A NOTE ON THE BI-LINEAR DIFFIE-HELLMAN ASSUMPTION

YACOV YACOBI

ABSTRACT. The Bi-linear Diffie-Hellman (BDH) intractability assumption is required to establish the security of new Weil-pairing based cryptosystems. BDH is reducible to most of the older believed-to-be-hard discrete-log problems and DH problems, but there is no known reduction from any of those problems to BDH. Let the bilinear mapping be $\hat{e} : G_1 \times G_1 \to G_2$, where G_1 and G_2 are cyclic groups. We show that a *many-one* reduction from any of the relevant problems to BDH has to include an efficient mapping $\varphi : G_2 \to G_1$, where $\varphi(g^x) = f(x)P$. Here $g \in G_2$, and $P \in G_1$ are generators of the corresponding cyclic groups. The function φ must be used in the reduction either before or after the call to oracle BDH. We show that if $f(x) = ax^n + b$ for any constants a, b, n, then φ could be used as an oracle for a probabilistic polynomial time solution for Decision Diffie-Hellman in G_2. Thus such a reduction is unlikely.

1. Introduction

The bi-linear Diffie-Hellman (BDH) intractability assumption is required to establish the security of new Weil-pairing based cryptosystems. BDH is reducible to most of the older believed-to-be-hard discrete-log problems and DH problems, but there is no known reduction from any of those problems to BDH, thus we have no evidence that the BDH problem is indeed hard. Let the bi-linear mapping be $\hat{e} : G_1 \times G_1 \to G_2$, where G_1 and G_2 are cyclic groups. Here G_1 is the cyclic group of points on elliptic curve (EC), and we follow the usual convention of calling the basic operation on EC "addition," its aggregate "multiplication," and unlike the convention, we call the inverse of multiplication on EC "division" (the convention is "discrete-log"). G_2 is a group of polynomials of degree 1 over some basic field, and the basic operation is multiplication modulo some degree 2 polynomial.

Let $g \in G_2$, and $P \in G_1$ be generators of the corresponding cyclic groups.

We observe that any polynomial *many-one* reduction (see eg [HU], pp. 212) from the relevant candidate hard problems to BDH must include an efficient mapping $\varphi : G_2 \to G_1$, where $\varphi(g^x) = f(x)P$. The function φ must be used in the reduction either before or after the call to oracle BDH. We also show that if efficient φ: $G_2 \to G_1$ exists with

1. (1) $f(x) = x$ then it could be used as an oracle to solve Computational Diffie-Hellman in G_2 in deterministic polynomial time.
2. (2) $f(x) = ax + b$, where $a \in \mathbb{Z}_q^*, b \in \mathbb{Z}_q$ then φ can be used as an oracle to solve Decision DH in G_2 in deterministic polynomial time.
3. (3) $f(x) = ax^n + b$, where $a \in \mathbb{Z}_q^*, b \in \mathbb{Z}_q$ and $0 < n < q - 1$, then φ could be used as an oracle to solve Decision DH in G_2 in probabilistic polynomial time.

Date: May 20, 2002.

Key words and phrases. Weil-pairing, bi-linear Diffie-Hellman, ID-based encryption.

Thus in all the above cases the existence of a reduction is unlikely.

2. Background

The definitions in this section are taken from [BF]. Let G_1 and G_2 be two cyclic groups of order q for some large prime q. We use $e(P,Q)$ and $\hat{e}(P,Q)$ for the unmodified and modified Weil-pairing, respectively. The latter is an efficient bi-linear mapping $\hat{e} : G_1 \times G_1 \to G_2$. This map satisfies the following properties:

(1) $\hat{e}(aP, bQ) = \hat{e}(P,Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in \mathbb{Z}$.
(2) If P is a generator of G_1 then $g = \hat{e}(P,P)$ is a generator of G_2. In the (unmodified) weil-pairing this is not true, since $e(P,P) = 1$ for every P.

