



RUHR-UNIVERSITÄT BOCHUM

Simple Chosen-Ciphertext Security from Low-Noise LPN

PKC 2014, 26.03.2014

Eike Kiltz

Daniel Masny

HGI, Ruhr Universität Bochum

Krzysztof Pietrzak

IST Austria

Outline

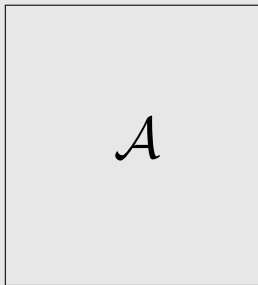
- 1 IND-CCA PKE
- 2 Low-Noise LPN
- 3 Applying Techniques of LWE-Based Schemes
- 4 Simple TBE Based on Low-Noise LPN

IND-CCA Secure PKE

A public key encryption $PKE = (Gen, Enc, Dec)$ is called IND-CCA secure, if for every PPT adversary \mathcal{A} holds:

IND-CCA Secure PKE

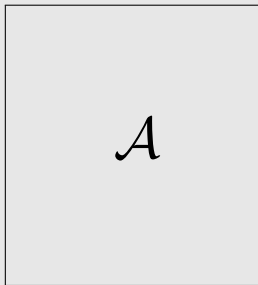
A public key encryption $PKE = (Gen, Enc, Dec)$ is called IND-CCA secure, if for every PPT adversary \mathcal{A} holds: For



$$\xleftarrow{pk} Gen(1^\kappa) \rightarrow (sk, pk);$$

IND-CCA Secure PKE

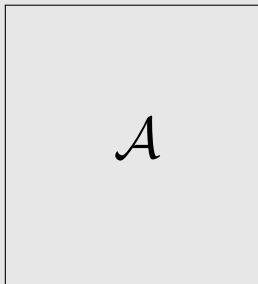
A public key encryption $PKE = (Gen, Enc, Dec)$ is called IND-CCA secure, if for every PPT adversary \mathcal{A} holds: For



$$\begin{array}{l} \xleftarrow{pk} \\ \xrightarrow{c} \\ \xleftarrow{m} \end{array} \quad \begin{array}{l} Gen(1^\kappa) \rightarrow (sk, pk); \\ Dec(sk, c) = m; \end{array}$$

IND-CCA Secure PKE

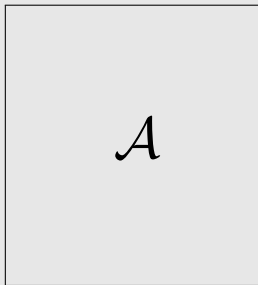
A public key encryption $PKE = (Gen, Enc, Dec)$ is called IND-CCA secure, if for every PPT adversary \mathcal{A} holds: For



$$\begin{array}{l}
 \xleftarrow{pk} \quad Gen(1^\kappa) \rightarrow (sk, pk); \\
 \xrightarrow{c} \quad Dec(sk, c) = m; \\
 \xleftarrow{m} \\
 \xrightarrow{m_0, m_1} \quad \{0, 1\} \rightarrow b; \\
 \xleftarrow{c^*} \quad Enc(pk, m_b) \rightarrow c^*;
 \end{array}$$

IND-CCA Secure PKE

A public key encryption $PKE = (Gen, Enc, Dec)$ is called IND-CCA secure, if for every PPT adversary \mathcal{A} holds: For



$$\begin{array}{l}
 \xleftarrow{pk} \quad Gen(1^\kappa) \rightarrow (sk, pk); \\
 \xrightarrow{c} \quad Dec(sk, c) = m; \\
 \xleftarrow{m} \\
 \xrightarrow{m_0, m_1} \quad \{0, 1\} \rightarrow b; \\
 \xleftarrow{c^*} \quad Enc(pk, m_b) \rightarrow c^*;
 \end{array}$$

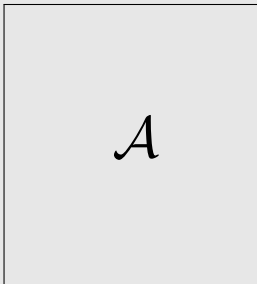
it is hard for \mathcal{A} to guess b .

