Post-quantum Key Exchange from LWE Problems

Jintai Ding

University of Cincinnati

08.2019
Introduction: PKC
The future – quantum threat – post-quantum cryptography
Lattice-based Key Exchange

What is this number?

2133562529 1600027351 1427593551 9420913291 4767425698
0668648182 4528580269 7571587504 8271600387 9286718814
4217660057 9559348458 0081495826 8691260056 0376434697
9087161398 8653520618 5442348052 5894942341 3033375605
8732136514 8876038644 3075342912 0129705489 0001670606
7393246389 8375697515 1734774577 2076420507 4793016726
4791679237 3351492517 3209625562 4512058040 6546060184
8036703111 8237059907 4873628794 2617311911 1255520806
0025609009 0478884806 3977173442 6254325175 1228479981
6060960213 2860929278 0435354785 7716957089 8641110787
9876456259 1930871508 8016517131 0668371684 8928958136
1754587749 9229988091 2892709869 7538006934 6521176840
9897604596 0758751
data packages (...) are digitally signed.
The number for Microsoft updates
PKC

- The number for Microsoft updates
- Digital signature based on RSA
Mathematics behind: the hardness of integer factorization

\[ n = pq. \]

\[ 15 = 3 \times 5. \]
Mathematics behind: the hardness of integer factorization

\[ n = pq. \]

\[ 15 = 3 \times 5. \]

The concept behind:

Public key Cryptography

IN (the god of) MATH WE TRUST!
Introduction: PKC
The future – quantum threat – post-quantum cryptography
Lattice-based Key Exchange

PKC

- Diffie-Hellman
- The inventors of the idea of PKC
  - Turing Prize 2016
Introduction: PKC
The future – quantum threat – post-quantum cryptography
Lattice-based Key Exchange

PKC

- Diffie-Hellman – Turing Prize 2016
- The inventors of the idea of PKC
- RSA – 2003 Turing prize
Symmetric system

Traditional symmetric systems

- The sender and receiver have the same keys:
  Ceasar Cipher, Enigma machines, DES, AES (Advanced Encryption Standards)
Symmetric system

Traditional symmetric systems

- The sender and receiver have the same keys: Ceasar Cipher, Enigma machines, DES, AES (Advanced Encryption Standards)

- The two parties must have a prior secure key exchange.
Why PKC

- Large computer networks
Why PKC

- Large computer networks
- Cost of key agreement
The idea was proposed in 1970s by Diffie-Hellmann
The idea was proposed in 1970s by Diffie-Hellmann.

Traditionally the information is symmetric. PKC is asymmetric.
The idea was proposed in 1970s by Diffie-Hellmann.

Traditionally the information is symmetric. PKC is asymmetric.

There are two sets of keys:
one public \((N, r)\): \(\gcd(r, (p - 1)(q - 1)) = 1\)
one private
\((d, p, q)\): \(d \times r = 1 \mod (p - 1)(q - 1)\)
Encryption
The public key is for encryption:

\[ e = m^r \mod N \]

The private key for decryption:

\[ a = e^d \mod N. \]

RSA: n is public and p,q is private.

One knows how to factor n, one can defeat RSA.
One can calculate \( d \) from \((p - 1)(q - 1)\).
Secure Communications

- To establish a shared key,
- the fast communication is done using the shared keys with symmetric systems – AES
- SSL, TLS, Online shopping (sending credit card securely on the internet)
Authentications

- To authenticate a message or a transaction
- The public key is to verify and anyone can verify.
- The private key is used to sign
- Software update, legal document, voting
Bitcoin-Blockchain

- An open ledger
- Decentralized
  Anyone can participate and can verify
  Good privacy and highly efficient in time
- Used as cryptocurrency
  ECDSA, Hash functions and Elliptic curve signature
  Public key (or address: the hash of public key) is to receive bitcoins
  Private key is used to send money, while the transaction can be verified by the public key
  Hash functions for address and POW – for synchronization
PKC and Quantum computer

- Quantum computer: quantum mechanics for computations

R. Feynman
PKC and Quantum computer

- Quantum computer: quantum mechanics for computations
  - R. Feynman

- In 1995, Quantum algorithm for factoring, discrete logarithm.
  - P. Shor
Can quantum computer really work?

Isaac Chuang

15 million dollars to show that

15 = 3 × 5.

The problem of scaling
The context of our work - PQC

- Shor’s quantum algorithm
- Post-quantum cryptography

*Develop public key cryptosystems that could resist future quantum computer attacks*
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

IAD 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
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
Forward Security

- RSA does not offer forward security since compromise of static private key allows decrypting the session keys.
- Possible to achieve forward security with RSA with ephemeral keys but expensive.
- Diffie Hellman offers forward security.

*Forward security: If static keys compromised, previous session keys remain secure.*
Introduction: PKC
The future – quantum threat – post-quantum cryptography
Lattice-based Key Exchange

Diffie-Hellman Key Exchange

\[ g^a \]

\[ g^b \]

\[ (g^b)^a \]

\[ (g^a)^b \]
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, Chebychev 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!!!