Without going into details of the definition of Weil-pairing we give some details of the modified Weil-pairing, which builds on top of Weil-pairing. Let p be a prime satisfying $p = 6q - 1$ for some large prime q (so in particular $p = 2 \bmod 3$). Let E be the elliptic curve defined by the equation $y^2 = x^3 + 1$ over $\mathbb{F}_p$. We have the following properties:

(1) The mapping $x \to x^3 + 1$ induces a permutation on $\mathbb{F}_p$ hence the group defined by the elliptic curve E over the group $\mathbb{F}_p$, denoted $E(\mathbb{F}_p)$, contains $p+1$ points. Let $G_1 \subset E(\mathbb{F}_p)$ denote the group of points of order $q = (p+1)/6$, and let $P \in G_1$ be a generator of G_1.
(2) Let $1 \neq \zeta \in \mathbb{F}_{p^2}$ be a solution of $x^3 - 1 = 0$ in $\mathbb{F}_{p^2}$. The map $\phi(x,y) = (\zeta x, y)$ is an automorphism of the group of points on the curve $E(\mathbb{F}_{p^2})$.
(3) $P \in E(\mathbb{F}_p)$ is linearly independent of $\phi(P) \in E(\mathbb{F}_{p^2})$. They generate a group isomorphic to $\mathbb{Z}_q \times \mathbb{Z}_q$, denoted $E[q]$.
(4) Let G_2 denote the subgroup of $\mathbb{F}_{p^2}$ containing all the elements of order $q = (p+1)/6$. The Weil-pairing on the curve $E/\mathbb{F}_{p^2}$ is a mapping $e : E[q] \times E[q] \to G_2$.
(5) The modified Weil-pairing is a bi-linear mapping $\hat{e} : G_1 \times G_1 \to G_2$, defined by $\hat{e}(P,Q) = e(P, \phi(Q))$.

The unmodified weil-pairing has the following symmetry: For all points $P, Q \in G_1$, $e(P,Q) = e(Q,P)^{-1}$, while the modified Weil-pairing has the simpler symmetry: $\hat{e}(P,Q) = \hat{e}(Q,P)$ (let R be a generator of G_1, then $\exists x, y$, integers, such that $P = xR, Q = yR$. $\hat{e}(P,Q) = \hat{e}(xR, yR) = \hat{e}(R,R)^{xy} = \hat{e}(yR, xR) = \hat{e}(Q,P)$).[1]

3. Believed-to-be-hard problems and known reductions

3.1. Problem definitions. In the definitions below DDH abbreviates Decision Diffie-Hellman, CDH stands for Computational Diffie-Hellman, and DL means Discrete-Log. Each appears with a subscript denoting the group in which it is defined. In addition we have the following two problems: Bi-linear Pairing Inversion (BPI), and Bi-linear Diffie-Hellman (BDH).

[1] The following is a tempting bogus proof to the contrary: For all points P, Q on the curve $\hat{e}(P,Q) = \hat{e}(Q,P)$, meaning $e(P,\phi(Q)) = e(Q,\phi(P))$. Let $Q = \phi(P)$. Then $e(P,\phi^2(P)) = e(\phi(P),\phi(P))$. But the right hand side equals 1, hence for all P, $e(P,\phi^2(P)) = 1$.But $\phi^2()$ and $\phi()$ are algebraically indistinguishable in G_1, hence for all $P, e(P,\phi(P)) = 1$, ie $\hat{e}(P,P) = 1$, contradicting the fact that if P has order q in G_1 then $\hat{e}(P,P)$ has order q in G_2. The problem with this "proof" is that the assignment $Q = \phi(P)$ is illegal. Q must be in $E(\mathbb{F}_p)$, while $\phi(P) \in E(\mathbb{F}_{p^2}) \backslash E(\mathbb{F}_p)$.

(1) DDH_{G_1}: Given: $P, xP, yP, zP \in G_1$; Decide: $xyP = zP$?
(2) DDH_{G_2}: Given: $g, g^x, g^y, g^z \in G_2$; Decide: $g^{xy} = g^z$?
(3) CDH_{G_1}: Given: $P, xP, yP \in G_{\mathbf{1}}$; Find: xyP.
(4) CDH_{G_2}: Given: $g, g^x, g^y \in G_2$; Find: g^{xy}.
(5) DL_{G_1}: Given: $P, xP \in G_{\mathbf{1}}$; Find: x.
(6) DL_{G_2}: Given: $g, g^x \in G_2$; Find: x.
(7) BPI: Given: $Q \in G_{\mathbf{1}}$, $\hat{e}(P, Q) \in G_{\mathbf{2}}$; Find: P .
(8) BDH: Given: $P, xP, yP, zP \in G_1$; Find: $\hat{e}(P, P)^{xyz}$.

