INVENTORS FOR THE DIGITAL WORLD

Sphinx

Should we start pre-standards work for a compact and provably secure packet format?

Harry Halpin, NEXTLEAP

What is Sphinx?

Privacy-Preserving Packets: **Coming out of anonymous communications research community**

- **Unlinkable**
- **Same size (unlike normal onion-wrapping)**
- **Routing Information Private**

Design Paper with Security Proofs: **Sphinx: A Compact and Provably Secure Mix Format** IEEE Security and Privacy 2009 by Ian Goldberg and George Danezis

Has withstood test of time: **All alternative proposals have shown to have security or privacy shortcoming, basic design unchanged after 10 years.**

Cryptographic Intuition

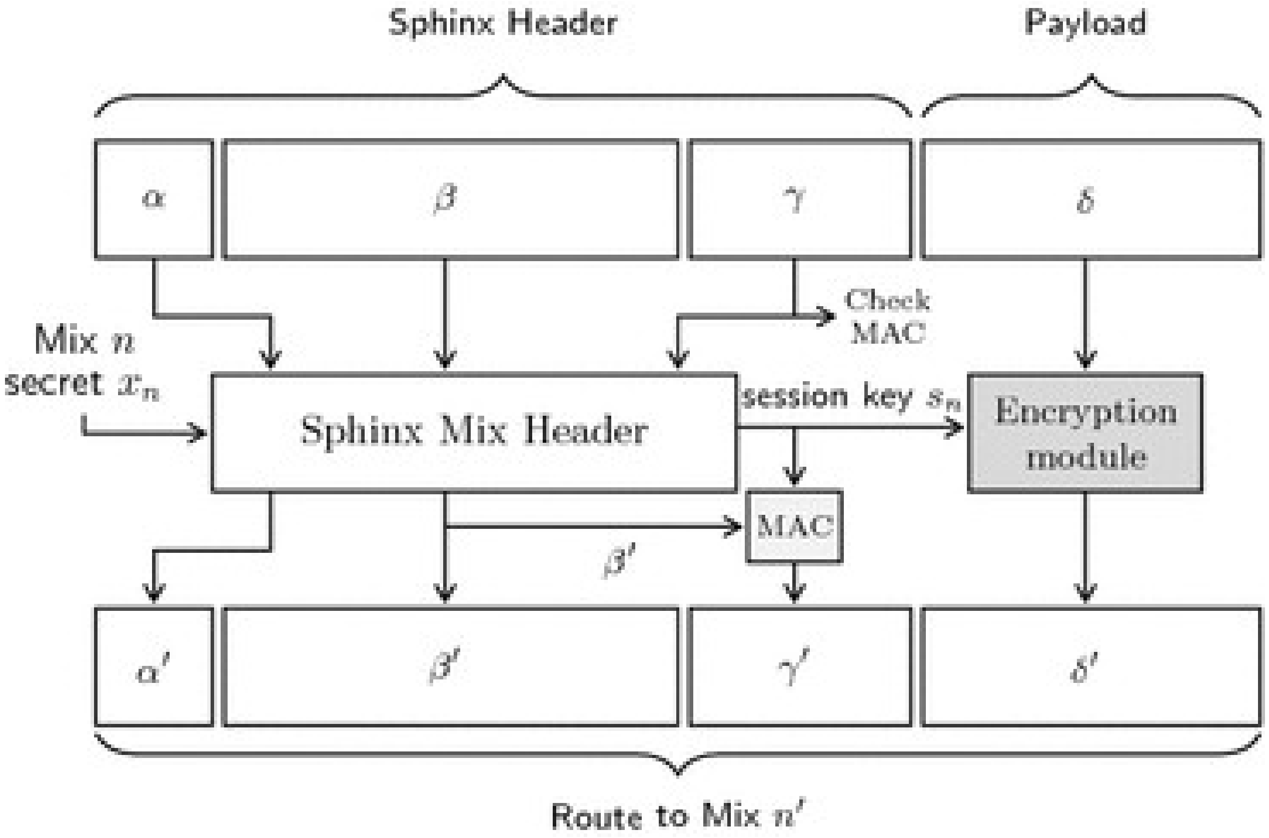
All packets must be same size: Use a wide blockcipher (to resist tagging attacks) and specify padding.

“Extended” Diffie-Hellman over a Network: Client creates the network path and a Diffie Hellman with a recipient. The curve point/group plus Diffie-Hellman with recipient then allows a point-to-point Diffie Hellman to be established for path that reveals only routing information to next hop.

