Unifom Generators and Combinatorial Designs

Alexis Bonnecaze
Université de la méditerrané,
IML, ERISCS
Marseille, France
Email: bonnecaze@univmed.fr

Pierre Liardet
Université de Provence,
LATP
Marseille, France
Email: liardet@cmi.univ-mrs.fr

***Abstract*—The concept of randomness is fundamental in many domains and in particular in cryptography. Intuitively, a system, which is unpredictable is more difficult to attack and as a consequence, creating sequences that look like random represents a major issue. In this paper, we first study theoretically how a source of symbols with positive entropy can be turned into a true random generator called Bernoulli. We concentrate on a special type of generators, which consists in randomly choosing k elements out of n elements. After studying some existing algorithms, which are of Las Vegas type, we introduce new constructions from a binary generator taken as a primary random source of symbols. Our method is based on combinatorial block designs and we construct algorithms of Monte Carlo type involving random walks. We analyze in detail properties of our general method. Several explicit constructions of k-out-of-n generators are given. We show that the speed of convergence to the uniform distribution is better than any known method using algorithms with bounded running times.**

***Keywords*-Random Generator; Design; k-out-of-n Algorithms; Markov Chain; Random Algorithms;**

I. Introduction

Random or pseudo-random generators of numbers are omnipresent in cryptography. The concept of randomness is used for various purposes. Salt and nonce are well known examples of random values. A nonce (number used once) is used to check the freshness of a message or as an initialization vector. In conjunction with password, salt is frequently used in order to complicate a dictionary attack. Many cryptographic primitives also require random or pseudorandom inputs like keys or values to make algorithms probabilistic. It is well known that digital signatures or challenges in authentication protocols require the use of random quantities. For these reasons, finding of good pseudo-random generator is a stake in first importance. There exist in the literature lots of pseudo-random generators, which imitate in some sense a sequence of independent random variables X_n, uniformly distributed like, for example, the Blum Blum Shub generator (BBS) [5].

However, some applications require more complex generators, called k-out-of-n generators. They consist in picking randomly k elements in a set of n elements. The need for such generators is multifarious. They help to reach load balancing in certain distributed systems like high-availability clusters for example. They are also useful in security protocols, like threshold signatures [27] or time-stamping schemes [6], [7]. Suppose we decide to create a service of authentication (signatures and time-stamps). Most of protocols use the concept of trusted third party even though it may be difficult to build a third party server that can be trusted. Indeed a server may be corrupted or victim of Denial of Service attacks (DoS). Moreover, the problem may not have a malicious origin but a hardware or software one. An important requirement of existing protocols is to prevent the server from failing. In fact, schemes relying on a unique third party server cannot be fully trusted. Therefore, such a scheme should use a multi-server architecture that could be described as follows: the protocol uses n third party servers. For each request to the system, k servers out of the n servers are randomly chosen to process the request. These k servers are said to be the active servers. In this configuration, an attacker does not know a priori what are the active servers for a given request. The attack is then much more difficult to operate. Moreover, the randomness of the generator allows the system to be load balancing.

The above example is at the origin of this work as we noticed that the construction of a uniform k-out-of-n generator from a unbiased (or not) 0-1 valued Bernoulli generator were not so much studied in the literature where uniform random number generators in the unit interval are commonly used. The underlying general problem is in fact to construct unbiased Bernoulli generators from a binary Bernoulli one, supporting a possible bias.

The paper is organized as follows. Section II recalls necessary background, regarding (true) random generators from a theoretical point of view. To this aim, basic properties of symbolic Bernoulli dynamical systems are given. They serve as theoretical models of random sources of symbols with positive entropies. Section III reviews some existing solutions of the k-out-of-n problem. Both first ones are very simple and of classical conception, while the third one called RANKSB algorithm in [17] is due to Nijenhuis and Wilf. This former algorithm leads to important bias in comparison to the uniform distribution. Most of them are Las Vegas algorithms. Section IV is devoted to our constructions that are supported by Monte Carlo algorithms, hence with

a bounded running times. Such algorithms approach the uniform distribution exponentially fast. We first propose a generator based on the existence of some special combinatorial objects, namely block t-designs or including some Steiner systems. This construction generates k elements from a set of n elements in a uniform fashion from a binary generator. It consists in randomly picking a block, called word, from the blocks of the design. This word is then modified in order to obtain a vector of weight k and length n with the desired property. We will see that it can be useful to introduce the notion of block codes, since codewords of a fixed Hamming weight in some codes hold a design.

The second type of constructions is based on random walks on a finite set following the action of a finite number of generators of a group acting transitively. The first construction uses the permutation group $\mathfrak{S}_n$. The second construction is very related to the method based on block designs. The main difference is in the way of randomly picking a word in the appropriate set. The algorithm makes use of the automorphism group of the design and executes a random walk on its blocks.

This article is an extended version of [1]. We present in detail our methods, including mathematical foundations. Compare to the conference version, our construction makes use of block designs and not just Steiner systems. Indeed, block designs are widespread and, as a consequence, our construction can be applied for a wide range of parameters k and n. In order to illustrate our methods, we give several explicit constructions. For each chosen couple of parameters k and n, we exhibit a correct design and calculate the speed of convergence of the generator.

II. Randomness and mathematical foundations

A. From a binary random number generator to a q-ary one

In [18], the NIST defines a random bit sequence as follows. “A random bit sequence could be interpreted as the result of the flips of an unbiased *fair* coin with sides that are labeled “0" and “1," with each flip having a probability of exactly $1/2$ of producing a “0" or “1." Furthermore, the flips are independent of each other: the result of any previous coin flip does not affect future coin flips". Similarly, replacing the set of issues $\{0,1\}$ by a given finite set $\mathcal{A}$, one can define a random $\mathcal{A}$-valued sequence as an $\mathcal{A}$-valued independent and identically distributed random variables $X_n(\cdot)$ with common law the uniform distribution on $\mathcal{A}$. For our purpose, it is convenient to translate these notions in term of Bernoulli dynamical systems. To this aim, we recall basic definitions and general results related to symbolic dynamical systems and entropy.

Assume that $\mathcal{A}$, also called alphabet, is equipped with the discrete topology, has q elements (or letters) with $q \geq 2$, and set $\Omega(\mathcal{A}) = \mathcal{A}^{\mathbb{Z}}$ the product space endowed with the usual compact product topology. Elements ω of $\Omega(A)$ are infinite bilateral sequences

$$\omega = (\dots, \omega_{-3}, \omega_{-2}, \omega_{-1}; \omega_0, \omega_1, \omega_2, \dots)$$

with origin pointed at ω_0. The alphabet $\mathcal{A}$ is usually endowed with the uniform distribution denoted by $U(\mathcal{A})$ but we also consider other distributions. The space $\Omega(\mathcal{A})$ is equipped with the σ-algebra of its Borel subsets. Any probability μ on $\mathcal{A}$, induces the infinite product probability μ^∞ on $\Omega(\mathcal{A})$, which is defined from cylinder sets. More precisely, for any $a := a_0 \dots a_{n-1}$ in A^n, let

$$[a] := \{\omega \in \Omega\,;\ \forall\, i \in \{0, \dots, n-1\},\ \omega_i = a_i\}$$

be the cylinder set of base a, then

$$\mu^\infty(\sigma^k[a]) = \mu(\{a_0\}) \dots \mu(\{a_{n-1}\})$$

for any $k \in \mathbb{Z}$. The shift $\sigma : \Omega(\mathcal{A}) \to \Omega(\mathcal{A})$ is defined by $\sigma(\omega)_n = \omega_{n+1}$. Now, the triplet $B(\mathcal{A}, \mu) := (\Omega(\mathcal{A}), \sigma, \mu^\infty)$ is by definition the Bernoulli (random generator) on $\mathcal{A}$ with source distribution μ. In case $\mu = U(\mathcal{A})$ we set $U^\infty(\mathcal{A})$ for μ^∞ and $B(\mathcal{A}, U(\mathcal{A}))$ is simply denoted by $B(\mathcal{A})$. Let $\pi_0 : \Omega(\mathcal{A}) \to \mathcal{A}$ be the central projection ($\pi_0(\omega) = \omega_0$), then the maps $B(\mathcal{A})_n := \pi_0 \circ \sigma^n$ defined on the probability space $(\Omega(\mathcal{A}), U^\infty(\mathcal{A}))$ form an $\mathcal{A}$-valued sequence of independent random variables identically distributed with distribution law $U(\mathcal{A})$. This corresponds to a q-ary random number generator for $q = \#\mathcal{A}$.