3.2. **Reductions.** We give only sketches that show worst case behavior. The more interesting cases of average case complexity usually requires more details, but discrete log and Diffie-Hellman problems have "self reducibility" property that makes this extra step painless: In all of those cases a problem is hard in the worst case iff it is hard on the average. We use $B \Rightarrow A$ to denote a polynomial Turing reduction from problem B to problem A (so A is the oracle). Note that although the known reductions presented here are Turing reductions, the main claim is narrowed down to many-one reductions. It is an open problem if it holds for Turing reductions. Most of the reductions below are many-one reductions (with the exception of reductions 5 and 7). At one point (Th. 1) we use this notation for a randomized reduction, but we state so explicitly. All the reductions below are trivial and have logarithmic complexity (most of them are actually simple isomorphisms).

(1) DDH_{G_1} is easy [JN]: Check whether $\hat{e}(xP, yP) = \hat{e}(P, zP)$
(2) $DL_{\mathbf{G_1}} \Rightarrow DL_{\mathbf{G_2}}$ [MOV]: Let $g = \hat{e}(P, P) \in G_2$. Then $g^x = \hat{e}(P, xP)$, hence an oracle for $DL_{\mathbf{G_2}}$, which is given g, g^x as above, also solves the given instance of $DL_{\mathbf{G_1}}$.
(3) $CDH_{\mathbf{G_1}} \Rightarrow DL_{\mathbf{G_1}}$: Given $P, xP \in G_1$, oracle $DL_{\mathbf{G_1}}$ outputs x. Compute $x(yP) = xyP$.
(4) $BDH \Rightarrow CDH_{\mathbf{G_1}}$: Give oracle $CDH_{\mathbf{G_1}}$ input P, xP, yP, and get back xyP. Then compute $\hat{e}(xyP, zP) = \hat{e}(P, P)^{xyz}$.
(5) $BDH \Rightarrow CDH_{\mathbf{G_2}}$: Let $g = \hat{e}(P, P) \in G_2$. Then $g^x = \hat{e}(P, xP), g^y = \hat{e}(P, yP)$, and $g^z = \hat{e}(P, zP)$. Use oracle $CDH_{\mathbf{G_2}}$ to compute g^{xy} given g^x and g^y, then use it again to compute g^{xyz} given g^{xy} and g^z.
(6) $DDH_{\mathbf{G_2}} \Rightarrow CDH_{\mathbf{G_2}}$: Use oracle $CDH_{\mathbf{G_2}}$ to compute g^{xy} and compare to g^z.
(7) $CDH_{\mathbf{G_2}} \Rightarrow BPI$: Pick any $Q \neq 1, Q \in G_1$. Use oracle BPI to find $P \in G_1$ such that $g = \hat{e}(P, Q)$. Then $g^x = \hat{e}(xP, Q)$. Use the oracle to find xP. Likewise $g^y = \hat{e}(yQ, P)$. Use the oracle to find yQ (with P now the fixed-point). Finally compute $g^{xy} = \hat{e}(xP, yQ)$.
(8) $BPI \Rightarrow DL_{\mathbf{G_2}}$: Let Q be the fixed element given in the input of BPI. Define $g = \hat{e}(Q, Q)$ and $g^x = \hat{e}(P, Q)$. It follows that $P = xQ$. Feed g and g^x to oracle $DL_{\mathbf{G_2}}$ and get back x. Compute $P = xQ$.

For more relevant reductions see [ERV]. We summarize the above reductions as follows (here slanted arrows mean reductions too):

$$\begin{array}{ccccccc} BDH & \Rightarrow & CDH_{G_1} & \Rightarrow & DL_{G_1} & \Rightarrow & DL_{G_2} \\ & \searrow & & & & \nearrow & \\ DDH_{G_2} & \Rightarrow & CDH_{G_2} & \Rightarrow & BPI & & \end{array}$$

4. Is there a reduction to BDH?

There is no known reduction from any of those problems to BDH. Note that a reduction from DL_{G_2} to BDH would in particular imply, a reduction from DL_{G_2} to DL_{G_1}, an open problem since 1985. Let the bi-linear mapping be $\hat{e} : G_1 \times G_1 \rightarrow G_2$. We observe that any polynomial *many-one* reduction (see eg [HU], pp. 212) from any of the above relevant problems to BDH requires an efficient mapping $\varphi : G_2 \rightarrow G_1$, where $\varphi(g^x) = f(x)P$, so that $g \in G_2$ and $P \in G_1$ are generators of the corresponding cyclic groups. The function φ must be used in the reduction either before or after the call to oracle BDH. We show that if $f(x) = ax^n + b$ for constants a, b, n, then φ could be used as an oracle for a probabilistic polynomial time solution to DDH_{G_2}.Thus such a reduction is unlikely.

