T-79.515 Cryptography: Special Topics / Pairing-based Cryptography: Bilinear Pairings

Sven Laur

January 22, 2003

1 Introduction

Since early days of public cryptography intractability of discrete logarithm has been a cornerstone of many encryption, signing and key agreement schemes. Many mathematicians have tackled the problem. Bilinear pairings where first used to reduce discrete logarithm problem in elliptic curve to finite field logarithm via MOV reduction [MOV]. Wail and Tate pairings enabled also to solve Decisional Diffie-Hellman problem in supersingular elliptic curves. First positive use of bilinear pairings was three-party Diffie-Hellman key agreement proposed by Joux[J00]. Since then several interesting ideas where proposed from signing and authentication to key escrow.

In the following we give a brief survey about basic properties of bilinear pairing and security issues around it which bases on [M02] and [BF01]. We state Bilinear Diffie-Hellman problem and explore connections between it and tree-party Diffie-Hellman protocol.

2 Definitions and implications

Definition 1. *Let $(G_1;+)$ be a cyclic additive group of order p and $(G_2;\cdot)$ a multiplicative group of order p where p is a prime*[1]*. A bilinear pairing on (G_1, G_2) is a function $\hat{e} : G_1 \times G_1 \to G_2$ which has following properties:*

1. *it is additive*

$$\forall R, S, T \in G_1 : \ \hat{e}(R+S, T) = \hat{e}(R, T) \cdot \hat{e}(S, T)$$
$$\forall R, S, T \in G_1 : \ \hat{e}(R, S+T) = \hat{e}(R, S) \cdot \hat{e}(R, T)$$

2. *it is non-constant*

$$\forall R \in G_1^* : \ e(R, R) \neq 1$$

3. *for all elements $R, S \in G_1$ pairing $\hat{e}(R, S)$ can be efficiently calculated*[2]*.*

Simple implications from the definition:

- Zero to one property
$$\hat{e}(S, 0) = \hat{e}(0, S) = 1.$$
- Minus to inverse property
$$\hat{e}(-S, T) = \hat{e}(S, -T) = \hat{e}(S, T)^{-1}.$$
- Bilinearity over $\mathbb{Z}$ and $\mathbb{Z}_p$
$$\forall a, b \in \mathbb{Z} : \ \hat{e}(aS, bT) = \hat{e}(S, T)^{ab}.$$
- Symmetry
$$\hat{e}(S, T) = \hat{e}(aP, bP) = \hat{e}(P, P)^{ab} = \hat{e}(T, S).$$
- Non-degeneracy
$$\forall T \in G_1 : \ \hat{e}(S, T) = 1 \quad \Rightarrow \quad S = 0.$$

Let P be the generator of G_1 then $\hat{e}$ is completely defined with the value $g = \hat{e}(P, P)$ and all values are in $\langle g \rangle$. As the order of G_2 is prime number it has only to subgroups $\{1\}$ and G_2, thus g is a generator of G_2. Bilinearity also means that second condition is equivalent to $\hat{e}(P, P) \neq 1$ because$\hat{e}(aP, aP) = g^{a^2}$ and

$$a^2 \equiv 0 \mod p \ \Rightarrow \ a \equiv 0 \mod p.$$

Non-degenerancy property follows from the bilinearity. Let $S = aP, \ a \in \mathbb{Z}_p$ then $1 = \hat{e}(S, P) = g^a$. As g is a generator of G_2, the claim follows.

[1] The restrictions are quite weird from the algebraic point of view

[2] Usually this means a polynomial time bound.

General form of bilinear pairings. As a pairing is completely defined with $g = \hat{e}(P,P)$ then it is easy to see

$$\hat{e}(S,T) = g^{\log_P S \log_P T}$$

so pairing and discrete logarithm are intimately connected.

Connection with scalar products. As G_1 is a cyclic group we can watch it as a vector space over finite field $\mathbb{F}_p$. As $G_2 \cong \mathbb{Z}_p$ we can simply define exponentiation on G_2 so that $G_2 \cong \mathbb{F}_q$ thus we can think of a bilinear pairing as a scalar product.

Corollary 1. *Bilinear pairings are in one-to-one correspondence with scalar products. Thus there is exactly $p-1$ different bilinear pairings.*

Proof. Let $\delta : G_2 \to \mathbb{F}_p$ be the isomorphism then clearly $\delta \circ \hat{e}$ is scalar product because it is bilinear and non-constant. Let $s : G_1 \times G_1 \to \mathbb{F}_p$ be the scalar product then $\delta^{-1} \circ s$ is a pairing because $s(P,P) \neq 0$.

