Post-quantum key exchange for the Internet

Douglas Stebila McMaster University

https://eprint.iacr.org/2016/1017

Acknowledgements

Collaborators

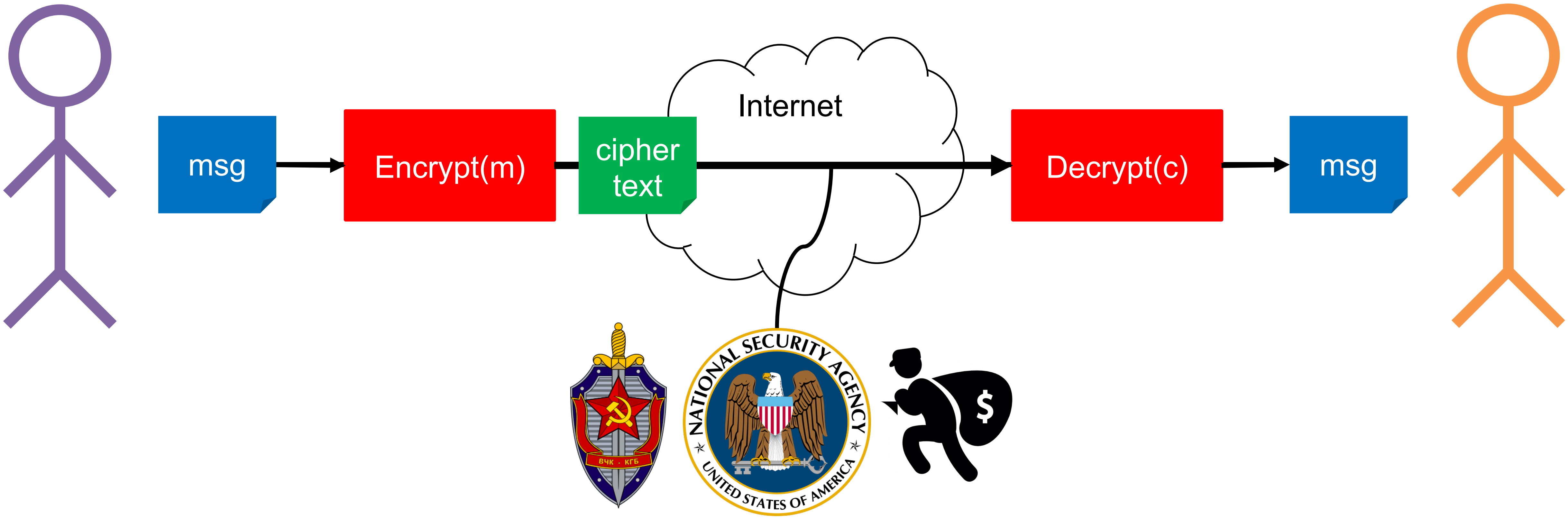
- Joppe Bos
- Craig Costello and Michael Naehrig
- Léo Ducas
- Ilya Mironov and Ananth Raghunathan
- Michele Mosca
- Valeria Nikolaenko

Support

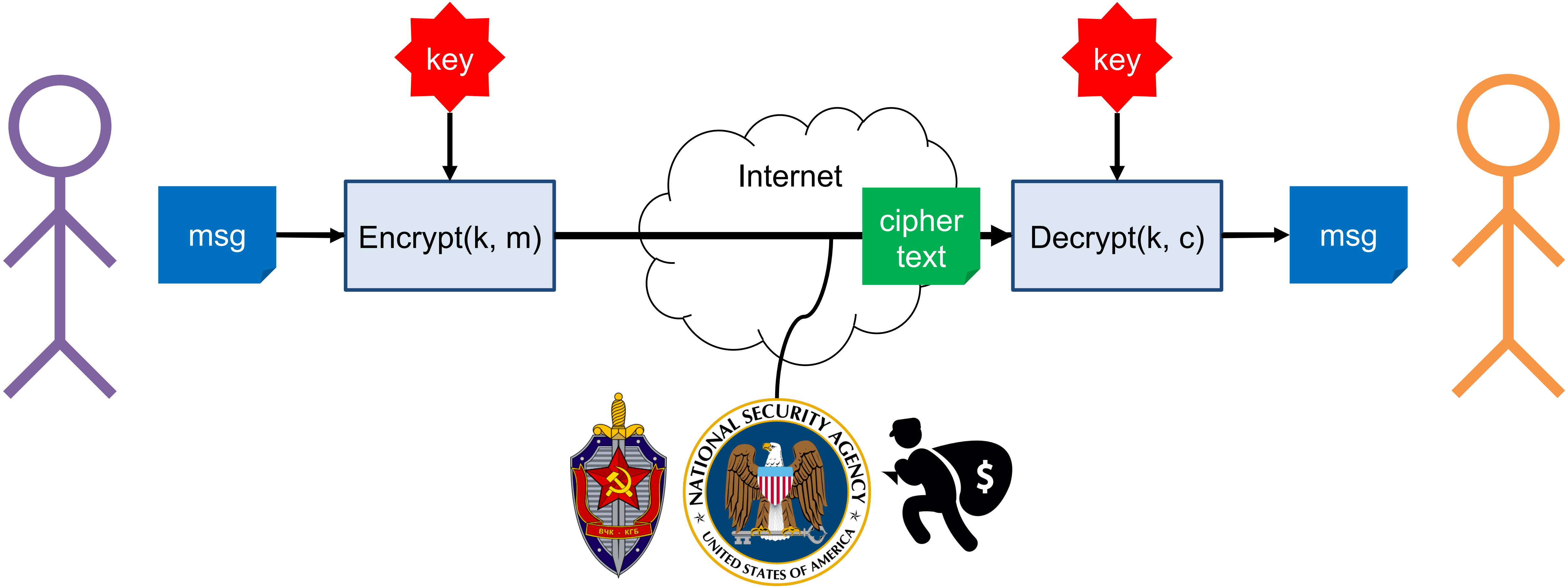
- Australian Research Council (ARC)
- Natural Sciences and Engineering Research Council of Canada (NSERC)
- Queensland University of Technology
- Tutte Institute for Mathematics and Computing

Background and motivation

Encryption


msg
Encrypt(m)
cipher text
Internet
Decrypt(c)
msg
NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA
$

Symmetric encryption



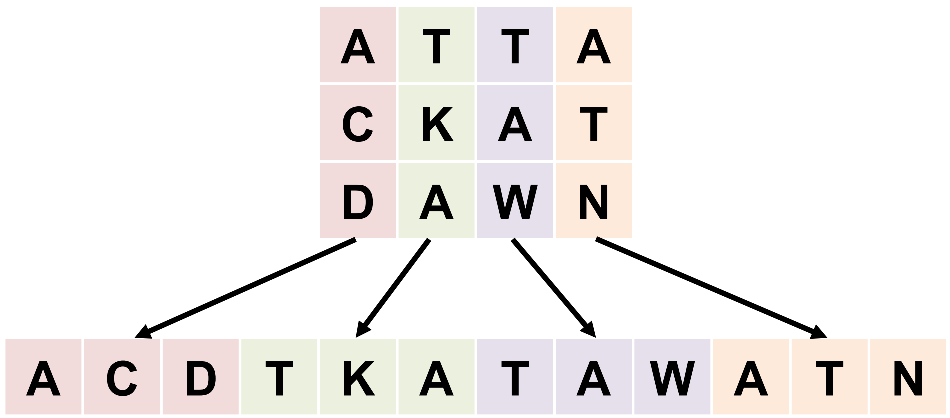
Symmetric encryption

Idea #1: Confusion

A	T	T	A	C	K		A	T		D	A	W	N
↓	↓	↓	↓	↓	↓		↓	↓		↓	↓	↓	↓
D	W	W	D	F	N		D	W		G	D	Z	Q

- A + 3 letters = D
- "Caesar cipher"
- Key: permutation on alphabet

Idea #2: Diffusion



- Diffusion window: 3 spots
- Key: permutation on columns

Symmetric encryption

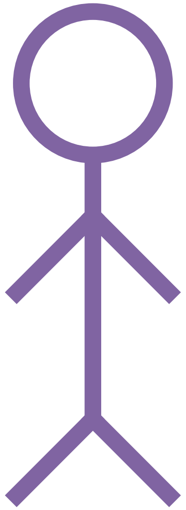
- Advanced Encryption Standard (AES)
  - Repeated rounds of (confusion then diffusion)
  - Different alphabet and column permutations derived from a common key

- But how do Alice and Bob establish a shared secret key in the first place?

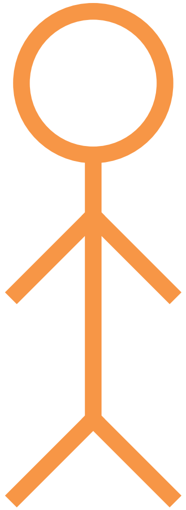
Key exchange – the Diffie–Hellman protocol

Let G be a cyclic group of prime order q, generated by g.

Candidate groups: $(\mathbb{Z}_p^*, \times)$, points on elliptic curve $(E/\mathbb{F}_p, +)$



$$x \xleftarrow{\$} \mathbb{Z}_q$$
$$X \leftarrow g^x$$

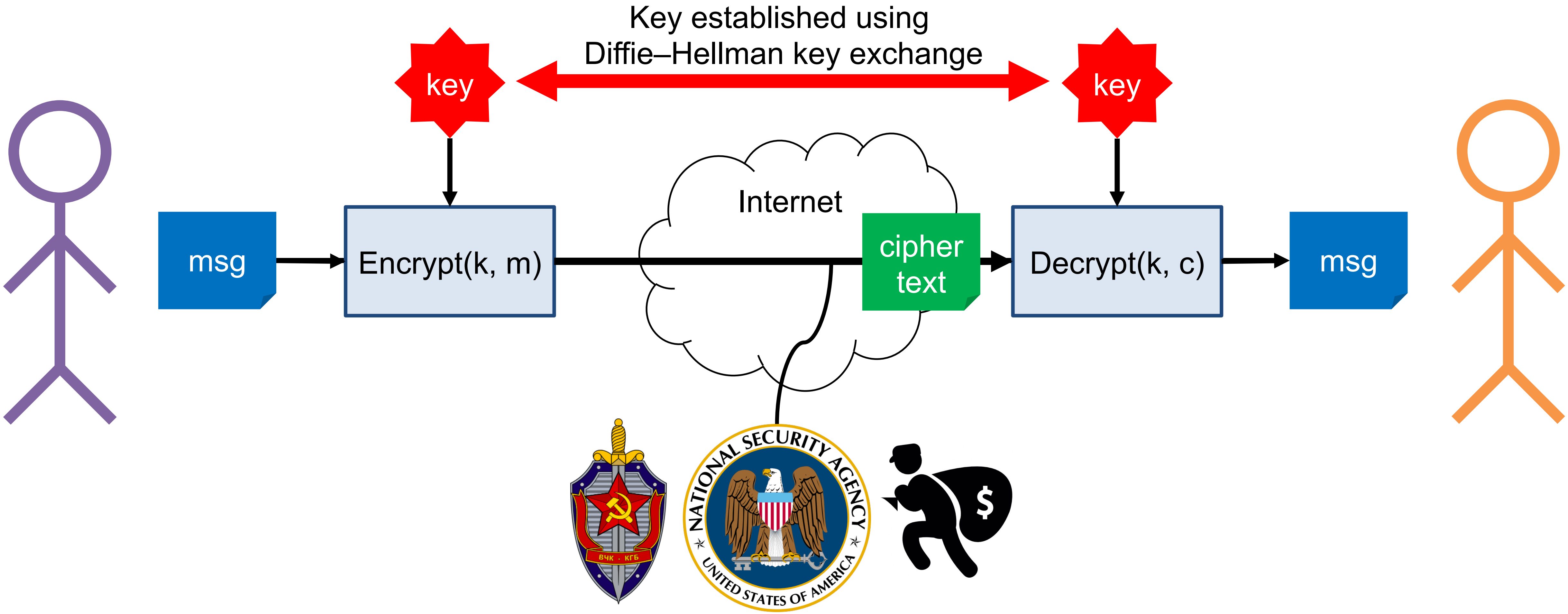


$$y \xleftarrow{\$} \mathbb{Z}_q$$
$$Y \leftarrow g^y$$

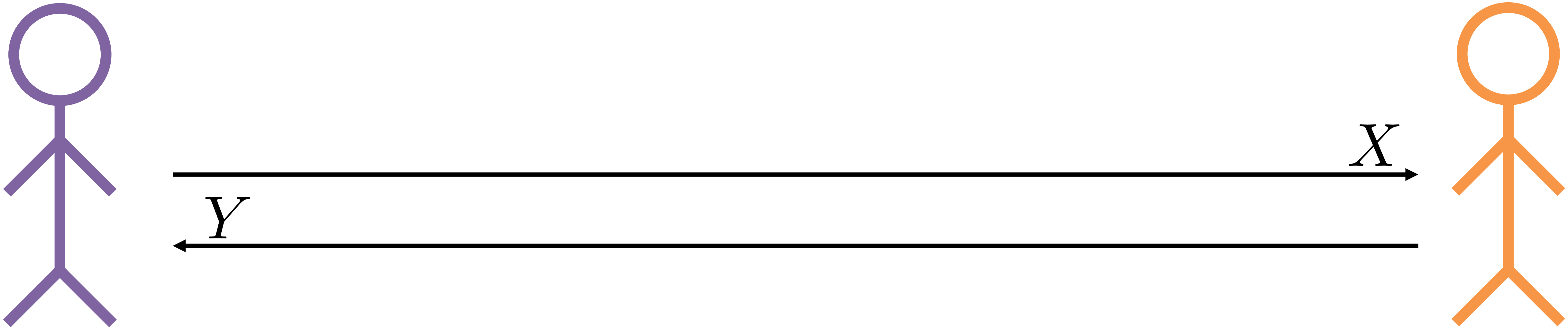
$$\xrightarrow{\quad X \quad}$$
$$\xleftarrow{\quad Y \quad}$$

