CS 594 – Advanced Crypto (Spring 2026)

Alex Block

Lecture 25

April 22, 2026

The Learning with Errors Assumption

Solving Linear Equations

Solving Linear Equations

- Groundwork: solving systems of linear equations mod q for $q \in \mathbb{Z}^+$.

Solving Linear Equations

- Groundwork: solving systems of linear equations mod q for $q \in \mathbb{Z}^+$.

$$3 = \quad x_1 \quad +2x_2 \quad -2x_3 \quad +6x_5$$

Solving Linear Equations

- Groundwork: solving systems of linear equations mod q for $q \in \mathbb{Z}^+$.

$$\begin{aligned} 3 &= x_1 + 2x_2 - 2x_3 + 6x_5 \\ 1 &= 4x_1 + 3x_2 - 4x_3 - x_4 + 6x_5 \end{aligned}$$

Solving Linear Equations

- Groundwork: solving systems of linear equations mod q for $q \in \mathbb{Z}^+$.

$$
\begin{array}{rrrrrr}
3 = & x_1 & +2x_2 & -2x_3 & & +6x_5 \\
1 = & 4x_1 & +3x_2 & -4x_3 & -x_4 & +6x_5 \\
4 = & -x_1 & -x_2 & +4x_3 & +3x_4 & -3x_5 \\
1 = & x_1 & -x_2 & +2x_3 & +3x_4 & +x_5 \\
5 = & 4x_1 & +5x_2 & +5x_3 & +3x_4 & +x_5
\end{array}
$$

Solving Linear Equations

- Groundwork: solving systems of linear equations mod q for $q \in \mathbb{Z}^+$.

$$
\begin{array}{rrrrrr}
3 = & x_1 & +2x_2 & -2x_3 & & +6x_5 \\
1 = & 4x_1 & +3x_2 & -4x_3 & -x_4 & +6x_5 \\
4 = & -x_1 & -x_2 & +4x_3 & +3x_4 & -3x_5 \\
1 = & x_1 & -x_2 & +2x_3 & +3x_4 & +x_5 \\
5 = & 4x_1 & +5x_2 & +5x_3 & +3x_4 & +x_5
\end{array}
$$

- How can we solve this system?

Solving Linear Equations

- Linear algebra, of course!

Solving Linear Equations

- Linear algebra, of course!

$$\begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \\ 5 \end{bmatrix}$$

Solving Linear Equations

- Linear algebra, of course!

$$\begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -2 & 0 & 6 \\ 4 & 3 & -4 & -3 & 6 \\ -1 & -1 & 4 & 3 & -3 \\ 1 & -1 & 2 & 3 & 1 \\ 4 & 5 & 5 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \bmod q$$

Solving Linear Equations

- Linear algebra, of course!

$$\begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -2 & 0 & 6 \\ 4 & 3 & -4 & -3 & 6 \\ -1 & -1 & 4 & 3 & -3 \\ 1 & -1 & 2 & 3 & 1 \\ 4 & 5 & 5 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \bmod q$$

$$\begin{bmatrix} 1 & 2 & -2 & 0 & 6 \\ 4 & 3 & -4 & -3 & 6 \\ -1 & -1 & 4 & 3 & -3 \\ 1 & -1 & 2 & 3 & 1 \\ 4 & 5 & 5 & 3 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \\ 5 \end{bmatrix} \bmod q = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Solving Linear Equations

- Linear algebra, of course!

$$\begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -2 & 0 & 6 \\ 4 & 3 & -4 & -3 & 6 \\ -1 & -1 & 4 & 3 & -3 \\ 1 & -1 & 2 & 3 & 1 \\ 4 & 5 & 5 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \bmod q$$

$$\begin{bmatrix} 1 & 2 & -2 & 0 & 6 \\ 4 & 3 & -4 & -3 & 6 \\ -1 & -1 & 4 & 3 & -3 \\ 1 & -1 & 2 & 3 & 1 \\ 4 & 5 & 5 & 3 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \\ 5 \end{bmatrix} \bmod q = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

- So long as the matrix in invertible mod q, we can solve in polynomial time.

Solving Approximate Linear Equations

- Note that for parameters m, n with $m \geqslant n$, for $\mathbf{t} \in \mathbb{Z}_q^m$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the problem of solving $\mathbf{t}^\top = \mathbf{A} \cdot \mathbf{x}^\top$ is not made harder by increasing m.

Solving Approximate Linear Equations

- Note that for parameters m, n with $m \geqslant n$, for $\mathbf{t} \in \mathbb{Z}_q^m$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the problem of solving $\mathbf{t}^\top = \mathbf{A} \cdot \mathbf{x}^\top$ is not made harder by increasing m.
  - I.e., though there may not always be a solution, if one exists we can still find it in polynomial time.

Solving Approximate Linear Equations

- Note that for parameters m, n with $m \geqslant n$, for $\mathbf{t} \in \mathbb{Z}_q^m$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the problem of solving $\mathbf{t}^\top = \mathbf{A} \cdot \mathbf{x}^\top$ is not made harder by increasing m.
  - I.e., though there may not always be a solution, if one exists we can still find it in polynomial time.
- **Question**: what if we want to solve an *approximate* linear system?

Solving Approximate Linear Equations

- Note that for parameters m, n with $m \geqslant n$, for $\mathbf{t} \in \mathbb{Z}_q^m$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the problem of solving $\mathbf{t}^\top = \mathbf{A} \cdot \mathbf{x}^\top$ is not made harder by increasing m.
  - I.e., though there may not always be a solution, if one exists we can still find it in polynomial time.
- **Question**: what if we want to solve an *approximate* linear system?

$$2 \approx \quad x_1 \quad +2x_2 \quad -2x_3 \quad +6x_5$$

Solving Approximate Linear Equations

- Note that for parameters m, n with $m \geqslant n$, for $\mathbf{t} \in \mathbb{Z}_q^m$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the problem of solving $\mathbf{t}^\top = \mathbf{A} \cdot \mathbf{x}^\top$ is not made harder by increasing m.
  - I.e., though there may not always be a solution, if one exists we can still find it in polynomial time.
- **Question**: what if we want to solve an *approximate* linear system?

$$
\begin{aligned}
2 \approx \quad & x_1 & +2x_2 & & -2x_3 & & & & +6x_5 \\
7 \approx \quad & 4x_1 & +3x_2 & & -4x_3 & & -x_4 & & +6x_5
\end{aligned}
$$

Solving Approximate Linear Equations

- Note that for parameters m, n with $m \geqslant n$, for $\mathbf{t} \in \mathbb{Z}_q^m$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the problem of solving $\mathbf{t}^\top = \mathbf{A} \cdot \mathbf{x}^\top$ is not made harder by increasing m.
  - I.e., though there may not always be a solution, if one exists we can still find it in polynomial time.
- **Question**: what if we want to solve an *approximate* linear system?

$$
\begin{aligned}
2 \approx \quad & x_1 & +2x_2 & & -2x_3 & & & & +6x_5 \\
7 \approx \quad & 4x_1 & +3x_2 & & -4x_3 & & -x_4 & & +6x_5 \\
1 \approx \quad & -x_1 & -x_2 & & +4x_3 & & +3x_4 & & -3x_5
\end{aligned}
$$

Solving Approximate Linear Equations

- Note that for parameters m, n with $m \geqslant n$, for $\mathbf{t} \in \mathbb{Z}_q^m$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the problem of solving $\mathbf{t}^\top = \mathbf{A} \cdot \mathbf{x}^\top$ is not made harder by increasing m.
  - I.e., though there may not always be a solution, if one exists we can still find it in polynomial time.
- **Question**: what if we want to solve an *approximate* linear system?

$$
\begin{array}{rrrrrr}
2 \approx & x_1 & +2x_2 & -2x_3 & & +6x_5 \\
7 \approx & 4x_1 & +3x_2 & -4x_3 & -x_4 & +6x_5 \\
1 \approx & -x_1 & -x_2 & +4x_3 & +3x_4 & -3x_5 \\
8 \approx & x_1 & -x_2 & +2x_3 & +3x_4 & +x_5 \\
2 \approx & 4x_1 & +5x_2 & +52x_3 & +3x_4 & +x_5 \\
8 \approx & 2x_1 & & +x_3 & -x_4 & +2x_5 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

Solving Approximate Linear Equations

- Here, $\approx$ means that the two sides only differ by a small amount.

Solving Approximate Linear Equations

- Here, $\approx$ means that the two sides only differ by a small amount.
  - For example, they only differ by $\{0, \pm 1\}$.

Solving Approximate Linear Equations

- Here, $\approx$ means that the two sides only differ by a small amount.
  - For example, they only differ by $\{0, \pm 1\}$.
  - In this case, for example, you want to find $\mathbf{x} \in \mathbb{Z}_q^5$ such that $\left|\mathbf{t}_i - \mathbf{A}_i \mathbf{x}^\top\right| \leqslant 1$ for all i.

↑ row i

Solving Approximate Linear Equations

- Here, $\approx$ means that the two sides only differ by a small amount.
  - For example, they only differ by $\{0, \pm 1\}$.
  - In this case, for example, you want to find $\mathbf{x} \in \mathbb{Z}_q^5$ such that $\left|\mathbf{t}_i - \mathbf{A}_i \mathbf{x}^\top\right| \leqslant 1$ for all i.
- We can rewrite this constraint by utilizing an error vector $\mathbf{e}$.

Solving Approximate Linear Equations

- Here, $\approx$ means that the two sides only differ by a small amount.
  - For example, they only differ by $\{0, \pm 1\}$.
  - In this case, for example, you want to find $\mathbf{x} \in \mathbb{Z}_q^5$ such that $\left|\mathbf{t}_i - \mathbf{A}_i \mathbf{x}^\top\right| \leqslant 1$ for all i.
- We can rewrite this constraint by utilizing an error vector $\mathbf{e}$.

$$\begin{bmatrix} 2 \\ 7 \\ 1 \\ 8 \\ 2 \\ 8 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 2 & -2 & 0 & 6 \\ 4 & 3 & -4 & -3 & 6 \\ -1 & -1 & 4 & 3 & -3 \\ 1 & -1 & 2 & 3 & 1 \\ 4 & 5 & 5 & 3 & 1 \\ 2 & 0 & 3 & -4 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ \vdots \end{bmatrix} \bmod q$$

t A x e

Solving Approximate Linear Equations

- Here, $\approx$ means that the two sides only differ by a small amount.
  - For example, they only differ by $\{0, \pm 1\}$.
  - In this case, for example, you want to find $\mathbf{x} \in \mathbb{Z}_q^5$ such that $\left|\mathbf{t}_i - \mathbf{A}_i \mathbf{x}^\top\right| \leqslant 1$ for all i.
- We can rewrite this constraint by utilizing an error vector $\mathbf{e}$.

$$\begin{bmatrix} 2 \\ 7 \\ 1 \\ 8 \\ 2 \\ 8 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 2 & -2 & 0 & 6 \\ 4 & 3 & -4 & -3 & 6 \\ -1 & -1 & 4 & 3 & -3 \\ 1 & -1 & 2 & 3 & 1 \\ 4 & 5 & 5 & 3 & 1 \\ 2 & 0 & 3 & -4 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ \vdots \end{bmatrix} \bmod q$$

- Here, we know that $\mathbf{e}_i \in \{0, \pm 1\}$ (i.e., they are small), but are otherwise *unknown*.

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.
  - The vector $\mathbf{e}$ is a *noise* or *error* vector.

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.
  - The vector $\mathbf{e}$ is a *noise* or *error* vector.

- Why are these systems of equations useful for crypto?

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.
  - The vector $\mathbf{e}$ is a *noise* or *error* vector.

- Why are these systems of equations useful for crypto?
  - On the one hand, for $\mathbf{A} \in \mathbb{F}^{m \times n}$ and *large enough* m, there is generally at most one possible $\mathbf{x}$ and $\mathbf{e}$ to solve $\mathbf{t}^\top = \mathbf{A}\mathbf{x}^\top + \mathbf{e}^\top$.

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.
  - The vector $\mathbf{e}$ is a *noise* or *error* vector.

- Why are these systems of equations useful for crypto?
  - On the one hand, for $\mathbf{A} \in \mathbb{F}^{m \times n}$ and *large enough* m, there is generally at most one possible $\mathbf{x}$ and $\mathbf{e}$ to solve $\mathbf{t}^\top = \mathbf{A}\mathbf{x}^\top + \mathbf{e}^\top$.
  - On the other hand, it appears to be *very hard* to *find* these solutions or even determine if one exists.

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.
  - The vector $\mathbf{e}$ is a *noise* or *error* vector.

- Why are these systems of equations useful for crypto?
  - On the one hand, for $\mathbf{A} \in \mathbb{F}^{m \times n}$ and *large enough* m, there is generally at most one possible $\mathbf{x}$ and $\mathbf{e}$ to solve $\mathbf{t}^\top = \mathbf{A}\mathbf{x}^\top + \mathbf{e}^\top$.
  - On the other hand, it appears to be *very hard* to *find* these solutions or even determine if one exists.
    - All known algorithms, even quantum ones, take *exponential* time!

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.
  - The vector $\mathbf{e}$ is a *noise* or *error* vector.
- Why are these systems of equations useful for crypto?
  - On the one hand, for $\mathbf{A} \in \mathbb{F}^{m \times n}$ and *large enough* m, there is generally at most one possible $\mathbf{x}$ and $\mathbf{e}$ to solve $\mathbf{t}^\top = \mathbf{A}\mathbf{x}^\top + \mathbf{e}^\top$.
  - On the other hand, it appears to be *very hard* to *find* these solutions or even determine if one exists.
    - All known algorithms, even quantum ones, take *exponential* time!
- These properties make such systems of equations very useful for crypto!

Noisy Linear Systems are Useful for Crypto

- We call approximate linear systems *noisy linear systems*.
  - The vector $\mathbf{e}$ is a *noise* or *error* vector.

- Why are these systems of equations useful for crypto?
  - On the one hand, for $\mathbf{A} \in \mathbb{F}^{m \times n}$ and *large enough* m, there is generally at most one possible $\mathbf{x}$ and $\mathbf{e}$ to solve $\mathbf{t}^\top = \mathbf{A}\mathbf{x}^\top + \mathbf{e}^\top$.
  - On the other hand, it appears to be *very hard* to *find* these solutions or even determine if one exists.
    - All known algorithms, even quantum ones, take *exponential* time!

- These properties make such systems of equations very useful for crypto!
  - Recall a one-way function $f\colon \{0,1\}^* \to \{0,1\}^\lambda$: easy to compute $f(x) = y$ for any x, but hard to find x' such that $f(x') = y$ when given $y = f(x)$ for random x.

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

Let $m \geqslant n$ and q be positive integers.

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$.

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *Learning with Errors* (LWE) assumption states that the following two distributions are computationally indistinguishable.

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *Learning with Errors* (LWE) assumption states that the following two distributions are computationally indistinguishable.

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{real}}}$

$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$

$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n}$

$\mathbf{e} \leftarrow \mathcal{E}^m$

$\mathbf{t} = \mathbf{M}\mathbf{s}^\top + \mathbf{e}^\top$

return $(\mathbf{M}, \mathbf{t})$.

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *Learning with Errors* (LWE) assumption states that the following two distributions are computationally indistinguishable.

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{real}}}$

$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$

$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n}$

$\mathbf{e} \leftarrow \mathcal{E}^m$

$\mathbf{t} = \mathbf{M}\mathbf{s}^\top + \mathbf{e}^\top$

return $(\mathbf{M}, \mathbf{t})$.

$\approx_c$

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *Learning with Errors* (LWE) assumption states that the following two distributions are computationally indistinguishable.

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{real}}}$
$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n}$
$\mathbf{e} \leftarrow \mathcal{E}^{m}$
$\mathbf{t} = \mathbf{M}\mathbf{s}^{\top} + \mathbf{e}^{\top}$
return $(\mathbf{M}, \mathbf{t})$.

$\approx_c$

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{rand}}}$
$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
$\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^{m}$
return $(\mathbf{M}, \mathbf{t})$.

The Learning with Errors Problem

- The *Learning with Errors* (LWE) formalizes this problem.

Definition 1 (The LWE Assumption)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *Learning with Errors* (LWE) assumption states that the following two distributions are computationally indistinguishable.

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{real}}}$
$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n}$
$\mathbf{e} \leftarrow \mathcal{E}^m$
$\mathbf{t} = \mathbf{M}\mathbf{s}^\top + \mathbf{e}^\top$
return $(\mathbf{M}, \mathbf{t})$.

