A quantum-safe circuit-extension handshake for Tor

John M. Schanck[1,2] William Whyte[1] Zhenfei Zhang[1]

Security Innovation, Wilmington, MA 01887, USA

University of Waterloo, Waterloo, Canada

April 2015

Outline

1. Key exchange protocols
2. Tor networks
3. The ntrutor protocol
4. A few other results

Key Exchange Protocols

Diffie-Hellman key exchange

$\hat{A}$		$\hat{B}$
$(x, X = g^x) \leftarrow \mathsf{DHGen}(1^\lambda)$		$(y, Y = g^y) \leftarrow \mathsf{DHGen}(1^\lambda)$
	$\xrightarrow{X}$	
	$\xleftarrow{Y}$	
$K = Y^x$		$K = X^y$

- Allow two users to agree keying material without an existing shared secret;

Key Exchange Protocols

KEA+: two way authenticated key exchange

$\hat{A} : (a, A = g^a)$	$\hat{B} : (b, B = g^b)$
$(x, X = g^x) \leftarrow \mathsf{DHGen}(1^\lambda)$	$(y, Y = g^y) \leftarrow \mathsf{DHGen}(1^\lambda)$
$\xrightarrow{X}$	
$\xleftarrow{Y}$	
$K = \mathsf{KDF}(Y^a \vert B^x \vert \hat{A} \vert \hat{B})$	$K = \mathsf{KDF}(A^y \vert X^b \vert \hat{A} \vert \hat{B})$

- Static (authenticated) keys: (a, A), (b, B)
- Ephemeral keys: (x, X), (y, Y)
- KDF: Key derivation function.

Key transport

Key Encapsulation Mechanism (KEM)

$\hat{A}$:		$\hat{B}$: (b, B)
$m \leftarrow_R \mathbb{M}$		
$c \leftarrow \text{Encrypt}_B(m)$		
	$\xrightarrow{c}$	
		$m = \text{Decrypt}_B(c)$
$K = \text{KDF}(m)$		$K = \text{KDF}(m)$

- One-way authentication and one-way anonymity
- No forward secrecy

Background

The Tor anonymity network

- Allows for improved privacy for Internet connections web browsing

- Run through a series of volunteer relays
- Increased interest following YOU-KNOW-WHO revelations

