Ding Key Exchange

Jintai Ding[1] Tsuyoshi Takagi[2] Xinwei Gao[3] Yuntao Wang[4]

1. University of Cincinnati, jintai.ding@gmail.com
2. University of Tokyo, takagi@mist.i.u-tokyo.ac.jp
3. Beijing Jiaotong University, xinwei.gao.7@yandex.com
4. Kyushu University, y-wang@math.kyushu-u.ac.jp

2018-04-12

Outline

Summary

Ding Key Exchange

An ephemeral Diffie-Hellman-like key exchange from RLWE problem

- Post-quantum key exchange protocol
  - Ephemeral-only Diffie-Hellman-like (forward secure), not KEM
  - Only one RLWE sample
  - Reduced communication cost
  - Parameter sets targeting AES-128/192/256 security
  - Drop-in replacement
  - Simple and elegant design

LWE & Ring-LWE-based Key Exchange Protocols

Key Exchange

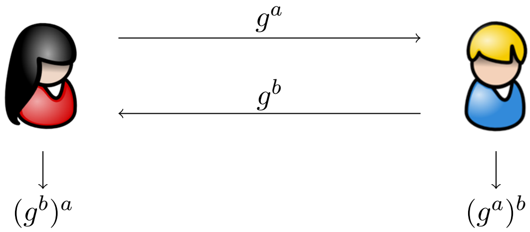
- Pre-2012: Various LWE & RLWE encryption (KEM) schemes with large ciphertext size. Framework of DH-like key exchange construction appeared. No concrete error reconciliation mechanism
- 2012: Ding et al. invented the first complete LWE & RLWE-based Diffie-Hellman-like key exchange protocols (DING12)
- 2014: Peikert tweaked DING12 reconciliation slightly
- 2015: Bos et al. implemented PKT14 (BCNS)
- 2016: Alkim et al. improved BCNS (NewHope)
- ...

LWE & Ring-LWE-based Key Exchange Protocols

Attacks (Key Reuse)

- 2015: NSA revealed key reuse issues for post-quantum encryption and key agreement
- 2016: Fluhrer proposed attack framework on Diffie-Hellman-like reconciliation-based key exchange
- 2016-2018: Ding et al. extended Fluhrer's attack in multiple works and proposed countermeasure

Diffie-Hellman Key Exchange



Generalizing DH

- DH works because maps $f(x) = x^a$ and $h(x) = x^b$ commute

$$f \circ h = h \circ f,$$

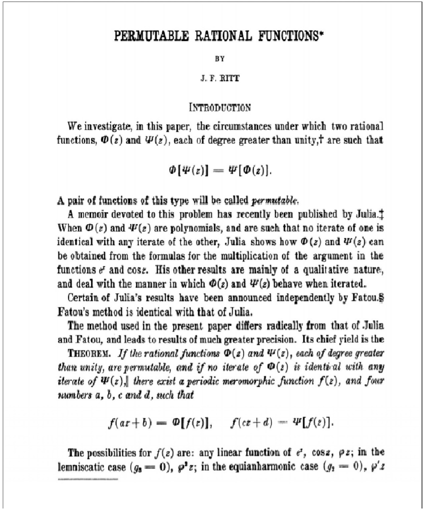
  $\circ$ – composition

  Nonlinearity

- Many attempts – Braid group etc.
- J. Ritt (1923) – Power polynomials, Chebychev polynomials and elliptic curve
- No direct post-quantum variant

Diffie-Hellman Key Exchange

Figure 1: J. Ritt



PERMUTABLE RATIONAL FUNCTIONS*

BY

J. F. RITT

INTRODUCTION

We investigate, in this paper, the circumstances under which two rational functions, $\Phi(z)$ and $\Psi(z)$, each of degree greater than unity,† are such that

$$\Phi[\Psi(z)] = \Psi[\Phi(z)].$$

A pair of functions of this type will be called *permutable*.

