A Simple and Efficient CCA-Secure Lattice KEM in the Standard Model

Xavier Boyen
QUT

Malika Izabachène
CEA List

Qinyi Li
Griffith University

SCN 2020

Key Encapsulation Mechanism (KEM)

- Key generation:
  - $(Pk, Sk) \longleftarrow KGen(1^{\lambda})$
  - Generates public key Pk and private key Sk
- Key encapsulation:
  - $(C, K) \longleftarrow Enc(Pk)$
  - Wraps secret session keys K into ciphertexts C using Pk
- Decapsulation:
  - $\perp$ or $K \longleftarrow Dec(Sk, C)$
  - Recovers secret keys K from ciphertexts C
- Correctness:
  - For honestly generated Pk, Sk, and C, decapsulation works
  - $Pr[K \longleftarrow Dec(Sk, C)] \geq 1 - negl(\lambda)$

- Used for hybrid encryption (with data encapsulation Mechanism)

CCA2-Secure KEM: Definition

- Preparation phase
  - (Pk, Sk) ⟵ KGen(1^λ), choose random b ⟵ {0,1}
  - Get a real key K_1* where (C*, K_1*) ⟵ Enc(Pk)
  - Choose a random key K_0*,
  - Adversary $\mathcal{A}$ obtains (Pk, C*, K_b*)
- Attacking phase
  - $\mathcal{A}$ sends chosen ciphertexts C_1, …,C_ℓ ($C_i \neq C^*$) to challenger $\mathcal{C}$
  - $\mathcal{C}$ replies Dec(Sk, C_i)
- Guessing phase
  - $\mathcal{A}$ outputs b’ as a guess of b
  - It wins if b’ = b
- Require $|Pr[b'=b] - ½|$ is negligible

CCA2-Secure KEM from Lattices

- CPA-Secure PKE + FO transformation [FO13]
  - Practical constructions, random oracle model
- Naor-Yung paradigm [NY90,PS19]
  - CPA-secure PKE + NIZK, less efficient
- Lossy trapdoor functions [PW08,LZL14,BL19]
  - Standard model, stronger lattice assumptions
- CPA-Secure tag-based encryption [CHK04, BK05,MP12]
  - Uses generic transformation, e.g., one-time signatures or MACs
  - Standard model
  - How to avoid generic transformations, i.e., have direct constructions

Our Contributions

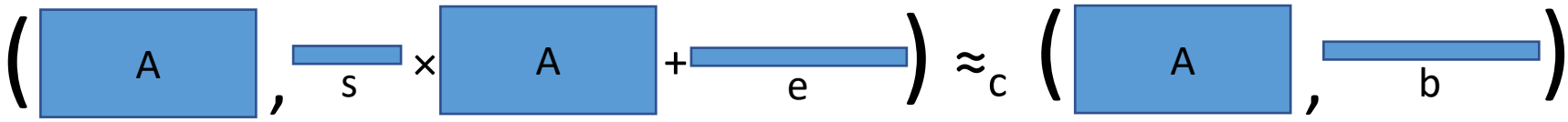
- A simple construction of CCA2-Secure KEM
  - Standard model under LWE problem
  - No generic transformation
  - Efficient as the best-known lattice CCA1-secure PKE [MP12]

	Modulus-to-noise ratio α	|pk|	|ct|	Security	Type
MP12 [24]	$\tilde{O}(1/n)$	$2(n \log q)^2$	$3n \log q + \lambda$	CCA1	PKE
MP12-MAC [24,9]	$\tilde{O}(1/n)$	$2(n \log q)^2 + \|\mathsf{pub}\|$	$3n \log q + \|\mathsf{tag}\| + \|\mathsf{com}\|$	CCA2	PKE
MP12-SIG [24,9]	$\tilde{O}(1/n)$	$2(n \log q)^2$	$3n \log q + \|\mathsf{sig}\| + \|\mathsf{vk}\|$	CCA2	PKE
This work	$\tilde{O}(1/n)$	$3(n \log q)^2$	$3n \log q + \lambda$	CCA2	KEM

λ :security parameter tag size of TBE, n: lattice dimension of LWE, q: modulus, |tag|: size of MAC tag, |com| size of commitment of MAC key, |sig|: size of a one-time signature, |vk|: size of verification key

