Advanced Encryption: IBE and Pairings

CS 276: Introduction to Cryptography

Sanjam Garg

April 1, 2026

Overview

1. Identity-Based Encryption (IBE)
2. Signatures from IBE
3. CCA-Secure Encryption from IBE
4. Zero-Knowledge Proofs

IBE Syntax

Definition 1 (Identity-Based Encryption)

An IBE scheme $\mathcal{E}_{id} = (G, K, E, D)$ has:

- $G(1^n)$: setup; outputs master public key mpk and master secret key msk.
- $K(msk, id)$: keygen; outputs secret key sk_{id} for identity id.
- $E(mpk, id, m)$: encrypts m under identity id, outputs ciphertext c.
- $D(sk_{id}, c)$: decrypts; outputs m or $\perp$.

IBE Correctness

Definition 2 (Correctness)

For all n, identities id, messages m:

$$\Pr\Big[(mpk, msk) \leftarrow G(1^n),\ sk_{id} \leftarrow K(msk, id),\ c \leftarrow E(mpk, id, m),\ m' \leftarrow D(sk_{id}, c)\Big] = 1 - \mathrm{negl}(n),$$

where the probability is over the randomness of G, K, E.

IBE Security: IBE-CPA Game

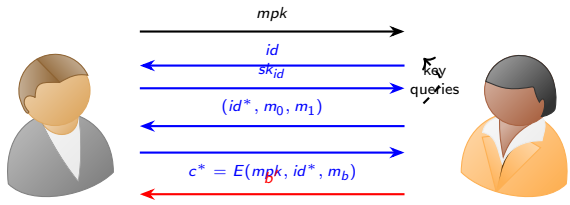
Game $\mathrm{Exp}^{\mathrm{IBE,CPA}}_{\Pi,\mathcal{A}}(n)$

1. Challenger runs $(mpk, msk) \leftarrow G(1^n)$, sends mpk to $\mathcal{A}$.
2. $\mathcal{A}$ may make **key queries** id and receive sk_{id} (for any $id \neq id^*$).
3. $\mathcal{A}$ outputs challenge identity id^* and messages (m_0, m_1).
4. Challenger samples $b \leftarrow \{0,1\}$, computes $c^* \leftarrow E(mpk, id^*, m_b)$, sends c^* to $\mathcal{A}$.
5. $\mathcal{A}$ may continue to make key queries for identities $id \neq id^*$.
6. Finally $\mathcal{A}$ outputs b'; output 1 if $b' = b$, else 0.

IBE-CPA: Challenger vs Adversary

$$\mathrm{Exp}^{\mathrm{IBE,CPA}}_{\Pi,\mathcal{A}}(n)$$

sample $(mpk, msk) \leftarrow G(1^n)$, $b \leftarrow \{0,1\}$



output 1 if $b' = b$; else 0

IBE Security Definition

Definition 3 (IBE-CPA security)

An IBE $\mathcal{E}_{id} = (G, K, E, D)$ is IBE-CPA secure if for every non-uniform PPT adversary $\mathcal{A}$:

$$\Pr[\mathrm{Exp}^{\mathrm{IBE,CPA}}_{\Pi,\mathcal{A}}(n) = 1] = \frac{1}{2} + \mathrm{negl}(n).$$

Equivalently, $\mathcal{A}$'s advantage over random guessing is negligible.

Digital Signatures from IBE

Digital Signatures from IBE

Construction

Given a secure IBE $\mathcal{E}_{id} = (G, K, E, D)$:

- $G'(1^n)$: run $(mpk, msk) \leftarrow G(1^n)$; verification key $vk = mpk$, signing key $sk = msk$.
- $S'(sk, m)$: treat m as an identity; output signature $\sigma \leftarrow K(sk, m)$.
- $V'(vk, m, \sigma)$: pick random r from IBE message space; set $c \leftarrow E(vk, m, r)$; accept iff $D(\sigma, c) = r$.

Digital Signatures from IBE

Construction

Given a secure IBE $\mathcal{E}_{id} = (G, K, E, D)$:

- $G'(1^n)$: run $(mpk, msk) \leftarrow G(1^n)$; verification key $vk = mpk$, signing key $sk = msk$.
- $S'(sk, m)$: treat m as an identity; output signature $\sigma \leftarrow K(sk, m)$.
- $V'(vk, m, \sigma)$: pick random r from IBE message space; set $c \leftarrow E(vk, m, r)$; accept iff $D(\sigma, c) = r$.

