A Simple Provably Secure (Authenticated) Key Exchange Scheme Based on the Learning with Errors Problem

Jintai Ding

April. 3, 2015

Joint work with O. Dagdelen, X. Lin, X. Xie, J. Zhang, Z. Zhang

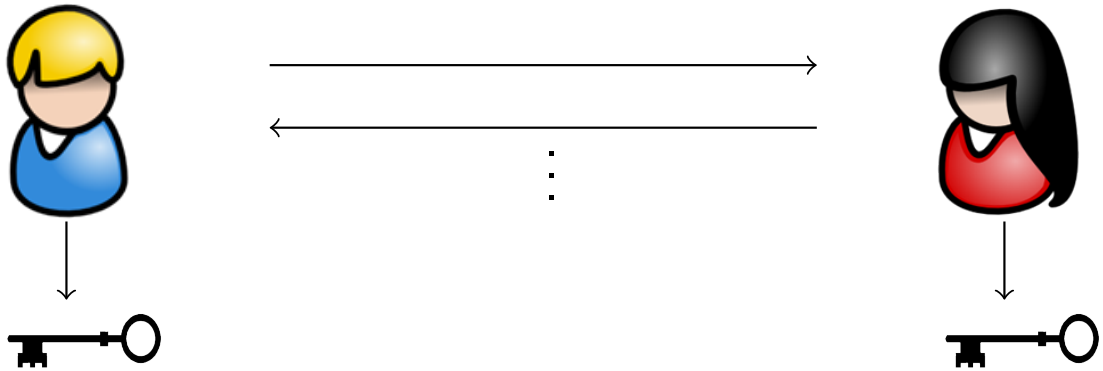
Key Transport from Encryption versus Key Exchange?

- Alice Uses Bob's public key to encrypt a random string and sends the ciphertext to Bob. Bob decrypts it and get the random string.
- In practice, public key encryption is only used to transmit random keys. (The key is only determined by one party)
- Using PKE can not guarantee forward security.
  - If the attacker gets the secret key, then he will learn every communication made before.

What's Key Exchange

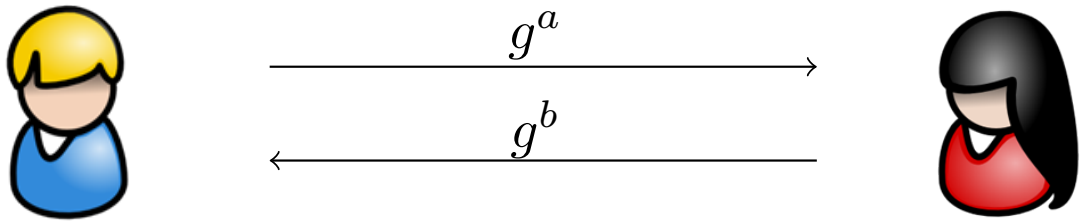
What's Key Exchange

What's Key Exchange

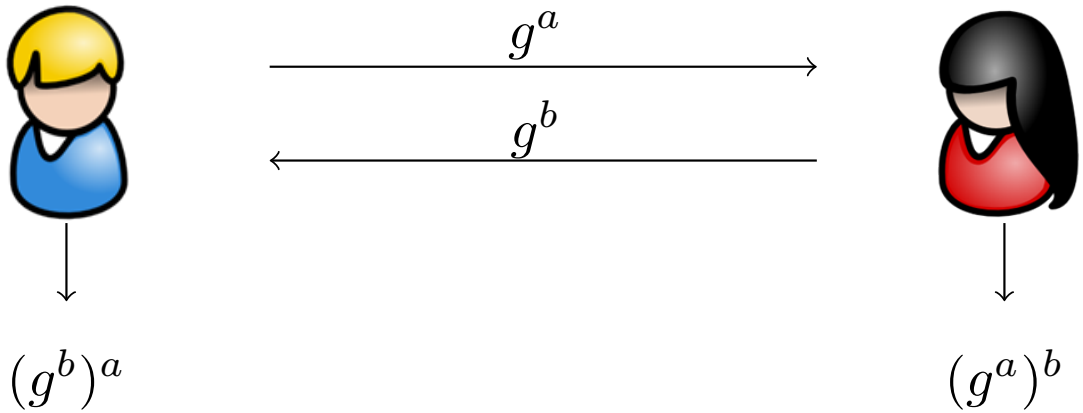


- Get a shared secret key in an insecure channel.

The Elegant Diffie-Hellman Protocol



The Elegant Diffie-Hellman Protocol



- Using $g^{ab} = (g^b)^a = (g^a)^b$.

Mathemtical structure behind

Motivation:

- Can we get a DH analogy from other mathematical tools?
- The case of Diffie-Hellmann:

$$(g^a)^b = (g^b)^a = g^{ab}$$

The commutativity of nonlinear operators.

Mathemtical structure behind

Motivation:

- Can we get a DH analogy from other mathematical tools?
- The case of Diffie-Hellmann:

$$(g^a)^b = (g^b)^a = g^{ab}$$

The commutativity of nonlinear operators.

Other similar attempts?

- Can we get a DH analogy from other mathematical structures?

  Many failed attempts to build new DH like protocols.
- Braid group and other finite groups
- Other nonlinear maps?
  MR1501252, Ritt, J. F. Permutable rational functions. Trans. Amer. Math. Soc. 25 (1923), no. 3, 399-448. 30D05

Mathemtical structure behind

Motivation: Linear case?

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

- To make it secure, we need to add "errors".
- We need to be able to remove "errors".

Motivation and Results

Motivation:

- Can we get a DH analogy from other mathematical tools?
- Can we get KE from lattices (say, LWE, which is apparently resistant to quantum attacks)?
- If so, we will get better efficiency and better security guarantees.

Motivation and Results

Motivation:

- Can we get a DH analogy from other mathematical tools?
- Can we get KE from lattices (say, LWE, which is apparently resistant to quantum attacks)?
- If so, we will get better efficiency and better security guarantees.

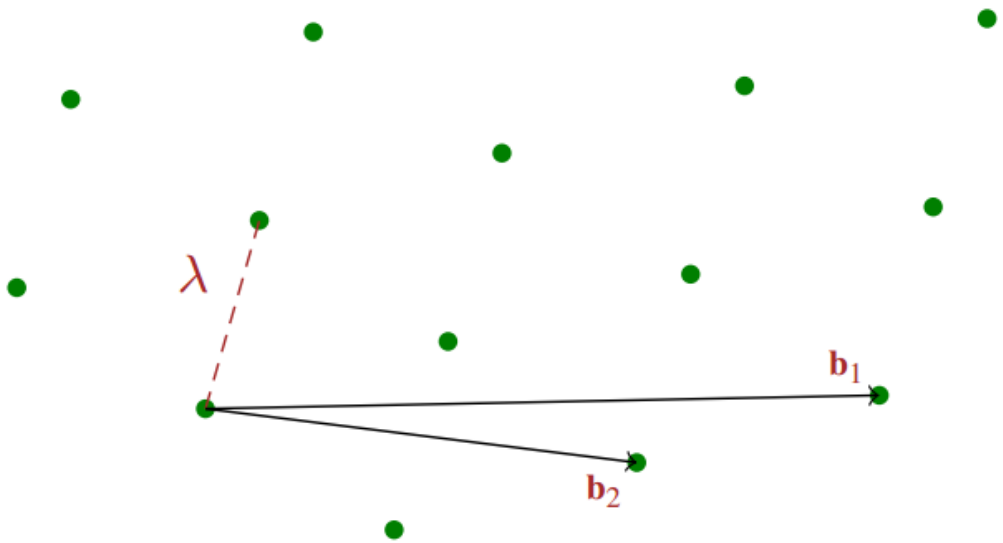
Our Results:

- An Efficient (2-round) key exchange protocol from LWE and RLWE.
- A new way to deal with approximate key exhange
- Extend to multi-party key exchange (without security proof).

