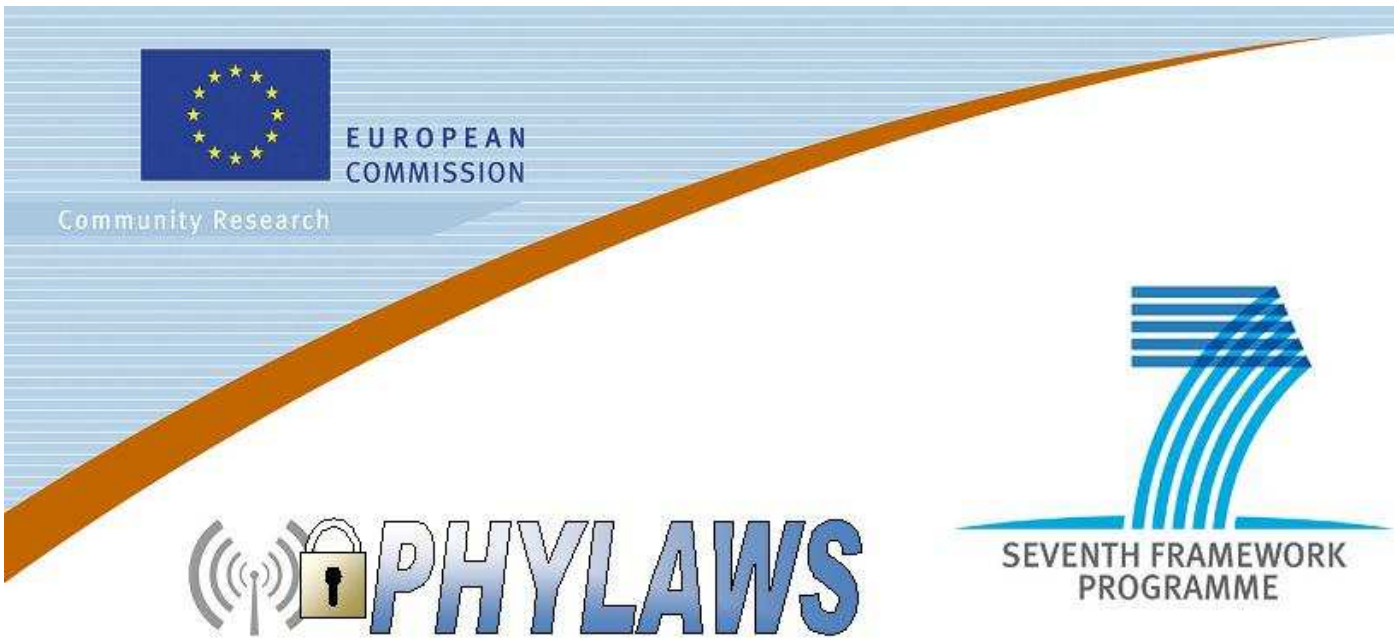

Project PHYLAWS (Id 317562)
PHYsical LAyer Wireless Security

Deliverable D.4.2

TRANSEC upgrades of existing RATs - simulation and analyses complements

Version 1.0 - 30 / 11 / 2015

Change History

Version	Date	Description	Affected Sections
0.a	2015-10-30	First drafts including only TCS contributions	All
1.0	2015-11-30	Revised first version	All

List of Contributors

Partner	Contributors
Thales	François Delaveau, Renaud Molière, Christiane Kameni
VTT	
CEL	
TPT	

Table of contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose of the deliverable

Deliverable D4.2 is a complement of deliverable D4.1. Deliverables D4.1 and D4.2 are relevant to task T4.1 of Work Package WP4, which is the outcome of WP2 (State of the Art, and especially Deliverable D2.4) and by academic studies and signal records performed in WP3, especially Deliverable D3.1, D3.2 and D3.3).

In WP2 we outlined why secured and early authentication of devices and of Channel State Information (CSI) are necessary to implement several further protections based on Physec concepts, such as:

- Artificial Noise (AN)
- Secret Key Generation (SKG)
- Secret Codes (SC)

In WP4 task T4.1, we developed a new exchange protocol based on Tag Signals (TS), called Interrogation and Acknowledgement Sequences (IAS), and demonstrated its proof of concept concerning:

- Secure pairing
- Resilience to passive eavesdropper and active radio hackers (such as intelligent jammers and Man in the Middle Systems)
- Accurate estimation of radio channels

The results of the present deliverable strengthen this proof of concept with additional simulations and experiments. In addition, we complete analyses and definitely conclude on the “tremendous” perspectives of the proposed TS and IAS schemes concerning:

- Improvement in secure pairing of devices
- Possibility of accurate CSI and radio measurements for:
    - Initiating AN, SKG and SC.
    - Supporting further users’ authentication cipher key negotiation.

This document is particularly in close relation with deliverables D2.4, D3.1, D4.1 and D4.3.

1.2 Scope of the deliverable

This deliverable is organized in the following manner:

- Section 1 introduces the content of the deliverable.
- Section 2 defines the terminology used in the deliverable, summarizes the main achievements of D4.1 and underlines the complements provided by deliverable D4.2.
- Section 3 investigates the random design of PN sequences for building Tag Signals, such as announced in Deliverable D4.1. Requirements, generation techniques, simulation protocols and performance are particularly studied in relation with the IAS protocol.
- Section 4 extends the CSI processing derived in deliverable D4.1 by describing estimation methods for Channel Frequency Response (CFR) and Channel Impulse Response (CIR), for LTE, UMTS and GSM.

- Section 5 completes deliverable D4.1 by developing estimation techniques of Signal to Interference + Noise ratio (SINR) and Signal to Noise ratio (SNR). Such measurements are crucial to control the radio advantage provided by tag signal or by other methods (such as artificial noise and beam forming), and to implement secrecy coding schemes (see deliverable D4.3).

- Section 6 concludes the deliverable by recapping its main results.

- Section 7 includes the references used for the redaction of the deliverable.

2 Definitions, achievements of D4.1 and complements of D4.2

2.1 Terms, concepts and definitions

Term	Definition
ALOHA	Access Local in HAwaï: random access method for communication networks designed by the Hawaï University in the 1970's. (Note : The expression ALOHA is derived from Hawaï language)
AN	Artificial Noise
AP	Access Point
BCCH	Broadcasting Control CHannel
BF	Beam-Forming
BTS	Base Transceiver Station
BSML	Binary Sequence of Maximal Length
CDMA	Code Division Multiple Access
CFR	Channel Frequency Response
CIR	Channel Impulse Response
COMSEC	Communication Security is relevant to the protection of the content of the user messages (voice, data). COMSEC applies either at the radio interface or at upper layer. COMSEC techniques involve ciphering, authentication and integrity control of signalling and users' data at several protocol layer and interfaces (examples are point to point ciphering of each user data flux, ciphering of IP packets, ciphering of artery, etc.).
CP	Cyclic Prefix
CRS	Cell-specific Reference Signal
CSC	Chaotic Spreading Code
CSI	Channel State Information
DL	Down Link – Nominal sense of the communication from a network/Base Station/Access Point toward a terminal
DM-RS	DeModulation-Reference Signal
DS	Direct-Sequence
DSS	Direct-Spread-Sequence
DSSS	Direct-sequence Spread Spectrum
eNodeB	Evolved Node B (equivalent to BTS in LTE)
FDD	Frequency-division duplexing (communicating devices operate in different carrier frequencies)

FFT	Fast Fourier Transform
Fu-Du	See Full-Duplex
Full-duplex	Two directional communication where both parties transmit simultaneously
FWD	ForWarD sense (of transmission)
GMSK	Gaussian Minimum Shift Keying – Modulation in GSM
GSM	Global System for Mobile communications (2G)
IAS	Interrogation and Acknowledgement Sequences
IFF	Identification of Friend and Foe
IFFT	Inverse Fast Fourier Transform
IJ	Intelligent Jammer
IoT	Internet of Things
LOS	Line Of Sight
LSFR	Linear Shift Feedback Register
LTE	Long Term Evolution (4G)
LTF	Long Training Field
M2M	Machine to Machine
MIMO	Multi-input multi-output: use of multiple antennas at both the transmitter and receiver
MISO	Multi-input single-output: use of multiple antennas at the transmitter
MITM	Man-In-The-Middle
NETSEC	Network Transmission Security: NETSEC is relevant to the protection of the signalling of the network. NETSEC applies mainly at the radio interface and at the medium access protocol layer, with request to upper protocol layers. NETSEC techniques involve mainly transmitter authentication protocols, integrity control and ciphering of signalling data.
OFDM	Orthogonal frequency division multiplex – Modulation technology where a signal is split into several narrowband channels at different frequencies.
OFDMA	Orthogonal Frequency Division Multiple Access
OVSF	Orthogonal Variable Spreading Factor
PHYSEC	Physical Layer Security is generic term that will be used in this project to design all kind of protection techniques that are based on the use of the physical layer sensing and/or measurement.
PLCP	Physical Layer Convergence Procedure
PN	Pseudo Noise

PRNG	Pseudo Random Noise Generator
PSS	Primary Synchronization Sequence
PUSCH	Physical Uplink Shared Channel
RAT	Radio Access Technology (e.g. FDMA, TDMA, CDMA, OFDM)
RB	Resource Block
RE	Resource Element
RTN	ReTurN sense (of transmission)
SC	Secrecy Code
SF	Spreading factor of DSSS signals at chip period T' and rate 1/T'. The following convention is applied in the document. - When the DSSS code is applied to a symbol stream at symbol period T (rate 1/T), SF is the ratio of the rate of the modulated signal and of the rate of the symbol stream: SF = T/T' - When the DSSS code is pure (i.e. when the symbol stream under the DSSS code is identically equal to the same value), SF is the processing gain linked the integration duration T_I of the receiving processing: SF=T_I/T'.
SIM	Self-Interference Mitigation
SIMO	Single-Input Multi-Output (use of multiple antennas at the receiver)
SINR	Signal-to-noise and interference ratio
SISO	Single-input single-output (use of single antennas at both the transmitter and receiver)
SKG	Secret Key Generation
SNR	Signal-to-Noise Ratio
SRS	Sounding Reference Signal
STF	Short Training Field
Superimposed	Placed or set over or above on something else. In communication, a signal transmitted at the same time and at the same frequency as another.
TDD	Time-division duplexing
TDMA	Time Division Multiple Access
TJ	Time Jitter
TRANSEC	Transmission Security: TRANSEC is relevant to the protection of the waveform against

	interception/direction finding of the radio signal, jamming of the user receiver, and intrusion attempts into the radio-communication access protocol. Applies mainly at the radio interface.
Trustworthy[1]	Secure, reliable and resilient to attacks and operational failures; ensures quality of service; protects user data; ensures privacy and provides trusted tools to support security management.
TS	Tag Signal : Low power signal, which is transmitted at the same time, at the same frame or slot, and at the same carrier than the user signal. Can be used e.g. to identify the sender.
TSNR	Tag to Signal + Noise Ratio
TSR	Tag to Signal Ratio
TTI	Time Transmission Interval
UE	User Equipment
UL	Up Link – Nominal sense of a communication from a terminal to a network/Base Station/Access Point
UMTS	Universal Mobile Telecommunications System (3G)
USS	Uncoordinated Spread Spectrum
W-CDMA	Wideband-CDMA

