AN OBSERVATION ABOUT VARIATIONS OF THE DIFFIE-HELLMAN ASSUMPTION

R. BHASKAR, K. CHANDRASEKARAN, S. LOKAM, P.L. MONTGOMERY, R. VENKATESAN, AND Y. YACOBI

Abstract. We generalize the Strong Boneh-Boyen (SBB) signature scheme to sign vectors (GSBB). We show that if a particular (but most natural) average case reduction from SBB to GSBB exists, then the Strong Diffie-Hellman (SDH) and the Computational Diffie-Hellman (CDH) have the same worst case complexity.

1. Introduction

Many researchers looked at the Boneh-Boyen signature scheme for Anonymous Credentials applications. In credential systems usually the credentials are represented as vectors. One can easily sign a vector using any ordinary digital signature scheme, by first hashing the vector into a relatively short message and then signing it. However, credential systems are very intricate, and have many additional requirements. The ability to sign vectors without destroying the algebraic structure may help accomplish some of those difficult requirements (but may also open the doors to new attacks, so one must be careful). Our complexity theoretic result may be of interest in such cases.

We generalize the Strong Boneh-Boyen (SBB) signature scheme to sign vectors (GSBB). There is a trivial worst-case polynomial time reduction from SBB to GSBB. We show that if an average case reduction exists *where the two problems are over the same elliptic curve group, and using the same generator*,

[[
Also, currently it works only for $n_{SBB} = n_{GSBB} = 0$.
]]

then the Strong Diffie-Hellman (SDH) and the Computational Diffie-Hellman (CDH) have the same worst case complexity.

2. The modified Weil pairings

Let E be Elliptic Curve group over field $\mathbb{F}$, both of order q for some large prime p. And let $\hat{e}$ be the modified Weil-pairing function $\hat{e} : E \times E \to \mathbb{F}$; $P \in E$ a generator of E. This map satisfies the following properties:

(1) $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in E$ and all $a, b \in \mathbb{F}$.
(2) If P is a generator of E then $g = \hat{e}(P, P)$ is a generator of $\mathbb{F}$.

In the (unmodified) Weil pairing the second requirement does not hold, since $e(P, P) = 1$ for every P. Throughout this paper we assume the *modified* Weil pairing. For more details on general bilinear groups see [1], sec. 2.2.

Date: November 26, 2008.

3. The Strong BB system

We start from a special case of [1], which is as strong as the general SBB, and then generalize it to sign vectors.

Global public parameters: EC group E over field $\mathbb{F}$,both of large prime order p, and a bilinear pairing function $\hat{e} : E \times E \to \mathbb{F}$; $P \in E$ a generator of E.

Secret key: x, y, random integers $\bmod p$.

Public key: $P \in E,\ U = xP,\ V = yP,\ z = \hat{e}(P, P) \in \mathbb{F},\ z \neq 1$.

Signing: To sign integer message $m \mod p$, pick a random $r \bmod p$. The signature is (σ, r), where

$$\sigma = \frac{1}{x + m + yr} P \in E$$

Verification:

$$\hat{e}(\sigma, U + mP + rV) = z$$

4. Generalization

A variation of the above signature system may be useful for signing vectors of credentials (a person has a vector of credentials; each entry in the vector is a credential). Suppose that the message is a vector $m = (m_1, m_2, ...m_t)$. Let the secret key be $\boldsymbol{x}, y$, where now $\boldsymbol{x} = (x_0, x_1, ...x_t)$, x_i, y are integers $\bmod p$ and the public key is $P \in E,\ U_i = x_i P,\ i = 0, 1, 2, ...t,\ V = yP,\ z = e(P, P) \in \mathbb{F},\ z \neq 1$.The signature is (σ, r), where

$$\sigma = \frac{1}{x_0 + \sum_{i=1}^{t} (x_i m_i) + yr} P \in E$$

Verification:

$$e(\sigma, U_0 + \sum_{i=1}^{t} m_i U_i + rV) = z$$

5. Reductions

To simplify the discussion we look at the case $t = 2$. The claims hold for $t > 2$ as well. α is the reduction from SBB (message=m) to GSBB, generalized system $(message = (1, m_1, m_2))$. Throughout this memo $r \in_R T$ means that element r is picked at random from set T with uniform distribution. Here is a concise description of the systems, that hints at a natural reduction:

