Optimal Randomness Extraction from a Diffie-Hellman Element

Céline Chevalier, Pierre-Alain Fouque, David Pointcheval, and Sébastien Zimmer

École Normale Supérieure, CNRS-INRIA, Paris, France
{Celine.Chevalier,Pierre-Alain.Fouque,David.Pointcheval,Sebastien.Zimmer}@ens.fr

Abstract. In this paper, we study a quite simple deterministic randomness extractor from random Diffie-Hellman elements defined over a prime order multiplicative subgroup G of a finite field $\mathbb{Z}_p$ (the truncation), and over a group of points of an elliptic curve (the truncation of the abscissa). Informally speaking, we show that the least significant bits of a random element in $G \subset \mathbb{Z}_p^*$ or of the abscissa of a random point in $\mathcal{E}(\mathbb{F}_p)$ are indistinguishable from a uniform bit-string. Such an operation is quite efficient, and is a good randomness extractor, since we show that it can extract nearly the same number of bits as the Leftover Hash Lemma can do for most Elliptic Curve parameters and for large subgroups of finite fields. To this aim, we develop a new technique to bound exponential sums that allows us to double the number of extracted bits compared with previous known results proposed at ICALP'06 by Fouque *et al.* It can also be used to improve previous bounds proposed by Canetti *et al.* One of the main application of this extractor is to mathematically prove an assumption proposed at Crypto '07 and used in the security proof of the Elliptic Curve Pseudo Random Generator proposed by the NIST. The second most obvious application is to perform efficient key derivation given Diffie-Hellman elements.

1 Introduction

Since Diffie and Hellman's seminal paper [10], many cryptographic schemes are based on the Diffie-Hellman technique: key exchange protocols [10] of course, but also encryption schemes, such as ElGamal [12] and Cramer-Shoup [9] ones, or pseudo-random generators, as the Naor-Reingold PRNG [23]. More precisely, the security of these schemes relies on the Decisional Diffie-Hellman assumption (DDH) [4], which means that there is no efficient algorithm that can distinguish the two distributions in G^3, (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), where a, b and c are chosen at random in $[\![1, q]\!]$, and $G = \langle g \rangle$ is a cyclic group, generated by g of prime order q. For many of the schemes whose security is based on the DDH assumption, the DH element is used as a shared random element of G. However, a perfectly random element of G is not a perfectly random bit string and sometimes, as in key derivation for example, it can be useful to derive a uniform bit string which could be used as a symmetric key. Therefore, from this random element of G, one has to find a way to generate a random bit string.

1.1 Related Work

One classical solution to derive a random-looking bit-string from the DH element is to use a hash function. One indeed gets a uniform bit-string, but in the random oracle model [2].

Another solution, secure in the standard model, is to use a randomness extractor, such as the one which has been proposed by Gennaro *et al.* in [15]. But one first needs to have some entropy in the DH element, whereas g^{ab} is totally determined by g^a and g^b. This entropy is computationally injected using a computational assumption, as the CDH and DDH assumptions.

The CDH assumption, which states that g^{ab} is difficult to compute from g^a and g^b, implies that several bits of g^{ab} are not known from the adversary. Therefore, from the adversary point of view, there is some randomness in it. So one solution is to prove the hardness of predicting the least significant bits of a DH element. This comes from the hardcore bit theory, where one tries to provide a reduction between an algorithm that predicts the least significant bits of the DH element to the recovery of the whole DH element: predicting these bits is thus as hard as solving the CDH problem. However, usually, only a small number of bits can be proved to be random-looking, given g^a and g^b [6, 5, 20].

This entropy can also be computationally created using the DDH assumption, which says that we have $\log_2(q)$ bits of entropy in the DH element, but one does not know where exactly: one cannot extract them directly out of the representation of the element in G. This is the goal of a randomness extractor. The *Leftover Hash Lemma* [17, 15] is the most famous randomness extractor. It is a *probabilistic* randomness extractor that can extract entropy for *any* random source which has sufficient min-entropy. The main drawback with the Leftover Hash Lemma is that it requires the use of pairwise independent hash functions, which are not used in practice, and extra perfect randomness. A computational version of this Leftover Hash Lemma, has also been proposed and analysed in [14], version which has the advantage of using pseudorandom functions for randomness extraction and not pairwise independent hash functions. However, it still requires the use of extra perfect randomness. The two previous solutions are generic: it could be interesting to find a *deterministic* solution dedicated to the randomness extraction from a random element in G, since it would prevent the use of extra randomness.

Definitely, the most interesting solution in this vein is to keep the least significant bits of the DH element and hope that the resulting bit-string is uniform, as it is proposed in many papers [6, 5, 20]. Truncation, as studied above and in this paper, is quite simple and *deterministic*, which is of high interest from a practical point of view, even if it is specific to DH distributions.

A first step in this direction was the analysis of Canetti *et al.* [8] which basically shows that the concatenation of the least significant bits of g^a, g^b and g^{ab} is close to the uniform distribution. This result was achieved using exponential sums techniques. However, Boneh [4] noted: "*This is quite interesting although it does not seem to apply to the security analysis of existing protocols. In most protocols, the adversary learns all of* g^a *and* g^b*.*" This result is statistical

and no cryptographic assumption is required, since some bits of a and b are free, when the view of the adversary is limited to some part of g^a and g^b. There is no chance to extend this result to our problem, since, as already noted, given the entire representation of g^a and g^b, there is no randomness at all in g^{ab}. However, under the DDH assumption, some entropy appears in the DH element, and so, one can expect to extract it into a bit-string that will be close to the uniform distribution, in a statistical sense.

At ICALP'06, Fouque *et al.* [13] use this idea and show that under the DDH assumption, the least significant bits of g^{ab} are nearly uniformly distributed, given g^a and g^b, if the group G is a large enough multiplicative subgroup (of prime order q) of a finite field (let say $\mathbb{Z}_p$), that is, q is not too small compared to p. The large q is the main drawback since q needs to be at least half the size of p, which makes the cryptographic protocol quite inefficient. To prove this result, the authors upper bound the statistical distance, evaluating directly the L_1 norm, using exponential sums.

Since elliptic curves cryptography uses large subgroup in practice, the same result for elliptic curve could be of practical interest. Gürel [16] studied the case of elliptic curves over quadratic extensions of a finite field, with a large fraction of bits, and over a prime finite field, but with similar limitations as above in the number of extracted bits. He also upper bounds directly the statistical distance by evaluating the L_1 norm, but using a sum of Legendre characters. His technique only uses the Legendre character, which is not enough in the case of $\mathbb{Z}_p$. Consequently, the technique of the authors of [13] needed to sum on all characters.