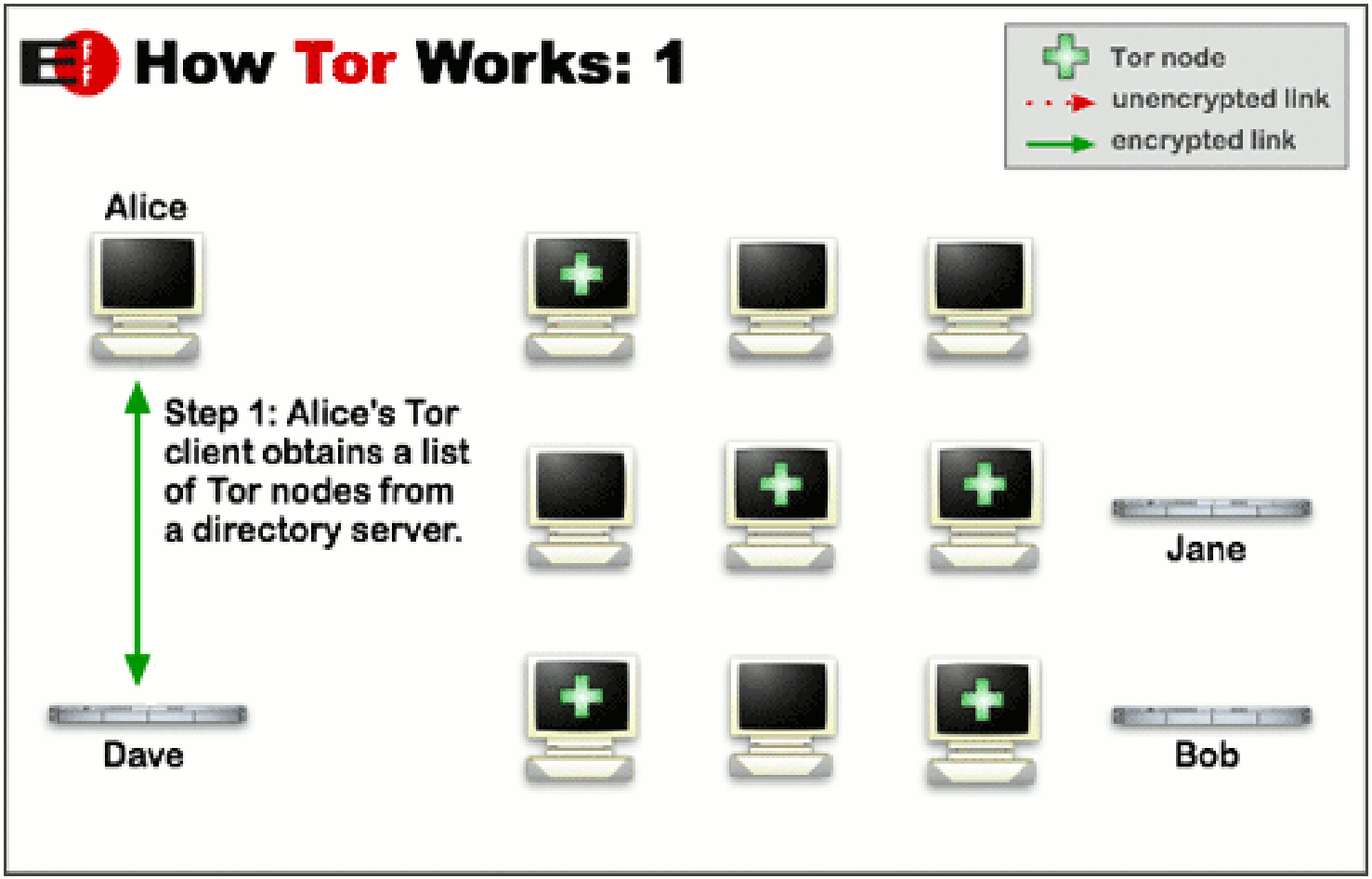

Figure source: torproject.org

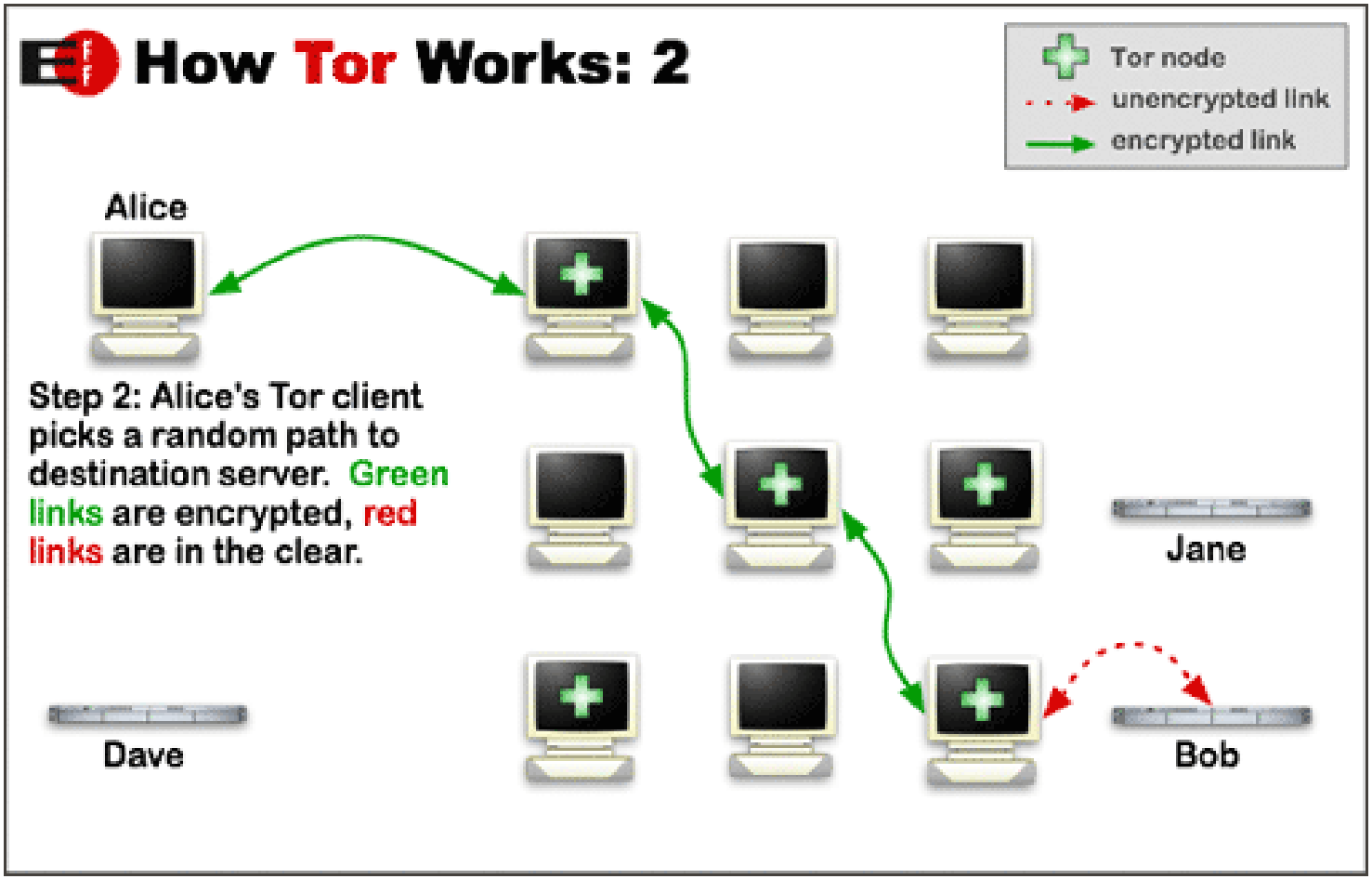

Figure source: torproject.org

Tor handshake

- The clients need to remain anonymous;
- The servers need to be authenticated;
- Originally uses Tor Authentication Protocol (TAP);
- Current best practice: ntor protocol;

ntor protocol

$\hat{A}$:		$\hat{B} : (b, B)$
$(x, X) \leftarrow \mathsf{DHGen}(1^\lambda)$		$(y, Y) \leftarrow \mathsf{DHGen}(1^\lambda)$
	$\xrightarrow{X}$	
		$s_1 = X^y \vert X^b$
		$(vk, K) = \mathsf{H}_1(s_1 \vert \hat{B} \vert X \vert Y)$
		$auth = \mathsf{H}_2(vk \vert \hat{B} \vert Y \vert X)$
	$\xleftarrow{Y, auth}$	
$s_1 = Y^x \vert B^x$		
$(vk, K) = \mathsf{H}_1(s_1 \vert \hat{B} \vert X \vert Y)$		
ensure $auth = \mathsf{H}_2(vk \vert \hat{B} \vert Y \vert X)$		

A need for quantum-safe handshake

- If you send something now
  - Encrypted with an algorithm that's later broken
  - And someone has stored your message
  - They can decrypt it
- Encryption needs to take into account the lifetime for which your data might remain sensitive
- Attacker who doesn't actively get involved at the time of the interaction, but passively records traffic for later analysis