B. Symbolic random sources, entropy and factors

To fix some notations and for convenience of the reader, we recall basic definitions and facts from ergodic theory. For more details and proofs we refer to the monographs [32] and [25], and specific references below. Our mathematical model of random source of letters in $\mathcal{A}$ should be identified to a symbolic dynamical system (SDS) with symbols in $\mathcal{A}$, that is to say a triple $(\Omega(\mathcal{A}), \sigma, \nu)$ where ν is a Borel measure on $\Omega(\mathcal{A})$, which is σ-invariant, *i.e.,* $\nu = \nu \circ \sigma^{-1}$. The entropy of such a system is, by the classical Kolmogorov-Sinai Theorem or by definition,

$$H(\sigma, \nu) := -\lim_N \frac{1}{N} \sum_{a \in \mathcal{A}^N} \nu([a]) \log \nu([a])$$

with the convention $0. \log(0) = 0$.

Let $\Omega' := (\Omega(\mathcal{A}'), \sigma', \nu')$ and $\Omega := (\Omega(\mathcal{A}), \sigma, \nu)$ be symbolic dynamical systems. Their direct product is the SDS $\Omega \times \Omega' = (\Omega(\mathcal{A} \times \mathcal{A}'), \sigma \times \sigma', \nu \otimes \nu')$ and so $H(\sigma \times \sigma', \nu \times \nu') = H(\sigma, \nu) + H(\sigma', \nu')$. This construction means that the source corresponding to Ω and Ω' are independent. If there is a measure preserving map $f : \Omega' \to \Omega$ commuting with the shifts, *i.e.,*

$$\begin{cases} \nu = \nu' \circ f^{-1}\,, \\ \sigma \circ f = f \circ \sigma' \ (\nu\text{-almost everywhere})\,, \end{cases} \tag{1}$$

then Ω is said to be a factor of Ω' with factor map f. In that case $H(\sigma, \nu) \leq H(\sigma', \nu')$. In the following, the value

$\frac{H(\sigma,\nu)}{H(\sigma',\nu')}$ will be called entropy rate and denoted by $\tau(\nu,\nu')$ or simply by τ. If f is invertible (up to negligible sets), with measure preserving inverse map, then Ω is said to be isomorphic to Ω' with conjugate map f and if Ω' is a Bernoulli random generator, then by extension Ω is also called a Bernoulli SDS.

We have $H(\sigma^{-1},\nu) = H(\sigma,\nu)$ and more generally $H(\sigma^{\pm k},\mu) = kH(\sigma,\nu)$ for any natural number k. In fact the k-th iterate $(\Omega(\mathcal{A}),\sigma^k,\nu)$, $k \neq 0$, is canonically identified with $(\Omega(\mathcal{A}^k),\sigma^{(k)},\nu^{(k)})$, where $\sigma^{(k)}$ is the shift on $\Omega(\mathcal{A}^k)$ and $\nu^{(k)}$ is induced by ν restricted to cylinder sets of $\Omega(\mathcal{A}^k)$ viewed as particular cylinder sets from $\Omega(\mathcal{A})$. Therefore $H(\sigma^{(k)},\nu^{(k)}) = kH(\sigma,\nu)$.

This construction has the following important consequence. Assume that we have got a binary source with positive entropy h, for example a source extracting from random jitter in an electrical circuit or quantum effects in semiconductors or timing of running current process, or a combination of these sources. Then, we derive a source of binary blocks of length k having entropy kh.

In case of a Bernoulli system $B(A,\mu)$ (hence $\nu = \mu^\infty$ with the above definition), its entropy is easy to compute:

$$H(\sigma,\mu^\infty) = -\sum_{a\in\mathcal{A}} \lambda(\{a\})\log\lambda(\{a\}).$$

In particular $H(\sigma,U^\infty(\mathcal{A})) = \log(\#\mathcal{A})$.

As a consequence of a deep result of Y. Sinai (see [28]), if the entropy $H(\sigma',\nu')$ of $\Omega' = (\Omega(A'),\sigma',\mu')$ is greater than or equal to $\log\#A$ then there exists a factor f from $(\Omega(A'),\sigma',\mu')$ onto the uniform Bernoulli generator $B(A)$. Properties (1) shows that f is determined by the central coordinate map $f_0 = \pi_0 \circ f$ since we have $f = (\dots, f_{-2}, f_{-1}; f_0, f_1, f_2, \dots)$ with $f_k = f_0 \circ \sigma^k$ ($k \in \mathbb{Z}$). In particular f_0 is constant equal to a on $C_a := f^{-1}([a])$. Moreover, all partitions $\{\sigma'^m(C_a);\ a \in A\}$ ($m \in \mathbb{Z}$) are independent in between. Hence, building such a partition is usually intractable by computer except in particular cases pointed out below. Another consequence of the above construction and Sinai's theorem is that, from any given random binary source of positive entropy, theoretically there exists a factor built from of a suitable power of this source, which is $B(\{0,1\})$, the factor map consisting in distributing binary sequences in two parts equally likely and independently in the time. This is the most hard problem to be solved in practice for constructing, from a suitable physical random source, a binary random generator according to the NIST definition.

Following results of D. S. Ornstein [19], we recall that the family of Bernoulli dynamical systems is remarkably stable. In particular, they are characterized by their entropy (two such systems of equal entropy are isomorphic), any direct product of Bernoulli systems and any non trivial iterate (and also any root) of a Bernoulli system are Bernoulli. Moreover any factor of a Bernoulli system is also Bernoulli. These properties imply that any probability algorithm that takes in input a Bernoulli source and output a random source of symbols always gives rise to a Bernoulli SDS, isomorphic to some $B(\mathcal{A},\mu)$. In this paper we propose algorithms that take as inputs the outcomes of an appropriated Bernoulli source $B(\{0,1\}^k)$ and output a random source of letters in a given alphabet $\mathcal{A}$ whose distribution of letters is exactly or approximate accurately the uniform distribution. We may distinguish three sorts of such random algorithms.

(A1) Algorithms that output the uniform distribution on $\mathcal{A}$ in a bounded running time.

(A2) Las Vegas algorithms: they output the uniform distribution on letters with unbounded running time but with a finite expectation.

(A3) Monte Carlo algorithms: they end in a bounded running time, output a distribution usually distinct to the uniform distribution but arbitrarily closed to it in term of total variation.

The following theorem depicts the first case.

Theorem 1: If an algorithm of type (A1) takes input from issues of the source $B(\{0,1\}^k)$ and produce a uniform Bernoulli source of entropy $\log_2 q$, then $q = 2^s$ with $s \leq k$.

Proof: By assumption, the algorithm can be identified to a factor maps f with central coordinate map f_0 having its values in $\{0,1,\dots,q-1\}$. That leads to the partition $\{f_0^{-1}(\{j\});\ 0 \leq j < q\}$ of $\Omega(\{0,1\}^k)$ with $U(\mathcal{A}^k)(f_0^{-1}(\{j\})) = 1/q$ and there exists an integer $L \geq 1$ such that each $f_0^{-1}(\{j\})$ is the union of some cylinder sets of the form $C_L(a) := \{\omega \in \Omega(\{0,1\}^k);\ (\forall i)(|i| > L \text{ or } \omega_i = a_{i+L})\}$, $a_0\cdots a_{2L} \in (\{0,1\}^k)^{2L+1}$. This implies that q divides 2^{kL} and since $\log q \leq k$ one has $q = 2^s$ with $s \leq k$. ■

Obviously, having in hands a uniform binary source, like tossing an unbiased coin, for cryptographic applications is impractical. But such an abstract uniform Bernoulli generator of binary sequences serves as a benchmark for evaluation of random generators and pseudo-random generators. In fact, security of most cryptographic algorithms and protocols using pseudo-random generators is based on the assumption that it is infeasible to distinguish use of a suitable binary pseudo-random number generator (PRNG) from use of a (truly) random number generator (RNG) defined as the SDS $B(\{0,1\})$. As an example, the pseudorandom generator BBS has been proven secure in the sense that an attacker cannot predict, in a reasonable time, the next bit of the outcome with a probability greater than $1/2$ (see [5]).

Putting apart the independency, the first major problem is then to construct generators G_n of elements (called states or symbols) of a finite set $\mathcal{A}$, such that the distribution law P_n of G_n converges to the uniform distribution $U(\mathcal{A})$ on $\mathcal{A}$ as n tends to infinity. In order to quantify this convergence, we use the total variation distance between P_n and $U(\mathcal{A})$.

This distance is defined by

$$d(P_n, U(\mathcal{A})) = \frac{1}{2} \sum_{a \in \mathcal{A}} \left| P_n(a) - \frac{1}{\#\mathcal{A}} \right|$$

where $P_n(a)$ is the probability that the generator G_n outcomes the state a.