1.2 Our Results

In this paper, we show that the following distributions are computationally indistinguishable

$$(aP, bP, U_k) \approx_C (aP, bP, \mathtt{lsb}_k(x(abP))),$$

where U_k is the uniform distribution on k-bit strings, $\mathtt{lsb_k}()$ is the function which truncates the k least significant bits of a bit-string and $x()$ is the abscissa function of points on an elliptic curve.

Under the DDH assumption, we know that $(aP, bP, abP) \approx_C (aP, bP, cP)$ for random scalars $a, b, c \in [\![1, q]\!]$, in the group G, generated by P of prime order q. Then, we prove, without any cryptographic or mathematical assumption, that

$$(aP, bP, U_k) \approx_S (aP, bP, \mathtt{lsb}_k(x(cP)))$$

in a statistical sense.

Actually, we first show this result for prime order multiplicative subgroups of finite fields. This result extends those of Canetti *et al.* and of Fouque *et al.* since we are able to extract twice the number of bits as before. This new result is achieved by introducing a new technique to bound the statistical distance. Whereas previous techniques directly tried to bound the L_1 norm, while it is hard to cope with the absolute value, we upper-bound the Euclidean L_2 norm,

which is much easier since only squares are involved. Finally, we are also able, in some cases, to improve our result using classical techniques on exponential sums. Then, the number of extracted bits can be made quite close to the number that the Leftover hash lemma can extract.

However, since the result still applies to large subgroups only, we extend it to Elliptic Curve groups. In general, the co-factor of EC groups is small: less than 8, and even equal to one for the NIST curves, over prime fields. We thus achieve our above-mentioned result using more involved techniques on exponential sums over functions defined on the points of the elliptic curve. More precisely, we can show that the 82 (resp. 214 and 346) least significant bits of the abscissa of a DH element of the NIST curves over prime fields of 256 (resp. 384 and 521) bits are indistinguishable from a random bit-string. They can thus be directly used as a symmetric key. To compare with Gürel's result in [16], for an elliptic curve defined over a prime field of 200 bits, Gürel extracts 50 bits with a statistical distance of 2^{-42}, while with the same distance, we can extract 102 bits. Note that Gürel's proof was easier to understand, but we did not manage to evaluate the L_2 norm of Legendre character sums and generalize his proof.

One main practical consequence of the result for elliptic curve is that, we can avoid the Truncated Point Problem (TPP) assumption used in the security proof of the NIST Elliptic Curve Dual Random Bit Generator (DRBG) [7, 24].

1.3 Organization of the paper

In Section 2, we review some notations and the definition of a deterministic randomness extractor as well as some results on the Leftover Hash Lemma. Then, in Section 3, we improve the results of Canetti *et al.* and of Fouque *et al.* using a new technique to bound exponential sums, using the Euclidean norm. In this section, we also improve the bound in some cases. Next, in Section 4, we prove the same kind of result for the group of points of an elliptic curve. Finally, in Section 5, we show some applications of our proofs to the security of the NIST EC DRBG [7, 24] and the key derivation from a DH element.

2 Notations

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

2.1 Measures of Randomness

To measure the randomness existing in a random variable, we use two different measures: the *min entropy* and the *collision entropy*. The min entropy measures the difficulty that an adversary has to guess the value of the random variable, whereas the collision entropy measures the probability for two elements drawn according this distrubtion to collide. In this paper, the collision entropy is used as an intermediate tool to establish results, which are then reformulated using min entropy.

Definition 1 (Min Entropy). *Let X be a random variable with values in a finite set $\mathcal{X}$. 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 $\log_2(N)$.

Definition 2 (Collision Entropy). *Let X and X' be two random independent and identically distributed variables with values in a finite set $\mathcal{X}$. The* collision probability *of X, denoted by* $\mathrm{Col}(X)$ *is the probability* $\Pr[X = X'] = \sum_{x\in\mathcal{X}}\Pr[X=x]^2$. *The* collision entropy *of X is:* $\mathbf{H}_2(X) = -\log_2(\mathrm{Col}(X))$.

The collision entropy is also called the Renyi entropy. There exists an easy relation between collision and min entropies: $\mathbf{H}_\infty(X) \leq \mathbf{H}_2(X) \leq 2\cdot\mathbf{H}_\infty(X)$. To compare two random variables we use the classical statistical distance:

Definition 3 (Statistical Distance). *Let X and Y be two random variables with values in a finite set $\mathcal{X}$. The* statistical distance *between X and Y is the value of the following expression:*

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

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.

Lemma 4. *Let X be a random variable with values in a set $\mathcal{X}$ of size $|\mathcal{X}|$ and $\varepsilon = \mathbf{SD}(X, U_\mathcal{X})$ the statistical distance between X and $U_\mathcal{X}$, the uniformly distributed variable over $\mathcal{X}$. We have:*

$$\mathrm{Col}(X) \geq \frac{1+4\varepsilon^2}{|\mathcal{X}|}. \tag{1}$$

Proof. This lemma, whose result is very useful in this work, is proved in Appendix A.

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, using a so-called *randomness extractor*. Presumably, the most famous randomness extractor is provided by the *Leftover Hash Lemma* [17, 19], which requires the use of *universal hash function families*.

Definition 5 (Universal Hash Function Family). *A universal hash function family $(h_i)_{i\in\{0,1\}^d}$ with $h_i : \{0,1\}^n \to \{0,1\}^k$, for $i \in \{0,1\}^d$, is a family of functions such that, for every $x \neq y$ in $\{0,1\}^n$, $\Pr_{i\in\{0,1\}^d}[h_i(x) = h_i(y)] \leq 1/2^k$.*

Let $(h_i)_{i\in\{0,1\}^d}$ be a universal hash function family, 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 X denote a random variable taking values in $\{0,1\}^n$, with i and X mutually independent and with X min entropy greater than m, that is $\mathbf{H}_\infty(X) \geq m$. The Leftover Hash Lemma (which proof can be found in [25]) states that $\mathbf{SD}(\langle i, h_i(X)\rangle, \langle i, U_k\rangle) \leq 2^{(k-m)/2-1}$.

In other words, if one wants to extract entropy from the random variable X, one generates a *uniformly distributed* random variable i and computes $h_i(X)$. The Leftover Hash Lemma guarantees a 2^{-e} security, if one imposes that

$$k \leq m - 2e + 2. \tag{2}$$

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 6 (Deterministic Extractor). *Let f be a function from $\{0,1\}^n$ into $\{0,1\}^k$. Let X be a random variable 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. We say that f is an (X,ε)-deterministic extractor if:*

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

2.3 Characters on Abelian Groups

We recall a standard lemma for character groups of Abelian groups.

