Xolotl

Compact mixnet format with hybrid anonymity

Jeff Burdges

28.6.2015

E-mail: Asynchronous messaging

- Email with GnuPG provides authenticity and confidentiality...
- ... but fails to provide forward secrecy, aka key erasure,
- ... and fails to *protect meta-data*

Ratchets provide forward-secrecy

Off-the-Record messaging:
 Rerun DH key exchange ocasionally

Silence Circle's SCIMP:
 Replace our key with its own hash
 No sessions!

Ratchets provide forward-secrecy

Off-the-Record messaging:
Rerun DH key exchange ocasionally

Silence Circle's SCIMP:
Replace our key with its own hash
No sessions!

Axolotl ratchet :
Weld these two together!

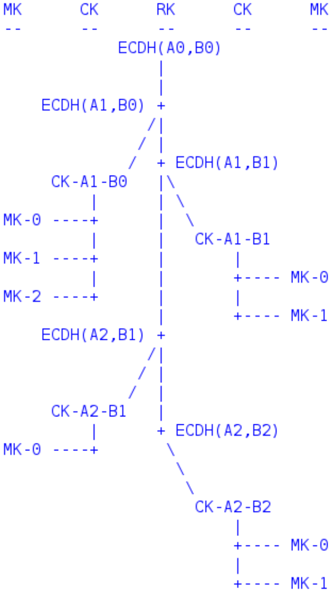
"[Axolotl] combines the .. forward secrecy [of] a hash iteration ratchet like SCIMP [with the] future secrecy .. of a DH ratchet like OtR" — Moxie Marlenspike

Axolotl ratchet by Trevor Perrin and Moxie Marlenspike

Approach:
  Run DH whenever possible
  Iterate key by hashing otherwise

2-step DH is less book keeping than OtR's 3-step DH

Header is one DH public key,
  which one can encrypt.



```
MK      CK      RK      CK      MK
--      --      --      --      --
            ECDH(A0,B0)
                |
                |
    ECDH(A1,B0) +
               /|
              / |
             /  + ECDH(A1,B1)
     CK-A1-B0   |\
         |      | \
MK-0 ----+      |  \
         |      |   CK-A1-B1
MK-1 ----+      |       |
         |      |       +---- MK-0
MK-2 ----+      |       |
                |       +---- MK-1
    ECDH(A2,B1) +
               /|
              / |
             /  |
     CK-A2-B1   |
         |      + ECDH(A2,B2)
MK-0 ----+       \
                  \
                   \
                    CK-A2-B2
                        |
                        +---- MK-0
                        |
                        +---- MK-1
```

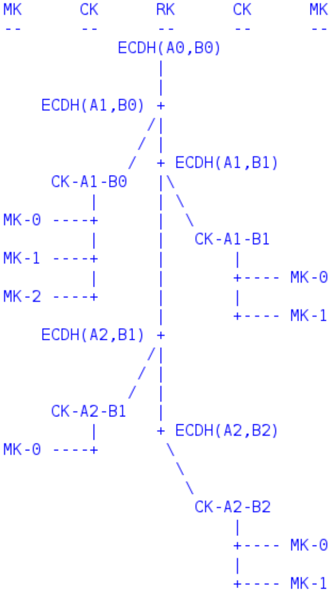
Axolotl ratchet by Trevor Perrin and Moxie Marlenspike

Approach:
 Run DH whenever possible
 Iterate key by hashing otherwise

2-step DH is less book keeping than OtR's 3-step DH

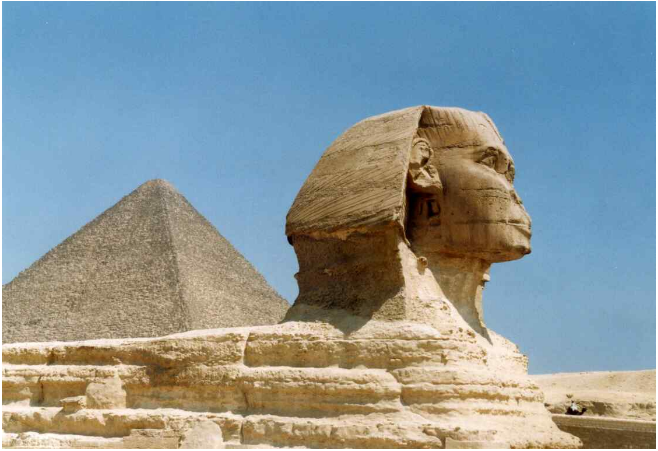
Header is one DH public key,
 which one can encrypt.

