Compressing Pairing Values

Koray Karabina

University of Waterloo

February 25, 2010

Outline

Bilinear pairings

Let $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, +)$ be additively written cyclic groups

Let $(\mathbb{G}_T, \cdot)$ be a multiplicatively written cyclic group

Assume that $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = n$, and n is prime

Let $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$ and $\mathbb{G}_T = \langle g \rangle$

A bilinear pairing is a function: $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ such that

$$
\begin{aligned}
e(P_1, P_2) &\neq 1 \\
e(a \cdot P_1, P_2) &= e(P_1, P_2)^a \\
e(P_1, b \cdot P_2) &= e(P_1, P_2)^b
\end{aligned}
$$

A Pairing-based crypto application

Scott, 2002: Authenticated ID-based key exchange

Two users A and B with their provable identities ID_A and ID_B

A trusted authority TA

A and B want to agree on a shared key K based on ID_A and ID_B

Setting:

- $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$
- $H : \{0,1\}^* \longrightarrow \mathbb{G}_1$

Interaction with TA:

- A proves her identity ID_A to TA
- B proves his identity ID_B to TA
- TA chooses a secret $s \in [0, n-1]$
- $TA \longrightarrow A : \; P_A = s \cdot H(ID_A)$
- $TA \longrightarrow B : \; P_B = s \cdot H(ID_B)$

ID-based key exchange cont'd

Protocol (IDB-KE):

- $A \longrightarrow B: \; ID_A$ and $B \longrightarrow A: \; ID_B$
- A chooses a secret $a \in [0, n-1]$
- B chooses a secret $b \in [0, n-1]$
- $A \longrightarrow B: \; g_A \; = \; e(P_A, \; H(ID_B))^a \; \in \; \mathbb{G}_T$
- $B \longrightarrow A: \; g_B \; = \; e(H(ID_A), \; P_B))^b \; \in \mathbb{G}_T$
- A computes

$$\begin{aligned} K &= g_B^a = e(H(ID_A), \; s \cdot H(ID_B))^{ba} \\ &= e(H(ID_A), \; H(ID_B))^{sab} \end{aligned}$$

- B computes

$$\begin{aligned} K &= g_A^b = e(s \cdot H(ID_A), \; H(ID_B))^{ab} \\ &= e(H(ID_A), \; H(ID_B))^{sab} \end{aligned}$$

A closer look at IDB-KE

Requirements:

- A (symmetric) bilinear pairing $e : \ \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$
- A hash function $H : \ \{0,1\}^* \longrightarrow \mathbb{G}_1$

Computations from A's point of view:

- A pairing evaluation: $e(P_A,\ H(ID_B)) \in \mathbb{G}_T$
- Exponentiation in $\mathbb{G}_T$: $g_A = e(P_A,\ H(ID_B))^a$ and g_B^a

Exchanged messages:

- g_A and $g_B \ \in \ \mathbb{G}_T$

Security requirements:

- *BDHP* is hard: Given $(\mathbb{G}_1 = \langle P \rangle, aP, bP, sP)$ compute $e(P,P)^{sab}$
- *DLP* is hard: Given $(\mathbb{G}_T = \langle g \rangle, g^a)$ compute a

A concrete setting for IDB-KE

Let $E : y^2 + y = x^3 + x$ be an elliptic curve defined over $\mathbb{F}_{2^{1223}}$

$E(\mathbb{F}_{2^{1223}}) = 5 \cdot n$ where n is a 1221-bit prime

$\mathbb{G}_1 \ = \ E(\mathbb{F}_{2^{1223}})[n]$, the set of n-torsion points of $E(\mathbb{F}_{2^{1223}})$

$\mathbb{G}_T \ = \mu_n \subset \mathbb{F}^*_{2^{4 \cdot 1223}}$, the set of n'th roots of unity

There exists a bilinear pairing $e : \ \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \ \mathbb{G}_T$ and

An efficiently computable hash function $H : \ \{0,1\}^* \longrightarrow \mathbb{G}_1$

- IDB-KE can be realized in this setting at 128-bit security level
- A and B have to exchange 4892-bit messages g_A and $g_B \in \mathbb{G}_T$