As the pairing is determined with $\hat{e}(P,P)$ and there is exactly $p-1$ generators in $\mathbb{Z}_p$, the claim follows. □

Example 1. *If $(G_2;\cdot) = (\mathbb{Z}_p;+)$ then resulting pairing is either inefficient or discrete logarithm problem is easy in G_1 because*

$$\begin{aligned}\hat{e}(xP,P) =& \hat{e}(P,P) + \cdots + \hat{e}(xP,P) = x\hat{e}(P,P)\\ \Rightarrow& \qquad \log_P Q = \frac{\hat{e}(Q,P)}{\hat{e}(P,P)}.\end{aligned}$$

3 Various reductions

It is quite easy to generalize the first example to formula

$$a = \log_P S = \log_g \hat{e}(P,S), \qquad g = \hat{e}(P,P)$$

By bilinearity

$$1 = \hat{e}(P,S)\hat{e}(P,aP)^{-1} = \hat{e}(P,S-aP)$$

and thus by non-degeneracy $S = aP$.

Corollary 2 (MOV-reduction). *If (G_1,G_2) have a bilinear pairing $\hat{e}$ then discrete logarithm problem in G_1 is no harder than in G_2*

Proof. This follows directly from the formula. □

Definition 2. *The Decision Diffie-Hellman problem(DDH) is to distinguish between two distributions $\langle P,aP,bP,abP\rangle$ and $\langle P,aP,bP,cP\rangle$ where $a,b,c \in \mathbb{Z}_n$ are picked randomly and n is the group order.*

Corollary 3. *If there exist bilinear pairing between G_1 and G_2 then DDH is easy in G_1.*

Proof. As $\hat{e}(aP,bP) = \hat{e}(P,abP)$ and $\hat{e}$ is non-degenerative

$$ab \equiv c \mod p \qquad \Leftrightarrow \qquad \hat{e}(aP,bP) = \hat{e}(P,cP)$$

□

The opposite is also true under some additional conditions. Clearly if there exist efficiently computable isomorphism $\delta : G_2 \to G_1$ then one can transform DDH from G_2 to G_1 and solve DDH in G_1 which is easy. Next we introduce all candidates for a such isomorphism.

Proposition 1. *Let $\hat{e}$ be the bilinear pairing then functions $f_Q : G_1 \to G_2$ defined by $f_Q(R) = \hat{e}(Q,R)$ form a set of all group homomorphisms $\mathrm{Hom}(G_1,G_2)$. And even more if $Q \neq 0$ then f_Q is an isomorphism.*

Proof. Clearly bilinearity and zero to one property assure f_Q is preserves group operations and thus f_Q is group homomorphism. Let P be the generator of G_1 then we have p distinct homomorphisms

$$f_{aP}(P) = \hat{e}(aP,P) = g^a,\ g = \hat{e}(P,P),\ a \in \mathbb{Z}_p$$

because g is a generator of G_2. As zero homomorphism is only non-isomorphism the claim is proved. □

Corollary 4. *If there exists $Q \in G_1^*$ such that equation $\hat{e}(Q,X) = h$ can be efficiently solved for arbitrary group element $h \in G_2$ then DDH is also easy in G_2 and discrete logarithm problems in G_1 and G_2 are polynomially equivalent.*

Proof. Solving the equation is equivalent to inverting the isomorphism f_Q and thus we have computationally efficient isomorphism f_Q^{-1}. The Diffie-Hellman condition in G_2 is equivalent

$$g^a \cdot g^b = g^c \quad \Leftrightarrow \quad f_Q^{-1}(g^a)f_Q^{-1}(g^b) = f_Q^{-1}(g^c)$$

and as DDH is easy in G_1 the claim follows. In the same way one can reduce discrete logarithm problem in G_2 to discrete logarithm problem in G_1. $\square$

Corollary 5. *If DDH is hard in G_2 then discrete logarithm problem may be harder in G_2 than in G_1.*

4 Diffie-Hellman problem

As DDH is easy for G_1 when we have a bilinear pairing $\hat{e}$ the it is natural to wonder if Diffie-Hellman problem itself is simple.

Definition 3. *The Computational Diffie-Hellman problem(CDH) is to calculate abP if P, aP and bP are given and $P \in G$, $a, b \in \mathbb{Z}_n$ are picked randomly and n is the group order.*

Following results obtained in [MW98] give a partial answer. They prove that under certain conditions ability to compute efficiently CDH implies ability to solve efficiently discrete logarithm. In the following we give a brief sketch of proof.

