Hardness of Distinguishing the MSB or LSB of Secret Keys in Diffie-Hellman Schemes

Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Sébastien Zimmer

CNRS-École normale supérieure – Paris, France –
{Pierre-Alain.Fouque,David.Pointcheval,Jacques.Stern,Sebastien.Zimmer}@ens.fr

Abstract. In this paper we introduce very simple deterministic randomness extractors for Diffie-Hellman distributions. More specifically we show that the k most significant bits or the k least significant bits of a random element in a subgroup of $\mathbb{Z}_p^\star$ are indistinguishable from a random bit-string of the same length. This allows us to show that under the Decisional Diffie-Hellman assumption we can deterministically derive a uniformly random bit-string from a Diffie-Hellman exchange in the standard model. Then, we show that it can be used in key exchange or encryption scheme to avoid the leftover hash lemma and universal hash functions.

Keywords: Diffie-Hellman transform, randomness extraction, least significant bits, exponential sums.

1 Introduction

Motivation. The Diffie-Hellman key exchange [15] is a classical tool allowing two entities to agree on a common random element in a group G. It maps a pair of group elements (g^x, g^y) to g^{xy}. Since x and y are randomly chosen, the latter value is uniformly distributed in G. However it is not secret from an information theoretic point of view since x and y are uniquely determined modulo $|G|$ and so is g^{xy}. That is why an additional computational assumption is needed to guarantee that no computationally bounded attacker can find this element with a significant probability. The Computational Diffie-Hellman assumption (CDH) basically expresses this security notion. However, it does not rule out the ability to guess some bits of g^{xy}.

To obtain a cryptographic key from g^{xy} we need that no information leaks and further assumptions are required. Among those, the DDH is perhaps the most popular assumption and allows cryptographers to construct secure protocols [4]. It states the intractability of distinguishing DH-triples (g^x, g^y, g^{xy}) from random triples (g^x, g^y, g^z). Under the decisional Diffie-Hellman assumption (DDH) one can securely agree on a random and private element. However, a problem remains: this element is a random element in G but not a random bit-string as is generally required in further symmetric use. The common secret will indeed thereafter be used as a symmetric key to establish an authentic and private channel. Hence, one has to transform this random element into a random-looking bit-string, i.e. extract the *computational entropy* injected by the DDH assumption in the Diffie-Hellman element. To solve this problem, different methods have been proposed.

Thanks to the Leftover Hash Lemma [23, 25], one can extract entropy hidden within g^z by means of a family of universal hash functions. This solution has the advantage of being proven in the standard model and does not require any cryptographic assumption. One can indeed easily construct such families [10], and they are furthermore quite efficient to compute. However it requires extra randomness which needs to be of good quality (unbiased) and independent of the random secret g^z. Consequently, in a key exchange protocol, this extra randomness either needs to be authenticated or hard-coded in the protocol. This solution is mostly theoretical and is not widely used in standard protocols for the simple reason that families of universal hash functions are not present in cryptographic softwares, while they would be quite efficient [16, 33].

In practice, designers prefer to apply hash functions, such as MD5 or SHA-1, to the Diffie-Hellman element. This solution can be proven secure under the CDH assumption in the random oracle model [2], under the assumption that the compression function acts as a random oracle [13], but not in the standard model (unless one makes additional non-standard assumptions [1, 16, 18]).

In this paper, we analyze a quite simple and efficient randomness extractor for Diffie-Hellman distributions. The security relies on the DDH assumption in the *standard model*.

Related Works. To extract randomness from a Diffie-Hellman secret, one approach is to focus on the distribution induced by the DDH assumption. In [9], Canetti *et al.* show that given the k most significant bits of g^x and g^y, one cannot distinguish, in the statistical sense, the k most significant bits of g^{xy} from a random k bit-string. As Boneh observes [4], this is quite interesting but cannot be applied to practical protocols because an adversary always learns all of g^x and g^y. Chevassut *et al.* [11, 12] review a quite simple and optimal randomness extractor but which can be applied to $\mathbb{Z}_p^\star$, with a safe prime p only. This randomness extractor is very efficient but requires high computational effort to compute g^x, g^y and g^{xy} because of the requirement of a large group. They also presented a new technique (TAU [12]) but which applies to specific elliptic curves only. Independently, Gürel [22] proved that, under the DDH assumption over an elliptic curve, the most significant bits of the Diffie-Hellman transform are statistically close to a random bit-string, when the elliptic curve is defined over a quadratic extension of a finite field. However, $\mathbb{Z}_p^\star$ is one of the most interesting group and in order to speed up the Diffie-Hellman key-exchange, the computations must be performed in a small subgroup. To this end, Gennaro *et al.* [18] prove that a family of universal hash functions can be used even in non-DDH groups, provided that the group contains a large subgroup where the DDH assumption holds. However, this result still requires the use of a family of universal hash functions.