$$k_A \leftarrow Y^x = g^{xy}$$
$$k_B \leftarrow X^y = g^{xy}$$

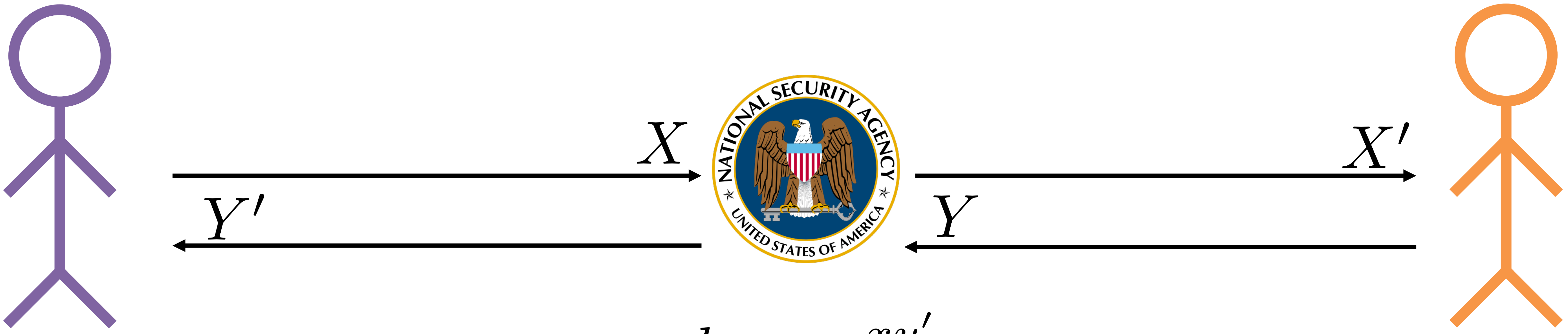
Key exchange + symmetric encryption



Man-in-the-middle attack



Man-in-the-middle attack



$k_1 = g^{xy'}$

$k_1 = g^{xy'}$
$k_2 = g^{x'y}$

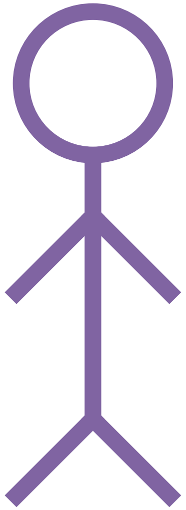
$k_2 = g^{x'y}$

Digital signatures

- The signer creates a pair of related keys
  - Signing key sk – kept private
  - Verification key vk – distributed publicly

- Anyone with a copy of the verification key should be able to check if a signature is valid
- Only the person with the signing key should be able to generate valid signatures

RSA digital signatures

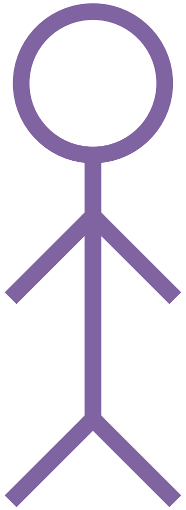


Key generation

1. Pick random primes p and q
2. Compute $n = pq$ and $\varphi(n) = (p-1)(q-1)$
3. Let $e = 3$
4. Compute $d = e^{-1} \bmod \varphi(n)$

- Signing key: $sk = (n, d)$
- Verification key: $vk = (n, e)$

RSA digital signatures

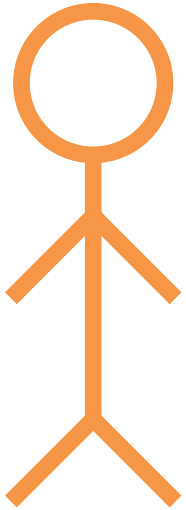


Sign message $m \in \mathbb{Z}_n$ using $sk = (n, d)$.

1. Compute $\sigma \leftarrow m^d \bmod n$

Verify message (m, σ) using $vk = (n, e)$.

1. Check if $\sigma^e \equiv m \bmod n$.

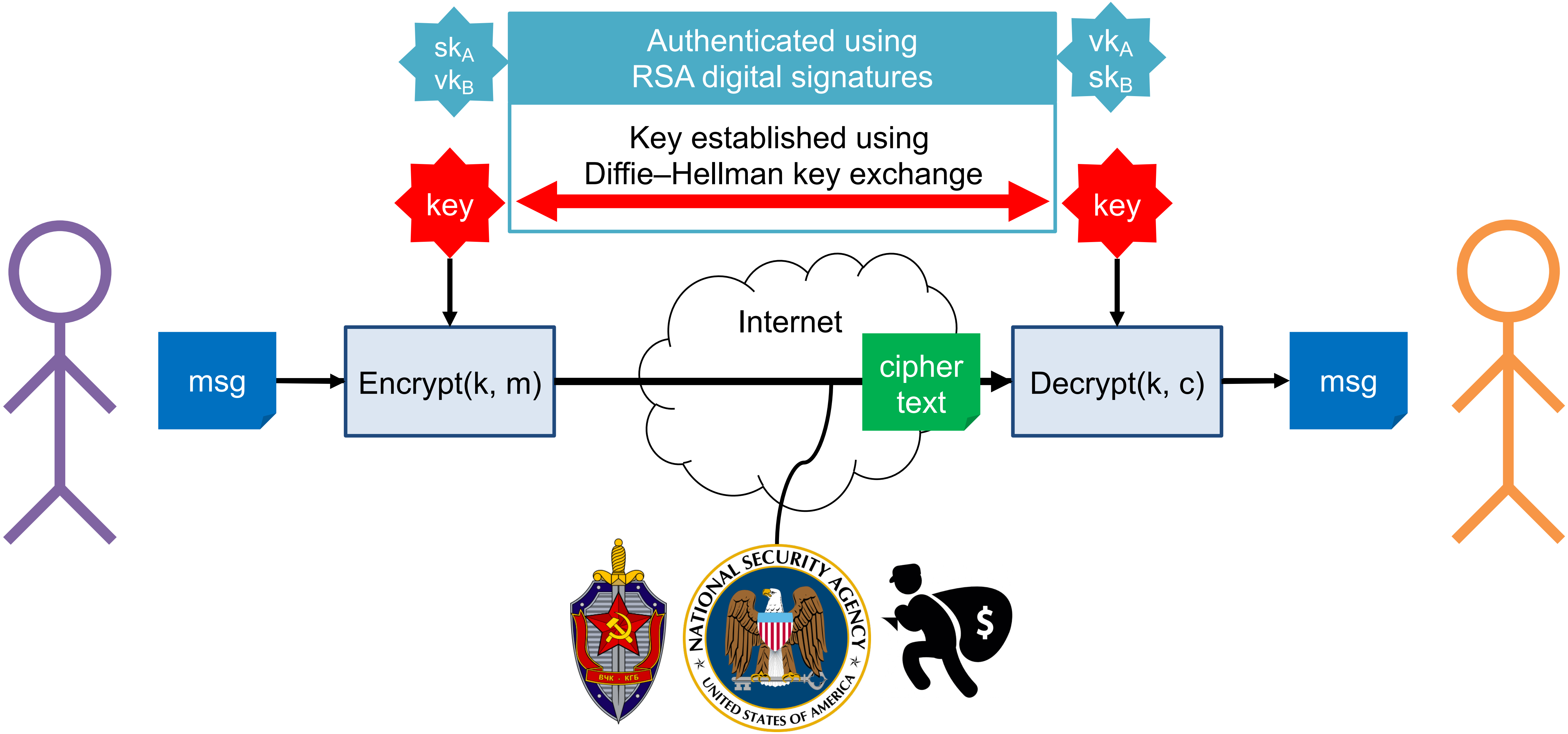


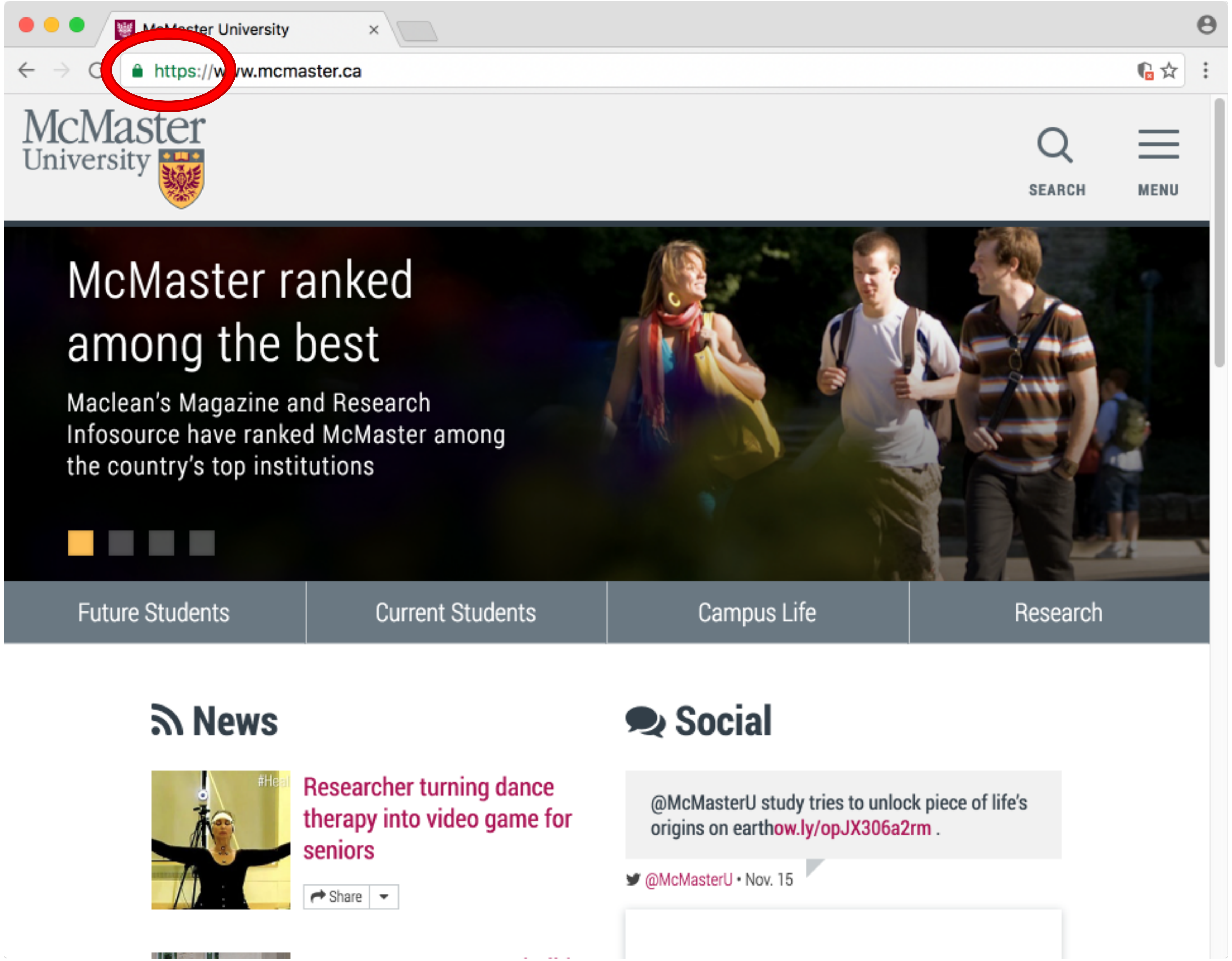
Why does verification work?

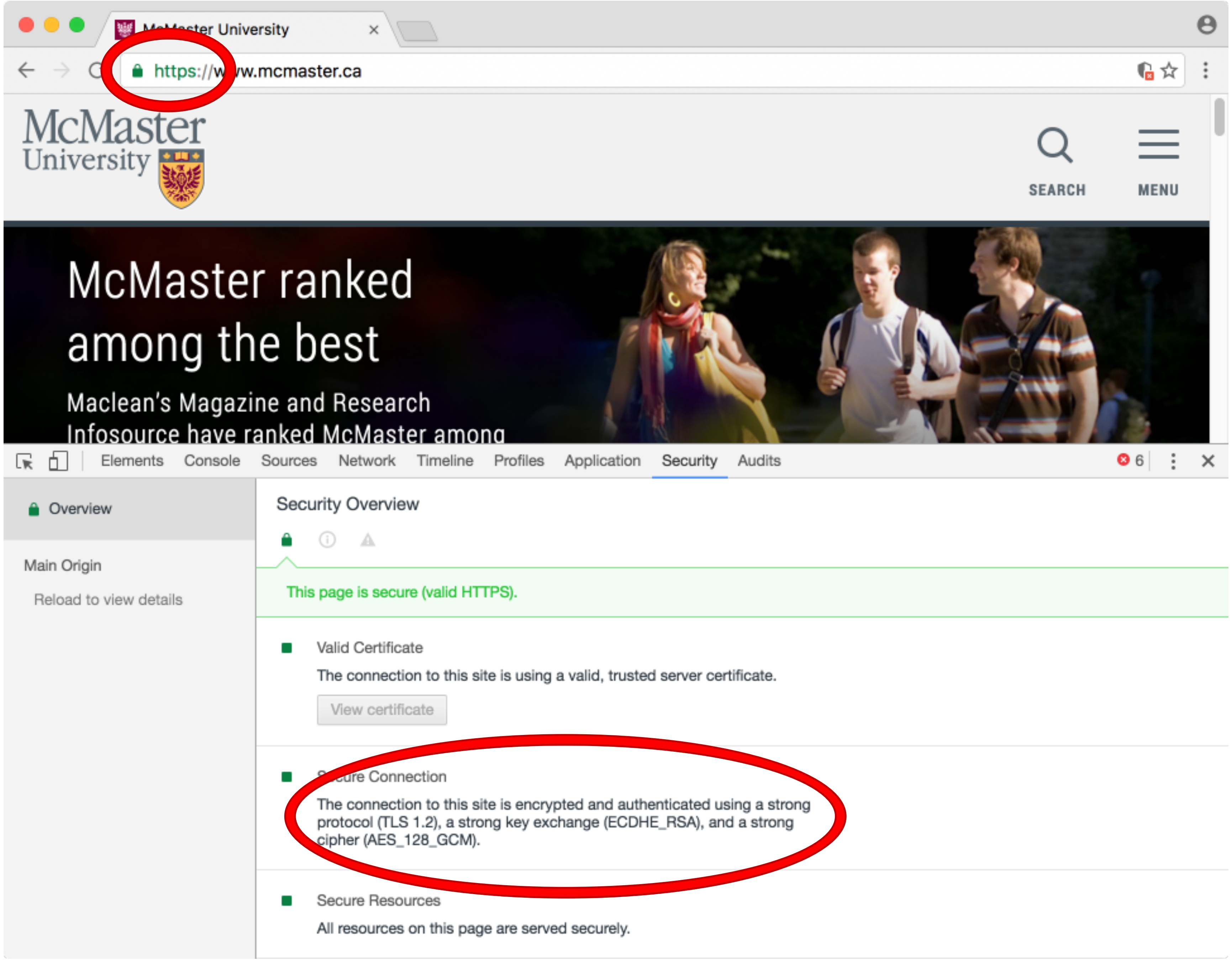
$$\sigma^e \equiv (m^d)^e \equiv m^{ed} \equiv m^1 \bmod n$$

since $ed \equiv 1 \bmod \varphi(n)$

Authenticated key exchange + symmetric encryption




McMaster University
https://www.mcmaster.ca
McMaster University
SEARCH
MENU
McMaster ranked among the best
Maclean's Magazine and Research Infosource have ranked McMaster among the country's top institutions
Future Students
Current Students
Campus Life
Research
News
Researcher turning dance therapy into video game for seniors
Share
Social
@McMasterU study tries to unlock piece of life's origins on earthow.ly/opJX306a2rm .
@McMasterU • Nov. 15


