Post-quantum Key Exchange from LWE

Jintai Ding, University of Cincinnati

Worcester Polytechnic Institute, September 2, 2016
The context of our work - PKC

- Diffie-Hellman’s revolution idea – Public key cryptography
- Symmetric systems versus Asymmetric systems
- The work of RSA – the critical role of mathematics
- The Internet and the PKCs
  Internet can not work without PKCs.
The context of our work - PQC

- Shor’s quantum algorithm
- Post-quantum cryptography
  Develop public key cryptosystems that could resist future quantum computer attacks
The Preparation for the Future

- The first Quantum-Safe-Crypto Workshop
  26 - 27 September, 2013

  **ETSI – the European Telecommunications Standards Institute** at SOPHIA ANTIPOLIS, FRANCE

- The second Quantum-Safe-Crypto Workshop
  6 - 2 October, 2014, Ottawa, Canada

  White paper

- The Quantum-Safe-Crypto Workshop at **NIST: National Institute of Standard of Technology**,
  April 7-8, 2015, Washington DC
Fig. 1. Seven stages in the development of quantum information processing. Each advancement requires mastery of the preceding stages, but each also represents a continuing task that must be perfected in parallel with the others. Superconducting qubits are the only solid-state implementation at the third stage, and they now aim at reaching the fourth stage (green arrow). In the domain of atomic physics and quantum optics, the third stage had been previously attained by trapped ions and by Rydberg atoms. No implementation has yet reached the fourth stage, where a logical qubit can be stored, via error correction, for a time substantially longer than the decoherence time of its physical qubit components.
A commercial for PQC from NSA

Cryptography Today

In the current global environment, rapid and secure information sharing is important to protect our Nation, its citizens and its interests. Strong cryptographic algorithms and secure protocol standards are vital tools that contribute to our national security and help address the ubiquitous need for secure, interoperable communications.

Currently, Suite B cryptographic algorithms are specified by the National Institute of Standards and Technology (NIST) and are used by NSA’s Information Assurance Directorate in solutions approved for protecting classified and unclassified National Security Systems (NSS). Below, we announce preliminary plans for transitioning to quantum resistant algorithms.

Background

1AD will initiate a transition to quantum resistant algorithms in the not too distant future. Based on experience in deploying Suite B, we have determined to start planning and communicating early about the upcoming transition to quantum resistant algorithms. Our ultimate goal is to provide cost effective security against a potential quantum computer. We are working with partners across the USG, vendors, and standards bodies to ensure there is a clear plan for
Practical Challenge

- Quantum computing will break many public-key cryptographic algorithms/schemes
  - Key agreement (e.g. DH and MQV)
  - Digital signatures (e.g. RSA and DSA)
  - Encryption (e.g. RSA)

- These algorithms have been used to protect Internet protocols (e.g. IPsec) and applications (e.g. TLS)

- NIST is studying “quantum-safe” replacements

What do we really need?—a slides of L. Chen from NIST
In PQC2016 in Japan, NIST make a call for quantum resistant algorithms by Dustin Moody

*Deadline: November 2017*
Post Quantum Needs – Functionality

- Key Exchange – for secure communications

- Signatures – for Authentication
Key Exchange Applications — SSL/TLS

- RSA
- Diffie–Hellman
- Our goal – replacements for post quantum world
Diffie-Hellman Key Exchange

\[ (g^b)^a \]  \hspace{2cm}  \[ g^a \]  \hspace{2cm}  \[ (g^a)^b \]
Generalizing DH

• DH works because maps $f(x) = x^a$ and $h(x) = x^b$ commute
  \[ f \circ h = h \circ f, \]

  \(\circ\) – composition

Nonlinearity

• Many attempts – Braid group etc
Generalizing DH

- When do we have commuting *nonlinear* maps?
  - Powers of $x$ (normal DH)
  - Iterates of a polynomial
  - J. Ritt (1923) – Power polynomials, Chebyshev polynomials. Elliptic curve
Who is J. Ritt: 1893-1951
Who is J. Ritt: 1923: PERMUTABLE RATIONAL FUNCTIONS

J. Ritt (1923) – Power polynomials, Chebychev polynomials. Elliptic curve
Generalizing DH

Our basic idea — adding "small" noise or perturbation:

- (Ring) LWE approximately commutes—use to build DH generalization

From

\[(s_1 \times a) \times s_2 = s_1 \times (a \times s_2)\]

to

\[(as_1 + e_1)s_2 \approx s_1 as_2 \approx (as_2 + e_2)s_1\]
A historical Note

Our basic idea — adding "small" noise or perturbation is not new!!!