A second line of research is to study usual cryptographic primitives in protocols and prove that they are good randomness extractors. Dodis *et al.* [16]

therefore tried to analyze the security of IPsec. They showed that NMAC, the cascade construction and $\mathsf{CBC\text{-}MAC}$ are probabilistic randomness extractors. This is the first formal study of the randomness extraction phase of Diffie-Hellman standards in the standard model. These extractors can be applied with several distributions, not only the Diffie-Hellman distributions. However, these results require the assumption that the compression functions of the hash-based constructions under review (the hash functions MD5 or SHA-1) are a family of almost universal hash functions, which is not realistic.

In [5, 6], Boneh and Venkatesan show that the k most significant bits or least significant bits of g^{xy} are hard to compute. Namely, they prove that given an oracle which takes as input (g^x, g^y) and returns the k most significant bits of g^{xy}, one can construct an algorithm to compute g^{ab} given (g^a, g^b). They can take into account faulty oracle which can fail with probability at most $1/\log p$. In order to use these results to show that these bits are hardcore bits, the oracle must correctly answer with probability better than $1/2^k + \epsilon$. Indeed, in this case, the oracle finds the k bits more frequently than by guessing them. However, the techniques used cannot take into account such faulty oracles. Moreover their proof is known to contain a gap which was fixed by Gonzales-Vasco and Shparlinski in [21]. The result of [5, 6] is improved in [21, 20] and in [3]. In the latter, it is shown that under the DDH assumption the two most significant bits of the Diffie-Hellman result are hard to compute. Our main result here tells that under the DDH assumption, a good distinguisher for the two distributions (g^a, g^b, U_k) and $(g^a, g^b, \mathsf{lsb}_k(g^{ab}))$ cannot exist.

Our Result. In this paper, we use the exponential sum techniques to analyze cryptographic schemes. These techniques date back to the beginning of the last century, but we borrowed them from [9, 8] where they are used for cryptographic purposes. They allow us to study a very simple deterministic randomness extractor. Deterministic extractors have been recently introduced in complexity theory by Trevisan and Vadhan [28]. We describe here a deterministic randomness extractor which is provably secure in the standard model, under classical assumptions. We focus on the distribution induced by the DDH in a prime subgroup G of $\mathbb{Z}_p^\star$, where p is prime and $|G| \gg \sqrt{p}$. We prove that the k least significant bits of a random element of G are statistically close to a perfectly random bit-string. In other words, we have a very simple *deterministic* randomness extractor which consists in keeping the k least significant bits of the random element and discarding the others. This extractor can be applied to Diffie-Hellman Key Exchange and El Gamal-based encryption schemes, under the DDH assumption. It does not need any family of universal hash functions neither any extra randomness. We also show that if p is sufficiently close below of a power of 2 by a small enough amount, the k most significant bits are also uniformly distributed.

Organization. In section 2, we present some definitions and results about entropy and randomness extraction. In section 3, we present and analyze our new randomness extractor. In section 4, we compare our extractor with other randomness extractors. In section 5, we present some natural and immediate applications of our extractor. In section 6, we relax the DDH assumption into the weaker CDH assumption and analyze the bit-string we can generate in that case.

2 Entropy and Randomness Extractors

First of all we introduce the notions used in randomness extraction. In the following, a randomness source is viewed as a probability distribution.