McMaster University
https://www.mcmaster.ca
McMaster University
SEARCH
MENU
McMaster ranked among the best
Maclean's Magazine and Research Infosource have ranked McMaster among
Elements Console Sources Network Timeline Profiles Application Security Audits
Overview
Main Origin
Reload to view details
Security Overview
This page is secure (valid HTTPS).
Valid Certificate
The connection to this site is using a valid, trusted server certificate.
View certificate
Secure Connection
The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA), and a strong cipher (AES_128_GCM).
Secure Resources
All resources on this page are served securely.

Why is this secure?

1. If AES symmetric encryption/decryption is secure,
   and no one else knows Alice and Bob's shared key,
   then their message is confidential.

2. If Diffie–Hellman key exchange is secure,
   and no one carried out a man-in-the-middle attack,
   then no one else knows Alice and Bob's shared key.

3. If RSA digital signatures are secure,
   and Alice and Bob have copies of each other's verification key,
   then they can confirm no one carried out a man-in-middle attack.

Reductionist security

- Relate the security of breaking the cryptosystem to the difficulty of solving some mathematical problem.

Factoring problem:

1. Pick two large random equal length primes p and q.
2. Compute $n = pq$
3. Given n, find p or q.

Reductionist security

~~**Goal**: If factoring is difficult, then forging RSA digital signatures is hard.~~

Try to prove this using contrapositive:

~~Given a polynomial time algorithm A for forging RSA digital signatures, then we can use A to construct a polynomial time algorithm B for factoring.~~

Thm: If factoring is easy, then forging RSA digital signatures is easy.

Currently, the best known method for forging RSA digital signatures is to factor n.

Assume RSA digital signatures are as hard as factoring.

Best known algorithm for factoring takes sub-exponential time.

Reductionist security

~~**Goal**: If computing discrete logarithms in G is difficult, then breaking Diffie–Hellman key exchange is hard.~~

Try to prove this using contrapositive:

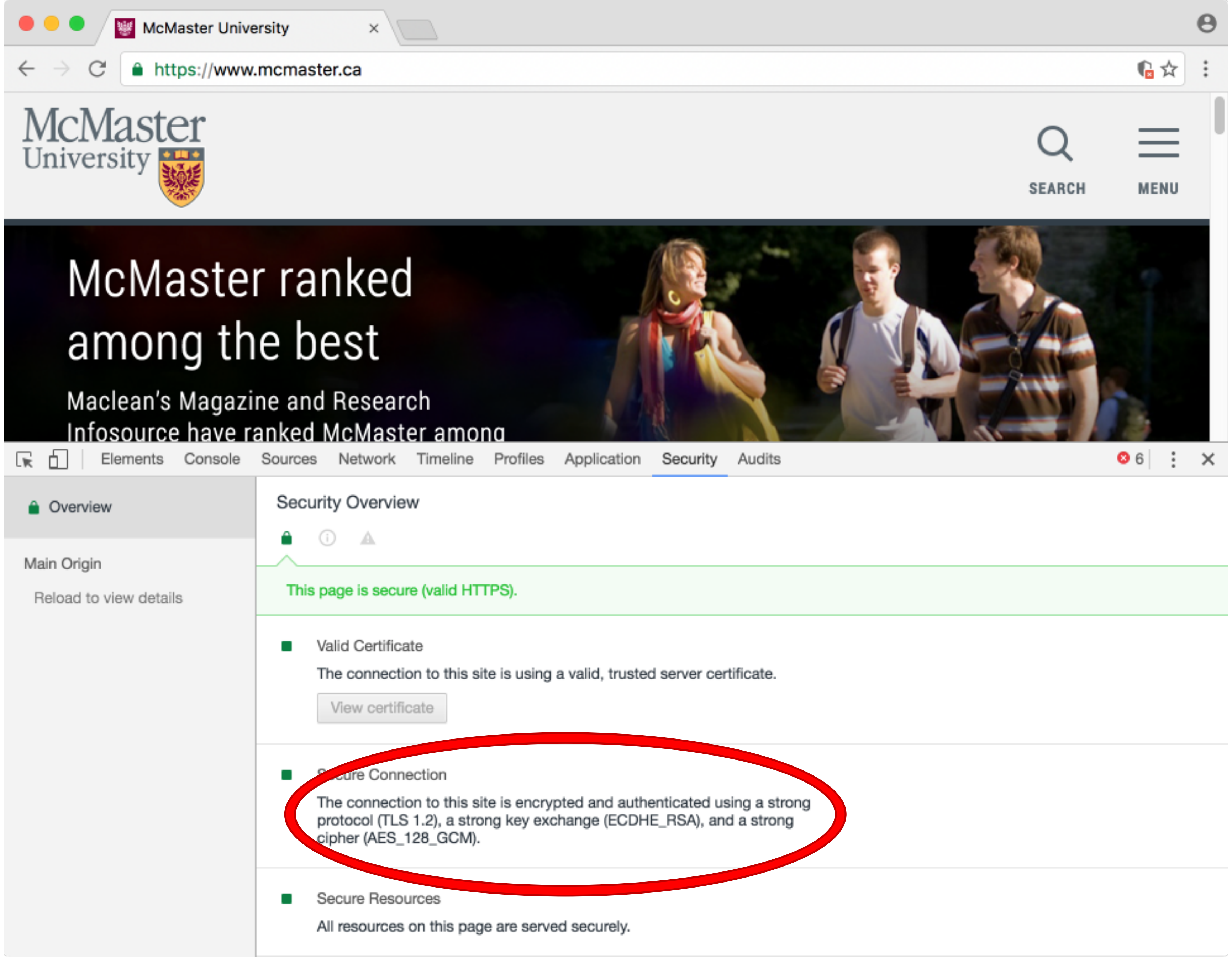
~~Given a polynomial time algorithm A for breaking Diffie–Hellman key exchange, then we can use A to construct a polynomial time algorithm B for discrete logarithms.~~

Thm: If computing discrete logarithms is easy, then breaking Diffie–Hellman key exchange is easy.

Currently, the best known method for breaking DH key exchange is to computing discrete logarithms.

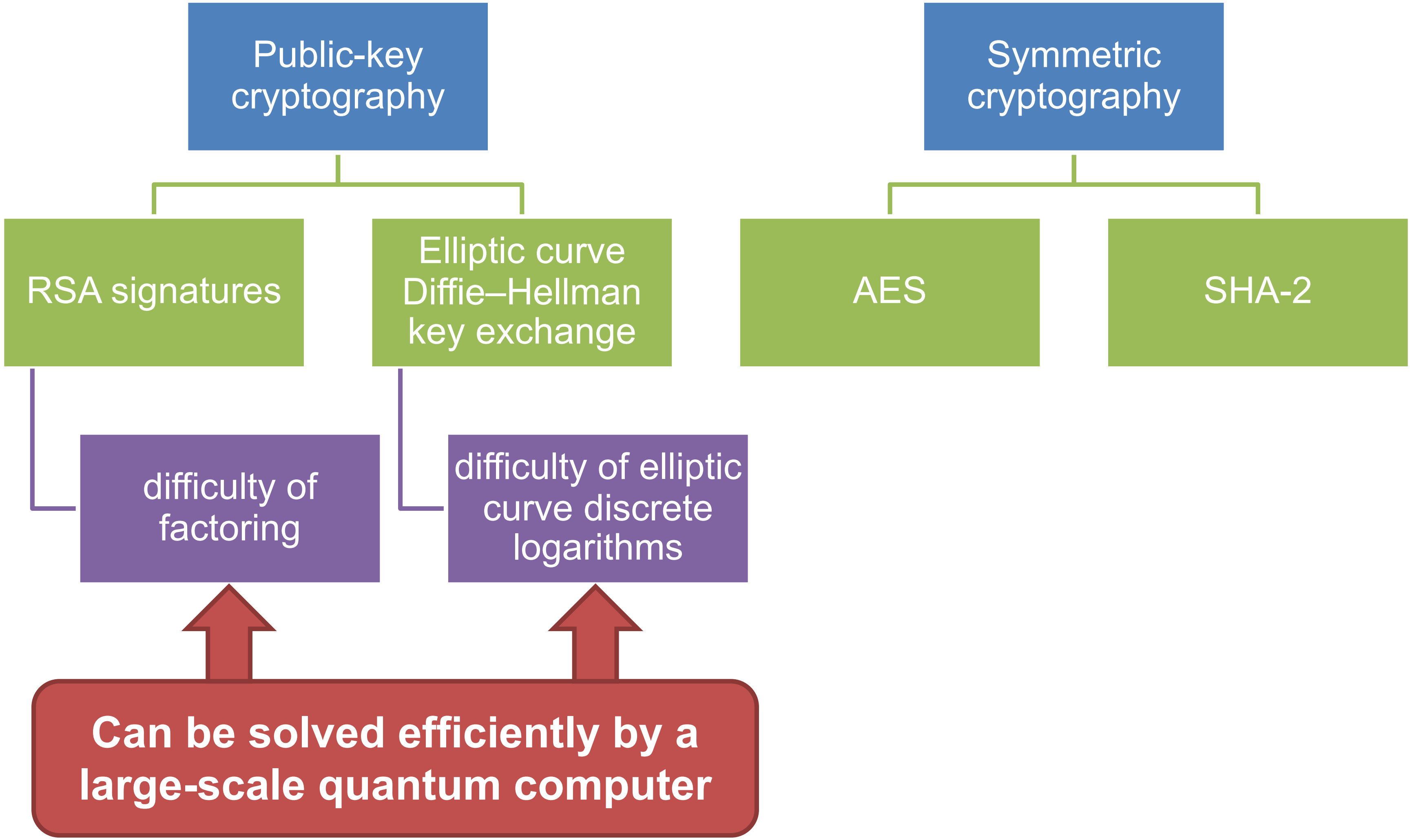
Assume DH key exchange is as hard as discrete logs.

Best known algorithm for discrete logs takes exponential time.


McMaster University
https://www.mcmaster.ca
McMaster University
SEARCH
MENU
McMaster ranked among the best
Maclean's Magazine and Research Infosource have ranked McMaster among
Elements Console Sources Network Timeline Profiles Application Security Audits
Overview
Main Origin
Reload to view details
Security Overview
This page is secure (valid HTTPS).
Valid Certificate
The connection to this site is using a valid, trusted server certificate.
View certificate
Secure Connection
The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA), and a strong cipher (AES_128_GCM).
Secure Resources
All resources on this page are served securely.

Contemporary cryptography

TLS-ECDHE-RSA-AES128-GCM-SHA256

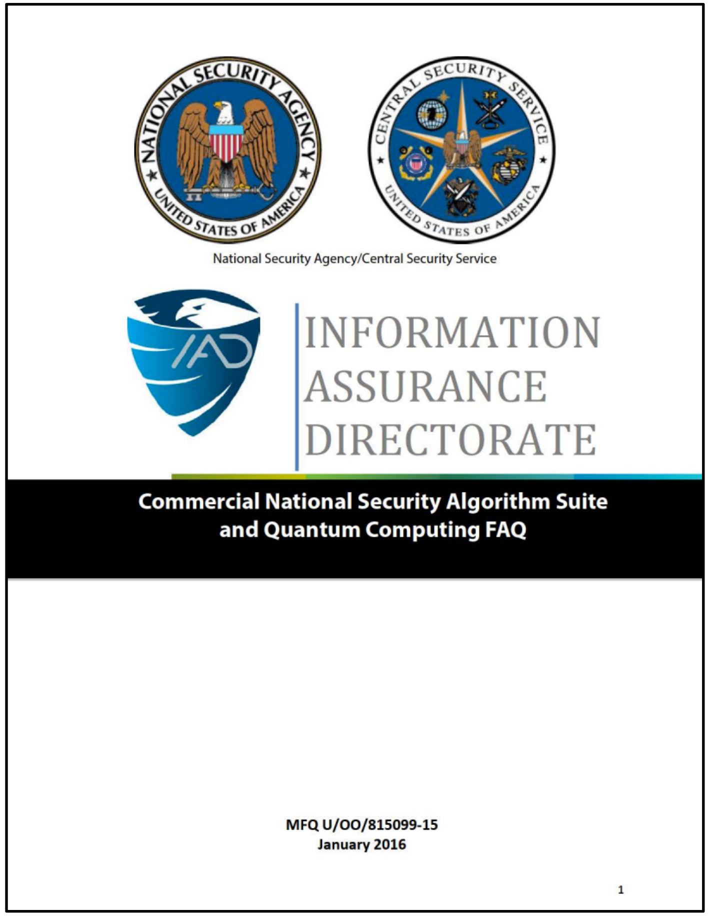


When will a large-scale quantum computer be built?