Tag-Based Encryption (TBE)

A tag-based encryption $TBE = (Gen, Enc, Dec)$ with a tag space $\mathcal{T} = \{0, 1\}^{\kappa}$ is called weakly secure, if for every PPT adversary \mathcal{A} holds:

Tag-Based Encryption (TBE)

A tag-based encryption $TBE = (Gen, Enc, Dec)$ with a tag space $\mathcal{T} = \{0, 1\}^\kappa$ is called weakly secure, if for every PPT adversary \mathcal{A} holds: For

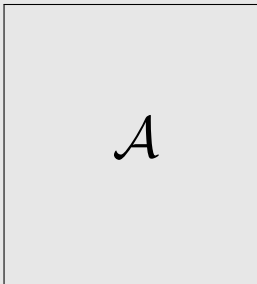


$$\begin{array}{c} \xrightarrow{\tau^* \in \mathcal{T}} \\ \xleftarrow{pk} \end{array}$$

$$Gen(1^\kappa) \rightarrow (sk, pk);$$

Tag-Based Encryption (TBE)

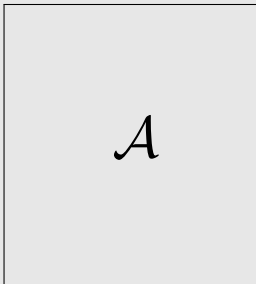
A tag-based encryption $TBE = (Gen, Enc, Dec)$ with a tag space $\mathcal{T} = \{0, 1\}^\kappa$ is called weakly secure, if for every PPT adversary \mathcal{A} holds: For



$$\begin{array}{l}
 \tau^* \in \mathcal{T} \\
 \xrightarrow{\quad} \\
 \xleftarrow{pk} \quad Gen(1^\kappa) \rightarrow (sk, pk); \\
 \tau \neq \tau^*, c \\
 \xrightarrow{\quad} \quad Dec(sk, \tau, c) = m; \\
 \xleftarrow{m}
 \end{array}$$

Tag-Based Encryption (TBE)

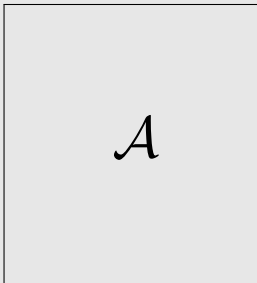
A tag-based encryption $TBE = (Gen, Enc, Dec)$ with a tag space $\mathcal{T} = \{0, 1\}^\kappa$ is called weakly secure, if for every PPT adversary \mathcal{A} holds: For



$$\begin{array}{l}
 \tau^* \in \mathcal{T} \\
 \xrightarrow{\quad} \\
 \xleftarrow{pk} \quad Gen(1^\kappa) \rightarrow (sk, pk); \\
 \tau \neq \tau^*, c \\
 \xrightarrow{\quad} \quad Dec(sk, \tau, c) = m; \\
 \xleftarrow{m} \\
 \xrightarrow{m_0, m_1} \quad \{0, 1\} \rightarrow b; \\
 \xleftarrow{c^*} \quad Enc(pk, \tau^*, m_b) \rightarrow c^*;
 \end{array}$$

Tag-Based Encryption (TBE)

A tag-based encryption $TBE = (Gen, Enc, Dec)$ with a tag space $\mathcal{T} = \{0, 1\}^\kappa$ is called weakly secure, if for every PPT adversary \mathcal{A} holds: For

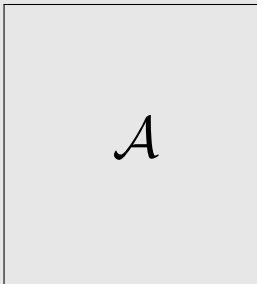


$$\begin{array}{l}
 \xrightarrow{\tau^* \in \mathcal{T}} \\
 \xleftarrow{pk} \quad Gen(1^\kappa) \rightarrow (sk, pk); \\
 \xrightarrow{\tau \neq \tau^*, c} \quad Dec(sk, \tau, c) = m; \\
 \xleftarrow{m} \\
 \xrightarrow{m_0, m_1} \quad \{0, 1\} \rightarrow b; \\
 \xleftarrow{c^*} \quad Enc(pk, \tau^*, m_b) \rightarrow c^*;
 \end{array}$$

it is hard for \mathcal{A} to guess b .

