# 1 Pairings / Bilinear maps

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of order $p$ prime, generated respectively by $g_1$, $g_2$ and $g_T$.

A **pairing** (or **bilinear map**) is a function $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ such that:

- (Bilinearity) $\forall a, b \in \mathbb{Z}_p,\ e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$
- (Non degeneracy) $\forall (x, y) \in \mathbb{G}_1 \times \mathbb{G}_2,\ e(x, y) = 1_{\mathbb{G}_T} \implies x = 1_{\mathbb{G}_1}$ or $y = 1_{\mathbb{G}_2}$.

## 1.1 Three types of pairings

There is three types of pairings:

- Type 1: $\mathbb{G}_1 = \mathbb{G}_2$
- Type 2: $\mathbb{G}_1 \neq \mathbb{G}_2$ but there exists an computationally efficient group morphism $\phi : \mathbb{G}_1 \longrightarrow \mathbb{G}_2$
- Type 3: $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no computationally efficient group morphism $\phi : \mathbb{G}_1 \longrightarrow \mathbb{G}_2$

In this class we will consider only type 1 pairings, hence $\mathbb{G}_1 = \mathbb{G}_2$ denoted $\mathbb{G}$.
In practice, $\mathbb{G}_1, \mathbb{G}_2$ are elliptic curves and $\mathbb{G}_T$ is a finite field.

What we know about pairings:

- We know groups without pairings, DDH can hold for these groups. (For instance, groups that are the codomain of a pairing typically do not have pairings.)
- We know some groups with pairings. eg: Weil pairings, Tate pairings.

## 1.2 Application of pairings: the 1-round 3-party key exchange

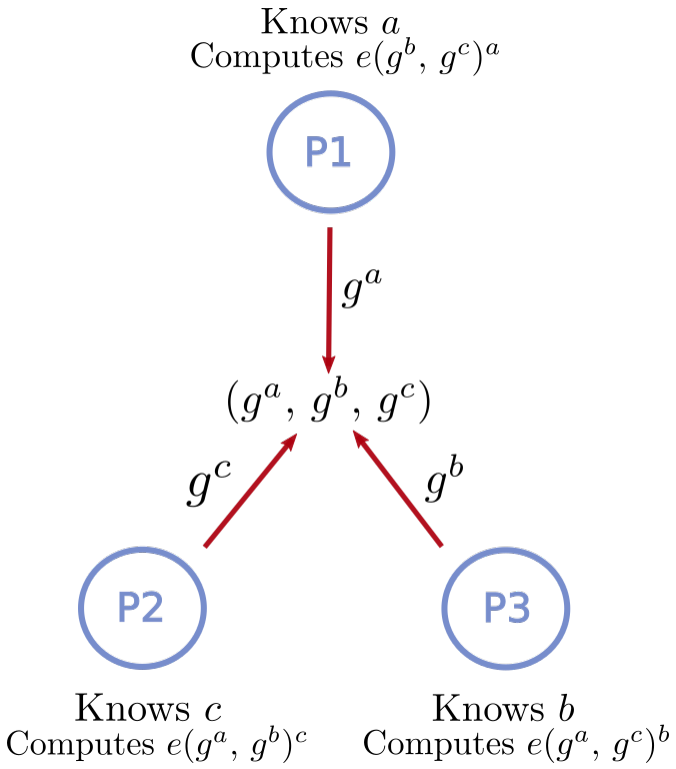


Figure 1: A 1-round 3-party key exchange based on pairings, the shared secret is $e(g, g)^{abc}$.

$P_1$, $P_2$ and $P_3$ want to compute a shared secret by communicating over an insecure channel.

In [1], Joux showed that pairings give you a 1-round 3-party key exchange (KE), given in Figure 1.

As with the 2-party KE, 3-party KE's security also relies on a hardness assumption. Instead of the DDH assumption, we'll have to introduce the DBDH assumption.

**Decisional Bilinear Diffie Hellman assumption (DBDH)** for a group $\mathbb{G}$
Given a pairing $e$ and $g \in \mathbb{G}$:

$$(e, g, g^a, g^b, g^c, e(g,g)^{abc}) =_c (e, g, g^a, g^b, g^c, e(g,g)^d)$$

for $d \leftarrow^{\$} \mathbb{Z}_p$.

Given this definition of DBDH, it is easy to show that if CDH is easy then so is DBDH. Indeed if one can compute $g^{ab}$ from $g^a$ and $g^b$, then one can simply evaluate the pairing $e$ at $(g^{ab}, g^c)$ to see wether the last element of the tuple is indeed $e(g, g)^{abc}$.