Lattices

Given m linear independent vectors $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_m] \in \mathbb{R}^{n \times m}$. A lattice $\mathcal{L}(\mathbf{B})$ consists of the integer combinations of $\mathbf{b}_i$'s.

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



Hard Problem

γ-SVP(Shortest Vector Problem)

Given a n-dimensional lattice $\mathcal{L}(\mathbf{B})$, find a non-zero lattice vector $\mathbf{v}$, such that $\|\mathbf{v}\| \leq \gamma \cdot \lambda$.

Hard Problem

γ-SVP(Shortest Vector Problem)

Given a n-dimensional lattice $\mathcal{L}(\mathbf{B})$, find a non-zero lattice vector $\mathbf{v}$, such that $\|\mathbf{v}\| \leq \gamma \cdot \lambda$.

- $\gamma = 1$; just the SVP problem.
- γ is constant (independent of n); γ-SVP is NP-hard.
- $\gamma \geq 2^n$; γ-SVP can be solved in polynomial time (LLL algorithm).
- $\gamma = \text{poly}(n)$; probably not NP-hard, but we do not have polynomial time algorithms (This is what we use in cryptography).

Learning with Errors (LWE) [Oded Regev 2005]

Goal: distinguishing "noisy inner products" from uniform.

$\mathbf{a}_1 \leftarrow \mathbb{Z}_q^n;$	$b_1 = \langle \mathbf{a}_1, \mathbf{s} \rangle + e_1 \mod q$
$\mathbf{a}_2 \leftarrow \mathbb{Z}_q^n;$	$b_2 = \langle \mathbf{a}_2, \mathbf{s} \rangle + e_2 \mod q$
$\vdots$	
$\mathbf{a}_m \leftarrow \mathbb{Z}_q^n;$	$b_m = \langle \mathbf{a}_m, \mathbf{s} \rangle + e_m \mod q$

$\mathbf{a}_1 \leftarrow \mathbb{Z}_q^n;$	$b_1 \leftarrow \mathbb{Z}_q$
$\mathbf{a}_2 \leftarrow \mathbb{Z}_q^n;$	$b_2 \leftarrow \mathbb{Z}_q$
$\vdots$	
$\mathbf{a}_m \leftarrow \mathbb{Z}_q^n;$	$b_m \leftarrow \mathbb{Z}_q$

In a matrix form

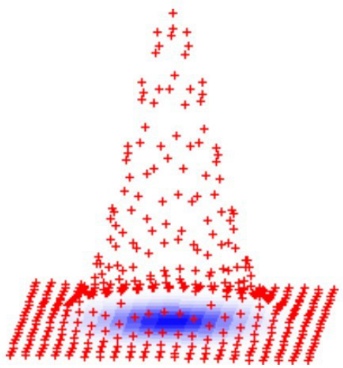
$$(\mathbf{A}, \mathbf{As} + \mathbf{e}) \approx_c (\mathbf{A}, \mathbf{b})$$

Where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $m = \text{poly}(n)$, $q = \text{poly}(n)$ and $e_i \leftarrow \chi$ is some distribution in $\mathbb{Z}$. e_i has small size, much smaller than q.

Theorem (Informal)[Reg'05]

Let χ be a discrete Gaussian distribution with parameter $0 < \alpha < 1$, s.t. $\alpha q \geq 2\sqrt{n}$. If there exists a polynomial time algorithm solves LWE problem, then there exists a quantum algorithm solves (n/α)-SVP problems for all n-dimension lattices.

- $\mathbf{s} \leftarrow \chi^n$ is as hard as standard LWE ($\mathbf{s} \leftarrow \mathbb{Z}_q^n$) [ACPS'09].



Notations

- We always consider $\mathbb{Z}_q$ for prime q, and $\mathbb{Z}_q = [-\frac{q-1}{2}, \frac{q-1}{2}]$.
- We always consider the LWE problem with $\mathbf{s} \leftarrow \chi$, i.e. $\mathbf{s}$ is much smaller than q.

Our Protocol (basic idea)

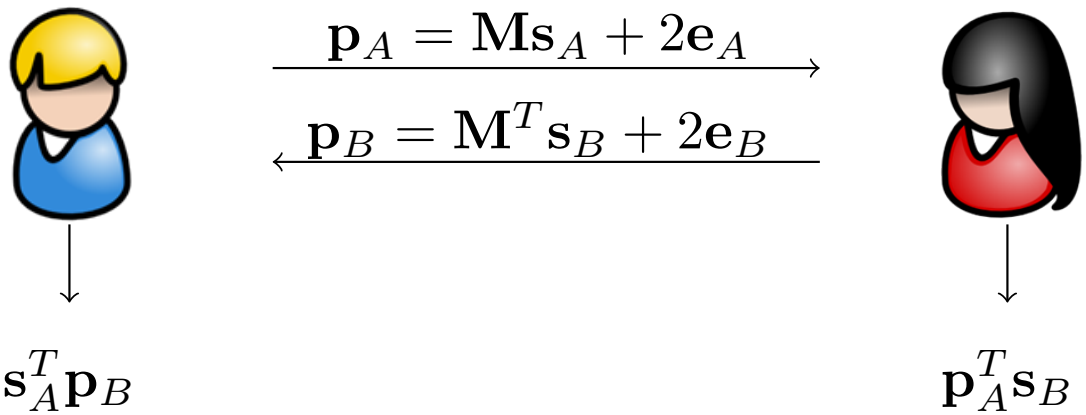
Public Parameter: $\mathbf{M} \leftarrow \mathbb{Z}_q^{n \times n}$

$$\xrightarrow{\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A}$$

$$\xleftarrow{\mathbf{p}_B = \mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_B}$$

Our Protocol (basic idea)

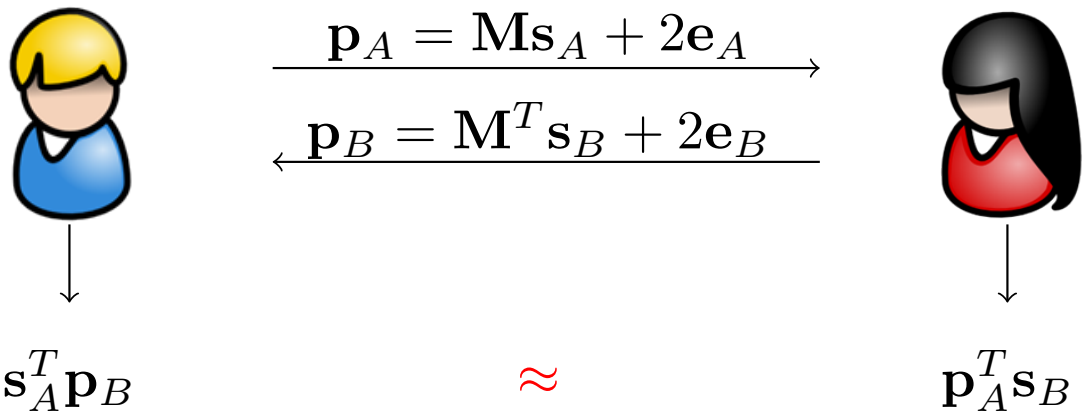
Public Parameter: $\mathbf{M} \leftarrow \mathbb{Z}_q^{n \times n}$