Tag-Based Encryption (TBE)

A tag-based encryption $TBE = (Gen, Enc, Dec)$ with a tag space $\mathcal{T} = \{0, 1\}^\kappa$ is called weakly secure, if for every PPT adversary \mathcal{A} holds: For



$$\begin{array}{l}
 \xrightarrow{\tau^* \in \mathcal{T}} \\
 \xleftarrow{pk} \quad Gen(1^\kappa) \rightarrow (sk, pk); \\
 \xrightarrow{\tau \neq \tau^*, c} \quad Dec(sk, \tau, c) = m; \\
 \xleftarrow{m} \\
 \xrightarrow{m_0, m_1} \quad \{0, 1\} \rightarrow b; \\
 \xleftarrow{c^*} \quad Enc(pk, \tau^*, m_b) \rightarrow c^*;
 \end{array}$$

it is hard for \mathcal{A} to guess b .

Tag-Based Encryption (TBE)

Why Tag-Based Encryption?

- ▶ A TBE is easier to construct than an IND-CCA PKE.

Tag-Based Encryption (TBE)

Why Tag-Based Encryption?

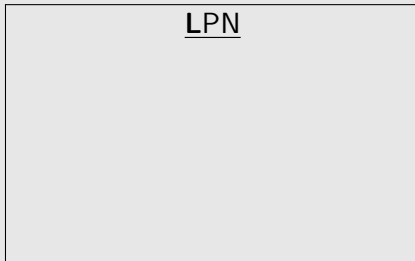
- ▶ A TBE is easier to construct than an IND-CCA PKE.
- ▶ There are generic transformations from a weakly secure TBE to a IND-CCA PKE [BK05, Kil06, BCHK07].

Outline

- 1 IND-CCA PKE
- 2 Low-Noise LPN
- 3 Applying Techniques of LWE-Based Schemes
- 4 Simple TBE Based on Low-Noise LPN

Learning Parity with Low-Noise

Given two distributions:



The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Learning Parity with Low-Noise

Given two distributions:

LPN

$$A \leftarrow \mathbb{Z}_2^{2n \times n};$$

Uniform

The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Learning Parity with Low-Noise

Given two distributions:

LPN

$$A \leftarrow \mathbb{Z}_2^{2n \times n};$$
$$s \leftarrow \mathbb{Z}_2^n;$$

Uniform

The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Learning Parity with Low-Noise

Given two distributions:

LPN

$$\begin{aligned} A &\leftarrow \mathbb{Z}_2^{2n \times n}; \\ s &\leftarrow \mathbb{Z}_2^n; \\ e &\leftarrow \mathcal{B}_p^{2n}, \text{ for } p \in \Theta(1/\sqrt{n}); \end{aligned}$$

Uniform

The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Learning Parity with Low-Noise

Given two distributions:

LPN

$$\begin{aligned}A &\leftarrow \mathbb{Z}_2^{2n \times n}; \\s &\leftarrow \mathbb{Z}_2^n; \\e &\leftarrow \mathcal{B}_p^{2n}, \text{ for } p \in \Theta(1/\sqrt{n}); \\b &= As + e;\end{aligned}$$

Uniform

The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Learning Parity with Low-Noise

Given two distributions:

LPN

$A \leftarrow \mathbb{Z}_2^{2n \times n};$
 $s \leftarrow \mathbb{Z}_2^n;$
 $e \leftarrow \mathcal{B}_p^{2n}, \text{ for } p \in \Theta(1/\sqrt{n});$
 $b = As + e;$
Output (A, b)

Uniform

The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Learning Parity with Low-Noise

Given two distributions:

LPN

$$A \leftarrow \mathbb{Z}_2^{2n \times n};$$

$$s \leftarrow \mathbb{Z}_2^n;$$

$$e \leftarrow \mathcal{B}_p^{2n}, \text{ for } p \in \Theta(1/\sqrt{n});$$

$$b = As + e;$$

Output (A, b)

Uniform

$$A \leftarrow \mathbb{Z}_2^{2n \times n};$$

$$b' \leftarrow \mathbb{Z}_2^{2n};$$

The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Learning Parity with Low-Noise

Given two distributions:

LPN

$$A \leftarrow \mathbb{Z}_2^{2n \times n};$$

$$s \leftarrow \mathbb{Z}_2^n;$$

$$e \leftarrow \mathcal{B}_p^{2n}, \text{ for } p \in \Theta(1/\sqrt{n});$$

$$b = As + e;$$

Output (A, b)

Uniform

$$A \leftarrow \mathbb{Z}_2^{2n \times n};$$

$$b' \leftarrow \mathbb{Z}_2^{2n};$$

Output (A, b')

The Low-Noise LPN assumption is: It is hard to distinguish (A, b) from (A, b') .