A classical method to solve this problem is to introduce a transitive and irreducible Markov chain of transition matrix T, with space of states $\mathcal{A}$, such that the uniform distribution on $\mathcal{A}$ is the stationary distribution of the chain. Constraints of the problem are essentially on the incidence matrix of the chain since each state a can only transit on a number $\tau(a)$ of states such that $0 < \tau(a) \leq \tau_{\max}$ where $\tau_{\max}$ is a small constant compare to $\#\mathcal{A}$. The stationary distribution is then approached by considering Markov random walk on a finite graph. In fact, the general theory of finite Markov chains shows that (by Perron-Frobenius's theorem) there exist two constants $C > 0$ and $\rho \in]0,1[$ such that for every pair of states (i, j), one has

$$\left| (T^n)_{ij} - \frac{1}{\#\mathcal{A}} \right| \leq C\rho^n \,. \tag{2}$$

When $\#\mathcal{A}$ is big, computation of constants C and ρ satisfying (2) is generally not effective, even if we assume that n is large enough. In fact, if m_0 is an integer such that for a constant $c \in]0,1]$ we have $(T^{m_0})_{ij} \geq c\frac{1}{\#\mathcal{A}}$ for all pairs of states (i, j), then inequality (2) becomes

$$\left| (T^n)_{ij} - \frac{1}{\#\mathcal{A}} \right| \leq (1-c)^{\lfloor n/m_0 \rfloor} \,. \tag{3}$$

More details concerning finite Markov chains can be found in [26] or [13]. Random walks on groups or finite graphs is treated in [23], [10] and a survey on recent results on the subject can be found in [24].

III. Existing algorithms

Construction of a k-out-of-n generator greatly depends on the requirements of the applications. They could involve the level of security, the amount of resources (CPU, memory, etc.) needed or the generators used as a primary source of randomness. In this former case, our reference, the generator BBS, corresponds to the abstract model $B(\{0,1\})$. Given the set $\mathcal{P}_k^n := \{F \subset E;\ \#F = k\}$ endowed with the uniform distribution, the ultimate goal is then to construct from $B(\{0,1\})$ a sequence of independent random variables X_m of distribution P_m such that $d(P_m, U(\mathcal{P}_k^n)) \leq Ce^{-cm}$ for $m \geq m_0$, where C, c and m_0 are explicit constants that can be used in practice. Now we review some standard k-out-of-n generators according to above classification $A1$–$A3$.

(1) An algorithm of type (A1) exists if and only if $\binom{n}{k}$ is a power of 2. This implies that $n = 2^s$ and $k = 1$. In that case, the algorithm is just the identity map: the output is equal to the input given by the generator $B(\{0,1\}^s)$ or by iterating $B(\{0,1\})$ s times.

(2) The most obvious algorithm is based on the construction of a set of k elements by randomly picking an integer between 1 and n, then renew the process to obtain an other element between 1 and n but distinct from the first one and so on. It is typically a Las Vegas algorithm. From a probabilistic point of view, this process needs an average of $n\left(\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-k+1}\right)$ random runs (see [14]), an average, which is $\mathcal{O}(n)$ for $1 \leq k \leq n/2$.

(3) The probably oldest probabilistic algorithm to uniformly and randomly pick k elements among n elements relies to the Fisher-Yates shuffle algorithm for generating a random permutation σ of $\mathcal{E}_n = \{1, \ldots, n\}$ but stoping the construction as soon as the values $\sigma(1), \ldots, \sigma(k)$ are constructed. The Fisher and Yates original method consists to randomly pick an element e_1 from $\mathcal{E}_n$ with the uniform distribution then pick an element from $\mathcal{E}_n \setminus \{e_1\}$ with the uniform probability and so on, k times. The underlying probabilistic model is based on the representation of integers in factorial basis. More precisely, let $I_k := \{0, \ldots, n-1\} \times \{0, \ldots, n-2\} \times \cdots \times \{0, \ldots, n-k\}$ equipped with the uniform probability. Any element $i := (i_1, i_2, \ldots, i_k)$ of I_k corresponds univocally to the integer $n_i \in \{0, \ldots, n! - 1\}$ given by its expansion in the factorial basis:

$$n_i = i_k.(n-k)! + i_{k-1}.(n-k+1)! + \cdots + i_2.(n-2)! + i_1.(n-1)!$$

and the corresponding subset $P_i = \{p_{i_1}, \ldots, p_{i_k}\}$ of $\mathcal{E}_n$ given by $p_{i_1} = i_1 + 1$, then p_{i_2} is the $(i_2 + 1)$-th coordinate of the n-tuple M_2 deduced from the n-tuple $M_1 = (1, 2, \ldots, n-1, n)$ by exchanging the n-th coordinate with the $(i_1 + 1)$-th one. For $2 \leq s \leq k - 1$, p_{i_s} is constructed by induction as the $(i_s + 1)$-th coordinate of the n-tuple M_s deduced from M_{s-1} by exchanging the $(n - s + 2)$-th coordinate with the $(i_{s-1} + 1)$-th coordinate. Notice that there exist $k!$ integers i, which give the same subset. The major drawback of this process is that it requires to have independent uniform generators on s letters, for $n - k < s \leq n$. The given construction leads to a factor map from $B(A', \mu')$ onto $B(A, \mu)$ with $A' = I_k$ and A equal to $\mathcal{P}_k^n$. The entropic rate is thus $\frac{\log \binom{n}{k}}{\log n! - \log(n-k)!}$.

A better version of the Fisher-Yates algorithm was introduced by R. Durstenfeld in [11] and later by D. Knuth in [14] with his Algorithm P (Shuffing) on page 145. A random number generator G uniformly distributed in $[0, 1]$ is used in Algorithm P. It can be formally obtained from $B(\{0,1\})$ by computing $G(\omega) = \sum_{n=0}^{\infty} \omega_n 2^{n-1}$ ($\omega \in \Omega(\{0,1\})$). G is applied for computing a random integer $k(\omega) = 1 + \lfloor jG(\omega) \rfloor$, between 1 and j. In this way, computation of $\lfloor jG(\omega) \rfloor$ could never stop (but only for n particular issues of ω). Nevertheless, the algorithm is of type (A2). Recently, R. Rolland in [22] proposed an algorithm of Las-Vegas type, which is analogous to

Fisher-Yates algorithm. This algorithm constructs a k-out-of-n random generator involving only a uniform Bernoulli random generator $B(\{0,1\}^\ell)$ with $n \leq 2^\ell$.

(4) RANKSB algorithm proposed in [17] takes into account some algorithmic constraints (in particular in terms of CPU and execution time), which are not verified by the preceding methods. We give a simplified version of RANKSB. It consists in subdividing the interval $[1, n]$ in k sub-intervals R_j with approximately the same length, and randomly choose the number r_j of elements to be selected in each R_j. If we don't take into account that $r_j \leq \#R_j$, the k-tuple $(r_1, \ldots, r_k)$ of integers $r_j \geq 0$ such that $r_1 + \cdots + r_k = k$ follows binomial law constructed from k independent runs of integers in $\{0, \ldots, n\}$, with the uniform law. In order to avoid that r_j be greater than $\#R_j$, we recompute the subdivision in sub-intervals R_j then in each R_j we select r_j elements using method (2), (3) or any others. The algorithm uses a source of entropy $k \log n$ corresponding to $B(\{1, ..., n\}^k)$, to obtain the factor corresponding to a shift of Bernoulli on the set of k-tuples $(r_1, \ldots, r_k)$ as above (distributed according to the binomial law), which is of entropy

$$H_k := -\sum_{r_1+\cdots+r_k=k} \frac{1}{k^k} \binom{k}{r_1, \ldots, r_k} \log\Big(\frac{1}{k^k}\binom{k}{r_1, \ldots, r_k}\Big).$$

When k is small, let us use (2) to pick the r_j elements in R_j so that the corresponding entropy rate is (in average) assumed closed to 1. The entropy rate of the algorithm is then $H_k/k \log n$. Since H_k is less than $\log k^k$, this entropy rate is then less than $\log k/\log n$. Notice that if n/k is small, the output distribution has a non negligible bias.

In the sequel we describe k-out-of-n generators, which are of Monte Carlo type.

IV. Proposed algorithms

In this section, we propose two types of constructions. The first one is based on some remarkable configurations of points in binary vectors spaces, namely, block designs [15]. It leads to optimal uniform generators and exists for a wide range of parameter values k and n since designs are very common. The second type of constructions is based on random walks on some groups or finite sets.

A. Block t-design based constructions

A combinatorial block t-design D with parameters t-(v, k, λ) is an incidence structure $(\mathcal{P}, \mathcal{B})$ (where elements of $\mathcal{B}$, called blocks, are subsets of $\mathcal{P}$) satisfying the following conditions:

- $\#\mathcal{P} = v$,
- $\forall\, B \in \mathcal{B}, \#B = k$,
- $\forall\, S \subset \mathcal{P}$ such that $\#S = t,\ \#\{B \in \mathcal{B};\ S \subset B\} = \lambda$.