Compression/decompression in $\mathbb{G}_T$

Let $q = 2^{1223}$ and $\mu_n \subset \mathbb{F}_{q^4}^*$ be as before

$\mathbb{F}_{q^2}$ is a 2-dim. vector space over $\mathbb{F}_q$ with a basis $\{1, w\}$

$\mathbb{F}_{q^4}$ is a 2-dim. vector space over $\mathbb{F}_{q^2}$ with a basis $\{1, \sigma\}$

$\mathbb{F}_{q^4}$ is a 4-dim. vector space over $\mathbb{F}_q$ with a basis $\{1, w, \sigma, w \cdot \sigma\}$

Therefore, if $g \in \mathbb{F}_{q^4}$ then

$$g = g_0 + g_1 \cdot w + g_2 \cdot \sigma + g_3 \cdot w\sigma$$

g is naturally represented by 4 elements in $\mathbb{F}_q$ (i.e., 4892-bits)

But if $g \in \mu_n \subset \mathbb{F}_{q^4}^*$ then we might hope to use fewer elements as

$$|\mu_n| = n \approx q \ll q^4$$

Compression/decompression in characteristic-two

Let $m = 1223$, $q = 2^m$ and $\mu_n \subset \mathbb{F}^*_{q^4}$

We observe that

$$q + 1 - \sqrt{2q} \;\equiv\; q^2 + 1 \;\equiv\; 0 \pmod{n}$$

That is, if $g \in \mu_n$ then

$$g^{q+1-\sqrt{2q}} \;=\; g^{q^2+1} \;=\; 1$$

Manipulating these relations we obtain compression/decompression maps

$$\mathcal{C} : \mu_n \longrightarrow \mathbb{F}_q \times \{0,1\}, \quad \mathcal{C}(g_0, g_1, g_2, g_3) = (c_g,\; i_g)$$
$$\mathcal{D} : \mathbb{F}_q \times \{0,1\} \longrightarrow \mu_n, \quad \mathcal{D}(c_g, i_g) = (g_0, g_1, g_2, g_3)$$

IDB-KE

- $A \longrightarrow B: \; ID_A$ and $B \longrightarrow A: \; ID_B$
- A chooses a secret $a \in [0, n-1]$
- B choses a secret $b \in [0, n-1]$
- $A \longrightarrow B: \; g_A \; = \; e(P_A, \; H(ID_B))^a \; \in \; \mathbb{G}_T$
- $B \longrightarrow A: \; g_B \; = \; e(H(ID_A), \; P_B))^b \; \in \mathbb{G}_T$
- A computes

$$\begin{aligned} K &= g_B^a = e(H(ID_A), \; s \cdot H(ID_B))^{ba} \\ &= e(H(ID_A), \; H(ID_B))^{sab} \end{aligned}$$

- B computes

$$\begin{aligned} K &= g_A^b = e(s \cdot H(ID_A), \; H(ID_B))^{ab} \\ &= e(H(ID_A), \; H(ID_B))^{sab} \end{aligned}$$

Modified IDB-KE

- $A \longrightarrow B: \ ID_A$ and $B \longrightarrow A: \ ID_B$
- A chooses a secret $a \in [0, n-1]$
- B choses a secret $b \in [0, n-1]$
- $A \longrightarrow B: \ \mathcal{C}(g_A) \ = \ \mathcal{C}(e(P_A, \ H(ID_B))^a) \ \in \ \mathbb{F}_q$
- $B \longrightarrow A: \ \mathcal{C}(g_B) \ = \ \mathcal{C}(e(H(ID_A), \ P_B))^b) \ \in \mathbb{F}_q$
- A computes $\mathcal{D}(\mathcal{C}(g_B))$ and

$$\begin{aligned} K &= g_B^a = e(H(ID_A), \ s \cdot H(ID_B))^{ba} \\ &= e(H(ID_A), \ H(ID_B))^{sab} \end{aligned}$$

- B computes $\mathcal{D}(\mathcal{C}(g_A))$

$$\begin{aligned} K &= g_A^b = e(s \cdot H(ID_A), \ H(ID_B))^{ab} \\ &= e(H(ID_A), \ H(ID_B))^{sab} \end{aligned}$$

Concluding remarks

- We have achieved factor-4 and factor-6 compression of pairing values
  - Our methods yield the ideal compression
- We have designed exponentiation algorithms that work with the compressed representations
  - The algorithms are faster than conventional exponentiation algorithms