$\approx_c$

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{rand}}}$
$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
$\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^{m}$
return $(\mathbf{M}, \mathbf{t})$.

- "Learning with Errors:" and adversary is trying to learn the secret vector $\mathbf{s}$ while getting noisy/error samples $\mathbf{M}_i\mathbf{s}^\top + \mathbf{e}_i$.

LWE Security and Parameters

LWE Security and Parameters

- Suppose we want to target λ-bits of security.

LWE Security and Parameters

- Suppose we want to target λ-bits of security.
- Unlike RSA or Discrete-log, there is no "one-size-fits-all" recommendation.

LWE Security and Parameters

- Suppose we want to target λ-bits of security.
- Unlike RSA or Discrete-log, there is no "one-size-fits-all" recommendation.
  - Issue: different constructions using LWE need different settings of m, n, q, and even $\mathcal{E}$.

LWE Security and Parameters

- Suppose we want to target λ-bits of security.
- Unlike RSA or Discrete-log, there is no "one-size-fits-all" recommendation.
  - Issue: different constructions using LWE need different settings of m, n, q, and even $\mathcal{E}$.

- The following parameter suggestions are a rough estimate for basic LWE applications in order to achieve approximately λ-bits of security.

LWE Security and Parameters

- Suppose we want to target λ-bits of security.
- Unlike RSA or Discrete-log, there is no "one-size-fits-all" recommendation.
  - Issue: different constructions using LWE need different settings of m, n, q, and even $\mathcal{E}$.

- The following parameter suggestions are a rough estimate for basic LWE applications in order to achieve approximately λ-bits of security.
  - $n = \lambda$;

LWE Security and Parameters

- Suppose we want to target λ-bits of security.
- Unlike RSA or Discrete-log, there is no "one-size-fits-all" recommendation.
  - Issue: different constructions using LWE need different settings of m, n, q, and even $\mathcal{E}$.

- The following parameter suggestions are a rough estimate for basic LWE applications in order to achieve approximately λ-bits of security.
  - $n = \lambda$;
  - $q \approx \lambda^2$;

LWE Security and Parameters

- Suppose we want to target λ-bits of security.
- Unlike RSA or Discrete-log, there is no "one-size-fits-all" recommendation.
  - Issue: different constructions using LWE need different settings of m, n, q, and even $\mathcal{E}$.

- The following parameter suggestions are a rough estimate for basic LWE applications in order to achieve approximately λ-bits of security.
  - $n = \lambda$;
  - $q \approx \lambda^2$;
  - $\mathcal{E} = \{-\sqrt{\lambda}, \ldots, -1, 0, 1, \ldots, \sqrt{\lambda}\} \subset \mathbb{Z}_q$.

uniform: output each element w.p. $\frac{1}{2\sqrt{\lambda}+1}$

Discrete Gaussian

LWE Security and Parameters

- Suppose we want to target λ-bits of security.
- Unlike RSA or Discrete-log, there is no "one-size-fits-all" recommendation.
  - Issue: different constructions using LWE need different settings of m, n, q, and even $\mathcal{E}$.

- The following parameter suggestions are a rough estimate for basic LWE applications in order to achieve approximately λ-bits of security.
  - $n = \lambda$;
  - $q \approx \lambda^2$;
  - $\mathcal{E} \subseteq \{-\sqrt{\lambda}, \ldots, -1, 0, 1, \ldots, \sqrt{\lambda}\} \subset \mathbb{Z}_q$.

- With these parameters, the LWE assumption is believed to hold for any $m = \text{poly}(\lambda)$, even against quantum adversaries!

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

Definition 2 (Short LWE)

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

Definition 2 (Short LWE)

Let $m \geqslant n$ and q be positive integers.

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

Definition 2 (Short LWE)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$.

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

Definition 2 (Short LWE)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *LWE with short secrets* assumption states that the following two distributions are computationally indistinguishable.

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

Definition 2 (Short LWE)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *LWE with short secrets* assumption states that the following two distributions are computationally indistinguishable.

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{sreal}}}$

$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$

$\mathbf{s} \xleftarrow{\$} \mathcal{E}^n$

$\mathbf{e} \leftarrow \mathcal{E}^m$

$\mathbf{t} = \mathbf{M}\mathbf{s}^\top + \mathbf{e}^\top$

return $(\mathbf{M}, \mathbf{t})$.

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

Definition 2 (Short LWE)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *LWE with short secrets* assumption states that the following two distributions are computationally indistinguishable.

$\mathcal{L}^{m,n,q}_{\mathsf{sreal}}$

$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$

$\mathbf{s} \xleftarrow{\$} \mathcal{E}^n$

$\mathbf{e} \leftarrow \mathcal{E}^m$

$\mathbf{t} = \mathbf{M}\mathbf{s}^\top + \mathbf{e}^\top$

return $(\mathbf{M}, \mathbf{t})$.

$\approx_c$

LWE with Short Secrets

- Note that we can also define LWE with respect to the secret vector being short and not uniform.
- This is called *LWE with short secrets*.

Definition 2 (Short LWE)

Let $m \geqslant n$ and q be positive integers. Let $\mathcal{E}$ be a distribution over $\mathbb{Z}_q$. Then, the *LWE with short secrets* assumption states that the following two distributions are computationally indistinguishable.

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{sreal}}}$
$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
$\mathbf{s} \xleftarrow{\$} \mathcal{E}^n$
$\mathbf{e} \leftarrow \mathcal{E}^m$
$\mathbf{t} = \mathbf{M}\mathbf{s}^\top + \mathbf{e}^\top$
return $(\mathbf{M}, \mathbf{t})$.

$\approx_c$

$\underline{\mathcal{L}^{m,n,q}_{\mathsf{srand}}}$
$\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
$\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^{m}$
return $(\mathbf{M}, \mathbf{t})$.

LWE with Short Secrets

- Perhaps surprisingly, the hardness of LWE does not change if we sample $\mathbf{s}$ from $\mathcal{E}^n$ instead of uniformly over $\mathbb{Z}_q^n$.

LWE with Short Secrets

- Perhaps surprisingly, the hardness of LWE does not change if we sample $\mathbf{s}$ from $\mathcal{E}^n$ instead of uniformly over $\mathbb{Z}_q^n$.

Lemma 1

If the LWE assumption is true, then the LWE assumption with short secrets is true.

LWE with Short Secrets

- Perhaps surprisingly, the hardness of LWE does not change if we sample $\mathbf{s}$ from $\mathcal{E}^n$ instead of uniformly over $\mathbb{Z}_q^n$.

Lemma 1

If the LWE assumption is true, then the LWE assumption with short secrets is true.

- Proof: suppose the LWE assumption holds.

LWE with Short Secrets

- Perhaps surprisingly, the hardness of LWE does not change if we sample $\mathbf{s}$ from $\mathcal{E}^n$ instead of uniformly over $\mathbb{Z}_q^n$.

Lemma 1

If the LWE assumption is true, then the LWE assumption with short secrets is true.

- Proof: suppose the LWE assumption holds. We proceed via a hybrid argument and start by defining two hybrids H_0 and H_1.

LWE with Short Secrets

- Perhaps surprisingly, the hardness of LWE does not change if we sample $\mathbf{s}$ from $\mathcal{E}^n$ instead of uniformly over $\mathbb{Z}_q^n$.

Lemma 1

If the LWE assumption is true, then the LWE assumption with short secrets is true.

- Proof: suppose the LWE assumption holds. We proceed via a hybrid argument and start by defining two hybrids H_0 and H_1.

$\underline{H_0}$
$(\mathbf{M}, \mathbf{t}) \leftarrow \mathcal{L}^{m,n,q}_{\mathsf{sreal}}$
return $(\mathbf{M}, \mathbf{t})$

$M \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
$s \leftarrow \mathcal{E}^n$ $\quad t = Ms^T + e^T$
$e \leftarrow \mathcal{E}^n$

LWE with Short Secrets

- Perhaps surprisingly, the hardness of LWE does not change if we sample $\mathbf{s}$ from $\mathcal{E}^n$ instead of uniformly over $\mathbb{Z}_q^n$.

Lemma 1

If the LWE assumption is true, then the LWE assumption with short secrets is true.

- Proof: suppose the LWE assumption holds. We proceed via a hybrid argument and start by defining two hybrids H_0 and H_1.

H_0

$(\mathbf{M}, \mathbf{t}) \leftarrow \mathcal{L}_{\mathsf{sreal}}^{m,n,q}$

return $(\mathbf{M}, \mathbf{t})$

H_1

$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$

$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$

$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$

return $(\mathbf{M}^*, \mathbf{t}^*)$

LWE with Short Secrets

- Perhaps surprisingly, the hardness of LWE does not change if we sample $\mathbf{s}$ from $\mathcal{E}^n$ instead of uniformly over $\mathbb{Z}_q^n$.

Lemma 1

If the LWE assumption is true, then the LWE assumption with short secrets is true.

- Proof: suppose the LWE assumption holds. We proceed via a hybrid argument and start by defining two hybrids H_0 and H_1.

$$\underline{H_0}$$
$$(\mathbf{M}, \mathbf{t}) \leftarrow \mathcal{L}_{\mathsf{sreal}}^{m,n,q}$$
$$\text{return } (\mathbf{M}, \mathbf{t})$$

$$\equiv$$

$$\underline{H_1}$$
$$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$$
$$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$$
$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$
$$\text{return } (\mathbf{M}^*, \mathbf{t}^*)$$

- Claim: $H_0 \equiv H_1$

LWE with Short Secrets

- Claim follows by the following observations.

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

    This implies:

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

This implies:

$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

This implies:

$$\begin{aligned}\mathbf{t}^* &= \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2 \\ &= (-\mathbf{M}_2\mathbf{M}_1^{-1})(\mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1) + (\mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top)\end{aligned}$$

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

This implies:

$$\begin{aligned}\mathbf{t}^* &= \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2 \\ &= (-\mathbf{M}_2\mathbf{M}_1^{-1})(\mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1) + (\mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top) \\ &= -\mathbf{M}_2\mathbf{s}^\top + \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top\end{aligned}$$

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

This implies:

$$\begin{aligned}
\mathbf{t}^* &= \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2 \\
&= (-\mathbf{M}_2\mathbf{M}_1^{-1})(\mathbf{M_1}\mathbf{s}^\top + \mathbf{e}_1) + (\mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top) \\
&= -\mathbf{M}_2\mathbf{s}^\top + \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top \\
&= \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{e}_2^\top
\end{aligned}$$

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

    This implies:

$$\begin{aligned}\mathbf{t}^* &= \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2 \\ &= (-\mathbf{M}_2\mathbf{M}_1^{-1})(\mathbf{M_1}\mathbf{s}^\top + \mathbf{e}_1) + (\mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top) \\ &= -\mathbf{M}_2\mathbf{s}^\top + \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top \\ &= \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{e}_2^\top\end{aligned}$$

- Moreover, since $\mathbf{M}_1, \mathbf{M}_2$ are uniformly random, so is $\mathbf{M}^*$ (which has dimension $m \times n$).

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

    This implies:

$$\begin{aligned}\mathbf{t}^* &= \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2 \\ &= (-\mathbf{M}_2\mathbf{M}_1^{-1})(\mathbf{M_1}\mathbf{s}^\top + \mathbf{e}_1) + (\mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top) \\ &= -\mathbf{M}_2\mathbf{s}^\top + \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top \\ &= \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{e}_2^\top\end{aligned}$$

- Moreover, since $\mathbf{M}_1, \mathbf{M}_2$ are uniformly random, so is $\mathbf{M}^*$ (which has dimension $m \times n$).
- $\mathbf{e}_1, \mathbf{e}_2$ are both drawn from $\mathcal{E}^m$.

LWE with Short Secrets

- Claim follows by the following observations.
  - By definition in H_1, we have

$$\mathbf{t}_1 = \mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1^\top \quad \mathbf{t}_2 = \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top$$

This implies:

$$\begin{aligned}\mathbf{t}^* &= \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2 \\ &= (-\mathbf{M}_2\mathbf{M}_1^{-1})(\mathbf{M}_1\mathbf{s}^\top + \mathbf{e}_1) + (\mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top) \\ &= -\mathbf{M}_2\mathbf{s}^\top + \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{M}_2\mathbf{s}^\top + \mathbf{e}_2^\top \\ &= \mathbf{M}^*\mathbf{e}_1^\top + \mathbf{e}_2^\top\end{aligned}$$

- Moreover, since $\mathbf{M}_1, \mathbf{M}_2$ are uniformly random, so is $\mathbf{M}^*$ (which has dimension $m \times n$).
- $\mathbf{e}_1, \mathbf{e}_2$ are both drawn from $\mathcal{E}^m$.
- Therefore, $(\mathbf{M}^*, \mathbf{t}^*)$ are distributed as $\mathcal{L}^{m,n,q}_{\mathsf{sreal}}$.

LWE with Short Secrets

- Next, we define hybrid H_2.

LWE with Short Secrets

- Next, we define hybrid H_2.

$\underline{H_1}$

$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$

$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$

$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$

return $(\mathbf{M}^*, \mathbf{t}^*)$

LWE with Short Secrets

- Next, we define hybrid H_2.

$\underline{H_1}$

$$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$$
$$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$$
$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$
return $(\mathbf{M}^*, \mathbf{t}^*)$

$\underline{H_2}$

$$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{rand}}^{(n+m),n,q}$$
$$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$$
$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$
return $(\mathbf{M}^*, \mathbf{t}^*)$

LWE with Short Secrets

- Next, we define hybrid H_2.

$\underline{H_1}$

$$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$$
$$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$$
$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$
return $(\mathbf{M}^*, \mathbf{t}^*)$

$\approx_c$

$\underline{H_2}$

$$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{rand}}^{(n+m),n,q}$$
$$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$$
$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$
return $(\mathbf{M}^*, \mathbf{t}^*)$

- Claim: $H_1 \approx_c H_2$.

LWE with Short Secrets

- Next, we define hybrid H_2.

$\underline{H_1}$

$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$

$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$

$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$

return $(\mathbf{M}^*, \mathbf{t}^*)$

$\approx_c$

$\underline{H_2}$

$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{rand}}^{(n+m),n,q}$

$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$

$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$

return $(\mathbf{M}^*, \mathbf{t}^*)$

- Claim: $H_1 \approx_c H_2$.
  - This follows by the LWE assumption!

LWE with Short Secrets

- Next, we define hybrid H_2.

$\underline{H_1}$

$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$

$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$

$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$

return $(\mathbf{M}^*, \mathbf{t}^*)$

$\approx_c$

$\underline{H_2}$

$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{rand}}^{(n+m),n,q}$

$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$

$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$

return $(\mathbf{M}^*, \mathbf{t}^*)$

- Claim: $H_1 \approx_c H_2$.
  - This follows by the LWE assumption!
- Finally, we define hybrid H_3

LWE with Short Secrets

- Next, we define hybrid H_2.

H_1

$$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{real}}^{(n+m),n,q}$$
$$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$$
$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$
return $(\mathbf{M}^*, \mathbf{t}^*)$

$\approx_c$

H_2

$$\left(\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}\right) \leftarrow \mathcal{L}_{\mathsf{rand}}^{(n+m),n,q}$$
$$\mathbf{M}^* = -\mathbf{M}_2\mathbf{M}_1^{-1}$$
$$\mathbf{t}^* = \mathbf{M}^*\mathbf{t}_1 + \mathbf{t}_2$$
return $(\mathbf{M}^*, \mathbf{t}^*)$

Uniform over $\mathbb{Z}_p^{m \times n}$

- Claim: $H_1 \approx_c H_2$.
  - This follows by the LWE assumption!
- Finally, we define hybrid H_3

H_3

$$(\mathbf{M}, \mathbf{t}) \leftarrow \mathcal{L}_{\mathsf{srand}}^{m,n,q}$$
return $(\mathbf{M}, \mathbf{t})$

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

Potential Issue!

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

Potential Issue!

What if $\mathbf{M}_1$ is not invertible over $\mathbb{Z}_q$?

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

Potential Issue!

What if $\mathbf{M}_1$ is not invertible over $\mathbb{Z}_q$?

- Except with low probability, $\mathbf{M}_1$ will be invertible

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

Potential Issue!

What if $\mathbf{M}_1$ is not invertible over $\mathbb{Z}_q$?

- Except with low probability, $\mathbf{M}_1$ will be invertible
  - But the probability of failure is *non-negligible*.

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

Potential Issue!

What if $\mathbf{M}_1$ is not invertible over $\mathbb{Z}_q$?