Computational Assumptions

- learning with errors (LWE) assumption
  - Let χ be a (noise) distribution over $\mathbb{Z}_q$
  - Secrete $s \leftarrow \mathbb{Z}_q^n$, $A \leftarrow \mathbb{Z}_q^{n \times m}$, noise $e \leftarrow \chi^m$, $b \leftarrow \mathbb{Z}_q^m$
  - $(A, sA+e) \approx_c (A, b)$



- Short integer solution (SIS)
  - $A \leftarrow \mathbb{Z}_q^{n \times m}$, $m = O(n \log q)$
  - Hard to find non-zero, low-norm vector $e \in \mathbb{Z}^n$ s.t. $Ae \equiv 0 \pmod q$

Gadget Trapdoors [MP12]

- Defining matrix F = [A|AR + TG]
  - A: random, wide matrix
  - R: low-norm, sufficiently unpredictable matrix
  - G: gadget matrix from [PM12]
  - T: square matrix, called tag

- If T full rank (invertible over $\mathbb{Z}_q$)
  - Efficiently recover s, e_0, e_1 from y = sF + e = s[A|AR+TG] + [e_0|e_1]

- If T=0
  - sF+ e = s[A|AR] + [e_0|e_1] is pseudorandom under LWE

Efficient CCA1-PKE [MP12]

- Pk = (A, A_1); Sk = R
  - Wide, random matrix A, low-norm unpredictable matrix R, A_1 = AR
- Enc(Pk,m)
  - LWE sample y = $s[A|A_1 + TG] + [e_0|e_1]$ for full rank T
  - Message m is encoded into noise terms e_1
  - Ciphertext Ct = (y, T)
- Dec(Sk, Ct)
  - $y = s[A|AR + TG] + [e_0|e_1]$
  - Recovers s, e_0 and e_1 using trapdoor R
  - Recover the message from e_1
- CCA1 Security game
  - Decryption query before seeing the challenge ciphertext
  - No decryption query after

Security of MP12

- In simulation, Pk = (A, $A_1 = AR - T^*G$); Sk = R
  - T* will be used for challenge ciphertext
  - A_1 completely hides T*
  - Any decryption query with $T \neq T^*$, can be answered
- Challenge ciphertext
  - Ct* = (y*, T*)
  - $y^* = s[A|A_1 + T^*G] + [e_0|e_1] = s[A|AR\ G] + [e_0|e_1]$
  - y* is pseudorandom under LWE
- CCA1 security
  - Decryption query $T \neq T^*$ before Ct* = (y*, T*) revealed
- CCA2 insecurity
  - Decryption query (y, T*) where $y \neq y^*$ can't be answered

Ideas

- $y^* = s[A|A_1+T^*G] + [e_0|e_1] = s[A|AR] + [e_0|e_1]$
  - Recall we can decrypt for any $T \neq T^*$
  - Want to bind y^* to T^*, i.e., modifying $y^* \rightarrow$ modifying T^*

- $H(sA+e_0, Ue_1) = T^*$ for random and *wide* matrix U
  - $sA+e_0$ uniquely binds s, A, and e_0, information-theoretically
  - Ue_1 uniquely binds e_1, computationally (by SIS)
  - $sA+e_0$ hides s and e_0 ,computationally (by LWE)
  - Ue_1 hides e_1 information-theoretically

$sA+e_0$, Ue_1 determine y^*

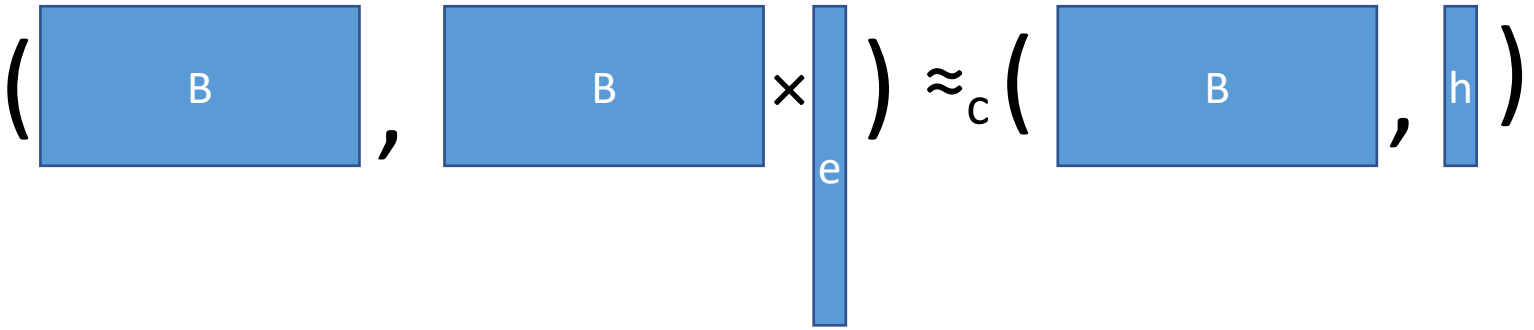
- Don't forget the other part of ct about e_1
  - LWE samples $s(AR) + e_1$