Definition 4. *Let g be the generator of group G and $p \in \mathbb{P}$ a divisor of group order. We say that element $a = g^{y'}$ represents implicitly element $y \in \mathbb{Z}_p$ if $y' \equiv y \mod p$ and write $y \rightsquigarrow a$.*

It is easy to understand that ability to solve computational Diffie-Hellman problem implies that we can do all algebraic $\mathbb{F}_p$ operations with implicit representations

$$\begin{aligned} y = z &\Leftrightarrow a^{\frac{|G|}{p}} = b^{\frac{|G|}{p}} \\ y + z &\rightsquigarrow a \cdot b \\ y - z &\rightsquigarrow a \cdot b^{|G|-1} \\ z \cdot y &\rightsquigarrow DH(a, b) \end{aligned}$$

where $y \rightsquigarrow a$ and $z \rightsquigarrow b$. Next we need an auxiliary group which is intended for resolving implicit representations ie. calculating discrete logarithm. The auxiliary group must meet following conditions.

Definition 5. *Partially defined parametric embedding $\mathcal{E}mb : \mathbb{F}_p \times F_p \to H$ is probababilistically α-efficient if it meets four conditions.*

1. *Calculation $\mathcal{E}mb(x, e)$ involves only algebraic operations in $\mathbb{F}_p$ and either succeeds or failure can be detected by algebraic operations;*

2. *If we calculate $\mathcal{E}mb(x, e)$ by taking e randomly from $\mathbb{F}_p$ until the algorithm does not fail then expected running time is at most α algebraic operations in $\mathbb{F}_p$.*

3. *We have an inverse map of $\mathcal{E}mb$ which extracts elements $\mathcal{E}xt : H \times \mathbb{F}_p \to \mathbb{F}_p$ so $\mathcal{E}xt(\mathcal{E}mb(x, e), e) = x$ if $\mathcal{E}mb(x, e) \in H$.*

4. *Extraction algorithm does at most α calculations in $\mathbb{F}_p$.*

Definition 6. *A group is defined strongly $(m.\alpha)$-algebraically over $\mathbb{F}_p$ if following three conditions are met*

1. *elements of H can be represented as m-tuples of elements of $\mathbb{F}_p$;*

2. *group operation must be carried out at most α algebraic operations;*

3. *exists probabilistically α-efficient partially defined parametric embedding $\mathcal{E}mb : \mathbb{F}_p \times F_p \to H$.*

Now we can state when CDH is equivalent for calculating discrete logarithm.

Theorem 2. *Let G_1 be the cyclic group of order $p \in \mathbb{P}$. If there exists a finite Abelian group H such that H is strongly (m, α)-algebraically defined over $\mathbb{F}_p$ and order $|H|$ is B smooth and known and m, α, B are* **poly**$(\log |G|)$ *then breaking the CDH with a generator P is probabilistic polynomial-time equivalent to computing discrete logarithms in G to the base g.*

Proof sketch

The following four stage algorithm creates isomorphism $\delta : (G; +) \to (\mathbb{Z}_p; +)$

1. Calculate implicit representation $\mathcal{E}mb(x, e) = c$.

2. Calculate discrete logarithm of c using only the implicit representation.

3. Calculate explicit representation of c by exponentiation.

4. Calculate $x = \mathcal{E}xt(\mathcal{E}mb(c, e), e)$.

Explanation It is possible to detect failure in the first stage even if we calculate with implicit values because algorithm uses only algebraic operations. The most costly stage is second stage where discrete logarithm is calculated by incremental revealing like in the Pohlig-Hellman algorithm. The exponentiation and extraction step are straightforward. The construction uses heavily implicit multiplication by querying the Diffie-Hellman oracle.

□

Final remark: Usually the suitable auxiliary group is an elliptic curve over $\mathbb{F}_p$ with required parameters. The construction of such auxiliary groups is direct security proof if discrete logarithm is believed to be hard in G_1. Such proofs constructions are presented in [JN01].

5 Bilinear Diffie-Hellman Problem

If we assume a little bit more than computational Diffie-Hellman is hard then one can construct various protocols and primitives. Following notion is a cornerstone to various cryptographic applications.

