

# Security of Biased Sources for Cryptographic Keys

Preda Mihăilescu

**Abstract.** Cryptographic schemes are based on keys which are highly involved in granting their security. It is in general assumed that the source producing these keys has uniformly distribution, that is, it produces keys from a given key space with equal probability. Consequently, deviations from uniform distribution of the key source may be regarded a priori as a potential security breach, even if no dedicated attack is known, which might take advantage of these deviations.

We propose in this paper a model for biased key sources and show that it is possible to prove some results about tolerance of biases, which have the property of being inherent to the bias itself and not requiring assumptions about unknown attacks, using these biases. The model is based on comparing the average case complexities of generic attacks to some number theoretical problems, with respect to uniform and to biased distributions.

We also show the connection to information entropy based analysis of biased sources, which was used in earlier works, for suggesting the tolerance of biased sources.

## 1. Introduction

Cryptographic schemes, both secret and public key ones, use keys or sets of keys upon which their security is based. It is assumed in general that the keys are produced by a random uniform distributed source, so that the probability for every key from a given key space is constant. The perception, that a random uniform distribution of the cryptographic keys is a minimal security constraint to be respected by a safe implementation, is therefore quite widespread. This perception is based on the following reasoning: breaking a scheme based on uniform random distributed keys requires not less than the general state of the art attack against the given scheme. If the key generation has some bias, on the other hand, the bias could make it vulnerable. One may suppose that some – still unknown – algorithm can exist, which takes advantage of the given bias in order to break the scheme considerably faster than the state of the art general solution would.

Purely uniformly distributed random sources are not easy to provide in practice and are also hard to prove. This may be due to the practical difficulty for creating random bits – as the Netscape flaw proved, several years ago – or to the fact that provably secure random sources may involve a considerable computational overhead. Several authors – Bach [1], [2], Peralta, Shoup [12], [14], etc. – therefore explicitly introduced the notion that *random bits are not for free* into the analysis of algorithms. It may also be that some key generation algorithm, which has slight distributional biases, enjoys further advantages which one would prefer to conserve, provided the security of the bias may be ascertained.

The primary conception that bias equals insecurity may therefore be too strong. One would like to have some simple measures allowing one to distinguish between *harmless* biases and possibly dangerous ones. It is the purpose of this paper to provide a simple model for biases, based on complexity and measure theories, which allows defining the tolerability of a bias by an intrinsic measure of the bias distribution. It is then shown that attacks, which would successfully take advantage of tolerable biases, implicitly improve upon the state of the art attack against the same scheme using uniformly distributed key sources. Of course, there is no general protection against such improvements, and avoiding small biases is therefore not the place in which to invest for obtaining better security.

Our concepts are naturally related to the powerful theory of *average case complexity*, initiated by Levin [8] and extended by Gurevitch [5], [6] Ben-David, Chor, Goldreich and Luby [3], etc. In terms of this theory, we compare the average case complexity of some unknown algorithm – which is optimally adapted to take advantage of a given biased distribution – with respect to two different distributions: the uniform and a given biased one. We thus derive bounds which are intrinsic to the biased distribution, either in terms of measure or of entropy. The class of polynomially bounded biases itself will be defined in the spirit of related average case complexity definitions.

By working with the model, it becomes clear that it is rather powerful in dealing with very general key sources. We focus our attention, however, primarily upon public key cryptographic schemes which base their security upon the assumption that one of the following number theoretic problems is hard:

IF The problem of factoring rational integers.

DL The discrete logarithm in the multiplicative group of a finite field.

EDL The discrete logarithm in the group structure of an elliptic curve over a finite field.

We shall thus not load the model with more generality than necessary for treating these cases in an unified way. Possible further generalizations are obvious and may be made by cryptographers when concrete problems which do not fit in our restricted presentation, raise similar distribution questions. We give examples for the problems which we take into consideration. These will show both useful biases, which are tolerable and such ones which are definitely not, with respect to this model and from the point of view of *common sense*.

## 2. Key and Token Spaces

We start with some examples which will illustrate several issues to be taken into account when formalizing notions like key, key space, key token, etc. We want to define these notions in the following sense: Let  $\mathcal{S}$  denote some source which generates *key tokens*  $\tau_i, i = 1, 2, \dots, k$  from given *token spaces*  $\mathcal{I}_i$ . These tokens are combined by a key generation function

$$\kappa : \mathcal{K}_0 = \mathcal{I}_1 \times \mathcal{I}_2 \times \dots \times \mathcal{I}_k \longrightarrow \mathcal{K},$$

where  $\mathcal{K}$  is a key space of some cryptographic scheme.

For the secret key, the combination can be the identity, since the key holder is allowed to know the single tokens out of which his key is built. The key generation function typically *hides* information when building public keys out of their tokens. We will not be concerned with secret keys, since attacks against these keys are physical or organizational and cannot be protected against cryptographically. Domain parameters are the common data to which individuals add secret tokens in order to produce key pairs. As such, they can display weaknesses which concern the whole domain. For instance if an elliptic curve with Frobenius trace 1 is chosen as a domain parameter for a scheme based on EDL, the curve is prone to the Smart attack [15]. Domain keys are therefore also part of the topic.