Ideas

- We need to show
  - (sA + e, Ue) computationally hide e and s
  - A SIS instance is added to an LWE instance
- Intuition
  - U is wide and random, e is sufficiently random
  - Ue leaks little info. about e and shouldn't help solving LWE
- How?
  - If knapsack LWE is hard (sA + e, Ue) is pseudorandom

Knapsack LWE

- $B \leftarrow \mathbb{Z}_q^{n \times m}$, $e \leftarrow \chi^m$, $h \leftarrow \mathbb{Z}_q^{n}$, where $m > n + \omega(\log n)$
- $(B, Be) \approx_c (B, h)$



- Equivalent to LWE [MM11]
- We use some samples from knapsack LWE to simulate LWE samples sA+e, the rest are used as Ue

Final Scheme

- Pk = (A, A_1, U); Sk = R
  - Wide, random matrix A, U, low-norm matrix R, A_1 = AR
- Enc(Pk)
  - Choose s, e_0, e_1, and set $sA+e_0$, Ue_1
  - Compute T = H($sA+e_0$, Ue_1)
  - Set y = $[sA+e_0 | s(A_1 + TG)+ e_1] = s[A|A_1 + TG] + [e_0|e_1]$
  - Random session key is encoded into secret s
  - Ct = (y, T)
- Dec(Sk, Ct)
  - $y = s[A|AR + TG] + [e_0|e_1]$
  - Recovers s, e_0 and e_1 using trapdoor R
  - If $T \neq H(sA+e_0, Ue_1)$, reject
  - Recover the message from s

Comparing to MP12 CCA1 Scheme

- Pk = (A, A_1, U); Sk = R
  - Wide, random matrix A, U, lo
- , and set $sA+e_0$, Ue_1
  - $(sA+e_0, Ue_1)$
  - $s(A_1 + TG)+ e_1] = s[A|A_1 + TG] + [e_0|e_1]$
  - Random session key is end
  - Ct = (y, T)
- Dec(Sk, Ct)
  - $y = s[A|AR + TG] + [e_0|e_1]$
  - Recovers s, e_0 and e_1 using trapdoor R
  - If $T \neq H(sA+e_0, Ue_1)$, reject
  - Recover the message from s

LWE assumption:
(1) Same LWE modulus-to-noise ratio
(2) Larger number of LWE samples (transformation from Knapsack LWE)

Pk: One more matrix, wide

Ct: The same size

Decryption: need recomputing the tag

Why KEM only & How to do PKE?

- Tag T* depends on the challenge message.
  - KEM can embed challenge session key (message) into Pk
  - In PKE game, challenge message is given after Pk is produced

- Idea of constructing PKE:
  - Fix T* first, when challenge message is given, choose s, e_0, e_1 to hit T*
  - In security proof, the matrix U can be sampled with a trapdoor
  - Given u and a trapdoor, can sample e_1 s.t. $Ue_1 = u$ (GPV preimage sampling [GPV08])
  - Lattice chameleon hash [CHKP10] $T^* = Ue_1 + Bx$ where x is binary, hash of $sA+e_0$

- Why didn't use for KEM scheme?
  - Requires larger LWE modulus-to-noise ratio (due to having trapdoor for U)

Summary

- Constructions of lattice KEM
  - Simple and direct
  - CCA security in standard model
  - As efficient as the best known CCA1 scheme [MP12]

- Techniques
  - Adding a SIS function to LWE noise to make the noise non-malleable

Thank you!