- $\mathbf{s}_A^T\mathbf{p}_B = \mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + 2\mathbf{s}_A^T\mathbf{e}_B \approx \mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_A^T\mathbf{s}_B = \mathbf{p}_A^T\mathbf{s}_B.$
  - note that $\mathbf{s}_A, \mathbf{s}_B, \mathbf{e}_A, \mathbf{e}_B$ are "small".
  - the difference between $\mathbf{s}_A^T\mathbf{p}_B$ and $\mathbf{p}_A^T\mathbf{s}_B$ is *even*

Our Protocol (basic idea)

Public Parameter: $\mathbf{M} \leftarrow \mathbb{Z}_q^{n \times n}$



- $\mathbf{s}_A^T\mathbf{p}_B = \mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + 2\mathbf{s}_A^T\mathbf{e}_B \approx \mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_A^T\mathbf{s}_B = \mathbf{p}_A^T\mathbf{s}_B.$
  - note that $\mathbf{s}_A, \mathbf{s}_B, \mathbf{e}_A, \mathbf{e}_B$ are "small".
  - the difference between $\mathbf{s}_A^T\mathbf{p}_B$ and $\mathbf{p}_A^T\mathbf{s}_B$ is *even*

Robust Extractors

Intuitively, a robust extractor enables to two parties to extract identical information from two close elements with some additional hint.

Definition (Robust Extractors)

An algorithm E is a robust extractor on $\mathbb{Z}_q$ with error tolerance δ with respect to a hint algorithm S, if the following holds:

- The deterministic algorithm E: for $x \in \mathbb{Z}_q$ and $\sigma \in \{0,1\}$, output $k = E(x,\sigma) \in \{0,1\}$.
- The hint algorithm S: for $y \in \mathbb{Z}_q$, output $\sigma \leftarrow S(y) \in \{0,1\}$.
- For any $x, y \in \mathbb{Z}_q$ such that $x - y$ is *even* and $|x - y| \leq \delta$, then $E(x,\sigma) = E(y,\sigma)$, where $\sigma \leftarrow S(y)$.
- If $y \xleftarrow{\$} \mathbb{Z}_q$ and $\sigma \leftarrow S(y)$, then $E(y,\sigma)$ is uniform conditioned on σ.

Note that the errors of x, y in the definition can be set to be multiple of t, where t is a small integer.

Our Robust Extractor

We first define two functions: for $q > 2$ is prime

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

The hint algorithm $S(y)$: $b \xleftarrow{\$} \{0,1\}, S(y) = \sigma_b(y)$.

The robust extractor $E(x, \sigma)$:

$$E(x, \sigma) = \left(x + \sigma \cdot \frac{q-1}{2} \bmod q \right) \bmod 2$$

Lemma

Let $q > 8$ be an odd integer, E is a robust extractor with respect to S with error tolerance $\frac{q}{4} - 2$.

Lemma

Let $q > 8$ be an odd integer, E is a robust extractor with respect to S with error tolerance $\frac{q}{4} - 2$.

For any $x, y \in \mathbb{Z}_q$, and $x - y \triangleq 2\varepsilon$, with $|2\varepsilon| \leq \frac{q}{4} - 2$.
Let $\sigma \leftarrow S(y)$, we have

$$|y + \sigma \cdot \frac{q-1}{2} \bmod q| \leq \frac{q}{4} + 1.$$

Therefore,

$$x+\sigma\cdot\frac{q-1}{2} \bmod q = y+\sigma\cdot\frac{q-1}{2}+2\varepsilon \bmod q = (y+\sigma\cdot\frac{q-1}{2}) \bmod q+2\varepsilon,$$

this implies

$$\begin{aligned} E(x,\sigma) &= x + \sigma \cdot \frac{q-1}{2} \bmod q \mod 2 \\ &= y + \sigma \cdot \frac{q-1}{2} \bmod q \mod 2 = E(y,\sigma) \end{aligned}$$

Removing the Approximation

Public Parameter: $\mathbf{M} \leftarrow \mathbb{Z}_q^{n \times n}$

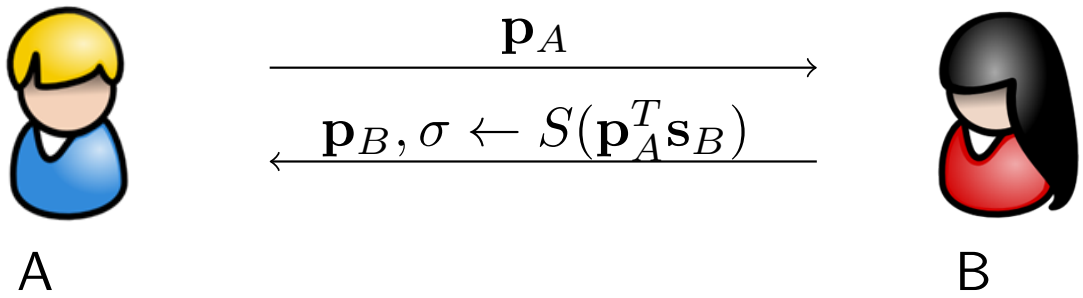
$$\xrightarrow{\mathbf{p}_A}$$

$$\xleftarrow{\mathbf{p}_B, \sigma \leftarrow S(\mathbf{p}_A^T \mathbf{s}_B)}$$

A B

Removing the Approximation

Public Parameter: $\mathbf{M} \leftarrow \mathbb{Z}_q^{n \times n}$



- A outputs $E(\mathbf{s}_A^T \mathbf{p}_B, \sigma)$
- B outputs $E(\mathbf{p}_A^T \mathbf{s}_B, \sigma)$

Correctness

- A has: $\mathbf{s}_A$ and $\sigma \leftarrow S(\mathbf{p}_A^T \mathbf{s}_B)$; B has: $\mathbf{s}_B$.
- Let $K_A = \mathbf{s}_A^T \mathbf{p}_B$ and $K_B = \mathbf{p}_A^T \mathbf{s}_B$.