It is known that a necessary condition for the existence of a t-design is that

$$b_s = \lambda \frac{\binom{v-s}{t-s}}{\binom{k-s}{t-s}},\ \text{for all } s \text{ satisfying } 0 \leq s < t,$$

and where b_s corresponds to the number of blocks that contain any s-set (*i.e.,* set of s elements) of points from $\mathcal{P}$. The Web site [9], maintained and regularly updated by Dan Gordon, gives a database of known t-designs.

A Steiner system is a particular case of a block design. It is simply a block design with parameters t-$(v, k, 1)$ and is currently denoted by $\mathrm{S}(t, k, v)$. A Steiner system for $t = 2$ is called balanced incomplete block design and for $v = s^2 + s+1$, $k = s+1$, the system corresponds to the combinatorial notion of finite projective plane. A necessary condition for the existence of a Steiner system is that the number

$$\frac{\binom{v-s}{t-s}}{\binom{k-s}{t-s}}$$

is an integer for all s satisfying $0 \leq s < t$. This condition is not sufficient and there is no Steiner system with, for example, parameters $(2, 6, 36)$, $(3, 7, 37)$ or $(5, 6, 18)$. In fact, there is no known general sufficient condition on the existence of Steiner systems. For $t = 2$ and 3, there exist infinitely many families of Steiner systems and for $t \geq 4$, we refer to [8] or [31].

There exist many ways to construct a design. The easiest one is probably to consider an error correcting code. Indeed, there exist codes whose words of a fixed weight hold designs. For example, words of weight 3 in the Hamming codes yield 2-designs and 5-designs can be constructed from Golay codes. Quadratic residue codes constitute another example of family of codes that yield designs. An other way to obtain designs is to consider Hadamard matrices. In this article, examples of such constructions are given. Finally, there exist many other constructions like, for example, recursive methods [20].

For any design D, its group of automorphisms, denoted by $\mathrm{Aut}(D)$, plays a fundamental role to understand the geometrical structure of D. Elements σ of this group belong to the permutation group $\mathfrak{S}(\mathcal{P})$ of $\mathcal{P}$ and verifies the following condition

$$\mathcal{B}^\sigma = \mathcal{B}$$

where $\mathcal{B}^\sigma$ denotes the set of $\sigma(B)$ with $B \in \mathcal{B}$.

A brief description of our construction of a k-out-of-n generator from a design can be expressed as follows. First, consider a k-(n, b, λ) design $(\mathcal{P}, \mathcal{B})$ and after indexing $\mathcal{P}$ by i, $1 \leq i \leq \#\mathcal{P}$ identify each block B in $\mathcal{B}$ with a binary string of length $\#\mathcal{P}$, the place of 1's indicating the elements of B. Now, choose randomly a block m from the blocks of the design. By construction, the Hamming weight of m is b. Then, randomly choose $b - k$ coordinate positions of 1 in m, replace the corresponding 1 by 0. We

get the output m'. Since we are only concerned by the set of positions of the 1's, we may identify it to $\{1,\ldots,b\}$ so that the elimination of $b-k$ digits 1 is done by selecting at random, independently of the random choice of m, a subset of $b-k$ elements in $\{1,\ldots,b\}$, issuing m' independently of the possible λ-blocks m that cover m'. Hence, the algorithm outputs a random word of length n with Hamming weight k. Notice that finding a k-out-of-n generator is equivalent to finding a $n-k$-out-of-n generator. Of course this algorithm needs uniform random generators on sets of symbols, which are not necessarily of cardinality a power of 2, hence it is usually of type (A2).

Example 1: As an example, we explain in detail the construction of a 5-out-of-24 generator, which is based on a $S(5,8,24)$. In this case, the blocks of the design can be represented as the words of weight 8 in the extended binary Golay code $\mathcal{G}_{24}$. Our construction is based on the following property [16, page 67]: every binary vector of Hamming weight 5 and length 24 is covered by exactly one word of $\mathcal{G}_{24}$ of weight 8. It turns out that a random generator in $\mathcal{P}_5^{24}$ is easily obtained from a random generator of the words of weight 8 of the Golay code (and vice versa).

We recall the main combinatorial properties of $\mathcal{G}_{24}$. In the sequel $W(m)$ denotes the Hamming weight of a binary string m (also called vector as element of the underlying vector space); the weight distribution of $\mathcal{G}_{24}$ is classical and given by the following table:

weights	0	8	12	16	24
number of words	1	759	2576	759	1

Table 1: Weight distribution of the words of the Golay code $\mathcal{G}_{24}$

A remarkable property of this code is that the set of words of a given weight forms the blocks of a design. Hence, the words of weight 8, called octads, form the blocks of a 5-$(24,8,1)$ design and words of weights 12 form the blocks of a 5-$(24,12,48)$ design. It is worth noticing that, in the case of octads, the parameter λ of the design is equal to 1. This means that a vector of length 24 and Hamming weight 5 is covered by exactly one octad of the code. Thus, octads form a Steiner system with parameters $S(5,8,24)$. Note that there exist in $\mathcal{G}_{24}$ other Steiner systems like $S(4,7,23)$, $S(3,6,22)$ or $S(2,5,21)$, leading to similar constructions of generators.

Since $\mathcal{G}_{24}$ decode at most three errors, it happens that when changing three bits from the value 1 to the value 0 in an octad, one can construct $\binom{8}{5}$ vectors of weight five. If we repeat this process for every octad, we obtain a total of $759\times\binom{8}{5}=\binom{24}{5}$ vectors of weights five, which is the cardinal of $\mathcal{P}_8^{24}$.

This leads to the following construction:

(a) choose a base $\{b_1,\ldots,b_{12}\}$ of $\mathcal{G}_{24}$;
(b) choose a random generator G on 12 bits (corresponding to a 12 iterations of a binary Bernoulli);
(c) pick a binary vector $g:=g_1\cdots g_{12}$ from the random generator G;
(d) compute the word $m(g):=\bigoplus_{j=1}^{12} g_j b_j$;
(e) if $W(m)=8$ then $m=0^{a_1}10^{a_2}1\ldots0^{a_8}10^{a_9}$ and do
  (e1) randomly pick a vector $x=x_1\ldots x_8$ of weight three and length eight (there exist $\binom{8}{3}=56$ such vectors. They can be kept in memory);
  (e2) compute $m'':=0^{a_1}(1\oplus x_1)0^{a_2}(1\oplus x_2)\ldots0^{a_8}(1\oplus x_8)0^{a_9}$;
(f) if $W(m)=16$, then $m:=m\oplus 1^{24}$ and go to (e);
(g) if $W(m)=0$ or $W(m)=24$, then go to (c);
(h) if $W(m)=12$, then ask G a binary vector $g':=g'_1\cdots g'_{12}$ and compute $m':=m(g')$
if $W(m')=8$ then $m:=m'$ and go to (e)
else if $W(m'\oplus m)=8$ then $m:=m'\oplus m$ and go to (e)
else go to (c);
(i) output m''.

This algorithm makes use of the generator $\Gamma=B_8\times B(\{1,\ldots,56\})$ where B_8 is the generator induced by $B(\{0,1\}^{12})$ on the set P_8 of words of weight eight or sixteen. Since the counting probability of P_8 is $\mu(P_8)=\frac{2\times759}{2^{12}}=0,37\ldots$, the entropy of Γ is equal to $h(\Gamma)=(12\log 2+\log 56)/\mu(P_8)=33.30\ldots$. It gives in output the uniform generator $B(\mathcal{P}_5^{24})$ with entropy $\log\binom{24}{5}$. Hence we obtain the entropy rate

$$\frac{\log\binom{24}{5}}{h(\Gamma)}=0,319\ldots.$$

In this example, we obtained the blocks of the design by considering the words of a fixed weight in the Golay code. This is not the only method as we will show later.

B. Random walks methods

In this section, we show how random walks can lead to a distribution as closed as desired to the uniform k-out-of-n generator. The first construction applies random walk on a finite group and in particular on the symmetric group $\mathfrak{S}_n$. However, this method is not practical for large values of n since the size of the group becomes huge. The second construction makes use of a Markov walk on the set of blocks of a k-design. In both case the convergence to the uniform distribution is exponential. We illustrate our method by some examples. We consider codes, like Hamming, Golay, or quadratic residue codes, in order to obtain the appropriate design. We also give an example related to Hadamard matrices.

1) Random walk on a finite group, generalities: this topic in cryptographic context was investigated by Sloane in a nice survey [29]. Let us introduce objects and notations necessary for our study.