2.1 Measures of Randomness

Definition 1 (Min Entropy). *Let X be a random variable with values in a set $\mathcal{X}$ of size N. The* guessing probability *of X, denoted by $\gamma(X)$, is the probability* $\max_{x\in\mathcal{X}}(\Pr[X=x])$. *The min entropy of X is: $H_\infty(X) = -\log_2(\gamma(X))$.*

For example, when X is drawn from the uniform distribution on a set of size N, the min-entropy is $\log_2(N)$. To compare two random variables we use the classical statistical distance:

Definition 2 (Statistical Distance). *Let X and Y be two random variables with values in a set $\mathcal{X}$ of size N. The* statistical distance *between X and Y is the value of the following expression:*

$$\mathbf{SD}(X,Y) = \frac{1}{2}\sum_{x\in\mathcal{X}} |\Pr[X=x] - \Pr[Y=x]|\,.$$

We denote by U_k a random variable uniformly distributed over $\{0,1\}^k$. We say that a random variable X with values in $\{0,1\}^k$ is *δ-uniform* if the statistical distance between X and U_k is upper bounded by δ.

2.2 From Min Entropy to δ-Uniformity

The most common method to obtain a δ-uniform source is to extract randomness from high-entropy bit-string sources. Presumably, the most famous randomness extractor is provided by the Leftover Hash Lemma [23, 25], which requires to introduce the notion of universal hash function families.

Definition 3 (Universal Hash Function Families). *Let $\mathcal{H} = \{h_i\}_i$ be a family of efficiently computable hash functions $h_i : \{0,1\}^n \to \{0,1\}^k$, for $i \in \{0,1\}^d$. We say that $\mathcal{H}$ is a universal hash function family if for every $x \neq y$ in $\{0,1\}^n$,*

$$\Pr_{i\in\{0,1\}^d}[h_i(x) = h_i(y)] \leq 1/2^k.$$

Theorem 4 (Leftover Hash Lemma). *Let $\mathcal{H}$ be a universal hash function family from $\{0,1\}^n$ into $\{0,1\}^k$, keyed by $i \in \{0,1\}^d$. Let i denote a random variable with uniform distribution over $\{0,1\}^d$, let U_k denote a random variable* *uniformly distributed in* $\{0,1\}^k$*, and let* A *denote a random variable taking values in* $\{0,1\}^n$*, with* i *and* A *mutually independent. Let* $\gamma = \gamma(A)$*, then:*

$$\mathbf{SD}(\langle i, h_i(A)\rangle, \langle i, U_k\rangle) \leq \frac{\sqrt{2^k \gamma}}{2}.$$

Proof. See [32].

The Leftover Hash Lemma extracts nearly all of the entropy available whatever the randomness sources are, but it needs to invest few additional truly random bits. To overcome this problem, it was proposed to use deterministic functions. They do not need extra random bits, but only exist for some specific randomness sources.

Definition 5 (Deterministic Extractor). *Let* f *be a function from* $\{0,1\}^n$ *into* $\{0,1\}^k$*. Let* $\mathcal{X}$ *be a set of random variables of min entropy* m *taking values in* $\{0,1\}^n$ *and let* U_k *denote a random variable uniformly distributed in* $\{0,1\}^k$*, where* U_k *and* X *are independent for all* $X \in \mathcal{X}$*. We say that* f *is an* (m,ε)*-deterministic extractor for* $\mathcal{X}$ *if for all* $X \in \mathcal{X}$*:*

$$\mathbf{SD}\left(f(X), U_k\right) < \varepsilon.$$

3 Randomness Extractor in a Subgroup of $\mathbb{Z}_p^\star$

In this section, we propose and prove the security of a simple randomness extractor for the Diffie-Hellman exchange in sufficiently large subgroups of $\mathbb{Z}_p^\star$. The main result of this section is theorem 7 which shows that least significant bits of a random element in G are statistically close to truly random bits. To prove this result, we apply the exponential sum techniques in order to find an upper bound on the statistical distance. It is very similar to the results of [27] who studies the distribution of fractional parts of ag^x/p in given intervals of $[0,1]$.

Our result does not require the DDH assumption. However, as it is precised in section 5, to apply it in a cryptographic protocol, the DDH assumption is needed to obtain a random element in the subgroup of $\mathbb{Z}_p^\star$.

3.1 Description of the Deterministic Extractor

Let p be an n-bit prime, that is $2^{n-1} < p < 2^n$, G a subgroup of $\mathbb{Z}_p^\star$ of order q with $q \gg \sqrt{p}$, ℓ the integer such that $2^{\ell-1} \leq q < 2^\ell$ and X a random variable uniformly distributed in G. In the following, we denote by k an integer, by s a k-long bit-string and the associated integer in $[\![0,\, 2^k - 1]\!]$, and by U_k a random variable uniformly distributed in $\{0,1\}^k$. If x is an integer, we denote by $\mathsf{lsb}_k(x)$ the k least significant bits of x and by $\mathsf{msb}_k(x)$ the k most significant bits of x.