Correctness

- A has: $\mathbf{s}_A$ and $\sigma \leftarrow S(\mathbf{p}_A^T \mathbf{s}_B)$; B has: $\mathbf{s}_B$.
- Let $K_A = \mathbf{s}_A^T \mathbf{p}_B$ and $K_B = \mathbf{p}_A^T \mathbf{s}_B$.

$$K_A - K_B = 2(\mathbf{s}_A^T \mathbf{e}_B - \mathbf{e}_A^T \mathbf{s}_B)$$

Correctness

- A has: $\mathbf{s}_A$ and $\sigma \leftarrow S(\mathbf{p}_A^T \mathbf{s}_B)$; B has: $\mathbf{s}_B$.
- Let $K_A = \mathbf{s}_A^T \mathbf{p}_B$ and $K_B = \mathbf{p}_A^T \mathbf{s}_B$.

$$K_A - K_B = 2(\mathbf{s}_A^T \mathbf{e}_B - \mathbf{e}_A^T \mathbf{s}_B)$$

If $|2(\mathbf{s}_A^T \mathbf{e}_B - \mathbf{e}_A^T \mathbf{s}_B)| \leq \frac{q}{4} - 2$, then we have

$$E(K_A, \sigma) = E(K_B, \sigma)$$

Correctness

- A has: $\mathbf{s}_A$ and $\sigma \leftarrow S(\mathbf{p}_A^T \mathbf{s}_B)$; B has: $\mathbf{s}_B$.
- Let $K_A = \mathbf{s}_A^T \mathbf{p}_B$ and $K_B = \mathbf{p}_A^T \mathbf{s}_B$.

$$K_A - K_B = 2(\mathbf{s}_A^T \mathbf{e}_B - \mathbf{e}_A^T \mathbf{s}_B)$$

If $|2(\mathbf{s}_A^T \mathbf{e}_B - \mathbf{e}_A^T \mathbf{s}_B)| \leq \frac{q}{4} - 2$, then we have

$$E(K_A, \sigma) = E(K_B, \sigma)$$

It is easy to check that the shared key is

$$\mathbf{s}_A^T \mathbf{M}^T \mathbf{s}_B + \sigma \cdot \frac{q-1}{2} \bmod q \mod 2.$$

Security

- We slightly change the protocol to prove the passive security based on LWE.
  - We set $K_A = \mathbf{s}_A^T \mathbf{p}_B + 2e_A \bmod q$ and $K_B = \mathbf{p}_A^T \cdot \mathbf{s}_B + 2e_B \bmod q$.
- The proof is given from a series of hybrid experiments.
- Note that $(\mathbf{A}, \mathbf{As} + \mathbf{2e} \bmod q) \approx_c (\mathbf{A}, \mathbf{b})$ for odd q.

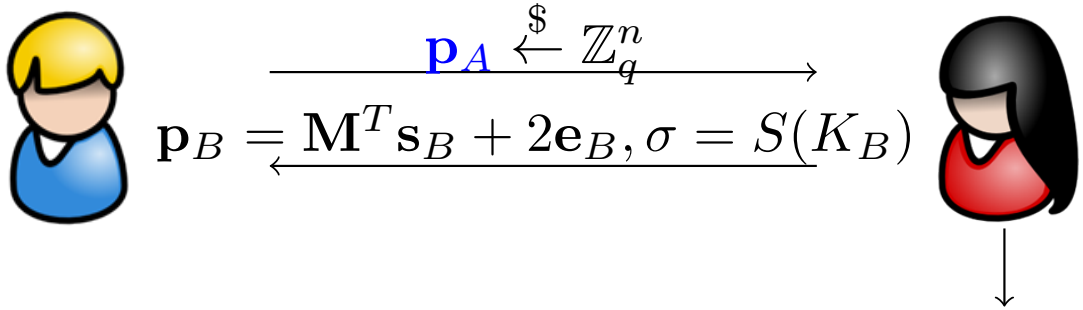
Proof Intuition

$$\xrightarrow{\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A}$$

$$\mathbf{p}_B = \mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_B, \sigma = S(K_B) \xleftarrow{}$$

$$K_B = \mathbf{p}_A^T\mathbf{s}_B + 2e_B$$

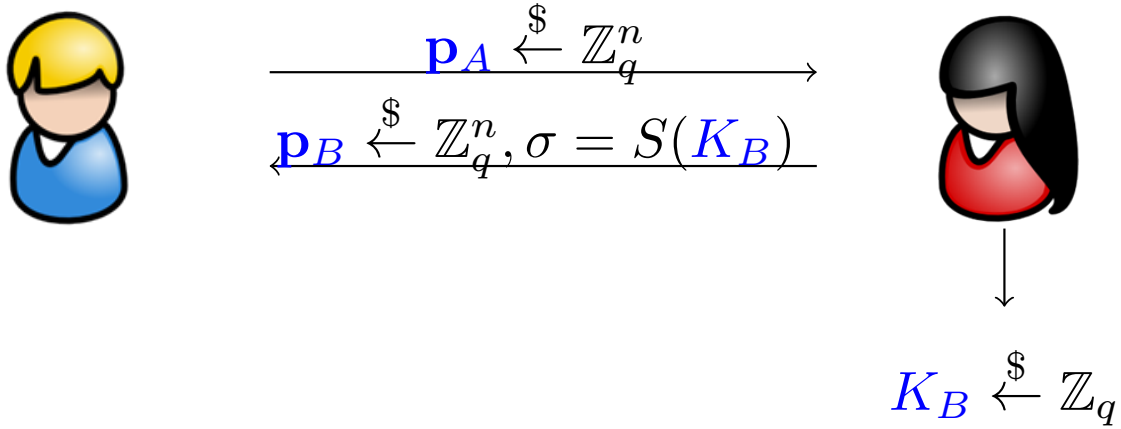
Proof Intuition



$$K_B = \mathbf{p}_A^T\mathbf{s}_B + 2e_B$$

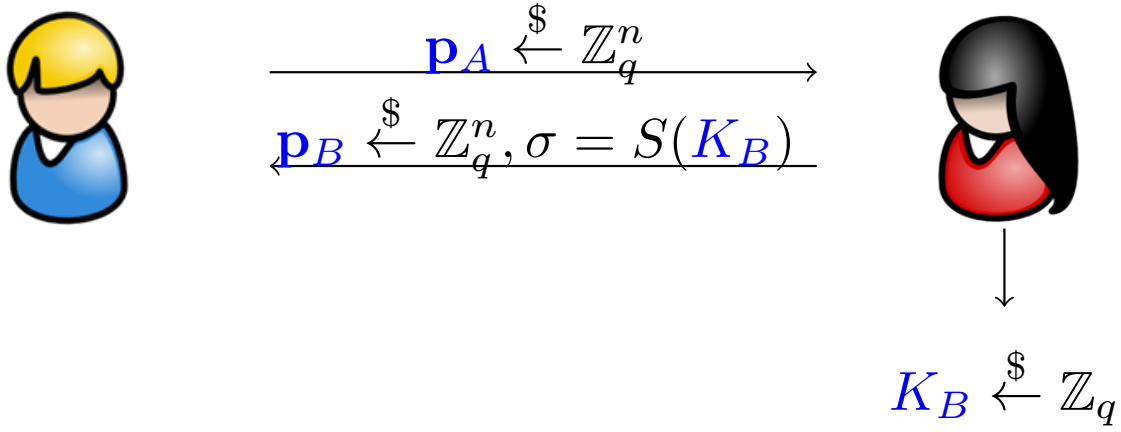
Replace $\mathbf{p}_A$ with uniform random one from $\mathbb{Z}_q^n$ (LWE assumption).