- Clifford Cocks – RSA, Malcolm Williamson – DH, 1973

- The forgotten inspiration of J. Ellis – ”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 \( \vec{s} \) from \( A, \vec{b} \).
- Hard to tell if \( \vec{s} \) even exists
- Reduction to lattice approximation problems
Discrete Gaussian
Definition

Let $n$ be a power of 2, $q \equiv 1 \pmod{2^n}$ 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

\[ p_A = as_A + 2e_A \]
\[ p_B = as_B + 2e_B \]

- Public \( a \in R_q \). Acts like generator \( g \) in DH.
Diffie-Hellman from Ideal Lattices

\[ p_A = as_A + 2e_A \]

\[ p_B = as_B + 2e_B \]

\[ k_A = s_Ap_B + 2e'_A \approx \]

\[ = aS_A S_B + 2S_Ae_B + 2e'_A \]

\[ k_B = p_As_B + 2e'_B \approx \]

\[ = aS_A S_B + 2S_Be_A + 2e'_B \]

- Public \( a \in R_q \). 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
Difference 4, both odd.
Wrap-around Illustrated

- But wait! If \( q = 17 \),
  \[ \mathbb{Z}_q = \{-8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8\} \]
- 11 becomes \(-6\), now parities disagree!
Rounding Intuition – Outer Region problem

Additional modular operation
Rounding Intuition

- inner region
- outer region

+ (q-1)/2

swaps the regions

- inner region values moved to outer region

outer values moved to inner region

- outer region

- inner region
Rounding Intuition – Region Division

Signal function $\text{Sig}(\cdot)$
Compensating for Wrap-Around

- $g = 2S_A e_B - 2S_B e_A + 2e'_A - 2e'_B$.
- Recall: $|g^{(j)}| < \frac{q}{8}$
- Define $E = \{-\lfloor \frac{q}{4} \rfloor, \ldots, \lfloor \frac{q}{4} \rfloor \}$. 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 $k_B^{(j)} \notin E$, $k_A^{(j)} + \frac{q-1}{2} \equiv k_B^{(j)} + \frac{q-1}{2}$.
Wrap-around Defeated

Define $\text{Sig}(v) = \begin{cases} 0 & v \in E, \\ 1 & v \notin E. \end{cases}$

and, $w_B^{(j)} = \text{Sig}(k_B^{(j)})$

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 $w_B = (w_B^{(0)}, w_B^{(1)}, \ldots, w_B^{(n-1)})$
- $\sigma_B = k_B + w_B \frac{q-1}{2} \pmod{2}$.
- $\sigma_A = k_A + w_B \frac{q-1}{2} \pmod{2}$. 
Obtaining shared secret from approximate shared secret:

\[
\begin{align*}
k_A &= (k_A^{(0)}, k_A^{(1)}, \ldots, k_A^{(n-1)}) \\
k_B &= (k_B^{(0)}, k_B^{(1)}, \ldots, k_B^{(n-1)}) \\
\tilde{g} &= (g^{(0)}, g^{(1)}, \ldots, g^{(n-1)}) \\
k_A - k_B &= 2\tilde{g} \\
&= k_A \equiv k_B \pmod{2}
\end{align*}
\]
Key Derivation

Obtaining shared secret from approximate shared secret:

\[ k_A = (k^{(0)}_A, k^{(1)}_A, \ldots, k^{(n-1)}_A) \]
\[ k_B = (k^{(0)}_B, k^{(1)}_B, \ldots, k^{(n-1)}_B) \]
\[ \tilde{g} = (g^{(0)}, g^{(1)}, \ldots, g^{(n-1)}) \]
\[ k_A - k_B = 2\tilde{g} \]
\[ k_A \equiv k_B \pmod{2} \]

- Each \( k^{(j)}_A = k^{(j)}_B + 2g^{(j)} \).
- 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
Ding’s paper:
Cryptology ePrint Archive: Report 2012/688
20121210:115748 (posted 10-Dec-2012 11:57:48 UTC)

Cryptology ePrint Archive: Report 2014/070, C. Peikert
Cryptology ePrint Archive: Report 2014/599, Joppe W. Bos
and Craig Costello and Michael Naehrig and Douglas Stebila
Cryptology ePrint Archive: Report 2015/1092, Erdem Alkim
and Leo Ducas and Thomas Poppelmann and Peter Schwabe
Comparison of Signal

Signal function $\text{Sig}(\cdot)$

Cross rounding $b = \langle \cdot, \cdot \rangle_2$
Proof Games

Proof proceeds by series of games:

- Begin with simulated protocol
- Adversary cannot distinguish from previous game
- Eventually, if original protocol can be distinguished from random, rLWE can be broken
- Important step to post quantum key exchange.
- Authenticated key exchange in Eurocrypt 2015.
Thank You

Any questions?

Thank Air Force and NSF for support.

Questions to jintai.ding@gmail.com