A random walk on a finite group G is currently defined by a probability Q on G and a homogeneous Markov chain

Γ_n, of space of states G, of transition matrix T given by $T_{g,h} = Q(gh^{-1}) = P(\Gamma_{n+1} = g|\Gamma_n = h)$. If the support of Q generates G, the chain is irreducible and its stationary distribution is the uniform distribution $U(G)$ on G. From the identity ($\Gamma_0 = \{e\}$), and a sequence of independent random variables X_n, the walk can be described inductively by $\Gamma_1 = X_1$, $\Gamma_n = X_n\Gamma_{n-1}$, the law of Γ_n being given by $Q^{(1)} = Q$ for Γ_1 and convolution product $Q^{(m)}(g) = Q*Q^{(m-1)}(g) = \sum_{h\in G} Q^{(m-1)}(gh^{-1})Q(h)$ for Γ_m (≥ 2). Here, we suppose that the chain is irreducible and aperiodic. Hence there exist an integer m_0 and a constant $c > 0$ such that $T^{m_0}_{g,h} \geq c\frac{1}{\#G}$ and then inequality (3) can be applied. This inequality can be translated in terms of distance (2), and can be improved in the case of symmetric walks ($Q(g) = Q(g^{-1})$) or for particular groups previously analyzed in a probabilistic way. For more details, see [2], [3], [10], [23], [24].

One application in cryptography is the popular stream parity de-skewer. Here G is the additive group $\{0,1\} := \mathbb{Z}/2\mathbb{Z}$ and $Q(0) = \frac{1}{2} - \beta$, $Q(1) = \frac{1}{2} + \beta$ with $0 \leq 2\beta < 1$. Then $T = \frac{1}{2}J + \beta S$ with $J := \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and $S := \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$. Using $JS = SJ = 0$, $J^2 = 2J$ and $S^2 = 2S$ we get the explicit formula

$$T^n = \frac{1}{2}J + (2\beta)^n S$$

and consequently

$$d(Q^n, U(\{0,1\})) = e^{-n\log(1/2\beta)}.$$

Moreover, for $k \geq 1$ fixed, the sequence of random variables $(\Gamma_{kn})_{n\geq 1}$ defines the Bernoulli SDS $B(\{0,1\}, Q^{(k)})$ whose entropy is, after simplification,

$$\begin{aligned} H_k &= \log 2 - \frac{1}{2}(1 + (2\beta)^k)\log(1 + (2\beta)^k) \\ &\quad + \frac{1}{2}(1 - (2\beta)^k)\log\Big(\frac{1}{1 - (2\beta)^k}\Big). \end{aligned}$$

Hence, $H_k - \log 2 \sim (2\beta)^{2k}$ as k tends to infinity.

2) Random transposition on the symmetric group $\mathfrak{S}_n$ and k-out-of-n generators: random walks on the symmetric group $\mathfrak{S}_n$ have been intensely studied (see the preceding references). Consider a uniform Bernoulli generator $B(\mathfrak{S}_n)$ on the group of permutations $\mathfrak{S}_n$. This generator defines a sequence of random variables $\Sigma_n(\cdot) = B(\mathfrak{S}_n)_n$. Then $C_n = \Sigma_n(\{1,\ldots,k\})$ is a sequence of random variables uniformly distributed in the set $\mathcal{P}^n_k$.

An interesting method to construct generators distinct from Fisher-Yates shuffle algorithm is to choose a set E of generators of $\mathfrak{S}_n$ and use a Bernoulli $B(E)$. A result of [10] states that the speed of convergence is in $e^{-\gamma}$ when the walk has a sufficiently large number of steps γ: more precisely, for $E = \{Id, (1,2), (1,2,\ldots,n), (n,n-1,n-2,\ldots,1)\}$ one has

$$d(G^{(36n^3(\log n+\gamma))}, U(\mathfrak{S}_n)) \leq \alpha e^{-\gamma}, \tag{4}$$

where $\alpha > 0$ is a universal constant and for all integers $\gamma \geq 0$. Then, random variables Γ_m (see above) represent a generator converging to the Bernoulli generator $B(\mathfrak{S}_n)$; it outputs, at each step, a permutation σ and $\sigma(\{1,\ldots,k\})$. Let $\Gamma_m[k]$ denotes this generator. Its distribution $Q^{(m)}_k$ on $\mathcal{P}^n_k$ is given, for each set A of k elements of $\mathcal{E}_n = \{1,\ldots,n\}$, by

$$Q^{(m)}_k(A) = \sum_{\substack{\sigma\in\mathfrak{S}_n \\ \sigma(\{1,\ldots,k\})=A}} Q^{(m)}(\sigma).$$

We observe that $\#\{\sigma \in \mathfrak{S}_n\,;\ \sigma(\{1,\ldots,k\}) = A\} = k!(n-k)!$ so that

$$\begin{aligned} d(\Gamma_m, U(\mathfrak{S}_n)) &= \frac{1}{2}\sum_{A\in\mathcal{P}^n_k}\Big|Q^{(m)}_k(A) - \frac{k!(n-k)!}{n!}\Big| \\ &\leq d(\Gamma_m[k], U(\mathcal{P}^n_k))\,. \end{aligned}$$

This last bound validates the construction of $\Gamma_m[k]$ with the same convergence property than that of Γ_m. However, this generator is not practical for large value of n, and in particular for $n = 24$.

3) Homogeneous symmetric random walk on a finite set: Let $\mathcal{M}$ be a finite set of elements and let E be a set of bijections of $\mathcal{M}$. We set

$$\mu := \#\mathcal{M} \quad \text{and} \quad \chi := \#E.$$

A general random walk on $\mathcal{M}$ with instructions in E is given by a distribution law $\mathcal{L}$ on $\mathcal{M}$ and a sequence of E-valued random variables $(X_n)_n$ of given distribution $\mathcal{P}_n$. An outcome $x = (x_k)_k$ of the random sequence $(X_k)_{k\geq 1}$ leads to a walk on $\mathcal{M}$ consisting to start from an initial point m_0 in $\mathcal{M}$, selected according to the law $\mathcal{L}$ (step 0) and the location of the walk after n steps is $m_n = x_n \circ x_{n-1} \circ \cdots \circ x_1(m_0)$. In the sequel, we only pay attention to a Markov symmetric homogeneous and uniform walk, that means a walk satisfying the following properties:

(j) The initial point m_0 is fixed.
(jj) Symmetry: for all Y in E the inverse map Y^{-1} belongs to E and the identity map belongs to E.
(jjj) The random variables X_n are independent, with uniform distribution $\mathcal{P}_n := U(E)$.

Therefore, the space of states of the corresponding Markov chain is $\mathcal{M}$ and the stochastic transition matrix T is given by

$$T_{i,j} := \frac{\#\{Y \in E\,;\ Y(i) = j\}}{\chi} \quad ((i,j) \in \mathcal{M}^2)\,.$$

According to property (jj), the matrix T is symmetric and so, has the uniform distribution on $\mathcal{M}$ for stationary probability. For our applications we assume that

(jv) The chain is mixing.

This assumption is equivalent to the fact that a power of S has all entries positive. Hence, we may define the following important parameter of the chain

$$\kappa := \min\{k \geq 0\,;\ \forall\,(i,j) \in \mathcal{M}^2,\ (T^k)_{i,j} > 0\}.$$

In other words κ is the minimum number of necessary steps to go from any state to any state.

Our next goal is to estimate the so called spectral hole of S. To this aim we use the symmetry of the chain by considering appropriated quadratic forms on the vector space $\mathbb{R}^{\mathcal{M}}$ equipped with the euclidean scalar product denoted by $\langle \xi | \xi' \rangle = \sum_{(i,j)\in\mathcal{M}2} \xi_i \xi'_j$, with norm $|| \cdot ||$. Each generator Y in E, acting on $\mathcal{M}$ is identified to an automorphism of $\mathbb{R}^{\mathcal{M}}$ permuting the canonical basis. It is represented by an orthogonal matrix, still denoted by Y. The action of Y applies at i will be denoted by $Y{\cdot}i$. Explicitly, Y is given by $Y_{i,j} = 1$ if $j = Y{\cdot}i$ and $Y_{i,j} = 0$ otherwise. The inverse Y^{-1} of Y corresponds to the transpose matrix Y^*. The stochastic transition matrix T defined above is now given by

$$T := \frac{1}{\chi} \sum_{Y\in E} Y.$$

By symmetry of T and Perron-Frobenius's theorem, T has μ eigenvalues λ_ν ($0 \le \nu < \mu$) whose the largest one is equal to 1, with multiplicity 1. Let ρ be the greater eigenvalue of T distinct from 1, we ordered real eigenvalues as follows:

$$-1 < \lambda_{\mu-1} \le \cdots \le \lambda_1 = \rho < \lambda_0 = 1\,.$$

Theorem 2: With the preceding definitions and notations, we have

$$-1 + \frac{2}{\chi} \le \lambda_{\mu-1} \quad \textit{and} \quad \rho \le 1 - \frac{4}{\kappa(\kappa+1)\chi}\,.$$

Proof.