Known PKE Constructions

A IND-CPA secure PKE by Alekhnovich [Ale03].

Known PKE Constructions

A IND-CPA secure PKE by Alekhnovich [Ale03].
A TBE by Döttling et al. [DMQN12].

Known PKE Constructions

A IND-CPA secure PKE by Alekhnovich [Ale03].

A TBE by Döttling et al. [DMQN12].

TBE by Döttling et al. [DMQN12]

- ▶ $B_1, \dots, B_q \subset pk, B_1, \dots, B_q \in \mathbb{Z}_2^{\Theta(n) \times n}$.

Known PKE Constructions

A IND-CPA secure PKE by Alekhnovich [Ale03].

A TBE by Döttling et al. [DMQN12].

TBE by Döttling et al. [DMQN12]

- ▶ $B_1, \dots, B_q \subset pk, B_1, \dots, B_q \in \mathbb{Z}_2^{\Theta(n) \times n}$.
- ▶ An encryption uses a B_τ which is derived from B_1, \dots, B_q .

Known PKE Constructions

A IND-CPA secure PKE by Alekhnovich [Ale03].

A TBE by Döttling et al. [DMQN12].

TBE by Döttling et al. [DMQN12]

- ▶ $B_1, \dots, B_q \subset pk, B_1, \dots, B_q \in \mathbb{Z}_2^{\Theta(n) \times n}$.
- ▶ An encryption uses a B_τ which is derived from B_1, \dots, B_q .
- ▶ This results in a large public key ($q \approx 400$).

Outline

- 1 IND-CCA PKE
- 2 Low-Noise LPN
- 3 Applying Techniques of LWE-Based Schemes**
- 4 Simple TBE Based on Low-Noise LPN

Lattice-Based Trapdoor Mechanism

LWE-Based Trapdoor Mechanism [MP12]

- ▶ $sk = T \in \{0, 1\}^{\omega(n) \times \omega(n)}$, $pk = (A, B := TA) \in \mathbb{Z}_p^{\omega(n) \times n}$.

Lattice-Based Trapdoor Mechanism

LWE-Based Trapdoor Mechanism [MP12]

- ▶ $sk = T \in \{0, 1\}^{\omega(n) \times \omega(n)}$, $pk = (A, B := TA) \in \mathbb{Z}_p^{\omega(n) \times n}$.
- ▶ $\tau, \tau^* \in \mathcal{T} = \mathbb{F}_{p^n} \subset \mathbb{Z}_p^{n \times n}$.

Lattice-Based Trapdoor Mechanism

LWE-Based Trapdoor Mechanism [MP12]

- ▶ $sk = T \in \{0, 1\}^{\omega(n) \times \omega(n)}$, $pk = (A, B := TA) \in \mathbb{Z}_p^{\omega(n) \times n}$.
- ▶ $\tau, \tau^* \in \mathcal{T} = \mathbb{F}_{p^n} \subset \mathbb{Z}_p^{n \times n}$.
- ▶ $c \approx As$, $c_1 \approx Bs + G\tau s$, for an error correction matrix G .

Lattice-Based Trapdoor Mechanism

LWE-Based Trapdoor Mechanism [MP12]

- ▶ $sk = T \in \{0, 1\}^{\omega(n) \times \omega(n)}$, $pk = (A, B := TA) \in \mathbb{Z}_p^{\omega(n) \times n}$.
- ▶ $\tau, \tau^* \in \mathcal{T} = \mathbb{F}_{p^n} \subset \mathbb{Z}_p^{n \times n}$.
- ▶ $c \approx As$, $c_1 \approx Bs + G\tau s$, for an error correction matrix G . To reconstruct $s \in \mathbb{Z}_p^n$, $c_1 - Tc \approx G\tau s$ is computed.

Lattice-Based Trapdoor Mechanism

LWE-Based Trapdoor Mechanism [MP12]

- ▶ $sk = T \in \{0, 1\}^{\omega(n) \times \omega(n)}$, $pk = (A, B := TA) \in \mathbb{Z}_p^{\omega(n) \times n}$.
- ▶ $\tau, \tau^* \in \mathcal{T} = \mathbb{F}_{p^n} \subset \mathbb{Z}_p^{n \times n}$.
- ▶ $c \approx As$, $c_1 \approx Bs + G\tau s$, for an error correction matrix G . To reconstruct $s \in \mathbb{Z}_p^n$, $c_1 - Tc \approx G\tau s$ is computed.
- ▶ $B' := TA - G\tau^*$ is as close to uniform as B .

