The Quantum Menace

By Collin Berman and Reid Bixler

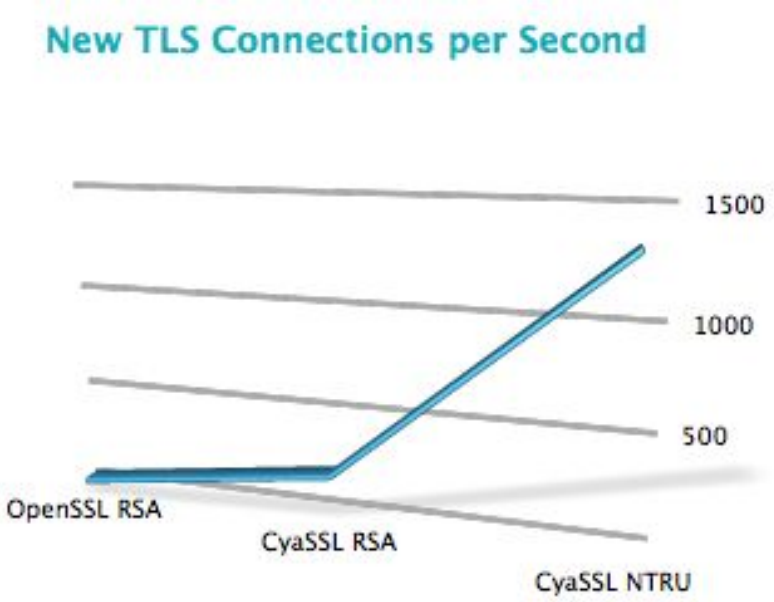
Refresher

- Many schemes resist attacks from quantum computers
  - Secret-key cryptography
  - Lattice-based cryptography
  - Hash-based cryptography
  - Code-based cryptography
  - Multivariate-quadratic-equations cryptography
- “We focus our work on the key exchange component, not authentication: we assume that a quantum computer does not currently exist so that the standard RSA-based authentication in TLS is secure for now“

wolfSSL

SUPPORTED ALGORITHMS

- Key exchange RSA, DSS, DH, NTRU
- Bulk encryption DES, 3DES, AES, ARC4, RABBIT, HC-128
- MAC: MD2, MD5, SHA-1, SHA-512, RIPEMD



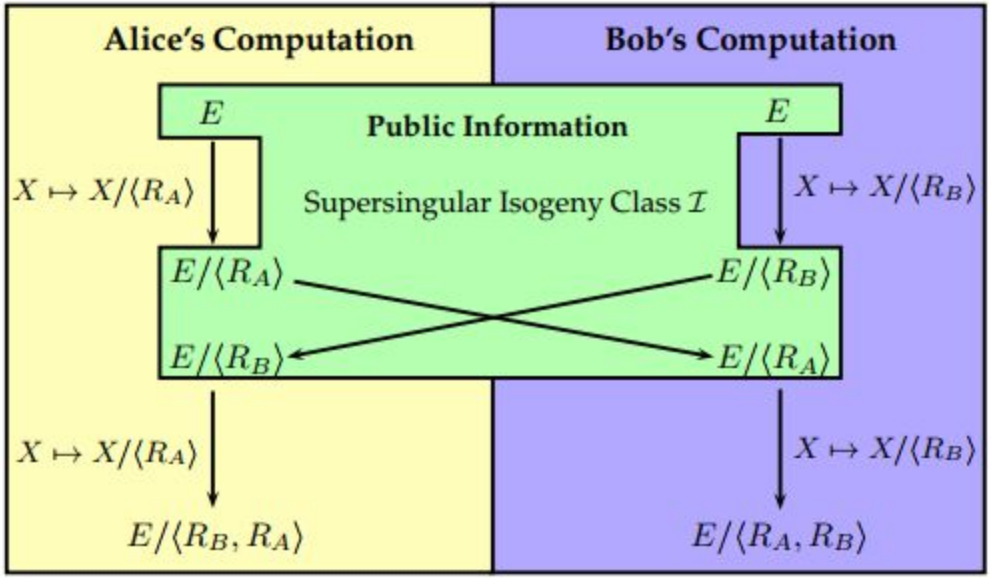
NTRU Leaks!