Lemma 7. *Let H be an Abelian group and $\hat{H} = \mathrm{Hom}(H, \mathbb{C}^*)$ its dual group. Then, for any element χ of $\hat{H}$, the following holds, where χ_0 is the trivial character:*

$$\frac{1}{|H|}\sum_{h\in H}\chi(h) = \begin{cases} 1 & \text{if } \chi = \chi_0 \\ 0 & \text{if } \chi \neq \chi_0 \end{cases}$$

In the following, we denote by e_p the character such that for all $x \in \mathbb{F}_p$, $e_p(x) = e^{\frac{2i\pi x}{p}} \in \mathbb{C}^*$.

2.4 Elliptic Curves

Let p be a prime and $\mathcal{E}$ be an elliptic curve over $\mathbb{F}_p$ given by the Weierstrass equation

$$y^2 + (a_1x + a_3)\cdot y = x^3 + a_2x^2 + a_4x + a_6.$$

We denote by $\mathcal{E}(\mathbb{F}_p)$ the group of elements of $\mathcal{E}$ over $\mathbb{F}_p$ and by $\mathbb{F}_p(\mathcal{E})$ the function field of the curve $\mathcal{E}$, defined as the field of fractions over the points

of $\mathcal{E}$: $\mathbb{F}_p(\mathcal{E}) = \mathbb{F}_p[X,Y]/\mathcal{E}(\mathbb{F}_p)$. It is generated by the functions x and y, satisfying the Weierstrass equation of $\mathcal{E}$, and such that $P = (x(P), y(P))$ for each $P \in \mathcal{E}(\mathbb{F}_p) \setminus \{O\}$. Let $f_a \in \mathbb{F}_p(\mathcal{E})$ be the application $f_a = a \cdot x$ where $a \in \mathbb{Z}_p^*$. If $f \in \mathbb{F}_p(\mathcal{E})$, we denote by $\deg(f)$ its degree, that is $\sum_{i=1}^{t} n_i \deg(P_i)$ if $\sum_{i=1}^{t} n_i P_i$ is the divisor of poles of f. Finally, we denote by $\Omega = \mathrm{Hom}(\mathcal{E}(\mathbb{F}_p), \mathbb{C}^*)$ the group of characters on $\mathcal{E}(\mathbb{F}_p)$, and by ω_0 the trivial character (such that $\omega_0(P) = 1$ for each P).

3 Randomness Extraction in Finite Fields

In this section, we first extends results from Fouque *et al.* [13], in order to extract bits from random elements in a multiplicative subgroup of a finite field. Then, we use the same techniques to improve the result of Canetti *et al.* [8].

3.1 Randomness Extraction

We study now the randomness extractor which consists in keeping the least significant bits of a random element from a subgroup G of $\mathbb{Z}_p^*$. The proof technique presented here allows us to extract twice the number of bits extracted by Fouque *et al.*. In the particular case when $q \geq p^{3/4}$, where q is the cardinal of G, we prove an even better result: one can extract as many bits as with the Leftover Hash Lemma. This means that, in the case when $q \geq p^{3/4}$, our extractor is as good as the Leftover Hash Lemma, but computationally more efficient and easiest to use in protocols, since it does not require extra perfect public randomness.

In the original paper, Fouque *et al.* upper bound directly the statistical distance between the extracted bits and the uniform distribution, using exponential sums. We still use them, but propose to apply exponential sum technique to upper bound the collision probability of the extracted bits. The Cauchy-Schwartz inequality allows to relate statistical distance and collision probability and to conclude. Since the distribution of extracted bits is very close to the uniform distribution, the Cauchy-Schwartz inequality is very tight. That is the reason why we do not lose much with our roundabout way. On the contrary, we are able to find a good upper bound of collision resistance, and thus the global upper bound is improved.

The result in the case when $q \geq p^{3/4}$ is elaborated on the same basic idea but requires more elaborated techniques on exponential sums to be established.

Theorem 8. *Let p be a n-bit prime, G a subgroup of $\mathbb{Z}_p^*$ of cardinal q (we denote $\ell = \log_2(q) \in \mathbb{R}$), U_G a random variable uniformly distributed in G and k a positive integer. We have:*

$$\mathbf{SD}(\mathsf{lsb}_k(U_G), U_k) \leq \begin{cases} 2^{3n/4-\ell-1} + 2^{(k-\ell)/2} & \left(\text{if } p^{3/4} \leq q\right) \\ 2^{(k+n+\log_2 n)/2-\ell} & \left(\text{if } (2^{-8}p)^{2/3} \leq q \leq p^{3/4}\right) \\ 2^{(k+n/2+\log_2 n+4)/2-5\ell/8} & \left(\text{if } p^{1/2} \leq q \leq (2^{-8}p)^{2/3}\right) \\ 2^{(k+n/4+\log_2 n+4)/2-3\ell/8} & \left(\text{if } (2^{16}p)^{1/3} \leq q \leq p^{1/2}\right). \end{cases}$$

We remind that these inequalities are non trivial only if they are smaller than 1.

Proof. We give here a sketch of proof of the theorem, the complete proofs are in the full version.

Let us define $K = 2^k$, $u_0 = \mathsf{msb}_{n-k}(p-1)$. 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 $(\mathbb{C}^*, \cdot)$. For all $a \in \mathbb{Z}_p^*$, let also introduce the following notation:

$$S(a, G) = \sum_{x \in G} e_p(ax).$$

The two main interests of exponential sums is that they allow to construct some characteristic functions and that in some cases we know good upper bounds for them. Thanks to these caracteristic functions one can evaluate the size of certain sets and, manipulating sums, one can upper bound the size of these sets.

