Cryptographic directions in Tor

Nick Mathewson
nickm@torproject.org

6 Jan 2016
Outline

• Where we started
• Where we are
• Where we're going – maybe.
Let's oversimplify Tor, in 1 slide.
We chose some reasonable-looking crypto in 2004...

- Relay encryption: AES-CTR + Truncated SHA1
  - End-to-end only

- Key negotiation: “TAP”.
  - (RSA1024 + DH1024 + AES-CTR)

- Links: TLS1.0
  - With DH1024, RSA1024, AES-CBC, SHA1.
...and we've replaced a lot of it...

- **Relay encryption:** AES-CTR + Truncated SHA1
  - End-to-end only

- **Key negotiation:** “TAP” “ntor”
  - (RSA1024 + DH1024 + AES-CTR)
  - Curve25519 + SHA256

- **Links:** TLS1.0
  - With DH1024, RSA1024, AES-CBC, SHA1.
...and we've replaced a lot of it...

- Relay encryption: AES-CTR + Truncated SHA1
  - End-to-end only

- Key negotiation: “TAP” “ntor”
  - (RSA1024 + DH1024 + AES-CTR)
  - Curve25519 + SHA256

- Links: TLS1.0 TLS >= 1.0...
  - With DH1024, RSA1024, AES-GCM, SHA1.
  - With ECDH (P256), RSA1024, AES-GCM
But work remains!

- Relay encryption: AES-CTR + Truncated SHA1
  - End-to-end only
  - Too Malleable!

- Key negotiation: “TAP” “ntor”
  - (RSA1024 + DH1024 + AES-CTR)
  - Curve25519 + SHA256
  - Not Postquantum Enough!

- Links: TLS1.0 TLS >= 1.0...
  - With DH1024, RSA1024, AES-GCM, SHA1.
  - With ECDH (P256), RSA1024, AES-GCM
  - Just no.
Malleable AES-CTR + end-to-end MAC allows tagging attacks.

XORs data into ciphertext

Recovers plaintext, and finds data.
Solution: Add a MAC at each hop?

Alice

Evil R1

XORs data into ciphertext

R2

Evil R3

Bob

Rejects ciphertext.

Observes: Circuit closed.
Solution: Add a MAC at each hop?

Alice

Evil R1

R2

Evil R3

XORs data into ciphertext

Rejects ciphertext.

Bob
But that leaks path length/position.
Solution: Chained wide-block SPRP?

Alice

Evil R1

R2

Evil R3

Garbled ciphertext

Observes: Circuit garbled.

Bob

XORs data into ciphertext
Single anonymity tool seeks SPRP for good times, encryption.

- **AEZ? (rogaway et al)**
  - CAESAR candidate
  - Based on AES round function—complex.
  - Fast with AESNI; less so if not??

- **HHFHFH? (djb et al)**
  - Feistel construction: simple, has proofs.
  - Instantiate with GF25519 / XChaCha20?
  - Slower than AEZ?? Need more data!

- Help?
Also let's do PQ circuit extension!

- Forward secrecy matters most.
- Needs to be fast-ish and small-ish.
- No less secure than current ntor approach.
  
  (approximately:)
  
  - Alice $\rightarrow$ Bob: “$g^x$, Bob.”
  - Bob $\rightarrow$ Alice: “$g^y$, H1($g^xy$, $g^xb$...)”.
  - Keys are: KDF($g^xy$, $g^xb$...)
Also let's do PQ circuit extension!

- Forward secrecy matters most.
- Needs to be fast-ish and small-ish.
- No less secure than current ntor approach.
  
  (approximately:)
  
  - Alice → Bob: “g^x, Bob, PQKey ”
  - Bob → Alice: “g^y, H1(g^xy, g^xb....), E(PQKey, N)”. 
  - Keys are: KDF(g^xy, g^xb, N ....)
Current candidates

- ntru?
- newhope?
- ___________?
Questions?

- Also see tor-dev mailing list for more discussion!

- Targeting 2016 deployment.

- Also, ask me about hidden service crypto.