Proof Intuition



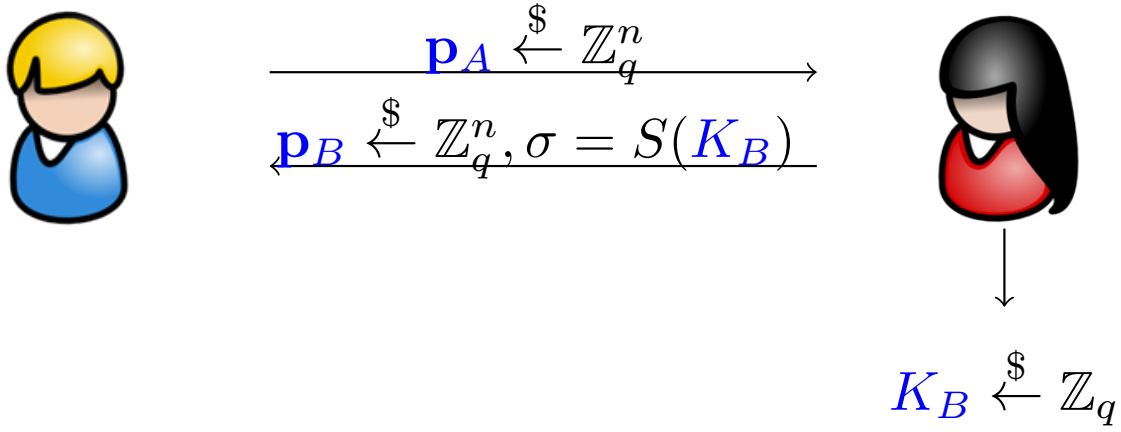
Replace $\mathbf{p}_A$ with uniform random one from $\mathbb{Z}_q^n$ (LWE assumption). Since $\mathbf{p}_A$ is uniform, we replace $\mathbf{p}_B$ and K_B with uniform ones (LWE assumption).

Proof Intuition



Replace $\mathbf{p}_A$ with uniform random one from $\mathbb{Z}_q^n$ (LWE assumption).
Since $\mathbf{p}_A$ is uniform, we replace $\mathbf{p}_B$ and K_B with uniform ones (LWE assumption).
Note that σ can always be computed.

Proof Intuition



Replace $\mathbf{p}_A$ with uniform random one from $\mathbb{Z}_q^n$ (LWE assumption).
Since $\mathbf{p}_A$ is uniform, we replace $\mathbf{p}_B$ and K_B with uniform ones (LWE assumption).
Note that σ can always be computed.
Now use the uniform property of robust extractors: $E(K_B, \sigma)$ is uniform, conditioned on σ.

Extend to RLWE

Ring Learning with Errors (RLWE) [LPR'10]:
Let $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = \mathbb{Z}_q[x]/(x^n + 1)$, $n = 2^k$ for $k \in \mathbb{Z}^+$.

Goal: distinguishing "noisy ring products" from uniform.

$a_1 \leftarrow R_q;$	$b_1 = a \cdot s + e_1 \in R_q$
$a_2 \leftarrow R_q;$	$b_2 = a_2 \cdot s + e_2 \in R_q$
$\vdots$	
$a_m \leftarrow R_q;$	$b_m = a_m \cdot s + e_m \in R_q$

$a_1 \leftarrow R_q;$	$b_1 \leftarrow R_q$
$a_2 \leftarrow R_q;$	$b_2 \leftarrow R_q$
$\vdots$	
$a_m \leftarrow R_q;$	$b_m \leftarrow R_q$

$s \leftarrow R_q$ and $e_i \leftarrow \chi$ is some distribution on R and $\|e_i\|$ is "small".

Key Exchange from RLWE

Public Parameter: $m \leftarrow R_q$

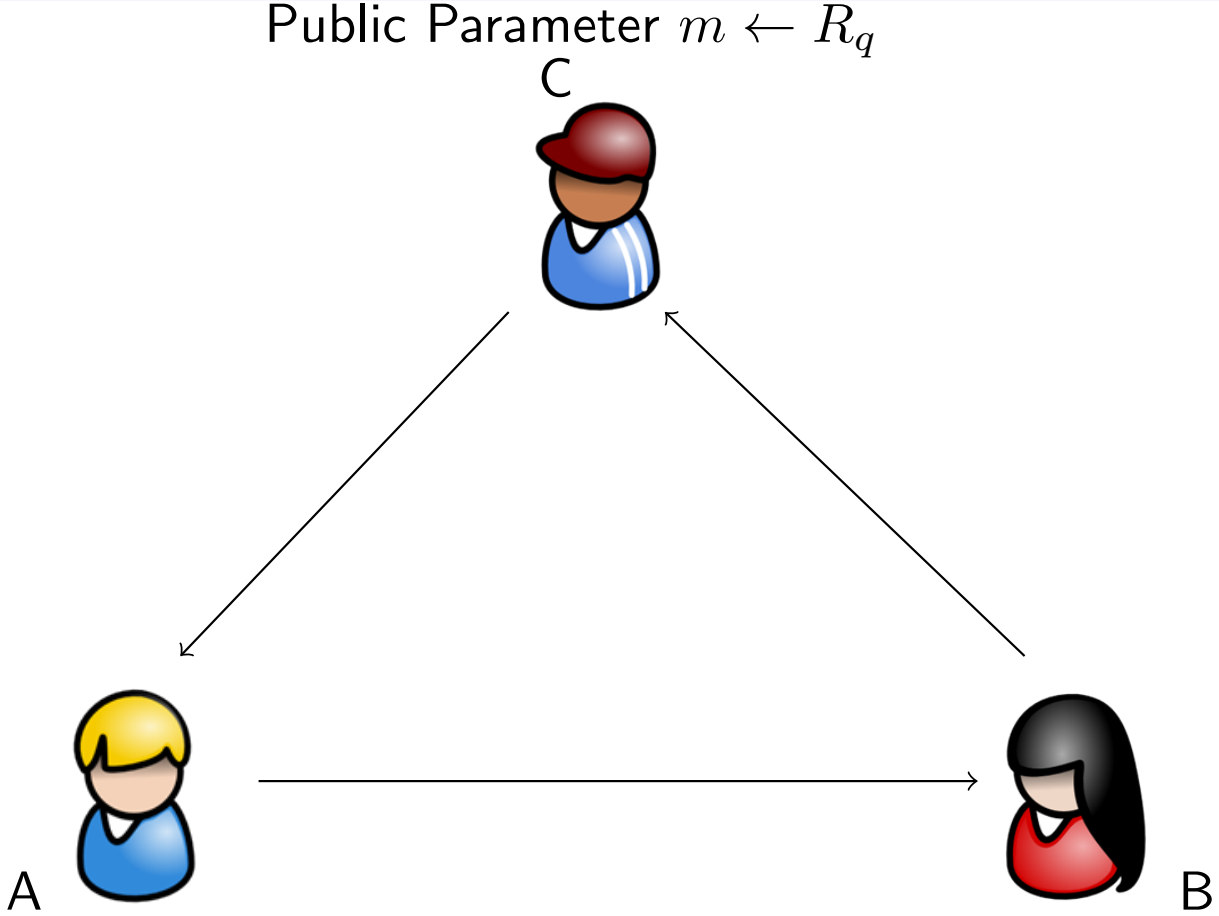
$$\xrightarrow{\quad p_A = ms_A + 2e_A \quad}$$