	SBB	**GSBB**
Secret	$x, y \in_R \mathbb{Z}_p$	$x_0, x_1, x_2, y \in_R \mathbb{Z}_p$
Public	$P \in E,\ U = xP, V = yP,$ $z = e(P, P) \neq 1$	$P \in E,\ U_i = x_i P,\ i = 0, 1, 2$ $V = yP,\ z = e(P, P) \neq 1$
Sign	$r \in_R \mathbb{Z}_p,\ \sigma = \frac{1}{x+m+yr} P$	$r \in_R \mathbb{Z}_p,\ \sigma = \frac{1}{x_0+m_1x_1+m_2x_2+yr} P$
Verify	$e(\sigma, U + mP + rV) = z?$	$e(\sigma, U_0 + m_1U_1 + m_2U_2 + rV) = z?$

5.1. **The restricted reduction.** In the following assignments, variables of any SBB instance appear on the left and mapped ($\rightarrow$) onto variables of GSBB.

$P \rightarrow P$, $U \rightarrow U_0$, $V \rightarrow V$ (these assignments imply $z \rightarrow z$, $x \rightarrow x_0$, $y \rightarrow y$); Pick any x_1, x_2, m_1, m_2 subject to the constraint: $m = x_1 m_1 + x_2 m_2$. The signature returned by oracle GSBB is the signature needed in the SBB instance. This restricted reduction is a worst case reduction. It says nothing about average case complexity [6], [4], [2], [7]. While *validity* and *efficiency* hold, *domination* does not hold. We elaborate on the latter.

Let (R, μ_1) and (S, μ_2) denote the distributional decision problems corresponding to SBB and GSBB, respectively. Here μ_i is the distribution function (and μ_i' is the corresponding density function). Let M be a probabilistic oracle Turing Machine that randomly reduces (R, μ_1) to (S, μ_2), The probability $\Pr[v := M(u)]$ is taken over $M's$ internal coin flips (the corresponding notation in [2] is $Ask_M(u, v)$).

Definition 1. [2] *Domination holds if there exists a constant, $c > 0$, such that for every $v \in \{0,1\}^*$,*

$$\mu_2'(v) \geq \frac{1}{|v|^c} \cdot \sum_{u \in \{0,1\}^*} \Pr[v := M(u)] \cdot \mu_1'(u).$$

In our restricted reduction, the machine, M, maps short strings to longer strings, they are longer by roughly a factor $t \geq 2$. Let $r > 1$ be the exact factor of expansion. Let $|u| = n$. Then $M : \{0,1\}^n \rightarrow \{0,1\}^{rn}$.

Here is an example of a uniform distribution μ:

$$\forall x \in \{0,1\}^*, \quad \mu'(x) = |x|^{-2} 2^{-|x|}.$$

We show that (S, μ_2) does not dominate (R, μ_1) with respect to the restricted reduction (playing the role of M) if both μ_1 and μ_2 are uniform (as above).

In the restricted reduction, for every $v \in S$, there is exactly one $u \in R$, such that $\Pr[v := M(u)] = 1$, and for all other values of u, $\Pr[v := M(u)] = 0$. This simplifies the domination condition to $\mu_2'(v) \geq \frac{1}{|v|^c} \mu_1'(u)$, where u is the particular value for which $\Pr[v := M(u)] = 1$. For $r > 1$ it is impossible that

$$\lim_{|x| \rightarrow \infty} r^2 |x|^{-2} 2^{-r|x|} \geq \frac{1}{r^c |x|^c} |x|^{-2} 2^{-|x|},$$

since for large enough $|x|$, the exponentials are the dominant factors. We summarize these observations as follows:

Lemma 1. *Let (R, μ_1) and (S, μ_2), be distributional decision problems, and let $M : \{0,1\}^n \rightarrow \{0,1\}^{rn}$, be a reduction from (R, μ_1) to (S, μ_2) with $r > 1$, such that for every $v \in S$, there is exactly one $u \in R$, such that $\Pr[v := M(u)] = 1$, and for all other values of u, $\Pr[v := M(u)] = 0$. Then domination does not hold for this reduction when both μ_1 and μ_2 are uniform.*

We can hope for domination, and hence for average case reduction, if $\Pr[v := M(u)] = o(\exp(-r|u|))$. This happens if we can choose u and v independently. This calls for a non-restricted reduction.