“I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031.”

— Michele Mosca, November 2015
https://eprint.iacr.org/2015/1075

Post-quantum cryptography in government



National Security Agency/Central Security Service

INFORMATION ASSURANCE DIRECTORATE

Commercial National Security Algorithm Suite and Quantum Computing FAQ

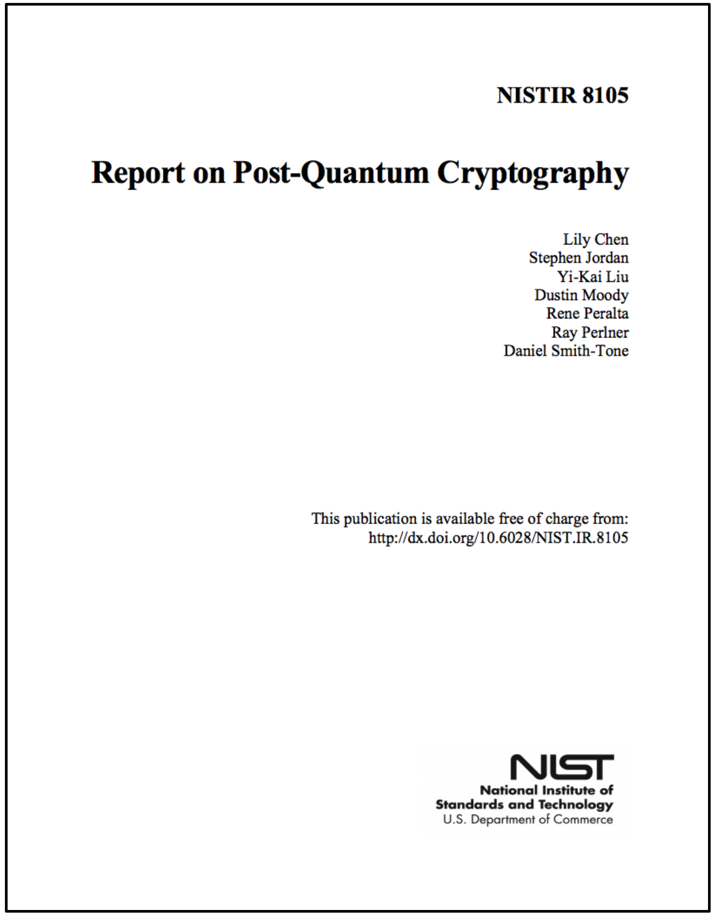
MFQ U/OO/815099-15
January 2016

1

Aug. 2015 (Jan. 2016)

"IAD will initiate a transition to quantum resistant algorithms in the not too distant future."

– NSA Information Assurance Directorate, Aug. 2015



NISTIR 8105

Report on Post-Quantum Cryptography

Lily Chen
Stephen Jordan
Yi-Kai Liu
Dustin Moody
Rene Peralta
Ray Perlner
Daniel Smith-Tone

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.8105

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Apr. 2016

NIST Post-quantum Crypto Project timeline

September 2016	Feedback on call for proposals
Fall 2016	Formal call for proposals
November 2017	**Deadline for submissions**
Early 2018	Workshop – submitters’ presentations
3–5 years	Analysis phase
2 years later	Draft standards ready

http://www.nist.gov/pqcrypto

Post-quantum / quantum-safe crypto

No known exponential quantum speedup

Hash-based

- Merkle signatures
- Sphincs

Code-based

- McEliece

Multivariate

- multivariate quadratic

Lattice-based

- NTRU
- learning with errors
- ring-LWE

Isogenies

- supersingular elliptic curve isogenies

Lots of questions

- Design better post-quantum key exchange and signature schemes
- Improve classical and quantum attacks
- Pick parameter sizes
- Develop fast, secure implementations
- Integrate them into the existing infrastructure

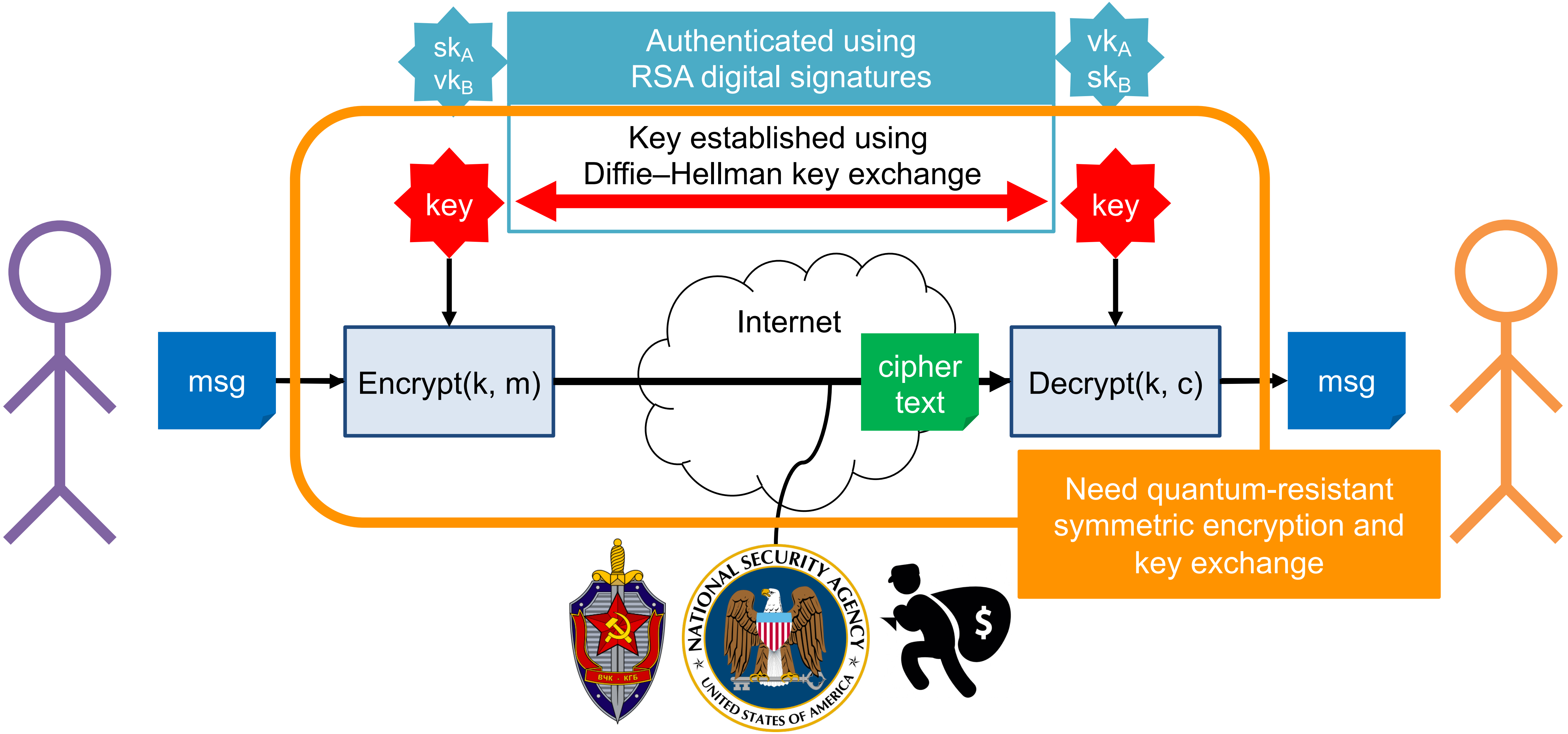
This talk

- Two key exchange protocols from lattice-based problems
  - BCNS15: key exchange from the ring learning with errors problem
  - Frodo: key exchange from the learning with errors problem
- Open Quantum Safe project
  - A library for comparing post-quantum primitives
  - Framework for easing integration into applications like OpenSSL

Why key exchange?

Premise: large-scale quantum computers don't exist right now, but we want to protect today's communications against tomorrow's adversary.

Authenticated key exchange + symmetric encryption



Need quantum-resistant symmetric encryption and key exchange

Why key exchange?

- AES encryption already quantum resistant
- Signatures still done with traditional primitives (e.g., RSA)
  - we only need authentication to be secure *now*
  - benefit: use existing RSA-based public key infrastructure
- Key agreement done with ring-LWE, LWE, …

Learning with errors problems

Solving systems of linear equations

$$\underbrace{\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}}_{\mathbb{Z}_{13}^{7\times 4}} \times \underbrace{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}_{\text{secret } \mathbb{Z}_{13}^{4\times 1}} = \underbrace{\begin{bmatrix} 4 \\ 8 \\ 1 \\ 10 \\ 4 \\ 12 \\ 9 \end{bmatrix}}_{\mathbb{Z}_{13}^{7\times 1}}$$

Linear system problem: given **blue**, find **red**

Solving systems of linear equations

secret

$$\underset{\mathbb{Z}_{13}^{7\times 4}}{\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}} \times \underset{\mathbb{Z}_{13}^{4\times 1}}{\begin{bmatrix} 6 \\ 9 \\ 11 \\ 11 \end{bmatrix}} = \underset{\mathbb{Z}_{13}^{7\times 1}}{\begin{bmatrix} 4 \\ 8 \\ 1 \\ 10 \\ 4 \\ 12 \\ 9 \end{bmatrix}}$$

Easily solved using Gaussian elimination (Linear Algebra 101)

Linear system problem: given **blue**, find **red**

Learning with errors problem

$$\underbrace{\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}}_{\substack{\textbf{random} \\ \mathbb{Z}_{13}^{7\times 4}}} \times \underbrace{\begin{bmatrix} 6 \\ 9 \\ 11 \\ 11 \end{bmatrix}}_{\substack{\textbf{secret} \\ \mathbb{Z}_{13}^{4\times 1}}} + \underbrace{\begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \\ 1 \\ 0 \\ -1 \end{bmatrix}}_{\substack{\textbf{small noise} \\ \mathbb{Z}_{13}^{7\times 1}}} = \underbrace{\begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix}}_{\mathbb{Z}_{13}^{7\times 1}}$$

Learning with errors problem

$$
\begin{array}{ccccccc}
\textbf{random} & & \textbf{secret} & & \textbf{small noise} & & \\
\mathbb{Z}_{13}^{7\times 4} & & \mathbb{Z}_{13}^{4\times 1} & & \mathbb{Z}_{13}^{7\times 1} & & \mathbb{Z}_{13}^{7\times 1} \\
\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix} & \times & \begin{bmatrix} \ \\ \ \\ \ \\ \ \end{bmatrix} & + & \begin{bmatrix} \ \\ \ \\ \ \\ \ \\ \ \\ \ \\ \ \end{bmatrix} & = & \begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix}
\end{array}
$$

Computational LWE problem: given **blue**, find **red**

Decision learning with errors problem

random $\mathbb{Z}_{13}^{7\times 4}$ **secret** $\mathbb{Z}_{13}^{4\times 1}$ **small noise** $\mathbb{Z}_{13}^{7\times 1}$ **looks random** $\mathbb{Z}_{13}^{7\times 1}$

$$\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix} \times \begin{bmatrix} \, \\ \, \\ \, \\ \, \end{bmatrix} + \begin{bmatrix} \, \\ \, \\ \, \\ \, \\ \, \\ \, \\ \, \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix}$$

Decision LWE problem: given **blue**, distinguish **green** from random

Toy example versus real-world example

$\mathbb{Z}_{13}^{7\times 4}$

4	1	11	10
5	5	9	5
3	9	0	10
1	3	3	2
12	7	3	4
6	5	11	4
3	3	5	0

$\mathbb{Z}_{4093}^{640\times 256}$

256

640

2738	3842	3345	2979	…
2896	595	3607		
377	1575			
2760				
…				

640 × 256 × 12 bits = **245 KiB**

Ring learning with errors problem

random

$\mathbb{Z}_{13}^{7\times 4}$

4	1	11	10
10	4	1	11
11	10	4	1
1	11	10	4
4	1	11	10
10	4	1	11
11	10	4	1

Each row is the cyclic shift of the row above

Ring learning with errors problem

random

$\mathbb{Z}_{13}^{7\times 4}$

4	1	11	10
3	4	1	11
2	3	4	1
12	2	3	4
9	12	2	3
10	9	12	2
11	10	9	12

Each row is the cyclic shift of the row above

…

with a special wrapping rule: *x* wraps to –*x* mod 13.

Ring learning with errors problem

random

$\mathbb{Z}_{13}^{7\times 4}$

4	**1**	**11**	**10**

Each row is the cyclic shift of the row above

…

with a special wrapping rule: *x* wraps to –*x* mod 13.

So I only need to tell you the first row.