In our case, we construct $\mathbb{1}(x, y, u) = \frac{1}{p} \times \sum_{a=0}^{p-1} e_p(a(g^x - g^y - Ku))$, where $\mathbb{1}(x, y, u)$ is the characteristic function which is equal to 1 if $g^x - g^y = Ku \bmod p$ and 0 otherwise. Therefore, we can evaluate $\mathrm{Col}(\mathsf{lsb}_k(U_G))$ where U_G is uniformly distributed in G:

$$\begin{aligned}\mathrm{Col}(\mathsf{lsb}_k(U_G)) &= \frac{1}{q^2} \times \left| \{(x, y) \in [\![0, q-1]\!]^2 \mid \exists u \leq u_0,\ g^x - g^y = Ku \bmod p\} \right| \\ &= \frac{1}{q^2 p} \times \sum_{x=0}^{q-1} \sum_{y=0}^{q-1} \sum_{u=0}^{u_0} \sum_{a=0}^{p-1} e_p(a(g^x - g^y - Ku)).\end{aligned}$$

Then we manipulate the sums, separate some terms ($a = 0$) and obtain:

$$\mathrm{Col}(\mathsf{lsb}_k(U_G)) = \frac{u_0 + 1}{p} + \frac{1}{q^2 p} \sum_{a=1}^{p-1} |S(a, G)|^2 \left(\sum_{u=0}^{u_0} e_p(-aKu) \right). \tag{3}$$

The last three bounds. From this point, the proof of the last three inequations is different from the proof of the first inequation. First, we give here the sketch of proof of the last three inequations of the theorem (we remind that the complete proof is given in the full version).

In Equation (3) we inject the absolute value, introduce $M = \max_a(|S(a, G)|)$, make classical manipulations and obtain:

$$\mathrm{Col}(\mathsf{lsb}_k(U_G)) \leq \frac{u_0 + 1}{p} + \frac{M^2 \log_2(p)}{q^2}.$$

We now use the Lemma 4 which gives a relation between the statistical distance ε of $\mathsf{lsb}_k(X)$ with the uniform distribution and the collision probability: $\mathrm{Col}(\mathsf{lsb}_k(U_G)) \geq \frac{1+4\varepsilon^2}{2^k}$. The previous upper bound, combined with some manipulations, gives:

$$2\varepsilon \leq \sqrt{2^k \cdot \mathrm{Col}(\mathsf{lsb}_k(U_G)) - 1} \leq \sqrt{\frac{2^k}{p}} + \frac{2^{k/2} M \log_2^{1/2}(p)}{q}. \tag{4}$$

We conclude the theorem using the following upper bounds for M:

$$M \leq \begin{cases} p^{1/2} & \left(\text{interesting if } p^{2/3} \leq q\right) \\ 4p^{1/4}q^{3/8} & \left(\text{interesting if } p^{1/2} \leq q \leq p^{2/3}\right) \\ 4p^{1/8}q^{5/8} & \left(\text{interesting if } 2^{16/3}p^{1/3} \leq q \leq p^{1/2}\right). \end{cases}$$

The first bound above is the famous Polya-Vinogradov bound that we recall in Theorem 9 (its proof is reminded in the full version). The other bounds are from [22, 18]. The last third bounds of the theorem can be easily deduced.

Theorem 9 (Polya-Vinogradov inequality). *Let p be a prime number, G a subgroup of $(\mathbb{Z}_p^*, \cdot)$. For all $a \in \mathbb{Z}_p^*$, we have:*

$$\left|\sum_{x \in G} e_p(ax)\right| \leq \sqrt{p}.$$

The first bound. We give now a sketch of proof of the first inequality, a precise proof is given in the full version. For that, we use a bit more elaborated results than previously: for all coset $\omega \in \mathbb{Z}_p^*/G$, and for all two representatives a and a' of the coset ω, we have $S(a, G) = S(a', G)$. Therefore we can naturally define $S(\omega, G)$.

To establish the first inequality, we use Equation (3) and manipulating sums we establish that:

$$\mathrm{Col}(\mathsf{lsb}_k\left(U_G\right)) = \frac{u_0+1}{p} + \frac{1}{q^2 p} \sum_{\omega \in \mathbb{Z}_p^*/G} |S(\omega, G)|^2 \sum_{u=0}^{u_0} S(-\omega K u, G).$$

Then we use the Polya-Vinogradov inequality combined with the inequality $\sum_{\omega \in \mathbb{Z}_p^*/G} |S(\omega, G)|^2 \leq p$ (the proof of this result is reminded in the full version) and show that:

$$\mathrm{Col}(\mathsf{lsb}_k\left(U_G\right)) \leq \frac{u_0+1}{p} + \frac{u_0\sqrt{p}+q}{q^2}.$$

Finally, we finish, as for previous inequalities, using that $\mathrm{Col}(\mathsf{lsb}_k\left(U_G\right)) \geq \frac{1+4\varepsilon^2}{2^k}$, and obtain:

$$2\varepsilon \leq 2^{(k-n+1)/2} + 2^{3n/4-\ell} + 2^{(k-\ell)/2}.$$

Since $\ell \leq n-1$, this gives the expected bound. □

Since the min entropy of U_G, as an element of $\mathbb{Z}_p^*$ but uniformly distributed in G, equals $\ell = \log_2(|G|) = \log_2(q)$, the previous proposition leads to:

Corollary 10. *Let e be a positive integer and let us suppose that one of these inequations is true:*

$$k \leq \begin{cases} \ell - (2e+2) & \text{and } 2^e \cdot p^{3/4} \leq q \\ 2\ell - (n + 2e + \log_2(n)) & \text{and } (2^{-8} \cdot p)^{2/3} \leq q \leq 2^e \cdot p^{3/4} \\ 5\ell/4 - (n/2 + 2e + \log_2(n) + 4) & \text{and } p^{1/2} \leq q \leq (2^{-8} \cdot p)^{2/3} \\ 3\ell/4 - (n/4 + 2e + \log_2(n) + 4) & \text{and } (2^{16} \cdot p)^{1/3} \leq q \leq p^{1/2}. \end{cases}$$

In this case, the application Ext_k is an $(U_G, 2^{-e})$-deterministic extractor.

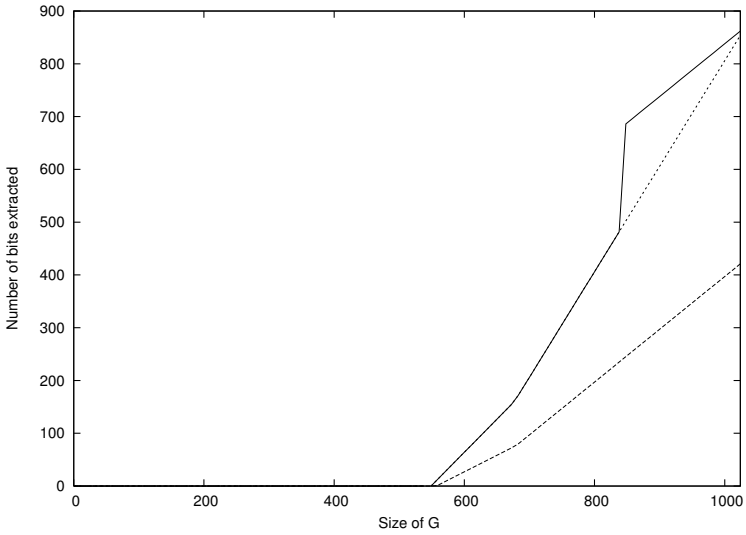

Fig. 1. This is the number of bits extracted according to the group size, for $n = 1024$ and $e = 80$. The long dash line represents Fouque *et al.* [13] result, the plain line is our results. Note the jump for $q = p^{3/4}$. The short dash line represents our result without particular improvement in the case $q \geq p^{3/4}$.