In this section we show that the k least significant bits of a random element g of G are statistically close to a truly random k-long bit-string provided that G is large enough. A direct consequence of this result is that the function from

$\mathbb{Z}_p^\star$ to $\{0,1\}^k$ which keeps only the k least significant bits of its input is a good deterministic extractor for a G-group source (that is for variables uniformly distributed in the group $G \subset \mathbb{Z}_p^\star$).

Definition 6. *The function* $\mathsf{Ext}_k : \{0,1\}^n \to \{0,1\}^k : c \mapsto \mathsf{lsb}_k(c)$ *is called an* (n,p,q,k)*-**extractor** for a G-group source.*

Theorem 7. *With the above notations of an* (n,p,q,k)*-extractor for a group source, we have:*

$$\mathbf{SD}(\mathsf{lsb}_k(X), U_k) < \frac{2^k}{p} + \frac{2^k\sqrt{p}\log_2(p)}{q} < 2^{k+n/2+\log_2(n)+1-\ell}.$$

This inequality is non trivial only if $k < \ell - n/2 - \log_2(n) - 1$.

Proof. Let us define $K = 2^k$, $H_s = \lfloor \frac{p-1-s}{K} \rfloor$ for $s \in [\![0,\ K-1]\!]$. Let denote by e_p the following character of $\mathbb{Z}_p$: for all $y \in \mathbb{Z}_p$, $e_p(y) = e^{\frac{2i\pi y}{p}} \in \mathbb{C}^*$. The character e_p is an homomorphism from $(\mathbb{Z}_p, +)$ in $(\mathbb{C}^*, \cdot)$. Since

$$\frac{1}{p} \times \sum_{a=0}^{p-1} e_p(a(g^x - s - Ku)) = \mathbb{1}(x,s,u),$$

where $\mathbb{1}(x,s,u)$ is the characteristic function which is equal to 1 if $g^x = s + Ku \bmod p$ and 0 otherwise, we have:

$$\begin{aligned}\Pr_{X\in G}[\mathsf{lsb}_k(X) = s] &= \frac{1}{q} \times \left|\{(x,u) \in [\![0,\ q-1]\!] \times [\![0,\ H_s]\!] \mid g^x = s + Ku \bmod p\}\right| \\ &= \frac{1}{qp} \times \sum_{x=0}^{q-1}\sum_{u=0}^{H_s}\sum_{a=0}^{p-1} e_p(a(g^x - s - Ku)).\end{aligned}$$

Let us change the order of the sums, and split sum on the a's in two terms:

1. the first one comes from the case $a = 0$, and is equal to $(H_s+1)/p$, that is approximately $1/2^k$,
2. the second one comes from the rest, and will be the principal term in the statistical distance in which we can separate sums over x and u.

Twice the statistical distance, that is 2Δ, is equal to:

$$\begin{aligned}&\sum_{s\in\{0,1\}^k} \left|\Pr_{X\in G}[\mathsf{lsb}_k(X) = s] - 1/2^k\right| \\ &\le \sum_{s\in\{0,1\}^k} \left|\frac{H_s+1}{p} - \frac{1}{2^k}\right| + \sum_{s\in\{0,1\}^k} \frac{1}{qp} \sum_{a=1}^{p-1} \left|\left(\sum_{x=0}^{q-1} e_p(ag^x)\right)\left(\sum_{u=0}^{H_s} e_p(-aKu)\right)\right|.\end{aligned}$$

For the first term, we notice that $\left|(H_s+1)/p - 1/2^k\right| \le 1/p$, since $K = 2^k$, $H_s = \lfloor \frac{p-1-s}{K} \rfloor$ and:

$$-\frac{1}{p} \le -\frac{1+s}{Kp} \le \left(1 + \left\lfloor \frac{p-1-s}{K} \right\rfloor\right)\frac{1}{p} - \frac{1}{K} \le \frac{K-(1+s)}{Kp} \le \frac{1}{p}.$$