Consider a many-one reduction $CDH_{G_1} \Rightarrow BDH$. First notice a domain "mismatch" between CDH_{G_1} and BDH. They both have their inputs in G_1, however, oracle $BDH's$ answer is in G_2, and in a many-one reduction must be mapped back into G_1 to be of use to CDH_{G_1}. Likewise there is a mismatch that needs an efficient mapping from G_2 to G_1 in a reduction from CDH_{G_2} to BDH (here the mismatch is in the input domains). In all the candidate problems defined above we have at least one of the above mismatches.

An efficient mapping $\varphi : G_2 \rightarrow G_1$ must cover any point in G_2, so it is natural to represent all those points using one generator g of G_2, and represent every point as some power g^x where $x \in [0, q)$. It is also true that every point in G_1 is the answer to some instance of the problem CDH_{G_1}. Hence there exists some function f on $\mathbb{Z}_q$ such that $\varphi(g^x) = f(x)P$, with P a generator in G_1. Note that if $(\exists Q, P)[g = \hat{e}(P, Q)]$, and $\varphi(g^x) = xP$, then φ, which is supposed to be easy to compute, is precisely BPI (assuming Q is given as part of the definition of φ),which is believed to be hard. So either such a reduction does not exist, or it becomes meaningless (the former is more likely).

We now analyze more general cases of the function φ. Let $g \in G_2$ and $P \in G_1$ be generators of the corresponding groups, and let $\varphi(g^x) = f(x)P$, for some function f defined on $\mathbb{Z}_q$. Without loss of generality we assume that $g = \hat{e}(P, P)$ (there exists some $Q \in G_1$ s.t. $g = \hat{e}(Q, Q)$ and some $a \in \mathbb{Z}_q^*$ s.t. $aQ = P$, and the function $f()$ can take care of it).

Lemma 1. *If efficient φ: $G_2 \rightarrow G_1$ exists with $f(x) = x$, then φ could be used as an oracle to solve CDH_{G_2} in polynomial time.*

Proof. Given a tuple $< g, g^x, g^y >$ as input to CDH_{G_2} use oracle φ three times to find $< P, xP, yP >$, respectively. Then compute $\hat{e}(xP, yP) = \hat{e}(P, P)^{xy} = g^{xy}$. ∎

Lemma 2. *If efficient φ: $G_2 \rightarrow G_1$ exists with $f(x) = ax+b$, where $a \in \mathbb{Z}_q^*, b \in \mathbb{Z}_q$ then φ can be used as an oracle to solve DDH_{G_2} in deterministic polynomial time.*

Proof. Given a tuple $< g, g^x, g^y, g^z >$ as input to DDH_{G_2} use oracle φ to find $< P, f(x)P, f(y)P, f(z)P >$, respectively. The value bP can be found by invoking the oracle one more time with $x = 0$ (namely, $\varphi(1) = bP$). Thus we can compute axP, ayP, azP, and we can also compute aP (use $x = 1$). Since $\exists a^{-1} \bmod q$, $\hat{e}(axP, ayP) = \hat{e}(azP, aP)$ iff $\hat{e}(xP, yP) = \hat{e}(zP, P)$, namely, iff $g^{xy} = g^z$. ∎

In the next lemma we assume that x, y, z (in the definition of DDH_{G_2}) are uniformly distributed in $\mathbb{Z}_q$.

Lemma 3. *If efficient* φ*:* $G_2 \to G_1$ *exists with* $f(x) = ax^n + b$*, where* $a \in \mathbb{Z}_q^*, b \in \mathbb{Z}_q$ *and* $0 < n < q-1$*, then* φ *could be used as an oracle to solve* DDH_{G_2} *with error probability* $\gcd(n, q-1)^{-1}$.