Lattice-Based Trapdoor Mechanism

LWE-Based Trapdoor Mechanism [MP12]

- ▶ $sk = T \in \{0, 1\}^{\omega(n) \times \omega(n)}$, $pk = (A, B := TA) \in \mathbb{Z}_p^{\omega(n) \times n}$.
- ▶ $\tau, \tau^* \in \mathcal{T} = \mathbb{F}_{p^n} \subset \mathbb{Z}_p^{n \times n}$.
- ▶ $c \approx As$, $c_1 \approx Bs + G\tau s$, for an error correction matrix G . To reconstruct $s \in \mathbb{Z}_p^n$, $c_1 - Tc \approx G\tau s$ is computed.
- ▶ $B' := TA - G\tau^*$ is as close to uniform as B .
- ▶ For $c'_1 \approx B's + G\tau s$:

$$c'_1 - Tc \approx G(\tau - \tau^*)s$$

and s is reconstructable for all $\tau \neq \tau^*$.

Lattice-Based Trapdoor Mechanism

LWE-Based Trapdoor Mechanism [MP12]

- ▶ $sk = T \in \{0, 1\}^{\omega(n) \times \omega(n)}$, $pk = (A, B := TA) \in \mathbb{Z}_p^{\omega(n) \times n}$.
- ▶ $\tau, \tau^* \in \mathcal{T} = \mathbb{F}_{p^n} \subset \mathbb{Z}_p^{n \times n}$.
- ▶ $c \approx As$, $c_1 \approx Bs + G\tau s$, for an error correction matrix G . To reconstruct $s \in \mathbb{Z}_p^n$, $c_1 - Tc \approx G\tau s$ is computed.
- ▶ $B' := TA - G\tau^*$ is as close to uniform as B .
- ▶ For $c'_1 \approx B's + G\tau s$:

$$c'_1 - Tc \approx G(\tau - \tau^*)s$$

and s is reconstructable for all $\tau \neq \tau^*$.

- ▶ For $\tau = \tau^*$ some instances remain hard.

A Trapdoor for Low-Noise LPN

Applying the Mechanism to LPN

- ▶ $A, B \in \mathbb{Z}_2^{2n \times n}$, $\mathcal{T} = \mathbb{F}_{2^n} \subset \mathbb{Z}_2^{n \times n}$.

A Trapdoor for Low-Noise LPN

Applying the Mechanism to LPN

- ▶ $A, B \in \mathbb{Z}_2^{2n \times n}$, $\mathcal{T} = \mathbb{F}_{2^n} \subset \mathbb{Z}_2^{n \times n}$.
- ▶ G is a binary error correction code.

A Trapdoor for Low-Noise LPN

Applying the Mechanism to LPN

- ▶ $A, B \in \mathbb{Z}_2^{2n \times n}$, $\mathcal{T} = \mathbb{F}_{2^n} \subset \mathbb{Z}_2^{n \times n}$.
- ▶ G is a binary error correction code.
- ▶ Sample T such that the noise in $c - Tc_1$ is small, while $B := TA$ is close to uniform.

A Trapdoor for Low-Noise LPN

Applying the Mechanism to LPN

- ▶ $A, B \in \mathbb{Z}_2^{2n \times n}$, $\mathcal{T} = \mathbb{F}_2^n \subset \mathbb{Z}_2^{n \times n}$.
- ▶ G is a binary error correction code.
- ▶ Sample T such that the noise in $c - Tc_1$ is small, while $B := TA$ is close to uniform.
- ▶ Either the noise is too big or B is not close to uniform.

A Trapdoor for Low-Noise LPN

Applying the Mechanism to LPN

- ▶ $A, B \in \mathbb{Z}_2^{2n \times n}$, $\mathcal{T} = \mathbb{F}_2^n \subset \mathbb{Z}_2^{n \times n}$.
- ▶ G is a binary error correction code.
- ▶ Sample T such that the noise in $c - Tc_1$ is small, while $B := TA$ is close to uniform.
- ▶ Either the noise is too big or B is not close to uniform.
- ▶ This approach does not immediately apply to LPN.