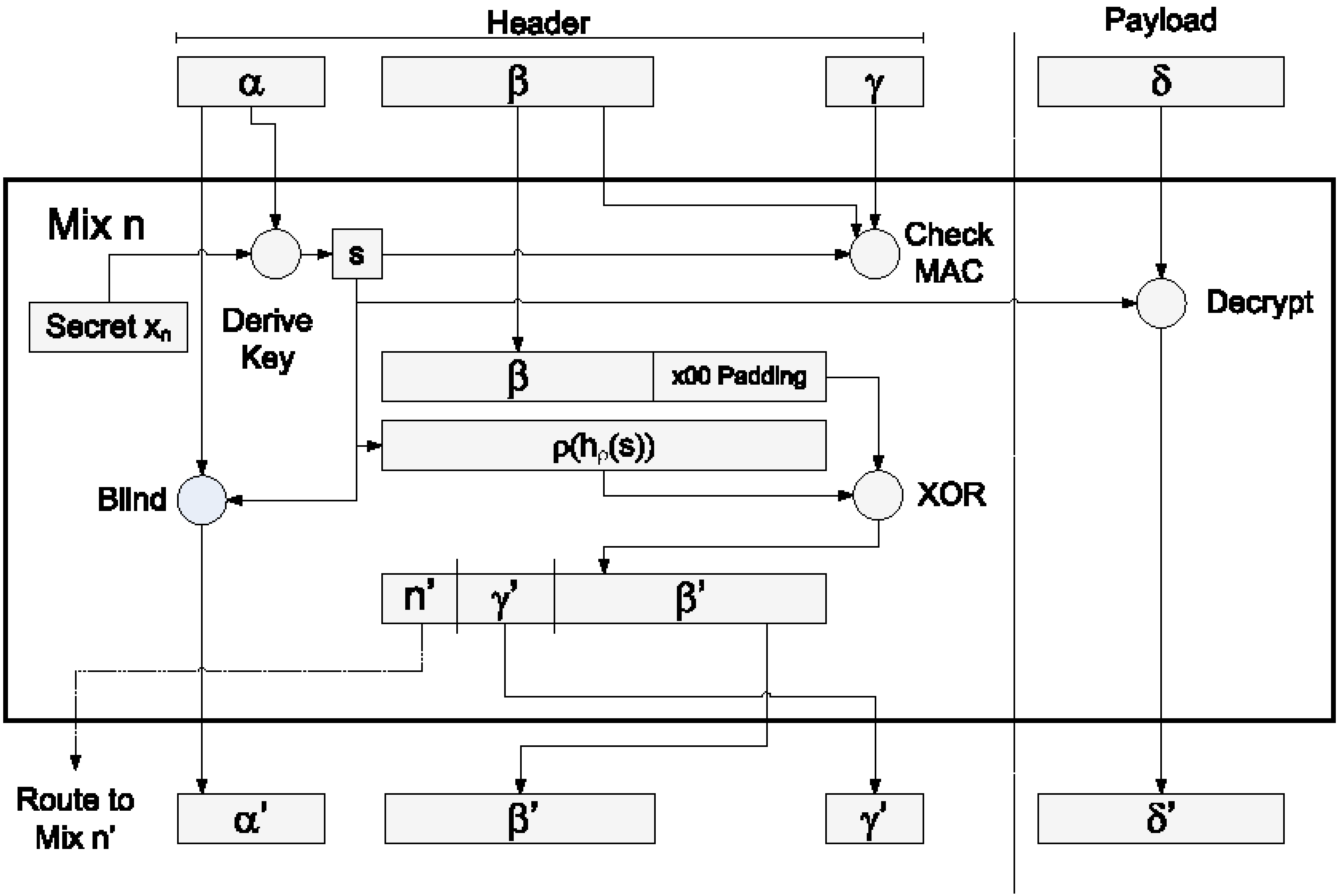
Encrypt-Then-Mac with De-blinding: At each hop, the “next hop” is deblinded.

Only at delivery is the payload unencrypted.

How Sphinx Works



How Sphinx Works



Use Cases

Blockchain Technology: **Lightning Network** uses Sphinx for routing direct p2p payment information. Released May 2018, > 1000 nodes carrying large amounts of Bitcoin traffic (Stellar, etc. also adopting)

Messaging System: **Next generation Bitmask messaging client uses Sphinx**, estimated 200,000 users with a test-net.

E-voting System: **State of Greece** (GR.NET) deploying and Estonia has interest in using for e-voting.

Privacy-preserving Statistics: **SAP** is using with privacy-enhanced GDPR compliant stats.

Why Standardize?

Lots of incompatible libraries: Maintenance of a core Python library with test-suite (with Java and Javascript compatibile libraries), incompatible versions in Go, C, Python – including Lighting Network incompatibilities.

Due to Incompatible Extensions: The issue is routing information is assumed constant length, not the case in real networks.

Due to Changing Cryptographic Requirements: Original paper had both RSA and Elliptic Curve versions, incompatible curves now being used. Wide blockcipher (LIONESS, AEZ, etc.) need standard (CFRG?)

Efficiency: Issues with optimizations for trusted setup

Work on a Spec has Begun

Reference Implementation:

https://github.com/UCL-InfoSec/sphinx

Draft Specification:

https://github.com/katzenpost/docs/blob/master/specs/sphinx.txt

Get in touch: harry.halpin@inria.fr

Mixnetworks.org

Panoramix-project.eu