- Fits known attacker pattern

A need for quantum-safe handshake

- If you send something now
  - Encrypted with an algorithm that's later broken
  - And someone has stored your message
  - They can decrypt it
- Encryption needs to take into account the lifetime for which your data might remain sensitive
- Attacker who doesn't actively get involved at the time of the interaction, but passively records traffic for later analysis
- Fits known attacker pattern

The ntrutor protocol

$\hat{A}$:
$(x, X) \leftarrow \mathsf{DHGen}(1^\lambda)$
$(pk_N, sk_N) \leftarrow \mathsf{NTRUGen}(1^\lambda)$

$\hat{B} : (b, B)$
$(y, Y) \leftarrow \mathsf{DHGen}(1^\lambda)$

$X, pk_N \longrightarrow$

$s_1 = X^y | X^b$
$s_2 \leftarrow_R \mathbb{M}$
$c \leftarrow \mathsf{NTRUEnc}(s_2, pk_N)$
$(vk, K) = \mathsf{H}(s_1 | \hat{B} | X | Y | s_2 | pk_N)$
$auth = \mathsf{H}_{\mathsf{mac}}(vk | \hat{B} | Y | X | c | pk_N)$

$\longleftarrow Y, c, auth$

$s_1 = Y^x | B^x$
$s_2 = \mathsf{NTRUDec}(c, sk_N)$
$(vk, K) = \mathsf{H}(s_1 | \hat{B} | X | Y | s_2 | pk_N)$
ensure $auth = \mathsf{H}_{\mathsf{mac}}(vk | \hat{B} | Y | X | c | pk_N)$