5.2. **A hypothetical non restricted reduction.** Given arbitrary GSBB signature (i.e., without the previous restriction $m = \sum_{i=1}^{t} x_i m_i$) the Randomized Turing Machine, M, that reduces SBB to GSBB has to efficiently compute an SBB signature for a given message m. The machine M, is given m, it tosses its internal coins, and picks pairs (x_i, m_i), $i = 1, 2, ...t$, with uniform distribution. It then computes b, s.t. $m = b + \sum_{i=1}^{t} x_i m_i$. Given

$$\sigma_2 = \frac{1}{x_0 + \sum_{i=1}^{t} x_i m_i + yr} P$$

it computes

$$\sigma_1 = \frac{1}{x + m + yr} P.$$

The above reduction can be summarized as follows:
Given: $P, b, \frac{1}{z+b}P$,
Find: $\frac{1}{z}P$.
`To make it fully general (subject only to the restriction that the two problems are over the same elliptic curve group, and using the same generator) we allow multiple oracle calls, so input is` $P, b_i, \frac{1}{z+b_i}P$, $i = 1, 2, ...f()$ `where` $f()$ `is polynomial in the security parameter). Same output. In the reduction we invoke oracle` $\gamma = 1{-}SDH$ $f()$ `times.` γ `is probabilistic, so each time it produces a new` b_i. `[[Also, we need to throw in all those probabilities..]]`

5.3. **Proof of security.** To show that GSBB is secure we need to consider the setting of Strong Existential Un-forgeability ([1], sec. 2.1). It allows multiple challenges to the signer, after which an attacker computes a signature on a new message.

SBB is existentially unforgeable. A random Turing reduction from SBB to GSBB would prove the same for GSBB, where both problems allow multiple challenges. Let the number of challenges of SBB be n_{SBB}, and the number of challenges to GSBB (in a reduction) be $n_{GSBB} = f(n_{SBB})$, where $f()$ is any polynomial. If a reduction M exists for any n_{SBB}, then it exists for $n_{SBB} = 0$. We show that if a particular form of the latter exists then there is a reduction from Computational Diffie-Hellman (CDH) to Strong Diffie-Hellman (SDH) that *succeeds with the same probability.* We proceed to define that special reduction, for the identity polynomial $f()$. The case of a general polynomial $f()$ is only slightly more complex.

[[This is the most general reduction `subject to the restriction that the two problems are over the same elliptic curve group, and using the same generator.`
]]
Let $z = x + m + yr \mod p$, and define $b \in [0, p)$ such that $z + b = x_0 + \sum_{i=1}^{t} x_i m_i + yr \mod p$. The machine that performs the reduction has a subroutine M with the following basic input/output relations:
Given: $P, b, \frac{1}{z+b}P$,
Find: $\frac{1}{z}P$.
It is very unusual to use a reduction or a subroutine of a reduction (M) as a problem to which we reduce another problem. Usually we draw directed graphs

that describe reduction relations among problems, where the problems are nodes and the reductions are directed edges. But a reduction can also be defined as a problem, to and from which we make other reductions. It is also unusual to have the OR / AND of problems as nodes[1]. We are about to do all of the above.

5.4. **Strong Diffie-Hellman Vs. Computational Diffie-Hellman.** The q-Strong Diffie-Hellman (q-SDH) problem was defined in [1] with all the trappings of average case complexity, and SBB was proven there as hard to forge as q-SDH. Moreover, [1] proved a high lower bound on its complexity for a generic ("black-box") group. Still, it is not known if a SDH oracle can help solving the much more mature Computational Diffie-Hellman (CDH) problem, published in 1976 (a reduction the other way exists).

We show that existence of *average case* reduction from (R, μ_1) to (S, μ_2) implies the worst-case equivalence of CDH and SDH.

Let M be a non restricted reduction, as defined in the previous subsection. We use 1-SDH as a short-hand for 1-SDH with $\epsilon = 1$ (a special case of q-SDH, but sufficient for worst case reductions). $\boldsymbol{I}$ =inversion in the elliptic curve group. The I/O relations defining these problems are:

	Given	**Find**
$\boldsymbol{M}$	$P, b, \frac{1}{z+b}P$	$\frac{1}{z}P$
$\boldsymbol{CDH}$	R, uR, vR	uvR
$\boldsymbol{1-SDH}$ $with\ \ \varepsilon = 1$	P, zP	$(c, \frac{1}{z+c}P)$
$\boldsymbol{Inversion}$	P, zP	$\frac{1}{z}P$

```
[[
M  is α  in the slides.
]]
```