1. The first inequality is easy to prove. By (jj), diagonal terms of T are equal to $1/\chi$, hence the matrix $\frac{\chi}{\chi-1}(T - \frac{1}{\chi}I)$ is stochastic with eigenvalues $\frac{\chi\lambda_\nu - 1}{\chi-1}$ between -1 and 1. In particular $-1 \le \frac{\chi\lambda_{\mu-1}-1}{\chi-1}$, which gives $-1 + \frac{2}{\chi} \le \lambda_{\mu-1}$, as expected.

2. The second inequality is more complex to prove. It relies on the comparison of two quadratic forms on $\mathcal{R}^{\mathcal{M}}$.

To every symmetric matrix A indexed on $\mathcal{M}$ we associate the quadratic form $Q_A(\xi) = \langle A\xi | \xi \rangle$. Eigenvalues $\alpha_0 \le \alpha_1 \le \cdots \le \alpha_{\mu-1}$ of A are given by Courant-Fisher theorem (also called mini-max theorem) [12]:

$$\alpha_\nu = \min_F \{ m(F)\,;\ \dim(F) = \nu + 1 \}, \quad 0 \le \nu < \mu,$$

the minimum being calculated over the set of all subspaces F of $\mathbb{R}^{\mathcal{M}}$ of dimension $\nu + 1$ and

$$m(F) := \max\{ \langle A\xi | \xi \rangle\,;\ ||\xi|| = 1,\ \text{and } \xi \in F \}.$$

A straightforward consequence of this theorem is

Corollary 1: Let Q_A and $Q_{A'}$ two quadratic forms on $\mathbb{R}^{\mathcal{M}}$, of symmetric matrices A and A' and eigenvalues λ_ν, λ'_ν respectively (indexed in decreasing order). If for a constant $C > 0$ we have $Q_{A'} \le CQ_A$, then $\lambda'_\nu \le C\lambda_\nu$ for every index ν, $0 \le \nu < \mu$.

For our purpose, choose $A = I - T$. Then we have

$$\begin{aligned} Q_{I-T}(\xi) &= \frac{1}{2}\sum_{i,j} (\xi_i - \xi_j)^2 T_{i,j} \\ &= \frac{1}{2\chi} \sum_{\substack{Y\in E \\ i\in\mathcal{M}}} (\xi_i - \xi_{Y\cdot i})^2 \end{aligned}$$

The bound will result from the following main lemma.

Lemma 1: Consider the symmetric matrix $B = I - \frac{1}{\mu}J$ where J has all its coefficients equal to 1. Then

$$Q_B \le \frac{\chi\kappa(\kappa+1)}{4} Q_{I-T}\,. \tag{5}$$

Proof. For every pair (i,j) of states, let $Y^{i,j} = Y^{i,j}_{k(i,j)} \cdots Y^{i,j}_0$ with $Y^{i,j}_0 = I$ be a composition of elements of E such that $j = Y^{i,j}{\cdot}i$ with $k(i,j)$ minimal. Set $|Y^{i,j}| := k(i,j)$. We have $|Y^{i,j}| \le \kappa$ and

$$\xi_i - \xi_j = \sum_{s=0}^{|Y^{i,j}|-1} (\xi_{Y^{i,j}_s \cdots Y^{i,j}_0 \cdot i} - \xi_{Y^{i,j}_{s+1}\cdots Y^{i,j}_0 \cdot i}).$$

Applying Cauchy-Schwarz inequality gives

$$\begin{aligned} (\xi_i - \xi_j)^2 &= |Y^{i,j}| \sum_{s=0}^{|Y^{i,j}|-1} (\xi_{Y^{i,j}_s \cdots Y^{i,j}_0 \cdot i} - \xi_{Y^{i,j}_{s+1}\cdots Y^{i,j}_0 \cdot i})^2 \\ &\le \chi |Y^{i,j}| \sum_{s=0}^{|Y^{i,j}|-1} (\xi_{Y^{i,j}_s \cdots Y^{i,j}_0 \cdot i} - \xi_{Y^{i,j}_{s+1}\cdots Y^{i,j}_0 \cdot i})^2 T_{\lambda_{ijs}} \end{aligned}$$

where $\lambda_{ijs} := Y^{i,j}_s \cdots Y^{i,j}_0 \cdot i, Y^{i,j}_{s+1} \cdots Y^{i,j}_0 \cdot i$.
Multiply these inequalities by $1/\mu$ and add all of them, first by summing on j and then on i. The summation on the left side simply gives

$$\frac{1}{\mu} \sum_{(i,j)\in\mathcal{M}^2} (\xi_i - \xi_j)^2 = 2Q_B(\xi).$$

Hence, the summation on the right side is greater than $2Q_B(\xi)$. By collecting all right terms according to the values taking by $|Y^{i,j}|$ for i fixed we get

$$(\chi/2)\frac{1}{\mu} \sum_{k=1}^{\kappa} k(2Q_{I-T})$$

since by minimality, there is no loop in the path going from i to j and constructed from $Y^{i,j}$. Therefore, summing over i now leads to the desired inequality $Q_B(\xi) \le \frac{\kappa(\kappa+1)\chi}{4} Q_{I-T}$.

Eigenvalues of $\frac{1}{\mu}J$ being 1 and 0, those of B are then 1 (with multiplicity $\mu - 1$) and 0. Then Corollary 1 gives $1 \le \frac{\kappa(\kappa+1)\chi}{4}(1 - \rho)$, which is the second inequality of theorem 2.

Using Theorem 2 and the fact that T is a symmetric stochastic matrix of order μ, we get the following inequality

$$\left|\left|\left| T^n - \frac{1}{\mu}J \right|\right|\right|_2 \le \left(1 - \frac{4}{\kappa(\kappa+1)\chi}\right)^n,$$

where $|||\cdot|||_2$ denotes the quadratic norm of operators. Now let $P_m^{(n)}$ be the distribution of the walk obtained from the state m_0, and set

$$d(n) := \max_{m\in\mathcal{M}} d(P_m^{(n)}, U(\mathcal{M})).$$

Let I_{m_0} be the column vector in $\mathbb{R}^{\mathcal{M}}$ with all entries 0 except the entry corresponding to m_0. From Cauchy-Schwarz inequality and symmetry of T^n,

$$\begin{aligned} d(P_m^{(n)}, U(\mathcal{M})) &= \frac{1}{2}\sum_{j\in\mathcal{M}}\Big|T^n_{i,m_0} - \frac{1}{\mu}\Big| \\ &\le \frac{1}{2}\sqrt{\mu}\Big(\sum_{j\in\mathcal{M}}\Big(T^n_{i,m_0} - \frac{1}{\mu}\Big)^2\Big)^{1/2} \\ &\le \frac{1}{2}\sqrt{\mu}\Big|\Big|\Big(T^n - \frac{1}{\mu}J\Big)I_{m_0}\Big|\Big|_2 . \end{aligned}$$

Since

$$\begin{aligned} ||(T^n - \frac{1}{\mu}J)I_{m_0}||_2 &\le |||(T^n - \frac{1}{\mu}J)|||_2||I_{m_0}||_2 \\ &\le \Big(1 - \frac{4}{\kappa(\kappa+1)\chi}\Big)^n, \end{aligned}$$

we obtain

$$d(n) \le \frac{\sqrt{\mu}}{2}\Big(1 - \frac{4}{\kappa(\kappa+1)\chi}\Big)^n$$

that can be transformed into

$$d(\lceil ab + b\gamma\rceil) \le e^{-\gamma}. \tag{6}$$

with

$$\begin{aligned} a &= \frac{1}{2}\log\mu - \log 2 \\ b &= -\frac{1}{\log\Big(1 - \frac{4}{\kappa(1+\kappa)\chi}\Big)}. \end{aligned}$$

This inequality exhibits the speed of convergence of the walk to the uniform distribution.

In the next subsection, we apply this general theory to the specific case of random walks on the blocks of a design.

4) Random walk using the automorphism group of a design: We now introduce an efficient uniform k-out-of-n generator, using a random walk on a block of a $k-$design. The walk consists in acting on the block a set E of appropriate generators of the automorphism group of the design.

Gk-n(N) Algorithm

INPUT : N
OUTPUT : a binary vector of Hamming weight k and length n
Choose a block m of weight b among the blocks of a $k-(n,b,\lambda)$ design. The automorphism group A of the design must be transitive on the blocks.
If A is $(b-k)$-transitive on the blocks,
then
(b.1) replace m by m', replacing the first $b-k$ coordinates equal to 1 in m by zeros
(b.2) randomly act on m' the generators of G, N times
(b.3) output the obtained word.
else
(c.1) randomly act on m the generators of G, N times, and obtain m'
(c.2) randomly choose k coordinates equal to 1 in m' using a k-out-of-b generator
(c.3) output the obtained word.