Replacing the Leftover Hash Lemma

Using a Trapdoor with Low Entropy

- ▶ Sample T from $\mathcal{B}_p^{2n \times 2n}$, $p \in \Theta(1/\sqrt{n})$.

Replacing the Leftover Hash Lemma

Using a Trapdoor with Low Entropy

- ▶ Sample T from $\mathcal{B}_p^{2n \times 2n}$, $p \in \Theta(1/\sqrt{n})$.
- ▶ $A, B = TA$ is computationally indistinguishable from uniform.

Replacing the Leftover Hash Lemma

Using a Trapdoor with Low Entropy

- ▶ Sample T from $\mathcal{B}_p^{2n \times 2n}$, $p \in \Theta(1/\sqrt{n})$.
- ▶ $A, B = TA$ is computationally indistinguishable from uniform.
- ▶ While switching $B = TA$ to $B' = TA - GT^*$, we lose access to the trapdoor T .

Replacing the Leftover Hash Lemma

Using a Trapdoor with Low Entropy

- ▶ Sample T from $\mathcal{B}_p^{2n \times 2n}$, $p \in \Theta(1/\sqrt{n})$.
- ▶ $A, B = TA$ is computationally indistinguishable from uniform.
- ▶ While switching $B = TA$ to $B' = TA - GT^*$, we lose access to the trapdoor T .
- ▶ How to answer decryption queries?

Two Trapdoors

The Two Trapdoors Approach

- ▶ We use two trapdoors $sk_0 = T_0$ and $sk_1 = T_1$.

Two Trapdoors

The Two Trapdoors Approach

- ▶ We use two trapdoors $sk_0 = T_0$ and $sk_1 = T_1$.
- ▶ $pk = (A, B_0, B_1)$

Two Trapdoors

The Two Trapdoors Approach

- ▶ We use two trapdoors $sk_0 = T_0$ and $sk_1 = T_1$.
- ▶ $pk = (A, B_0, B_1)$
- ▶ sk_0 is a trapdoor for (A, B_0) and sk_1 for (A, B_1) .

Switching the Public Key

Switching the Public Key

- ▶ $pk = (A, B_0, B_1)$ is computationally indistinguishable from $pk' = (A, B'_0, B'_1)$.

Switching the Public Key

Switching the Public Key

- ▶ $pk = (A, B_0, B_1)$ is computationally indistinguishable from $pk' = (A, B'_0, B'_1)$.
- ▶ Switch $pk = (A, B_0, B_1)$ to (A, B_0, B'_1) while having access to sk_0 .

Switching the Public Key

Switching the Public Key

- ▶ $pk = (A, B_0, B_1)$ is computationally indistinguishable from $pk' = (A, B'_0, B'_1)$.
- ▶ Switch $pk = (A, B_0, B_1)$ to (A, B_0, B'_1) while having access to sk_0 .
- ▶ sk_1 is now a trapdoor for all $\tau \neq \tau^* \in \mathcal{T}$.

Switching the Public Key

Switching the Public Key

- ▶ $pk = (A, B_0, B_1)$ is computationally indistinguishable from $pk' = (A, B'_0, B'_1)$.
- ▶ Switch $pk = (A, B_0, B_1)$ to (A, B_0, B'_1) while having access to sk_0 .
- ▶ sk_1 is now a trapdoor for all $\tau \neq \tau^* \in \mathcal{T}$.
- ▶ Switch (A, B_0, B'_1) to $pk' = (A, B'_0, B'_1)$ while having access to sk_1 .

Switching the Public Key

Switching the Public Key

- ▶ $pk = (A, B_0, B_1)$ is computationally indistinguishable from $pk' = (A, B'_0, B'_1)$.
- ▶ Switch $pk = (A, B_0, B_1)$ to (A, B_0, B'_1) while having access to sk_0 .
- ▶ sk_1 is now a trapdoor for all $\tau \neq \tau^* \in \mathcal{T}$.
- ▶ Switch (A, B_0, B'_1) to $pk' = (A, B'_0, B'_1)$ while having access to sk_1 .
- ▶ sk_0 and sk_1 are now trapdoors for all $\tau \neq \tau^* \in \mathcal{T}$.