This means that if one wants a 2^{-e} security, and if $(2^{-8} \cdot p)^{2/3} \leq q \leq 2^e \cdot p^{3/4}$, one can extract k bits with $k \leq 2(\ell - (n/2 + e + \log_2 n/2))$.

In most practical cases, the second bound is the most appropriate. However, sometimes it is one of the others. For example, with $n = 1024$, $\ell = 600$ and $e = 80$, the second bound says that we can extract 6 bits. Using the third bound given in the theorem above we can actually extract 64 bits.

If one wants to extract a 256-bit string, for the same values of n and e, one needs a group of size greater than 2^{756}. The figure 1 presents our upper bounds and also the original upper bounds of Fouque *et al.* [13], in the case when $n = 1024$ and $e = 80$.

3.2 Truncated Inputs

Our above result proves that given g^a and g^b, the least significant bits of g^{ab} are globally indistinguishable from a random bit-string, under the Decisional Diffie-Hellman problem.

But our technique can be applied to other results which upper-bound statistical distances using character sums. One of them is the result of Canetti *et al.* [8], which studies some statistical properties of Diffie-Hellman distribution. They show that if one takes a proportion of the least significant bits of g^x, g^y, g^{xy}, then one obtains a distribution whose statistical distance from uniform is exponentially small. Basically, it shows that given the least significant bits of g^a

and g^b, the least significant bits of g^{ab} are globally indistinguishable from a random bit-string, without any computational assumption.

More precisely, if k_1, k_2, k_3 are three integers and U_1, U_2, U_3 three independent random variables uniformly distributed in respectively $\{0,1\}^{k_1}$, $\{0,1\}^{k_2}$, $\{0,1\}^{k_3}$, then, using the notations as in previous subsection, their Theorem 9 inequality, can be restated as follows:

$$\mathbf{SD}\left(\left(\mathsf{lsb}_{k_1}\left(g^x\right), \mathsf{lsb}_{k_2}\left(g^y\right), \mathsf{lsb}_{k_3}\left(g^{xy}\right)\right), (U_1, U_2, U_3)\right) \ll \frac{2^{k_1+k_2+k_3} p^{1/4} \log_2^{1/3}(p)}{q^{1/3}}.$$

Using our techniques, we can prove a better upper-bound:

Theorem 11. *Let p be a prime, G a subgroup of $\mathbb{Z}_p^*$ of cardinal q and X, Y two independent random variables uniformly distributed in $\{1, \ldots, q\}$. If k_1, k_2, k_3 are three integers and U_1, U_2, U_3 three independent random variables uniformly distributed in respectively $\{0,1\}^{k_1}$, $\{0,1\}^{k_2}$, $\{0,1\}^{k_3}$, then we have:*

$$\mathbf{SD}\left(\left(\mathsf{lsb}_{k_1}\left(g^x\right), \mathsf{lsb}_{k_2}\left(g^y\right), \mathsf{lsb}_{k_3}\left(g^{xy}\right)\right), (U_1, U_2, U_3)\right) \ll \frac{2^{\frac{k_1+k_2+k_3}{2}} p^{1/4} \log_2^{1/3}(p)}{q^{1/3}}.$$

Proof (Sketch of the proof.). First, find an upper bound for the collision entropy $\mathrm{Col}\left(\mathsf{lsb}_{k_1}\left(g^x\right), \mathsf{lsb}_{k_2}\left(g^y\right), \mathsf{lsb}_{k_3}\left(g^{xy}\right)\right)$ using exponential sum techniques and the inequality of Theorem 8 of [8]. Conclude with Lemma 4. □

4 Randomness Extraction in Elliptic Curves

We now show how the randomness extractor studied in the previous section, which consisted in keeping the least significant bits of a random element from a subgroup G of $\mathbb{Z}_p^*$, can be extended to another structure, that is the group of elements of an elliptic curve. The main idea is to evaluate the element "M" of the proof of Theorem 8, which is the upper bound of $S(a, G)$ as defined in the previous section.

4.1 A bound for $S(a, G)$

If G is a subgroup of $\mathcal{E}(\mathbb{F}_p)$, $f \in \mathbb{F}_p(\mathcal{E})$ and $\omega \in \Omega$, we define

$$S(\omega, f, G) = \sum_{P \in G} \omega(P) e_p(f(P)).$$

In particular, since $e_p \in \Omega$,

$$S(a, G) = S(\omega_0, f_a, G) = \sum_{P \in G} e_p(f_a(P)).$$

The objective of this section is to show the following result:

Theorem 12. *Let $\mathcal{E}$ be an elliptic curve over $\mathbb{F}_p$ and $f \in \mathbb{F}_p(\mathcal{E})$. Then,*

$$S(\omega, f, \mathcal{E}(\mathbb{F}_p)) \leq 2 \deg(f) \sqrt{p}.$$

As a consequence, if $a \in \mathbb{Z}^*$, $S(a, \mathcal{E}(\mathbb{F}_p)) \leq 4\sqrt{p}$.

More specifically, in the case where the co-factor is not equal to 1, we are interested in its corollary. Note that in the case of $\mathbb{F}_p$, all the curves recommended by the NIST have co-factor equal to 1. The proof of this corollary can be found in the full version.

Corollary 13. *Let $\mathcal{E}$ be an elliptic curve over $\mathbb{F}_p$, $a \in \mathbb{Z}^*$ and G a subgroup of $\mathcal{E}(\mathbb{F}_p)$. Then,*

$$S(\omega, f, G) \leq 2\deg(f)\sqrt{p} \qquad \text{and} \qquad S(a, G) \leq 4\sqrt{p}.$$

Proof (of Theorem 12). For sake of simplicity, we only show the case where $\omega = \omega_0$ in order to use easier notations. We follow the proof of Bombieri in [3] and Kohel and Shparlinski in [21], by first considering $S_m(f, \mathcal{E}(\mathbb{F}_p)) = S(\sigma \circ f, \mathcal{E}(\mathbb{F}_{p^m}))$ where σ is the trace from $\mathbb{F}_{p^m}$ to $\mathbb{F}_p$. Note that for our needs, the interesting sum corresponds to $m = 1$.