**Example 2.1.** *For the DL problem, the usual tokens are  $\tau_1 = p$ , a large prime, and  $\tau_2 = \alpha \in \mathbb{F}_p^*$ , an element generating a subgroup of large size in the multiplicative group  $\mathbb{F}_p^*$ . In the RSA scheme,  $\tau_1 = p, \tau_2 = q$  are two primes so that  $\kappa(p, q) = p \cdot q = n$  is one part of the public key. The other part, the public exponent is sometimes fixed, but may in general be regarded as a key token. For a fixed public modulus size of  $2m$  bits, the key space may be defined in different ways. It is in most cases contained in the set:  $S = \{p, q : 2^{2m-1} < N := p \cdot q < 2^{2m}\}$ , with  $p$  and  $q$  primes, although the condition  $N < 2^{2m}$  can be found sometimes relaxed to,  $N < 2^{2m+l}$ , with, e.g.  $l = 7$ . Even for  $l = 0$ , the primes  $p$  and  $q$  are subject to additional conditions which can be more or less severe. One set of conditions concerns the sizes of  $p$  and  $q$  and it ranges from  $p, q > 2^{(1-\varepsilon)m}$  for some fixed fraction  $\varepsilon$  to  $2^m > p, q > \sqrt{2} \cdot 2^{m-1}$ . Other conditions restrict the  $\gcd(p-1, q-1)$ . The resulting key spaces therefore have slight differences.*

Finally, for the EL problem, the tokens  $\tau_1 = p$ , with  $p$  prime some integer,  $\tau_{2,3} = A, B$  are the parameters of an elliptic curve which is a domain key:

$$\kappa(p; A, B) = \mathcal{E}_{p,h} : Y^2 = X^3 + A \cdot X + B.$$

An important implicit token is  $N(p; A, B) = \#\mathcal{E}_p$ , the number of points of the elliptic curve  $\mathcal{E}_p$  which can be evaluated faster by specifying an order  $\Lambda = [1, f\omega] \subset \mathcal{O}(\mathbb{K})$  in the integers of an imaginary quadratic extension  $\mathbb{K}$ , in which  $\mathcal{E}_p$  will have complex multiplication. This specification fixes  $A$  and  $B$  and herewith a possible way to determine a curve consists of specifying  $\mathbb{K}$  and  $f$ . For computational reasons, the discriminant of  $\mathbb{K}$  must be polynomial in  $\log p$ , and therefore the methods

relying on fixing the complex multiplication produce an exponentially thin key subspace of all the smooth elliptic curves modulo  $p$ . This can be produced by choosing  $A$  and  $B$  randomly. After that, one counts the number of points on the resulting curve  $\mathcal{E}$ , subject to some restrictions which do not affect the size of the key space.

Primarily, both keys and tokens are *sets of parameters*. We shall restrict the notion of *key* only by one condition. A key is an organization level of parameters at which an attack against a public key scheme can be conducted. In this sense, both public and domain parameters are considered to be keys. Also, a key *need not* be the *complete public key* of a given algorithm. For instance for the RSA algorithm, the *public modulus* is a key against which an individual factoring attack can be carried out, although it requires one additional *token*, the public exponent, in order to complete the public key. The range of possible attacks may increase by using this token too, but the modulus in itself is definitely a meaningful key, while the public exponent is not. A secret key is equal to a public key plus information about the building tokens. A domain key is not a key in the restricted sense, since it has insufficient information in order to be the object of an attack. Its choice may however influence the possibility of attacking public keys which are built using the domain key. This is the case, for instance, for most supersingular elliptic curves.

We define our notions accordingly:

**Definition 2.2.** A token is a parameter or a parameter-tuple  $\tau \in \mathbb{Z}^k$ , for  $k \geq 1$ . A key is a set of parameters  $K \in \mathbb{Z}^N$ , built by a key generation function acting on a product of token spaces. A key is subject to attacks that aim to reveal the tokens it is built from. Keys are defined by:

1. *Algorithm and functional type.* The algorithm describes a public key scheme for which the key is used. The type defines the keys' functionality – e.g. RSA public modulus.
2. *Size parameters.*
3. *Token dependencies.*

A key space  $\mathcal{K}$  is the set of all keys sharing the same definitions 1.-3. together with a set of conditional token distributions reflecting the token dependencies.

While the first two items completely define the functionality of a key, the token dependencies may, as shown in the example, induce variations of the key spaces for the same functional keys. In particular, token dependencies may be empty, meaning that all tokens are statistically independent, or they may be given by a set of conditional probabilities. It happens that more tokens are bound by a reciprocal condition. For instance, the primes of which an RSA public modulus is built cannot be *too close* to each other, etc. In such cases, the last generated token can depend upon the choice of the previous ones.

We consider generic attacks against the underlying *hard* problem and their average case complexity. It is our purpose to compare the average case complexities in dependency of different distributions of the keys. Firstly, this means that keys shall be defined so that they are in one to one correspondence with instances of the

problem. Secondly the key spaces must be invariant under different key generation strategies. Finally, conditional distributions need to be defined in every case. With these, a distribution of the keys can be derived from the distributions of single. Note that our definitions meet these conditions; however, we can only compare sources at the level of identical key spaces. In particular, variations induced by token restrictions on key spaces for the same functional keys, are not object of our approach. They can be addressed by related methods, but we shall not consider this problem here.

### 3. Distributions

We shall give some definitions of distributions and attack strategies which are based on average case complexity using [3], [5].

If  $\mathcal{K}$  is a key space, let  $s : \mathcal{K} \rightarrow \mathbb{N}$  be the size parameter and  $L = s(\mathcal{K}) \subset \mathbb{N}$ . For  $n \in L$ , let  $\mathcal{K}_n$  be the key set of size  $n$ . The function  $s$  needs not be surjective, but for  $n, m \in L$ ,  $n \neq m \Rightarrow \mathcal{K}_n \neq \mathcal{K}_m$ . We assume additionally that  $\mathcal{K}$  is endowed with an enumeration function  $\phi : \mathcal{K} \rightarrow \mathbb{N}$  with

$$x \in \mathcal{K}_n, y \in \mathcal{K}_m, n < m \Rightarrow \phi(x) < \phi(y).$$

We wish next to define size independent distributions on the key space, in accordance with average case complexity. If  $\mu : \mathcal{K} \rightarrow [0, 1]$  is a probability density function, then  $\mu_n := \mu(k | s(k) = n)$  is a conditional density defined on the space of size  $n$  keys. Since the key space of different sizes are disjoint, the event  $s(K) = n$  has a probability function  $\rho : \mathbb{N} \rightarrow [0, 1]$  with

$$\begin{aligned} \rho(n) &= 0, \quad \text{for } n \notin L \\ \mu(k) &= \rho(s(k)) \cdot \mu_{s(k)}(k), \quad \text{for } k \in \mathcal{K} \text{ and} \\ \sum_{n>1} \rho(n) &= 1. \end{aligned} \tag{1}$$

It is also reasonable to assume that there is a polynomial  $b \in \mathbb{Z}[x]$  such that  $\rho(n) \geq \frac{1}{d(n)}$ ,  $\forall n \in L$  (see [5], Proposition 1.1): In most practical cases, the set  $L$  may even be considered to be finite. Note that the function  $\rho$  allows building averages over finite and countably infinite sets  $L$ . It has no specific empirical correspondence. We may therefore assume that it is shared by *all* probability functions describing key generation sources for  $\mathcal{K}$ .