- Except with low probability, $\mathbf{M}_1$ will be invertible
  - But the probability of failure is *non-negligible*.
  - How do we get negligible failure probability?

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

Potential Issue!

What if $\mathbf{M}_1$ is not invertible over $\mathbb{Z}_q$?

- Except with low probability, $\mathbf{M}_1$ will be invertible
  - But the probability of failure is *non-negligible*.
  - How do we get negligible failure probability?

- Easy fix: replace $(n+m)$ with $(2n+m)$ in H_1 and H_2.

LWE with Short Secrets

- Claim: $H_2 \equiv H_3$.
- The claim follows by the following observations.
  - In H_2, since $\mathbf{t}_2$ is uniform over $\mathbb{Z}_q^m$ and independent of $\mathbf{M}^*$, it acts as a one-time pad.
  - This implies that $\mathbf{t}^*$ is also uniformly distributed over $\mathbb{Z}_q^m$.
  - Thus, $(\mathbf{M}^*, \mathbf{t}^*)$ in H_2 is identically distributed as $\mathcal{L}_{\mathsf{srand}}^{m,n,q}$.

Potential Issue!

What if $\mathbf{M}_1$ is not invertible over $\mathbb{Z}_q$?

- Except with low probability, $\mathbf{M}_1$ will be invertible
  - But the probability of failure is *non-negligible*.
  - How do we get negligible failure probability?

- Easy fix: replace $(n+m)$ with $(2n+m)$ in H_1 and H_2.
  - Then, prune down the $2n \times m$ matrix $\mathbf{M}_1$ to *invertible* $\hat{\mathbf{M}}_1 \in \mathbb{Z}_q^{n \times n}$.

Key Exchange from LWE

Big Picture: DDH Key Exchange

Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.

Big Picture: DDH Key Exchange

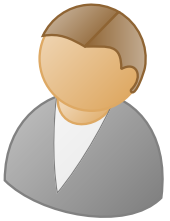
- The intuition behind LWE Key Exchange comes from DDH Key Exchange.

Alice

Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.

Alice



Bob

Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.

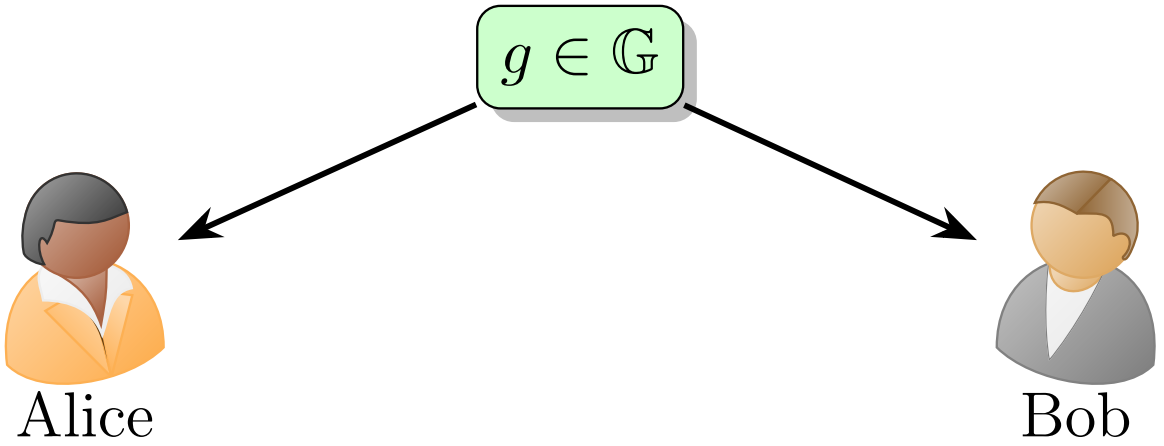
$g \in \mathbb{G}$

Alice

Bob

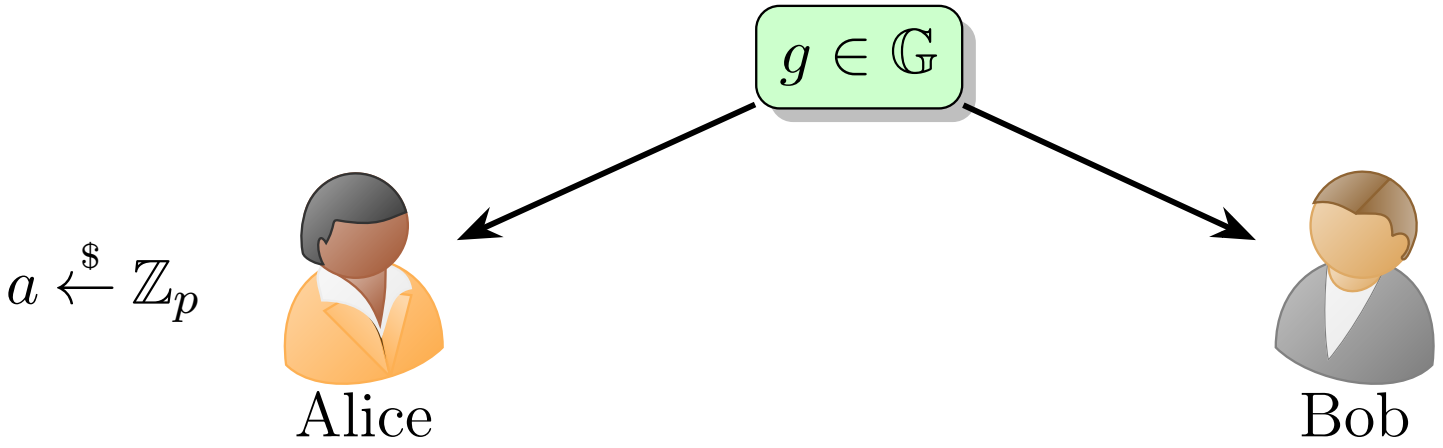
Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



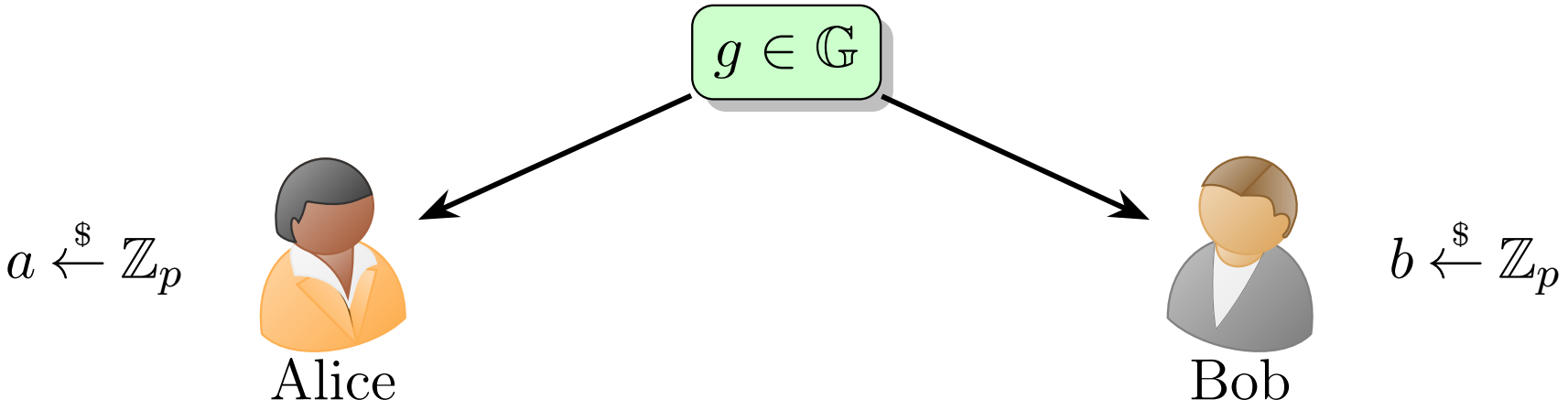
Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



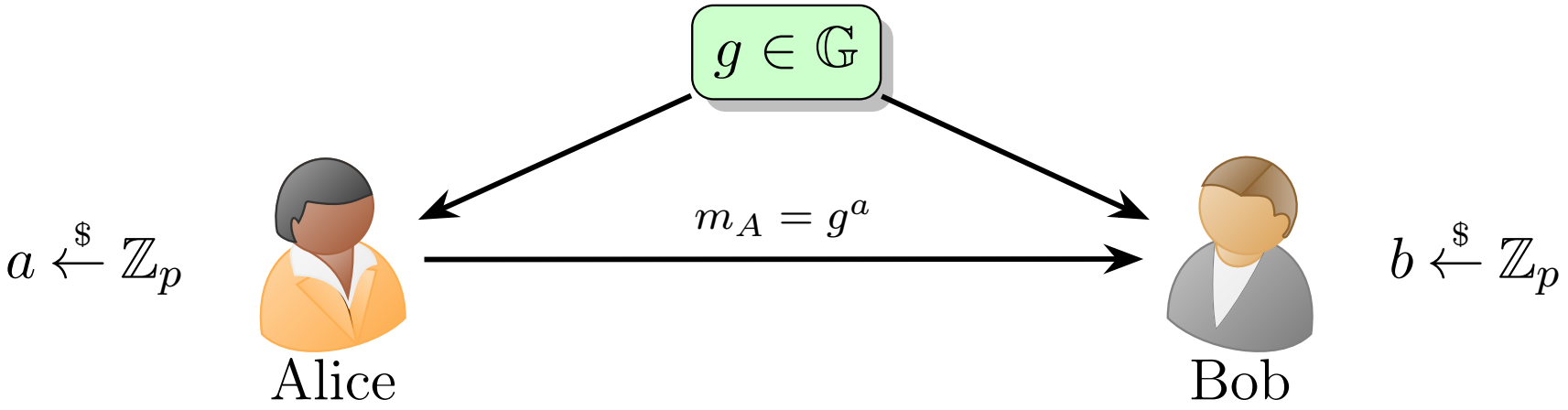
Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



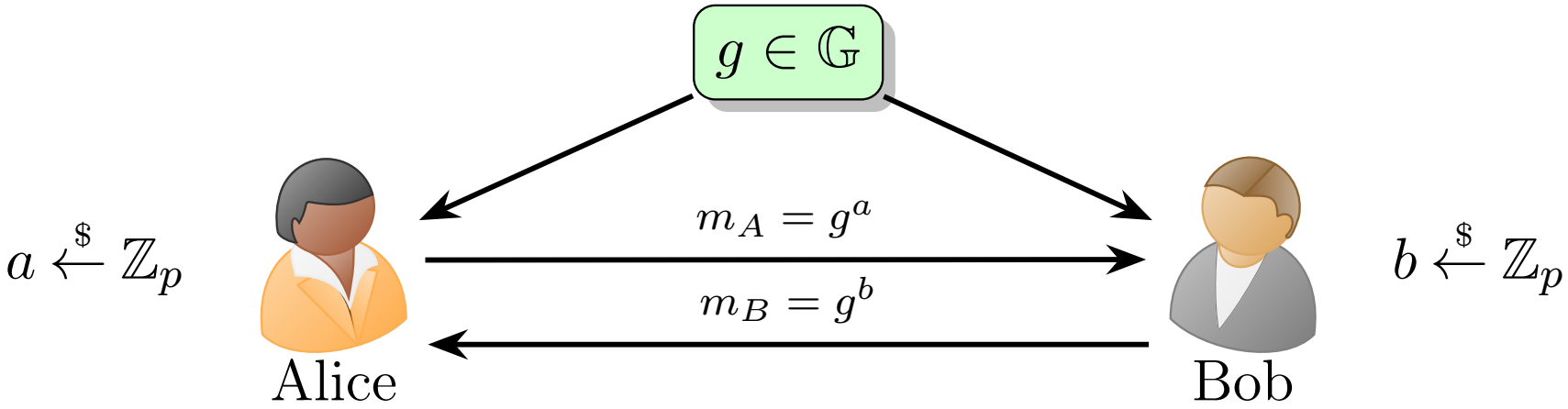
Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



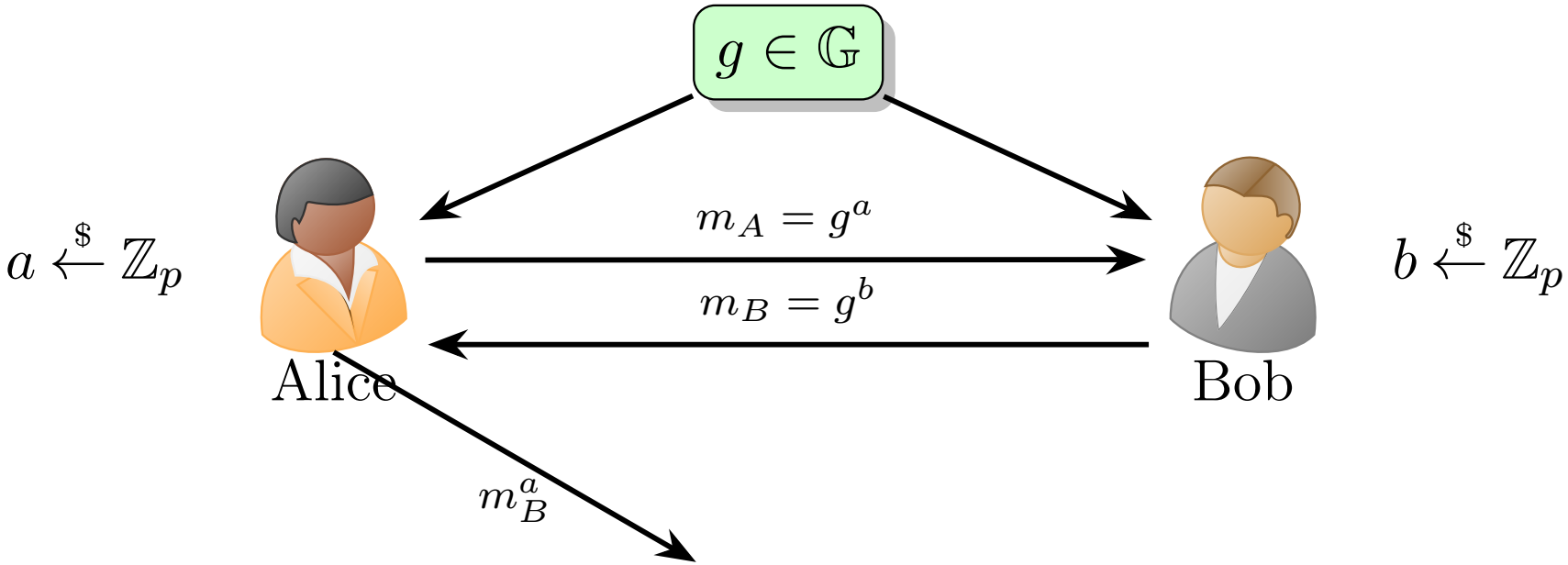
Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



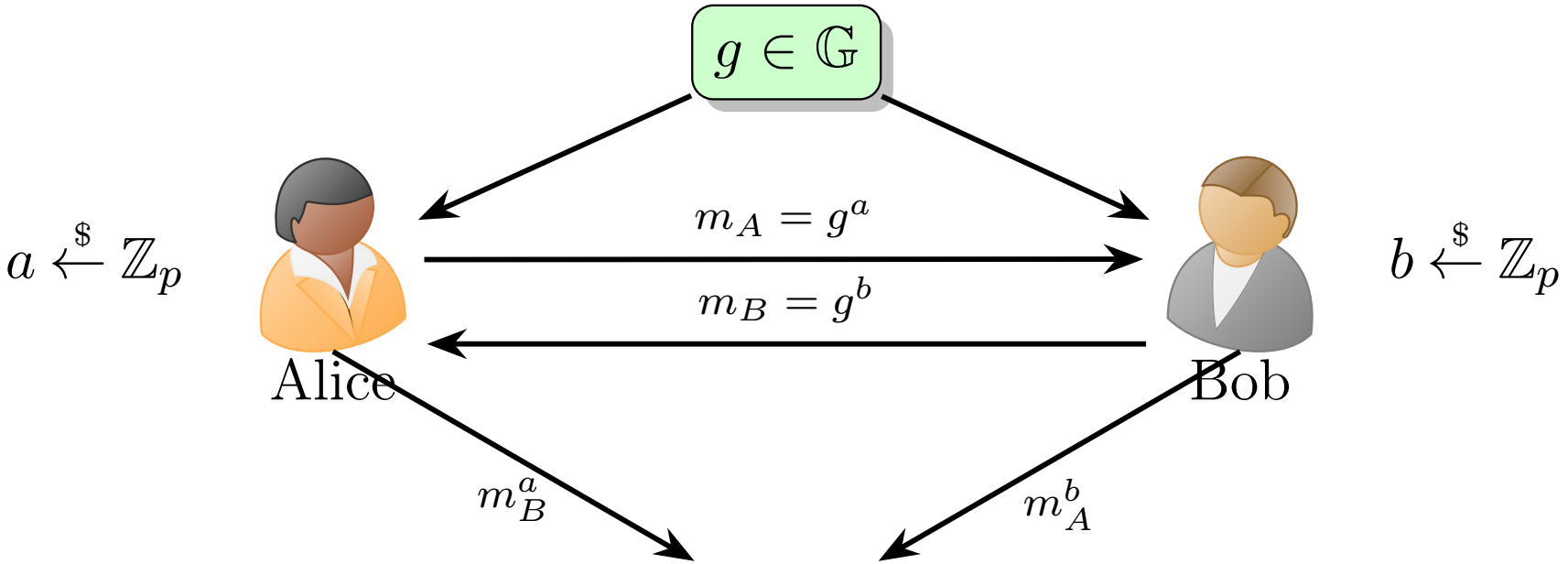
Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