- GCHQ – Communications-Electronics Security Group (CESG)  
  - James Elias – "Invention of non-secret encryption" 1969  
  - Clifford Cocks – RSA, Malcolm Williamson – DH, 1973

- The forgotten inspiration of J. Elias –  
  "Ellis said that the idea first occurred to him after reading a paper from World War II by someone at Bell Labs describing a way to protect voice communications by the receiver adding (and then later subtracting) random noise (possibly this 1944 paper[4] or the 1945 paper co-authored by Claude Shannon)"  
  - Wikipedia
Learning with Errors [2006, Regev]

\[
\begin{pmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_m
\end{pmatrix} = \begin{pmatrix}
  a_{11} & a_{12} & \cdots & a_{1n} \\
  a_{21} & a_{22} & \cdots & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{m1} & a_{m2} & \cdots & a_{mn}
\end{pmatrix} \begin{pmatrix}
  s_1 \\
  s_2 \\
  \vdots \\
  s_n
\end{pmatrix} + \begin{pmatrix}
  e_1 \\
  e_2 \\
  \vdots \\
  e_m
\end{pmatrix}
\]

- Approximate system over \( \mathbb{Z}_q \)
- Hard to find \( \mathbf{s} \) from \( \mathbf{A}, \mathbf{b} \).
- Hard to tell if \( \mathbf{s} \) even exists
- Reduction to lattice approximation problems
Definition

Let $n$ be a power of 2, $q \equiv 1 \pmod{2n}$ prime. Define the ring

$$R_q = \frac{\mathbb{Z}_q[x]}{(x^n + 1)}.$$

- Again, $b = as + e$ hard to find $s$
- Hard to distinguish from uniform $b$
- Approximation problems on ideal lattices
- More efficient than standard LWE
Diffie-Hellman from Ideal Lattices

- Public $a \in R_q$. Acts like generator $g$ in DH.

$p_A = as_A + 2e_A$

$p_B = as_B + 2e_B$
Diffie-Hellman from Ideal Lattices

\[
\begin{align*}
p_A &= as_A + 2e_A \\
p_B &= as_B + 2e_B
\end{align*}
\]

\[
\begin{align*}
k_A &= s_Ap_B = aS_AS_B + 2S_Ae_B \quad \approx \quad k_B &= p_As_B = aS_AS_B + 2S_Be_A
\end{align*}
\]

- Public \( a \in \mathbb{R}_q \). Acts like generator \( g \) in DH.
- Each side’s key is only \textit{approximately} equal to the other.
- Difference is even—same low bits.
- No authentication—MitM.
Wrap-around Illustrated

- Difference 2, both even.
Wrap-around Illustrated

- Difference 2, both even.
- But wait! If $q = 5$, $\mathbb{Z}_q = \{-2, -1, 0, 1, 2\}$.
- 4 becomes $-1$, now parities disagree!
Compensating for Wrap-Around

- \( g = 2S_A e_B - 2S_B e_A \).
- Recall: \( |g^{(j)}| < \frac{q}{8} \)
- Define \( E = \{-\left\lfloor \frac{q}{4} \right\rfloor, \ldots, \left\lfloor \frac{q}{4} \right\rfloor\} \). Middle half of \( \mathbb{Z}_q \).
- If \( k_B^{(j)} \in E \), no wrap-around occurs; \( k_A^{(j)} \equiv k_B^{(j)} \).
- If \( k_B^{(j)} \notin E \), then \( k_B^{(j)} + \frac{q-1}{2} \in E \)
- If \( k_B^{(j)} \notin E \), \( k_A^{(j)} + \frac{q-1}{2} \equiv k_B^{(j)} + \frac{q-1}{2} \).
Define \( w_B^{(j)} = \begin{cases} 0 & k_B^{(j)} \in E, \\ 1 & k_B^{(j)} \notin E. \end{cases} \) Then \( k_B^{(j)} + w_B^{(j)} \frac{q-1}{2} \in E. \)

