Frodo: Take off the ring! Practical, Quantum-Secure Key Exchange from LWE

Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, Douglas Stebila
Quantum computer breaks public key crypto

Public key crypto (key agreement & signatures)
   RSA, DH, DSA
   ECDH, ECDSA

Symmetric key crypto
   AES-128

Hash functions
   SHA-256, SHA-3
Quantum computer breaks public key crypto

In the presence of a quantum computer:

- **Public key crypto (key agreement & signatures)**
  - RSA, DH, DSA
  - ECDH, ECDSA
    - No longer secure
  - Feb 2016: NIST calls for proposals

- **Symmetric key crypto**
  - AES-128
    - Needs longer keys

- **Hash functions**
  - SHA-256, SHA-3
    - Needs longer output
TLS protocol

Client

Client Hello

Client Key Exchange
Finished

Server

Server Hello
Certificate Chain

Server Key Exchange

Finished

Established a shared key K

Encrypted with K

http GET

http RESPONSE
TLS protocol

**Client**

- Client Hello
- Client Key Exchange
- Finished

**Server**

- Server Hello
- Certificate Chain
- Server Key Exchange
- Server Authentication w. digital signatures

**Past stays secure**

**NEED a NEW key agreement**

|key| x 2

**Payload Encryption**

**Key Agreement**
When to expect a quantum computer?

- **Google + UCSB**
  - predicts in **15 years** (Matteo Mariantoni at PQCrypto 2014)

- **Microsoft Research**
  - largest area of investment
    - (Mar 2016, quote from the president of MSR)

- **Intel**
  - invests **$50 million** (Sep 2014)

- **IBM**
  - publicly available quantum computer w. **5 qubits** (May 2016)

- **NSA**
  - invested **$80 million** (Jan 2014, E. Snowden through Washington Post)
  - suggests moving towards quantum-secure crypto! (Aug 2015)
Learning with Errors (LWE): new foundation for key agreement

For a random $A$, random small $x$ and $e$

$$(A, Ax + e) \approx (A, \text{random})$$

Learning with Errors (LWE): new foundation for key agreement

- LWE considered to be quantum resistant
- LWE has worst-case to average-case reductions
- A new (3rd) type of assumption (RSA: factoring, DH: solving discrete logarithm)
- Other crypto primitives from LWE (FHE, ABE, etc.)
Ring-LWE: an alternative assumption

Ring-LWE has additional structure (as well as NTRU)

- Matrices have additional structure (each row is a cyclic shift of the row above)
- Save communication (4KiB vs. 11KiB)
- More efficient computation
- Prior work: [LP10, DXL12, P14, BCNS15, “New Hope”16]
- “New Hope” is integrated in Chrome Canary (Jul 2016)

Recent security gap between structured and unstructured lattices found[^CDW16]
(not immediately applicable to ring-LWE)

LWE has NO additional structure
Be Careful with the Ring
DH key agreement

Diffie-Hellman key agreement

Client

Choose random x

Choose random y

\( g^x \)

\( g^y \)

\( g^{xy} \)

Server

Choose random x

\( g^y \)

\( g^{xy} \)

\( (g, g^x, g^y, g^{xy}) \)

looks like

\( (g, g^x, g^y, \text{random}) \)
DH key agreement translates to LWE

**Diffie-Hellman key agreement**

**Client**
- Choose random $y$
- Choose random $x$

**Server**
- $g^x$
- $g^y$

\[(g, g^x, g^y, g^{xy})\]
looks like\n\[(g, g^x, g^y, \text{random})\]

**LWE key agreement** [*DXL12*]

**Client**
- Choose random small $y$, $e'$

**Server**
- Choose random small $x$, $e$
- $Ax+e$
- $yA+e'$

\[(A, Ax+e, yA+e', \text{msb}*(yAx))\]
looks like\n\[(A, Ax+e, yA+e', \text{random})\]

---


LWE gives approximate key agreement

Client

\[ \text{seed} \leftarrow \text{Uniform} \]
\[ A := \text{PRG(\text{seed})} \]
\[ X, E \leftarrow \text{Gaussian}_\sigma \]
\[ YAX + E'X \]
\[ YAx + YE \]