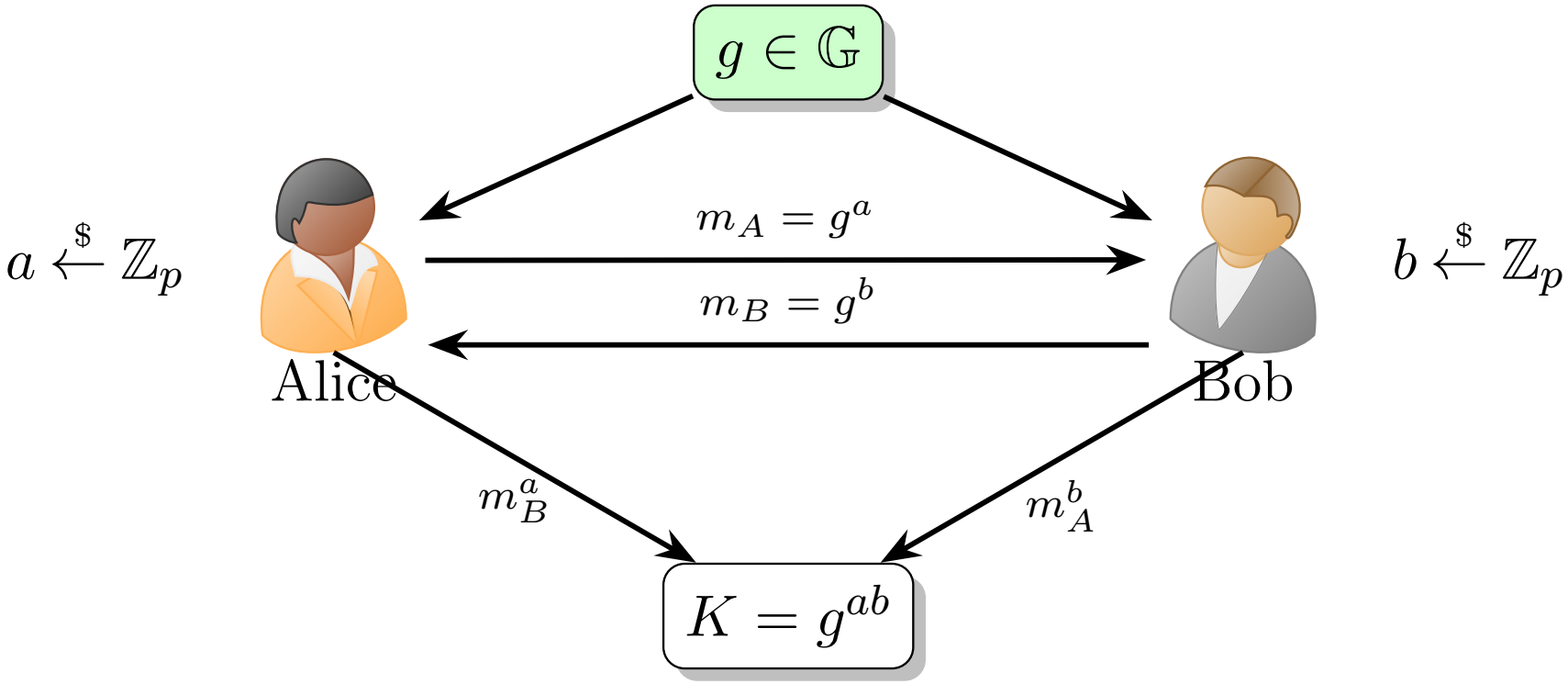
Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



Big Picture: DDH Key Exchange

- The intuition behind LWE Key Exchange comes from DDH Key Exchange.



Motivation: Vector-Matrix-Vector Products

Motivation: Vector-Matrix-Vector Products

- LWE Key Exchange will try to replicate DDH via the following observation about Vector-Matrix-Vector products.

Motivation: Vector-Matrix-Vector Products

- LWE Key Exchange will try to replicate DDH via the following observation about Vector-Matrix-Vector products.

$$\mathbf{a} \cdot \mathbf{G} \cdot \mathbf{b}^\top = (\mathbf{a} \cdot \mathbf{G}) \cdot \mathbf{b}^\top = \mathbf{a} \cdot \left(\mathbf{G} \cdot \mathbf{b}^\top \right)$$

Motivation: Vector-Matrix-Vector Products

- LWE Key Exchange will try to replicate DDH via the following observation about Vector-Matrix-Vector products.

$$\mathbf{a} \cdot \mathbf{G} \cdot \mathbf{b}^\top = (\mathbf{a} \cdot \mathbf{G}) \cdot \mathbf{b}^\top = \mathbf{a} \cdot \left(\mathbf{G} \cdot \mathbf{b}^\top \right)$$

- Here, $\mathbf{G}$ would be a public matrix over $\mathbb{Z}_q$, $\mathbf{a}$ is a vector sampled by Alice, and $\mathbf{b}$ is a vector sampled by Bob.

Motivation: Vector-Matrix-Vector Products

- LWE Key Exchange will try to replicate DDH via the following observation about Vector-Matrix-Vector products.

$$\mathbf{a} \cdot \mathbf{G} \cdot \mathbf{b}^\top = (\mathbf{a} \cdot \mathbf{G}) \cdot \mathbf{b}^\top = \mathbf{a} \cdot \left(\mathbf{G} \cdot \mathbf{b}^\top \right)$$

- Here, $\mathbf{G}$ would be a public matrix over $\mathbb{Z}_q$, $\mathbf{a}$ is a vector sampled by Alice, and $\mathbf{b}$ is a vector sampled by Bob.

- Building inspiration from DDH, we obtain the following protocol with this idea.

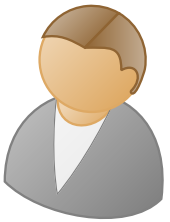
Matrix Key Exchange

Matrix Key Exchange

Alice

Matrix Key Exchange

Alice

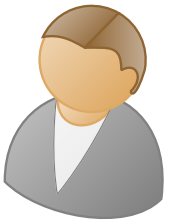


Bob

Matrix Key Exchange

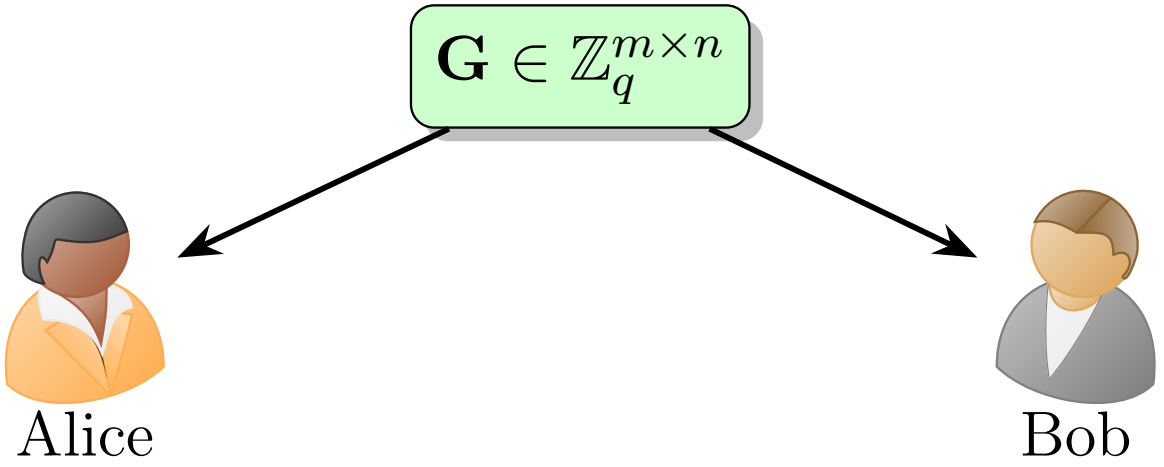
$$\mathbf{G} \in \mathbb{Z}_q^{m \times n}$$