This sum comes from the character $e_p \circ f$, which defines an Artin-Schreier extension (informally, an extension of degree p) of the function field $\mathbb{F}_p(\mathcal{E})$, and then an Artin-Schreier covering of $\mathcal{E}(\mathbb{F}_p)$. An easy way to evaluate this sum is to consider the L-function related to this Artin-Schreier covering. L-functions are a standard means to assemble several elements in a unique object (a series), in the same manner as a generating power series, see for example [26, chap. 14]. Bombieri shows that this L-function is defined as follows, for $t \in \mathbb{C}$ such that $|t| < q^{-1}$:

$$L(t, f, \mathcal{E}(\mathbb{F}_p)) = \exp\left(\sum_{m=1}^{+\infty} S(f, \mathcal{E}(\mathbb{F}_p))t^m/m\right).$$

By the Artin conjecture, which proof was given by Weil in [27] (see the full version), this function is a polynomial of degree $D = \deg(f)$. Denote its D complex roots (not necessarily distincts) by $\theta_i = {\omega_i}^{-1}$. Then, we have the two following equations:

$$L(t, f, \mathcal{E}(\mathbb{F}_p)) = \sum_{i=0}^{+\infty} \frac{1}{i!}\left(\sum_{m=1}^{+\infty} S_m(f, \mathcal{E}(\mathbb{F}_p))t^m/m\right)^i$$

$$L(t, f, \mathcal{E}(\mathbb{F}_p)) = \prod_{i=1}^{D}(1 - \omega_i t).$$

The first equation can be rewritten the following way:

$$1 + \sum_{m=1}^{+\infty} S_m(f, \mathcal{E}(\mathbb{F}_p))t^m/m + \frac{1}{2}\sum_{m=1}^{+\infty}\sum_{n=1}^{+\infty} \frac{S_m(f, \mathcal{E}(\mathbb{F}_p))S_n(f, \mathcal{E}(\mathbb{F}_p))}{m\ n} t^{m+n} + \ldots$$

If we consider the coefficient of the polynomial of order 1, we obtain:

$$S_1(f, \mathcal{E}(\mathbb{F}_p)) = -\sum_{i=1}^{D} \omega_i.$$

The Riemann hypothetis for function fields (see [27] for the proof and the full version for the statement) shows that each zero of the above L-function verifies $|\theta_i| = 1/\sqrt{p}$. This boils down to $|S_1(f, \mathcal{E}(\mathbb{F}_p))| \leq \deg(f)\sqrt{p}$, which is the result required. Finally, we conclude by remarking that $\deg(f_a) = 2$. □

4.2 Randomness Extraction

We now show an equivalent of Theorem 8:

Theorem 14. *Let p be a n-bit prime, G a subgroup of $\mathcal{E}(\mathbb{F}_p)$ of cardinal q generated by P_0, q being a ℓ-bit prime, U_G a random variable uniformly distributed in G and k a positive integer. We have:*

$$\mathbf{SD}(\mathsf{lsb}_k\left(U_G\right), U_k) \leq 2^{(k+n+\log_2 n)/2+3-\ell}.$$

Proof. We follow the proof of Theorem 8, by constructing $\mathbb{1}(r,s,u) = \frac{1}{p} \times \sum_{a=0}^{p-1} e_p(a(f(rP_0) - f(sP_0) - Ku))$, where $\mathbb{1}(r,s,u)$ is the characteristic function which is equal to 1 if $f(rP_0) - f(sP_0) = Ku \bmod p$ and 0 otherwise. Therefore, we can evaluate $\mathrm{Col}(\mathsf{lsb}_k\left(x(U_G)\right))$ where U_G is uniformly distributed in G, exactly in the same way, and inject $M \leq 4\sqrt{p}$ in Equation (4) to obtain:

$$2\varepsilon \leq \sqrt{\frac{2^k}{p} + \frac{2^{k/2+2}\sqrt{p}\sqrt{\log_2(p)}}{q}}.$$

We conclude as before using the two inequalities $2^{n-1} < p \leq 2^n$ and $2^{\ell-1} < q \leq 2^\ell$ and remarking that the first term is negligible with respect to the second one:

$$2\varepsilon \leq 2^{(k-n-1)/2} + 2^{(k+n+\log_2(n))/2+3-\ell}. \qquad \square$$

Using the co-factor $\alpha = |\mathcal{E}(\mathbb{F}_p)| \,/\, |G| \leq 2^{n-\ell}$ of the elliptic curve, we obtain the following result:

Corollary 15. *Let e be a positive integer and let us suppose that this inequation is true:*

$$k \leq 2\ell - (n + 2e + \log_2(n) + 6) = n - (2\log_2(\alpha) + 2e + \log_2(n) + 6).$$

In this case, the application Ext_k is an $(U_G, 2^{-e})$-deterministic extractor.

5 Applications

Our extractor can be applied in every protocol which generates (possibly under the DDH assumption) a uniformly distributed element in a subgroup of $\mathbb{Z}_p^*$ or a random point over an elliptic curve, while a random bit-string is required afterwards. Our results are indeed quite useful in cryptographic protocols and primitives where one has to extract entropy from a Diffie-Hellman element.

5.1 Key Extraction

The most well known cryptographic primitive where randomness extractors are required is the key extraction phase of a key exchange protocol in order to create a secure channel. The key exchange can be either interactive (classical 2-party or group key exchange) or non-interactive (the Key Encapsulation Mechanism of an

hybrid encryption scheme). After a Diffie-Hellman key exchange (or ElGamal-like key encapsulation) performed over a group G, the parties share a common Diffie-Hellman element, which is indistinguishable from a uniformly distributed element in G granted the DDH assumption. However, they need a uniformly distributed bit-string to key a symmetric primitive: one thus extracts entropy from the DH element using a randomness extractor.

The two most well-known tools used for this task are hash functions (seen as random oracles [2]) and universal hash functions (in order to apply the Leftover Hash Lemma). Hash functions are the most often adopted solution, because of their flexibility and efficiency. However, they have a significant drawback: the validity of this technique holds in the random oracle model only. On the contrary, the Leftover Hash Lemma shows that the use of universal hash functions is secure in the standard model and that, if the cardinal of the group G is equal to q and if one wants to achieve a security of 2^{-e}, then one can extract $k = \log_2 q - 2e$ bits. However this solution requires some extra, public and perfectly random bits, which increases both time and communication complexities of the underlying protocols.