For a fixed key size  $s(k) = n$ , a uniform source  $\mathcal{S}_u$  will draw uniform random distributed instances from each token space  $\mathcal{I}_i$ , so that the probability of drawing any token  $\tau_i \in \mathcal{I}_i$  will be the same:  $\ell_{n,i} = \frac{1}{|\mathcal{I}_i|}$ , where  $|\mathcal{I}_i|$  is the number of elements in  $\mathcal{I}_i$ . If the token spaces are independent, the probability  $\ell_n = \prod_i \ell_{n,i}$ . Otherwise,  $\ell_n$  is still constant, but depends on the set of conditional probabilities 3 in definition 1. Note that the constant  $\ell_n$  is the probability of the uniform distribution at each key  $k \in \mathcal{K}_n$ . We may thus write  $\ell_n(k)$  when the accent is on

the uniform distribution and  $\ell_n$  when we are performing computations with the (constant) value of its probability function.

We shall assume that the token spaces are independent and let thus  $\mu_{n,i}$  be the probability function on token space  $\mathcal{I}_i$  for key size  $n$ . We write for the number of elements  $m(n) := |\mathcal{K}_n|$  and  $m_i(n) := |\mathcal{I}_i|$ . Thus  $\ell_n \cdot m(n) = 1$  and  $\ell_{n,i} \cdot m_i(n) = 1$ . If  $X \subset \mathcal{K}_n$ , and  $g : \mathcal{K}_n \rightarrow \mathbb{R}$ , we shall write  $\sum_X g(k) := \sum_{k \in X} g(k)$ . The probability of the event that a key  $k \in X$  will be denoted by

$$\mathcal{P}(X) := \sum_X \mu(k) = P(k|k \in X).$$

Let  $(S, \ell)$  be a randomized unique solution search problem [3] – such as all number theoretical problems relevant for cryptography may be considered – and  $\sigma_S$  its standard (or *state of the art*) solution for the uniform distribution  $\ell$ . Let  $f_S(k)$  be the operation count of  $\sigma_S$  when applied to the key  $k \in \mathcal{K}$  and

$$\Theta_n = \sum_{k \in \mathcal{K}_n} f_S(k) \ell_n(k) \quad (2)$$

the expected operation count of  $\sigma_S$  for size  $n$  keys and  $\Theta = \sum_{n \in S} \rho(n) \Theta_n$ .

Let  $\mu_n$  be the probability functions corresponding to some *biased* key generation source  $\mathcal{S}$  on  $\mathcal{K}_n$ ,  $n \in L$ . The local bias of  $\mathcal{S}$  at  $k \in \mathcal{K}_n$  is given by  $\lambda_n(k) := \mu_n(k)/\ell_n$  and is normed by  $\sum_{k \in \mathcal{K}_n} \lambda_n(k) = 1/\ell_n \sum_{\mathcal{K}_n} \mu_n(k) = m(n)$ . This is an obvious measure of the deviation from uniform distribution. The most likely way to take advantage of biases consists of finding efficient attacks for the region of the token/key space where a given source has extreme bias. We consider briefly the local bias on token space  $\mathcal{I}_i$  with size  $m_{n,i} = 1/\ell_{n,i} = |\mathcal{I}_i|$ . First note that an important consequence of bias is the existence of keys which may never be produced by a given source. Let

$$E_{\mathcal{I}_i} := \{k_i \in \mathcal{I}_i : \mu_{n,i}(k_i) > 0\} \quad (3)$$

be the *effective subspace* of  $\mathcal{I}$  for the source  $\mathcal{S}$ . If  $E_n$  is defined consequently for the space  $\mathcal{K}_n$ , we let  $e(n) = \frac{|E_n|}{m(n)}$ , the function measuring the relative weight of the effective subspace for different key sizes and

$$\mu'_{n,i}(k) = \mu_{n,i}(k|k \in E_{\mathcal{I}_i}) = e(n)^{-1} \cdot \mu_{n,i}(k)$$

be the conditional probability that a given key, which has a positive chance to be produced, is generated by source  $\mathcal{S}$ . We define the complement  $Z_{\mathcal{I}_i} = \mathcal{I}_i \setminus E_{\mathcal{I}_i}$  of the effective set to be the *hidden subspace* of  $\mathcal{I}_i$ , and similarly for  $\mathcal{K}_n$  and for  $\mathcal{K}$ .

The following definition splits a token set in three subsets, two of which have extremely large and small biases, respectively, while the third one contains the tokens with bias close to 1:

**Definition 3.1.** For  $c > 1$ , let

$$\begin{aligned} A_{\mathcal{I}}(b) &:= \{k_i \in \mathcal{I} : \lambda_{n,i}(k_i) \geq m_i(n)^b\}, \\ B_{\mathcal{I}}(b) &:= \{k_i \in \mathcal{I} : \lambda_{n,i}(k_i) \leq m_i(n)^{-b}\}, \\ C_{\mathcal{I}}(b) &:= \mathcal{I} \setminus (A_{\mathcal{I}}(b) \cup B_{\mathcal{I}}(b) \cup Z_{\mathcal{I}}). \end{aligned} \quad (4)$$

When the token space  $\mathcal{I} = \mathcal{I}_i$  is indexed, we shall write for simplicity:  $X_i(b) := X_{\mathcal{I}_i}(b)$ , for any of  $X = A, B, C, E, Z$ . Also, when  $\mathcal{I} = \mathcal{K}_n$ , there will be no subscript, thus  $X(\varepsilon) = X_{\mathcal{K}}(\varepsilon)$ .