# 2 Identity-based encryption

## 2.1 IBE scheme

Communicating with someone using a public-key encryption system requires a previous knowledge of his/her public key. We would like to be able to do so but only knowing one's identity, which can be handy in practical situations. This is why introduce Identity-based encryption.

**Identity-based encryption** (or **IBE**)
An **identity-based encryption scheme** is a tuple of four PPT algorithms ($\mathsf{Setup}$, $\mathsf{KeyGen}$, $\mathsf{Enc}$, $\mathsf{Dec}$) such that:

- $\mathsf{Setup}(1^\lambda)$ outputs a pair $(m_{pk}, m_{sk})$ of master public key and master secret key,
- $\mathsf{KeyGen}(m_{sk}, ID)$, for $ID \in \{0,1\}^*$ outputs a security key $s_{ID}$ for identity $ID$,
- $\mathsf{Enc}(m_{pk}, ID, m)$ outputs a ciphertext $ct$,
- $\mathsf{Dec}(s_{ID}, ct)$ outputs a plaintext message $m'$

and such that the following holds:

- (correctness) $\mathbb{P}[\mathsf{Dec}(s_{ID}, \mathsf{Enc}(m_{pk}, ID, m)) = m \mid S] = 1 - negl(\lambda)$
  where $S = \left\{(m_{pk}, m_{sk}) \leftarrow \mathsf{Setup}(1^\lambda) \ \ \& \ \ s_{ID} \leftarrow \mathsf{KeyGen}(m_{sk}, ID)\right\}$

Here we consider that a trusted party is present. It will generate the master public key and the master secret key in order to produce security keys for every ID. Each ID will receive its $sk_{ID}$ and use it to decrypt messages adressed to it.

## 2.2 Security notion

We want to adapt the notion of **IND-CPA** security to IBE schemes.

**IND-ID-CPA** Game:

- **Setup:**
  Challenger runs $(m_{pk}, m_{sk}) \leftarrow^{\$} \mathsf{Setup}(1^\lambda)$.
- **Query phase 1:**
  Adversary $\mathcal{A}$ can adaptively ask secret keys for different IDs, $ID_1, \ldots, ID_g$ and receives $sk_{ID_1}, \ldots, sk_{ID_g}$ from challenger.
- **Challenge phase:**
  Adversary $\mathcal{A}$ picks $(ID^*, m_0, m_1)$ and sends them to the challenger. If $ID^* \neq ID_1, \ldots, ID_g$ challenger replies with $\mathsf{Enc}(m_{pk}, ID^*, m_b)$ for $b$ taken uniformly at random in $\{0, 1\}$, else challenger returns $\perp$.
- **Query phase 2:**
  Same as phase 1 but $ID^*$ cannot be queried.
- **Output:**
  Adversary $\mathcal{A}$ outputs $b'$ and wins if $b = b'$.

We say that an IBE scheme is **IND-ID-CPA secure** if

$$\left| \mathbb{P}[\mathcal{A} \text{ wins }] - \frac{1}{2} \right| = negl(\lambda)$$

# 3 Boneh-Franklin IBE scheme

Let $\mathbb{G}$, $\mathbb{G}_T$ be groups and $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ a pairing. Let $H : \{0, 1\}^* \longrightarrow \mathbb{G}$ denote a random oracle.

The **Boneh-Franklin IBE scheme**:

- $\mathsf{Setup}(1^\lambda)$:
  Generate $(\mathbb{G}, \mathbb{G}_T, e, g)$ and sample $s \leftarrow^{\$} \mathbb{Z}_p$.
  Output $(m_{pk} = g^s, m_{sk} = s)$.
- $\mathsf{KeyGen}(m_{sk}, ID)$:
  Compute $h = H(ID)$
  Output $(s_{ID} = h^s)$
- $\mathsf{Enc}(m_{pk}, ID, m)$:
  Compute $h_{ID} = H(ID)$
  Pick $r \leftarrow^{\$} \mathbb{Z}_p$
  Output $ct = (ct_1, ct_2) = (g^r, e(g^s, h_{ID}^r)m$
- $\mathsf{Dec}(s_{ID}, ct)$:
  Compute $m = \frac{ct_2}{e(ct_1, s_{ID}}$

# References

[1] Joux A. (2000) A One Round Protocol for Tripartite DiffieHellman. In: Bosma W. (eds) Algorithmic Number Theory. ANTS 2000. Lecture Notes in Computer Science, vol 1838. Springer, Berlin, Heidelberg