Definition 7. *Let $\hat{e} : G_1 \to G_2$ be a bilinear pairing then Bilinear Diffie-Hellman problem(BDH) is as follows: Given $\langle P, aP, bP, cP\rangle$ for some random $a, b, c \in \mathbb{Z}_p^*$, compute $W = \hat{e}(P, P)^{abc} \in G_2$. The algorithm $\mathcal{A}$ has an advantage ϵ in solving BDH if*

$$\Pr\left[\mathcal{A}(P, aP, bP, cP) = W\right] \geq \epsilon$$

Next result summarize that BDH is harder than CDH.

Corollary 6. *Let $\hat{e}$ be a bilinear pairing on (G_1, G_2). If computational DH is easy in G_1 or G_2 then BDH is also easy.*

PROOF. If CDH is easy in G_1 then one can calculate abP from aP and bP and thus also

$$\hat{e}(P, P)^{abc} = \hat{e}(abP, cP).$$

If CDH is easy in G_2 then one can calculate elements $g = \hat{e}(P, P)$, $g^{ab} = \hat{e}(aP, bP)$ and $g^c = \hat{e}(P, cP)$ and then solve the corresponding Diffie-Hellman problem in G_2 and obtain g^{abc}. □

Nothing else is known about hardness of BDH. If one believes in BDH the one can construct several interesting protocols and primitives:

- one round 3-party Diffie-Hellman key agreement
- key escrow protocol for ElGamal encryption scheme
- short signatures
- identity based encryption
- identity based signatures
- various multi-party signing protocols

6 Three-party key agreements

Standard three-party Diffie-Hellman key agreement has too rounds and must be interactive.

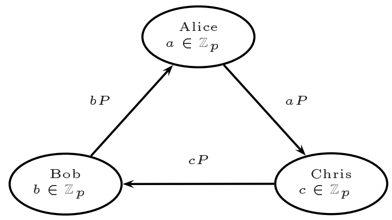

Figure 1: First round of standard 3-party DH

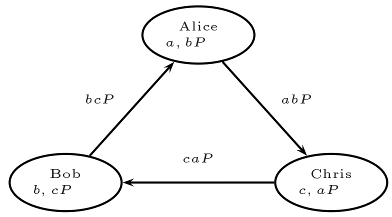

Figure 2: Second round of standard 3-party DH

Corollary 7. *Standard three-party Diffie-Hellman key agreement protocol is secure against passive adversary if and only if following problem is computationally hard in group G. Given $P, aP, bP, cP, abP, acP, bcP$ calculate $abcP$.*

PROOF. It is evident □

Three-party one round Diffie-Hellman key agreement: The use of a bilinear pairing makes it possible to reduce number of rounds to one ie. make the scheme static. The agreed key is

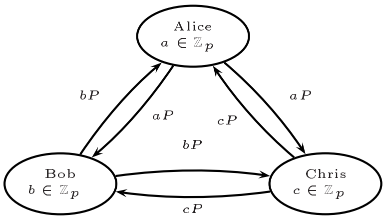

Figure 3: Static 3-party DH

$K = \hat{e}(P,P)^{abc} = \hat{e}(aP,bP)^c$ which makes it possible to obtain in the one run. This makes it possible to send broadcast messages. The security depends directly on BDH.

Corollary 8. *Three-party Diffie-Hellman key agreement protocol is secure against passive adversary if and only BDH hard in group G.*

PROOF. It is evident. □

It is possible to engage man-in-the-middle attack:

- Dan is impersonating Alice to Bob and Chris;
- Dan is impersonating Chris and Bob to Alice.

Thus authentication mechanism for the keys is needed.

References

[M02] A. Menses Constructive Applications of the Weil and Tate Pairings, *unpublished lecture notes*, 2002.

[MOV] A. Menezes, T. Okamoto and S. Vanstone, Reducing elliptic curve logarithms in a finite field,*IEEE Transactions on Information Theory*, Vol.39, pp.1639-1646, 1993.

[BF01] D. Boneh, and M. Franklin, Identity-based encryption from the Weil pairing, *Crypto'2001, Lecture Notes in Computer Science, Springer-Verlag*, 2001.

[J00] A. Joux, A one-round protocol for tripartite Diffie-Hellman, *ANTS-IV, Lecture Notes in Computer Science, Springer-Verlag*, 2000.

[JN01] A. Joux, and K. Nguyen, Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups *Cryptology ePrint Archive*, 2001 **http://eprint.iacr.org**

[MW98] U.M. Maurer, and S. Wolf, The Relationship Between Breaking the Diffie-Hellman Protocol and Computing Discrete Logarithms, *SIAM Journal on Computing*, 1999.