Identity-based encryption: Public-key encryption where "public key" can be an arbitrary string (e.g., email address)

Setup: Output master public key mpk
master secret key msk

KeyGen (msk, id): Output secret key sk_{id} for id

Encrypt (mpk, id, m): Encrypts message m to user id

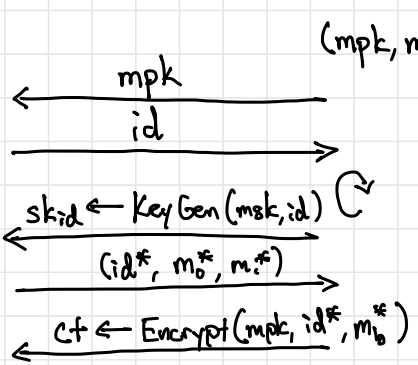
Decrypt (sk_{id}, ct): Decrypts ciphertext ct using identity key

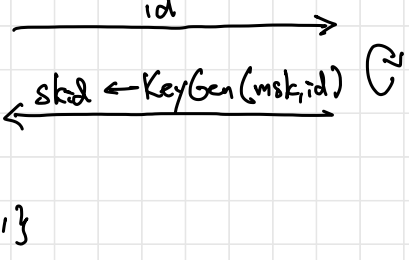
Correctness: $(mpk, msk) \leftarrow$ Setup
$sk_{id} \leftarrow$ KeyGen (msk, id)
$ct \leftarrow$ Encrypt (mpk, id, m)
$\Rightarrow$ Decrypt (sk_{id}, ct) = m

Encryption only requires knowledge of mpk and the user identity
No need to distribute or store individual public keys.

Generally, think of $id \in \{0,1\}^\lambda$ (λ-bit string)
↳ mpk "compresses" 2^λ public keys into a single short key

Security:





Scheme is (adaptively) secure if for all efficient adversaries A that does not query for a key on id^*: A

$$\left| Pr[b' = 1 \mid b = 0] - Pr[b' = 1 \mid b = 1] \right| \leq negl.$$

Namely, adversary cannot break semantic security for any identity for which it does not know the secret key.

Boneh-Franklin IBE scheme: secret key for an id will be BLS signature on id and decryption will "verify" possession of signature

"witness encryption for knowledge of BLS signature"

Setup: $k \xleftarrow{R} \mathbb{Z}_p$ $\quad mpk: g^k \quad msk: k$

KeyGen (msk, id): $sk_{id} = H(id)^{msk}$

Encrypt (mpk, id, m): assume $m \in G_T$

$r \xleftarrow{R} \mathbb{Z}_p$

$ct = (g^r, e(mpk, H(id))^r \cdot m)$

Decrypt (sk_{id}, ct): $\quad ct = (u, v)$ $\quad$ output $v / e(u, sk_{id})$

ElGamal-style encryption:

$ct = (g^r, h^r \cdot m)$

user public key is now

$e(g^k, H(id))$

Correctness: $$\frac{ct_1}{e(u, sk_{id})} = \frac{e(g^k, H(id))^r \cdot m}{e(g^r, H(id)^k)} = m \cdot \frac{e(g, H(id))^{kr}}{e(g, H(id))^{kr}} = m$$

Security: Relies on DBDH assumption:

$$(g, g^x, g^y, g^z, e(g,g)^{xyz}) \overset{c}{\approx} (g, g^x, g^y, g^z, e(g,g)^r)$$

and modeling H as a random oracle.

here, we will consider stronger setting where A tries to distinguish encryption of m^* from random

Suppose A can break security of Boneh-Franklin. We assume the following:

1. A makes at most Q queries
2. A queries random oracle on id before making a key-generation query on id
3. A queries random oracle on id^* (challenge identity)

We use A to construct an adversary B that breaks DBDH:

1. Let (g, u, v, w, T) be the DBDH challenge.
2. Guess an index $i^* \xleftarrow{R} [Q]$ (to plant the DBDH challenge)
3. Set $mpk = u$ and give mpk to A.
4. A now makes queries:
   - Random oracle query on id: If this is query i^*, reply with $sk_{id} = v$. Otherwise, sample $\alpha_{id} \xleftarrow{R} \mathbb{Z}_p$ and set $H(id) := g^{\alpha_{id}}$.
   - Key-generation query on id: If id is the $(i^*)^{th}$ identity queried to random oracle, then abort. Otherwise, reply with $sk_{id} = u^{\alpha_{id}}$.
5. A outputs a challenge (id^*, m). If id^* is not the $(i^*)^{th}$ identity queried to random oracle, then abort. Otherwise, let $ct^* = (w, m \cdot T)$.
6. A continues making queries. Answer as before.
7. A outputs a bit $b' \in \{0,1\}$ which B outputs.

Observe: let $u = g^x$, $v = g^y$, $w = g^z$. Then, $sk_{id} = g^{\alpha_{id} x} = H(id)^x$, which is distributed as in the real scheme.

Suppose $T = e(g,g)^{xyz}$. Then

$$ct = (g^z, e(g,g)^{xyz} \cdot m)$$

$$= (g^z, e(g^x, g^y)^z \cdot m) = (g^z, e(mpk, H(id^*))^z \cdot m)$$

[honestly-generated ciphertext]

Suppose $T = e(g,g)^r$. Then,

$$ct = (g^z, g^r \cdot z)$$

↑ uniformly random since $r \xleftarrow{R} \mathbb{Z}_p$.

If A can distinguish ciphertexts from random with advantage ε, then B breaks DBDH with advantage ε/Q (whenever guess is correct).

IBE does change the threat model of public-key encryption

PKE: everyone generates their own public key

IBE: central authority issues keys — needs a long-term secret key

↳ Can achieve variant of IBE without key escrow problem called registration-based encryption (RBE)

Here, users generate keys themselves and central authority aggregates the keys instead.