The source  $\mathcal{S}$  has polynomially bounded bias on the token space  $\mathcal{I}_i$  if there is a  $\delta \in (0, 1)$  such that:

$$\begin{aligned} \forall \varepsilon > 0 \quad \exists n_0 \in \mathbb{N} : \quad \mathcal{P}(A_i(\varepsilon)) &\leq \delta/2, \quad \forall n > n_0, \\ |B_i(\varepsilon)| &\leq \delta/2 \cdot \frac{m_i(n)^{1+\varepsilon}}{\Theta_n}. \end{aligned} \quad (5)$$

The bias of  $\mathcal{S}$  is polynomially bounded on the key space, iff (5) holds for each token space uniformly.

Note that our definition is more restrictive than requiring that the  $\ell$ -average of the bias is polynomial. This condition would in fact allow the probability function to be centered on a small subset of  $\mathcal{I}$  with very high local biases, which we do not want to happen. It follows from (5), that

$$\begin{aligned} |C_i(\varepsilon)| &> m_i(n)(1 - \delta m_i(n)^{-\varepsilon}) \quad \text{and} \\ \mathcal{P}(C_i(\varepsilon)) &> 1 - \delta, \end{aligned} \quad (6)$$

for  $n > n_0(\varepsilon)$ . Evidently the bound on the sum of probabilities of events with large and small biases also limits the size of the set of these events.

**Definition 3.2.** We say that the bias is conditionally polynomially bounded, if it is polynomially biased for the conditional probability  $\mu'_{n,i}$ . Equivalently, the restriction of the source to the effective set:  $S/E$  is polynomially biased. The function  $e(n)$  is in this case the conditional distribution. We say the source  $\mathcal{S}$  is **polynomially dense**, if

$$\forall \varepsilon > 0, \quad \exists n_0 : \quad e(n) > m(n)^{-\varepsilon}, \quad \forall n > n_0.$$

## 4. Strategies

If  $f : \mathcal{K} \rightarrow \mathbb{R}$  is a positive valued function, we define the expectations

$$E(f, \mu_n) := \sum_{k \in \mathcal{K}_n} f(k) \mu_n(k). \quad (7)$$

An attacker who tries to break the scheme for which keys were generated using the source  $\mathcal{S}$ , will use some strategy  $\sigma$  based on one or a combination of algorithms. Let  $\sigma$  be given by an oracle which acts optimally in accordance to the bias of  $\mu$  and let  $f = f_{\sigma} : \mathcal{K} \rightarrow \mathbb{R}$  be a function giving the expected number of operations required by  $\sigma$  for solving  $(S, \mu)$  for a given key  $k \in \mathcal{K}$ .

For fixed  $n$ , the average running time of the attacker's strategy, when the keys are generated by  $\mathcal{S}$  will thus be

$$\theta_n(\mathcal{S}) = E(f, \mu_n).$$

By (2), the state of the art strategy has running time  $\theta_n = E(f_S, \ell_n)$ . It is natural to assume that  $\theta_n$  is superpolynomial in  $\log m(n)$ , which is a measure of the uncertainty about the uniform distributed input key. We can also assume for any strategy  $\sigma$ , without restriction of generality, that

$$f_\sigma(k) \leq \Theta_n, \quad \forall k \in \mathcal{K}_n. \quad (8)$$

Since the average case run time of the state of the art algorithm  $\sigma_S$  is superpolynomial with respect to the uniform distribution, it is natural to define a biased distribution as tolerable, to be one with respect to which the average case run time does not differ from  $\Theta_n$  by more than a polynomial amount.

**Definition 4.1.** *Let  $\mathcal{S}, \mathcal{S}'$  be two sources for the key space  $\mathcal{K}$  with  $m(n) = |\mathcal{K}_n|$ ,  $\mu, \mu'$  their probability functions and  $f, f'$  be the run time functions for some optimal strategy oracles for  $\mathcal{S}$  with respect to the probability functions. We say  $\mathcal{S} \sim \mathcal{S}'$  are polynomially equivalent if*

$$\forall \varepsilon > 0, \exists n_0 : n > n_0 \Rightarrow m(n)^{-\varepsilon} E(f, \mu_n) < E(f', \mu'_n) < m(n)^\varepsilon E(f, \mu_n),$$

for all  $f, f'$ . The source  $\mathcal{S}$  is said to have **tolerable bias**, iff  $\mathcal{S} \sim \mathcal{S}_u$ . The bias is **conditionally tolerable**, if the source is polynomially dense and the bias is tolerable on the effective set  $E(\mathcal{S})$ .

The definition of tolerable biases can be verified by using only their intrinsic distribution and without making assumptions about individual attacks. In particular, biases which can be proved to be tolerable bring no risks, while nothing can be said about biases which cannot be proved to be tolerable. This is a clear advantage. The condition for the density  $e_n(\mathcal{S}) = 1$  is however very restrictive and dropping it to the condition of polynomial bounded density will allow more flexible definitions of bias tolerance.

The main result about security of biased distributions is:

**Theorem 4.2.** *Polynomially bounded biases are tolerable.*

*Proof.* Let us consider the single token space case first (e.g. DL domain parameters). The proof will take advantage of the fact that the sets of keys with extreme low biases have at most polynomial contribution to the average case complexity. The expectations  $E(f, \mu_n)$  are essentially determined by the values which  $f$  takes on  $C(\varepsilon)$ .