Intuition

IBE keygen for identity m becomes **signing** on m; verification uses IBE encryption/decryption as a consistency check. Security: a forgery on (m, σ) gives an IBE break for identity m.

Signatures from IBE: Security Sketch

Theorem 4

If $\mathcal{E}_{id} = (G, K, E, D)$ is a secure IBE scheme with super-polynomial message space, then the derived scheme $\mathcal{S} = (G', S', V')$ is a secure (EUF-CMA) digital signature scheme.

Signatures from IBE: Reduction

Reduction $\mathcal{B}$ (from SIG to IBE)

Let $\mathcal{A}$ break the signature scheme; build $\mathcal{B}$ that breaks IBE-CPA:

1. $\mathcal{B}$ receives mpk from the IBE challenger; forwards $vk = mpk$ to $\mathcal{A}$.
2. On signing query m_i, $\mathcal{B}$ queries IBE keygen for identity m_i to obtain sk_{m_i} and returns it as the signature σ_i.
3. Eventually $\mathcal{A}$ outputs a forgery (m, σ) with m not previously queried to the signing oracle.
4. $\mathcal{B}$ picks random t_0, t_1 in the IBE message space and uses the IBE challenger to get $c^* = E(mpk, m, t_b)$.
5. $\mathcal{B}$ computes $t' = D(\sigma, c^*)$ and outputs $b' = 1$ if $t' = t_1$, else 0.

CCA1 Encryption from IBE

CCA1 Encryption from IBE

Construction (CPA/CCA1 from IBE)

Let (G, K, E, D) be a CPA-secure IBE.

- KeyGen(1^n): $(pk, sk) \leftarrow G(1^n)$; set $pp = pk$, $msk = sk$.
- Enc(pp, m):
  1. Sample fresh identity $id \leftarrow \mathcal{ID}$ (e.g. random string).
  2. Compute $c = E(pp, id, m)$ and output (id, c).
- Dec$(msk, (id, c))$:
  1. Compute $sk_{id} \leftarrow K(msk, id)$.
  2. Output $D(sk_{id}, c)$.

CCA1 Encryption from IBE

Construction (CPA/CCA1 from IBE)

Let (G, K, E, D) be a CPA-secure IBE.

- KeyGen(1^n): $(pk, sk) \leftarrow G(1^n)$; set $pp = pk$, $msk = sk$.
- Enc(pp, m):
  1. Sample fresh identity $id \leftarrow \mathcal{ID}$ (e.g. random string).
  2. Compute $c = E(pp, id, m)$ and output (id, c).
- Dec$(msk, (id, c))$:
  1. Compute $sk_{id} \leftarrow K(msk, id)$.
  2. Output $D(sk_{id}, c)$.

Why CCA1 (sketch)

Challenge ciphertext uses a fresh id^*; in the CCA1 game the adversary never learns sk_{id^*} (no decryption queries after challenge), so IBE security transfers to the PKE scheme.

CCA2 Encryption from IBE + Signatures

Setting

Let (G, K, E, D) be a CPA-secure IBE scheme and $(\mathsf{Gen}_{\mathsf{sig}}, \mathsf{Sign}, \mathsf{Verify})$ a (one-time) secure signature scheme.

CCA2 Encryption from IBE + Signatures

Setting

Let (G, K, E, D) be a CPA-secure IBE scheme and $(\mathsf{Gen}_{\mathsf{sig}}, \mathsf{Sign}, \mathsf{Verify})$ a (one-time) secure signature scheme.

Key generation

- $\mathrm{KeyGen}(1^n)$: run $(pk, sk) \leftarrow G(1^n)$; set public parameters $pp = pk$, master secret $msk = sk$.

CCA2 Encryption from IBE + Signatures (algorithms)

CCA2 Encryption from IBE + Signatures (algorithms)

Encryption and decryption

- Enc(pp, m):
  1. Generate signing keypair $(vk_s, sk_s) \leftarrow \mathsf{Gen}_{\mathsf{sig}}(1^n)$.
  2. Let $id = vk_s$ (identity is the verification key).
  3. Compute $c = E(pp, id, m)$ (IBE-encrypt under identity id).
  4. Compute signature $\sigma = \mathsf{Sign}(sk_s, c)$.
  5. Output ciphertext (id, c, σ).
- Dec$(msk, (id, c, \sigma))$:
  1. If $\mathsf{Verify}(id, c, \sigma) = 0$, reject ($\perp$).
  2. Compute $sk_{id} \leftarrow K(msk, id)$ and output $D(sk_{id}, c)$.