A memoir devoted to this problem has recently been published by Julia.‡ When $\Phi(z)$ and $\Psi(z)$ are polynomials, and are such that no iterate of one is identical with any iterate of the other, Julia shows how $\Phi(z)$ and $\Psi(z)$ can be obtained from the formulas for the multiplication of the argument in the functions e^z and $\cos z$. His other results are mainly of a qualitative nature, and deal with the manner in which $\Phi(z)$ and $\Psi(z)$ behave when iterated.

Certain of Julia's results have been announced independently by Fatou.§ Fatou's method is identical with that of Julia.

The method used in the present paper differs radically from that of Julia and Fatou, and leads to results of much greater precision. Its chief yield is the

THEOREM. *If the rational functions $\Phi(z)$ and $\Psi(z)$, each of degree greater than unity, are permutable, and if no iterate of $\Phi(z)$ is identical with any iterate of $\Psi(z)$,‖ there exist a periodic meromorphic function $f(z)$, and four numbers a, b, c and d, such that*

$$f(az+b) = \Phi[f(z)], \quad f(cz+d) = \Psi[f(z)].$$

The possibilities for $f(z)$ are: any linear function of e^z, $\cos z$, $\wp z$; in the lemniscatic case ($g_3 = 0$), $\wp^2 z$; in the equianharmonic case ($g_2 = 0$), $\wp' z$

Figure 2: 1923

Basic Ideas

- A.B.C. three matrices:

  $$(A \times B) \times C = A \times (B \times C)$$

- The idea of LWE:

  Adding errors in the process.

Approximate Diffie-Hellman from RLWE

$$\xrightarrow{p_A = as_A + 2e_A}$$

$$\xleftarrow{p_B = as_B + 2e_B,\ \boxed{???}}$$

$$k_A = s_A p_B = as_A s_B + 2s_A e_B \quad \approx \quad k_B = p_A s_B = as_A s_B + 2s_B e_A$$

- Public $a \in R_q$ uniformly random. Error e is small
- k_A only *approximately* equals to k_B
- Difference is even – same low bits $\rightarrow$ mod 2 simultaneously, but not that simple
- Need to send additional small information – We call it “Signal”

Approximate Diffie-Hellman from RLWE

$$\xrightarrow{\quad p_A = as_A + 2e_A \quad}$$

$$\xleftarrow{\quad p_B = as_B + 2e_B, \boxed{???} \quad}$$

$$\downarrow \qquad\qquad\qquad\qquad \downarrow$$

$$k_A = s_A p_B = as_A s_B + 2s_A e_B \quad \approx \quad k_B = p_A s_B = as_A s_B + 2s_B e_A$$

- Need to send additional small information – We call it "Signal"