The truncation of the bit-string representation of the random element is definitely the most efficient randomness extractor, since it is deterministic, and it does not require any computation. However, the original results presented in [13, 16] were not as good as the Leftover Hash Lemma, from the number of extracted bit point of view. One could extract much less than $\log_2 q - 2e$ bits. In this paper, for large subgroups of $\mathbb{Z}_p^*$ (when the order q is larger than $p^{3/4} \cdot 2^e$), one extracts up to $\log_2 q - 2e$ bits, which is as good as the Leftover Hash Lemma. For large subgroups of an elliptic curve over $\mathbb{F}_p$, one extracts $n - 2e - \log_2(n) - 2\log_2(\alpha) - 6$ bits where α is the co-factor of the elliptic curve, which is not far from the Leftover Hash Lemma since, in practice, α is very small (often equal to 1). And then, for usual finite field size (p between 256 and 512), one can extract approximately $n - 2e - 16$.

Even with our improvement, the simple extractor may seem not very practical for subgroups of $\mathbb{Z}_p^*$, since quite large subgroups are needed. Indeed to generate a 256-bit string, with a 80-bit security and a 1024-bit prime p, one requires a 725-bit order subgroup, when the Leftover Hash Lemma would need a 416-bit order subgroup only: the time for exponentiation is approximately doubled. Note that, however, one saves on the time of the generation of extra randomness. Anyway, on elliptic curves, the improvement is quite meaningful, since groups in use are already quite large. The NIST elliptic curves have co-factor 1, and then on the 256-bit finite field elliptic curve, one can extract 82 bits, with a 80-bit security. On the 384-bit finite field, 214 bits can be extracted, while we can get 346 bits on the 521-bit field. This is clearly enough as symmetric key material for both privacy and authentication, without any additional cost.

We insist on the fact that it can apply for interactive key exchange, but also for the ElGamal [11] or Cramer-Shoup [9] encryption schemes.

5.2 NIST Random Generator

The very recent NIST SP 800-90 elliptic curve Dual Random Bit Generator (DRBG) [24] has been approved in the ANSI X9.82 standard in 2006. Based on the elliptic curves, the design of this random bit generator (RBG) is adapted from the Blum-Micali generator. At Crypto '07, Brown and Gjøsteen [7] adapted the Blum-Micali RNG security proof to show that the DRBG output is indistinguishable from a uniformly distributed bit-string for all computationally limited adversaries. In this section, we show that our result allows to improve this security result at two different places. The first improvement reduces the number of assumptions on which the security proof relies. The second one decreases the implicit security bound given in [7].

Getting Rid of TPP Assumption. The security result of [7] holds under three computational assumptions: the classical decisional Diffie-Hellman problem (DDH), the new x-logarithm problem (XLP) (which states that, given an elliptic curve point, it is hard to determine whether the discrete logarithm of this point is congruent to the x-coordinate of an elliptic curve point), and the truncated point problem (TPP). The latter TPP states that, given a k-bit string, it is hard to tell if it was generated by truncating the x-coordinate of a random elliptic curve point or if it was chosen uniformly at random. This problem is exactly the problem we studied in this paper. In section 4, we proved that this problem is indeed hard if the elliptic curve is defined over $\mathbb{F}_p$ (where p is an n-bit prime) and if $k = n - 2e - 2\log_2(\alpha) - \log_2(n)$ bits are kept after the truncation (we remind that e denotes the expected security level and α the cofactor of the elliptic curve). Therefore, our result strengthens the security proof of [7] since thanks to it, when the elliptic curve is defined over $\mathbb{F}_p$ of appropriate size, the TPP assumption actually holds, and thus their security proof relies on the DDH and XLP assumptions only.

It is interesting to note that Brown and Gjøsteen [7], when making their highly heuristic assumptions, estimated that the expected number of bits that could be kept after truncation would be approximately $k = n - 2e - C$ where C is some constant (if the cofactor of the elliptic curve is equal to 1). Our result confirms this heuristic analysis, but is more precise since it *proves* that *in all cases* we can keep *at least* $k = n - 2e - \log_2(n)$ bits. However, we recall Brown and Gjøsteen's warning and recommend to skip $2e + \log_2(n)$ bits of the elliptic curve point abscissa in the ECRNG.

Improvement of the Security Bound. Finally, our result also allows to improve the security bound of [7]. For the sake of clarity, this security bound is not explicitly stated in [7], but can be recovered from the proof. At the very last stage of the proof, the TPP assumption is used to show that if $Z_1, \ldots, Z_m$ are uniformly distributed points on the elliptic curve and if $b_1, \ldots, b_m$ are uniformly distributed k-bit strings, then $(\mathsf{lsb}_k(Z_1), \ldots, \mathsf{lsb}_k(Z_m))$ is indistinguishable from $(b_1, \ldots, b_m)$. If any adversary has a probability of successfully distinguishing

$\mathsf{lsb}_k(Z_1)$ from b_1 smaller than δ, a classical hybrid argument implies that any adversary has a probability of successfully distinguishing $(\mathsf{lsb}_k(Z_1), \ldots, \mathsf{lsb}_k(Z_m))$ from $(b_1, \ldots, b_m)$ smaller than $m \cdot \delta$. This bound can be improved to $\sqrt{2m/\pi} \cdot \delta$.

First, notice that in our case, δ is equal to $2^{(k+\log_2 n+2\log_2(\alpha)-n)/2}$. Using a result that can be found in [1], one can show that the advantage of the best adversary in distinguishing to two above m-uples is approximately equal to $\sqrt{m \cdot (2^k \cdot \mathrm{Col}(\mathsf{lsb}_k(Z_1)) - 1)/2\pi}$, if $2^k \cdot \mathrm{Col}(\mathsf{lsb}_k(Z_1)) - 1 \ll 1$. The latter expression $2^k \cdot \mathrm{Col}(\mathsf{lsb}_k(Z_1)) - 1$ is exactly the one we upper-bounded in the proof in Section 4: it is smaller than $2^{k+\log_2(n)+2\log_2(\alpha)-n+2} = 4\delta^2$. This implies that, if $\delta \ll 1$, the advantage of the best adversary in distinguishing $(\mathsf{lsb}_k(Z_1), \ldots, \mathsf{lsb}_k(Z_m))$ from $(b_1, \ldots, b_m)$ is upper bounded by $\sqrt{2m/\pi} \cdot \delta$. We thus improve the bound from [7] by a factor $\sqrt{m}$.