For the second term, we introduce $M = \max_a \left(\left| \sum_{x=0}^{q-1} e_p(ag^x) \right| \right)$, and show that:

$$\begin{aligned}
\sum_{a=1}^{p-1} \left| \sum_{u=0}^{H_s} e_p(-aKu) \right| &= \sum_{a=1}^{p-1} \left| \sum_{u=0}^{H_s} e_p(-au) \right| = \sum_{a=1}^{p-1} \left| \frac{1 - e_p(-a(H_s+1))}{1 - e_p(-a)} \right| \\
&= \sum_{a=1}^{p-1} \left| \frac{\sin(\frac{\pi a(H_s+1)}{p})}{\sin(\frac{\pi a}{p})} \right| = 2 \sum_{a=1}^{\frac{p-1}{2}} \left| \frac{\sin(\frac{\pi a(H_s+1)}{p})}{\sin(\frac{\pi a}{p})} \right| \\
&\leq 2 \sum_{a=1}^{\frac{p-1}{2}} \left| \frac{1}{\sin(\frac{\pi a}{p})} \right| \leq \sum_{a=1}^{\frac{p-1}{2}} \left| \frac{p}{a} \right| \leq p \log_2(p).
\end{aligned}$$

The first equality results from a change of variables. The second equality comes from the fact that $[\![0,\, H_s]\!]$ is an interval, therefore the sum is a geometric sum. We use the inequality $\sin(y) \geq 2y/\pi$ if $0 \leq y \leq \pi/2$ for the second inequality. In summary we have:

$$2\Delta \leq \frac{2^k}{p} + \frac{2^k M \log_2(p)}{q}. \tag{1}$$

Using the bound $M \leq \sqrt{p}$ that can be found in [26], $2^{n-1} < p < 2^n$ and $2^{\ell-1} \leq q < 2^\ell$, we obtain the expected result.

Consequently, since the min entropy of X, as an element of $\mathbb{Z}_p^\star$ but randomly distributed in G, equals $\log_2(|G|) = \log_2(q)$, the previous proposition leads to:

Corollary 8. *Let e be a positive integer and let suppose that we have $\log_2(q) > m = n/2 + k + e + \log_2(n) + 1$. Then the application Ext_k is an $(m, 2^{-e})$-deterministic extractor for the G-group distribution.*

3.2 Improvements

One drawback of the previous result is that we need a subgroup of order at least $\sqrt{p}$. In order to have more efficient Diffie-Hellman key exchange, one prefers to use smaller subgroups. Therefore to improve the results obtained on this random extractor, one idea would be to find a better bound than $\sqrt{p}$ on $M = \max_a \left(\left| \sum_{x=0}^{q-1} e_p(ag^x) \right| \right)$. There are several results which decrease this bound, as these from [7, 24]. Many of them are asymptotic, and do not explicit the constants involved. However, by looking carefully at the proof in [24] or [26] we can find them:

Theorem 9 ([26]). *With the notations of the previous subsection, if $q \geq 256$ then, for all $x \in \mathbb{Z}_p^\star$, we have:*

$$M \leq \begin{cases} p^{1/2} & \big(\textit{interesting if } p^{2/3} \leq q\big) \\ 4p^{1/4}q^{3/8} & \big(\textit{interesting if } p^{1/2} \leq q \leq p^{2/3}\big) \\ 4p^{1/8}q^{5/8} & \big(\textit{interesting if } 256 \leq q \leq p^{1/2}\big) \end{cases}$$

The bound $\sqrt{p}$ is always valid whatever p and q are. Yet, if $\sqrt{p} < q < p^{2/3}$, the second bound is better and similarly to the third bound. For example, with $n = 2048$, $\ell = 1176$ and $e = 80$, theorem 7 says that we can extract 60 bits. Using the second bound given in the theorem above with the equation 1 we obtain that $k \leq 5\ell/8 - (e + n/4 + \log_2(n) + 3)$. It means that we can actually extract 129 bits and obtain a bit-string of reasonable size. However, in most practical cases, the classical bound $\sqrt{p}$ is the most appropriate.

Moreover when G is the group of quadratic residues, Gauss has proven that $\left|\sum_{x=0}^{p-1} e_p(ax)\right| = \sqrt{p}$, for all $a \in \mathbb{Z}_p^\star$. Therefore, $\left|\sum_{x=0}^{q-1} e_p(ag^x)\right| \geq (\sqrt{p} - 1)/2$. This means that in the case of safe primes and with this proof technique, our result is nearly optimal.