Ring learning with errors problem

$$\mathbb{Z}_{13}[x]/\langle x^4+1\rangle$$

	$4 + 1x + 11x^2 + 10x^3$	**random**
×	$6 + 9x + 11x^2 + 11x^3$	**secret**
+	$0 - 1x + 1x^2 + 1x^3$	**small noise**
=	$10 + 5x + 10x^2 + 7x^3$	

Ring learning with errors problem

$$\mathbb{Z}_{13}[x]/\langle x^4 + 1 \rangle$$

	$4 + 1x + 11x^2 + 10x^3$	**random**
$\times$		**secret**
$+$		**small noise**
$=$	$10 + 5x + 10x^2 + 7x^3$	

Computational ring-LWE problem: given **blue**, find **red**

Decision ring learning with errors problem

$\mathbb{Z}_{13}[x]/\langle x^4+1\rangle$

	$4 + 1x + 11x^2 + 10x^3$	**random**
×		**secret**
+		**small noise**
=	$10 + 5x + 10x^2 + 7x^3$	**looks random**

Decision ring-LWE problem: given **blue**, distinguish **green** from random

Decision ring learning with errors problem **with small secrets**

$\mathbb{Z}_{13}[x]/\langle x^4 + 1 \rangle$

	$4 + 1x + 11x^2 + 10x^3$	**random**
×	$1 + 0x - 1x^2 + 2x^3$	**small secret**
+		**small noise**
=	$10 + 5x + 10x^2 + 7x^3$	**looks random**

Decision ring-LWE problem: given blue, distinguish green from random

Problems

Computational LWE problem	Decision LWE problem
Computational ring-LWE problem	Decision ring-LWE problem

with or without short secrets

Key agreement from ring-LWE

Bos, Costello, Naehrig, Stebila.
Post-quantum key exchange for the TLS protocol from the ring learning with errors problem.
IEEE Symposium on Security & Privacy (S&P) 2015.

https://www.douglas.stebila.ca/research/papers/SP-BCNS15/

Decision ring learning with errors problem with short secrets

Definition. Let n be a power of 2, q be a prime, and $R_q = \mathbb{Z}_q[X]/(X^n+1)$ be the ring of polynomials in X with integer coefficients modulo q and polynomial reduction modulo X^n+1. Let χ be a distribution over R_q.

Let $s \xleftarrow{\$} \chi$.

Define:

- $O_{\chi,s}$: Sample $a \xleftarrow{\$} \mathcal{U}(R_q)$, $e \xleftarrow{\$} \chi$; return $(a, as+e)$.
- U: Sample $(a, b') \xleftarrow{\$} \mathcal{U}(R_q \times R_q)$; return (a, b').

The *decision R-LWE problem with short secrets* for n, q, χ is to distinguish $O_{\chi,s}$ from U.

Hardness of decision ring-LWE

worst-case approximate shortest (independent) vector problem (SVP/SIVP) on ideal lattices in *R*

poly-time [LPR10]

search ring-LWE

poly-time [LPR10]

decision ring-LWE

tight [ACPS09]

decision ring-LWE with short secrets

[LPR10] Lyubashevsky, Piekert, Regev. *EUROCRYPT 2010.*
[ACPS15] Applebaum, Cash, Peikert, Sahai. *CRYPTO 2009.*
[CKMS16] Chatterjee, Koblitz, Menezes, Sarkar. ePrint 2016/360.

Lattices

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subseteq \mathbb{Z}_q^{n \times n}$ be a set of linearly independent basis vectors for $\mathbb{Z}_q^n$. Define the corresponding <u>lattice</u>

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} z_i \mathbf{b}_i : z_i \in \mathbb{Z} \right\} .$$

(In other words, a lattice is a set of *integer* linear combinations.)

Define the <u>minimum distance</u> of a lattice as

$$\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\| .$$

Shortest vector problem

The shortest vector problem (SVP) is: given a basis $\mathbf{B}$ for some lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, find a shortest non-zero vector, i.e., find $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$.

The decision approximate shortest vector problem (GapSVP_γ) is: given a basis $\mathbf{B}$ for some lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$ where either $\lambda_1(\mathcal{L}) \leq 1$ or $\lambda_1(\mathcal{L}) > \gamma$, determine which is the case.

Shortest vector problem

- Can solve GapSVP_γ using lattice reduction algorithm like LLL
- Runtime depends on approximation factor gamma
- No known classical or quantum algorithm can get polynomial approximation factor in polynomial runtime

Hardness of decision ring-LWE

worst-case approximate shortest (independent) vector problem (SVP/SIVP) on ideal lattices in *R*

poly-time [LPR10]

search ring-LWE

poly-time [LPR10]

decision ring-LWE

tight [ACPS09]

decision ring-LWE with short secrets

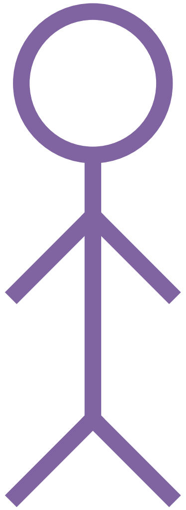
- GapSVP parameter gamma depends on LWE parameters n, q, and error distribution χ
- Estimate parameters based on runtime of lattice reduction algorithms e.g. [APS15]
- (Ignore non-tightness.) [CKMS16]

[LPR10] Lyubashevsky, Piekert, Regev. *EUROCRYPT 2010.*
[ACPS15] Applebaum, Cash, Peikert, Sahai. *CRYPTO 2009.*
[CKMS16] Chatterjee, Koblitz, Menezes, Sarkar. ePrint 2016/360.

Basic ring-LWE-DH key agreement (unauthenticated)

Based on Lindner–Peikert ring-LWE public key encryption scheme

public: uniform a in $R_q = \mathbf{Z}_q[x]/(x^n+1)$

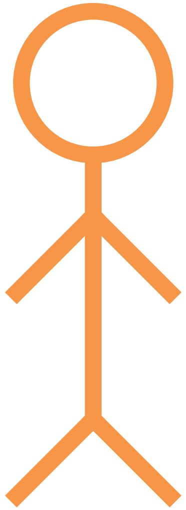


Alice

secret:
random "small" s, e in R_q

Bob

secret:
random "small" s', e' in R_q



$$b = a \bullet s + e \longrightarrow$$

$$\longleftarrow b' = a \bullet s' + e'$$

Alice shared secret:
$s \bullet b' = s \bullet (a \bullet s' + e') \approx s \bullet a \bullet s'$

Bob shared secret:
$b \bullet s' \approx s \bullet a \bullet s'$

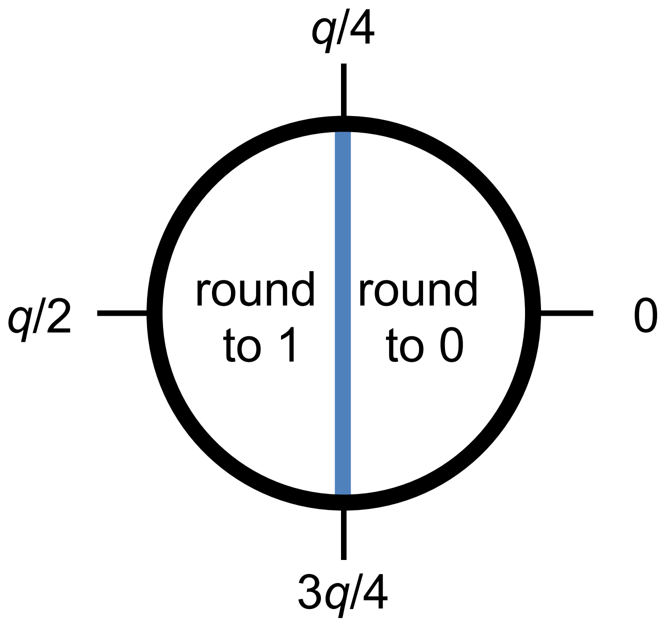
These are only approximately equal ⇒ need rounding

Rounding

- Each coefficient of the polynomial is an integer modulo q
- Treat each coefficient independently

Basic rounding

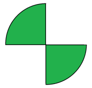
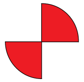
- Round either to 0 or $q/2$
- Treat $q/2$ as 1

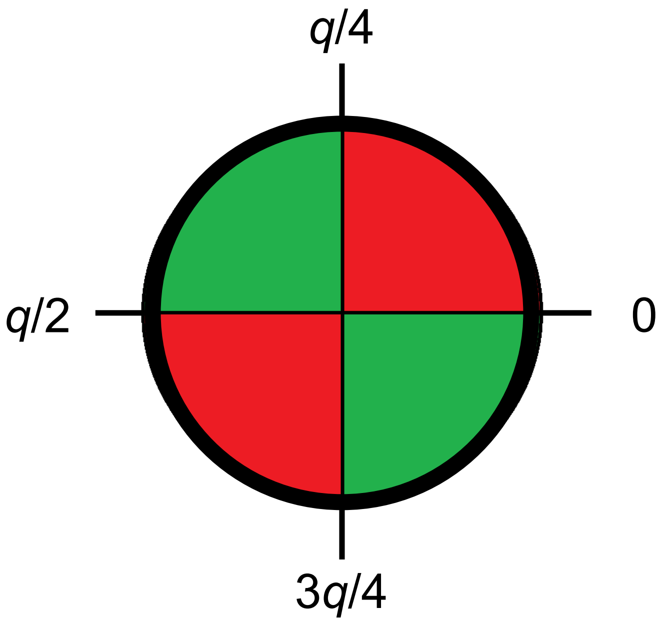


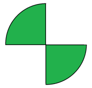
This works most of the time: prob. failure 2^{-10}.

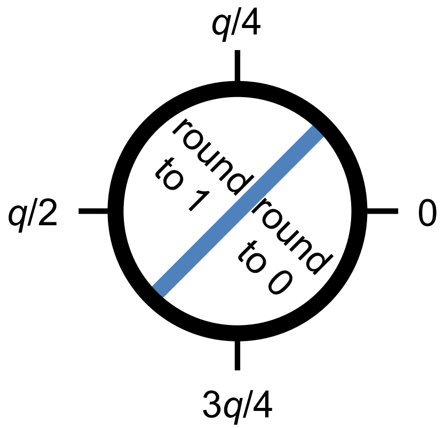
Not good enough: we need exact key agreement.

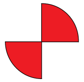
Better rounding

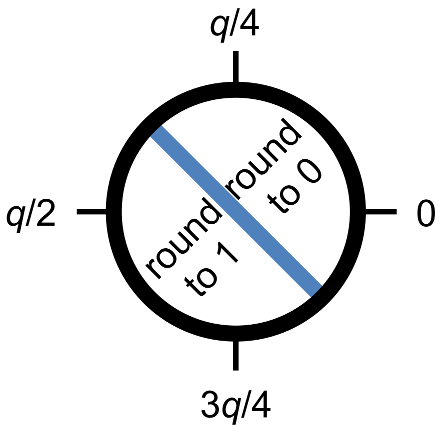
Bob says which of two regions the value is in:  or 



If 

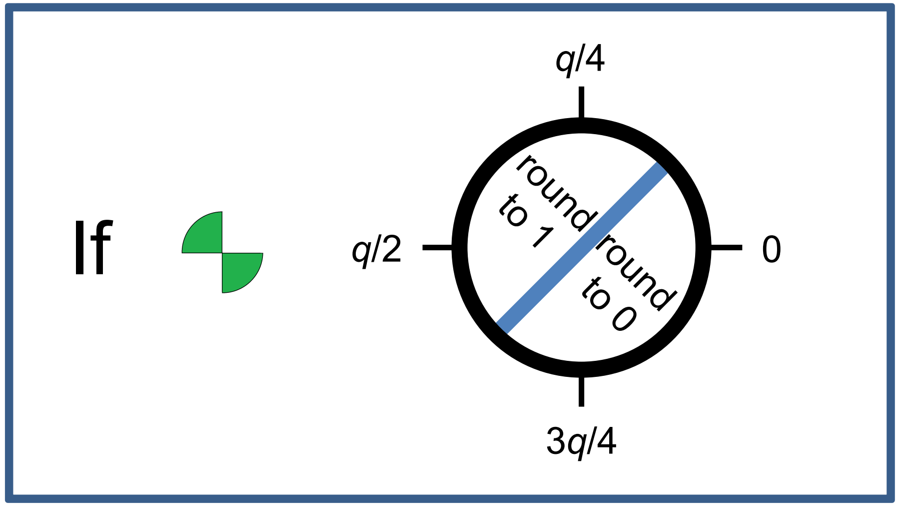


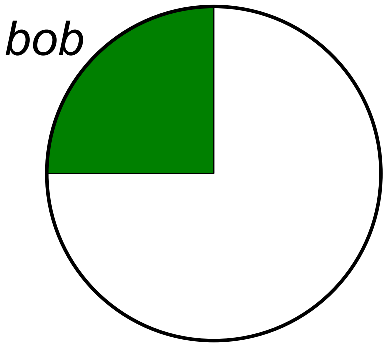
If 

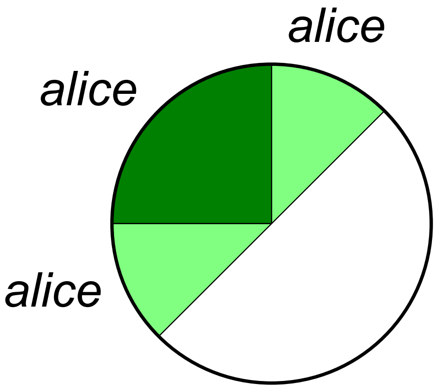


Better rounding

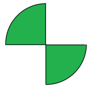
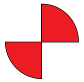
- If | *alice* – *bob* | ≤ *q*/8, then this always works.







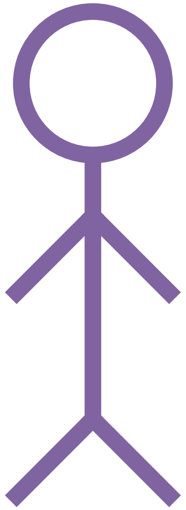
- For our parameters, probability | *alice* – *bob* | > *q*/8 is less than $2^{-128000}$.

- Security not affected: revealing  or  leaks no information

Exact ring-LWE-DH key agreement (unauthenticated)

Based on Lindner–Peikert ring-LWE public key encryption scheme

public: uniform a in $R_q = \mathbf{Z}_q[x]/(x^n+1)$



Alice

secret:
random "small" s, e in R_q

$b = a \cdot s + e$ →

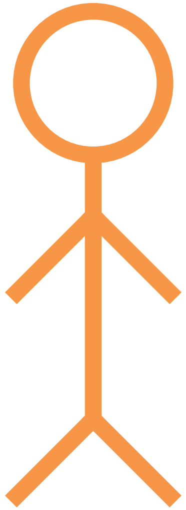
← $b' = a \cdot s' + e'$, or

shared secret:
round($s \cdot b'$)

Bob

secret:
random "small" s', e' in R_q

shared secret:
round($b \cdot s'$)



Thm: Key exchange is secure if decision ring learning with errors problem is hard.