[1] According to the European Commission Work Programme 2011-2012 for ICT

2.2 Recall of the content of D4.1 – Complements provided by D4.2

2.2.1 Transec solutions studied in D4.1

Deliverable D4.1 provided a survey of existing Transec schemes related to Phylaws project and introduced innovative schemes for both secure pairing and radio channel estimation. It thus presented spread spectrum modulated radio signals, Uncoordinated Spread Spectrum (USS) and Time Jitter (TJ) schemes, system for Identification Friend or Foe (IFF), Artificial Noise (AN) and Full-Duplex (Fu-Du). Combined with Tag Signals (TS, introduced in Deliverable D2.4 and studied in Deliverable D4.1) these techniques are at the core of the Interrogation and Acknowledgement Sequences (IAS) protocol, dedicated to Transec improvements at the start of wireless communications.

2.2.2 IAS protocol

2.2.2.1 Presentation of the protocol

In deliverable D4.1 was presented a new protocol based on tag signals and Interrogation and Acknowledgement Sequences (IAS). The description of the complete IAS scheme is shown in Figure 1. This protocol enables a secure pairing between Alice and Bob without any prior shared knowledge. It both provides secure pairing between Alice and Bob and accurate estimations of the legitimate CIR. Therefore, it also provides suitable conditions for the implementation of other Physec schemes such as Secret Key Generation (SKG), Secrecy Coding (SC) and Artificial Noise (AN). Moreover, this new Transec protocol may provide additional protections for users' authentication, and for symmetric and asymmetric ciphering.

Full description of the scheme and considerations about its resilience against any kind of attacks are available in §4.1.2 of deliverable D4.1.

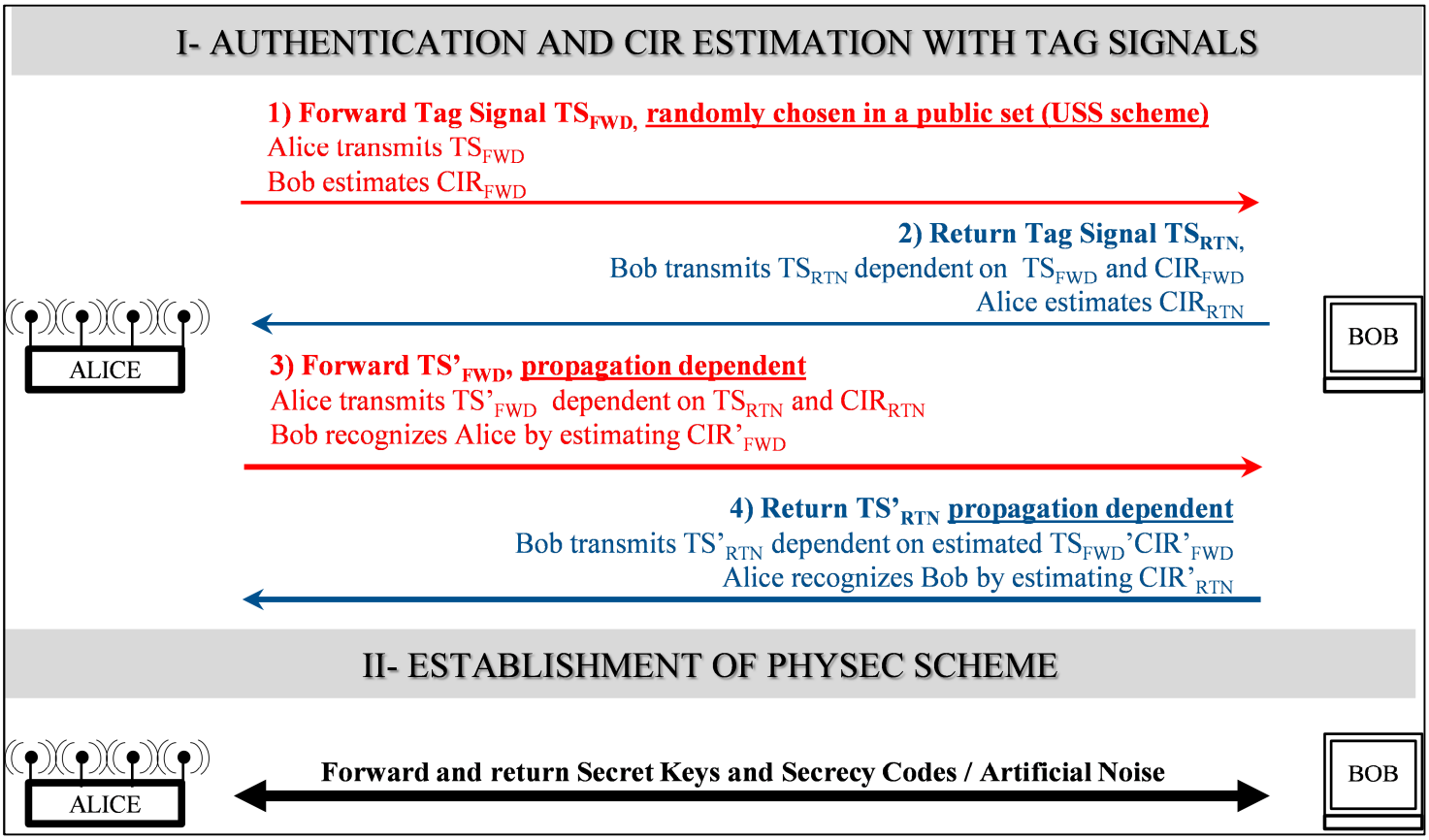

Figure 1: Protocol for secure pairing and implementation of PHYSEC with tag signals

One of the main principles of this protocol is the adaptive design of tag signals in function of on-going CIR measurements under the assumption of channel reciprocity. This construction can be done by two ways:

- Tag signals are directly designed from CIR measurements. This solution provides the highest level of security. However, it is difficult for such technique to provide suitable sequences, i.e, sequences with good correlation properties that facilitate detection and further CIR measurements. Moreover, the risk of mismatch between the sequence generated by Alice (resp. Bob) and the sequence researched by Bob (resp. Alice) is high. This risk would cause too much missed detection or would induce complex post-processing for compensation.
- Tag signals are selected from CIR measurements in pre-computed sets of sequences. These sets contain low-correlated codes to solve the problem of detection. Besides, the size and the number of these sets are large enough to prevent any attack from Eve. Channel estimation is thus considered as a key that determines the set and the sequences to use by Alice and Bob.

2.2.2.2 Focus on random generation of low-correlated sequences

For security reasons (detailed in D4.1 and summarized in §3.2.1.1), the sets shall not contain classical PN sequences such as Gold or Kasami sequences and shall be generated as arbitrarily as possible.

Starting from this statement, the following crucial questions arise concerning the construction of pre-computed sets of random sequences:

- What techniques can be used to generate the sequences of a set?
- What level of cross-correlation and autocorrelation can be expected in a set?
- Which parameters shall be taken into account in the design of a set?
- How large can be the set for a certain value of cross-correlation between the codes?
- How long does the generation process take?

All these questions were stated in deliverable D2.4. The complete study is done in §3 of this deliverable.

2.2.3 Reference signals for CIR and CFR

2.2.3.1 Recall on signal processing for CIR and CFR estimation for WiFi

In deliverable D4.1, theoretical background and algorithms for estimation of Channel Impulse Response (CIR) or Channel Frequency Response (CFR) are given and applied for WiFi links. Performance for estimating CIR by TS is assessed and compared to the use of sounding reference signals in WiFi standard. Comparisons revealed a decisive advantage in favour of TS in realistic environment for sequences longer than 2^{14} (=16384) samples.

2.2.3.2 Extension to LTE, UMTS and GSM RATs

Estimations of CIR and CFR can also be performed for other waveforms than WiFi. To do so, suitable reference signals must be defined for each RAT considered in Phylaws project, i.e. LTE (4G), UMTS (3G) and GSM (2G). The CIR and CFR estimations will be used for SKG and AN implementation.

References signals of 2G, 3G and 4G communication systems and their properties are presented in §4 of this deliverable.

3 Design of Tag Signals with random sequences

3.1 Properties of classical PN sequences

As specified in previous deliverable (see D2.4 and D4.1), tag signals are designed upon PN sequences. Among the advantages of such sequences suitable for TS, we can list:

- The high Spreading Factor (SF) to overcome interferences and dominant signals
- The stealth to avoid detection or jamming by an eavesdropper
- The high combinatory to allow multi-user access (such as in CDMA) and to take into account radio channel measurements in its design (such as proposed in the IAS protocol)
- The good correlation properties and time resolution for precise synchronization and accurate channel measurements

§4.3.3 of deliverable D4.1 describes known techniques for PN generation such as Binary Shift Maximal Length (BSML) sequences, Kasami (small and large set) and Gold sequences.

Such techniques are known to provide sets of PN sequences of good correlation properties. Considering sequences of order m' and length $M = 2^{m'} - 1$, Table 1 summarizes the properties of each set.