We compare  $E(f, \mu_n)$  to  $E(f, \ell_n)$  for given  $\varepsilon > 0$ . It is natural to split the evaluation of the expected values in three parts  $\sum_X$ , for  $X = A, B, C$ . Let  $\varepsilon' > 0$

be such that  $m(n)^{\varepsilon'} + 1 < m(n)^\varepsilon$  for  $n > n_0(\varepsilon)$ . For such  $n$ ,

$$\begin{aligned} \sigma(\mathcal{S}) &= \sum_{k \in \mathcal{K}_n} \mu_n(k) \cdot f(k) = \left( \sum_{C(\varepsilon')} + \sum_{A(\varepsilon')} + \sum_{B(\varepsilon')} \right) (\mu_n(k) \cdot f(k)) \\ &< m(n)^{\varepsilon'} \cdot \sum_{C(\varepsilon')} \ell_n f(k) + \sum_{A(\varepsilon') \cup B(\varepsilon')} \mu_n(k) f(k) \\ &= (|C(\varepsilon')|) m(n)^{-1+\varepsilon'} E(f, \ell_n) + \sum_{A(\varepsilon') \cup B(\varepsilon')} \mu_n(k) f(k). \end{aligned}$$

The contribution of the set  $A(\varepsilon') \cup B(\varepsilon')$  is easily bounded using (8) and the bounds (6), so  $\sum_{A(\varepsilon') \cup B(\varepsilon')} \mu_n(k) f(k) < \delta E(f, \ell_n)$  and:

$$\sigma(\mathcal{S}) < (m(n)^{\varepsilon'} + \delta) E(f, \ell_n) < m(n)^\varepsilon E(f, \ell_n)$$

Conversily, let  $\varepsilon > 0$  and  $n > n_0(\varepsilon)$ . Then:

$$\begin{aligned} \sigma(\mathcal{S}) &= \sum_{k \in \mathcal{K}_n} \mu_n(k) \cdot f(k) = \left( \sum_{C(\varepsilon) \cup A(\varepsilon)} + \sum_{B(\varepsilon)} \right) (\mu_n(k) \cdot f(k)) \\ &> m(n)^{-\varepsilon} \sum_{C(\varepsilon) \cup A(\varepsilon)} \ell_n f(k) + \sum_{B(\varepsilon)} \mu_n(k) \cdot f(k) \\ &> m(n)^{-\varepsilon} E(f, \ell_n) - m(n)^{-\varepsilon} \ell_n \sum_{B(\varepsilon)} (1 - \lambda_n(k)) f(k) \\ &> m(n)^{-\varepsilon} (E(f, \ell_n) - |B(\varepsilon)| \ell_n \Theta_n) \\ &> m(n)^{-\varepsilon} E(f, \ell_n) - \delta/2. \end{aligned}$$

It follows that the expectations  $E(f, \ell_n) \sim E(f, \mu_n)$  for this case.

Assuming that the token space probabilities  $\mu_{n,i}$  are independent, the general proof follows by induction on the number of token spaces.  $\square$

The statement for conditional bounded biases is a direct consequence of this proposition:

**Corollary 4.3.** *A polynomially dense source with conditionally polynomially bounded bias is conditionally tolerable.*

*Proof.* The source is dense and by the previous proposition, its restriction to the effective set has tolerable bias. By definition then, it has conditionally tolerable bias.  $\square$

**Remark 4.4.** *Note that the condition that density  $e(n) \rightarrow 1$ , which follows from (6), is necessary in the proof of theorem 4.2. Without it, there is technically no means for bounding the expected value of the uniformly distributed strategy on the hidden space. It is however a strongly restrictive condition and it is sensible to loosen it by allowing more general sources, which are asymptotically close to  $e(n) = 1$ , even if the density does not converge to 1. We shall see that the choice made in the definition of conditionally tolerable biases is natural from a further perspective, the one of entropy measures.*

## 5. Entropy

It has been argued in earlier research [4], [11], that biases can be tolerated when the *entropy* of the biased source is asymptotically equal to the entropy of the uniform distributed source. We shall investigate in this section the connection between the entropy approach and bias bounds.

According to Shannon's definition, the entropy of an uniform distributed source is:

$$H_n(\mathcal{S}_u) := - \sum_{k \in \mathcal{K}_n} \ell_n \cdot \log \ell_n = \log \ell_n.$$

A biased source  $\mathcal{S}$  has the entropy

$$H_n(\mathcal{S}) := - \sum_{k \in \mathcal{K}_n} \mu_n(k) \cdot \log \mu_n(k).$$

When the source  $\mathcal{S}$  is fixed and  $X \subset \mathcal{K}_n$  we shall also write

$$H_n(X) := - \sum_X \mu_n(k) \cdot \log \mu_n(k).$$

The following lemma sets a very useful relation between the size and probability of a subset of the effective set on the one hand, and the entropy of  $\mathcal{S}$  on that set.

**Lemma 5.1.** *Let  $D \subset E_n$  be a subset of the effective set and*

$$c := |D|/m(n), \quad M := \sum_{k \in D} \mu'_n(k).$$

*Let  $H_n(D) := - \sum_{k \in D} \mu'_n(k) \log(\mu'_n(k))$ . Then*

$$H_n(D) \leq M (\log m(n) - \log(M/c)).$$

*Proof.* We apply Jensen's inequalities to the concave function  $-x \log x$  on  $(0, 1)$ .

$$- \frac{\sum_{k \in D} \mu'_n(k) \log(\mu'_n(k))}{cm(n)} \leq - \frac{M}{cm(n)} \log \left( \frac{M}{cm(n)} \right),$$

so  $H_n(D) \leq -M \log \left( \frac{M}{cm(n)} \right) = M (\log m(n) - \log(M/c))$ . □

In general, if  $\mathcal{C} \subset \mathcal{K}_n$  is a subset on which  $\mu_n(k) = p$  is constant, then its entropy is

$$H_n(\mathcal{C}) = -|\mathcal{C}|p \log p. \tag{9}$$

One may expect that sources with  $\lim_{n \rightarrow \infty} H_n(\mathcal{S})/H_n(\mathcal{S}_u) = 1$  have a controlled bias. This notion is formalized by:

**Definition 5.2.** *Let  $\mathcal{S}$  be a source for the key space  $\mathcal{K}$ , such that:*

$$\forall \varepsilon > 0, \exists n_0 \in \mathbb{N}: \quad H_n(\mathcal{S}) > H_n(\mathcal{S}_u) \cdot (1 - \varepsilon), \quad \forall n > n_0. \tag{10}$$

*If (10) holds, we call the bias of  $\mathcal{S}$  **entropy bounded**.*

In the following we investigate the relation between entropy bounded and polynomially bounded biases.