We give in the sequel, examples of constructions for various parameters k and n.

Example 2: Generator 5-out-of-24 associated to the Mathieu group M_{24}

Let G_{24} be the extended binary Golay code. The Mathieu group, M_{24}, is the automorphism group of G_{24} and can be generated by the following four permutations acting on the coordinates of the words of the Golay code:

$$S : i \mapsto i+1, \quad V : i \mapsto 2i, \quad U : i \mapsto -1/i$$

and

$$W : \begin{cases} \infty \mapsto 0,\ 0 \mapsto \infty, \\ i \mapsto -(i/2)2 & \text{if } i \text{ is a quadratic residu modulo 23,} \\ i \mapsto (2i)2 & \text{otherwise.} \end{cases}$$

G5-24(N) Algorihm

INPUT : N
OUTPUT : a binary vector of Hamming weight 5 and length 24
(a) choose an octad of G_{24} : m
(b) replace m by m', replacing the first three coordinates equal to 1 in m by zeros
(c) randomly act on m' the four generators or their inverse or the identity, N times
(d) output the obtained word.

Note that M_{24} is 5-transitive on octads. This is why step (b) can be done before acting the generators.

We have now to explicitly construct the random generator of octads. Since the size of M_{24} is huge ($\#M_{24} = 210.33.5.7.11.23$), the speed of convergence of a walk on the Mathieu group would be mediocre.

Thus we introduce a Markov walk on the set $\mathcal{M}$ of octads by the action of the four aforementioned generators of M_{24}: S, V, U et W. Let I be the identity. Now we make the walk symmetrical by taking the following transition set

$$E := \{I, S, S^{-1}, U, V, V^{-1}, W, W^{-1}\},$$

with the uniform probability.

To show that *G5-24(N) Algorihm* realizes a uniform 5-out-of-24 generator asymptotically with exponential speediness, we determine equation (6) with the correct parameters.

We have to calculate the minimal number of times we have to act elements of E on a given octad in order to obtain all the octads. Since the walk is symmetric, this number corresponds to κ. Taking into account that the identity belongs to E, Table 2 shows that $\kappa = 7$.

Number of octads	Numbers of steps
683	6
76	7

Table 2: number of steps to obtain, during the walk, all octads from a specific octad

With the above notations, we have $\mu = 759$, $\chi = 8$, and $\kappa = 7$. We obtain

$$d(292 + 111\gamma) \leq e^{-\gamma}.$$

The following histogram (Fig. 1) gives a statistical view of what is going on in the case of a very short walk. It represents the number $N(f)$ of octads obtained f times during 7590 walks of length 11 (7590 is equal to ten times the number of octads). The distribution is rather good.

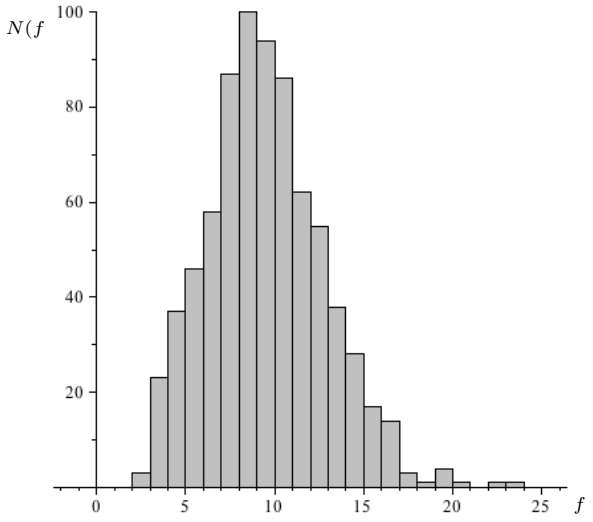

Figure 1. Number $N(f)$ of octads obtained f times during 7590 walks of length 11

Example 3: Random walk on a ternary Golay code

In the previous example, we focused on the binary Golay code. There also exists a ternary Golay code with parameters $[12, 6, 6]$ whose words of Hamming weight equal to 6 yield a 5-$(12, 6, 1)$ design. The number of blocks of the design being equal to 132. Thus, with a similar construction, we can obtain a 5-out-of-12 generator. The automorphism group of the design is of order $95040 = 2^6.3^3.5.11$ and is 5-transitive on the blocks of the design. This group is generated by the following four permutations on the set of 12 coordinates.

$A1 = (5, 9, 12, 7)(6, 10, 11, 8)$ of order 4
$A2 = (3, 12, 7, 9)(4, 6, 10, 8)$, of order 4
$A3 = (1, 3)(4, 8)(7, 11)(9, 12)$ of order 2
$A4 = (2, 4, 5, 8)(6, 9, 10, 12)$ of order 4

and we consider

$$E := \{A1,\ (A1)^{-1},\ A2,\ (A2)^{-1},\ A3,\ A4,\ (A4)^{-1},\ I\},$$

where I represents the identity permutation. We have $\chi = 8$ and $\kappa = 6$ from Table 3.

Number of blocks	Numbers of steps
74	5
58	6

Table 3: number of steps to obtain, during the walk, all blocks from a specific block

With the appropriate parameters, equation (6) becomes

$$d(146 + 83\gamma) \leq e^{-\gamma}.$$

Example 4: Generator 2-out-of-31

In this example, we consider a 2-$(31, 7, 7)$ design whose automorphism group A is not 2-transitive on its blocks. The number of blocks is 155. The group A is a permutation group acting on the set of 31 coordinates. It is of order $465 = 3.5.31$ and is generated by the following two permutations

$A1 = (1, 16, 15, 13, 9)(2, 18, 19, 21, 25)(3, 20, 23, 29, 10)$
$(4, 22, 27, 6, 26)(5, 24, 31, 14, 11)(7, 28, 8, 30, 12)$
and
$A2 = (2, 29, 10, 5, 20, 6, 17, 15, 21, 3, 26, 19, 9, 8, 11)$
$(4, 23, 28, 13, 27, 16, 18, 12, 30, 7, 14, 24, 25, 22, 31)$.

Consider the set of generators

$$E := \{A1,\ (A1)^{-1},\ A2,\ (A2)^{-1},\ I\}$$

and act the elements of E on a block m, N times, in order to obtain a random block m'. This block can be represented by a vector of weight 7 and length 31.
Computation gives $\kappa = 6$ and with the appropriate parameters, equation (6) becomes

$$d(95 + 52\gamma) \leq e^{-\gamma}\,.$$

Then we have to randomly choose 2 coordinates equal to 1 in m' using a 2-out-of-7 generator. To do this we consider a 2-$(7, 3, 1)$ design. The automorphism group of the design is $PSL(2, 7)$ and is 2-transitive on its blocks. We just have to replace the first 1 in m' by a zero and carry out a walk on this vector. We then obtain a random vector of weight 2 and length 31. Notice that generators 2-out-of-31 and 2-out-of-7 can be executed in parallel.

Example 5: 3-out-of-16 generator.

It is important to choose a correct design in order to carry out the walk. In fact, to obtain a uniform generator, our method requires to consider designs with automorphism group that is transitive on the blocks.

We introduce here an example of automorphism group of a design, which is not transitive on the blocks. The design is constructed from a Hadamard matrix. Recall that the order of a Hadamard matrix is necessarily a multiple of 4 and Sylvester construction shows that there exist Hadamard matrices of order 2^i for all positive integer i. Let H be the following Hadamard matrix of order 16