Switching the Public Key

Switching the Public Key

- ▶ $pk = (A, B_0, B_1)$ is computationally indistinguishable from $pk' = (A, B'_0, B'_1)$.
- ▶ Switch $pk = (A, B_0, B_1)$ to (A, B_0, B'_1) while having access to sk_0 .
- ▶ sk_1 is now a trapdoor for all $\tau \neq \tau^* \in \mathcal{T}$.
- ▶ Switch (A, B_0, B'_1) to $pk' = (A, B'_0, B'_1)$ while having access to sk_1 .
- ▶ sk_0 and sk_1 are now trapdoors for all $\tau \neq \tau^* \in \mathcal{T}$.
- ▶ Once pk' is used, decrypting ciphertexts with tag τ^* is hard, given sk_0 and sk_1 .

Outline

- 1 IND-CCA PKE
- 2 Low-Noise LPN
- 3 Applying Techniques of LWE-Based Schemes
- 4 Simple TBE Based on Low-Noise LPN**

A TBE Based on Low-Noise LPN

The Construction

- ▶ $Gen(1^k)$: Output $sk := T_0$, $pk := (A, B_0 := T_0A, B_1 := T_1A, C)$
for $T_0, T_1 \leftarrow \mathcal{B}_p^{2n \times 2n}$ and $A, C \leftarrow \mathbb{Z}_2^{2n \times n}$.

A TBE Based on Low-Noise LPN

The Construction

- ▶ $Gen(1^k)$: Output $sk := T_0$, $pk := (A, B_0 := T_0A, B_1 := T_1A, C)$ for $T_0, T_1 \leftarrow \mathcal{B}_p^{2n \times 2n}$ and $A, C \leftarrow \mathbb{Z}_2^{2n \times n}$.
- ▶ $Enc(pk, \tau, m)$: Sample randomness $s \leftarrow \mathbb{Z}_2^n$. Output

$$\begin{aligned}
 c &:= As, & c_0 &:= (B_0 + G\tau)s, \\
 c_1 &:= (B_1 + G\tau)s, & c_2 &:= Cs + Gm.
 \end{aligned}$$

A TBE Based on Low-Noise LPN

The Construction

- ▶ $Gen(1^k)$: Output $sk := T_0$, $pk := (A, B_0 := T_0A, B_1 := T_1A, C)$ for $T_0, T_1 \leftarrow \mathcal{B}_p^{2n \times 2n}$ and $A, C \leftarrow \mathbb{Z}_2^{2n \times n}$.
- ▶ $Enc(pk, \tau, m)$: Sample randomness $s \leftarrow \mathbb{Z}_2^n$. Output

$$\begin{aligned}
 c &:= As, & c_0 &:= (B_0 + G\tau)s, \\
 c_1 &:= (B_1 + G\tau)s, & c_2 &:= Cs + Gm.
 \end{aligned}$$

- ▶ $Dec(sk, \tau, (c, c_0, c_1, c_2))$: Reconstruct s from $c_0 - T_0c \approx G\tau s$. Check consistency of c_1 with s . Reconstruct m from $c_2 - Cs \approx Gm$. Output m .

Summary

- ▶ We construct a TBE, which can be transformed to an IND-CCA PKE.
- ▶ The security is based on the Low-Noise LPN assumption.
- ▶ pk is computationally indistinguishable from pk' .
- ▶ For pk' , decrypting ciphertexts associated with τ^* is hard.
- ▶ While switching pk to pk' two trapdoors are used to answer decryption queries.









RUHR-UNIVERSITÄT BOCHUM

Many thanks for your attention!

QUESTIONS?

References

-  **Michael Alekhnovich.**
More on average case vs approximation complexity.
In *44th Annual Symposium on Foundations of Computer Science*, pages 298–307. IEEE Computer Society Press, October 2003.
-  **Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz.**
Chosen-ciphertext security from identity-based encryption.
SIAM Journal on Computing, 36(5):1301–1328, 2007.
-  **Dan Boneh and Jonathan Katz.**
Improved efficiency for CCA-secure cryptosystems built using identity-based encryption.
In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer, February 2005.
-  **Nico Döttling, Jörn Müller-Quade, and Anderson CA Nascimento.**
Ind-cca secure cryptography based on a variant of the lpn problem.
In *Advances in Cryptology–ASIACRYPT 2012*, volume 7658, pages 485–503. Springer, 2012.
-  **Eike Kiltz.**
Chosen-ciphertext security from tag-based encryption.
In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, March 2006.
-  **Daniele Micciancio and Chris Peikert.**
Trapdoors for lattices: Simpler, tighter, faster, smaller.
In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, April 2012.