Serdica J. Computing **3** (2009), 309–318

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

AN OBSERVATION ABOUT VARIATIONS OF THE DIFFIE-HELLMAN ASSUMPTION

Raghav Bhaskar, Karthekeyan Chandrasekaran, Satyanaryana V. Lokam, Peter L. Montgomery, Ramarathnam Venkatesan, Yacov Yacobi

Abstract. We generalize the Strong Boneh-Boyen (SBB) signature scheme to sign vectors; we call this scheme 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 have looked at the Boneh-Boyen signature scheme for Anonymous Credentials applications. In credential systems the credentials are usually 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

ACM Computing Classification System (1998): F.2, E.3.

Key words: Digital signatures, Boneh-Boyen signatures, Vector signatures, Strong Diffie-Hellman, Computational Diffie-Hellman, Average Case Complexity.

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.

In complexity theory, we argue about relations between the computational complexities of two problems using *reductions*. That problem A is *efficiently reducible* to problem B, means that we could use B as a subroutine (*oracle*) in an algorithm that solves A efficiently. An algorithm is *efficient* if it runs in a time polynomial in the length of its input. A reduction can be *worst-case* or *average-case*. The former implies only that the most difficult instances of A are at least as hard as the most difficult instances of B for a given probability distribution. The latter establishes that the average instances of A are at least as hard as the average instances of B. Such reductions are more interesting and more difficult to establish. The Computational Diffie-Hellman (CDH) assumption was introduced in 1976 [4] and is assumed to be hard. It is the basis for the classic Diffie-Hellman cryptosystems. The Strong Diffie-Hellman assumption (SDH) was introduced in [3], and is the foundation for modern pairing-based cryptosystems initially introduced in that paper and subsequently used in many others. It is known that SDH is reducible to CDH, but it is not known whether CDH is reducible to SDH even in the worst case.

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 a *particular* average case reduction exists then the Strong Diffie-Hellman (SDH) and the Computational Diffie-Hellman (CDH) have the same worst case complexity.

2. The modified Weil pairings. Let p be a prime and $\mathbb{F} = \mathbb{Z}_p$ the finite field of order p. Let E be an elliptic curve of order p over $\mathbb{F}$. Let $P \in E$ be a generator of E. The modified Weil-pairing function $\widehat{e} : E \times E \to \mathbb{F}^*$ 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 $\mathbb{F}$.

For more details on general bilinear groups see [3, Section 2.2].

3. The Strong BB system. We start from a special case of [3], which is as strong as the general SBB, and then generalize it to sign vectors.

Global public parameters: An elliptic curve group E over field $\mathbb{F}$, both of a 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 an integer message $m \bmod p$, 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 $m = (m_1, m_2, \ldots, m_t) \in \mathbb{Z}_p^t$. Let the secret key be $\boldsymbol{x}, y$, where now $\boldsymbol{x} = (x_0, x_1, \ldots x_t)$, x_i, y are integers $\bmod p$ and the public key is $P \in E$, $U_i = x_i P$, $i = 0, 1, 2, \ldots, t$, $V = yP$, $z = \hat{e}(P, P) \in \mathbb{F}$, $z \neq 1$. The signature is (σ, r), where

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

Verification:

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

5. Reductions. To simplify the discussion we look at the case $t = 2$. The claims hold for $t > 2$ as well. Let α be the reduction from the SBB system, with message=m, to the GSBB generalized system where the message is a vector $(1, m_1, m_2)$). Throughout this paper, $r \in_R T$ means that element r is picked at

random from finite set T with uniform distribution. Here is a concise description of the systems, which 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	$P \in E,\ U = xP, V = yP,$ $z = \hat{e}(P,P) \neq 1$	$P \in E,\ U_i = x_i P,\ i = 0,1,2$ $V = yP,\ z = \hat{e}(P,P) \neq 1$
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	$\hat{e}(\sigma, U + mP + rV) = z?$	$\hat{e}(\sigma, U_0 + m_1U_1 + m_2U_2 + rV) = z?$

5.1. The restricted reduction. In the following assignments, variables of any SBB instance appear on the left and are 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: $m = x_1m_1 + x_2m_2$. 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 [7], [5], [2], [8].

An average case reduction from problem A to problem B should be *valid, efficient*, and the *domination* property should hold. Roughly speaking, *Valid* means that the given reduction algorithm with oracle B usually solves A. *Efficient* means that on the average it runs in polynomial time in the input length. *Domination* assures that instances of A map "evenly" into the space of instances of B (i.e., that they do not all map into a small subset of all instances of B).