Table 1. Experiments on NTRUSIGN-251 without perturbation, using NTRU symmetries.

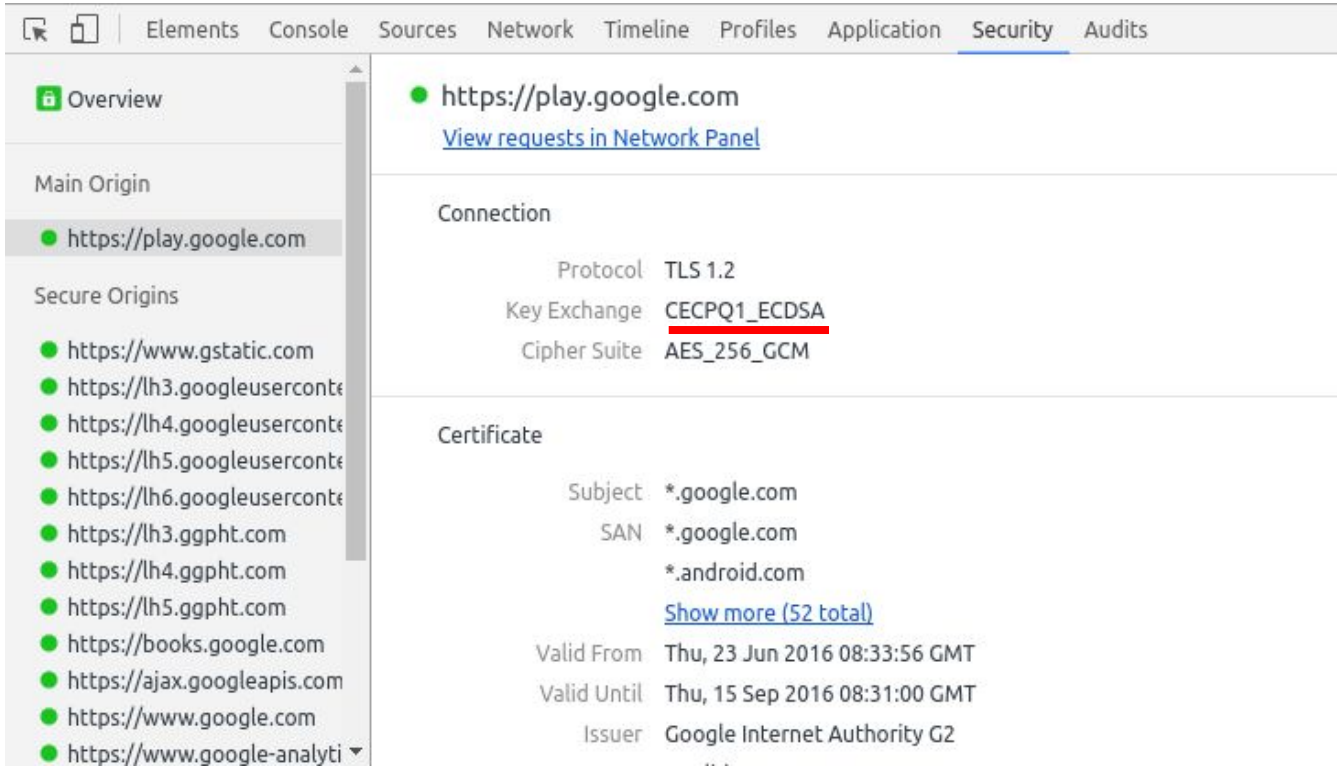
Number of signatures	Expected number of descents to recover the secret key
1,000	2
500	40
400	100

is indeed the case in practice (see Table 1): as few as 400 signatures are enough in practice to recover the secret key, though the corresponding 100,400 parallelepiped samples are not independent. This means that the previous number of 90,000 signatures required by the attack can be roughly divided by $N = 251$. Hence, NTRUSIGN without perturbation should be considered totally insecure.