Table 1: Properties of known PN sequences

Type of PN Sequences	Condition	Maximal cross-correlation (absolute value)	Family size
BSML	$\forall m' \in \mathbf{N}$	1	1
Gold sequences	$m' \in \mathbf{N}$ / mod(m',4) ≠ 0	t(m')	M+2
Small Kasami Set	$m' \in \mathbf{N}$ / mod(m',2) = 0	$\frac{1 + t(m')}{2}$	$\sqrt{M+1}$
Large Kasami Set	$m' \in \mathbf{N}$ / mod(m',2) = 0 & mod(m',4) ≠ 0	t(m')	(M+2)*(1+$\sqrt{M+1}$)

Where t(m') is defined by Eq.3- 1:

$$t(m') = \begin{cases} 1 + 2^{\frac{m'+1}{2}} \text{ if } m' \text{ is odd} \\ 1 + 2^{\frac{m'+2}{2}} \text{ if } m' \text{ is even} \end{cases} \quad \text{Eq.3- 1}$$

Small Kasami and Gold sequences are included in large Kasami set. It can be noted that small Kasami set offers better correlation properties but has less sequences than large Kasami family.

The cross-correlation properties of the classical sets will be compared to the performance obtained by the random design of PN sequences in §3.2.2.3.4.

3.2 Random design of PN sequences

3.2.1 Need and challenge of random PN sequences

3.2.1.1 Need of random designed PN sequences

Classical sets of PN sequences offer poor Transec protection because their determinism can be exploited by Eve to retrieve the original sequences from only few samples [1, 2]. For that reason breaking the determinism is interesting to goal in the design of arbitrary DSS codes. In this case, no prior knowledge can be exploited by Eve to determine the

sequence selected by Alice and Bob. Besides, random PN design gives the possibility to generate code which length is not restricted to a value $2^{m'}$-1 for some m'. This is obviously not the case of Linear Feedback Shift Register (LSFR) based sequences where the length is always linked to m'.

3.2.1.2 Challenges of random design of PN sequences

The main challenges of an arbitrary code design are:

- To keep the number of codes high enough for combinatory and allocation purposes.
- To keep the correlation properties of the set good enough to enhance synchronization and channel measurement accuracy.

We recall that periodic cross-correlation between 2 vectors $x = [x_1,\ldots,x_M]$ and $y = [y_1,\ldots,y_M]$ is calculated by Eq.3- 2.

$$R_{x,y}(i) = x * T^i y' = \sum_{l=1}^{M} x_l * \overline{y_{i+l}} \qquad \text{Eq.3- 2}$$

Where T^i is the operator that circularly shifts to the left a vector by i samples, y' is the conjugate transpose of vector y and $\overline{y_i}$ is the conjugate of scalar y_i.

The problem of generating large set of low-correlated sequences has been widely studied [3, 4, 5]. The upper bound on maximum periodic cross-correlation values that can be reached for a group of N sequences of length M, known as the periodic Welsh bound [3], is given by Eq.3- 3.

$$R_{Welsh} = M * \sqrt{\frac{N-1}{MN-1}} \qquad \text{Eq.3- 3}$$

The maximum normalized periodic cross-correlation, defined as the ratio between the cross-correlation and the length of the sequence is thus obtained by Eq.3- 4.

$$r_{Welsh} = \frac{R_{Welsh}}{M} = \sqrt{\frac{N-1}{MN-1}} \qquad \text{Eq.3- 4}$$

It can be noticed that, when the number of sequences N is fixed, R_{max} is equivalent to $M^{1/2}$ as M tends to infinity, while r_{max} is equivalent to $M^{-1/2}$. Regarding the number N we target (a few thousand at least), a suitable approximation of the Welsh bound is $r_{Welsh} = \sqrt{M}$

Figure 2 represents the evolution of the normalized and the un-normalized Welsh bounds for a set of 10000 sequences in function of the length of the sequence. These curves will be a reference for the design of low-correlated random sequences.

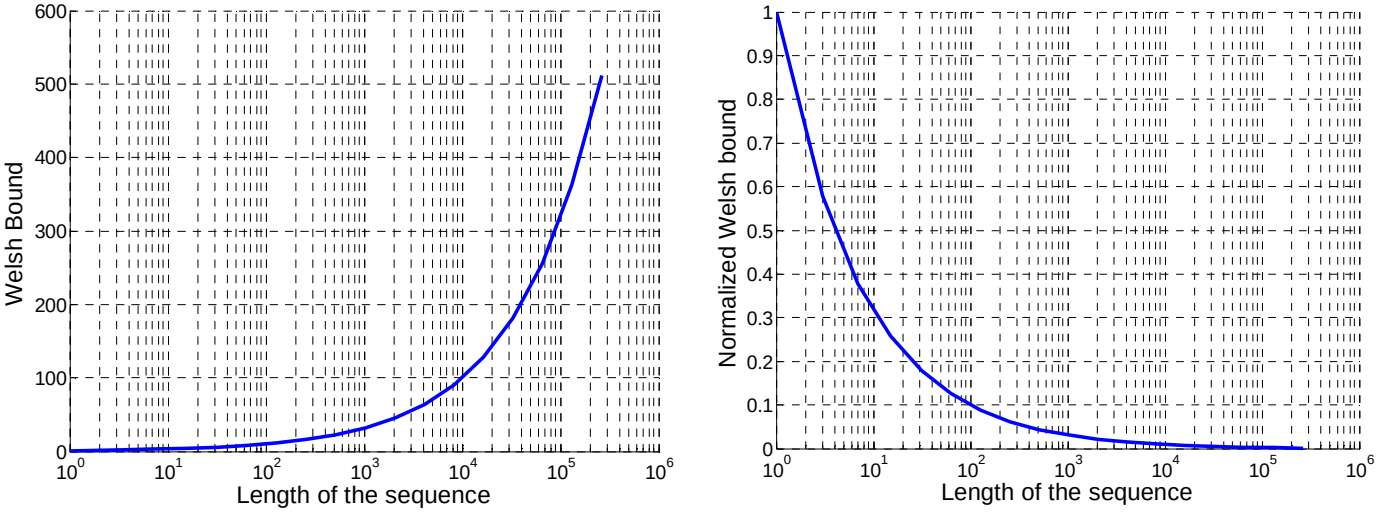

Figure 2: Representation of normalized (right) and un-normalized (left) Welsh bound for a set of 10^4 sequences

From Eq.3- 3 and Eq.3- 4, it appears that for the orders of magnitude considered in this study (concerning the number of sequences (100<N<10000) and the length of the sequences (1000<M<100000), N has a limited impact on the maximum correlation value. For that reason, in the remainder of this deliverable, the size of the set considered for calculating the Welsh bound is often fixed to 10000 sequences.

3.2.2 Simulation process

3.2.2.1 Objective of the study

The objective of this study is to determine whether the generation of random sequences can lead to a set of low-correlated PN codes. To do so, numerous ways to generate random sequences are explained in §3.2.2.2. Moreover, metrics and selection process must be implemented to correctly create the set. These methods are described in §3.2.2.3. Detailed descriptions and results of the simulations are presented in §3.2.2.3.4.

3.2.2.2 Generation of the sequence

3.2.2.2.1 Pseudo Random Number Generator

In this study, two methods are used to generate the random sequences. The first one is based on Pseudo Random Number Generator (PRGN).

The main advantages of such technique are:

- The good randomness properties of the generated sequences due to the intrinsic nature of the PRNG. Among them, we can list:
  - The good autocorrelation properties of the generated codes
  - The uniform distribution of the bits through the code. This property is particularly suitable to build large set of low-correlated sequences and for spectrum efficiency purposes.
- Generation is easy, fast, and well-known.

The main drawback of PRNG is their tendency to be periodic after long time, what is due to the digital processors. This is of course not suitable when generating large amount of sequences in order to select the least correlated ones.

In the following simulations, the PRNG is the standard PRNG provided by Matlab.

3.2.2.2.2 Chaotic Spreading Code

The second technique consists in Chaotic Spreading Code (CSC) generation. The principle of such technique is to use a chaotic process to generate the codes. Detailed information on CSC including algorithms, boundaries and implementations can be found in [6, 7, 8, 9, 10].

In the following simulations, the chaotic map used is the well-known tent map defined by Eq.3- 5.

$$y_{n+1} = \mathrm{a} * \min(y_n, 1 - y_n) \qquad \text{Eq.3- 5}$$

Where $0<x_0<1$ and $1<a<2$ are initial conditions and are randomly chosen.

The binary chaotic code x is generated from sequence y using Eq.3- 6.

$$x_n = \begin{cases} 1 \text{ if } \mathrm{y}_n > 0.5 \\ -1 \text{ if } \mathrm{y}_n < 0.5 \end{cases} \qquad \text{Eq.3- 6}$$

The main advantages of CSC are:

- The simplicity of implementation, even in digital hardware [11, 12].
- The sensitivity to initial conditions that prevents periodicity of the generation process (unlike PRNG) and allows a wider diversity of code.
- The non-periodicity of the generated sequences

However, contrary to PRNG, this generation technique does not guarantee good correlation properties even for autocorrelation. As an example, Figure 3 shows the autocorrelation of 10^4 samples sequences generated by a PRGN (in blue) and by a chaotic tent map (in red) for different initials conditions.

The top left and right hand autocorrelation figures show the high correlation level that can be obtained from a chaotic process. However, the bottom right hand figure demonstrates that some initial conditions (a, x_0) can lead to similar values than PRNG. This observation suggests that CSC process can be longer than PRNG to find small or medium set of low-correlated sequences. However, its wider diversity could be beneficial for finding very large set of such sequences. This conjecture will be verified in the following simulations.

