Cryptographic Trilinear Maps

Lukas Zobernig

SRC 2018

Introduction

Working on **Cryptography**, **Obfuscation** $\Rightarrow$ interested in multilinear maps as primitive

Outline

- DLP & Diffie-Hellman
- Weil Pairing
- Application of the Weil Pairing
- Candidate Trilinear Map

DLP & Diffie-Hellman

Definition (Discrete Logarithm Problem (DLP))

We say the **DLP** is hard in a group G if given a pair $g, g^x \in G$ it is hard to find $x \in \mathbb{Z}$.

Two-Party Key exchange [Diffie-Hellman, 1976]

Alice and Bob want to exchange a key (given a public group and generator g):

- Alice publishes $A = g^a$
- Bob publishes $B = g^b$

Both are able to calculate a shared key:

- Alice finds $k = B^a = g^{ab}$
- Bob finds $k = A^b = g^{ab}$

Weil Pairing on Curves

Definition (Weil pairing [Weil, 1940])

On an projective, non-singular curve C of genus $g > 0$ exists a non-degenerate, Galois invariant pairing

$$e_m : C[m] \times C[m] \to \mu_m$$

such that $\forall P, Q \in C[m]$ and $\forall a \in \mathbb{Z}$:

$$\begin{aligned} e_m(aP, Q) = e_m(P, aQ) &= e_m(P, Q)^a, \\ e_m(P, P) &= 1 \end{aligned}$$

- $C[m] = \{P \in C | mP = 0\}$ is the m-torsion group
- μ_m is the group of m-th roots of unity ($\cong$ cyclic group C_m)
- Pairing can be **efficiently** computed [Miller, 1986]

Pairing Based Cryptography

Three-Party Key Exchange [Joux, 2000]

Alice, Bob, and Charlie want to exchange a key (given a public curve and point P):

- Alice publishes $A = aP$
- Bob publishes $B = bP$
- Charlie publishes $C = cP$

Everyone is able to calculate a shared key:

- Alice finds $k = e(B, C)^a = e(P, P)^{abc}$
- Bob finds $k = e(A, C)^b = e(P, P)^{abc}$
- Charlie finds $k = e(A, B)^c = e(P, P)^{abc}$

Hardness assumption DLP: given points P and $Q = xP$ it is hard to find x

Multilinear Maps

Definition (Cryptographic n-linear map)

A map

$$f : G_1 \times \cdots \times G_n \to G_T$$

for groups G_i, G_T with hard DLP such that $\forall g_i \in G_i$ and $\forall a_i \in \mathbb{Z}$:

$$f(a_1 g_1, \cdots, a_n g_n) = f(g_1, \cdots, g_n)^{a_1 \cdots a_n}.$$

Major open problem, such maps would give:

- Exciting new cryptographic primitives
- **Obfuscation** (requires at least trilinear map)

[Boneh, Silverberg, 2003] suggest problems with generalising Weil pairing $\Rightarrow$ **surprise** construction from Weil pairing [Huang, 2017]

Candidate Trilinear Map

Explicit construction

- Start with Weil pairing on (supersingular) high-genus curve C
- Define trilinear map

$$\begin{aligned} f_m : C[m] \times C[m] \times \mathrm{End}(C) &\to \mu_m, \\ (P, Q, \varphi_c) &\mapsto e_m(P, \varphi_c(Q)) \end{aligned}$$

- Endomorphism φ_c encodes $c \in \mathbb{Z}$: $\varphi_c = c + x\lambda$ for random $x \in \mathbb{Z}$, fixed $\lambda \in \mathrm{End}(C)$
- Points $A = \lambda(B)$ for random B are public and used to encode $a, b \in \mathbb{Z}$ via $P = aA$, $Q = bB$

Candidate Trilinear Map

Pitfalls

- **Private encoding**: recover c from φ_c if λ known $\Rightarrow$ λ needs to be private, not everyone can encode in $\mathrm{End}(C)$
- Possible **algebraic attacks**:
  - by evaluating φ_c on points in $C[m']$ for small m'
  - if $a, b, c \in \mathbb{Z}$ are related $\Rightarrow$ may only encode *uniform* elements
- Current obfuscation constructions at least have *some* algebraic relations on a, b, c
- Problematic to efficiently represent φ_c ($\deg \varphi_c$ **large**)

Important research question: how hard is the DLP in $\mathrm{End}(C)$?

Thank you for your attention!