**Proposition 5.3.** *Polynomially bounded biases are entropy bounded.*

*Proof.* Let  $\mathcal{S}$  be a source with polynomial bounded bias and  $\varepsilon > 0$ . The relation (6) implies  $\sum_{A(\varepsilon) \cup B(\varepsilon)} \mu_n(k) < \delta$  and  $|A(\varepsilon) \cup B(\varepsilon)| < \delta m(n)$ . We shall estimate from below the entropy of  $\mathcal{S}$  on the set  $\mathcal{C}(\varepsilon)$ . We are expecting that  $\mathcal{C}(\varepsilon)$  generates the bulk of  $H_n(\mathcal{S})$  so it will be sufficient to show that (10) holds with  $H_n(\mathcal{C}(\varepsilon))$  replacing  $H_n(\mathcal{S})$ . Let  $\varepsilon, \varepsilon' > 0$  be such that

$$(1 - 2\varepsilon')H_n(\mathcal{S}_u) + \log(1 - \delta) > (1 - \varepsilon)H_n(\mathcal{S}_u)$$

and  $n > n_0(\varepsilon')$ . Note that  $1/2 > \mu_n(k) \geq \ell_n m(n)^{-\varepsilon'}$  for  $k \in \mathcal{C}(\varepsilon')$  – which implies local convexity of an auxiliary function we shall use.

We also let  $M := \mathcal{P}(\mathcal{C}(\varepsilon'))$ , so  $M > 1 - \delta$ , by (6), and  $t_n(k) := 1/M \mu_n(k)$ , a set of weights adding up to 1. We apply Jensen's inequality this time to the convex function  $-\log x$ :

$$H_n(\mathcal{C}(\varepsilon')) = M \sum_{C(\varepsilon')} t_n(k) \cdot (-\log \mu_n(k)) \geq -\log \left( \frac{\sum_{C(\varepsilon')} \mu_n(k)^2}{M} \right).$$

Using the bound  $|\mathcal{C}(\varepsilon')| < m(n)$ , we have

$$\sum_{C(\varepsilon')} \mu_n(k)^2 < |\mathcal{C}(\varepsilon')| m(n)^{2\varepsilon'} \ell_n^2 < m(n)^{-(1-2\varepsilon')}.$$

It follows that  $H_n(\mathcal{C}(\varepsilon')) > (1 - 2\varepsilon')H_n(\mathcal{S}_u) + \log(1 - \delta)$  and a fortiori

$$H_n(\mathcal{S}) \geq H_n(\mathcal{C}(\varepsilon')) > (1 - 2\varepsilon)H_n(\mathcal{S}_u) + \log(1 - \delta) > (1 - \varepsilon)H_n(\mathcal{S}_u).$$

This completes the proof.  $\square$

This proposition indicates a connection between (10) and previous criteria for bias bounding. We show that (10) is in fact equivalent to conditional polynomial bounds.

**Theorem 5.4.** *A source  $\mathcal{S}$  verifies condition (10) iff it is polynomially dense and has conditionally polynomial bounded bias.*

*Proof.* Let  $\mathcal{S}$  be a source with effective space  $E_n$  and polynomially bounded bias on  $E_n$ . Let  $\varepsilon > 0$  and  $\varepsilon' > 0$  be such that  $(1 - \varepsilon')^2 > 1 - \varepsilon$ . By applying the previous proposition, for  $n > n_0(\varepsilon')$ ,

$$\begin{aligned} H_n(\mathcal{S}) &> (1 - \varepsilon')H_n(\mathcal{S}_u/E_n) = (1 - \varepsilon') \log(e(n)m(n)) \\ &> (1 - \varepsilon') \log \left( m(n)^{1-\varepsilon'} \right) = (1 - \varepsilon')^2 H_n(\mathcal{S}_u). \end{aligned}$$

Finally, by choice of  $\varepsilon'$ , it follows that (10) holds. This shows that dense sources with conditionally polynomially bounded biases are entropy bounded.

We now show that the converse also holds. Suppose that  $\mathcal{S}$  verifies (10) for some  $\varepsilon > 0$ . We show first that  $\mathcal{S}$  is polynomially dense. We have  $H_n(\mathcal{S}) =$

$H_n(B(\varepsilon)) + H_n(A(\varepsilon) \cup C(\varepsilon))$ . We shall show that  $|A(\varepsilon) \cup C(\varepsilon)| > m(n)^{1-\varepsilon}$  which implies a fortiori that  $\mathcal{S}$  is polynomially dense. Note that

$$\mathcal{P}(B(\varepsilon)) \leq m(n)^{-\varepsilon} \ell_n |B(\varepsilon)| < m(n)^{-\varepsilon}.$$

Since by lemma 5.1,  $H_n(B(\varepsilon)) \leq \mathcal{P}(B(\varepsilon)) \log(m(n)) < m(n)^{-\varepsilon} H_n(\mathcal{S}_u)$ , it follows that  $H_n(A(\varepsilon) \cup C(\varepsilon)) > (1 - \varepsilon) H_n(\mathcal{S}_u)$ . But

$$H_n(A(\varepsilon) \cup C(\varepsilon)) \leq (1 - \mathcal{P}(B(\varepsilon))) \cdot \log(m(n) - |B(\varepsilon)|) < \log(m(n) - |B(\varepsilon)|).$$