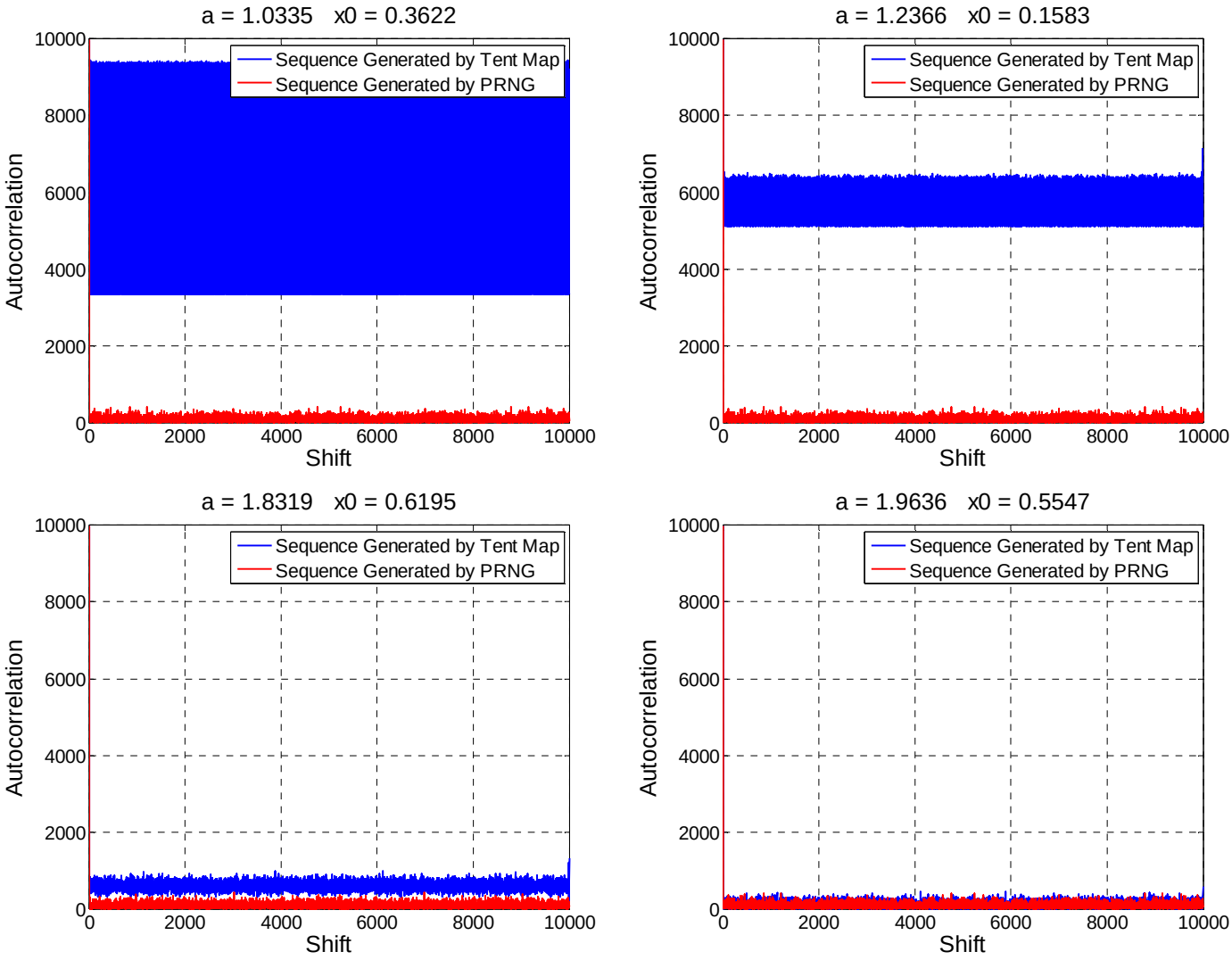

Figure 3: Autocorrelation of PRGN and chaotic sequences for different initial conditions

3.2.2.3 Metrics

3.2.2.3.1 Correlation metrics

In order to assess the correlation level in a generated set, correlation metrics are defined. They are expressed as a percentage of the length of the sequence (corresponding to the value of the autocorrelation peak).

In a given set $\boldsymbol{\sigma}$ of N sequences, the maximum correlation factor of the set, r_{max}, is given by Eq.3- 7. It corresponds to the maximal value of cross-correlation or autocorrelation between two sequences within the set. r_{max} indicates thus the worst value of the generated family. The value of r_{max} is limited by Welsh bound (see Eq.3- 4).

$$r_{max} = \frac{100}{\mathrm{M}} * \max_{x,y \in \boldsymbol{\sigma}}(R_{x,y}) \qquad \text{Eq.3- 7}$$

Where $R_{x,y}(i)$ is defined by Eq.3- 2 with i≠0 when x=y.

In a given set **σ** of N sequences, the minimum correlation factor of the set, r_{min}, is defined by Eq.3- 8. It corresponds to the minimal value of cross-correlation or autocorrelation between two sequences within the set. r_{min} indicates thus the best value of the generated family.

$$\mathrm{r}_{min} = \frac{100}{\mathrm{M}} * \min_{x,y \in \boldsymbol{\sigma}}(\mathrm{R}_{\mathrm{x,y}}) \qquad \text{Eq.3- 8}$$

Where $R_{x,y}(i)$ is defined by Eq.3- 2 with i≠0 when x=y.

In a given set **σ** of N sequences, the average correlation factor of the set, r_{mean}, is defined by Eq.3- 9 where E|.| denotes the expectation operator. It corresponds to the average value of cross-correlation or autocorrelation between two sequences within the set. r_{mean} indicates thus the global performance of the generated family

$$\mathrm{r}_{mean} = \frac{100}{\mathrm{M}} * \mathrm{E}_{x,y \in \boldsymbol{\sigma}}\left|\mathrm{R}_{\mathrm{x,y}}\right| \qquad \text{Eq.3- 9}$$

Where $R_{x,y}(i)$ is defined by Eq.3- 2 with i≠0 when x=y.

As an example, Figure 4 illustrates the evolution of the maximal correlation factor of Gold set, large and small Kasami sets and Welsh bound (for a set of 10^4 sequences). When large Kasami exists (m' even and not divisible by 4), its correlation factor is similar to Gold bound, as indicated in Table 1.

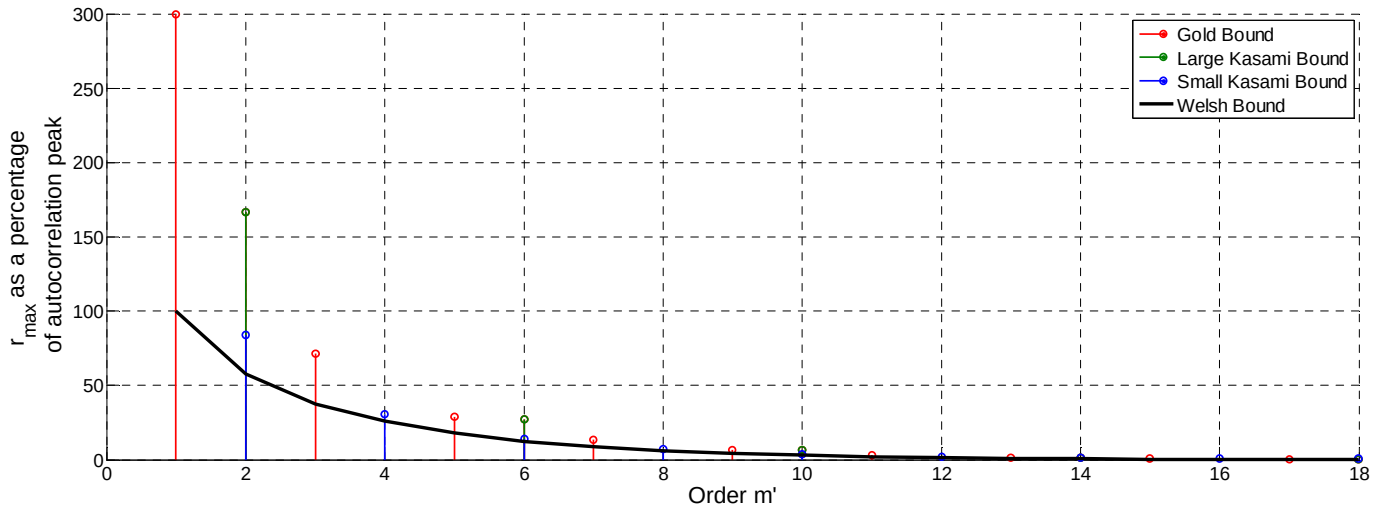

Figure 4: Maximum correlation factor of classical sets compared to Welsh bound in function of the order m'

The three correlation factors defined above can also be compared to the relevant Welsh bound $r_{Welsh}(M,N)$ where M is the length of the sequence and N the number of sequences in the set. In Eq.3- 10 we define the maximum excess to Welsh bound factor. In this case, the set will exceed the Welsh bound at most $x_{max/Welsh}$ times.

$$x_{max/Welsh} = \frac{r_{max}}{100 * r_{Welsh}} \qquad \text{Eq.3- 10}$$

where $R_{x,y}(i)$ is defined by Eq.3- 2 with i≠0 when x=y and r_{Welsh} is defined by Eq.3- 4.

Similarly, the minimum excess to Welsh bound factor and the average excess to Welsh bound factor can also be defined by respectively replacing r_{max} by r_{min} and r_{mean} in Eq.3- 10.

As an example, Figure 5 illustrates the evolution of the maximum excess to Welsh bound of Gold, large Kasami and small Kasami sets. When large Kasami exists (m' even and not divisible by 4), its correlation factor is similar to Gold

bound, as indicated in Table 1. It can be observed that small Kasami sets come closer to Welsh bound when the order increases, as its excess to Welsh bound factor tends to 1. Odd orders of Gold set tend to 1.42. Even orders of Gold set, and large Kasami set exceed the Welsh bound by a factor 2 with large order.