Problem M is an average case reduction. Yet we can view it as a function (a problem defined by input/output) and analyze worst case reductions to it and from it. In the next few bullet items, all the reductions are worst case reductions. We show that:

(1) Oracles M and $1-SDH$ together solve I (Lemma 2 below, see also [1], last paragraph of sec. 2.3),
(2) Problem $1-SDH$ is reducible to problem CDH (well known, see also Lemma 3 below),
(3) Problems I and CDH are reducible to each other (see Lemma 4, and [3]),
(4) We conclude from the above observations that if M is efficiently computable then problems CDH and $1-SDH$ are worst case reducible to each other in polynomial time. This problem has been open for a few years now.

Lemma 2. *Using oracles M and $1-SDH$ we can solve I.*

Proof. Given I's input call oracle $1-SDH$ to find $(c, \frac{1}{z+c}P)$. Then call oracle M to find $\frac{1}{z}P$. □

[1]Joux [5] has done OR before.

For the sake of self containment here is a reduction from $1 - SDH$ to CDH. Notation: Let $A = w_1 R, \quad B = w_2 R$. Define $\beta_R(A, B) = w_1 w_2 R$.

Lemma 3. *There exists a worst-case reduction from problem* $1 - SDH$ *to problem* CDH.

Proof. Given $1 - SDH's$ input, compute $R = P + zP = (1 + z)P$.
$\beta_R(P, P) = \beta_R(\frac{1}{1+z}R, \frac{1}{1+z}R) = \frac{1}{(1+z)^2}R = \frac{1}{1+z}P$.
This solves $1 - SDH$ for $c = 1$. □

Lemma 4. CDH *and* I *are polynomially reducible to each other.*

Proof. (a) CDH is polynomially reducible to I : Given oracle I, which constructs $(1/z)P$ from P and zP, first construct z^2P. We do it as follows: Compute $(1/z)P$, $(1 + z)P$, $(1/(1 + z))P$, $(1/z)P - (1/(1 + z))P = (1/(z + z^2))P$, $(z + z^2)P$, z^2P.

To get CDH (given R, uR, vR, find uvR) we use the above squarer oracle twice as follows: Compute $(u + v)R$, and $(u - v)R$. Call the squarer twice to compute $(u + v)^2R$, and $(u - v)^2R$. Their difference is $4uvR$. 4^{-1} exists modulo odd group order hence we can find uvR.

(b) I is polynomially reducible to CDH : To solve I, where we are given P and zP and want to find $(1/z)P$, let $R = zP$, and let $u = v = 1/z$. This is an unknown value, but we know $uR = vR = (1/z)zP = P$. Call oracle $\beta_R(uR, vR) = uvR = (1/z)(1/z)zP = (1/z)P$. □

6. Open problems

(1) Is there an average case reduction from SBB to GSBB?
(2) `[[Can there be an average case reduction from SBB to GSBB (subject only to the restriction that the two problems are over the same elliptic curve group, and using the same generator) that would not imply that problem` M `is easy?]]`

7. Acknowledgement

We would like to thank Adi Shamir for very helpful discussions.

References

[1] Dan Boneh and Xavier Boyen: Short Signatures Without Random Oracles; Proc. Eurocrypt'04, Springer Verlag LNCS, 2004.
[2] Shai Ben David, Benny Chor, Oded Goldreich, and Michael Luby: On the Theory of Average Case Complexity, Proceedings of the twenty-first annual ACM symposium on Theory of computing, pp. 204 - 216, 1989, ISBN:0-89791-307-8
[3] Feng Bao, Robert H. Deng, and Huafei Zhu: Variations of Diffie-Hellman Problem.
[4] Yuri Gurevich: Average Case Complexity; in Automata, Languages and Programming, LNCS, Springer, Vol. 510/1991, pp. 615-628.
[5] Antoine Joux: The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems. ANTS 2002: 20-32
[6] Leonid Levin: Average Case Complete Problems, SIAM J. Compute. 1986, pp. 285-286.
[7] Leonid Levin and Ramarathnam Venkatesan : Random instances of a graph coloring problem are hard, Annual ACM Symposium on Theory of Computing archive, Proceedings of the twentieth annual ACM symposium on Theory of computing, pp: 217 - 222 , 1988, ISBN:0-89791-264-0

Microsoft Research