Acknowledgements

This work was supported in part by the French ANR-07-SESU-008-01 PAMPA Project and the European ICT-2007-216646 ECRYPT II Contract.

References

1. T. Baignères, P. Junod, and S. Vaudenay. How far can we go beyond linear cryptanalysis? In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 432–450. Springer, Dec. 2004.
2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993.
3. E. Bombieri. On exponential sums in finite fields. In *American Journal of Mathematics*, volume 88, pages 71–105. The Johns Hopkins University Press, 1966.
4. D. Boneh. The decision Diffie-Hellman problem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *LNCS*. Springer, 1998. Invited paper.
5. D. Boneh and I. Shparlinski. On the unpredictability of bits of the elliptic curve Diffie-Hellman scheme. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 201–212. Springer, Aug. 2001.
6. D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 129–142. Springer, Aug. 1996.
7. D. R. L. Brown and K. Gjøsteen. A security analysis of the NIST SP 800–90 elliptic curve random number generator. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 466–481. Springer, Aug. 2007.
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. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, Aug. 1998.
10. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

11. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Aug. 1985.
12. T. ElGamal. On computing logarithms over finite fields. In H. C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, pages 396–402. Springer, Aug. 1986.
13. P.-A. Fouque, D. Pointcheval, J. Stern, and S. Zimmer. Hardness of distinguishing the MSB or LSB of secret keys in Diffie-Hellman schemes. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 240–251. Springer, July 2006.
14. P.-A. Fouque, D. Pointcheval, and S. Zimmer. HMAC is a randomness extractor and applications to TLS. In M. Abe and V. D. Gligor, editors, *ASIACCS*, pages 21–32. ACM, 2008.
15. R. Gennaro, H. Krawczyk, and T. Rabin. Secure Hashed Diffie-Hellman over non-DDH groups. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 361–381. Springer, May 2004.
16. N. Gürel. Extracting bits from coordinates of a point of an elliptic curve. Cryptology ePrint Archive, Report 2005/324, 2005. `http://eprint.iacr.org/`.
17. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
18. 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.
19. R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proc. of the 30th FOCS*, pages 248–253. IEEE, New York, 1989.
20. D. Jetchev and R. Venkatesan. Bits security of the elliptic curve diffie-hellman secret keys. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 75–92. Springer, Aug. 2008.
21. D. R. Kohel and I. E. Shparlinski. On exponential sums and group generators for elliptic curves over finite fields. In *Proceedings of the 4th International Symposium on Algorithmic Number Theory*, volume 1838 of *LNCS*, pages 395–404, 2000.
22. S. V. Konyagin and I. Shparlinski. *Character Sums With Exponential Functions and Their Applications*. Cambridge University Press, Cambridge, 1999.
23. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, Oct. 1997.
24. NIST. Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Mar. 2007. NIST Special Publications 800-90. `http://csrc.nist.gov/publications/PubsSPs.html`.
25. V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, Cambridge, 2005.
26. L. Washington. *Elliptic Curves: Number Theory and Cryptography*. CRC Press, 2003.
27. A. Weil. Sur les courbes algébriques et les variétés qui s'en déduisent. In *Actualités scientifiques et industrielles, Publications de l'institut de Mathématique de l'université de Strasbourg*, volume 1041, Paris, 1948. Hermann.

A Relation Between Collision Probability and Statistical Distance

In this section we prove the following lemma. The proof is taken from [25] and is given here for the sake of completeness. Note that this lemma is a consequence of the Cauchy-Schwarz inequality, which implies that the smaller the statistical distance is, the tighter the inequation is (if X is uniformly distributed, then the inequality is an equality).

Lemma 4. Let X be a random variable with values in a set $\mathcal{X}$ of size $|\mathcal{X}|$ and $\varepsilon = SD(X, U_{\mathcal{X}})$ be the statistical distance between X and $U_{\mathcal{X}}$ a random variable uniformly distributed over $\mathcal{X}$. We have:

$$\mathrm{Col}(X) \geq \frac{1 + 4\varepsilon^2}{|\mathcal{X}|}.$$

To prove the lemma we need the following result which states that norm 1 is smaller than norm 2.

Lemma 16. *Let $\mathcal{X}$ be a finite set and $(\alpha_x)_{x\in\mathcal{X}}$ a sequence of real numbers. We have:*

$$\frac{\left(\sum_{x\in\mathcal{X}} |\alpha_x|\right)^2}{|\mathcal{X}|} \leq \sum_{x\in\mathcal{X}} \alpha_x^2. \tag{5}$$

Proof. This inequality is a direct consequence of Cauchy-Schwarz inequality:

$$\sum_{x\in\mathcal{X}} |\alpha_x| = \sum_{x\in\mathcal{X}} |\alpha_x| \cdot 1 \leq \sqrt{\sum_{x\in\mathcal{X}} \alpha_x^2} \cdot \sqrt{\sum_{x\in\mathcal{X}} 1^2} \leq \sqrt{|\mathcal{X}|}\sqrt{\sum_{x\in\mathcal{X}} \alpha_x^2}.$$

The result can be deduced easily. □

If X is a random variable with values in $\mathcal{X}$ and if we consider that $\alpha_x = \Pr[X = x]$, then, since the sum of probabilities is equal to 1, and since $\mathrm{Col}(X) = \sum_{x\in\mathcal{X}} \Pr[X = x]^2$, we have:

$$\frac{1}{|\mathcal{X}|} \leq \mathrm{Col}(X). \tag{6}$$

We are now able to prove the above Lemma 4.

Proof. If $\varepsilon = 0$ the result is an easy consequence of equation Equation (6). Let assume that ε is different from 0. Let define $q_x = |\Pr[X = x] - 1/|\mathcal{X}||/(2\varepsilon)$, we have $\sum_x q_x = 1$. According to equation Equation (5), we have:

$$\begin{aligned}\frac{1}{|\mathcal{X}|} \leq \sum_{x\in\mathcal{X}} q_x^2 &= \sum_{x\in\mathcal{X}} \frac{\left(\Pr[X = x] - 1/|\mathcal{X}|\right)^2}{4\varepsilon^2} = \frac{1}{4\varepsilon^2}\left(\sum_{x\in\mathcal{X}} \Pr[X = x]^2 - 1/|\mathcal{X}|\right) \\ &\leq \frac{1}{4\varepsilon^2}\left(\mathrm{Col}(X) - 1/|\mathcal{X}|\right).\end{aligned}$$

The lemma can be deduced easily. □