Server

\[ \text{seed} \leftarrow \text{Uniform} \]
\[ A := \text{PRG(\text{seed})} \]
\[ X, E \leftarrow \text{Gaussian}_\sigma \]
\[ YAX + E'X \]

Security:
LWE + secure PRG

Secrets and noise:
Gaussians

A: always fresh

A: pseudorandom
Parameters are chosen to minimize communication

- modulus $q \in [2^{10}, ..., 2^{16}]$
- dimension $n \in [256, ..., 900]$
- noise deviation $\sigma \in [1,2]$ over reals
- number of extracted bits $B \in [1,..., \log q]$

Search for $(q,n,\sigma,B)$ that minimizes communication and
- classical/quantum attacks run in $> 2^{128}$
- failure probability $< 2^{-32}$
We suggest two sets of parameters

<table>
<thead>
<tr>
<th>Recommended</th>
<th>Paranoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q = 2^{15}$</td>
<td>$q = 2^{15}$</td>
</tr>
<tr>
<td>$n = 752$</td>
<td>$n = 864$</td>
</tr>
<tr>
<td>$\sigma = 1.75$</td>
<td>$\sigma = 1.75$</td>
</tr>
<tr>
<td>$B = 4$</td>
<td>$B = 4$</td>
</tr>
<tr>
<td>130 bits of quantum security</td>
<td>161 bits of quantum security</td>
</tr>
<tr>
<td>Total communication: <strong>22.57 KiB</strong></td>
<td>Total communication: <strong>25.93 KiB</strong></td>
</tr>
</tbody>
</table>

Reaches complexity lower bound of $2^{138}$ for sieving algorithms.
Table noise distribution minimizes security loss

- [BLLSS15]: bound the security loss when substituting distributions (using Renyi divergence)
- “NewHope”: substitute Gaussian for Binomial
- Our work: find optimal discrete distributions minimizing security loss and the number of uniformly random bits

Example:

- Needs only 12 random bits per sample
- Look-up table size: 14 Bytes

Approximation to Gaussian with std 1.75

---

Our implementation

● Constant time, pure C based on OQS framework[1]
● Compared with:
  ○ RSA 3072
  ○ ECDHE nistp256
  ○ and all implemented quantum resistant protocols
● New lattice ciphersuites in OpenSSL:
  - LWE_(RSA or ECDSA)_WITH_AES_256_GCM_SHA384
  - LWE_ECDHE_(RSA or ECDSA)_WITH_AES_256_GCM_SHA384

[1] Open Quantum Safe project by Michele Mosca and Douglas Stebila
openquantumsafe.org
## Standalone performance of key agreement (one sided)

<table>
<thead>
<tr>
<th>Method</th>
<th>Speed (ms)</th>
<th>Network (KiB)</th>
<th>Quantum security</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA 3072</td>
<td>4</td>
<td>0.3</td>
<td>-</td>
</tr>
<tr>
<td>ECDHE nistp256</td>
<td>0.7</td>
<td>0.03</td>
<td>-</td>
</tr>
<tr>
<td>NewHope (RLWE)</td>
<td>0.2</td>
<td>2</td>
<td>206</td>
</tr>
<tr>
<td>NTRU EES743EP1</td>
<td>0.3–1.2</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>Frodo (LWE)</td>
<td>1.4</td>
<td>11</td>
<td>130</td>
</tr>
<tr>
<td>SIDH</td>
<td>35–400</td>
<td>0.5</td>
<td>128</td>
</tr>
<tr>
<td>McBits (McEliece)</td>
<td>0.5</td>
<td>360</td>
<td>161</td>
</tr>
</tbody>
</table>

Most widely used ciphers:
- RSA 3072
- ECDHE nistp256
- NewHope (RLWE)
- NTRU EES743EP1
- Frodo (LWE)
- SIDH

Lattice based ciphers:
- McBits (McEliece)

Others:

First 6 rows: x86_64, 2.6GHz Intel Xeon E5 (Sandy Bridge) - Google n1-standard-4
McBits results from source paper [BCS13]
## Comparison of lattice-based key agreements to ECDHE