SIDH


Alice's Computation
Bob's Computation
E
Public Information
E
$X \mapsto X/\langle R_A \rangle$
Supersingular Isogeny Class $\mathcal{I}$
$X \mapsto X/\langle R_B \rangle$
$E/\langle R_A \rangle$
$E/\langle R_B \rangle$
$E/\langle R_B \rangle$
$E/\langle R_A \rangle$
$X \mapsto X/\langle R_A \rangle$
$X \mapsto X/\langle R_B \rangle$
$E/\langle R_B, R_A \rangle$
$E/\langle R_A, R_B \rangle$

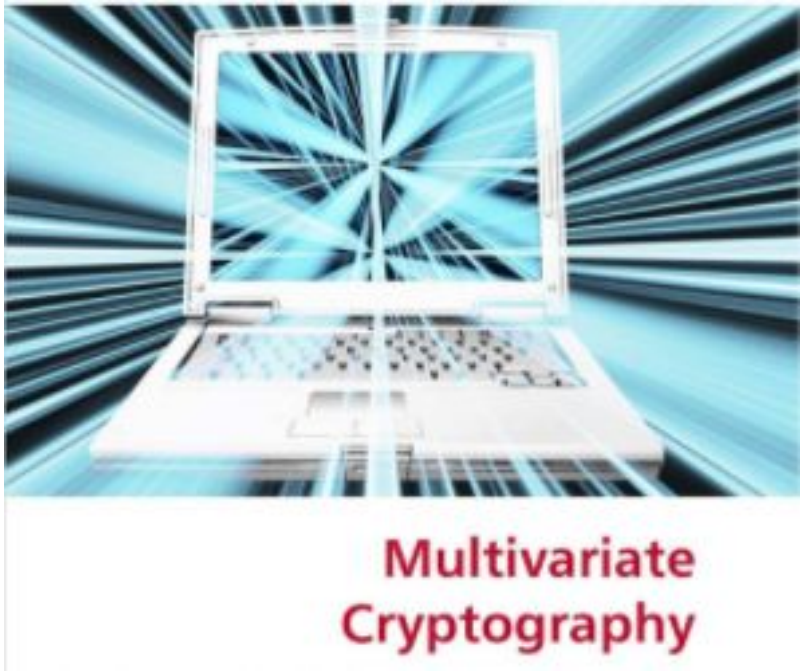
NewHope


Elements
Console
Sources
Network
Timeline
Profiles
Application
Security
Audits
Overview
Main Origin
https://play.google.com
Secure Origins
https://www.gstatic.com
https://lh3.googleusercont
https://lh4.googleusercont
https://lh5.googleusercont
https://lh6.googleusercont
https://lh3.ggpht.com
https://lh4.ggpht.com
https://lh5.ggpht.com
https://books.google.com
https://ajax.googleapis.com
https://www.google.com
https://www.google-analyti
https://play.google.com
View requests in Network Panel
Connection
Protocol TLS 1.2
Key Exchange CECPQ1_ECDSA
Cipher Suite AES_256_GCM
Certificate
Subject *.google.com
SAN *.google.com
*.android.com
Show more (52 total)
Valid From Thu, 23 Jun 2016 08:33:56 GMT
Valid Until Thu, 15 Sep 2016 08:31:00 GMT
Issuer Google Internet Authority G2

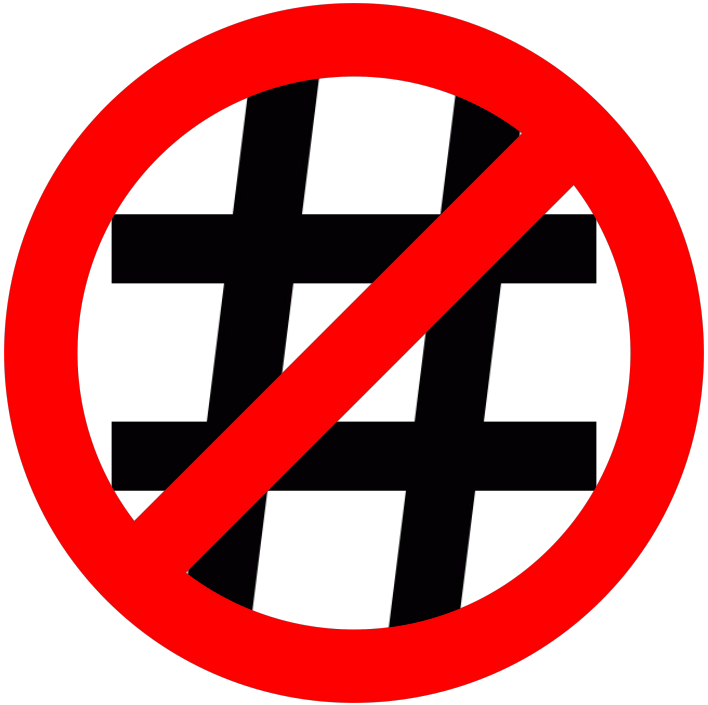
FRODO - Take Off the Ring!