In the previous reduction, *validity* and *efficiency* hold, but *domination* does not hold. We elaborate on the latter. The following definition of Distributional Decision Problem is taken from [2] (where it is likewise defined for search problems).

Definition 1. *A Distributional Decision Problem is a pair (D,μ) where* $D : \{0,1\}^* \rightarrow \{0,1\}$ *and* $\mu : \{0,1\}^* \rightarrow [0,1]$ *is a probability distribution function.*

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 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 2. [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 $M : \{0,1\}^n \to \{0,1\}^{rn}$.

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

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

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 $M : \{0,1\}^n \to \{0,1\}^{rn}$ 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 an 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), $i = 1, 2, \ldots, t$, 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: $P, b, \dfrac{1}{z+b}P$,

Find: $\dfrac{1}{z}P$.

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 should allow multiple oracle calls, so the input becomes $P, b_i, \dfrac{1}{z+b_i}P$, $i = 1, 2, \ldots f()$, where $f()$ is polynomial in the security parameter. The output stays the same. Then in the reduction we invoke the oracle $\gamma :=$ 1-SDH $f()$ times. The oracle γ is probabilistic, so each time it produces a new b_i.

5.3. Proof of security. To show that GSBB is secure we need to consider the setting of Strong Existential Un-forgeability ([3, Section 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: $P, b, \dfrac{1}{z+b}P$,

Find: $\dfrac{1}{z}P$.

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 [3] with all the trappings of average case complexity, and SBB was proven there as hard to forge as q-SDH. Moreover, [3] proved a high lower bound on its complexity for a generic ("blackbox") group. Still, it is not known if an 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 the existence of *an 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,

[1] Joux [6] has done OR before.

but sufficient for worst case reductions). $\boldsymbol{I}$ = inversion in the elliptic curve group. In the following short problem description, the group orders are given, but we omit them from the descriptions. 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
1-SDH *with* $\varepsilon = 1$	P, zP	$\left(c, \frac{1}{z+c}P\right)$
$\boldsymbol{Inversion}$	P, zP	$\frac{1}{z}P$

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 list below, all the reductions are worst case reductions. We show that:

1. Oracles M and 1-SDH together solve I (Lemma 2 below, see also [3], 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 [1]),
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 $\left(c, \frac{1}{z+c}P\right)$. Then call oracle M to find $\frac{1}{z}P$. □

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

P r o o f. 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.*

P r o o f. (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 the known 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?

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

REFERENCES

[1] Bao F., R. H. Deng, H. Zhu. Variations of Diffie-Hellman Problem. In: LNCS, **2836**, Springer-Verlag, 2003, 301–312.

[2] David S. B., B. Chor, O. Goldreich, M. Luby. On the Theory of Average Case Complexity. In: Proceedings of the twenty-first annual ACM symposium on Theory of computing, 1989, 204–216.

[3] Boneh D., X. Boyen. Short Signatures Without Random Oracles. In: Proceedings of Eurocrypt'04, LNCS, **3027**, Springer-Verlag, 2004, 56–73.

[4] Diffie W., M. Hellman. New Directions in Cryptography, *IEEE Trans. on Inf. Th.*, **IT-22** (1976), 644–654.

[5] Gurevich Y. Average Case Complexity. In: 18th International Colloquium on Automata, Languages and Programming (ICALP91), LNCS, **510**. Springer-Verlag, 1991, 615–628.

[6] Joux A. The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems. ANTS 2002, 20–32.

[7] Levin L. Average Case Complete Problems, *SIAM J. Computing*, **15** (1986), Issue 1, 285–286.

[8] Levin L., R. Venkatesan. Random instances of a graph coloring problem are hard. In: Proceedings of the twentieth annual ACM symposium on Theory of computing, 1988, 217–222.

Raghav Bhaskar
Satyanarayana V. Lokam
Microsoft Research India
196/36 2nd Main Rd, Sadashivnagar
Bangalore – 560080, INDIA.
e-mail: {rbhaskar,satya}@microsoft.com

Karthekeyan Chandrasekaran
2116 KACB, 266 Ferst Drive
Georgia Tech, Atlanta
GA 30332, USA
e-mail: karthe@gatech.edu

Peter L. Montgomery
Ramarathnam Venkatesan
Yacov Yacobi
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
e-mail: {petmon,venkie,yacov}@microsoft.com

Received June 12, 2009
Final Accepted August 26, 2009