$\hat{A}$:		$\hat{B} : (b, B)$
$(x, X) \leftarrow \mathsf{DHGen}(1^\lambda)$		$(y, Y) \leftarrow \mathsf{DHGen}(1^\lambda)$
$(pk_N, sk_N) \leftarrow \mathsf{NTRUGen}(1^\lambda)$		
	$\xrightarrow{X, pk_N}$	
		$s_1 = X^y \vert X^b$
		$s_2 \leftarrow_R \mathbb{M}$
		$c \leftarrow \mathsf{NTRUEnc}(s_2, pk_N)$
		$(vk, K) = \mathsf{H}(s_1 \vert \hat{B} \vert X \vert Y \vert s_2 \vert pk_N)$
		$auth = \mathsf{H}_{\mathsf{mac}}(vk \vert \hat{B} \vert Y \vert X \vert c \vert pk_N)$
	$\xleftarrow{Y, c, auth}$	
$s_1 = Y^x \vert B^x$		
$s_2 = \mathsf{NTRUDec}(c, sk_N)$		
$(vk, K) = \mathsf{H}(s_1 \vert \hat{B} \vert X \vert Y \vert s_2 \vert pk_N)$		
ensure $auth = \mathsf{H}_{\mathsf{mac}}(vk \vert \hat{B} \vert Y \vert X \vert c \vert pk_N)$		

ntrutor protocol

- Efficiency
- Provable security
  - eCK model with extension to passive quantum attackers
  - multi CCA model
- Forward secrecy
- Reuse existing design as much as possible
- Easy to migrate to
- Add no identifiers

SecurityInnovation® EMBEDDED SECURITY BUSINESS UNIT

	TAP	ntor	ntrutor
client → server bytes	186	84	693
server → client bytes	148	64	673
client computation (stage 1)	280 μs	84 μs	272 μs
server computation	771 μs	263 μs	307 μs
client computation (stage 2)	251 μs	180 μs	223 μs
total computation time	1302 μs	527 μs	802 μs
% client	40.8%	50.1%	61.7%

Table : Performance comparison of TAP, ntor, and ntrutor

- We have a prototype implementation as a proof of concept
- Available on github: https://github.com/NTRUOpenSourceProject/ntru-tor
- But this can't simply be implemented as written
  - Tor packets are limited to 512 bytes
  - Tor handshake messages are limited to one packet
- Solutions:
  - Change one of the above
  - Both have been discussed within the Tor project in other contexts

A few other results

Quantum-Safe Hybrid (QSH) Ciphersuite for TLS

INTERNET-DRAFT
Intended Status: Experimental
Expires: <Expiry Date>

J. M. Schanck
Security Innovation & U. Waterloo
W. Whyte
Security Innovation
Z. Zhang
Security Innovation
March 2015

Quantum-Safe Hybrid (QSH) Ciphersuite
for Transport Layer Security (TLS)

Abstract

This document describes the Quantum-Safe Hybrid ciphersuite, a new ciphersuite providing modular design for quantum-safe cryptography to be adopted in the handshake for the Transport Layer Security (TLS) protocol. In particular, it specifies the use of the NTRUEncrypt encryption scheme in a TLS handshake.

SecurityInnovation® EMBEDDED SECURITY BUSINESS UNIT

NTRU

Quantum-Safe Hybrid (QSH) Ciphersuite for TLS

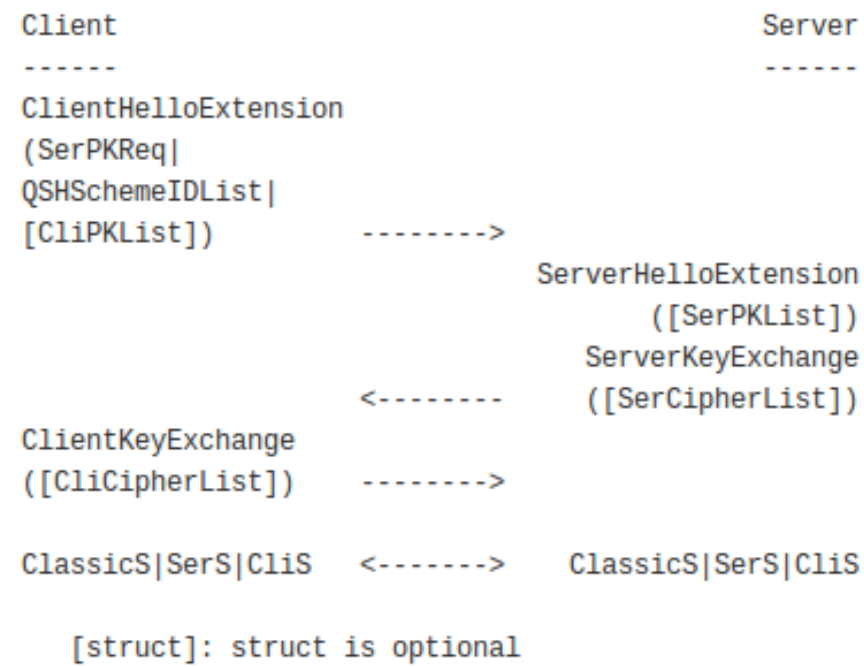

Figure 2: Additional cryptographic data in TLS handshake

- QSHSchemeIDList = {[NTRU]|[LWE]|[HFE]| ...}
- CliPKList = {[*NTRUcpk*]|[*LWEcpk*]| [*HFEcpk*]| ...}
- Server chooses NTRU and HFE
- SerCipherList = {[$\text{Enc}_{NTRUcpk}$(SerS1)]| [Enc_{HFEcpk}(SerS2)]| ...}