3.3 Other Result

The theorem presented in the previous section considers least significant bits. A similar result for most significant bits can be proved with the same techniques. We have the following theorem, whose proof is omitted by lack of space:

Theorem 10. *Let δ be $(2^n - p)/2^n$. If p, m, k and e are integers such that $3\delta < 2^{-e-1}$ and $\log_2(|G|) > m = n/2 + k + e + \log_2(n) + 1$, then the function $\mathsf{msb}_k(\cdot)$ is a $(m, 2^{-e})$-deterministic extractor for the G-group distribution.*

The first assumption on p to be close by below to a power of 2 is easily justified by the fact that the most significant bit is highly biased whenever p is just above a power of 2. Indeed in this case, with high probability, the most significant bit is equal to 0.

4 Comparisons

In the literature other randomness extractors proven secure in the standard model are also available.

4.1 The Leftover Hash Lemma

A famous one is the leftover hash lemma which is presented in subsection 2.2. If one uses a universal hash function family, we can extract up to $\log_2(|G|) - 2e + 2$ bits from a random element in G. With our extractor, the number of random bits extracted is approximately $\log_2(|G|) - (n/2 + \log_2(n) - e + 1)$. However, the leftover hash lemma needs the use of a universal hash function family and extra truly random bits.

In practice we can derandomize it by fixing the key of the hash function. Shoup [32] proved that in this case, there is a linear loss of security in the number of calls of the hash function.

4.2 An Optimal Randomness Extractor for Safe Prime Groups

To extract randomness from a random element of a subgroup of $\mathbb{Z}_p^\star$, where $p = 2q+1$ is a safe prime (q is also a prime), there is another deterministic extractor reviewed in [11, 12]. Let $G = \langle g \rangle$ denote the subgroup of quadratic residues of $\mathbb{Z}_p^\star$, and let g^x be a random element in G. To extract the randomness of g^x, the extractor needs this function f:

$$f(g^x) = \begin{cases} g^x & \text{if } g^x \leq (p-1)/2 \\ p - 1 - g^x & \text{otherwise} \end{cases}$$

This function is a bijection from G to $\mathbb{Z}_q$. To obtain a random bit-string, one has to truncate the result of f. The composition of f and the truncation is a good deterministic extractor. As f is a bijection, in some sense it is optimal : all the randomness is extracted. However this simple extractor is very restrictive because it can be applied only with a safe prime when our extractor can be used with a significantly larger set of primes. Moreover our extractor is more efficient than this simple one.

5 Applications

The DDH assumption allows to find to our extractor some natural applications in cryptographic protocols. It can indeed be applied in every protocol which generates a random element in a subgroup of $\mathbb{Z}_p^\star$ and where a randomness extractor is needed.

5.1 Key Exchange Protocol

Our extractor is designed to extract entropy from a random element in a group G. It is exactly what is obtained after a Diffie-Hellman key exchange performed in a DDH group G, where G is a subgroup of $\mathbb{Z}_p^\star$.

This means that we have an efficient solution to the problem of agreeing on a random bit-string which is based on the following simple scheme, provably secure in the standard model under the DDH assumption: Alice sends g^x, Bob sends g^y and they compute $\mathsf{lsb}_k(g^{xy})$.

The multiplicative group $\mathbb{Z}_p^\star$ is not a DDH group but if $p = \alpha q + 1$ with q a large prime and α small then the subgroup of $\mathbb{Z}_p^\star$ with q elements may be assumed a DDH group (in such a group, the DDH assumption is reasonable.) Therefore in this case we can extract up to $k = n/2 - (e + \log_2(n) + 2 + \log_2(\alpha))$ bits from an $n - \log_2(\alpha)$ min entropy source.

In practice, the security parameters are often $n = 1024$, $e = 80$. Hence we can extract approximately $420 - \log_2(\alpha)$ bits at the cost of two exponentiations modulo an integer of 1024 bits. It means that if we need a 128-long bit-string, the subgroup should have approximately 2^{731} elements.

5.2 Encryption Schemes

El Gamal Encryption Scheme [17]. In the El Gamal encryption scheme, the message must be an element of a cyclic group G of order q. Alice generates a random element x in $\mathbb{Z}_q$ and publishes $y = g^x$ where g is a generator of G. To encrypt the message m, she generates a random element r of $\mathbb{Z}_q$ and computes (g^r, my^r). This scheme is proven IND-CPA secure if $m \in G$. However in practice messages are often bit-strings and not elements from G. One solution to avoid this problem is to extract the randomness from y^r and xor the generated bit-string with the message. This way, the encryption scheme is still IND-CPA secure. Our extractor can be used in this context to extract randomness.