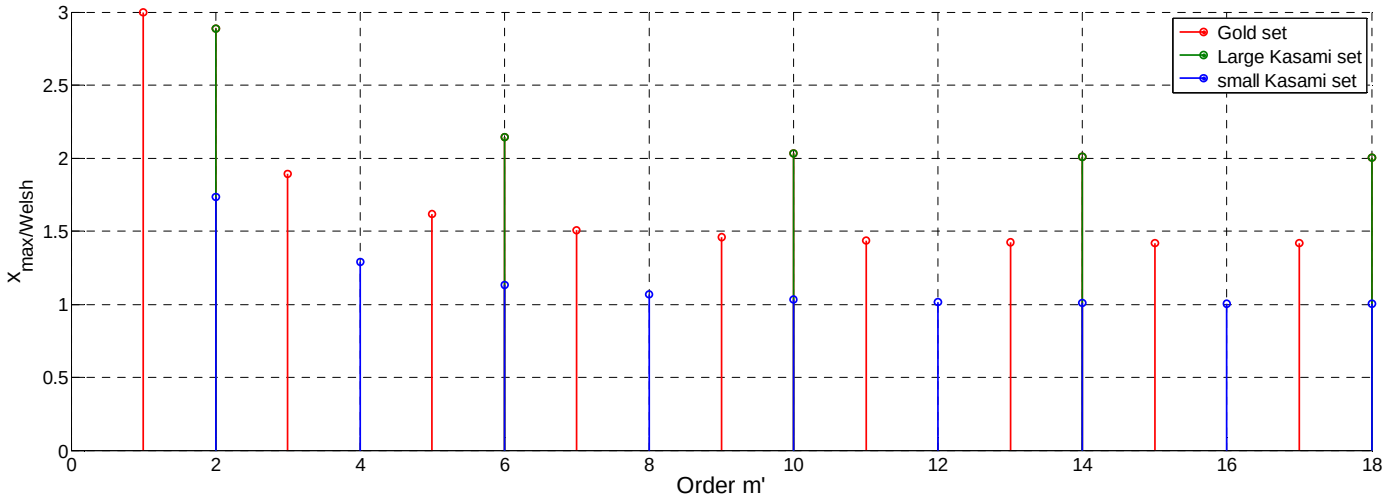

Figure 5: Excess to Welsh bound in function of the order m' for classical sets

The factor $x_{max/Welsh}$ enables to evaluate the level of the side-lobes of the generated sequences. Indeed, for the ideal Welsh bound case, the level of the side-lobe is -10*log(R_{Welsh}). With an excess to Welsh bound factor, the level of the side-lobes is thus given by Eq.3- 11.

$$SL = -10 * \log(R_{Welsh}) + 10 * \log(x_{max/Welsh}) \qquad \text{Eq.3- 11}$$

Table 2 gives the level of side-lobes for different length of codes and excess to Welsh bound factors.

Table 2: Side lobe level for different excess to Welsh bound and length of code

Length (in samples)	Side lobe for Welsh Bound (in dB)	Excess to Welsh bound	Side lobe Level (in dB)
1000	-15.0	2	-12.0
		4	-9.0
		6	-7.2
10000	-20.0	2	-17.0
		4	-14.0
		6	-12.2
100000	-25.0	2	-22.0
		4	-19.0
		6	-17.2

3.2.2.3.2 Technique for fast computation of the correlation factor

Computation of correlation factor $R_{x,y}$ as defined by Eq.3- 2 is quite long, especially when dealing with large sets of sequences.

Time can be saved by taking advantage of the properties of Fourier transform. A faster way to compute $R_{x,y}$ is given by Eq.3- 12.

$$R_{x,y} = \max\left(\mathrm{IFFT}\left(\mathrm{FFT}(x) .* \overline{\mathrm{FFT}(y)}\right)\right) \quad \text{Eq.3- 12}$$

Where x.*y is defined by Eq.3- 13.

$$x.*y = \begin{pmatrix} x_1 * y_1 \\ \vdots \\ x_M * y_M \end{pmatrix} \quad \text{Eq.3- 13}$$

3.2.2.3.3 Autocorrelation specific factor

A further step in the selection process of the codes could specifically consider the value of $R_{x,x}$ around the main peak. Indeed, low side lobes of autocorrelation function around the synchronization position would improve detection and channel measurements.

Such behaviour could be evaluated by the metric given by Eq.3- 14

$$AC = \frac{100}{M} * \max_{i=1,2,n-2,n-1}\left(R_{x,x}(i)\right) \quad \text{Eq.3- 14}$$

However, for simplicity reasons, this metric is not considered in the following simulations.

3.2.2.3.4 Spectrum efficiency factor

Metrics for the evaluation of the randomness distribution and the spectrum uniformity of the generated codes could also be implemented. In practice a verification of the balance of the chips distribution of the DSS codes is enough. Additional test such as NIST tests [13] could be considered to confirm the randomness quality of the DSS codes sets. However, for simplicity reasons, this metric is not considered in the following simulations.

3.2.3 Simulations results

3.2.3.1 Objectives and parameters

The main motivations of the simulations are the following:

- What kind of correlation factor can be obtained when generating large sets of random sequences?
- How long does the construction of low-correlated random codes take?

Simulations presented in §3.2.3.2 deals with the first question while the second question is evaluated by §3.2.3.3

The aim of the study is not to generate the definitive set of tag signals sequences but to assess the feasibility of the presented approach and to determine what kind of performance can be expected.

In the following, we evaluate the capabilities of the generation process with the following parameters:

- Set contains 100, 1000, or 10000 sequences
- Length of the sequences varies from 2512 samples (SF = 34 dB) to 100000 samples (SF = 50 dB).

3.2.3.2 Correlation performance of randomly generated sequences

3.2.3.2.1 Simulation protocol

The aim of this first simulation is to define performance boundaries when generating sets of random sequences without any selection process. The objective is to determine correlation threshold that can help in a further selection algorithm (to be set in §3.2.3.3).

To do so, after the generation of N sequences, the following parameters are calculated:

- r_{max}, as defined by Eq.3- 7. This coefficient corresponds to the upper limit of a threshold.
- r_{min}, as defined by Eq.3- 8. This coefficient reflects the lower limit that can be expected by a selection process of the code.
- r_{mean}, as defined by Eq.3- 9 This coefficient is an interesting threshold for the selection process.

Observations are averaged over a number of runs depending on the number of sequences in the set:

- 10 runs are drawn for sets of 10000 sequences
- 100 runs are drawn for sets of 1000 sequences
- 1000 runs are drawn for sets of 100 sequences

3.2.3.2.2 Results for PRNG

Results for generation using PRGN is given by Figure 6 in function of the SF and of the number of samples in the code.

Information about the curves is given below:

- The solid line represents r_{mean}
- The upper dashed line represents r_{max}
- The lower dashed line represents r_{min}
- The dotted and circled line represents the corresponding Welsh bound. Note that the Welsh bound curves are almost superimposed as the size of the set has a limited impact on the limit (see § 3.2.1.1). That is why only the Welsh bound for 10000 sequences can be observed.

For complexity reasons, correlation values for sets of 10000 sequences longer than $6.3*10^4$ samples were not calculated.

From Figure 6 it can be first observed that the number of sequences in the set as no impact on the mean correlation factor (the different solid curves are almost superimposed). In that way, the evolution of r_{mean} is similar to the Welsh bound (see §3.2.1.2). However, the minimal and maximal correlation factor depends on the number of codes in the set. The larger the family, the wider is the difference between r_{min} and r_{max}.

Moreover, the difference between the minimum correlation factors for different family size is thinner than the difference between maximum correlation factors. Thus, for SF = 42 dB, the difference between the minimum correlation factors for 100 sequences and 10000 sequences is 0.14% whereas the difference between maximum correlation factor is 1.25%. This suggests that the r_{min} calculated with this simulation are very close to the lower boundaries of correlation level that can be obtained using PRNG.

Besides, one can notice that r_{mean} is closer to r_{min} than to r_{max}. Indeed, for SF = 42 dB and for 10000 sequences in the set, r_{mean} - r_{min} = 0.72% whereas r_{mean} - r_{min} = 2.57%. This means that the majority of the correlation factors are under $r_{mean,\sigma}$. This suggests that $r_{mean,\sigma}$ could be a good upper threshold for a selection process.

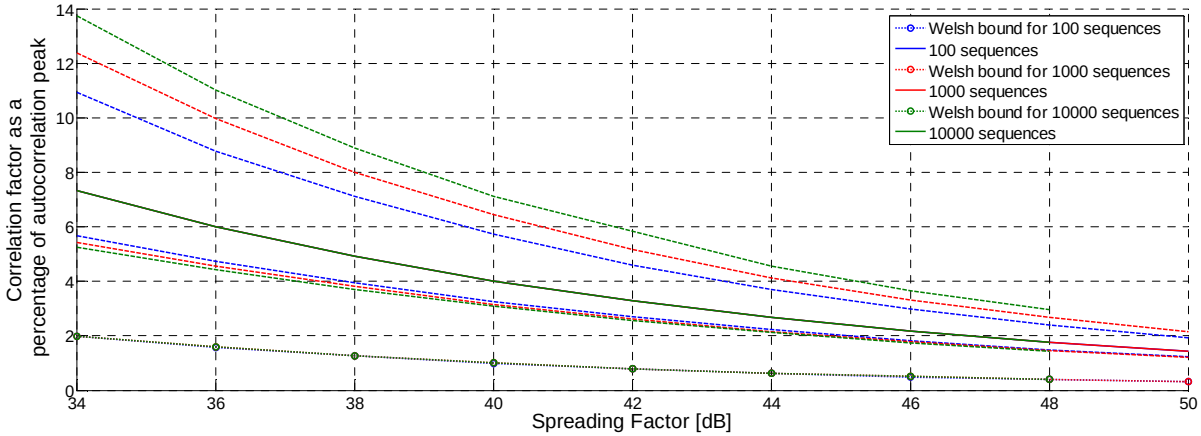

Figure 6: Correlation factors for sets of sequences using PRNG in function of the SF

Figure 7 represents the evolution of the different excess to Welsh bound factors in function of the SF.

Information about the curves is given below:

- The solid line represents $x_{mean/Welsh}$
- The upper dashed line represents $x_{max/Welsh}$
- The lower dashed line represents $x_{min/Welsh}$

For complexity reasons, correlation values for sets of 10000 sequences with SF > 48 dB were not calculated.

It can be first noticed that excess to Welsh bound factors increases with the length of sequences. Moreover, the curves do not seem to reach an upper limit which is the opposite behaviour to classical PN families as observed in Figure 5. Furthermore, correlation performance of PRNG sequences is slightly worse than for classical PN sequences. Indeed, for SF = 42 dB, what corresponds roughly to an order m' = 14, small Kasami set has an maximum excess to Welsh bound factor close to 1 while large Kasami and Gold sets have a factor close to 2. Without any selection process, the best value obtained with PRNG is 3.3 and can go up to 7.3. Compared to Gold set, the increase in terms of side lobes is between 1.8 dB and 5.5 dB (see Table 2).