Alice

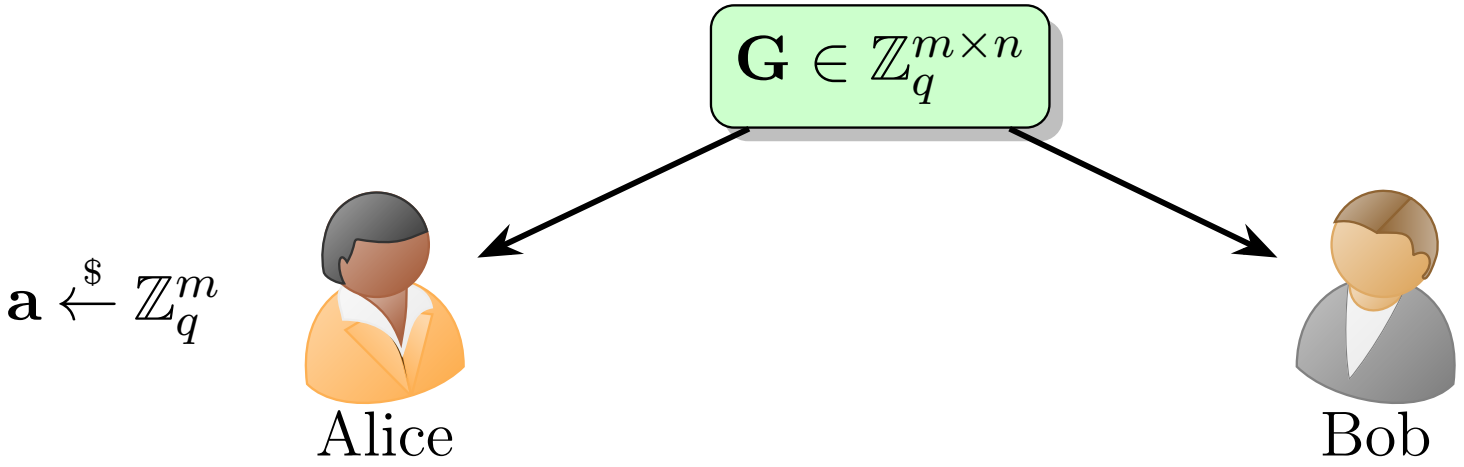


Bob

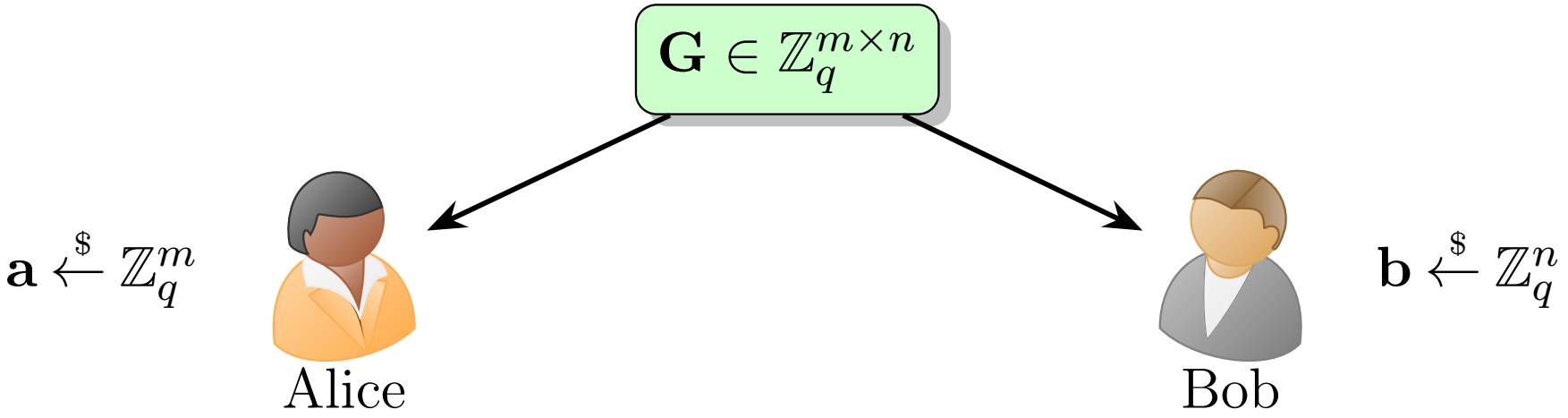
Matrix Key Exchange



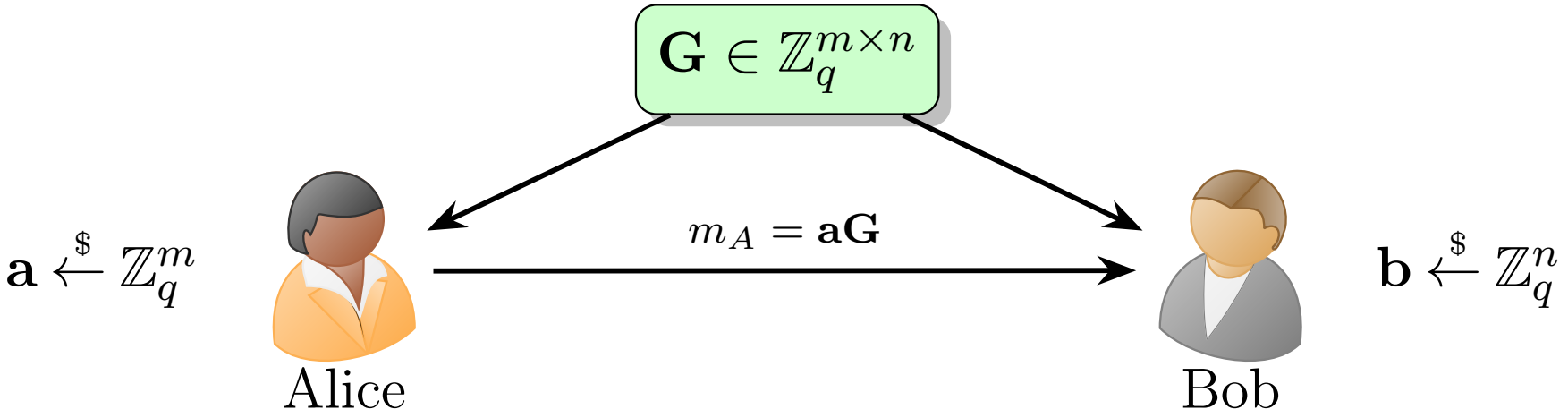
Matrix Key Exchange



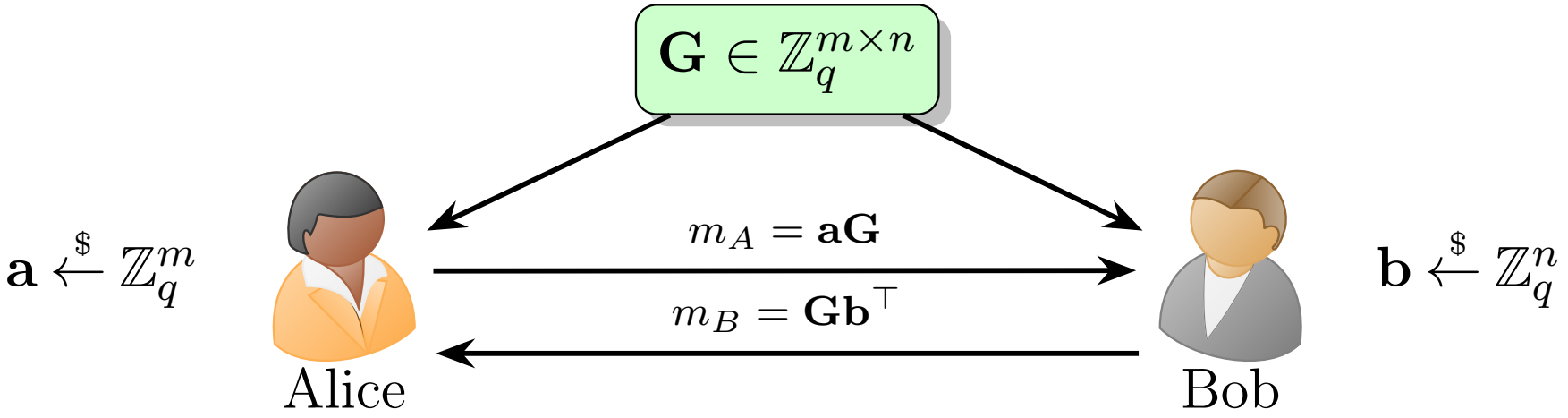
Matrix Key Exchange



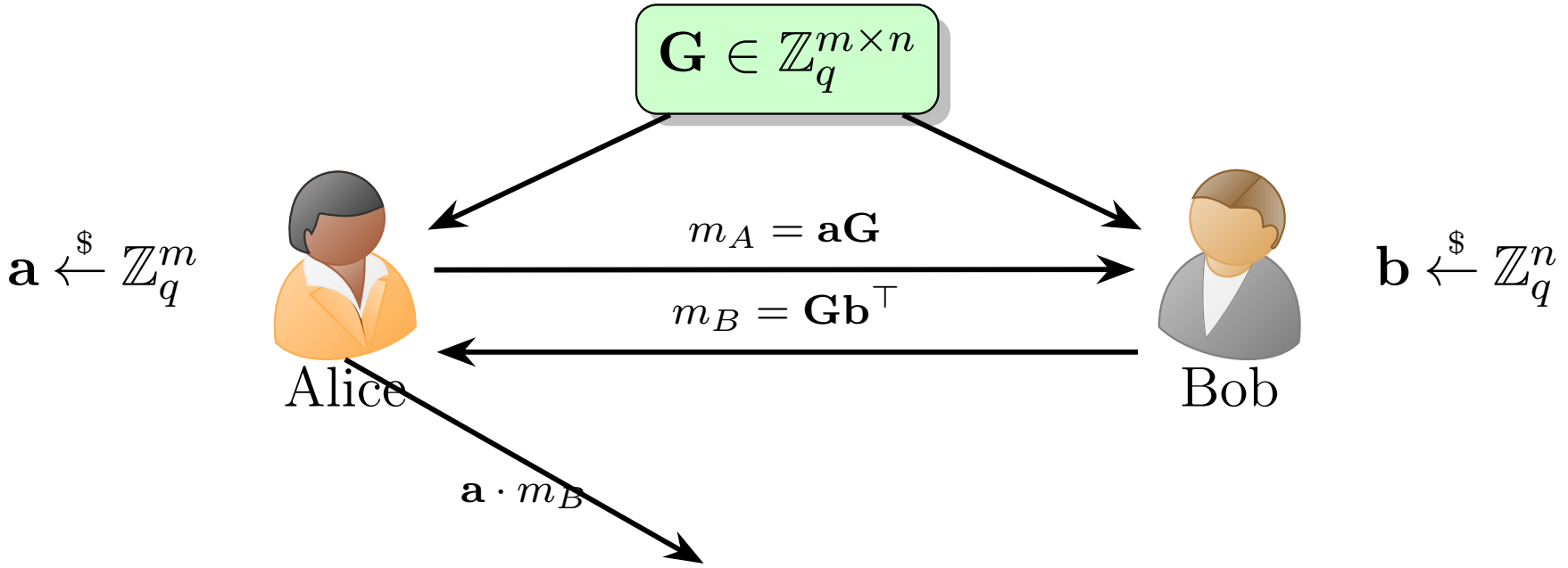
Matrix Key Exchange



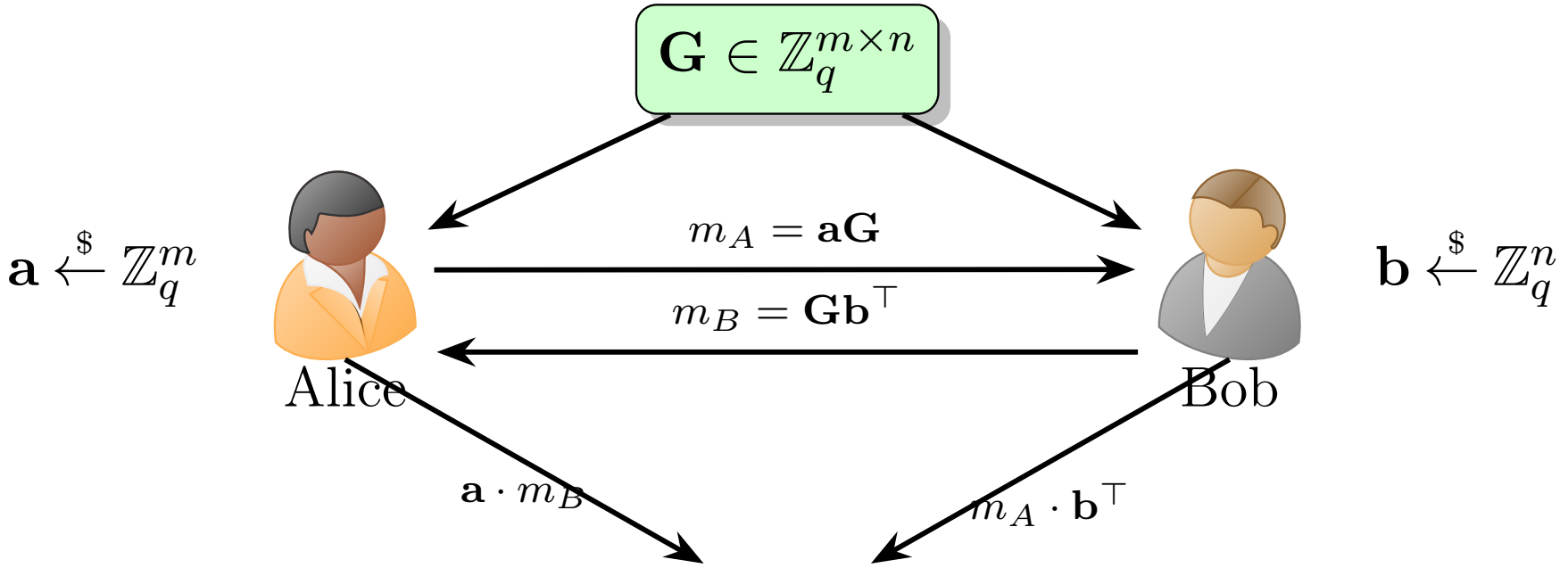
Matrix Key Exchange



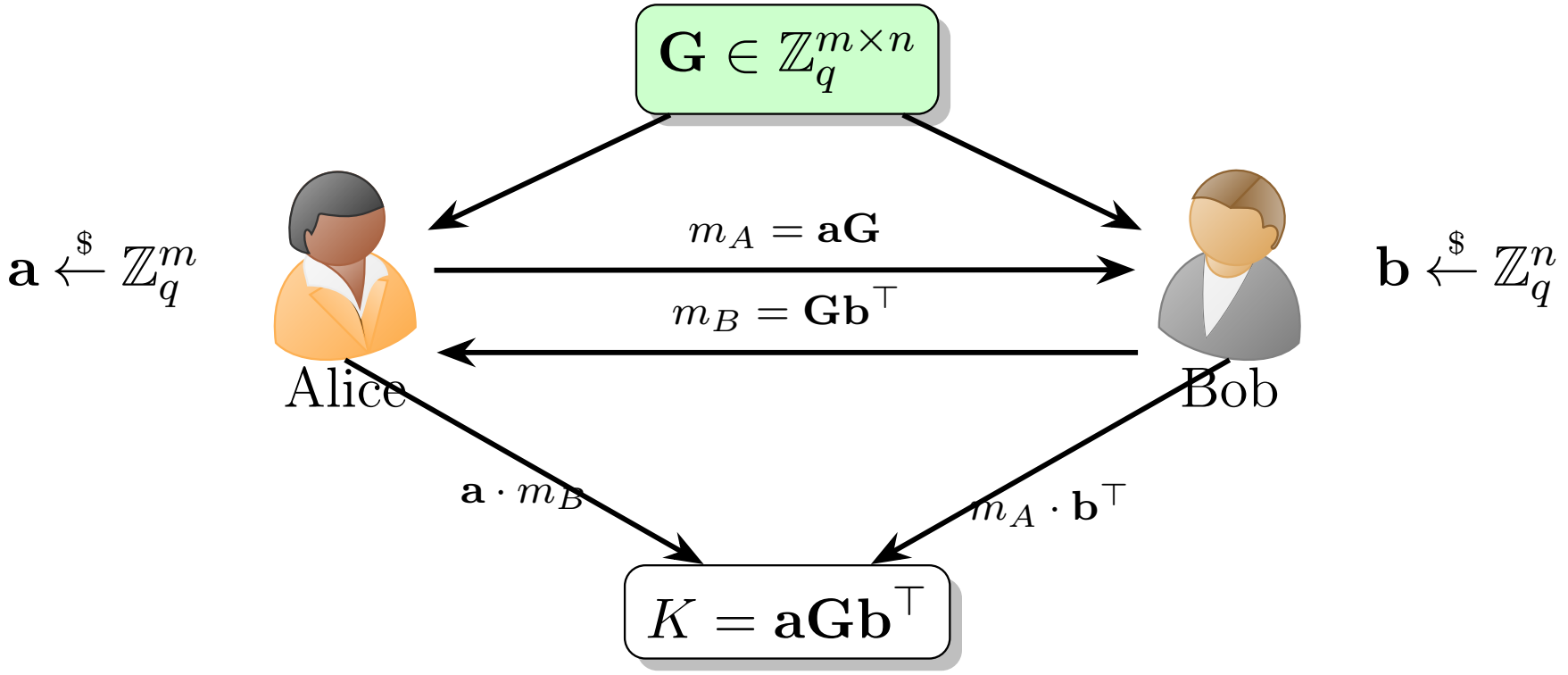
Matrix Key Exchange



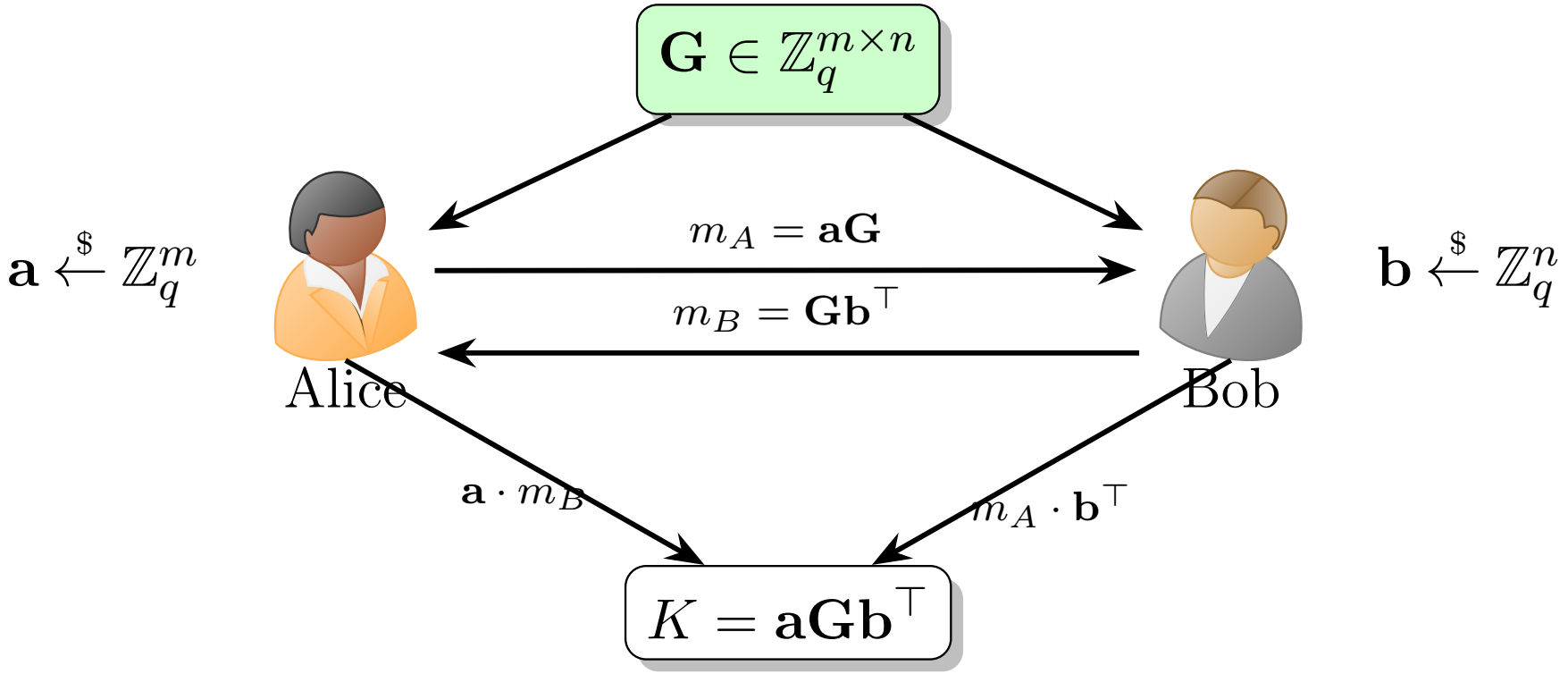
Matrix Key Exchange



Matrix Key Exchange

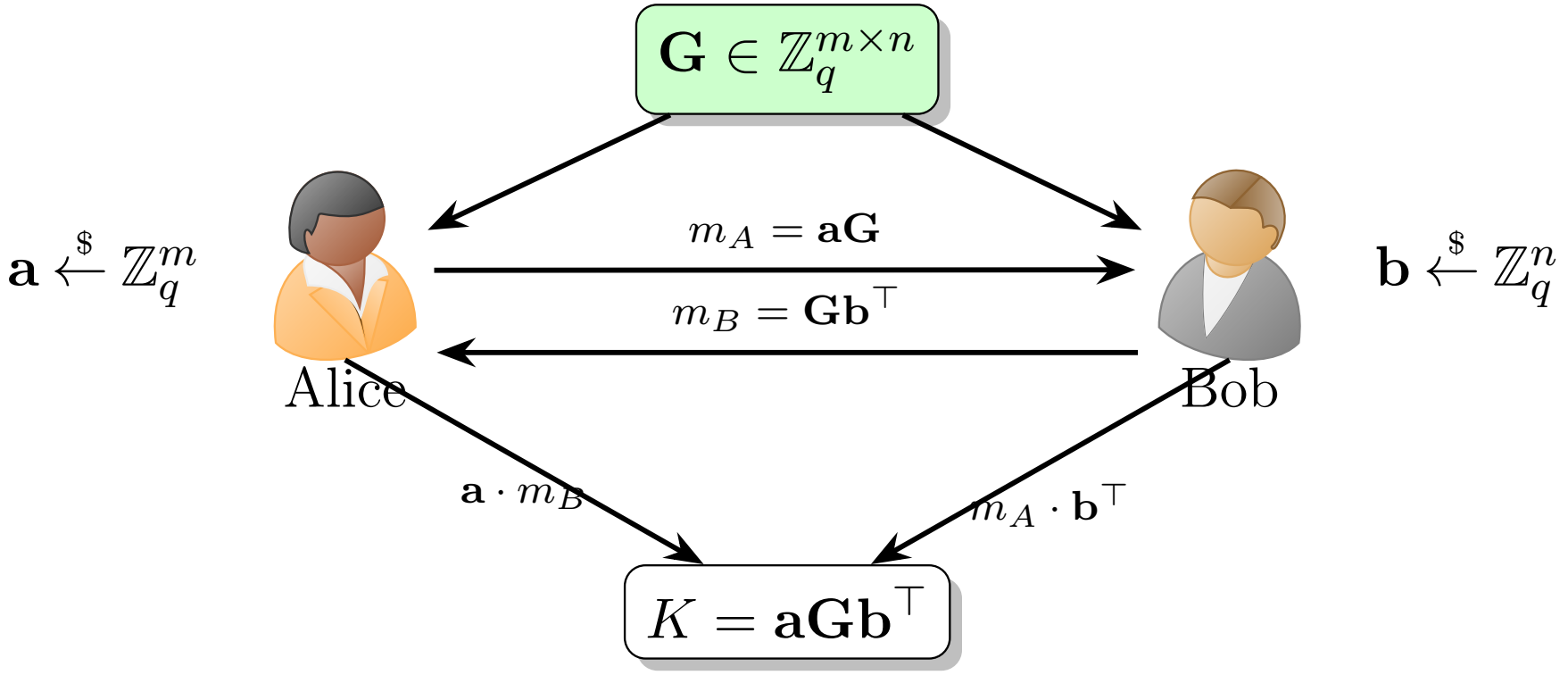


Matrix Key Exchange



Huge Issue!