Approximate Diffie-Hellman Key Exchange from RLWE

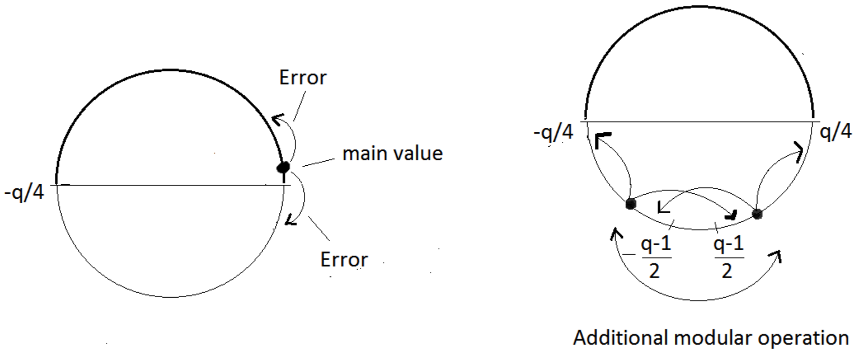


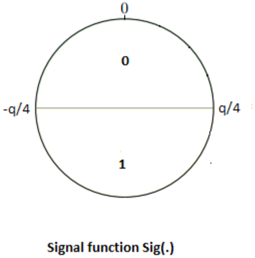
Figure 3: Mismatch

Protocol Construction

Party i		Party j
$seed \xleftarrow{\$} \{0,1\}^{128}$ $a = \text{Derive\_a}() \in R_q$ Public key: $p_i = a \cdot s_i + 2e_i \in R_q$ Private key: $s_i \in R_q$ where $s_i, e_i \xleftarrow{\$} D_{\mathbb{Z}^n,\sigma}$ $p'_i = \text{Round}(p_i, p, q)$	$\xrightarrow{p'_i,\, seed}$	$a = \text{Derive\_a}() \in R_q$ Public key: $p_j = a \cdot s_j + 2e_j \in R_q$ Private key: $s_j \in R_q$ where $s_j, e_j \xleftarrow{\$} D_{\mathbb{Z}^n,\sigma}$ $p'_j = \text{Round}(p_j, p, q)$
$p''_j = \text{Recover}(p'_j, p, q) \in R_q$ $k_i = p''_j \cdot s_i \in R_q$ $sk_i = \text{Mod}_2(k_i, w_j) \in \{0,1\}^n$	$\xleftarrow{p'_j,\, w_j}$	$p''_i = \text{Recover}(p'_i, p, q) \in R_q$ $k_j = p''_i \cdot s_j \in R_q$ $w_j = \text{Sig}(k_j) \in \{0,1\}^n$ $sk_j = \text{Mod}_2(k_j, w_j) \in \{0,1\}^n$

Figure 4: Ding Key Exchange

Protocol Construction



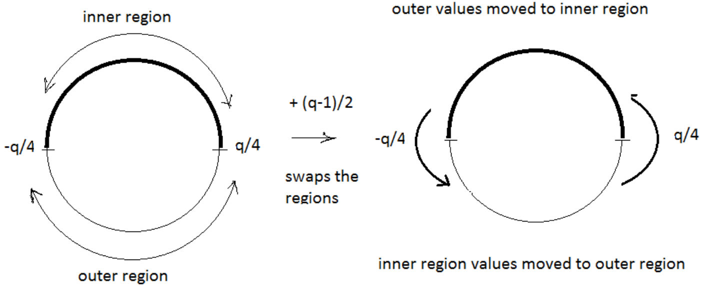


Figure 5: Rounding

Protocol Construction

Hint Function $\sigma_0(x), \sigma_1(x)$

Hint functions $\sigma_0(x)$, $\sigma_1(x)$ from $\mathbb{Z}_q$ to $\{0,1\}$ are defined as:

$$\sigma_0(x) = \begin{cases} 0, x \in [-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor] \\ 1, otherwise \end{cases}, \ \sigma_1(x) = \begin{cases} 0, x \in [-\lfloor \frac{q}{4} \rfloor + 1, \lfloor \frac{q}{4} \rfloor + 1] \\ 1, otherwise \end{cases}$$

Signal Function Sig()

For any $y \in \mathbb{Z}_q$, $\mathrm{Sig}(y) = \sigma_b(y)$, where $b \xleftarrow{\$} \{0,1\}$. If $\mathrm{Sig}(y) = 1$, we say y is in the outer region, otherwise y is in the inner region.

Reconciliation Function $\mathrm{Mod}_2()$

$\mathrm{Mod}_2()$ is a deterministic function with error tolerance $\delta = \frac{q}{4} - 2$. For any x in $\mathbb{Z}_q$ and $w = \mathrm{Sig}(x)$, $\mathrm{Mod}_2(x, w) = (x + w \cdot \frac{q-1}{2} \mod q) \mod 2$.

Protocol Construction

Rounding Function Round()

- Reduce communication cost using rounding technique.
- **Round public key $as + 2e$ to drop least significant bits.**

Recovering Function Recover()

- Recover rounded public key to R_q.
- Error term $2e'$ now contains random and deterministic “errors”.

Correctness

- $\|k_i - k_j\|_\infty \leq \frac{q}{4} - 2$.
- Generate n-bit final shared key.