Quantum-Safe Hybrid (QSH) Ciphersuite for TLS

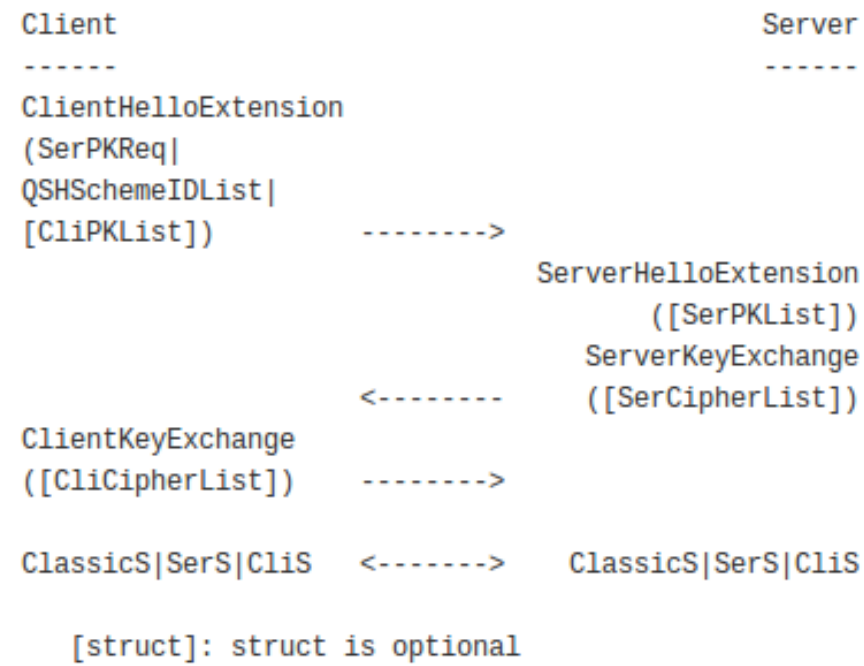

Figure 2: Additional cryptographic data in TLS handshake

- Depending on SerPKReq, server may include a list of his pks
- SerPKList = $\{[NTRUspk]|\ [HFEspk]|\ \dots\}$
- Client encapsulates his secret with server's pks
- Final premaster secret:

  ClassicS|SerS1|SerS2|...CliS1|CliS2|...

An update on NTRU challenge

NTRU Challenge

Calling All Cryptographers!

Security Innovation is pleased to present the NTRU Challenge, which has been created to increase the world's understanding of the shortest vector problem in NTRU lattices, and to encourage and stimulate further research in the security analysis of NTRU-based cryptosystems. We hope it will provide additional information to users of NTRU public-key cryptosystems to help them select suitable key lengths for a desired level of security.

The Challenge is to compute the NTRU private keys from the given list of public keys and associated system parameters. This is the type of problem a hacker faces who wishes to defeat an NTRU-based cryptosystem. There will be several challenges comprised of a number of security levels, some of which can be solved in a day, some in a few months and some which are considered to be computationally intractable.

Click here for a complete set of rules.

Click for Challenges

Machine readable files can be found here.

Over $90,000 in potential Cash Prizes!

The first correct submission for each NTRU Challenge will receive the following prizes (see contest rules for details)

- **NTRU Challenge N=107 through N=211 (11 Challenges total)** - Prize of $1,000 for **each** Challenge
- **NTRU Challenge N=227 through N=401 (16 Challenges total)** - Prize of $5,000 for **each** Challenge
- **NTRU Hall of Fame** - Innovative and Unique solutions may be chosen for the NTRU Hall of Fame and a trip to a major Cryptography Conference, including Airfare, Hotel and Conference Registration fees

- Given an NTRU public key, find the secret key.
- Some are "easy" to solve.
- Two challenges ($N = 107, 113$) have been solved so far, using BKZ.

SecurityInnovation® EMBEDDED SECURITY BUSINESS UNIT

NTRU