Why CCA-Secure? (Sketch)

Intuition

- Each ciphertext "names" its own identity $id = vk_s$; c is an IBE ciphertext to that identity.
- Decryption first checks $\text{Verify}(id, c, \sigma)$, so only **signed** ciphertexts are ever decrypted.
- In CCA2, to ask decryption on a modified challenge, the adversary must forge a new valid (id, c, σ), contradicting signature security; otherwise IBE security applies as in CCA1.

Why CCA-Secure? (Sketch)

Intuition

- Each ciphertext "names" its own identity $id = vk_s$; c is an IBE ciphertext to that identity.
- Decryption first checks Verify(id, c, σ), so only **signed** ciphertexts are ever decrypted.
- In CCA2, to ask decryption on a modified challenge, the adversary must forge a new valid (id, c, σ), contradicting signature security; otherwise IBE security applies as in CCA1.

Notes

- Requires a slightly stronger signature notion: freshness for tuples (m^*, σ^*) (even if m^* appeared before) – satisfied by deterministic or one-time signatures (e.g. BLS).
- This is a generic transform: CPA-secure IBE + one-time signatures $\Rightarrow$ CCA2-secure PKE.

Proving Computation Integrity

Zero-Knowledge Proofs, Interactive Proofs, and ZK for NP

Interactive Proof Systems

Definition 5 (Interactive proof for language L)

An interactive protocol $(\mathcal{P}, \mathcal{V})$ is an interactive proof system for L if $\mathcal{V}$ is PPT and:

- **Completeness**: for all $x \in L$, $\Pr[\mathcal{V} \text{ accepts after interacting with } \mathcal{P}(x)] = 1$.
- **Soundness**: for all $x \notin L$ and all (possibly unbounded) $\mathcal{P}^*$, $\Pr[\mathcal{V} \text{ accepts after } \mathcal{P}^*(x)] \leq \mathsf{negl}(|x|)$.

Interactive Proof Systems

Definition 5 (Interactive proof for language L)

An interactive protocol $(\mathcal{P}, \mathcal{V})$ is an interactive proof system for L if $\mathcal{V}$ is PPT and:

- **Completeness**: for all $x \in L$, $\Pr[\mathcal{V} \text{ accepts after interacting with } \mathcal{P}(x)] = 1$.
- **Soundness**: for all $x \notin L$ and all (possibly unbounded) $\mathcal{P}^*$, $\Pr[\mathcal{V} \text{ accepts after } \mathcal{P}^*(x)] \leq \mathsf{negl}(|x|)$.

We allow probabilistic prover/verifier and interaction; this can prove statements where non-interactive succinct certificates are not known.

GI vs GNI: Why Interaction Matters

Graph Isomorphism (GI)

For $GI = \{(G_0, G_1) \mid G_0 \cong G_1\}$, a prover can send an isomorphism w directly; verifier checks in polynomial time.

GI vs GNI: Why Interaction Matters

Graph Isomorphism (GI)

For $GI = \{(G_0, G_1) \mid G_0 \cong G_1\}$, a prover can send an isomorphism w directly; verifier checks in polynomial time.

Graph Non-Isomorphism (GNI)

For $GNI = \{(G_0, G_1) \mid G_0 \ncong G_1\}$, no short witness is known. Interaction helps:

- Verifier sends a random relabeling H of one graph.
- If $G_0 \ncong G_1$, prover identifies which graph produced H.
- If $G_0 \cong G_1$, even unbounded prover can only guess with probability $1/2$.

Repeat k times to make soundness error 2^{-k}.

Interactive Proof for GNI

One round

1. $\mathcal{V}$ samples $b \leftarrow \{0,1\}$ and random permutation π, sets $H = \pi(G_b)$, sends H.
2. $\mathcal{P}$ outputs b' (which graph H came from).
3. $\mathcal{V}$ accepts iff $b' = b$.

Interactive Proof for GNI

One round

1. $\mathcal{V}$ samples $b \leftarrow \{0,1\}$ and random permutation π, sets $H = \pi(G_b)$, sends H.
2. $\mathcal{P}$ outputs b' (which graph H came from).
3. $\mathcal{V}$ accepts iff $b' = b$.

Analysis

- If $(G_0, G_1) \in GNI$, unbounded $\mathcal{P}$ always answers correctly.
- If $(G_0, G_1) \notin GNI$, distributions are identical; any prover succeeds with probability at most $1/2$.

Sequential repetition gives negligible soundness error.