Parameter Choices

Table 1: Parameter Choices

n	σ	q	p	Claimed Security Level	NIST Security Category	Failure Probability
512	4.19	120833	7551	AES-128	I	2^{-60}
1024	2.6	120833	7551	AES-192 AES-256	III V	2^{-60}

Communication Cost

Table 2: Communication Cost

n	Party $i \to j$ (Byte)	Party $j \to i$ (Byte)	Total (Byte)	Claimed Security Level	NIST Security Category
512	848	896	1744	AES-128	I
1024	1680	1792	3472	AES-192 AES-256	III V

Passive Security

- Notion: Adversary cannot distinguish transcripts of the protocol from uniform random
- Submitted as KEM $\rightarrow$ IND-CPA claimed
- No key reuse

Advantages

- Ephemeral key exchange – One RLWE sample and forward secure
- Reduced communication cost
- DH-like key exchange vs KEM
- Longer final shared key
- Flexible parameter choices
- Simple and elegant design

Limitations

- Larger communication cost compared with current public key cryptosystems
- ...

Applications

- Drop-in replacement for protocols/applications that use DH(E)/ECDH(E) etc.
- TLS, SSH, IPsec, VPN
- End-to-end applications (secure messaging, audio/video calling etc.)
- Client-server applications
- ...

Two Estimators Used in Our Cryptic Analysis

1. Progressive BKZ (pBKZ) Simulator [Aono et al., 2016]: Four relevant parameters:

- blocksize β
- GSA constant r
- ENUM search radius coefficient α
- ENUM search success probability p

Input: basis B, the target β (or target r).
Output: optimal runtime t_{pBKZ} of pBKZ while the reduced basis achieves target r.

2. BKZ with Sieve [Albrecht et al., 2017]:
Input: dimension of a basis B, the blocksize β.
Output: asymptotic runtime $t_{BKZ-Sieve}$ to get BKZ-β reduced basis.

$$t_{BKZ-Sieve} = 8 \cdot n \cdot 2^{0.292\beta+16.4} (\text{Flops})$$

Rescaling

Let $z = \text{Recover}(\text{Round}(a \cdot s + 2e, p, q), p, q) = as + 2e + d = as + 2f \in R_q$, where $s, e \xleftarrow{\$} D_{\mathbb{Z}^n,\sigma}$ and $2f = 2e + d$.
The attack on the protocol is given z and a, output private key s.
This problem is equivalent to:

$$
\begin{aligned}
& z = a \cdot s + 2f \mod q \\
\Leftrightarrow\ & 2^{-1}z = 2^{-1}a \cdot s + f \mod q \\
\Leftrightarrow\ & z'' = a'' \cdot s + f \mod q
\end{aligned}
$$

Standard deviation of term f is denoted as σ_f. Note that f no longer follows discrete Gaussian distribution.

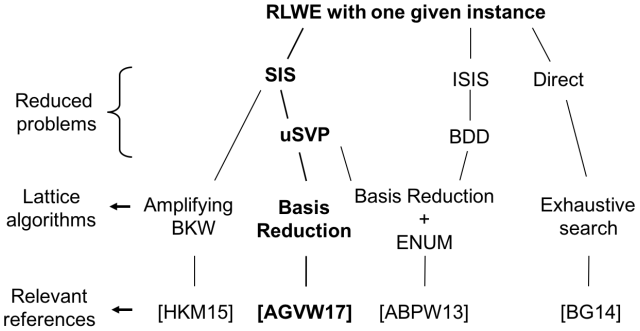
Number of Samples

Our security analysis is based on the fact:

ONLY ONE RLWE sample $(a, b = a \cdot s + e \mod q) \in (R_q, R_q)$ is given.

Some other security analysis are actually based on more samples.

Attack Choice



Possible attacks on search RLWE problem with only one given instance. Relevant references [HKM15], [AGVW17], [ABPW13] and [BG14] are [Herold et al., 2015], [Albrecht et al., 2017], [Aono et al., 2013] and [Bai and Galbraith, 2014] in reference respectively.