Together, the last two inequalities yield:  $\log(m(n) - |B(\varepsilon)|) > \log(m(n))^{1-\varepsilon}$ , showing, as announced, that

$$\frac{|C(\varepsilon) \cup A(\varepsilon)|}{m(n)} > m(n)^{-\varepsilon}. \quad (11)$$

We shall assume without restriction of generality, that  $\mu_n(k) \geq m(n)^{-\varepsilon} \ell_n$ , for  $n$  sufficiently large, while  $\mathcal{S}$  verifies (10). We still have to prove that

$$M := \mathcal{P}(A(\varepsilon)) < \delta/2.$$

Suppose this is not the case and there is a constant  $d \geq \delta/2 > 0$  so that  $\forall n_1 > 0, \exists n > n_1 : M \geq d$  (note that  $M$  depends on  $n$ ). Let  $c$  be the fraction of the keys  $k \in \mathcal{K}_n$  with  $\lambda_n(k) > m(n)^\varepsilon$ . Thus  $cm(n) = |A(\varepsilon)|$ . Since  $1 > \mathcal{P}(A(\varepsilon)) > m(n)^\varepsilon \ell_n cm(n)$  it follows that  $c < m(n)^{-\varepsilon}$ . By using again lemma 5.1,

$$\begin{aligned} H_n(\mathcal{S}) &\leq M \log(cm(n)) + (1 - M) \log(m(n)(1 - c)) \\ &< \log(m(n)) + M \left( \log \left( \frac{c}{1 - c} \right) \right). \end{aligned}$$

Since  $\log(x/(1 - x))$  is increasing on the interval  $(0, 1)$ , we have  $\log \left( \frac{c}{1 - c} \right) < -\varepsilon \log(m(n))$  and

$$H_n(\mathcal{S}) < \log(m(n)) (1 - \varepsilon M) < \log(m(n))(1 - d\varepsilon).$$

For  $n_1 > n_0(\varepsilon d/2)$ , we have a contradiction to (10), which completes the proof.  $\square$

**Corollary 5.5.** *The bias of a source  $\mathcal{S}$  is conditionally tolerable iff it is entropy bounded.*

*Proof.* This is a consequence of corollary 1 and theorem 2.  $\square$

## 6. Practical Applications

In the paper [4], Brandt and Darmgård investigate the distribution of primes produced by sequential search starting from a randomly distributed start point. In [11] we investigate the distribution of a source for prime numbers which searches sequentially in an arithmetic progression with relatively large ratio. Both distributions have in common a bias due to the sequential search. It stems from the fact that the probability that the source outputs a given prime is directly proportional to the length of the gap between that prime  $p$  and its predecessor  $q$ . The larger the

gap length  $p - q$ , the higher the probability that the starting point will hit inside the interval  $(q, p)$ . It is shown in both cases, based upon a classical conjecture of Hardy and Littlewood, that the resulting distribution is closely approximated by a Poisson distribution. The two biases were studied in the papers [4], [11] by using the entropy approach.

The search in arithmetic progressions generates only a polynomially dense subset of all primes with fixed length. In fact, depending on the ratio of the progression, certain primes of given length will not be produced at all. This feature is shared, for different reasons, by the *Gordon strong primes*. Gordon strong primes  $p$  [7] are defined by the requirements that there are two primes  $q$  and  $r$  of size  $k$  bits, such that

$$q|p - 1 \quad \text{and} \quad r|p + 1, \quad (12)$$

a condition which was useful for security, prior to the discovery of the elliptic curve factoring algorithm [9]. An additional condition on  $q - 1$  having a large prime factor of given size was imposed too. We shall drop this condition for simplicity, as it is not relevant to our analysis. The behaviour of the different sources of primes is summarised in:

**Theorem 6.1.** *The biases of the incremental search source  $\mathcal{S}_i$  and of the source  $\mathcal{S}_p$  for search in arithmetic progressions are entropy bounded. A uniformly distributed source  $\mathcal{S}_g$  for Gordon strong primes (that is, a source producing uniformly distributed primes subject to the above conditions) verifies:*

$$H_n(\mathcal{S}_u) - H_n(\mathcal{S}_g) < 2 \log k \quad (13)$$

*uniformly for all  $n > 2k$ .*

*Proof.* The first statement is [4], theorem 3. The second is proved in [11], §4. Note that the statement holds also for procedures of prime generation similar to the one in [11], such as Shawe-Taylor's algorithm [16]. Let  $G_n$  be the set of all  $n$  bit primes satisfying condition (12) for some uniformly distributed  $q, r$ . The statement (13) follows from  $H_n(\mathcal{S}_g) = \log(|G_n|)$ .  $\square$

The bias for Gordon primes is asymptotically tolerable. For sizes of cryptographic primes it is however comparable to the bias of  $\mathcal{S}_p$ . Note that  $\mathcal{S}_i$  is dense, while  $\mathcal{S}_p$  is only polynomially dense, which suggests a lower entropy for the second source. The source  $\mathcal{S}_g$  is also dense, but has a hidden subspace which is relatively large for common values of  $n$ . Its bias is thus more important for smaller size of primes generated.

In [10], Maurer gives an algorithm for recursive generation of provable primes, which uses the Dickman function in order to *approximate* the uniform distribution of the primes produced. This can be done, as suggested by Bach [10], with the help of a source of random bits whose distribution is related to the Dickman function. Under this condition, the uniform distribution can be approximated asymptotically with infinite accuracy. Based on several heuristic assumptions concerning the

behaviour of the factorization of  $p - 1$  for random primes  $p$ , the entropy of the source  $\mathcal{S}_m$  for primes generated by Maurer's algorithm has thus  $H_n(\mathcal{S}_n) = H_n(\mathcal{S}_u)$ .