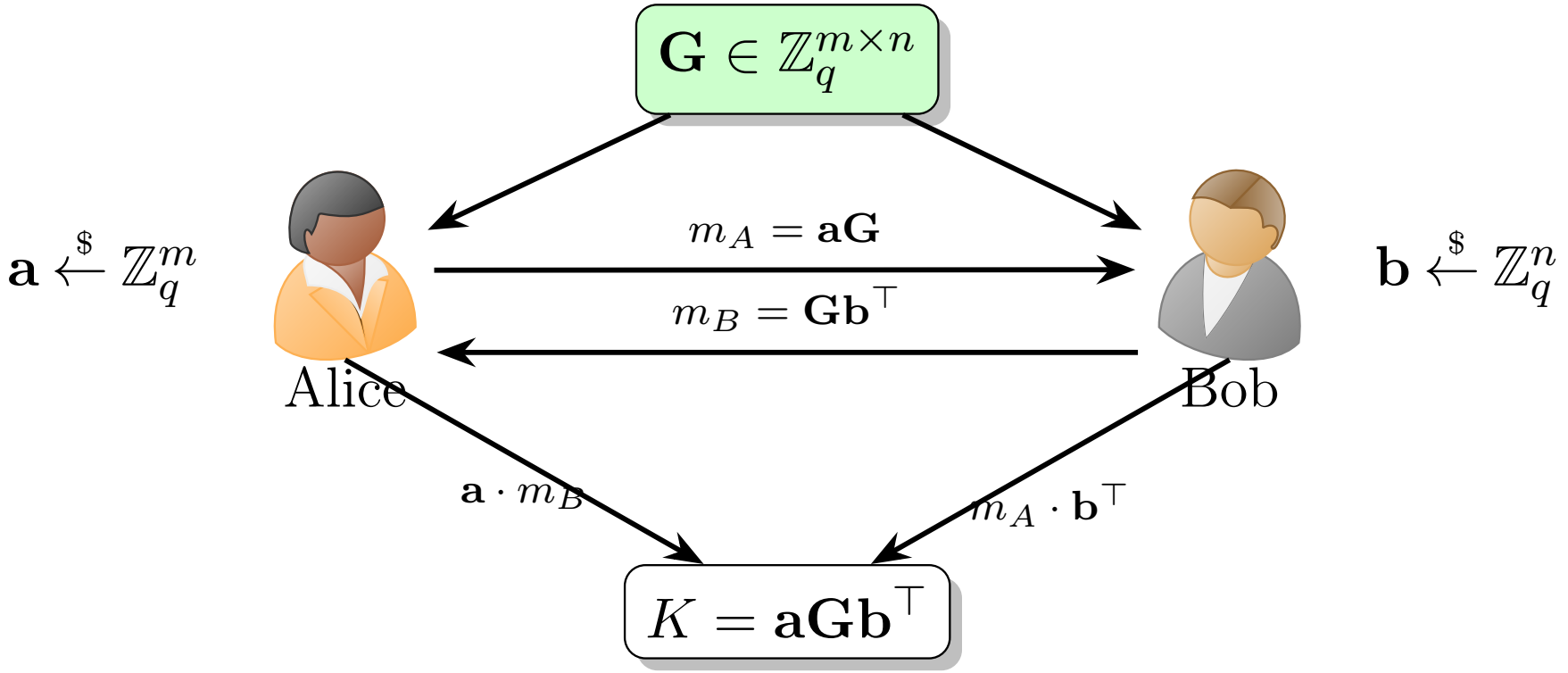
Matrix Key Exchange



Huge Issue!

Scheme is not secure!

Matrix Key Exchange



Huge Issue!

Scheme is not secure!

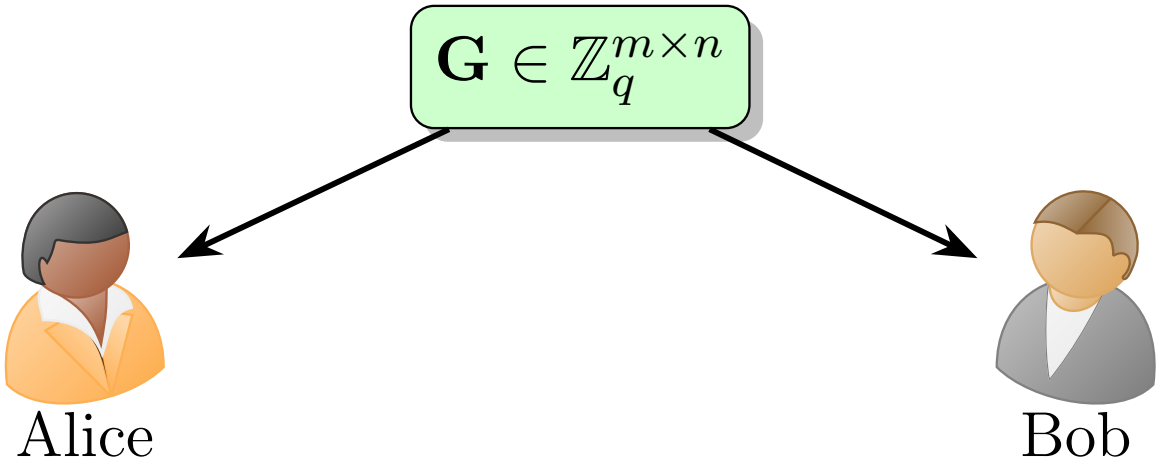
- m_A and m_B do not hide the secret vectors of Alice and Bob!

LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.

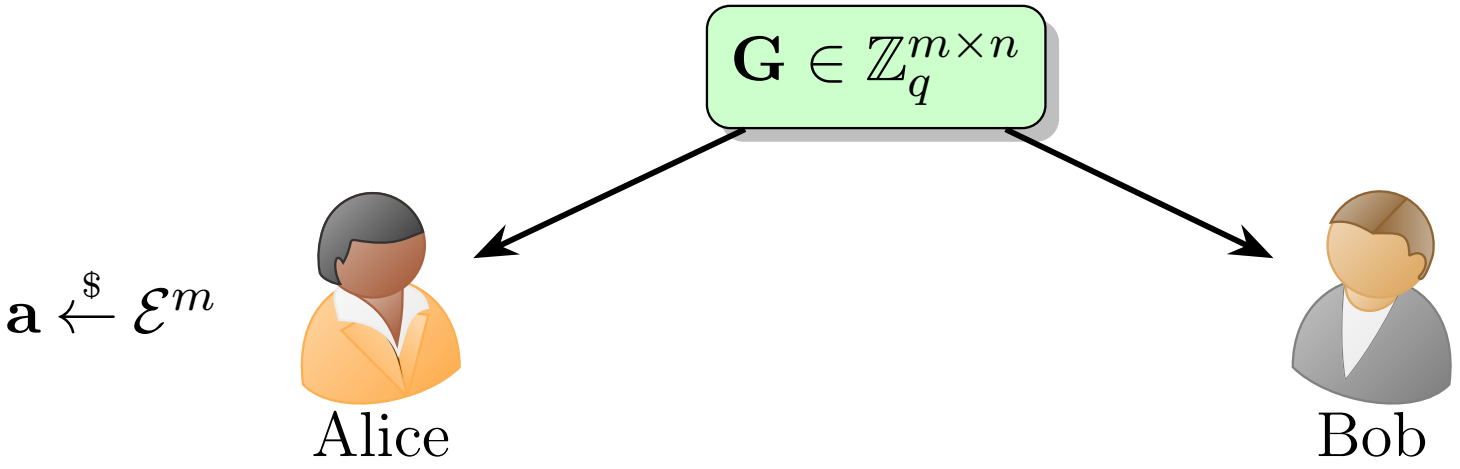
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



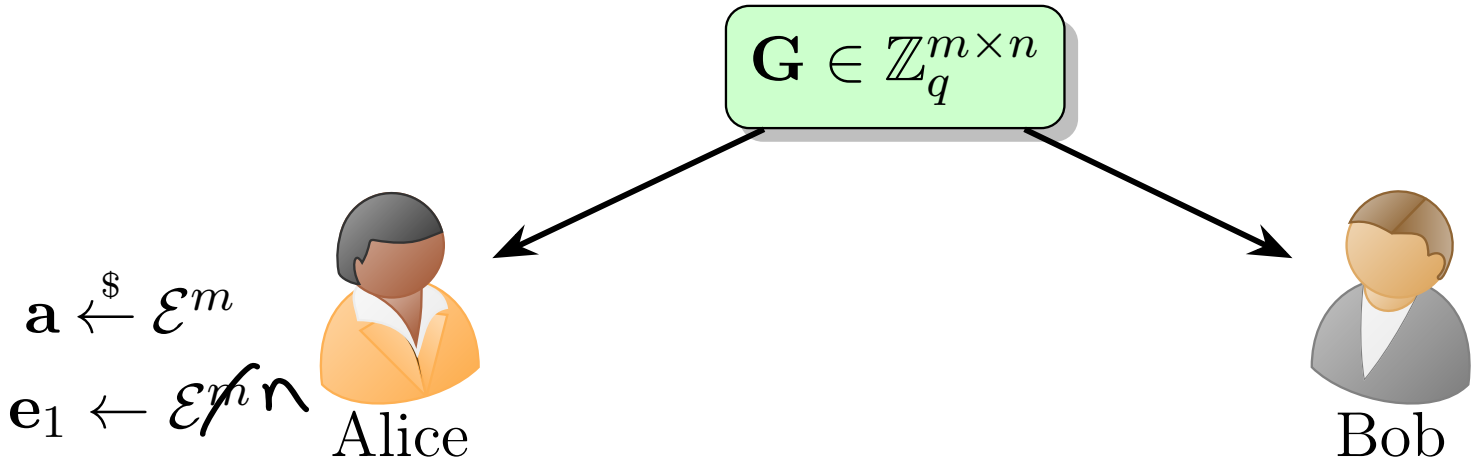
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



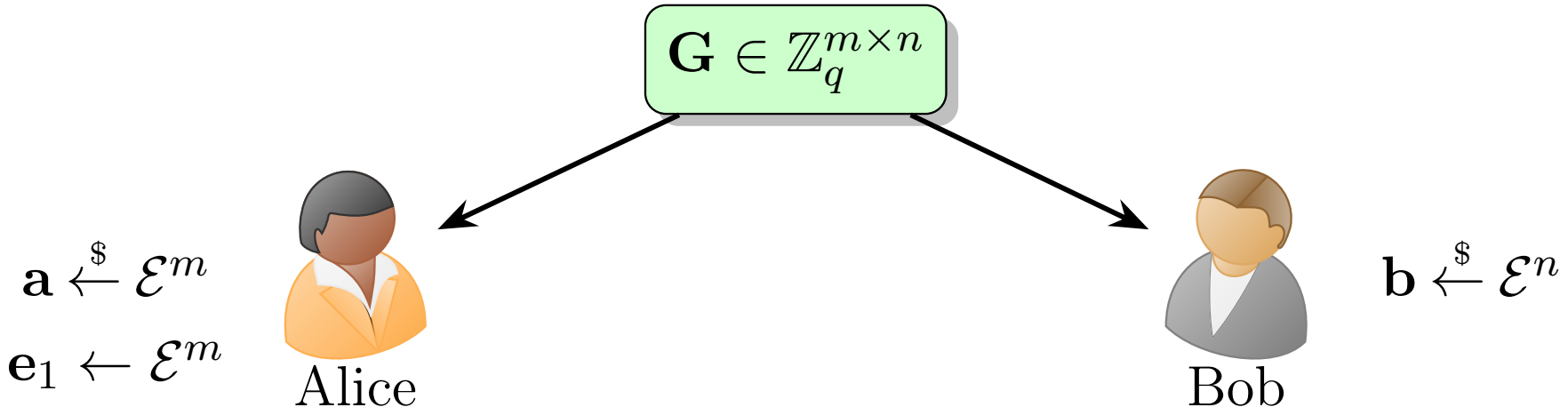
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



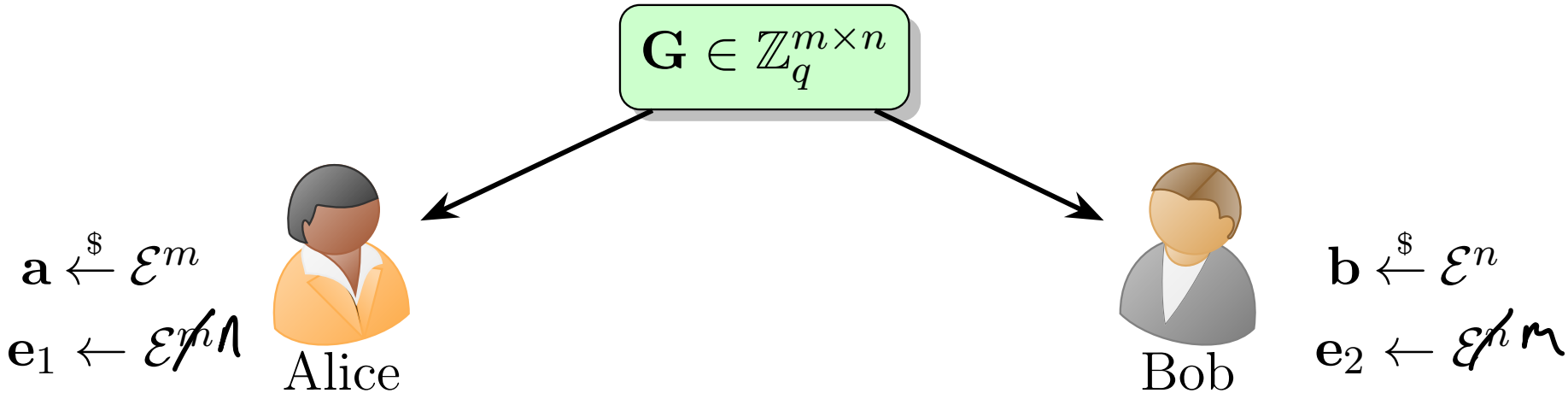
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



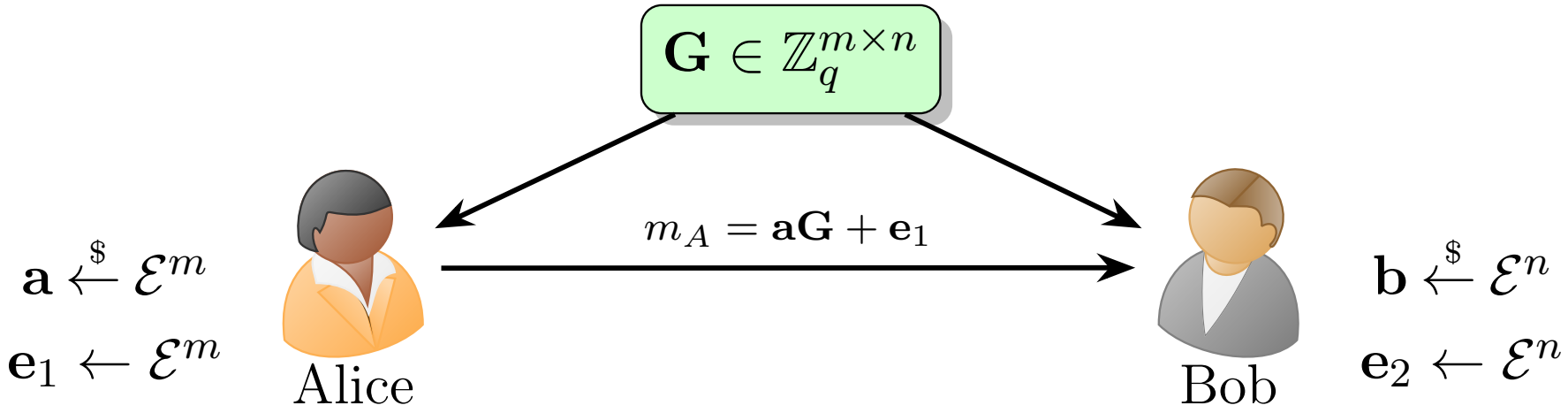
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