Also, \( k_B^{(j)} + w_B^{(j)} \frac{q-1}{2} \equiv k_A^{(j)} + w_B^{(j)} \frac{q-1}{2} \pmod 2. \)

- \( k_B^{(j)} + w_B^{(j)} \frac{q-1}{2} \pmod q \pmod 2 = k_A^{(j)} + w_B^{(j)} \frac{q-1}{2} \pmod q \pmod 2. \)
- Wrap-around correction \( w_B = (w_B^{(0)}, w_B^{(1)}, \ldots, w_B^{(n-1)}) \)
- \( \sigma_B = k_B + w_B \frac{q-1}{2} \pmod 2. \)
- \( \sigma_A = k_A + w_B \frac{q-1}{2} \pmod 2. \)
Rounding Intuition – Region Division

- Inner region
- Outer region

- \(-\frac{q-1}{2}\)
- \(-\frac{q}{4}\)
- 0
- \(\frac{q}{4}\)
- \(\frac{q-1}{2}\)
Rounding Intuition – Inner Region

- (q-1)/2
- q/4
0
q/4
(q-1)/2

Main value
Rounding Intuition – Outer Region problem

The problem – Additional modular operation

Errors

-\frac{(q-1)}{2} \quad -\frac{q}{4} \quad 0 \quad \frac{q}{4} \quad \frac{(q-1)}{2}

Main value

Jintai Ding AKE from rLWE
Rounding Intuition

By adding \((q-1)/2\), two regions swap.
Authentication: HMQV – To Resist Man-in-the-middle Attack and Achieve Forward Security

- Static keys $a, b$; tied to each party’s identity.
Authentication: HMQV – To Resist Man-in-the-middle Attack and Achieve Forward Security

- Static keys $a, b$; tied to each party’s identity.
- Ephemeral keys $x, y$: forward security.
Authentication: HMQV – To Resist Man-in-the-middle Attack and Achieve Forward Security

- Static keys $a$, $b$; tied to each party’s identity.
- Ephemeral keys $x$, $y$: **forward security**.
- Publicly derivable computations $d$, $e$. 
Motivation
Lattice-based Key Exchange
The Provable Security
Implementations

Lattice Diffie-Hellman
HMQV
Lattice HMQV

Authentication: HMQV – To Resist Man-in-the-middle Attack and Achieve Forward Security

\[ (g^y(g^b)^e)x + da = g^{(y+eb)(x+da)} = (g^x(g^a)^d)y + eb \]

- Static keys \( a, b \); tied to each party’s identity.
- Ephemeral keys \( x, y \): **forward security**.
- Publicly derivable computations \( d, e \).
- Shared key is \( K = H(\sigma_A) = H(\sigma_B) = H(g^{(y+be)(x+da)}) \).
HMQV from Ideal Lattices

\[ p_A = a s_A + 2e_A \]

\[ p_B = a s_B + 2e_B \]

- \( p_A, p_B \) as above. Public, static keys for authentication.
Motivation
Lattice-based Key Exchange
The Provable Security
Implementations

Lattice Diffie-Hellman
HMQV
Lattice HMQV

HMQV from Ideal Lattices

\[ p_A = as_A + 2e_A, \quad x_A = ar_A + 2f_A \]
\[ p_B = as_B + 2e_B, \quad y_B = ar_B + 2f_B \]