Neutral against Shor's algorithm
 running on a quantum computer.



```
MK      CK      RK      CK      MK
--      --      --      --      --
            ECDH(A0,B0)
                |
                |
    ECDH(A1,B0) +
               /|
              / |
             /  + ECDH(A1,B1)
     CK-A1-B0   |\
         |      | \
MK-0 ----+      |  \
         |      |   CK-A1-B1
MK-1 ----+      |       |
         |      |       +---- MK-0
MK-2 ----+      |       |
                |       +---- MK-1
    ECDH(A2,B1) +
               /|
              / |
             /  |
     CK-A2-B1   |
         |      + ECDH(A2,B2)
MK-0 ----+       \
                  \
                   \
                    CK-A2-B2
                        |
                        +---- MK-0
                        |
                        +---- MK-1
```

Sphinx by George Danezis and Ian Goldberg



An asychnornous mixnet allows us defeat corrolation attacks that best Tor

Sphinx by George Danezis and Ian Goldberg

Sphinx is a *compact* packet format for mix networks.

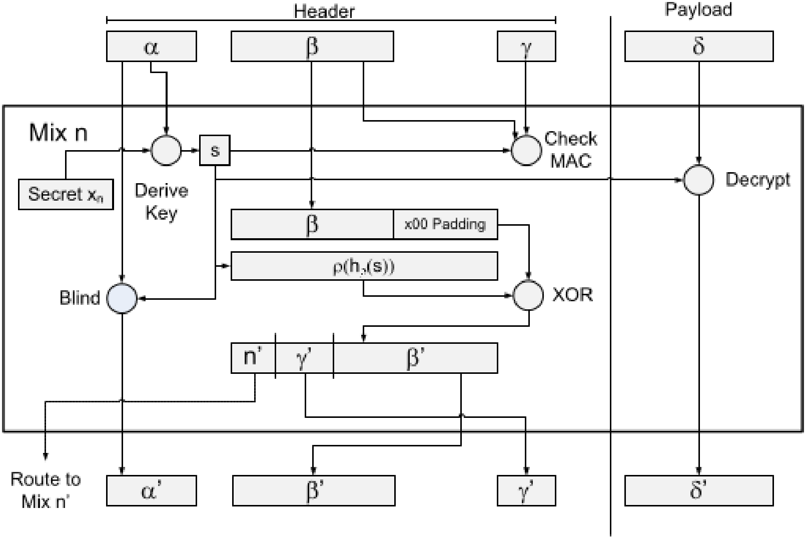
Sphinx is provably secure in the universal composability model [Camenisch & Lysyanskaya '05, Canetti '01]

1. Provides correct onion routing
2. Integrity, meaning immunity to long-path attacks
3. Security, including
   wrap-resistance* and
   indistinguishability of forward and reply messages

   Replay protection implemented by Bloom filter

* Wrap-resistance helps prevent nodes from acting as decryption oracles.

Sphinx by George Danezis and Ian Goldberg



The processing of a Sphinx message $((\alpha, \beta, \gamma), \delta)$ into $((\alpha', \beta', \gamma'), \delta')$

Post-quantum Sphnx?

A quantum computer factored 15 without cheating last year.

That might not sound like much progress for 20 years, but
one should expect slow progress to continue, and
one cannot expect negative results.

We could worry for many decades even if they are impossible!

Post-quantum Sphnx?

We have two seemingly post-quantum key exchanges :

- Ring learning with errors and
- Super-singular isogenies Diffie Hellman

In both cases, we need a blinding operations persumably based on the key exchange operation, but..