Nevertheless, similarly to correlation factors, the size of the family does not affect much the mean excess to Welsh bound factor even if a slight decrease can be noted when the set is larger. Moreover, the difference between minimum and maximum excess to Welsh bound increases with the size of the set.

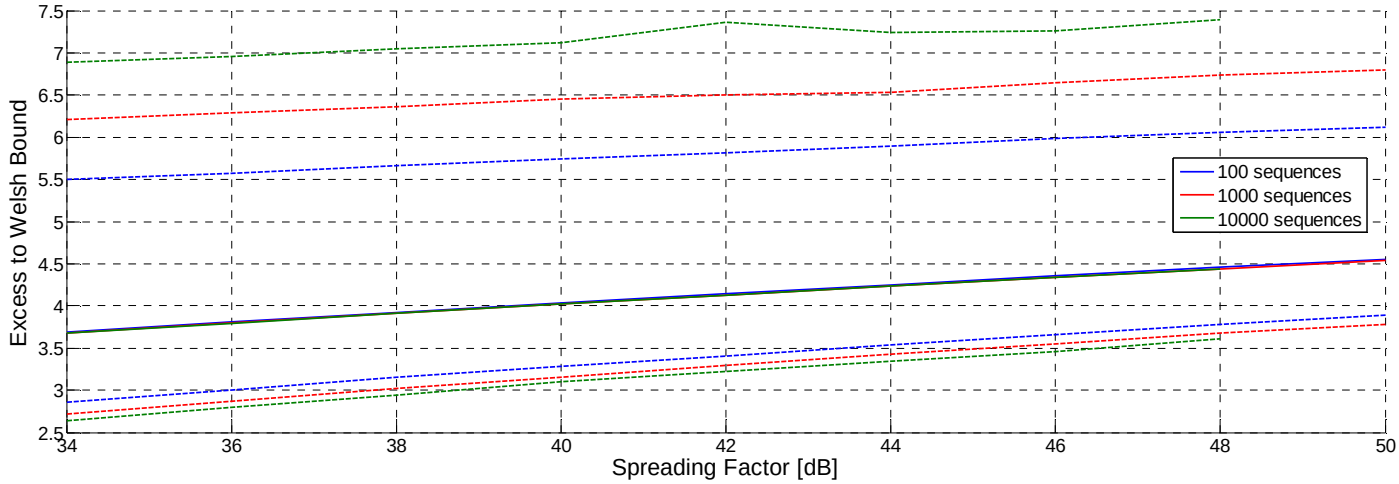

Figure 7: Excess to Welsh bound for sets of sequences using PRNG in function of the SF

3.2.3.2.3 Results for CSC

Results for CSC generation in terms of correlation and excess to Welsh bound factors are given by Figure 8. Only sets of 100 sequences are considered with SF from 34 to 42 dB. As CSC generation does not provide intrinsically low-correlated codes, maximum correlation factors can reach very high value, up to 95% of the autocorrelation peak. Furthermore, for SF = 42 dB some sequences in the set can exceed by 120 times the Welsh bound. This observation highlights the need for a use of extra selection process to build the set with this method.

The average correlation and excess to Welsh bound factors are superior to PRNG method. This suggests that the selection process can be longer with this method.

The minimum correlation and excess to Welsh bound factors of CSC sequences can reach lower values than for PRNG method. Indeed, for SF = 38 dB, $x_{min/Welsh}$ = 2.67 for CSC whereas $x_{min/Welsh}$ = 3.17 for PRNG. This suggests that for such SF, CSC can provide better set of codes in terms of correlation despite a longer generation times. However, this advantage decreases with longer sequences. However, from our simulations, this advantage seems to decrease with longer sequences. Thus, for SF = 42 dB, $x_{min/Welsh}$ = 3.39 for CSC while $x_{min/Welsh}$ = 3.42 for PRNG.

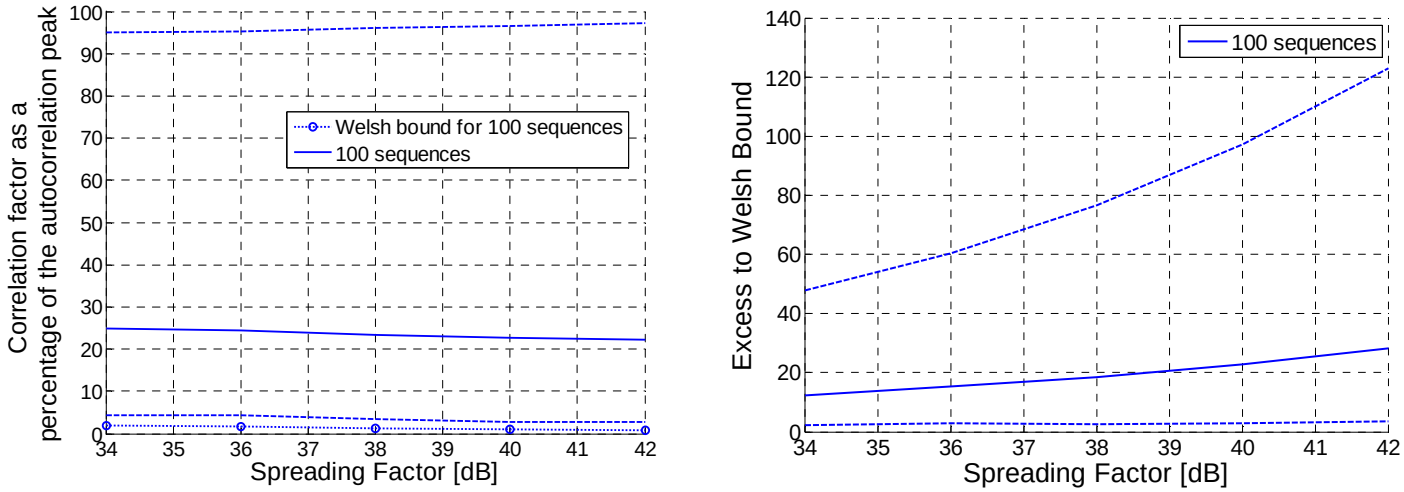

Figure 8: Correlation and excess to Welsh bound factors for a CSC generation

3.2.3.2.4 Conclusion on correlation performance of randomly generated sequences

This first study has highlighted the following points:

- Generation of random sets based on PRNG provides quite good results in terms of correlation even if performance is decreased compared to classical PN sequences families. Improvements can certainly be brought by a selection process applied on the generated codes. According to the results, we cannot expect to have a maximum excess to Welsh bound lower than 4 even with the selection process. The sets with $x_{max/Welsh}$ = 7 seems to be easy to obtain.
- Generation of random sets based on CSC needs selection process to procure low-correlated codes. No a priori threshold can be deduced from previous simulations. As a consequence, the same thresholds will be set for both generation techniques.

3.2.3.3 Complexity for generating sets of low-correlated random sequences

3.2.3.3.1 Simulation protocol

The aim of this second simulation is to evaluate the complexity of generating sets of low-correlated random sequences. The main objective is to determine how many sequences have to be generated to obtain a set with a given maximum excess to Welsh bound.

For simplicity, only sequences of SF = 42 dB are considered. This corresponds to roughly 16 000 samples. This value is selected according to studies performed in deliverable D4.1 where this length appeared to be suitable for tag signal.

Figure 9 presents the selection algorithm implemented to estimate the complexity. The first sequence of the family is always the code [-1,1,-1,1,…,-1,1]. This sequence has a perfect autocorrelation function and a balanced number of -1 and 1. This sequence helps thus to select the best random code regarding these two conditions. Of course, this first sequence is not counted up in the size of the set.

The following parameters are kept in memory:

- i_{Run} which is the number of generated sequences since the beginning of the simulation
- N which is the number of sequences in the current set

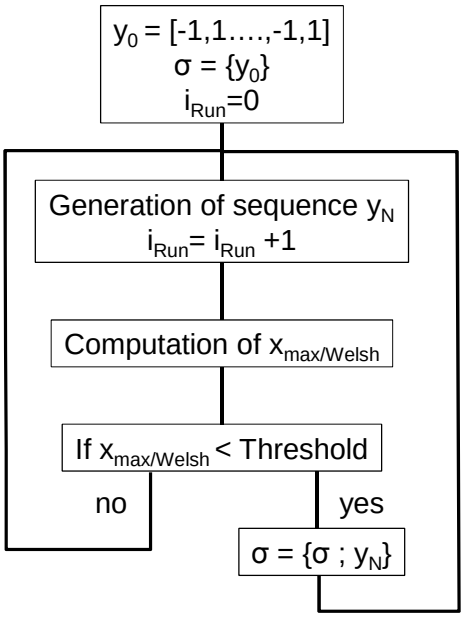

Figure 9: Simulation algorithm for evaluating the generation complexity

3.2.3.3.2 Evaluation of computational complexity

For computation time reason, the generation of low-correlated sets was most often stopped after only few days. Nevertheless, some attempts were performed over several weeks.

Finally, these simulations were enough to study the computational complexity and to forecast the time required for generating larger sets of non-deterministic tag signals (required for a better resilience to advanced attackers).

From our simulations, we elaborated a simplified and model that proposes a relation between the number of sequences to generate, y, in order to obtain a set of x low-correlated sequences in the set. The relationship between x and y is given by Eq.3- 15, where λ and μ are scale parameters. This model includes an exponential relationship for large values of x and y, and a linear relationship for low values of x and y. The initial value is thus y=1 when x=1. Note that in the protocol described Figure 9, the number of sequences to generate, y, is accessible through the parameter i_{Run}.

$$y(x) = \mu(x-1) + e^{\lambda x} \qquad \text{Eq.3- 15}$$

The asymptotic behaviours of Eq.3- 15 are:

- λ is positive and close to 0. μ is close to 1. λ and μ depend on the threshold concerning the correlation factor (see § 3.2.2.3).
- λ drives the exponential behaviour of the model ($y(x) \sim e^{\lambda x} - 1$) for large values of x and y
- μ drives the linear behaviour of the model ($y(x) \sim 1 + \mu(x-1)$) for low values of x and y, with an initial value y=1 for when x = 1.

The parameters λ and μ can be estimated from the simulations with non-linear regression techniques using weighting and 2D optimization.