- \( p_A, p_B \) as above. Public, static keys for authentication
- \( x_A, y_B \) same form. Forward secrecy.
HMQV from Ideal Lattices

\[ p_A = a s_A + 2e_A, \quad x_A = a r_A + 2f_A \]

\[ p_B = a s_B + 2e_B, \quad y_B = a r_B + 2f_B \]

\[ k_A = (p_B d + y_B)(s_A c + r_A) + 2d g_A \approx (a S_B d + a r_B)(s_A c + r_A) \]

\[ k_B = (p_A c + x_A)(s_B d + r_B) + 2c g_B \approx (a S_A c + a r_A)(s_B d + r_B) \]

- \( p_A, p_B \) as above. Public, static keys for authentication.
- \( x_A, y_B \) same form. Forward secrecy.
- \( c, d \) publicly derivable; \( g_A, g_B \) random, small.
Key Derivation

Obtaining shared secret from approximate shared secret:

\[ k_A = (k_A^{(0)}, k_A^{(1)}, \ldots, k_A^{(n-1)}) \]
\[ k_B = (k_B^{(0)}, k_B^{(1)}, \ldots, k_B^{(n-1)}) \]
\[ \tilde{g} = (g^{(0)}, g^{(1)}, \ldots, g^{(n-1)}) \]

\[ k_A - k_B = 2\tilde{g} \]

\[ k_A \equiv k_B \pmod{2} \]
Key Derivation

Obtaining shared secret from approximate shared secret:

\[ k_A = (k_A^{(0)}, k_A^{(1)}, \ldots, k_A^{(n-1)}) \]
\[ k_B = (k_B^{(0)}, k_B^{(1)}, \ldots, k_B^{(n-1)}) \]
\[ \tilde{g} = (g^{(0)}, g^{(1)}, \ldots, g^{(n-1)}) \]

\[ k_A - k_B = 2\tilde{g} \]
\[ k_A \equiv k_B \pmod{2} \]

- Each \( k_A^{(j)} = k_B^{(j)} + 2g^{(j)} \).
- Each \( g^{(j)} \) is small (\( |g^{(j)}| < \frac{q}{8} \)).
- Matching coefficients differ by small multiple of 2
- Take each coefficient mod 2, get n bit secret
HMQV from Ideal Lattices—Corrected
HMQV from Ideal Lattices—Corrected

\[ p_A, x_A \]

\[ k_B \]
Motivation
Lattice-based Key Exchange
The Provable Security
Implementations

Lattice Diffie-Hellman
HMQV
Lattice HMQV

HMQV from Ideal Lattices—Corrected

\[ p_A, x_A \]

\[ p_B, y_B, w_B \]

\[ k_B \]
HMQV from Ideal Lattices—Corrected

\[ p_A, x_A \]

\[ p_B, y_B, w_B \]

\[ k_A \]

\[ k_B \]
HMQV from Ideal Lattices—Corrected

\[ p_A, x_A \]

\[ p_B, y_B, w_B \]

\( k_A \)

\[ \sigma_A = \sigma_B \]

\( k_B \)
HMQV from Ideal Lattices—Corrected

\[ p_A, x_A \rightarrow \quad p_B, y_B, w_B \]

\[ k_A \quad \sigma_A = \sigma_B \quad k_B \]

\[ H \]

Key
Proof Games

Proof proceeds by series of games:
- Begin with simulated protocol
- Replace one hash output with true random value, back-program random oracle
- Adversary cannot distinguish from previous game
- Eventually, if original protocol can be distinguished from random, rLWE can be broken
- The modification using rejecting sampling
Forward Security

- If static keys compromised, previous session keys remain secure
- Notion captured in proof by giving adversaries ability to corrupt static key
- Use Bellare–Rogaway model restricted to two-pass
Quantum Hardness