Cramer-Shoup Encryption Scheme [14, 31]. The Cramer-Shoup encryption scheme is an improvement of the El Gamal encryption scheme which is IND-CCA secure. The principle is the same as in El Gamal, it hides m multiplying it with a random element h^r of G. The security proof requires that m is in G. In order to use bit-string messages, we can use the same solution: extract randomness from h^r with our extractor and xor the result with m.

6 Other Assumptions

In this section, we apply our result under various assumptions, related to the DDH one. First, we make a stronger assumption, the so-called Short Exponent Discrete Logarithm, which allows quite efficient DH-like protocols. Then, we relax the DDH assumption to the CDH one.

6.1 The s-DLSE Assumption

To speed up our randomness extractor, we can use a group in which the additional Short Exponent Discrete Logarithm (DLSE) assumption holds. First introduced in [34], it is formalized in [29] and [18] as follows:

Assumption 1 (s-DLSE [29]) *Let s be an integer, $\mathcal{G} = \{G_n\}_n$ be a family of cyclic groups where each G_n has a generator g_n and $ord(G_n) = q_n > 2^n$. We say that the s-DLSE Assumption holds in $\mathcal{G}$ if for every probabilistic polynomial time Turing machine I, for every polynomial $P(\cdot)$ and for all sufficiently large n we have that* $\Pr_{x \in_R [\![1,\, 2^s]\!]} [I(g_n, q_n, s, g_n^x) = x] \leq 1/P(n)$.

As explained in [18], current knowledge tends to admit that in a group of prime order, for a 2^{-e} security level, we can choose $s \geq 2e$. The usual security parameter of $e = 80$ leads to $s \geq 160$, which is quite reasonable, from a computational cost.

Gennaro *et al.* prove in [18] that under the s-DLSE and the DDH assumption, the two following distributions are computationally indistinguishable:

$$\left\{(g^x, g^y, Z) \middle|\, x, y \in_R [\![1,\, 2^s]\!],\ Z \in_R G\right\} \text{ and } \left\{(g^x, g^y, g^{xy}) \middle|\, x, y \in_R [\![1,\, 2^s]\!]\right\}.$$

This result allows us to use our extractor with the latter distribution and in that way be computationally more efficient.

6.2 The CDH Assumption

In practice, to apply our extractor, we need to work in a group where the DDH assumption is true. It is more difficult to extract entropy in a group where only the CDH assumption is supposed to hold. As precised in the introduction, in the random oracle model, it is possible to extract entropy using hash functions such as MD5 or SHA-1. Yet, in the standard model under the CDH assumption, we currently know how to extract only $O(\log \log p)$ bits and not a fixed fraction of $\log_2(p)$ as we prove in this paper under the DDH assumption. This bound of $O(\log \log p)$ bits is an indirect application of the Goldreich-Levin hard-core predicate [19], using the Shoup's trick [30].

Acknowledgement. The authors would like to thank I. Shparlinski for his helpful remarks and the anonymous reviewers for their comments.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *CT – RSA '01*, LNCS 2020, pages 143–158. Springer-Verlag, Berlin, 2001.
2. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, 1993.
3. I. F. Blake, T. Garefalakis, and I. E. Shparlinski. On the bit security of the Diffie-Hellman key. In *Appl. Algebra in Engin., Commun. and Computing*, volume 16, pages 397–404, 2006.
4. D. Boneh. The Decision Diffie-Hellman Problem. In J. P. Buhler, editor, *Algorithmic Number Theory Symposium (ANTS III)*, LNCS 1423, pages 48–63. Springer-Verlag, Berlin, 1998.
5. D. Boneh and R. Venkatesan. Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Schemes. In *Crypto '96*, LNCS 1109, pages 129–142. Springer-Verlag, Berlin, 1996.
6. D. Boneh and R. Venkatesan. Rounding in Lattices and its Cryptographic applications. In *Proc. of ACM-SIAM SODA'97*, pages 675–681, 1997.
7. J. Bourgain and S. V. Konyagin. Estimates for the Number of Sums and Products and for Exponential Sums Over Subgroups in Fields of Prime Order. *Comptes Rendus Mathmatiques*, 337:75–80, 2003.
8. R. Canetti, J. Friedlander, S. Konyagin, M. Larsen, D. Lieman, and I. Shparlinski. On the Statistical Properties of Diffie-Hellman Distributions. *Israel Journal of Mathematics*, 120:23–46, 2000.
9. R. Canetti, J. Friedlander, and I. Shparlinski. On Certain Exponential Sums and the Distribution of Diffie-Hellman Triples. *Journal of the London Mathematical Society*, 59(2):799–812, 1999.
10. L. Carter and M. Wegman. Universal Hash Functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.
11. O. Chevassut, P. A. Fouque, P. Gaudry, and D. Pointcheval. Key derivation and randomness extraction. Cryptology ePrint Archive, Report 2005/061, 2005. `http://eprint.iacr.org/`.
12. O. Chevassut, P. A. Fouque, P. Gaudry, and D. Pointcheval. The twist-augmented technique for key exchange. In *PKC '06*, LNCS 3958, pages 410–426. Springer-Verlag, Berlin, 2006.