In such iterative processing, (λ,μ) can be initiated by the couple (λ_0,μ_0):

- The scale parameter μ can be initiated by considering the linear progression of y(x) for the first values of x: $\mu_0 = \partial_x(\mathrm{y(x)})$
- The scale parameter λ can be initiated by considering the large values of x and y by calculating Eq.3- 16.

$$\lambda_0 = \lim_{x\to\infty} \frac{\ln(\mathrm{y(x)})}{\mathrm{x}} \qquad \text{Eq.3- 16}$$

In the asymptotical part of the model (when $\mathrm{x} \to \infty$), to get a set of x+1 sequences from a set of x sequences, Δy(x) additional sequences must be generated such as defined by Eq.3- 17.

$$\Delta\mathrm{y(x)} \sim \mathrm{e}^{\lambda(x+1)} - \mathrm{e}^{\lambda x} \sim \left(e^{\lambda} - 1\right) * \mathrm{e}^{\lambda x} \sim \left(e^{\lambda} - 1\right) * \mathrm{y(x)} \qquad \text{Eq.3- 17}$$

One additional sequence is serially compared to the x sequences that are already selected in the set. If the correlation level with one of these sequences is above the threshold, the additional sequence is discarded and a new one is generated. A new candidate sequence is added to the set only if it has successfully been compared to every current sequence.

Of course, many candidate sequences are rejected after a reduced number of comparisons. The average number of such comparisons is noted z(x) in the following. To study this item we propose to model z(x) with a proportional law.

$$\mathrm{z(x)} \sim \alpha.\mathrm{x} \;\; ; \;\; 0 < \alpha \leq 1$$

Then, the number c_α(x) of comparisons to obtain a set of x+1 sequences from a set of x sequences is given by Eq.3-18.

$$c_\alpha(\mathrm{x}) \sim \Delta\mathrm{y(x)} \,.\, \alpha.\mathrm{x} \sim \alpha.\left(e^{\lambda} - 1\right).\mathrm{y(x)}.x \qquad \text{Eq.3- 18}$$

Thus, the total number of comparisons to make to generate a set with X sequences is given by Eq.3- 19.

$$\mathbf{C}_\alpha(\mathbf{X}) \sim \sum c(x) \sim \alpha.\left(e^{\lambda} - 1\right)\sum_{x=1}^{X} x.\mathrm{e}^{\lambda x} \qquad \text{Eq.3- 19}$$

The cross-correlation and the comparison algorithms are performed with Matlab 2014. We used a stand-alone DELL PRECISION T1650 computer [14] with the following features, which CPU was entirely dedicated to our simulations:

- Processor: Intel Core i7 (3.7 GHz)
- Chipset: Intel C216
- Operating systems: Genuine Windows 7 Professional 64-Bit
- Memory RAM is 32GB Error Correction Coding at Frequency 1600MHz
- Storage controller: Intel Rapid Storage Controller 11.0 supporting SATA 6Gb/s and host based RAID 0/1/5/10

With this computer, the average comparison time between 2 sequences (computation of cyclic cross-correlation and comparison to threshold) was measured to $\tau = 1.3*10^{-3}$ s.

Then, from our simulations, it appears that the proportional law for z(x) matches reasonably with the computation when choosing α = ½.

Finally, the computation time required for generating larger sets can be extrapolated by Eq.3- 20.

$$\mathrm{T(X)} = \tau.\mathbf{C}_{\frac{\mathbf{1}}{\mathbf{2}}}(\mathbf{X}) \qquad \text{Eq.3- 20}$$

3.2.3.3.3 Results for PRNG

3.2.3.3.3.1 $x_{max/Welsh}$ = 7 and $x_{max/Welsh}$ = 6 – case of a limited number x of selected sequences

Figure 10 shows the number y(x) of sequences to generate in order to obtain a set with $x_{max/Welsh}$ = 6 (right) or 7 (left). For this first simulation, the generation process was stopped when the set reached x=10 000 sequences. It can be observed that this simulation case entirely corresponds to the linear part of the complexity model obtaining such set is quite fast because almost every run provides a code below the threshold. In particular, 10 000 runs are necessary when $x_{max/Welsh}$ = 7 (which corresponds exactly to μ=1, λ=0 in our model) while 11 621 runs are required for $x_{max/Welsh}$ = 6. (which corresponds to μ=1.0882, λ=8.77 10^{-4} in our model). Such low complexity was already expected from §3.2.3.2.2 Figure 7.

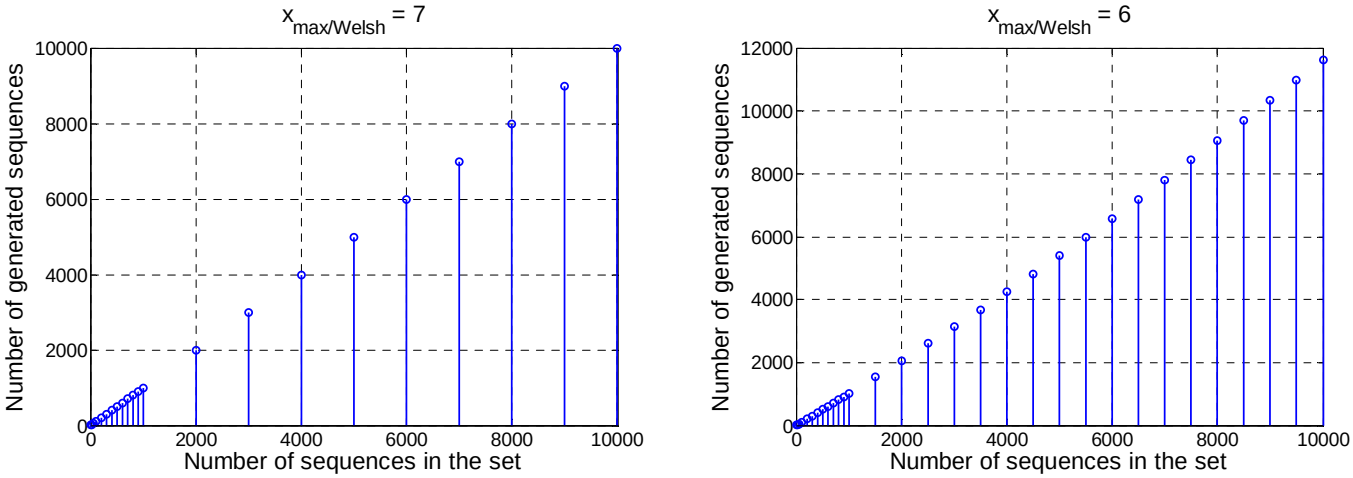

Figure 10: Number of generated sequences for a set with $x_{max/Welsh}$ = 7 (left) and $x_{max/Welsh}$ = 6 (right) - Case of PRNG – linear scale for y-axis

3.2.3.3.3.2 $x_{max/Welsh}$ = 7 and $x_{max/Welsh}$ = 6 – Unlimited number x of selected sequences

The results presented below were obtained for long simulations over three weeks. In this case, the number of sequences in the set was not limited to 10 000 contrary to §3.2.3.3.3.1, but constraints occur only on computing duration (limited here to three weeks).

Figure 11 shows the number of sequences to generate in order to obtain a set with $x_{max/Welsh}$ = 6 (right) or 7 (left).

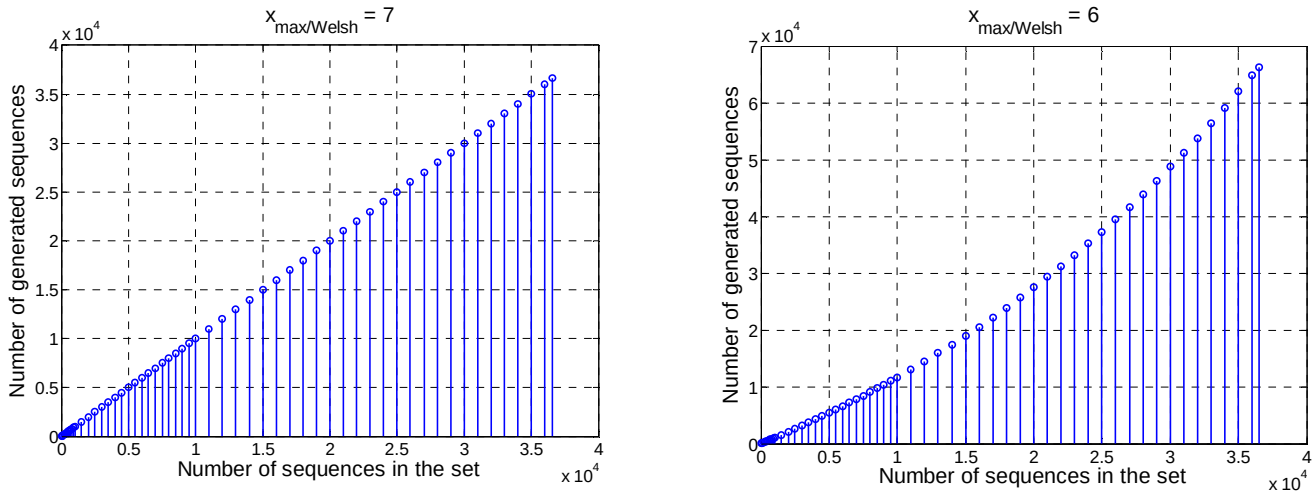

Figure 11: Number of generated sequences for a set with $x_{max/Welsh}$ = 7 (left) and $x_{max/Welsh}$ = 6 (right) – Case of PRNG – linear scale for y-axis

First of all, the seeds of the PRNG being different in every simulation (in order to avoid dependence on the initial value), it is observed that the computational complexity of the first runs of these long simulations follows the same behaviour described in the previous paragraph. In particular

- with $x_{max/Welsh}$ = 7, 10 002 sequences were generated to obtain a set of 10 000 sequences (compared to 10 000 runs in the previous paragraph).
- With $x_{max/Welsh}$ = 6, 11 687 generations were necessary to obtain such large set (compared to 11 621 in the previous paragraph).