$$p_B = ms_B + 2e_B, \vec{\sigma} \leftarrow S(p_A s_B) \xleftarrow{\quad}$$

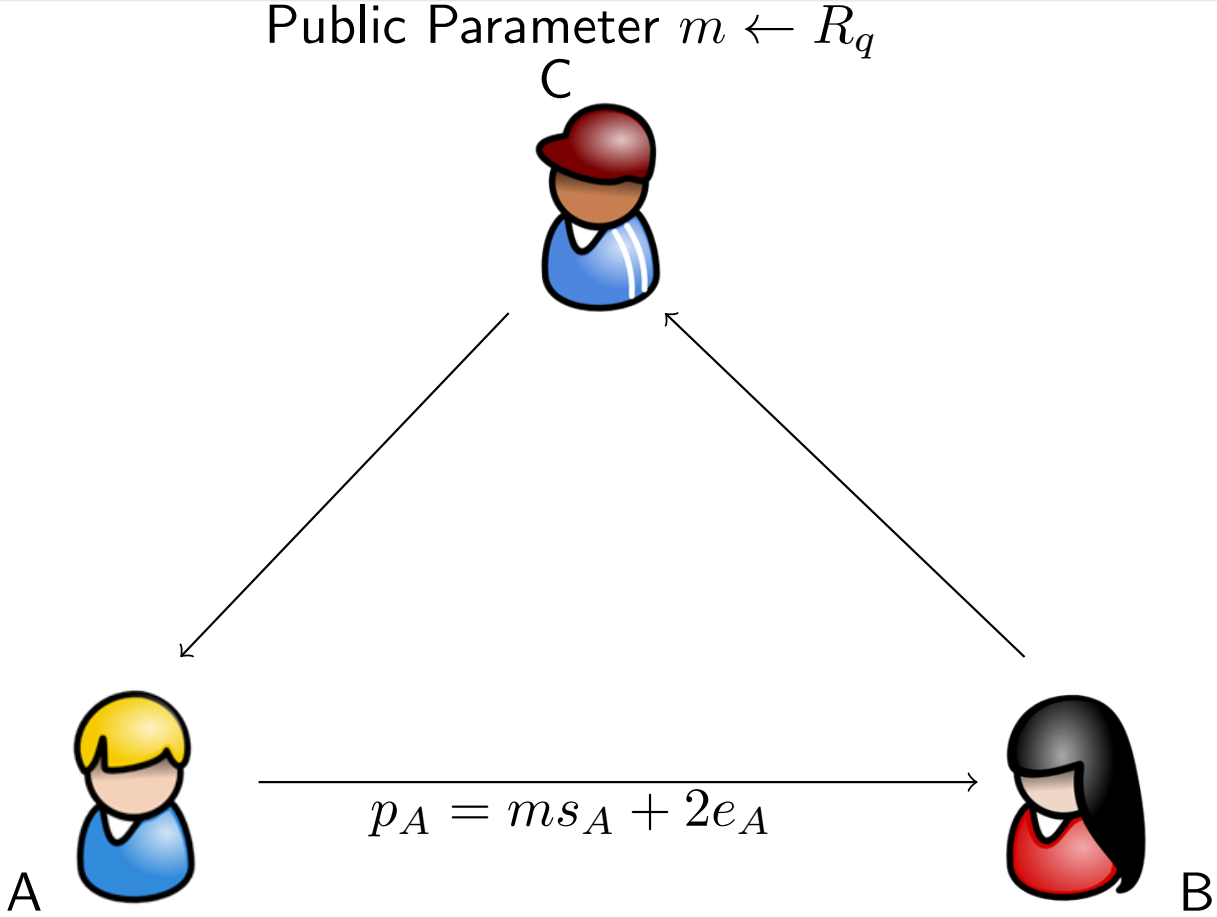
A B

- $\vec{\sigma}_b(a = \sum_{i=0}^{n-1} a_i X^i \in R_q) = \sum_{i=0}^{n-1} \sigma_b(a_i) X^i \in R_2$.
- $S(a) : b \xleftarrow{\$} \{0,1\}, S(a) = \vec{\sigma}_b(a)$.
- A outputs $E(s_A p_B, \vec{\sigma})$.
- B outputs $E(s_B p_A, \vec{\sigma})$.
- The shared secret key is $(s_A m s_B + \frac{q-1}{2}\vec{\sigma} \bmod q) \bmod 2 \in R_2$.