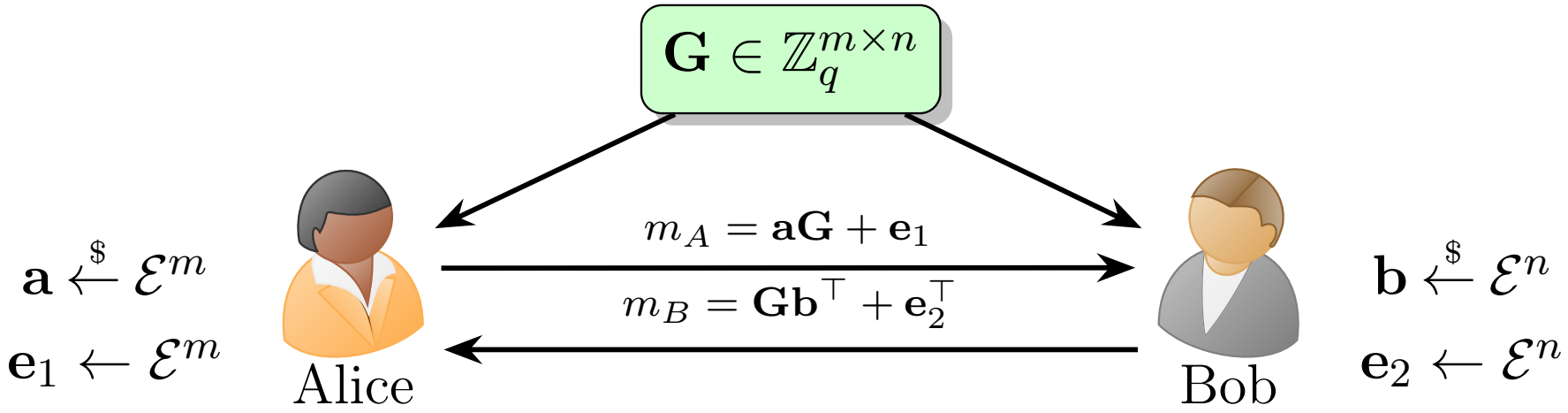
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



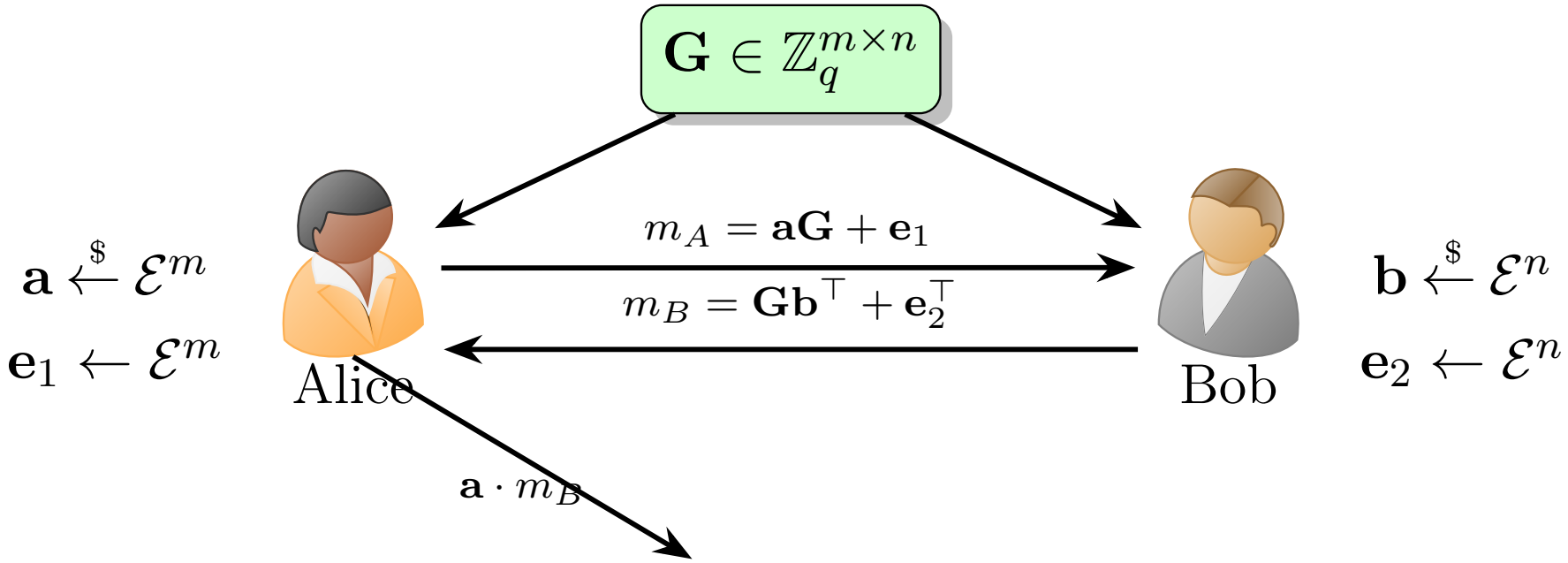
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



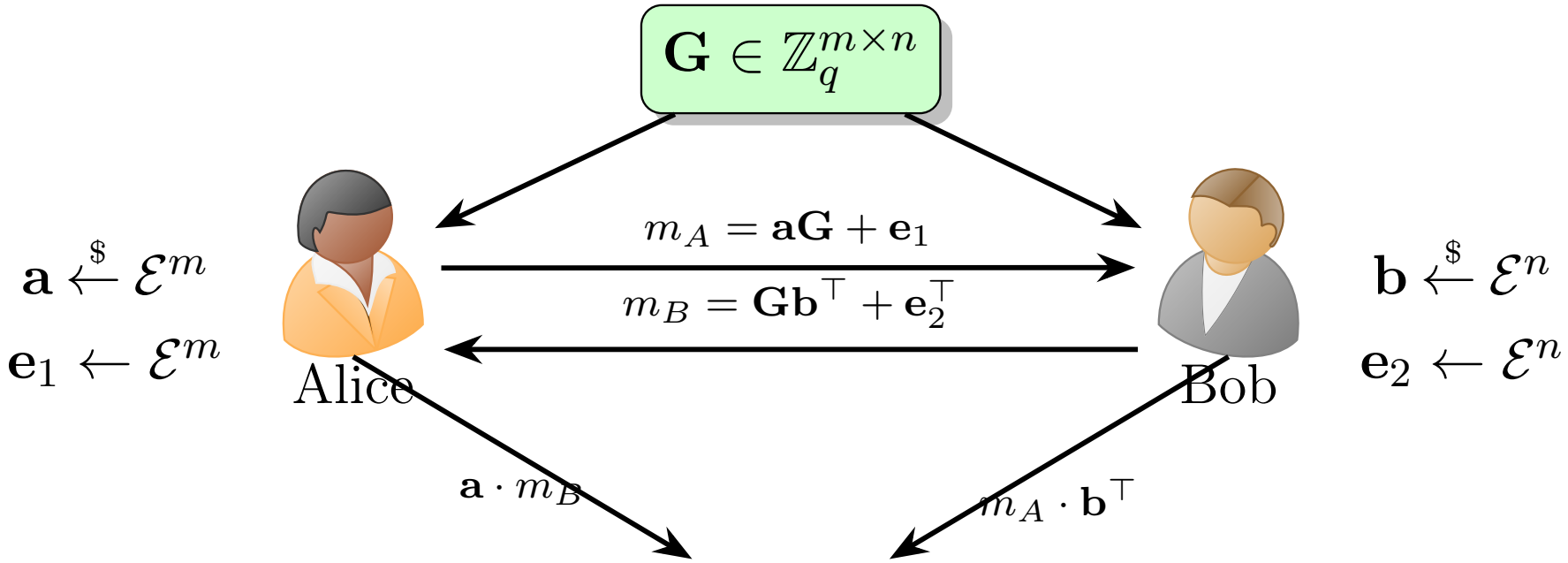
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



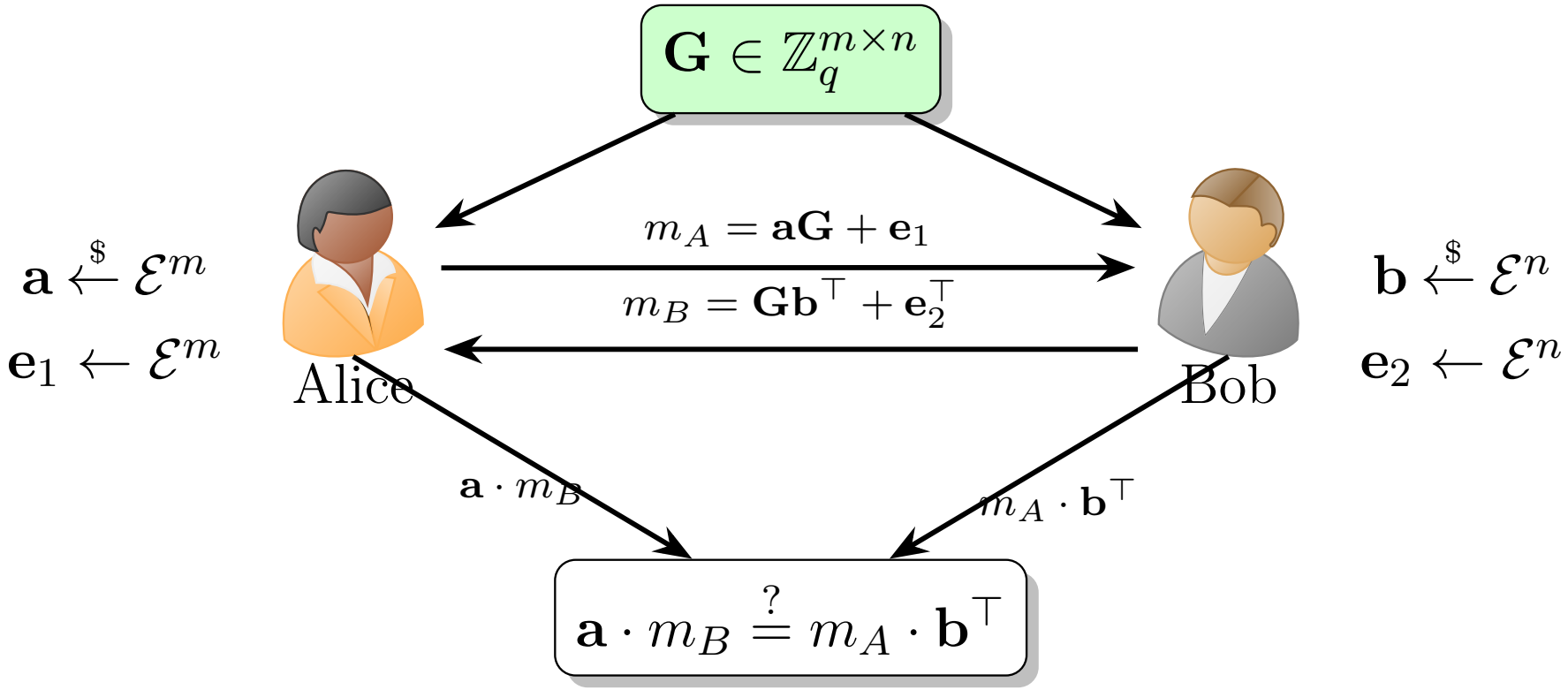
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



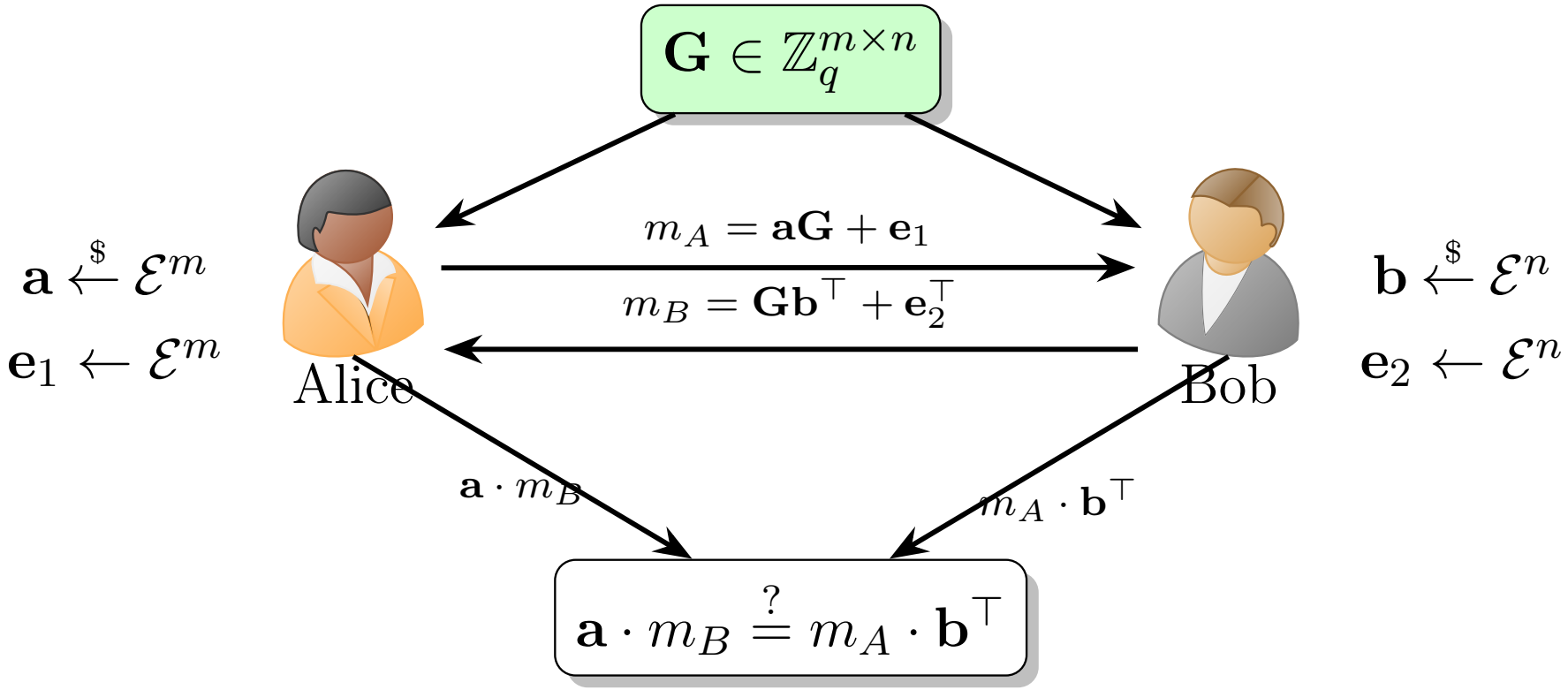
LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



LWE Key Exchange

- To fix this, let's modify the scheme by using the short LWE assumption.



- Issue: cannot send the error vectors, otherwise security is trivially broken!

LWE Key Exchange

- Let's take a closer look at what each party outputs.