Code-Based and McEliece/McBits

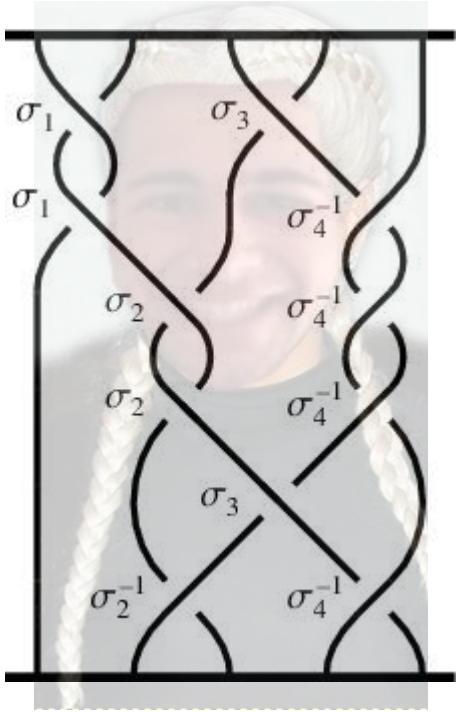
Ore Diffie-Hellman: Multivariate Crypto



No Hash Public Key Crypto!



Braids with Collin



Criteria (from NIST)

- Security (bits)
- Communication (bytes)
- Keygen (ms)
- Adoption-Ready
- Constant-Time
- ????

Postquantum TLS Analysis

	Security (bits)	Communication (bytes)	Keygen (ms)	Adoption-Ready	Constant Time	??
WolfSSL (NTRU)	128	2 049	2.249			??
SIDH	128	1 128	900	X	X	??
NewHope	256	3 872	0.31	X	X	??
Frodo	130	22 584	2.6	X	X	??
McBits	128	1 046 738	N/A		X	??
ODH	111	1 027 000	324 800 000 000 (primops)			??

AES 128 vs AES 256

Key size	Time to Crack
56-bit	399 seconds
128-bit	1.02×10^{18} years
192-bit	1.872×10^{37} years
256-bit	3.31×10^{56} years

BUT WAIT...

RSA IS QUANTUM-RESISTANT!

		Encryption	*Decryption*		
Key Size	**Bytes**		**Rem. tree**	**Cube root**	**CRT tree**
1MB	2^{20}	0.3	0.2	4.8	25.0
10MB	$2^{23.3}$	5	6	18	262
100MB	$2^{26.6}$	77	261	177	2851
1GB	2^{30}	654	812	1765	33586
4GB	2^{32}	3123	2318	8931	101309
8GB	2^{33}	6689	7214	17266	212215
16GB	2^{34}	18183	20420	34376	476798
32GB	2^{35}	29464	62729	62567	N/A
128GB	2^{37}	150975	N/A	N/A	N/A
256GB	2^{38}	362015	N/A	N/A	N/A

Table 4.1. Encryption and decryption times—We measure wall clock time in seconds on `lattice0` for encryption and the three stages of decryption: reducing the ciphertext modulo each prime factor, computing a cube root modulo each prime, and reconstructing the plaintext modulo the product.

The aggregate wall-clock time used by individual multiply jobs was about 1,239,626 seconds, and the elapsed time for the terabyte key generation was about four days. The final multiplication of two 512 GB integers took 176,223 seconds in wall-clock time, using 3.166TB of RAM and 2.5 TB of swap storage.