- Proof uses Random Oracle Model—quantum implications not fully understood
- Important step to post quantum key exchange
<table>
<thead>
<tr>
<th>Parameters</th>
<th>$n$</th>
<th>Security (expt.)</th>
<th>$\alpha$</th>
<th>$\gamma$</th>
<th>$\log \frac{\beta}{\alpha}$</th>
<th>$\log q$ (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I*</td>
<td>1024</td>
<td>80 bits</td>
<td>3.397</td>
<td>101.919</td>
<td>8.5</td>
<td>40</td>
</tr>
<tr>
<td>II</td>
<td>2048</td>
<td>80 bits</td>
<td>3.397</td>
<td>161.371</td>
<td>27</td>
<td>78</td>
</tr>
<tr>
<td>III</td>
<td>2048</td>
<td>128 bits</td>
<td>3.397</td>
<td>161.371</td>
<td>19</td>
<td>63</td>
</tr>
<tr>
<td>IV</td>
<td>4096</td>
<td>128 bits</td>
<td>3.397</td>
<td>256.495</td>
<td>50</td>
<td>125</td>
</tr>
<tr>
<td>V</td>
<td>4096</td>
<td>192 bits</td>
<td>3.397</td>
<td>256.495</td>
<td>36</td>
<td>97</td>
</tr>
<tr>
<td>VI</td>
<td>4096</td>
<td>256 bits</td>
<td>3.397</td>
<td>256.495</td>
<td>28</td>
<td>81</td>
</tr>
</tbody>
</table>
## Communication Overheads

<table>
<thead>
<tr>
<th>Choice of Parameters</th>
<th>Size (KB)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pk</td>
<td>sk (expt.)</td>
<td>init. msg</td>
</tr>
<tr>
<td>I*</td>
<td>5 KB</td>
<td>0.75 KB</td>
<td>5 KB</td>
</tr>
<tr>
<td>II</td>
<td>19.5 KB</td>
<td>1.5 KB</td>
<td>19.5 KB</td>
</tr>
<tr>
<td>III</td>
<td>15.75 KB</td>
<td>1.5 KB</td>
<td>15.75 KB</td>
</tr>
<tr>
<td>IV</td>
<td>62.5 KB</td>
<td>3 KB</td>
<td>62.5 KB</td>
</tr>
<tr>
<td>V</td>
<td>48.5 KB</td>
<td>3 KB</td>
<td>48.5 KB</td>
</tr>
<tr>
<td>VI</td>
<td>40.5 KB</td>
<td>3 KB</td>
<td>40.5 KB</td>
</tr>
</tbody>
</table>

The bound $6\alpha$ with $\text{erfc}(6) \approx 2^{-55}$ is used to estimate the size of secret keys.
Motivation
Lattice-based Key Exchange
The Provable Security
Implementations

Table: Timings of Proof-of-Concept Implementations in ms (The figures in the parentheses indicate the timings with pre-computing. For comparison, by simply using the “speed” command in openssl on the same machine, the timing for dsa1024 signing algorithm is about 0.7 ms, and for dsa2048 is about 2.3 ms).
We build a simple AKE based on RLWE.
They are provably secure.
We can prove the Forward Security of the AKE.
Our preliminary implementations are very efficient.
Our AKE are strong candidates for the post-quantum world.
Password authenticated Key Exchange (PAKE)
Authentication protocol using the signal functions.
Work in Progress - Authentication Protocol

Fig. 1. Authenticated Protocol

Prover (P)
- Sample $s, e \leftarrow \chi_\alpha$
- Secret Key: $s \in R_q$
- Public Key: $p = as + e \in R_q, a$
- Sample $s_1, e_1 \leftarrow \chi_\alpha$
- Compute: $p_1 = as_1 + e_1 \in R_q$
- Sample $g_p \leftarrow \chi_\alpha$
- Compute $k_p = (s_1 + bs)x + g_p$
- $w = \text{Sig}(k_p)$

Verifier (V)
- Set: $x = as' + e' \in R_q$
- Random Challenge bit $b \leftarrow \{-1, 1\}$
- Sample $s', e' \leftarrow \chi_\alpha$
- Sample $g_v \leftarrow \chi_\alpha$
- Compute $k_v = (p_1 + bp)s' + g_v$
- Verify if $w$ match with the value of $k_v$
Thank You

Thank NIST and NSF for support!

Thank you!

You can email your questions or comments to jintai.ding@gmail.com