- anonymity is far more delicate than cryptography,
- blinding is more fragile than key exchange,
- fewer researchers will ever study blinding, ..

And blinding is weakened by using multiple systems!

Ratchet for Sphinx

Idea : Axolotl is neutral against Shor.

Can we integrate a ratchet with Sphinx?

Axolotl won't work because :

- Relays never message users
- Cannot reuse curve elements

Ideas :

- Relays share new keys with the whole network for replay protection
- Users should learn what messages made it eventually

Xolotl
Sphinx + Axolotl

Relay key replacement

Replay protection requires that relays replace keys regularly.

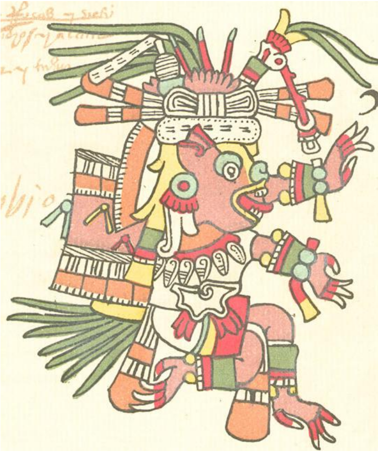
Key lifetime = SURB lifetime

Longer lifetime improves:

- Delivery convenience

Shorter lifetime improves:

- Throughput
- Memory footprint
- Forward-secrecy

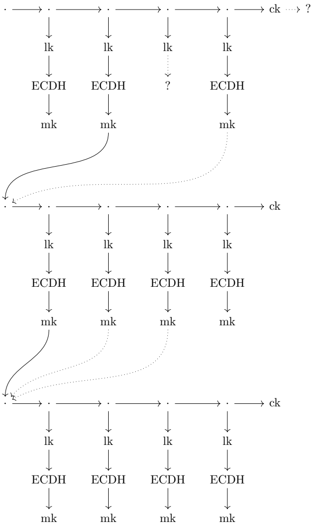


Acknowledging ratchet state

Idea: Client directs ratchet state

- Chain keys evolve like Axolotl, producing leaf keys.
- Create message keys by hashing a leaf key with a Sphinx ECDH

$$\mathrm{mk} = H(\mathrm{lk}, H'(\mathrm{ECDH}(\mathrm{u},\mathrm{r})))$$



Acknowledging ratchet state

Idea: Client directs ratchet state

Chain keys evolve like Axolotl, producing leaf keys.

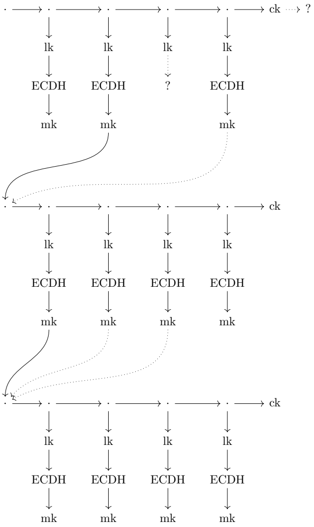
Create message keys by hashing a leaf key with a Sphinx ECDH

$$\mathrm{mk} = H(\mathrm{lk}, H'(\mathrm{ECDH}(\mathrm{u}, \mathrm{r})))$$

Packets identify the message key from which their chain started.

And their leaf key sequence no.

And parent max sequence no.



Wait. Aren't ratchets only pseudononymous?

We cannot use the Xolotl ratchet for every mixnet hop, but the ratchet should be suitable for certian situations.

Guard nodes can use a session ratchet initialized with:

- post-quantum key exchange, or
- another longer term ratchet, maybe.

Third hop out of a five hope circut:

- Long-term ratchet is okay, but only pseudonymous
- Initializing from a longer term ratchet is okay

Other hops require greater care

$$\text{User} \to \text{Guard} \to \text{Anon} \to \text{Pseudo} \to \text{Anon} \to \text{Cross} \to \cdots$$