptable
* ephemeral key exchange
  + same key used only for short-duration session
  + forward secrecy: secrets broken in the future don't unlock past contents

**Terminology**

* elliptic curve (EC) keys: assymetric key pair, much stronger than RSA
* elliptic curve Diffie-Hellman (ECDH/DH): how two pairs of EC keys generate a shared secret key
* key derivation function (KDF): can "stretch" or "shrink" high-entropy bytes to yield symmetric key(s)

**What is the "Signal Protocol"?**

* initial session setup via [X3DH](https://whispersystems.org/docs/specifications/x3dh/)
* iterative message key generation via the [double ratchet](https://whispersystems.org/docs/specifications/doubleratchet/)

Also, session management across devices via [Sesame](https://whispersystems.org/docs/specifications/sesame/)

**Session setup**

* goal is for Alice & Bob to share 32-byte secret, used for subsequent message encryption
* Alice & Bob each have a set of (EC) identity key pairs, with public keys published to central server
  + identity (IK): unique & constant for user
  + signed pre-key (SPK) : periodically changing (e.g., weekly/monthly) and signed with identity key
  + one-time pre-keys (OPK): each used only for one session initialization
* Alice initiates session with Bob by generating an ephemeral key pair EK\_a and calculating
  + DH1 = DH(IK\_a, SPK\_b)
  + DH2 = DH(EK\_a, IK\_b)
  + DH3 = DH(EK\_a, SPK\_b)
  + DH4 = DH(EK\_a, OPK\_b)
  + shared key: SK = KDF(DH1 || DH2 || DH3 || DH3)

**Double ratchet**

* goal is for both Alice & Bob to generate same unique encryption key for each message
  + e.g., key(s) for AES256 cipher in GCM mode
  + allows for asynchronous communication
* after establishing session, Alice generates initial root & chain keys from KDF(SK)
* when new message key is needed, "symmetric ratchet" is used
  + Message Key = KDF(Chain Key, constant)
  + Chain Key = KDF(Chain Key, constant)
* ephemeral keys (EK\_a & EK\_b) are replaced after each message round trip: "DH" ratchet
  + new shared secret SK generated from DH keys
  + new root & chain keys generated from KDF(SK)
* "DH" rachet means that SK changes with each round trip communication
  + temporary breaches in SK don't compromise all future communications

**Resources**

(Read these and you'll actually learn how the SP works.)

* [WhatsApp Encryption Overview](https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf)
* [X3DH](https://whispersystems.org/docs/specifications/x3dh/)
* [double ratchet](https://whispersystems.org/docs/specifications/doubleratchet/)
* [Sesame](https://whispersystems.org/docs/specifications/sesame/)
* [Advanced cryptographic ratcheting](https://whispersystems.org/blog/advanced-ratcheting/)
* [A Formal Security Analysis of the Signal Messaging Protocol](https://eprint.iacr.org/2016/1013.pdf)
* [Signal Android Client Code](https://github.com/WhisperSystems/Signal-Android)