Secure Hybrid Encryption from Hard Learning Problems in the Standard Model

Xavier Boyen
QUT

Malika Izabachène
Cosmian, Paris

Qinyi Li
Griffith University

PQCrypto 2021

Our Results

- CCA2-Secure hybrid encryption systems in the standard model under LWE/Low-noise LPN

- A KEM w/o CCA2 security plus a CCA-secure DEM (à la Kurosawa-Desmedt system [KD04])

- Outperform known standard model CCA2-secure PKE (with assumptions that symmetric primitives are secure, e.g., AES, HMAC)

Hybrid Encryption

- Public-key encryption (PKE)
  - No need to pre-share secret keys
  - Inefficient when encrypting long messages

- Symmetric-key (secret-key) encryption (SKE)
  - Efficient when encrypting long messages
  - Requires shared secret keys

- Hybrid encryption (HE)
  - Combine the merits of PKE and SKE
  - Use PKE to warp a random session key k, which is short
  - Use SKE with k to encrypt the actual message.

Key Encapsulation Mechanism (KEM)

- Key generation:
  - (pk, sk) $\longleftarrow$ KEM.Gen(1^λ)
  - Generates public key pk and private key sk
- Key encapsulation:
  - (c, k) $\longleftarrow$ KEM.Enc(pk)
  - Wraps secret session keys k into ciphertexts c using pk
- Decapsulation:
  - $\perp$ or k $\longleftarrow$ KEM.Dec(sk, c)
  - Recovers secret keys k from ciphertexts c
- Correctness:
  - For honestly generated pk, sk, and c, decapsulation works
  - $\Pr[k \longleftarrow \text{KEM.Dec}(sk, c)] \geq 1 - \text{negl}(\lambda)$

- Security:
  - (KEM.Enc(pk), k) $\approx_c$ KEM.Enc(pk), r) where r is a random session key
  - Under chosen-plaintext attacks or chosen ciphertext attacks

Data Encapsulation Mechanism (DEM)

- Data encapsulation:
  - $c \longleftarrow \text{DEM.Enc}(k,M)$
- Decapsulation:
  - $\perp$ or $M \longleftarrow \text{DEM.Dec}(k, c)$
- Security: for random k
  - $\text{DEM.Enc}(k, M_0) \approx_c \text{DEM.Enc}(k, M_1)$
  - Under chosen-plaintext attack or chosen ciphertext attack

Hybrid Encryption: Syntax

- Keygen(1^λ): Key generation
  - (pk, sk) ⟵ KEM.Gen(1^λ)
  - Generates public key Pk = pk and private key Sk = sk
- Enc(Pk, M): Encryption
  - (c, k) ⟵ KEM.Enc(Pk), c' ⟵ KEM.Enc(k,M)
  - Return Ct = (c, c')
- Dec(Pk, Sk, Ct): Decryption
  - Parse Ct = (c,c')
  - k ⟵ KEM.Dec(sk, c), M⟵ DEM.Dec(k,c')
- Correctness
  - Correctness of KEM and DEM

Hybrid Encryption: CCA2 Security

- Preparation phase
  - Challenger $\mathcal{C}$ generates (Pk, Sk) and gives the adversary $\mathcal{A}$ Pk
- Attacking phase 1
  - $\mathcal{A}$ adaptively sends chosen ciphertexts $Ct_1, \ldots, Ct_\ell$ to $\mathcal{C}$
  - $\mathcal{C}$ replies Dec(Sk, Ct_i)
- Challenge phase
  - $\mathcal{A}$ sends M_0, M_1 to $\mathcal{C}$
  - $\mathcal{C}$ flips a fair coin $b \in \{0,1\}$, and sends $Ct^* \longleftarrow Enc(Pk, M_b)$ to $\mathcal{A}$
- Attacking phase 2
  - $\mathcal{A}$ adaptively sends chosen ciphertexts $Ct_{\ell+1}, \ldots, Ct_{\ell'}$ to $\mathcal{C}$
  - Restriction: $Ct_i \neq Ct^*$
  - $\mathcal{C}$ replies Dec(Sk, Ct_i)
- Guessing phase
  - $\mathcal{A}$ outputs b' and wins if b' = b
- Secure if adv = $|\Pr[b'=b] - ½|$ is negligible

How to Obtain CCA2-Secure HE in the Standard Model?

- Generic security composition: A CCA2-secure KEM plus a CCA-secure DEM give CCA2-secure HE [CS03]
  - CCA2-secure DEM (simple and efficient):
  - CCA2-secure KEM (non-trivial and less efficient):
    - Naor-Young paradigm, lossy trapdoor function, hash proof systems, TBE/IBE plus BCHK transformation……
- Kurosawa-Desmedt system [KD04] based on decisional Diffie-Hellman (DDH) problem:
  - A more efficient KEM without CCA2 security
  - Combining with DEM gives a more efficient HE system
  - Post-quantum examples? (This work)

Our HE Constructions

- KEM: uses the state-of-the art tag-based encryption (TBE)
  - LWE TBE from [MP12] and low-noise LPN TBE from [KMP14] (which are not CCA2-secure by themselves)
- DEM: standard construction, very efficient
  - An unforgeable MAC plus a CPA-secure symmetric cipher
- Exploit properties of LWE/LPN and their trapdoors
- Proof ideas stem from Boneh-Katz transformation
  - BK-transformation uses universal hash-based commitment + MAC
  - Ours uses LWE/LPN ciphertext as commitment

Computational Problem

- Decisional learning with errors (LWE) problem [Reg05]
  - Let χ be a (noise) distribution over $\mathbb{Z}_q$
  - $s \longleftarrow \mathbb{Z}_q^{\,n}$, $A \longleftarrow \mathbb{Z}_q^{\,n\times m}$, $e \longleftarrow \chi^m$, $b \longleftarrow \mathbb{Z}_q^{\,m}$ (m>n)
  - $(A, sA + e) \approx_c (A, b)$

- Viewing LWE as a kind of commitment of the secret s
  - Computational hiding: $(A, sA + e) \approx_c (A, b)$
  - Statistical binding: for m> n sA+e uniquely determines s

- LPN problem has similar properties.