In practice, random sources are most often simulated by some computational process and it is tempting to wish to bound biases of computational random sources by using the concepts described here. It is however inherent to the problem setting, that such bounds can only be given based on some *model* of the computational bias and not upon statistical data of the same. Indeed, assuming that all actual computations which can be done are polynomial in the size of input, it is hardly conceivable to achieve a statistical estimation of the entropy of a computational random source beyond such a bound.

We give an example of a biased source for elliptic curves, which is not conditionally tolerable. Let  $p$  be a given prime and consider sources which produce elliptic curves  $\mathcal{E}(a, b) \pmod{p}$ , together with  $N = \#\mathcal{E}$ . The uniform distributed source  $\mathcal{S}$  will produce uniformly distributed parameters  $a, b$  and then compute  $N$  with some recent variant of the Schoof algorithm, see e.g. [13]. The complex multiplication source  $\mathcal{S}_{CM}$  chooses a complex multiplication field  $\mathbb{K}$  from a list of fields with polynomial discriminant and easily finds  $N = p + 1 \pm \mathbf{Tr}(\pi)$ , with  $\mathbf{Tr}(\pi) = \pi + \bar{\pi}$ , provided that  $p$  is a norm in  $\mathbb{K} : p = \pi \cdot \bar{\pi}$ . If  $\mathcal{R}$  is the space of curves produced by  $\mathcal{S}_{CM}$  and  $\mathcal{K}$  the space of elliptic curves over  $\mathbb{F}_p$ , then  $\frac{\#\mathcal{R}}{\#\mathcal{K}} = \frac{\mathcal{O}(g(\log p)^2)}{\mathcal{O}(p^2)}$ , where  $g$  is the polynomial bounding the discriminants of the complex multiplication fields. If the two sources are compared with respect to the full key space  $\mathcal{K}$ , it is obvious that  $\mathcal{S}_{CM}$  is far from being *dense* in this space, and its bias is thus not conditionally tolerable in our terminology. Although no EL algorithm dedicated for curves with CM in fields with small discriminant is known or made plausible, the source  $\mathcal{S}_{CM}$  is obviously incompatible with our general notion of tolerable biases.

## 7. Conclusions

We have given a model for measuring biases of non uniformly distributed key sources. The model provides the means for upper bounding the run time gain of an algorithm which makes use of the bias for breaking a scheme faster than by use of the state of the art algorithm in presence of uniform distributed keys. This is done by comparing the average case complexity of *any* algorithm with respect to the two distributions: uniform and biased. The two average complexities are related by functions depending only on the bias. Since there is no implicit assumption about the algorithms involved, the model gives an universal tool for evaluating upper bounds of run time gains which can be expected in presence of some biased sources. If these bounds are considered as irrelevant from the point of view of complexity theory, a sound proof of the security of respective biased sources results.

We have also shown the connection to entropy measure of biases and proved that some sources of prime numbers considered by earlier papers are tolerable in the sense defined here.

**Acknowledgments:** I thank the anonymous referee for pointing out the connection to the theory of average case complexity and providing valuable references and I thank him and I. Shparlinski for the encouragement to develop the ideas exposed in this paper. Thanks go also to U. Maurer for valuable discussions about the general framework for the exposition of the ideas in this paper. I am grateful to R. Silverman for his careful comments and suggestions.

## References

- [1] E. Bach: *Realistic analysis of some randomized algorithms*, J. Comput. Sys. Sci. **42** (1992), pp. 30-53.
- [2] E. Bach: *Explicit bounds for primality testing and related problems*, Math. Comp. **55** (1990) pp. 355-380.
- [3] S. Ben-David, B. Chor, O. Goldreich and M. Luby: *On the Theory of Average Case Complexity*, J. of Computer and System Sciences, bf 44, (1992), pp. 193-219.
- [4] J. Brandt and I. Damgård: *On generation of probable primes by incremental search*, Proceedings CRYPTO92, Lecture Notes in Computer Science, **740**, (1992), pp. 358-370.
- [5] Y. Gurevich: *Average Case Complexity*, J. of Computer and System Sciences, **42**, (1991), pp. 346-98.
- [6] Y. Gurevich: *Matrix decomposition problem is complete for the average case*, Proceedings 31-st IEEE Symp. on Foundations of Computer Science, (1990), pp. 802-811.
- [7] J. Gordon: *Strong primes are easy to find*, Advances in Cryptology - EUROCRYPT '84, Lecture Notes in Computer Science, vol. **209**, (1984), pp. 216-223.
- [8] L. Levin: *Average Case Complete Problems*, SIAM J. on Computing, **15** (1986), pp. 285-6.
- [9] H.W. Lenstra Jr.: *Factoring Integers With Elliptic Curves*, Annals of Mathematics, Vol. **126**, (1987), pp. 649-673.
- [10] U. Maurer: *Fast generation of prime numbers and secure public-key cryptographic parameters*, Journal of Cryptology, **8**, Nr. **3**, (1995), pp. 123-155
- [11] P. Mihăilescu: *Fast generation of provable primes using search in arithmetic progressions*, Proceedings CRYPTO94, Lecture Notes in Computer Science, **839**, (1994), pp. 282-293.
- [12] R. Peralta and V. Shoup: *Primality testing with fewer random bits*, Computational Complexity, **3** (1993), pp. 355-367.
- [13] R. Schoof: *Counting points on elliptic curves over finite fields*, J. de Théorie des Nombres, Bordeaux, **7**, (1995), 219-254.

- [14] V. Shoup: *Removing randomness from computational number theory*, PhD Thesis, University of Wisconsin - Madison (1989).
- [15] N. Smart: *The Discrete Logarithm Problem on Elliptic Curves of Trace One*. Journal of Cryptology **12 (3)**: pp. 193-196 (1999)
- [16] J. Shawe - Taylor: *Generating strong primes*, Electronics Letters, **Vol. 22**, No. 16, (1986), pp. 875-77.

Institut für wissenschaftliches Rechnen, ETH, 8090 Zürich  
*E-mail address*: `mihailcs@inf.ethz.ch`