LWE Key Exchange

- Let's take a closer look at what each party outputs.
- Alice's output:

LWE Key Exchange

- Let's take a closer look at what each party outputs.
- Alice's output:

$$\mathbf{a} \cdot m_B = \mathbf{a}\left(\mathbf{G}\mathbf{b}^\top + \mathbf{e}_2^\top\right) = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{a}\mathbf{e}_2^\top.$$

noise

LWE Key Exchange

- Let's take a closer look at what each party outputs.
- Alice's output:

$$\mathbf{a} \cdot m_B = \mathbf{a}\left(\mathbf{G}\mathbf{b}^\top + \mathbf{e}_2^\top\right) = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{a}\mathbf{e}_2^\top.$$

- Bob's output:

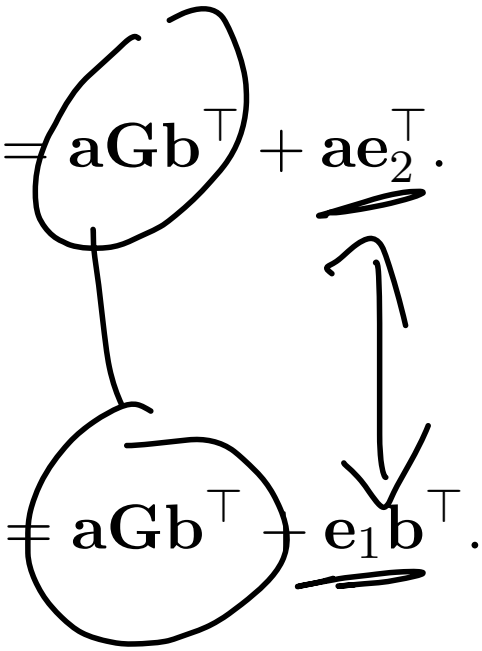
LWE Key Exchange

- Let's take a closer look at what each party outputs.
- Alice's output:

$$\mathbf{a} \cdot m_B = \mathbf{a}\left(\mathbf{G}\mathbf{b}^\top + \mathbf{e}_2^\top\right) = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{a}\mathbf{e}_2^\top.$$

- Bob's output:

$$m_A \cdot \mathbf{b}^\top = (\mathbf{a}\mathbf{G} + \mathbf{e}_1) \cdot \mathbf{b}^\top = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{e}_1\mathbf{b}^\top.$$



LWE Key Exchange

- Let's take a closer look at what each party outputs.
- Alice's output:

$$\mathbf{a} \cdot m_B = \mathbf{a}\left(\mathbf{G}\mathbf{b}^\top + \mathbf{e}_2^\top\right) = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{a}\mathbf{e}_2^\top.$$

- Bob's output:

$$m_A \cdot \mathbf{b}^\top = (\mathbf{a}\mathbf{G} + \mathbf{e}_1) \cdot \mathbf{b}^\top = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{e}_1\mathbf{b}^\top.$$

- We can calculate the difference between these two outputs as:

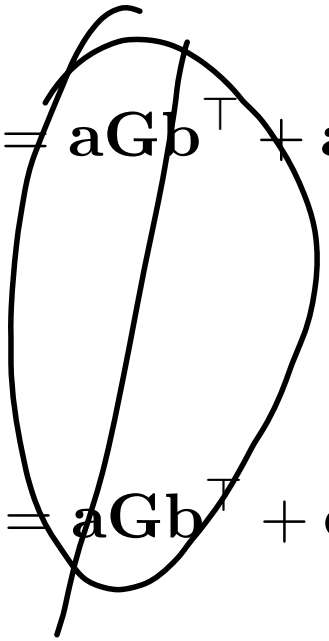
LWE Key Exchange

- Let's take a closer look at what each party outputs.
- Alice's output:

$$\mathbf{a} \cdot m_B = \mathbf{a}\left(\mathbf{G}\mathbf{b}^\top + \mathbf{e}_2^\top\right) = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{a}\mathbf{e}_2^\top.$$

- Bob's output:

$$m_A \cdot \mathbf{b}^\top = (\mathbf{a}\mathbf{G} + \mathbf{e}_1) \cdot \mathbf{b}^\top = \mathbf{a}\mathbf{G}\mathbf{b}^\top + \mathbf{e}_1\mathbf{b}^\top.$$



- We can calculate the difference between these two outputs as:

$$\epsilon := \left|\mathbf{a}m_B - m_A\mathbf{b}^\top\right| = \left|\mathbf{a}\mathbf{e}_2^\top - \mathbf{e}_1\mathbf{b}^\top\right|.$$

LWE Key Exchange: Reconciliation

- Notice: $(\mathbf{a}, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2)$ are all *short vectors* sampled from the noise distribution.

LWE Key Exchange: Reconciliation

- Notice: $(\mathbf{a}, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2)$ are all *short vectors* sampled from the noise distribution.
- Therefore, if we appropriately define $\mathcal{E}$, we can ensure that $\Pr[\epsilon \geqslant q/4] = \mathsf{negl}(\lambda)$.

LWE Key Exchange: Reconciliation

- Notice: $(\mathbf{a}, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2)$ are all *short vectors* sampled from the noise distribution.
- Therefore, if we appropriately define $\mathcal{E}$, we can ensure that $\Pr[\epsilon \geqslant q/4] = \mathsf{negl}(\lambda)$.
- So long as $\epsilon < q/4$, Alice and Bob can perform *reconciliation*.

LWE Key Exchange: Reconciliation

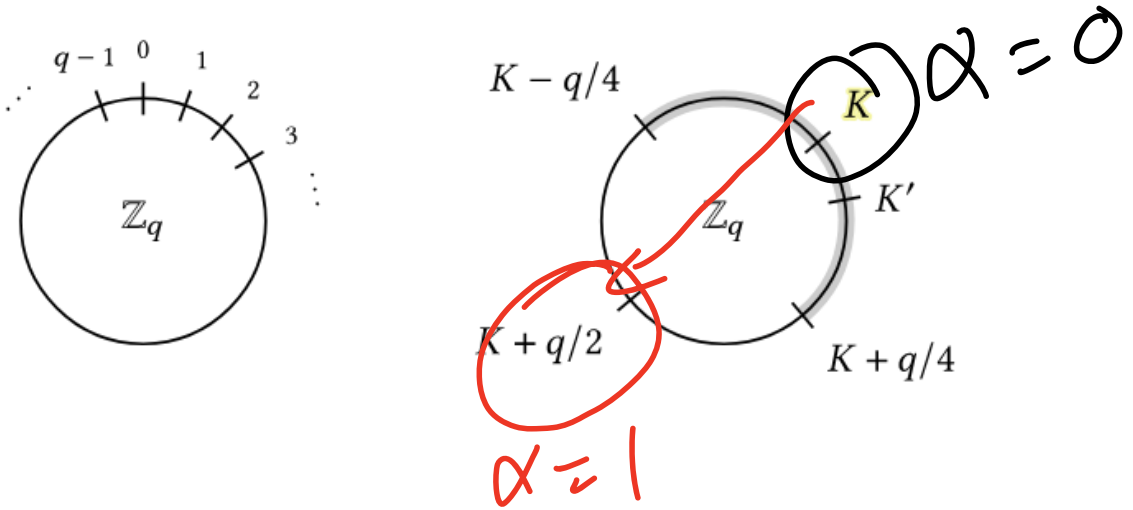
- Notice: $(\mathbf{a}, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2)$ are all *short vectors* sampled from the noise distribution.
- Therefore, if we appropriately define $\mathcal{E}$, we can ensure that $\Pr[\epsilon \geqslant q/4] = \mathsf{negl}(\lambda)$.
- So long as $\epsilon < q/4$, Alice and Bob can perform *reconciliation*.
  - Let $K := \mathbf{a} m_B$ (Alice's value) and $K' = m_A \mathbf{b}^\top$ (Bob's value).

LWE Key Exchange: Reconciliation

- Notice: $(\mathbf{a}, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2)$ are all *short vectors* sampled from the noise distribution.
- Therefore, if we appropriately define $\mathcal{E}$, we can ensure that $\Pr[\epsilon \geqslant q/4] = \mathsf{negl}(\lambda)$.
- So long as $\epsilon < q/4$, Alice and Bob can perform *reconciliation*.
  - Let $K := \mathbf{a} m_B$ (Alice's value) and $K' = m_A \mathbf{b}^\top$ (Bob's value).
  - Though Alice doesn't know K', Alice *does know* it lies within the shaded region in the following diagram:

LWE Key Exchange: Reconciliation

- Notice: $(\mathbf{a}, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2)$ are all *short vectors* sampled from the noise distribution.
- Therefore, if we appropriately define $\mathcal{E}$, we can ensure that $\Pr[\epsilon \geqslant q/4] = \mathsf{negl}(\lambda)$.
- So long as $\epsilon < q/4$, Alice and Bob can perform *reconciliation*.
  - Let $K := \mathbf{a} m_B$ (Alice's value) and $K' = m_A \mathbf{b}^\top$ (Bob's value).
  - Though Alice doesn't know K', Alice *does know* it lies within the shaded region in the following diagram:



LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0, 1\}$.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0, 1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0, 1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0,1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.
  - Alice and Bob respectively output α, β.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0,1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.
  - Alice and Bob respectively output α, β.
- Notice that except with negligible probability, we have $\alpha = \beta$.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0,1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.
  - Alice and Bob respectively output α, β.

- Notice that except with negligible probability, we have $\alpha = \beta$.
- Why does this work?

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0,1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.
  - Alice and Bob respectively output α, β.

- Notice that except with negligible probability, we have $\alpha = \beta$.
- Why does this work?
  - If $\alpha = 0$, then $R = K$, and by assumption $|K - K'| \leqslant q/4$ (except with negligible probability).

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0,1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.
  - Alice and Bob respectively output α, β.

- Notice that except with negligible probability, we have $\alpha = \beta$.
- Why does this work?
  - If $\alpha = 0$, then $R = K$, and by assumption $|K - K'| \leqslant q/4$ (except with negligible probability). In this case, Bob outputs $\beta = 0$.

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0,1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.
  - Alice and Bob respectively output α, β.

- Notice that except with negligible probability, we have $\alpha = \beta$.
- Why does this work?
  - If $\alpha = 0$, then $R = K$, and by assumption $|K - K'| \leqslant q/4$ (except with negligible probability). In this case, Bob outputs $\beta = 0$.
  - If $\alpha = 1$, then $R = K + \lfloor q/2 \rfloor$, and so $|K - K'| \gg q/4$ (except with negligible probability).

LWE Key Exchange: Reconciliation

- Alice and Bob will both use this fact and intuition to agree upon a (very short) key.
- The following round of communication is added to the Key Exchange protocol we defined before.
  - Alice samples a bit $\alpha \xleftarrow{\$} \{0,1\}$.
  - Alice defines $R := (K + \alpha \lfloor q/2 \rfloor) \bmod q$ and sends R to Bob.
  - Bob computes bit β as 0 if $|K' - R| < q/4$ and 1 otherwise.
  - Alice and Bob respectively output α, β.

- Notice that except with negligible probability, we have $\alpha = \beta$.
- Why does this work?
  - If $\alpha = 0$, then $R = K$, and by assumption $|K - K'| \leqslant q/4$ (except with negligible probability). In this case, Bob outputs $\beta = 0$.
  - If $\alpha = 1$, then $R = K + \lfloor q/2 \rfloor$, and so $|K - K'| \gg q/4$ (except with negligible probability). In this case, Bob outputs $\beta = 1$.

LWE Key Exchange

Lemma 2

LWE Key Exchange

Lemma 2

The defined key exchange protocol with reconciliation is passively secure if the LWE assumption holds.

LWE Key Exchange

Lemma 2

The defined key exchange protocol with reconciliation is passively secure if the LWE assumption holds.

Efficiency?

LWE Key Exchange

Lemma 2

The defined key exchange protocol with reconciliation is passively secure if the LWE assumption holds.

Efficiency?

The careful listener will notice this is *incredibly inefficient*.

LWE Key Exchange

Lemma 2

The defined key exchange protocol with reconciliation is passively secure if the LWE assumption holds.

Efficiency?

The careful listener will notice this is *incredibly inefficient*.

- To get a λ-bit secret key, one would need to repeat the protocol (in parallel) λ times.

LWE Key Exchange

Lemma 2

The defined key exchange protocol with reconciliation is passively secure if the LWE assumption holds.

Efficiency?

The careful listener will notice this is *incredibly inefficient*.

- To get a λ-bit secret key, one would need to repeat the protocol (in parallel) λ times.
- Horribly expensive.

LWE Key Exchange in Practice

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

1. Matrix-matrix-matrix multiplication versus vector-matrix-vector multiplication.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

1. Matrix-matrix-matrix multiplication versus vector-matrix-vector multiplication.
   - Our above protocol utilizes a vector-matrix-vector product, resulting in a single scalar being computed by each party.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

1. Matrix-matrix-matrix multiplication versus vector-matrix-vector multiplication.
   - Our above protocol utilizes a vector-matrix-vector product, resulting in a single scalar being computed by each party.
   - In practice, this is extended to a matrix-matrix-matrix product.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

1. Matrix-matrix-matrix multiplication versus vector-matrix-vector multiplication.
   - Our above protocol utilizes a vector-matrix-vector product, resulting in a single scalar being computed by each party.
   - In practice, this is extended to a matrix-matrix-matrix product.
   - This will allow Alice and Bob to approximately agree on an entire matrix of values.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.
   - Ring LWE is a variant of the LWE assumption which assumes additional structure on the (otherwise random) matrix $\mathbf{G}$.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.
   - Ring LWE is a variant of the LWE assumption which assumes additional structure on the (otherwise random) matrix $\mathbf{G}$.
   - Roughly speaking, ring LWE assumed that a matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ is composed of several $n \times n$ block matrices $\mathbf{G}_1, \ldots, \mathbf{G}_k$, where $m = kn$.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.
   - Ring LWE is a variant of the LWE assumption which assumes additional structure on the (otherwise random) matrix $\mathbf{G}$.
   - Roughly speaking, ring LWE assumed that a matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ is composed of several $n \times n$ block matrices $\mathbf{G}_1, \ldots, \mathbf{G}_k$, where $m = kn$.
   - Then, for each $\mathbf{G}_i$, the first row is sampled uniformly at random as $(a_1, \ldots, a_n)$, and then each row $j > 1$ is defined to be:

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.
   - Ring LWE is a variant of the LWE assumption which assumes additional structure on the (otherwise random) matrix $\mathbf{G}$.
   - Roughly speaking, ring LWE assumed that a matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ is composed of several $n \times n$ block matrices $\mathbf{G}_1, \ldots, \mathbf{G}_k$, where $m = kn$.
   - Then, for each $\mathbf{G}_i$, the first row is sampled uniformly at random as $(a_1, \ldots, a_n)$, and then each row $j > 1$ is defined to be:

$$(-a_{n-j+1}, \ldots, -a_n, a_1, a_2, \ldots, a_{n-j})$$

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.
   - Ring LWE is a variant of the LWE assumption which assumes additional structure on the (otherwise random) matrix $\mathbf{G}$.
   - Roughly speaking, ring LWE assumed that a matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ is composed of several $n \times n$ block matrices $\mathbf{G}_1, \ldots, \mathbf{G}_k$, where $m = kn$.
   - Then, for each $\mathbf{G}_i$, the first row is sampled uniformly at random as $(a_1, \ldots, a_n)$, and then each row $j > 1$ is defined to be:

$$(-a_{n-j+1}, \ldots, -a_n, a_1, a_2, \ldots, a_{n-j})$$

   - This gives 2 major benefits:

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.
   - Ring LWE is a variant of the LWE assumption which assumes additional structure on the (otherwise random) matrix $\mathbf{G}$.
   - Roughly speaking, ring LWE assumed that a matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ is composed of several $n \times n$ block matrices $\mathbf{G}_1, \ldots, \mathbf{G}_k$, where $m = kn$.
   - Then, for each $\mathbf{G}_i$, the first row is sampled uniformly at random as $(a_1, \ldots, a_n)$, and then each row $j > 1$ is defined to be:

$$(-a_{n-j+1}, \ldots, -a_n, a_1, a_2, \ldots, a_{n-j})$$

   - This gives 2 major benefits:
     - We only need n $\mathbb{Z}_q$ elements to specify each $\mathbf{G}_i$, so $kn = m$ elements overall.

LWE Key Exchange in Practice

- There are a few techniques we can employ to get actually efficient key-exchange from LWE.
- These techniques are (intuitively) used by ML-KEM!

2. Using *ring LWE*.
   - Ring LWE is a variant of the LWE assumption which assumes additional structure on the (otherwise random) matrix $\mathbf{G}$.
   - Roughly speaking, ring LWE assumed that a matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ is composed of several $n \times n$ block matrices $\mathbf{G}_1, \ldots, \mathbf{G}_k$, where $m = kn$.
   - Then, for each $\mathbf{G}_i$, the first row is sampled uniformly at random as $(a_1, \ldots, a_n)$, and then each row $j > 1$ is defined to be:

$$(-a_{n-j+1}, \ldots, -a_n, a_1, a_2, \ldots, a_{n-j})$$

   - This gives 2 major benefits:
     - We only need n $\mathbb{Z}_q$ elements to specify each $\mathbf{G}_i$, so $kn = m$ elements overall.
     - There are more efficient algorithms for multiplying these matrices.

Wrapping Up

That's All!

That's All!

- Thank you for joining me in this class!

That's All!

- Thank you for joining me in this class!
- I had a great time, I have learned a lot.

That's All!

- Thank you for joining me in this class!
- I had a great time, I have learned a lot.
- I certainly hope you have learned a lot!

That's All!

- Thank you for joining me in this class!
- I had a great time, I have learned a lot.
- I certainly hope you have learned a lot!

Please fill out your course evaluations!

That's All!

- Thank you for joining me in this class!
- I had a great time, I have learned a lot.
- I certainly hope you have learned a lot!

Please fill out your course evaluations!

Final project presentations are next week!

Thank you!