Proof. Given a tuple $< g, g^x, g^y, g^z >$ as input to DDH_{G_2} use oracle φ to find $< P, f(x)P, f(y)P, f(z)P >$, respectively. The values aP, bP can be found as before. Thus we can compute ax^nP, ay^nP, az^nP. Since $\exists a^{-1} \bmod q$, $\hat{e}(ax^nP, ay^nP) = \hat{e}(az^nP, aP)$ iff $\hat{e}(x^nP, y^nP) = \hat{e}(z^nP, P)$, namely, iff $g^{(xy)^n} = g^{z^n}$. If $xy = z$ then $(xy)^n = z^n$, but the converse is true only with probability d^{-1}, where $d = \gcd(n, q-1)$ (since x, y, z are uniformly distributed in $\mathbb{Z}_q$, and there are d roots to $z^n - A = 0 \bmod q$, see e.g. [AG] Th. 7, pp. 22). ∎

This suggests a probabilistic algorithm for DDH_{G_2}, using φ as oracle. It is tempting to first try the following simple probabilistic reduction: pick just one random $r \in (0, q-1)$, Compute $< g, g^{xr}, g^{yr}, g^{zr^2} >$, Call oracle φ four times to compute $< aP, ax^{nr}P, ay^{nr}P, az^{nr^2}P >$, and then if $\hat{e}(ax^{nr}P, ay^{nr}P) \neq \hat{e}(az^{nr^2}P, aP)$, answer 'no,' and terminate, else, answer 'yes.' But this would not create an even distribution of instances, and we will not be able to precisely state the error probability. Hence we need a slightly more complex reduction that would create a uniform distribution as follow:

Algorithm 1.

Repeat k times:
    Pick uniformly at random $r_1, r_2, s_1, s_2 \in (0, q-1)$;
    Compute $< g, g^{xr_1+s_1}, g^{yr_2+s_2}, g^{zr_1r_2+xr_1s_2+yr_2s_1+s_1s_2} >$;
    Call oracle φ four times and compute $U_1 = aP, U_2 = a(xr_1 + s_1)^nP$,
    $U_3 = a(yr_2 + s_2)^nP, U_4 = a(zr_1r_2 + xr_1s_2 + yr_2s_1 + s_1s_2)^nP$;
    If $\hat{e}(U_2, U_3) \neq \hat{e}(U_1, U_4)$, answer 'no,' and terminate;
Else, answer 'yes.'

The 3-tuple $< U_2, U_3, U_4 >$ is uniformly distributed in a domain isomorphic to $\mathbb{Z}_q^3$. We conclude that if the answer is 'yes' then the error probability is $\varepsilon = (1 - d^{-1})^k$. A 'no' answer is always correct. We thus showed:

Theorem 1. *If efficient* φ*:* $G_2 \to G_1$ *exists with* $f(x) = ax^n + b$*, where* $a \in \mathbb{Z}_q^*, b \in \mathbb{Z}_q$ *and* $0 < n < q-1$*, then* φ *could be used as an oracle to solve* DDH_{G_2} *in probabilistic polynomial time.*

Open problem:

(1) Which functions $f()$ are potentially useful in a reduction to BDH?
(2) Do all the potentially useful functions $f()$ lead to contradictions similar to the above?
(3) Is it possible to strengthen the main claim from many-one reductions to polynomial Turing reductions?
(4) Is it possible to strengthen the main claim to PR reductions (see eg [JVL] pp. 246)?

Acknowledgement: I would like to thank Peter Montgomery for the improvement reflected in Algorithm-1, and for other very helpful comments on earlier manuscript. I also thank Kristin Lauter for helping me understand many issues concerning Weil pairing.

5. References

[AG]W.W. Adams and L.J. Goldstein: *Introduction to Number Theory,* Prentice-Hall, 1976.

[BF] D. Boneh and M. Franklin: *Identity-based encryption from the Weil-pairing,* in Proc. Crypto 2001.

[ERV] Eric R. Verheul: *Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystem,* in Advances in Cryptology, Brigit Pfitzmann (Ed.), LNCS 2045, pp. 195-210.

[HU] J.E. Hopcroft and J.D. Ullman: *Introduction to Automata Theory, Languages, and Computation,* Addison Wesley 1979.

[JN] A. Joux and K. Nguyen: *Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups,* in eprint.iacr.org.

[JVL] *Handbook of Theoretical Computer Science, Vol. A, Algorithms and Complexity*, J. van Leeuwen ed., MIT Press 1990.

[MOV] A. Menezes, T. Okamoto, and S. Vanstone: *Reducing elliptic curve logarithms to logarithms in a finite field,* IEEE Trans. IT, Vol. 39,pp. 1639-1646, 1993.

Microsoft Research, One Microsoft Way, Redmond, WA 98052
E-mail address: `yacov@microsoft.com`