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 TBE/CCA1-PKE [MP12]

- pk = (A, A_1); sk = R
  - Wide, random matrix A, low-norm unpredictable matrix R, $A_1 = AR$
- Enc(pk,m)
  - Choose random full rank T*
  - LWE samples

$$y = [y_0 | y_1] = s[A \,| A_1 + T^*G] + [e_0 |\ e_1],$$
$$z = sU + e_2 + m \lfloor q/2 \rfloor$$

  - Ciphertext c = (y, z, T*)
- Dec(sk, c)
  - $y = s[A|AR + T^*G] + [e_0|e_1]$
  - Recovers s, e_0 and e_1 using trapdoor R
  - Recover the message m from z
- CCA1 security notion:
  - 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
  - $Ct^* = (y^*, z^*, T^*)$
  - $y^* = s[A|A_1 + T^*G] + [e_0|e_1] = s[A|AR] + [e_0|e_1]$
  - $Z^* = sU + e_2 + m\lfloor q/2 \rfloor$
  - y^*, z^* are 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

Our Construction

- $Pk = (A, A_1, U)$; $Sk = R$
  - $A_1 = AR$
- Enc(Pk, m)
  - Choose k, s, e_0, e_1, e_2, and compute $y_0 = sA+e_0$, $z = sU+e_2+k$ (encapsulating k)
  - Compute $T = H(sA+e_0, sU+e_2+k)$
  - Set $y = [y_0|y_1] = [y_0|s(A_1 + TG)+ e_1] = s[A|A_1 + TG] + [e_0|e_1]$
  - $(k_1,k_2) = KDF(k)$, ψ = SKE.Enc(k_1, m), $\tau = MAC(k_2,y\|z\|\psi)$
  - $Ct = (y, z, \psi, \tau)$
- Dec(Sk, Ct)
  - Set $T = H(y_0, z)$
  - $y = [y_0|y_1] = s[A|AR + TG] + [e_0|e_1]$; Recovers s, e_0, e_1 and k using trapdoor R
  - $(k_1,k_2) = KDF(k)$, m = SKE.Dec(k_1, ψ)
  - Return m if $\tau = MAC(k_2,y\|z\|\psi)$.

Our Construction

-

KEM Part
MP12 encryption: $y = [y_0|y_1] = s[A|A_1 + TG] + [e_0|e_1]$

- Enc(Pk, m)
  - Choose k, s, e_0, e_1, e_2, and compute y_0 … $z = sU+e_2+k$ (encapsulating k)
  - Compute $T = H(sA+e_0, sU+e_2+k)$
  - Set $y = [y_0|y_1] = [y_0|s(A_1 + TG)+ e_1] = s[A|A_1 + TG] + [e_0|e_1]$
  - (k_1,k_2) = KDF(k), ψ = SKE.Enc(k_1, m), $\tau = \mathrm{MAC}(k_2, y\|z\|\psi)$
  - Ct = (y, z, ψ
- Dec(Sk, Ct)
  - Set T =
  - … + TG] + [e_0
  - (k_1,k_2) = KDF(k), m = SKE.Dec(k_1, ψ)
  - Return m if $\tau = \mathrm{MAC}(k2, y\|z\|\psi)$.

Commitment of session key k

DEM Part
Symmetric encryption: ψ = SKE.Enc(k_1, m)
MAC: $\tau = \mathrm{MAC}(k_2, y\|z\|\psi)$

CCA2 Security (Idea)

- Challenge Ciphertext:
  - $y = [y_0|y_1] = [y_0|s(A_1 + TG) + e_1] = s[A|A_1 + TG] + [e_0|e_1]$; $z = sU + e_2 + k\lfloor q/2 \rfloor$
  - $(k_1,k_2) = \mathrm{KDF}(k)$, $\psi = \mathrm{SKE.Enc}(k_1, m)$, $\tau = \mathrm{MAC}(k_2, y\|z\|\psi)$

- Preventing adversary from crafting the challenge ciphertext to a valid decryption query

- $T = H(sA+e_0, sU+e_2+k\lfloor q/2 \rfloor) = H(y_0, z)$
  - **LWE statistical binding**: modifying k changes T => can answer decryption queries
  - **LWE computational hiding**: k is hidden
  - Without knowing k, modifying y, z, ψ, τ => a MAC forgery
  - So, decryption queries are not helpful

Summary

- Constructions of hybrid encryption for LWE/LPN
  - CCA2 security in standard model
  - Avoid generic transformation
  - Non-CCA2-secure KEMs

- Techniques
  - Explore that LWE/LPN are commitment schemes (statistical binding and computational hiding)

Thank you!