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & - & - & - & - & - & - & - & - \\
1 & 1 & 1 & 1 & - & - & - & - & 1 & 1 & 1 & 1 & - & - & - & - \\
1 & 1 & - & - & 1 & 1 & - & - & 1 & 1 & - & - & 1 & 1 & - & - \\
1 & 1 & - & - & - & 1 & 1 & - & 1 & - & - & 1 & - & - & 1 & 1 \\
1 & 1 & - & - & 1 & - & - & 1 & - & 1 & 1 & - & - & - & 1 & 1 \\
1 & 1 & - & - & - & - & 1 & 1 & - & - & 1 & 1 & 1 & 1 & - & - \\
1 & - & - & 1 & - & - & 1 & 1 & 1 & - & - & 1 & - & - & - & 1 \\
1 & - & - & 1 & 1 & 1 & - & - & - & - & 1 & 1 & 1 & - & - & 1 \\
1 & - & 1 & - & 1 & - & 1 & - & 1 & - & 1 & - & 1 & - & 1 & - \\
1 & - & 1 & - & 1 & - & - & 1 & 1 & - & - & 1 & - & 1 & - & 1 \\
1 & - & 1 & - & - & 1 & 1 & - & - & 1 & 1 & - & - & 1 & - & 1 \\
1 & - & 1 & - & - & 1 & - & 1 & - & 1 & - & 1 & 1 & - & 1 & - \\
1 & - & - & 1 & 1 & - & 1 & - & - & 1 & - & 1 & - & 1 & 1 & - \\
1 & - & - & 1 & - & 1 & - & 1 & 1 & - & 1 & - & - & 1 & 1 & - \\
1 & 1 & 1 & 1 & - & - & - & - & - & - & - & - & 1 & 1 & 1 & 1
\end{bmatrix}$$

where the symbol "$-$" stands for "-1". This matrix yields a 3-$(16, 8, 3)$ design with 30 blocks. The automorphism group is of order $2688 = 2^7.3.7$ and is generated by the following permutations on the set of 16 coordinates:
$A1 = (1, 5)(2, 10)(3, 12, 15, 8, 14, 6)(4, 7, 16, 11, 13, 9)$
$A2 = (2, 3, 4)(6, 8, 7)(9, 12, 10)(13, 16, 15)$
$A3 = (5, 10)(6, 9)(7, 12)(8, 11)$.
When acting the generators on the blocks, we obtain 2 orbits. One of order 28 and the other one of order 2. It means that from a given block, it is not possible to obtain all other blocks during the walk.

Let us now consider the design D whose blocks are the words of weight 12 of the extended Hamming code of parameters $[16, 12, 3]$. This is a 3-$(16, 12, 55)$ design with 140 blocks. The automorphism group of D is a permutation group A acting on a set of cardinality 16. It is of order $322\,560 = 2^{10}.3^2.5.7$ and is generated by the following automorphisms:
$A1 = (5, 13)(6, 10)(7, 16)(9, 15)$
$A2 = (5, 16)(6, 9)(7, 13)(10, 15)$
$A3 = (3, 6)(4, 7, 10, 12)(8, 13, 15, 14)(9, 11)$
$A4 = (1, 5)(3, 6)(7, 12)(8, 15)$
$A5 = (2, 10, 13, 12, 4)(3, 16, 11, 8, 15)(5, 7, 9, 6, 14)$.
Consider the set of 8 generators

$$E := \{A1,\ A2,\ A3,\ (A3)^{-1},\ A4,\ A5,\ (A5)^{-1},\ I\}.$$

Then, choose a block m of the design and, since $\mathrm{Aut}(D)$ is 3-transitive, just perform a walk on its first 3 coordinates using the elements of E. Here $\kappa = 6$ and so we get a random word of weight 3 and length 16 as desired.

Number of blocks	Numbers of steps
4	4
119	5
19	6

Table 4: number of steps to obtain, during the walk, all blocks from a specific block

The speed of convergence is given by

$$d(148 + 83\gamma) \leq e^{-\gamma}.$$

V. Conclusion

In this paper, we introduced a type of generators, which has not been deeply studied in the literature. Yet, k-out-of-n generators have a wide practical interest, particularly for developing secure applications. Our constructions make use of t-designs in order to obtain uniformity and run random walks in order to control the accuracy of convergence.

We proposed methods to efficiently construct such generators and studied in detail special cases. The speed of convergence of our generators is better than any known k-out-of-n generators.

Acknowledgment

This work has partially been supported by ANR NUGET.

References

[1] Bonnecaze A. and Liardet P., *Efficient Uniform k-out-of-n Generators*, ICSNC 2010, Fifth International Conference on Systems and Networks Communications (ICNS 2010), pp. 177-182.

[2] Aldous D., *Random walks on finite groups and rapidly mixing Markov chains*, Séminaire de Probabilités XVII (1981/82), Lecture Notes in Mathematics, 1059, Springer, Berlin (1983), pp. 243-297.

[3] Aldous D., *Shuffling cards and stopping times*, the American Mathematical Monthly, 93, (1986), pp. 333-348.

[4] Atlas of Finite Group Representations. `http://brauer.maths.qmul.ac.uk/Atlas/v3/spor/M24/`, January 2011.

[5] Blum L., Blum M. and Shub M., *A Simple Unpredictable Pseudo-Random Number Generator*, SIAM Journal on Computing, volume 15, May 1986, pp. 364-383.

[6] Bonnecaze A., Liardet P., Gabillon A. and Blibech K., *Threshold Signature For Distributed Time Stamping Scheme*, Annals of Télécommunications, Volume 62, N. 11-12 (2007), pp. 1353-1364.

[7] Bonnecaze A. and Trebuchet P., *Secure Time-Stamp Schemes: A Distributed Point Of View*, Annals of Télécommunications, Volume 61, N. 5-6, (2006), pp. 662-681.

[8] Colbourn C. and Mathon R., *Steiner Systems*, in Handbook of Combinatorial Designs, second edition, Discrete Mathematics and Its Applications (2007), pp. 102-110.

[9] `http://www.ccrwest.org/cover.html`, January 2011

[10] Diaconis P. and Saloff-Coste L., *Comparison techniques for random walks on finite groups*, The Annals of Probability, Volume 21, No 4, (1993), pp. 2131-2156.

[11] Durstenfeld R., *Algorithm 235: Random permutation*, Communications of the Association for Computing Machinery, volume 7, issue 7, (1964), pp. 420.

[12] Horn R.A. and Johnson R.C., *Matrix analysis*, Cambridge Press, Cambridge, 2nd edition, (2008).

[13] Kemeny J. and Snell L., *Finite Markov chains*, Van Nostrand company, Princeton, (1960).

[14] Knuth D.E., *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Addison-Wesley, 2nd edition, (1998).

[15] van Lint J.H. and Wilson R.M, *A Course in Combinatorics*, Cambridge University Press (1992).

[16] MacWilliams F.J. and Sloane N.J.A., *The Theory of Error-Correcting Codes* North-Holland, Eight impression, (1993).

[17] Nijenhuis A. and Wilf H.S., *Combinatorial Algorithms for Computers and Calculators*, Academic Press, Inc., 2nd edition, (1978).

[18] NIST: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications: `http://csrc.nist.gov/publications/nistpubs/800-22-rev1/SP800-22rev1.pdf`, January 2011.

[19] Ornstein D.S., *Ergodic theory, randomness and dynamical systems*, Yale Mathematical Monographs No. 5, Yale Univ, (1974).

[20] Ray-Chaudhuri D.K. and Tianbao Z., *A recursive method for construction of designs*, Discrete Mathematics, Volumes 106-107, (1992), pp. 399-406.

[21] Rolland R., *Sécurité des générateurs pseudo-aléatoires*, `http ://www.acrypta.fr.`, January 2011.

[22] Rolland R., *Personal communication*, April 20, 2010.

[23] Saloff-Coste L., Lectures on finite Markov chains, in Lectures on Probability Theory and Statistics, Ecole d'été de Probabilités de Saint-Flour XXVI-1996, E. Giné, G.R. Grimmett and L. Saloff-Coste (Authors), Lecture Notes in Math, No. 1665, pp. 301-413.

[24] Saloff-Coste L., *Random Walks on Finite Groups*, Probability on discrete structures, 263–346, Encyclopaedia of Mathematical Sciences, 110, Springer, Berlin. Harry Kesten Editor, (2004), pp. 263–346.

[25] Shields P., *The Ergodic Theory of Discret Sample Paths*, Graduate Studies in Mathematics: 13, American Mathematical Society, (1996).

[26] Seneta E., *Non-negative Matrices and Markov Chains*, Springer Series in Statistics, Springer, Revised Printing (2006).

[27] Shoup V., *Practical Threshold Signatures*, EUROCRYPT'00: Proceedings of the 19th international conference on Theory and application of cryptographic techniques, Springer-Verlag (1999), pp. 207-220.

[28] Sinai Ya. G., *On a weak isomorphism of transformations with invariant measure*, Matematidheskii. Sbornik. (N.S.), 63 (105), No. 1 (1964), pp. 23-42.

[29] Sloane N.J.A., *Encrypting by Random Rotations*, Thomas Beth (Ed.): Cryptography, Proceedings of the Workshop on Cryptography, Burg Feuerstein, Germany, March 29 - April 2, 1982. Lecture Notes in Computer Science 149 Springer (1983), pp. 71-128.

[30] Steiner J., (1853), *Combinatorische Aufgabe*, Journal fur die Reine und Angewandte Mathematik 45, pp. 181-182 .

[31] Steiner Systems, `http://www.ccrwest.org/cover/steiner.html`, January 2011.

[32] Walters P., *An Introduction to Ergodic Theory*, Graduate texts in mathematics: 79, Springer-Verlag (1982).

[33] Yao A., *Theory and applications of trapdoor functions*. In Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science (1982), pp. 80-91.