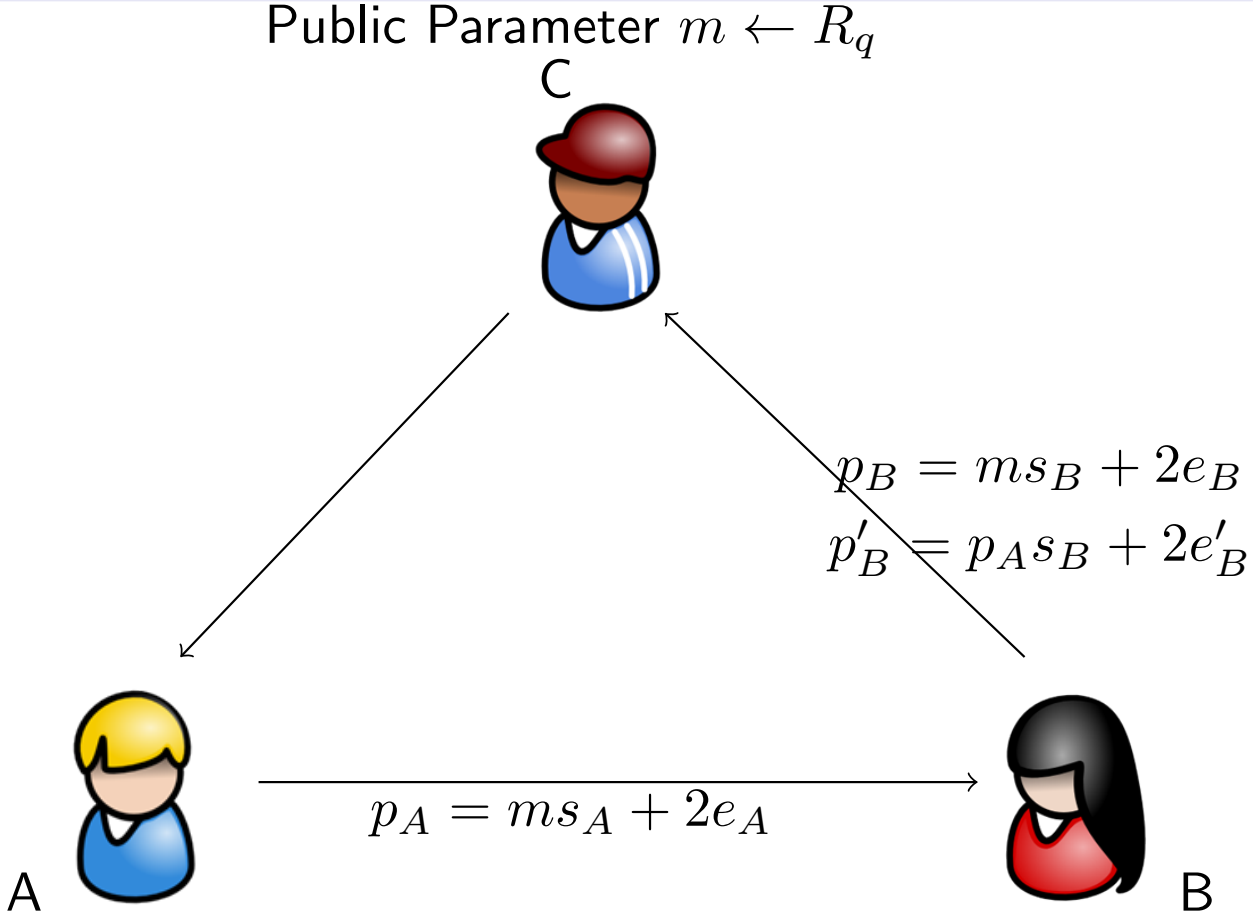
Multi-party Key Exchange



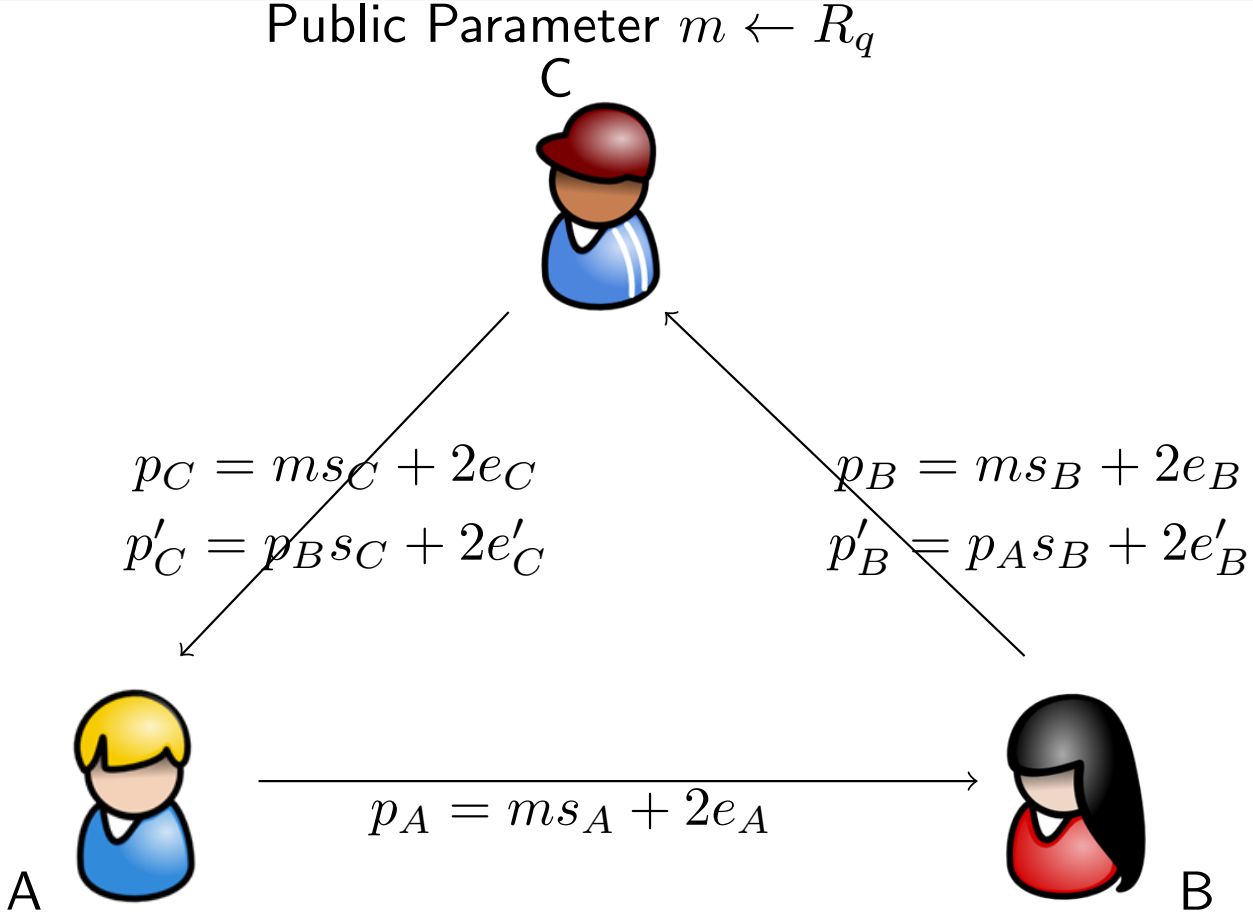
Multi-party Key Exchange



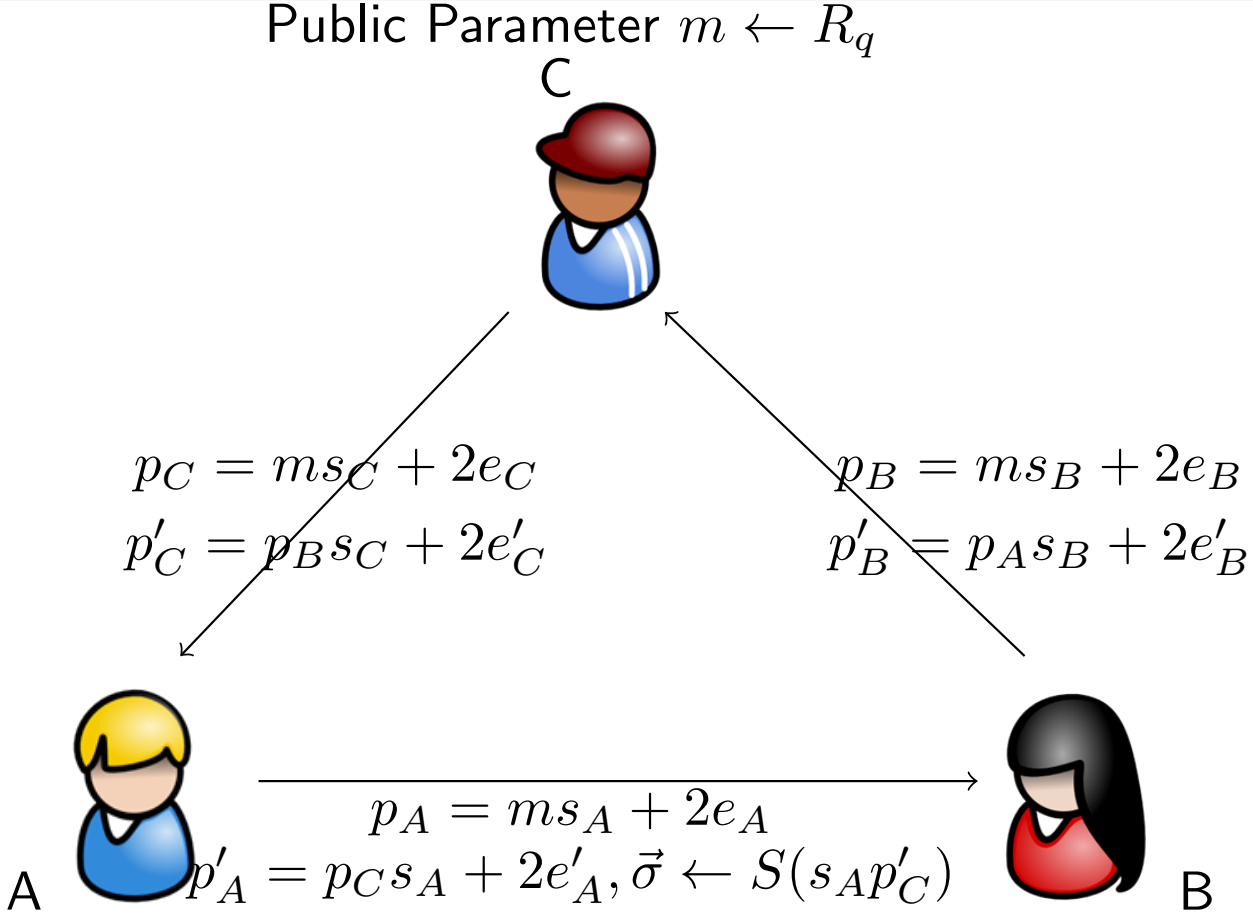
Multi-party Key Exchange



Multi-party Key Exchange



Multi-party Key Exchange



- A outputs $E(s_A p'_C, \vec{\sigma})$
- B outputs $E(s_B p'_A, \vec{\sigma})$
- C outputs $E(s_C p'_B, \vec{\sigma})$
- The shared key is $(s_A s_B s_C m + \frac{q-1}{2}\vec{\sigma} \bmod q) \bmod 2 \in R_2$.