Parameters

160-bit classical security,
80-bit quantum security

- n = 1024
- $q = 2^{32}–1$
- χ = discrete Gaussian with parameter sigma = 8/sqrt(2π)

- Failure: 2^{-12800}

- Total communication: 8.1 KiB

Implementation aspect 1:
Polynomial arithmetic

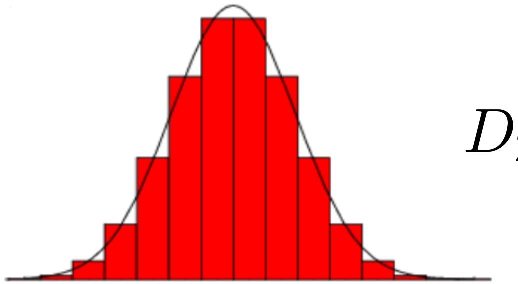
- Polynomial multiplication in $R_q = \mathbf{Z}_q[x]/(x^{1024}+1)$ done with Nussbaumer's FFT:

If $2^m = rk$, then

$$\frac{R[X]}{\langle X^{2^m}+1\rangle} \cong \frac{\left(\frac{R[Z]}{\langle Z^r+1\rangle}\right)[X]}{\langle X^k - Z\rangle}$$

- Rather than working modulo degree-1024 polynomial with coefficients in $\mathbf{Z}_q$, work modulo:
  - degree-256 polynomial whose coefficients are themselves polynomials modulo a degree-4 polynomial,
  - or degree-32 polynomials whose coefficients are polynomials modulo degree-8 polynomials whose coefficients are polynomials
  - or …

Implementation aspect 2:
Sampling discrete Gaussians



$$D_{\mathbb{Z},\sigma}(x) = \frac{1}{S} e^{-\frac{x^2}{2\sigma^2}} \qquad \text{for } x \in \mathbb{Z}, \sigma \approx 3.2, S = 8$$

- Security proofs require "small" elements sampled within statistical distance 2^{-128} of the true discrete Gaussian
- We use inversion sampling: precompute table of cumulative probabilities
- Choosing a good distribution and sampling efficiently is a challenge

Key agreement from LWE

Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila.
Frodo: Take off the ring! Practical, quantum-safe key exchange from LWE.
ACM Conference on Computer and Communications Security (CCS) 2016.

https://eprint.iacr.org/2016/659

Ring-LWE

$$\mathbb{Z}_{13}^{7\times 4}$$

4	**1**	**11**	**10**

Cyclic structure

$\Rightarrow$ Save communication,
more efficient computation

4 KiB representation

LWE

$$\mathbb{Z}_{4093}^{640\times 256}$$

256

640

2738	**3842**	**3345**	**2979**	**...**
2896	**595**	**3607**		
377	**1575**			
2760				
...				

640 × 256 × 12 bits = **245 KiB**

Ring-LWE

$$\mathbb{Z}_{13}^{7\times 4}$$

$$\begin{array}{|c|c|c|c|}\hline 4 & 1 & 11 & 10 \\ \hline \end{array}$$

Cyclic structure

$\Rightarrow$ Save communication,
more efficient computation

4 KiB representation

LWE

$$\mathbb{Z}_{2^{15}}^{752\times 8}$$

$$752\left\{\overbrace{\begin{array}{cccc c} 2738 & 3842 & 3345 & 2979 & \cdots \\ 2896 & 595 & 3607 & & \\ 377 & 1575 & & & \\ 2760 & & & & \\ \cdots & & & & \end{array}}^{8}\right.$$

752 × 8 × 15 bits = 11 KiB

Why consider (slower, bigger) LWE?

Generic vs. ideal lattices

- Ring-LWE matrices have additional structure
  - Relies on hardness of a problem in **ideal** lattices
- LWE matrices have no additional structure
  - Relies on hardness of a problem in **generic** lattices

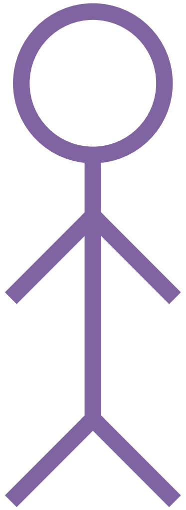
- Currently, best algorithms for ideal lattice problems are essentially the same as for generic lattices
  - Small constant factor improvement in some cases
  - Very recent quantum polynomial time algorithm for Ideal-SVP (http://eprint.iacr.org/2016/885) but not immediately applicable to ring-LWE

If we want to eliminate this additional structure, can we still get an efficient protocol?

Exact LWE-DH key agreement (unauthenticated)

Based on Lindner–Peikert LWE public key encryption scheme

public: uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$

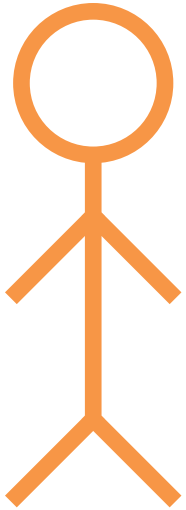


Alice

secret:
random "small"
$\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^{n \times m}$

Bob

secret:
random "small"
$\mathbf{S}', \mathbf{E}' \in \mathbb{Z}_q^{m \times n}$



$\mathbf{B} \leftarrow \mathbf{A}\mathbf{S} + \mathbf{E}$ →

← $\mathbf{B}' \leftarrow \mathbf{S}'\mathbf{A} + \mathbf{E}'$, or

Alice: shared secret:
round($\mathbf{B}'\mathbf{S} \approx \mathbf{S}'\mathbf{A}\mathbf{S}$)

Bob: shared secret:
round($\mathbf{S}'\mathbf{B} \approx \mathbf{S}'\mathbf{A}\mathbf{S}$)

Thm: Key exchange is secure if decision learning with errors problem is hard.

Performance

Implementations

Our implementations

- Ring-LWE BCNS15
- LWE Frodo

Pure C implementations

Constant time

Compare with others

- RSA 3072-bit (OpenSSL 1.0.1f)
- ECDH `nistp256` (OpenSSL)

Use assembly code

- Ring-LWE NewHope
- NTRU `EES743EP1`
- SIDH (Isogenies) (MSR)

Pure C implementations

Standalone performance

	Speed		Communication		Quantum Security
RSA 3072-bit	Fast	4 ms	Small	0.3 KiB	
ECDH `nistp256`	Very fast	0.7 ms	Very small	0.03 KiB	
Ring-LWE BCNS	Fast	1.5 ms	Medium	4 KiB	80-bit
Ring-LWE NewHope	Very fast	0.2 ms	Medium	2 KiB	206-bit
NTRU `EES743EP1`	Fast	0.3–1.2 ms	Medium	1 KiB	128-bit
SIDH	Very slow	35–400 ms	Small	0.5 KiB	128-bit
LWE Frodo Recom.	Fast	1.4 ms	Large	11 KiB	130-bit
McBits*	Very fast	0.5 ms	Very large	360 KiB	161-bit

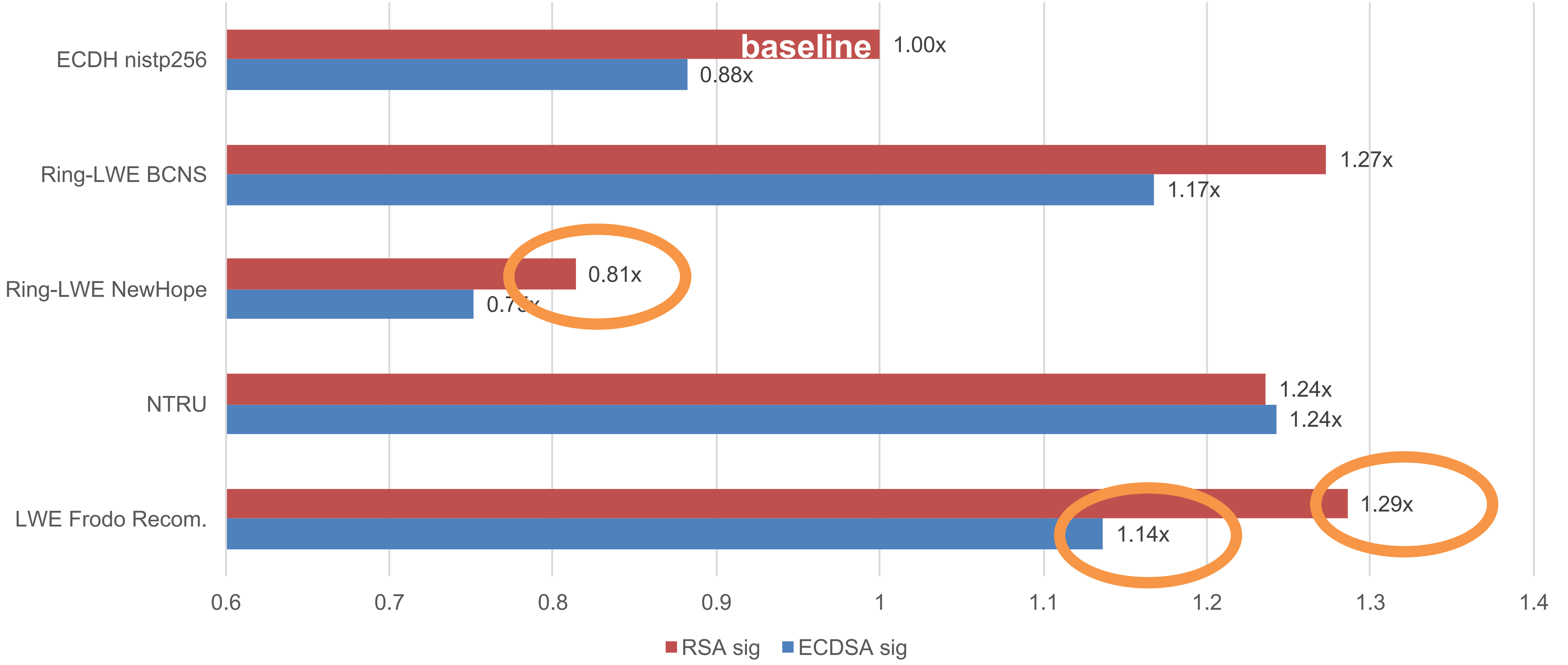
First 7 rows: x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – Google `n1-standard-4`
* McBits results from source paper [BCS13]

Note somewhat incomparable security levels

TLS handshake latency

compared to RSA sig + ECDH nistp256

smaller (left) is better



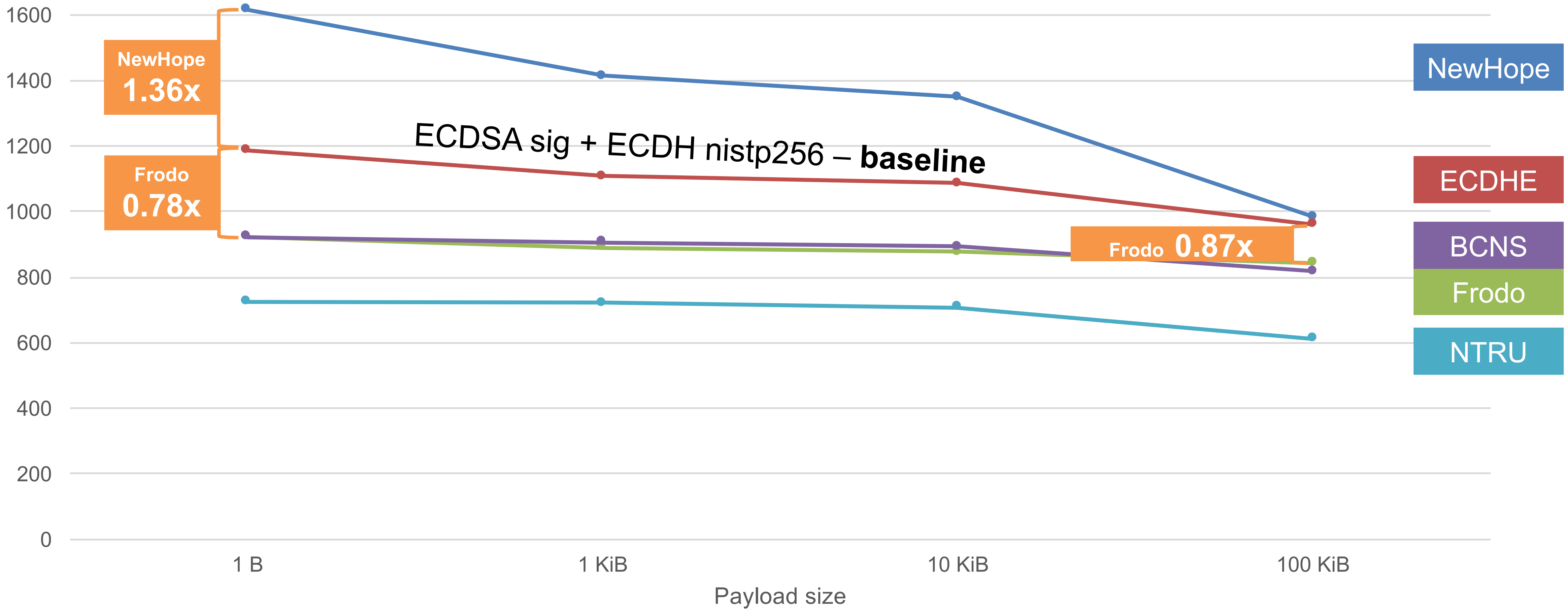
x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – server Google `n1-standard-4`, client `-32`

Note somewhat incomparable security levels

TLS connection throughput

ECDSA signatures

bigger (top) is better



x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – server Google `n1-standard-4`, client `-32`

Note somewhat incomparable security levels

Open Quantum Safe

- Open source C library
- Common interface for key exchange and digital signatures

1. Collect post-quantum implementations together
   - Our own software
   - Thin wrappers around existing open source implementations
   - Contributions from others

2. Enable direct comparison of implementations

3. Support prototype integration into application level protocols
   - Don’t need to re-do integration for each new primitive – how we did Frodo experiments