“2016 estimation”

The “2016 estimation” in [Albrecht et al., 2017] states that if the Gaussian Heuristic and GSA hold for BKZ-β reduced basis and

$$\sqrt{\beta/d} \cdot \|(\mathbf{e}|1)\|_2 \approx \sqrt{\beta}\sigma \leq \delta^{2\beta-d} \cdot \mathrm{Vol}(L_{(\mathbf{A},q)})^{1/d}. \tag{1}$$

then error e can be found by BKZ-β with root Hermite Factor δ.
Equation (1) originates from NewHope [Alkim et al., 2016] and was corrected in [Albrecht et al., 2017].

Our Simulator for Parameter Choice

Input: dimension n and modulus q in RLWE(n, q, σ_f) case from Ding Key Exchange.
Output: lower bound of σ_f required in Ding Key Exchange.

Step 1. A short vector $\|\mathbf{b}_1\| = \delta^d \cdot \det(\mathbf{B})^{1/d}$ is assumed to be inside of the BKZ-β reduced basis **B** of dimension d [Chen, 2013], where the rHF is

$$\delta = (((\pi\beta)^{1/\beta}\beta)/(2\pi e))^{1/(2(\beta-1))}. \tag{2}$$

We pre-compute the expected δ for $\beta = 10, \cdots, n$ and rewrite equation (1) as

$$\sqrt{\beta \cdot ({\sigma_e}^2 + {\sigma_f}^2)} \leq \delta^{2\beta-2n-1} \cdot q^{n/(2n+1)}. \tag{3}$$

In our case, $d = 2n + 1$ and $\mathrm{Vol}(L_{(\mathbf{A},q)}) = q^n$.

Our Simulator for Parameter Choice

Input: dimension n and modulus q in RLWE(n, q, σ_f) case from Ding Key Exchange.
Output: lower bound of σ_f required in Ding Key Exchange.

Step 2. for β from 10 to n, input (n, β), compute T_{BKZ} (t_{pBKZ} and $t_{BKZ-Sieve}$) from two BKZ runtime estimators respectively.

$$\begin{aligned} \textbf{(practical) bit operations} \text{ of } \mathrm{RLWE}(n, q, \sigma_f) = & \quad \log_2(t_{pBKZ} \times 2.7 \times 10^9 \times 64). \\ & \text{and} \\ \textbf{(lower bound) bit operations} \text{ of } \mathrm{RLWE}(n, q, \sigma_f) = & \quad \log_2(t_{BKZ-sieve} \times 64) \end{aligned} \tag{4}$$

$$\log_2(t_{pBKZ}(secs)) = \begin{cases} 0.003924 \cdot \beta^2 - 0.568 \cdot \beta + 41.93 & (n = 512) \\ 0.004212 \cdot \beta^2 - 0.6886 \cdot \beta + 55.49 & (n = 1024) \end{cases} \tag{5}$$

Combine with Step 1, we can get the lower bound of σ_f in RLWE(n, q, σ_f) which covers security of AES-128/192/256 using equations (4), (2) and (3).

Parameter Choice for Ding Key Exchange Protocol

Table 3: Our simulation data and parameter settings covering security of AES-128/192/256

Security level (n, q, σ)	AES-128 (512,120833,4.19)		AES-192 and AES-256 (1024,120833,2.6)	
Method	pBKZ	2016 estimation	pBKZ	2016 estimation
Logarithmic computational complexity	319.14	142.27	1473.09	279.05
Blocksize	330	366	660	831
GSA Const.	0.983		0.991	
σ (for s and e) of our parameter choice	4.19		2.6	
σ_f	4.92		4.72	
bits security	145.59		282.37	

Conclusion

- Ding Key Exchange – An ephemeral-only Diffie-Hellman-like RLWE + Rounding key exchange
- Reduced communication cost, flexible parameter choices covering security of AES-128/192/256 and forward secure
- Drop-in replacement of Diffie-Hellman key exchange and variants

Thanks for your attention!

Q & A