13. J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgard Revisited : How to Construct a Hash Function. In *Crypto '05*, LNCS 3621, pages 430–448. Springer-Verlag, Berlin, 2005.
14. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.
15. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.
16. Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin. Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes. In *Crypto '04*, LNCS, pages 494–510. Springer-Verlag, Berlin, 2004.
17. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
18. R. Gennaro, H. Krawczyk, and T. Rabin. Secure Hashed Diffie-Hellman over Non-DDH Groups. In *Eurocrypt '04*, LNCS 3027, pages 361–381. Springer-Verlag, Berlin, 2004.
19. O. Goldreich and L.A. Levin. A Hard-Core Predicate for all One-Way Functions. In *Proc. of the 21st STOC*, pages 25–32. ACM Press, New York, 1989.
20. M. I. Gonzalez Vasco, M. Näslund, and I. E. Shparlinski. New results on the hardness of Diffie-Hellman bits. In *PKC '04*, LNCS 2947, pages 159–172, 2004.
21. M. I. Gonzalez Vasco and I. E. Shparlinski. On the security of Diffie-Hellman bits. In *Proc. Workshop on Cryptography and Computational Number Theory Singapore, 1999*, pages 331–342. Birkhauser, 2001.
22. N. Gürel. Extracting bits from coordinates of a point of an elliptic curve. Cryptology ePrint Archive, Report 2005/324, 2005. `http://eprint.iacr.org/`.
23. J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom Generator from any One-Way Function. *SIAM Journal of Computing*, 28(4):1364–1396, 1999.
24. D. R. Heath-Brown and S. Konyagin. New bounds for Gauss sums derived from k^{th} powers, and for Heilbronn's exponential sum. *Q. J. Math.*, 51(2):221–235, 2000.
25. R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proc. of the 30th FOCS*, pages 248–253. IEEE, New York, 1989.
26. S. V. Konyagin and I. Shparlinski. *Character Sums With Exponential Functions and Their Applications*. Cambridge University Press, Cambridge, 1999.
27. N. M. Korobov. The distribution of digits in periodic fractions. *Mat. Sb. (N.S.)*, 89(131):654–670, 672, 1972.
28. L. Trevisan and S. Vadhan. Extracting Randomness from Samplable Distributions. In *Proc. of the 41st FOCS*, pages 32–42. IEEE, New York, 2000.
29. S. Patel and G. Sundaram. An Efficient Discrete Log Pseudo Random Generator. In *Crypto '98*, LNCS 1462. Springer-Verlag, Berlin, 1998.
30. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt '97*, LNCS 1233, pages 256–266. Springer-Verlag, Berlin, 1997.
31. V. Shoup. Using Hash Functions as a Hedge against Chosen Ciphertext Attack. In *Eurocrypt '00*, LNCS 1807, pages 275–288. Springer-Verlag, Berlin, 2000.
32. V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, Cambridge, 2005.
33. V. Shoup and T. Schweinberger. ACE: The Advanced Cryptographic Engine. Manuscript, March 2000. Revised, August 14, 2000.
34. P. C. van Oorschot and M. J. Wiener. On Diffie-Hellman Key Agreement with Short Exponents. In *Eurocrypt '96*, LNCS 1070, pages 332–343. Springer-Verlag, Berlin, 1996.