Summary

Post-quantum key exchange for the Internet

Douglas Stebila

McMaster University

- Lots of fun math in public key cryptography
  - Number theory
  - Groups, rings
  - Lattices
  - Elliptic curves
- Learning with errors problem
  - Difficulty based on lattice problem
  - Ring variant for smaller communication
- Building key exchange from LWE
  - Ring-LWE is fast and fairly small
  - LWE can achieve reasonable key sizes and runtime with more conservative assumption

Ring-LWE key exchange

- https://eprint.iacr.org/2014/599

LWE key exchange

- https://eprint.iacr.org/2016/659

Open Quantum Safe

- https://openquantumsafe.org/
- https://eprint.iacr.org/2016/1017

> "Thank God number theory is unsullied by any application."
>
> — Leonard Dickson (1874–1954)

Appendix

More on LWE and ring-LWE key exchange

Lyubashevsky, Peikert, Regev

Eurocrypt 2010

- Public key encryption from ring-LWE

Ding, Xie, Lin

ePrint 2012

- Key exchange from LWE and ring-LWE

Lindner, Peikert

ePrint 2010, CT-RSA 2011

- Public key encryption from LWE and ring-LWE
- Key exchange from LWE

Peikert

PQCrypto 2014

- Key encapsulation mechanism based on ring-LWE

Ring-LWE-DH key agreement

Public parameters		
Decision R-LWE parameters q, n, χ		
$a \xleftarrow{\$} \mathcal{U}(R_q)$		
Alice		**Bob**
$s, e \xleftarrow{\$} \chi$		$s', e' \xleftarrow{\$} \chi$
$\boxed{b \leftarrow as + e} \in R_q$	$\xrightarrow{b}$	$\boxed{b' \leftarrow as' + e'} \in R_q$
		$e'' \xleftarrow{\$} \chi$
		$\boxed{v \leftarrow bs' + e''} \in R_q$
		$\overline{v} \xleftarrow{\$} \mathrm{dbl}(v) \in R_{2q}$
	$\xleftarrow{b', c}$	$c \leftarrow \langle \overline{v} \rangle_{2q,2} \in \{0,1\}^n$
$k_A \leftarrow \mathrm{rec}(\boxed{2b's}, c) \in \{0,1\}^n$		$k_B \leftarrow \lfloor \overline{v} \rceil_{2q,2} \in \{0,1\}^n$

Secure if decision ring learning with errors problem is hard.

Sampling is expensive

Operation	Cycles	
	constant-time	non-constant-time
sample $\xleftarrow{\$} \chi$	1 042 700	668 000
FFT multiplication	342 800	—
FFT addition	1 660	—
$\mathrm{dbl}(\cdot)$ and crossrounding $\langle\cdot\rangle_{2q,2}$	23 500	21 300
rounding $\lfloor\cdot\rceil_{2q,2}$	5 500	3,700
reconciliation $\mathrm{rec}(\cdot,\cdot)$	14 400	6 800

“NewHope”

Alkim, Ducas, Pöppelman, Schwabe.
USENIX Security 2016

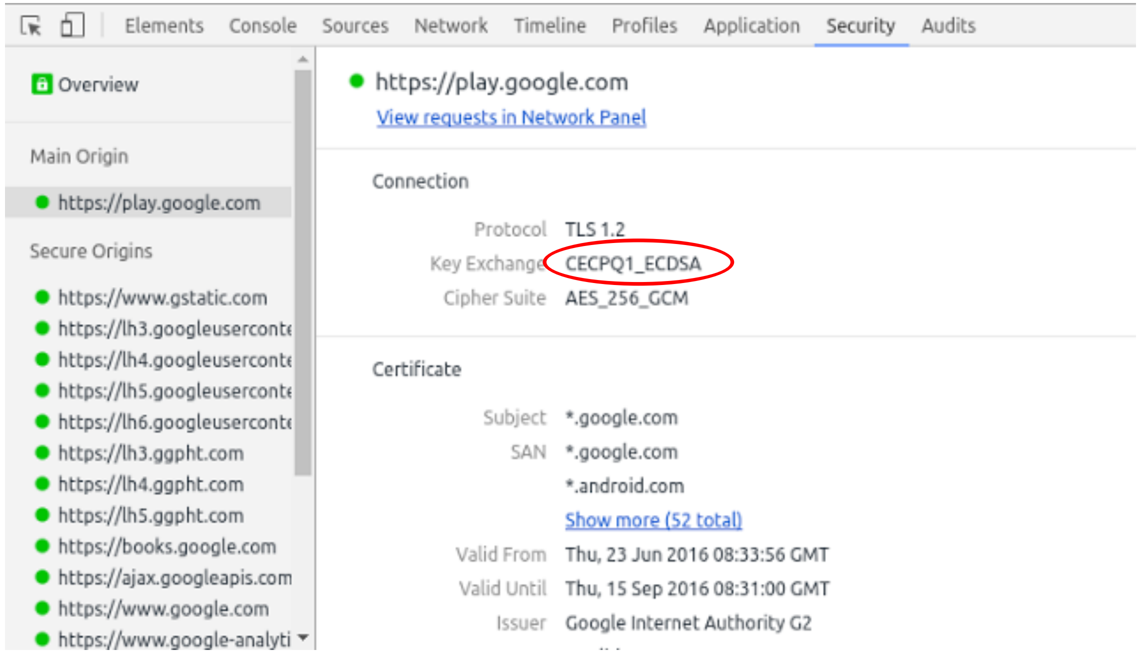
- New parameters
- Different error distribution
- Improved performance
- Pseudorandomly generated parameters

- Further performance improvements by others [GS16,LN16,…]

Google Security Blog

Experimenting with Post-Quantum Cryptography

July 7, 2016



https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

Decision learning with errors problem with short secrets

Definition. Let $n, q \in \mathbb{N}$. Let χ be a distribution over $\mathbb{Z}$.

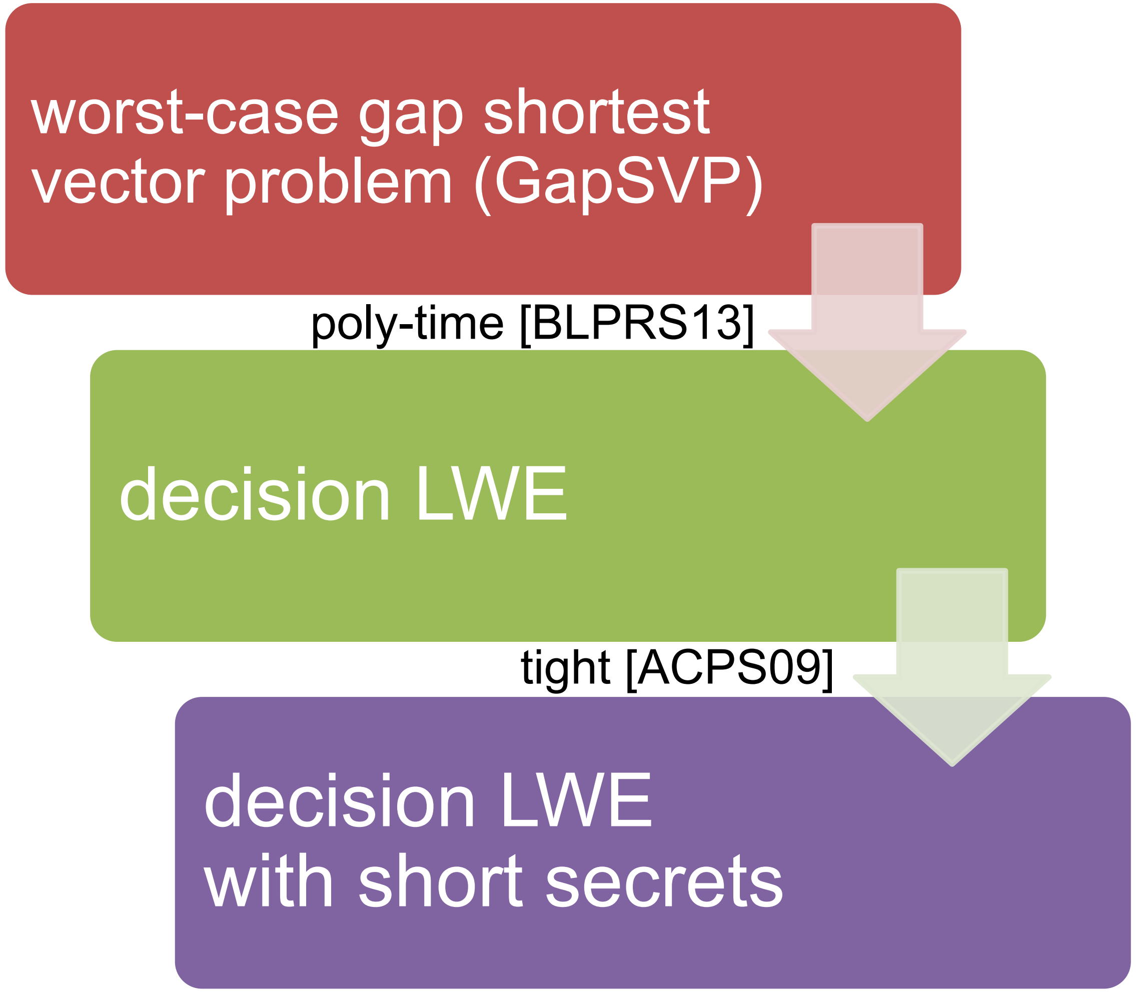
Let $\mathbf{s} \xleftarrow{\$} \chi^n$.

Define:

- $O_{\chi,\mathbf{s}}$: Sample $\mathbf{a} \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n)$, $e \xleftarrow{\$} \chi$; return $(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + e)$.
- U: Sample $(\mathbf{a}, b') \xleftarrow{\$} \mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$; return $(\mathbf{a}, b')$.

The *decision LWE problem with short secrets* for n, q, χ is to distinguish $O_{\chi,\mathbf{s}}$ from U.

Hardness of decision LWE



Practice:

- Assume the best way to solve DLWE is to solve LWE.
- Assume solving LWE involves a lattice reduction problem.
- Estimate parameters based on runtime of lattice reduction algorithms.
- (Ignore non-tightness.)

"Frodo": LWE-DH key agreement

Based on Lindner–Peikert LWE key agreement scheme

Alice		Bob
$\text{seed}_{\mathbf{A}} \xleftarrow{\$} U(\{0,1\}^s)$		
$\mathbf{A} \leftarrow \text{Gen}(\text{seed}_{\mathbf{A}})$		
$\mathbf{S}, \mathbf{E} \xleftarrow{\$} \chi(\mathbb{Z}_q^{n \times \overline{n}})$		
$\boxed{\mathbf{B} \leftarrow \mathbf{AS} + \mathbf{E}}$	$\xrightarrow[\in \{0,1\}^s \times \mathbb{Z}_q^{n \times \overline{n}}]{\text{seed}_{\mathbf{A}}, \mathbf{B}}$	
		$\mathbf{A} \leftarrow \text{Gen}(\text{seed}_{\mathbf{A}})$
		$\mathbf{S}', \mathbf{E}' \xleftarrow{\$} \chi(\mathbb{Z}_q^{\overline{m} \times n})$
		$\boxed{\mathbf{B}' \leftarrow \mathbf{S}'\mathbf{A} + \mathbf{E}'}$
		$\mathbf{E}'' \xleftarrow{\$} \chi(\mathbb{Z}_q^{\overline{m} \times \overline{n}})$
		$\boxed{\mathbf{V} \leftarrow \mathbf{S}'\mathbf{B} + \mathbf{E}''}$
	$\xleftarrow[\in \mathbb{Z}_q^{\overline{m} \times n} \times \mathbb{Z}_2^{\overline{m} \times \overline{n}}]{\mathbf{B}', \mathbf{C}}$	$\mathbf{C} \leftarrow \langle \mathbf{V} \rangle_{2^B}$
$K \leftarrow \text{rec}(\boxed{\mathbf{B}'\mathbf{S}}, \mathbf{C})$		$K \leftarrow \lfloor \mathbf{V} \rceil_{2^B}$

A generated pseudorandomly

Uses two matrix forms of LWE:
- Public key is n x n matrix
- Shared secret is m x n matrix

Secure if decision learning with errors problem is hard

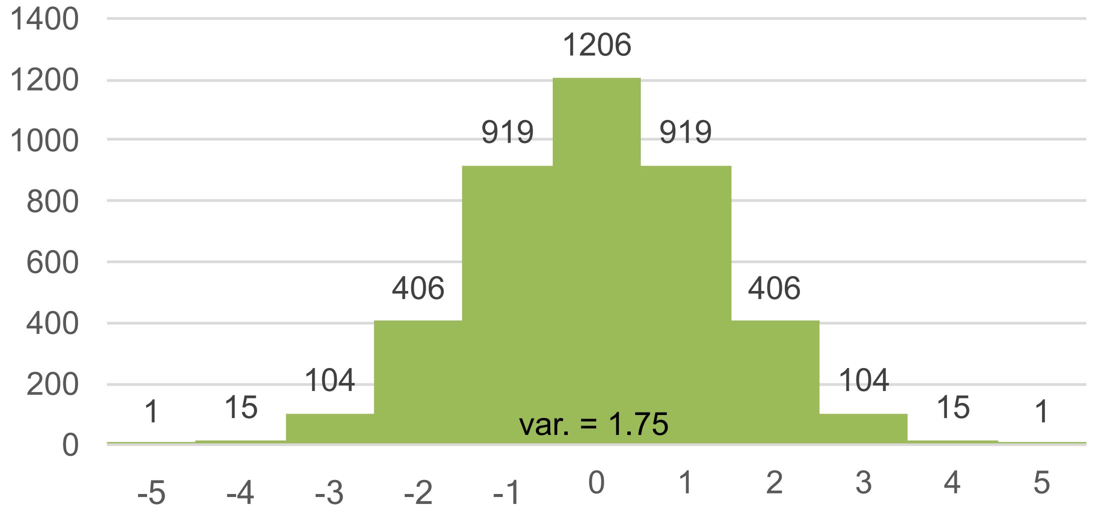
(and Gen is a secure PRF).