- The correctness is similar to the previous protocols.
- The security proof involves some "circular" problem, we leave it as an open problem.

- This scheme is secure under passive attacks, but how about man-in-the-middle attacks?
- In this case, we need an autenticated KE.
  Traditionally, we use digital signature.
  Can we do without digital signature?
- We can build an authenticated key exchange (AKE) protocol, which can be seen as an HMQV-like AKE from lattices.
- The protocol is simple since it does not involve any other cryptographic primitives to achieve authentication (e.g., signatures) and the system is also very efficient.

Eurocrypt 2015

AKE from ring-LWE

Party i

Public Key: $p_i = as_i + 2e_i \in R_q$
Secret Key: $s_i \in R_q$
where $s_i, e_i \leftarrow_r \chi_\alpha$

Party j

Public Key: $p_j = as_j + 2e_j \in R_q$
Secret Key: $s_j \in R_q$
where $s_j, e_j \leftarrow_r \chi_\alpha$

AKE from ring-LWE

Party i

Public Key: $p_i = as_i + 2e_i \in R_q$
Secret Key: $s_i \in R_q$
where $s_i, e_i \leftarrow_r \chi_\alpha$

$x_i = ar_i + 2f_i \in R_q$
where $r_i, f_i \leftarrow_r \chi_\beta$

x_i ——————————

Party j

Public Key: $p_j = as_j + 2e_j \in R_q$
Secret Key: $s_j \in R_q$
where $s_j, e_j \leftarrow_r \chi_\alpha$

AKE from ring-LWE

Party i

Public Key: $p_i = as_i + 2e_i \in R_q$
Secret Key: $s_i \in R_q$
where $s_i, e_i \leftarrow_r \chi_\alpha$

$x_i = ar_i + 2f_i \in R_q$
where $r_i, f_i \leftarrow_r \chi_\beta$

x_i

---

y_j, w_j

Party j

Public Key: $p_j = as_j + 2e_j \in R_q$
Secret Key: $s_j \in R_q$
where $s_j, e_j \leftarrow_r \chi_\alpha$

$y_j = ar_j + 2f_j \in R_q$
$k_j = (p_i c + x_i)(s_j d + r_j) + 2g_j$
where $r_j, f_j, g_j \leftarrow_r \chi_\beta$
$w_j = \mathsf{Cha}(k_j) \in \{0,1\}^n$
$\sigma_j = \mathsf{Mod}_2(k_j, w_j) \in \{0,1\}^n$
$sk_j = H_2(i, j, x_i, y_j, w_j, \sigma_j)$

$$c = H_1(i, j, x_i) \in R, d = H_1(j, i, y_j, x_i) \in R$$

AKE from ring-LWE

Party i		Party j
Public Key: $p_i = as_i + 2e_i \in R_q$ Secret Key: $s_i \in R_q$ where $s_i, e_i \leftarrow_r \chi_\alpha$		Public Key: $p_j = as_j + 2e_j \in R_q$ Secret Key: $s_j \in R_q$ where $s_j, e_j \leftarrow_r \chi_\alpha$
$x_i = ar_i + 2f_i \in R_q$ where $r_i, f_i \leftarrow_r \chi_\beta$	$\xrightarrow{x_i}$	
		$y_j = ar_j + 2f_j \in R_q$ $k_j = (p_i c + x_i)(s_j d + r_j) + 2g_j$ where $r_j, f_j, g_j \leftarrow_r \chi_\beta$ $w_j = \mathsf{Cha}(k_j) \in \{0,1\}^n$ $\sigma_j = \mathsf{Mod}_2(k_j, w_j) \in \{0,1\}^n$ $sk_j = H_2(i, j, x_i, y_j, w_j, \sigma_j)$
	$\xleftarrow{y_j, w_j}$	
$k_i = (p_j d + y_j)(s_i c + r_i) + 2g_i$ where $g_i \leftarrow_r \chi_\beta$ $\sigma_i = \mathsf{Mod}_2(k_i, w_j) \in \{0,1\}^n$ $sk_i = H_2(i, j, x_i, y_j, w_j, \sigma_i)$		

$$c = H_1(i, j, x_i) \in R, d = H_1(j, i, y_j, x_i) \in R$$

AKE from ring-LWE

Intuition for Security:

1. We can prove the security of the system

AKE from ring-LWE

Intuition for Security:

1. We can prove the security of the system
2. We can prove the forward security of the system

AKE from ring-LWE

Intuition for Security:

1. We can prove the security of the system
2. We can prove the forward security of the system
3. We did preliminary implementation and it is very efficient.

AKE from ring-LWE

Intuition for Security:

1. We can prove the security of the system
2. We can prove the forward security of the system
3. We did preliminary implementation and it is very efficient.
4. Parameters for implementation:

Parameters	n	Security (expt.)	α	γ	$\log \frac{\beta}{\alpha}$	$\log q$ (bits)
I*	1024	80 bits	3.397	101.919	8.5	40
II	2048	80 bits	3.397	161.371	27	78
III	2048	128 bits	3.397	161.371	19	63
IV	4096	128 bits	3.397	256.495	50	125
V	4096	192 bits	3.397	256.495	36	97
VI	4096	256 bits	3.397	256.495	28	81

AKE from ring-LWE

Communication Overheads:

Choice of Parameters	Size (KB)			
	pk	sk (expt.)	init. msg	resp. msg
I*	5 KB	0.75 KB	5 KB	5.125 KB
II	19.5 KB	1.5 KB	19.5 KB	19.75 KB
III	15.75 KB	1.5 KB	15.75 KB	16 KB
IV	62.5 KB	3 KB	62.5 KB	63 KB
V	48.5 KB	3 KB	48.5 KB	49 KB
VI	40.5 KB	3 KB	40.5 KB	41 KB

The bound 6α with $\mathrm{erfc}(6) \approx 2^{-55}$ is used to estimate the size of secret keys.

AKE from ring-LWE

Timings:

Parameters	Initiation	Response	Finish
I	3.22 ms (0.02 ms)	8.50 ms (4.69 ms)	5.23 ms (4.73 ms)
II	12.00 ms (0.04 ms)	29.33 ms (14.64 ms)	17.28 ms (14.61 ms)
III	10.33 ms (0.04 ms)	25.83 ms (13.46 ms)	15.58 ms (13.40 ms)
IV	83.61 ms (0.08 ms)	156.58 ms (39.86 ms)	73.11 ms (39.73 ms)
V	61.74 ms (0.08 ms)	117.81 ms (32.58 ms)	55.64 ms (32.20 ms)
VI	25.42 ms (0.08 ms)	62.31 ms (31.32 ms)	36.80 ms (31.29 ms)

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 believe our systems are very suitable for practical applications and they have very strong security.

Summary

- We build KE and AKE based on LWE and RLWE.
- They are provably secure against both classical and quantum attacks.
- We can prove the Forward Security of the AKE.
- Our preliminary implementations are very efficient.
- Our KE and AKE are strong candidates for quantum-safe crypto.

Thank You!