<table>
<thead>
<tr>
<th></th>
<th>Speed (one side)</th>
<th>Network (total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECDHE nistp256</td>
<td>0.7ms</td>
<td>0.06 KiB</td>
</tr>
<tr>
<td>NewHope (RLWE)</td>
<td>0.3x</td>
<td>3.9 KiB</td>
</tr>
<tr>
<td>NTRU EES743EP1</td>
<td>1x</td>
<td>2.1 KiB</td>
</tr>
<tr>
<td>Frodo (LWE)</td>
<td>2x</td>
<td>22.7 KiB</td>
</tr>
</tbody>
</table>

Cert chain for [https://www.google.com](https://www.google.com) is 3KiB
Switching to Hybrids

LWE_ECDHE_(RSA or ECDSA)_WITH_AES_256_GCM_SHA384

- Use both post-quantum key-agreement and traditional key-agreement together
- Example:
  - ECDHE + NewHope
    - Used in Google experiment*
  - ECDHE + Frodo
- Session key is secure if at least one problem is hard
- Use post-quantum: to prevent future (quantum) attacks
- Use ECDHE: to prevent classical attacks against post-quantum primitives

Throughput for TLS - hybrid (with ECDHE)

![Graph showing throughput for TLS with different key exchange methods.]

- ECDHE: 1.5 connections per second
- ECDHE+NewHope: 1.2 connections per second
- ECDHE+Frodo (not shown)
- ECDHE+NTRU (not shown)

x86_64, 2.6GHz Intel Xeon E5 (Sandy Bridge) - Google n1-standard-4
Throughput for TLS - hybrid (with ECDHE)

- Connections per second
- Throughput for TLS - hybrid (with ECDHE)

x86_64, 2.6GHz Intel Xeon E5 (Sandy Bridge) - Google n1-standard-4
Contributions

● Key-agreement protocol from LWE
● Implementation of LWE key agreement
  ○ OpenSSL integration
  ○ Micro/macro benchmarks
● Better rounding, extracting more bits
● New methods for noise sampling
● Scripts for parameters’ search
● All code is open source: github.com/lwe-frodo
  github.com/open-quantum-safe
● And…. Frodo took off the ring! :}
Thank you!
eprint.iacr.org/2016/659.pdf
Generalized rounding equalizes the keys

Toy example:
\[ q = 2^7 \]
\[ E < 2^3 \]

**TASK:** derive a common key from \( K \) and \( K' \), where \( E = K' - K \) is small

**SOLUTION:** take the most significant bits

**PROBLEM:** they can be altered by the carry from \( E \)

**FIX:** make the client send an indicator bit*

---

LWE with generalized rounding gives exact key agreement

Client

\[
\begin{align*}
A &:= \text{PRG}(\text{seed}) \in \mathbb{Z}_q^{n \times n} \\
Y, E' &\leftarrow \chi(\mathbb{Z}_q^{m \times n}) \\
B' &:= YA + E' \in \mathbb{Z}_q^{m \times n}
\end{align*}
\]

\[
K = YB = YAX + YE
\]

Server

\[
\begin{align*}
\text{seed} &\leftarrow U(\{0, 1\}^\lambda) \\
A &:= \text{PRG}(\text{seed}) \in \mathbb{Z}_q^{n \times n} \\
X, E &\leftarrow \chi(\mathbb{Z}_q^{n \times m}) \\
B &:= AX + E \in \mathbb{Z}_q^{n \times m}
\end{align*}
\]

\[
B', C \in \mathbb{Z}_2^{m \times m}
\]

\[
K' = B'X = S'AX + E'X
\]

pre-master key

rounding
Prior work built key agreement from **ring-LWE**

**Ding, Xie, Lin 2012**  
*ePrint 2012*  
Key agreement from LWE and **ring-LWE**

**Peikert 2014**  
PQCrypto 2014  
More efficient key agreement mechanism based on **ring-LWE**

**BCNS 2015**  
Key agreement from **ring-LWE**:  
- Selected parameters  
- Integrated into TLS  
- Measured performance

“**NewHope**” 2016  
Alkim, Ducas, Pöppelman, Scwabe  
USENIX Security 2016  
Key agreement from **ring-LWE**:  
- New parameters  
- Different error distribution  
- Improved performance

---

Further improvements: GS16, LN16, ...
LWE handshake in more details

Client

Choose random small $x, e$

Server

$\cdot m$ is chosen s.t. $K$ has enough entropy to derive the key

Choose random small $y, e'$