Rounding

- We extract 4 bits from each of the 64 matrix entries in the shared secret.
  - More granular form of previous rounding.

Parameter sizes, rounding, and error distribution all found via search scripts.

Error distribution



- Close to discrete Gaussian in terms of Rényi divergence (1.000301)
- Only requires 12 bits of randomness to sample

Parameters

All known variants of the sieving algorithm require a list of vectors to be created of this size

"Recommended"

- 144-bit classical security, 130-bit quantum security, 103-bit plausible lower bound

- $n = 752$, $m = 8$, $q = 2^{15}$
- χ = approximation to rounded Gaussian with 11 elements

- Failure: $2^{-38.9}$

- Total communication: 22.6 KiB

"Paranoid"

- 177-bit classical security, 161-bit quantum security, 128-bit plausible lower bound

- $n = 864$, $m = 8$, $q = 2^{15}$
- χ = approximation to rounded Gaussian with 13 elements

- Failure: $2^{-33.8}$

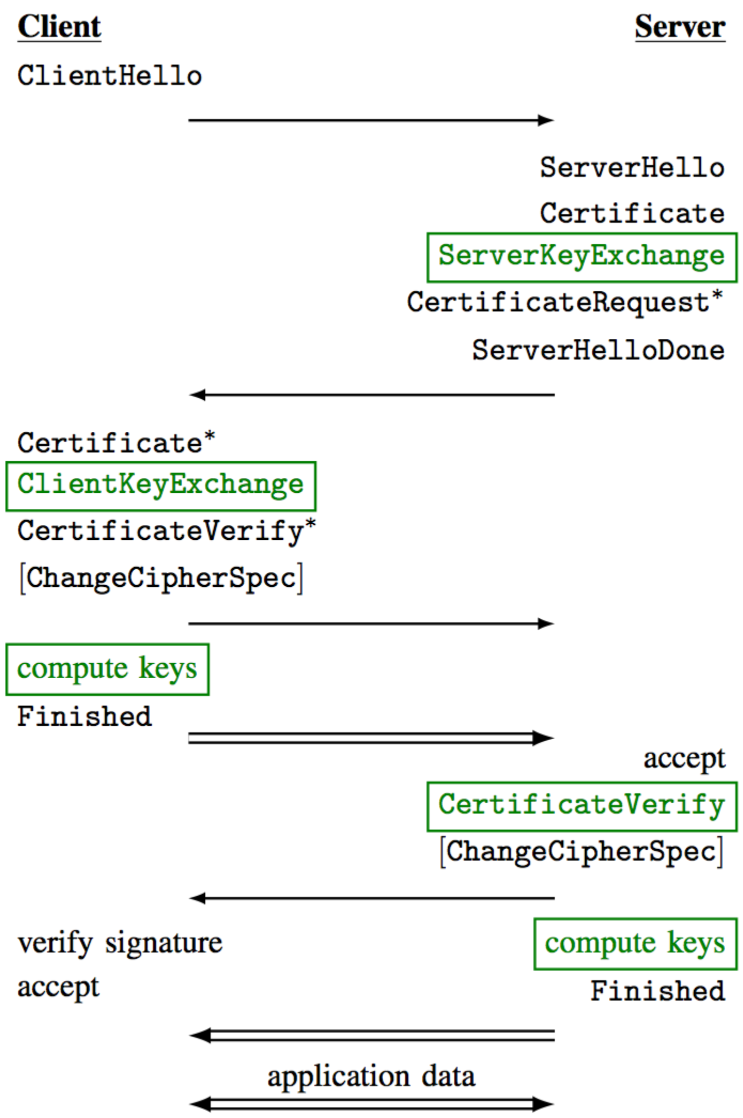
- Total communication: 25.9 KiB

TLS integration and performance

Integration into TLS 1.2

New ciphersuite:
`TLS-KEX-SIG-AES256-GCM-SHA384`

- `SIG` = RSA or ECDSA signatures for authentication
- `KEX` = Post-quantum key exchange
- AES-256 in GCM for authenticated encryption
- SHA-384 for HMAC-KDF



Security within TLS 1.2

Model:

- authenticated and confidential channel establishment (ACCE) [JKSS12]

Theorem:

- signed LWE/ring-LWE ciphersuite is ACCE-secure if underlying primitives (signatures, LWE/ring-LWE, authenticated encryption) are secure

Interesting provable security detail:

- TLS proofs use active security of unauthenticated key exchange (IND-CCA KEM or PRF-ODH assumption)
- Doesn't hold for basic BCNS15/Frodo/NewHope protocols
- Solution:
  - move server's signature to end of TLS handshake OR
  - use e.g. Fujisaki–Okamoto transform to convert passive to active security KEM

TLS performance

Handshake latency

- Time from when client sends first TCP packet till client receives first application data
- No load on server

Connection throughput

- Number of connections per second at server before server latency spikes

Hybrid ciphersuites

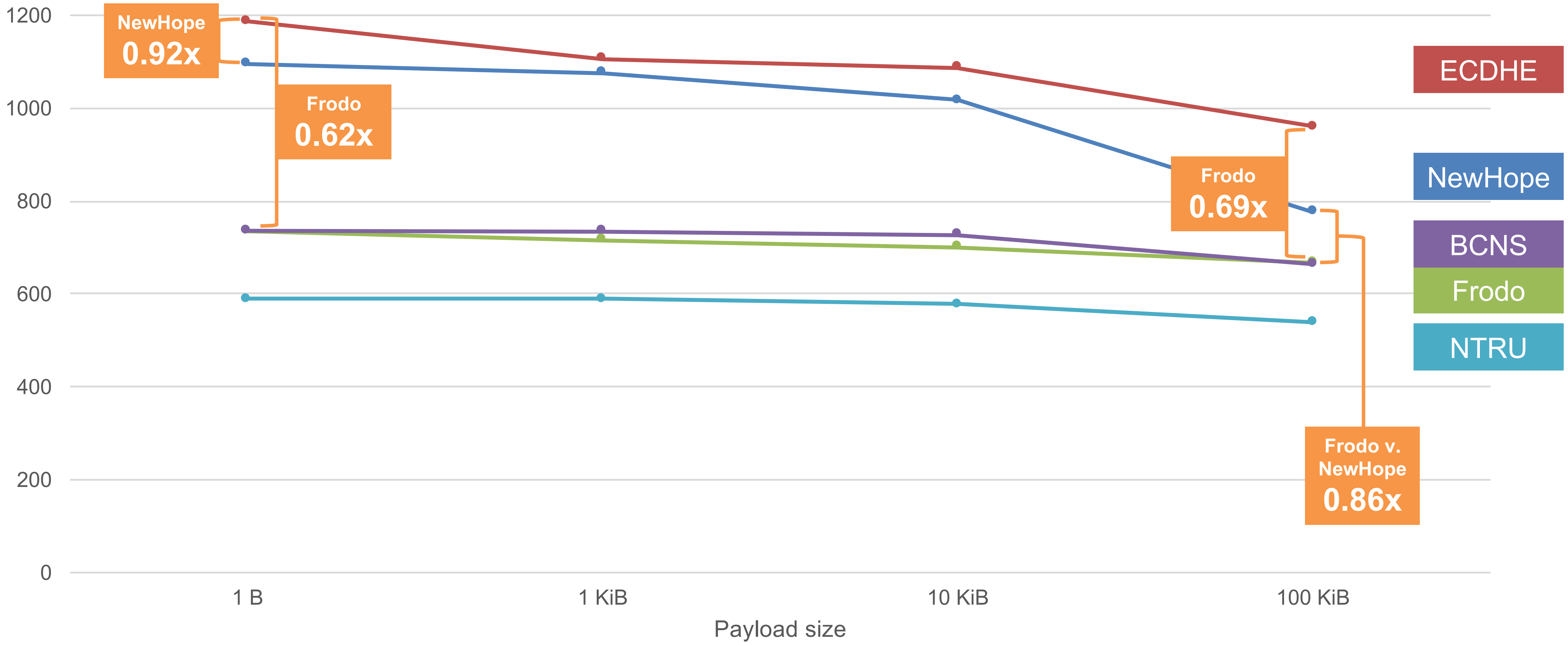
- Use both post-quantum key exchange and traditional key exchange
- Example:
  - ECDHE + NewHope
    - Used in Google Chrome experiment
  - ECDHE + Frodo

- Session key secure if either problem is hard
- Why use post-quantum?
  - (Potential) security against future quantum computer
- Why use ECDHE?
  - Security not lost against existing adversaries if post-quantum cryptanalysis advances

TLS connection throughput – hybrid w/ECDHE

ECDSA signatures

bigger (top) is better



x86_64, 2.6 GHz Intel Xeon E5 (Sandy Bridge) – server Google `n1-standard-4`, client `-32`

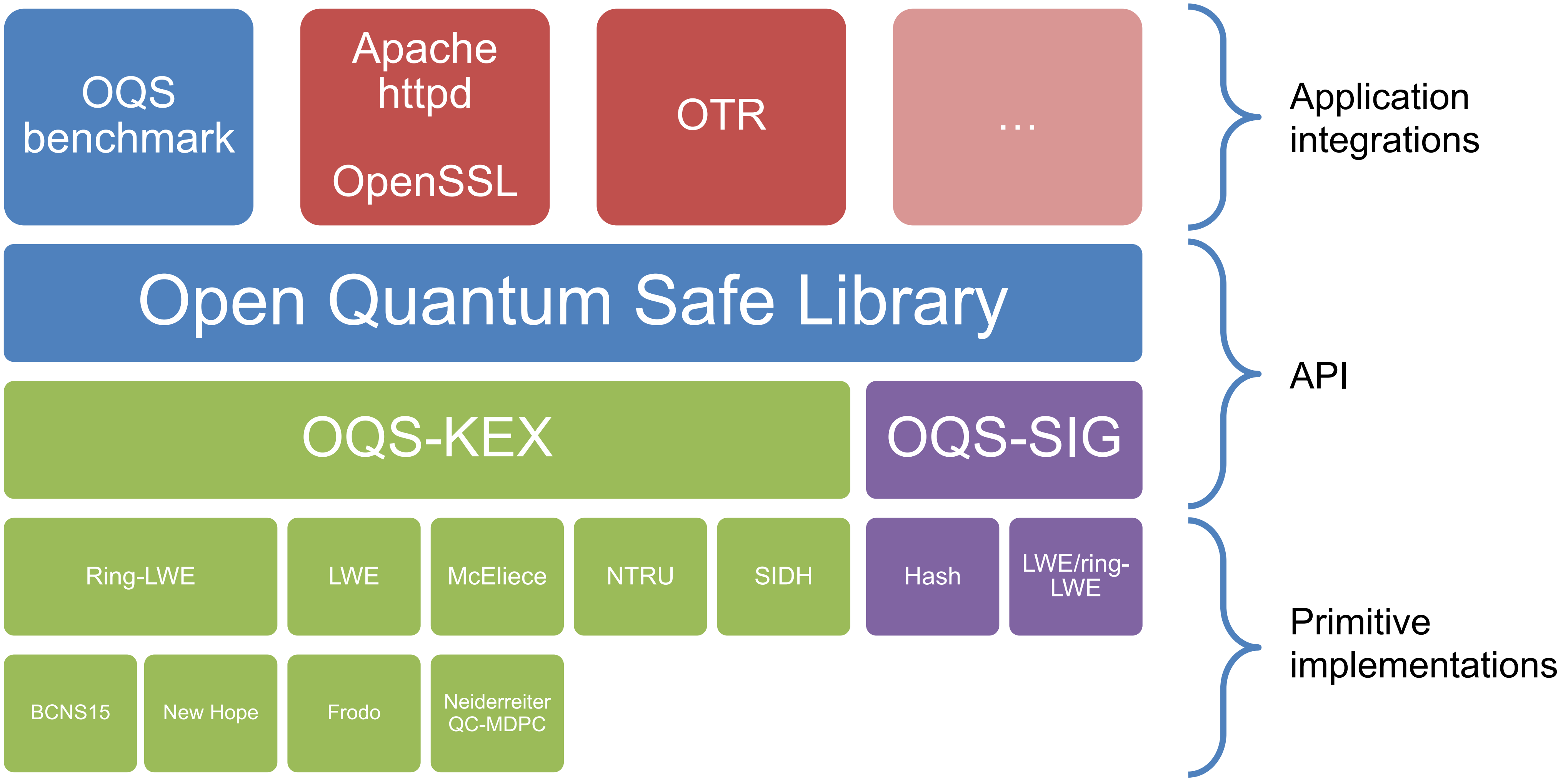
Note somewhat incomparable security levels

Open Quantum Safe

Collaboration with Mosca et al., University of Waterloo

https://openquantumsafe.org/

Open Quantum Safe architecture



Current status

- liboqs
  - ring-LWE key exchange using BCNS15
  - ring-LWE key exchange using NewHope*
  - LWE key exchange using Frodo
  - [alpha] code-based key exchange using Neiderreiter with quasi-cyclic medium-density parity check codes
- OpenSSL
  - integration into OpenSSL 1.0.2 head

Coming soon

- liboqs
  - benchmarking
  - key exchange:
    - SIDH, NTRU*
- Integrations into other applications
  - libotr

(* via wrappers)

OQC contributors and acknowledgements

Project leaders

- Michele Mosca and Douglas Stebila

Planning & discussions

- Scott Vanstone and Sherry Shannon Vanstone (Trustpoint)
- Matthew Campagna (Amazon Web Services)
- Alfred Menezes, Ian Goldberg, and Guang Gong (University of Waterloo)
- William Whyte and Zhenfei Zhang (Security Innovation)
- Jennifer Fernick, David Jao, and John Schanck (University of Waterloo)

Software contributors

- Mike Bender
- Tancrède Lepoint (SRI)
- Shravan Mishra (IQC)
- Christian Paquin (MSR)
- Alex Parent (IQC)
- Douglas Stebila (McMaster)
- Sebastian Verschoor (IQC)

+ Existing open-source code