Case of $x_{max/Welsh}$ = 7: It can be observed that obtaining such large set with $x_{max/Welsh}$ = 7 is quite easy because almost every run provides a code below the threshold value. In particular, only 36 027 runs are necessary to obtain a set of 36 000 sequences. This simulation case mainly corresponds to the linear part of the complexity model with μ very close to 1 (1.0001) and λ very close to 0 ($8.8*10^{-5}$). In practice, the can extrapolate from these results that when considering with $x_{max/Welsh}$ = 7, advanced computer technology and optimized simulation algorithm, sets of several millions of full arbitrary Tag Signals should be achieved.

Case of $x_{max/Welsh}$ = 6: For $x_{max/Welsh}$ = 6, the situation is quite different because the selection process takes significantly longer: Three weeks computation and 64 865 runs allowed to obtain a set of 36 000 sequences. The shape of the curve seems to be asymptotically exponential what justifies the model derived in § 3.2.3.3.2. This case mixes the linear relationship of the computational complexity model up for x < 10 000, and the exponential relationship of the model for x > 10 000.

The value of the scale parameter λ for $x_{max/Welsh}$ = 6 can be assessed by considering the asymptotic behaviour of Figure 12 for large x values. Taking into account the value $\lambda = 3.1*10^{-4}$ is coherent with the low computationnal complexity observed in the right part of Figure 11.

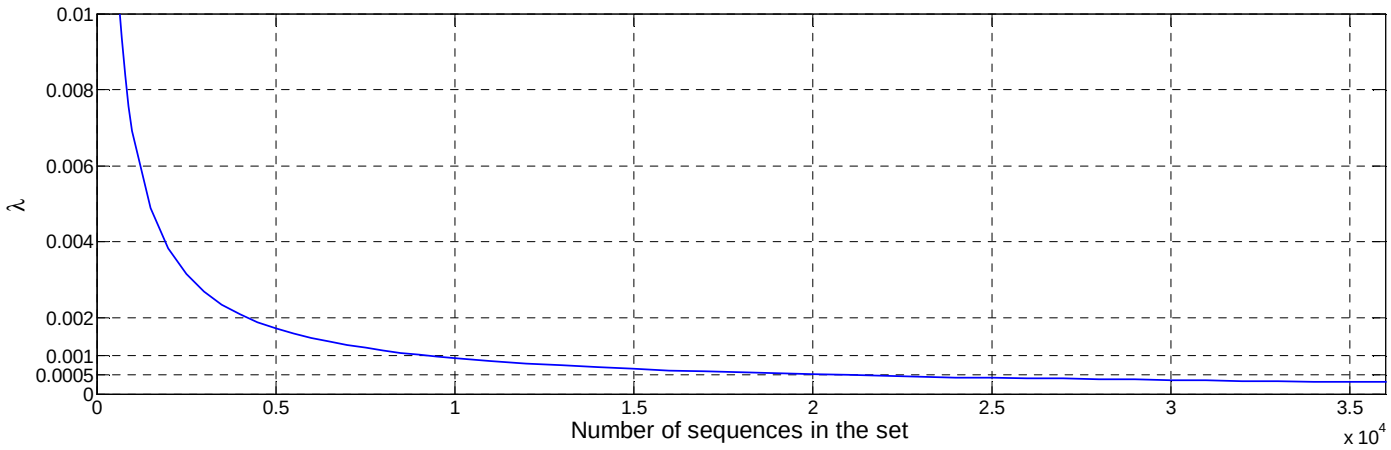

Figure 12: Evolution of the scale parameter λ in function of the size of the set for $x_{max/Welsh}$ = 6 – Case of PRNG

With $\lambda = 3.1*10^{-4}$ and $\tau = 1.3*10^{-3}$ the computational time provided by Eq.3- 20 is 20.5 days. This value is very close to the real time of our simulation (around 20 days), what allows extrapolation of a maximum evaluation of the computational complexity with the model derived in §3.2.3.3.2.

With such value of λ, Table 3 forecasts the maximal order magnitude of computation duration to obtain larger sets with $x_{max/Welsh}$ = 6. Sets of 50 000 sequences seem to correspond to a practical limit, when considering the highest computer technology and optimized simulation algorithm.

Table 3: Measurement and *extrapolation* of computational complexity for $x_{max/Welsh}$ = 6 ($\lambda=3.1*10^{-4}$) - Case of PRNG

Size of the set	36000	*40 000*	*50 000*	*60 000*
Number of generation	64 865	*242 801*	*$5.39*10^{6}$*	*$1.20*10^{8}$*
Time required	≈ 20.5 days	*67 days*	*5.2 years*	*140 years*

3.2.3.3.3.3 $x_{max/Welsh} = 5$

Figure 13 shows the number of sequences to generate to obtain a set with $x_{max/Welsh} = 5$. These results reveal the high computational complexity of generating a set with such a reduced correlation level. Indeed, even at low x values, 159 generations are necessary to obtain a set of 100 sequences. This number rises to 9604 for a set of 500 sequences. Finally, more than $7*10^5$ runs must be performed to obtain a set of 1000 sequences.

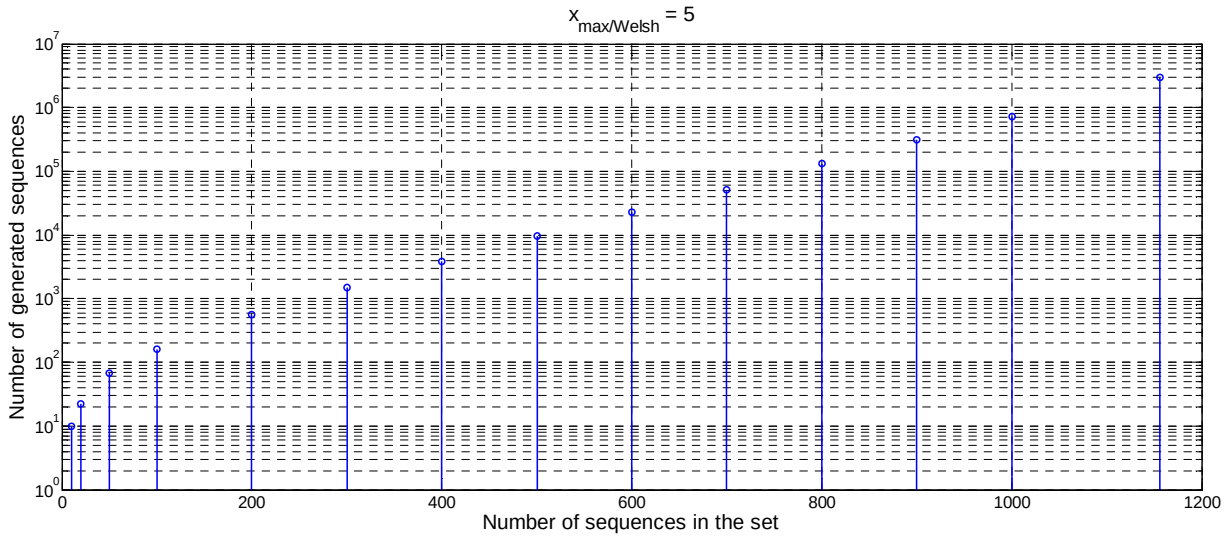

Figure 13: Number of generated sequences to obtain a set with $x_{max/Welsh} = 5$ – Case of PRNG – logarithmic scale for y-axis

The asymptotic behaviour of in Figure 14 (for large number x of selected sequences in the set) shows that the scale parameter λ tends to $1.29*10^{-2}$. This value can be used to estimate the order of magnitude of maximal complexity for the generation of larger set (see Table 4). Roughly, such results show that with such exponential increase, sets larger than 1500 codes can hardly be generated.

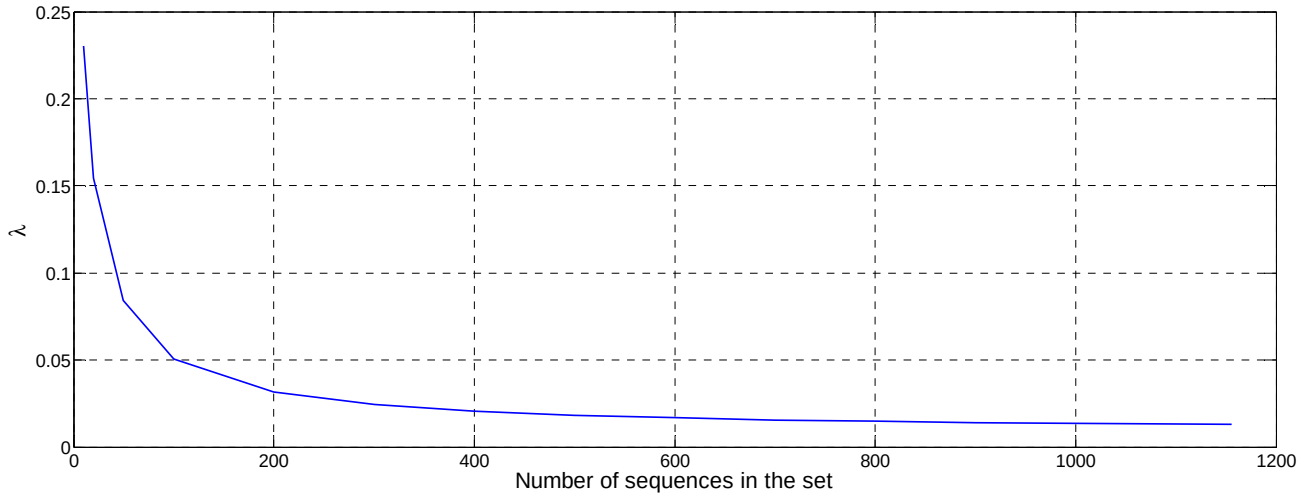

Figure 14: Evolution of the scale parameter λ in function of the size of the set for $x_{max/Welsh} = 5$ – Case of PRNG

Table 4: Measurement and *extrapolation* of computational complexity for $x_{max/Welsh} = 5$ ($\lambda=1.29*10^{-2}$) – Case of PRNG

Size of the set	1000	*1 300*	*1 400*	*1 500*	*1 600*
Number of generation	$7*10^5$	*$1.92*10^7$*	*$6.97*10^7$*	*$2.53*10^8$*	*$9.20*10^8$*
Time required	≈ 3.5 days	*176 days*	*1.9 year*	*7.4 years*	*29 years*

3.2.3.3.4 $x_{max/Welsh} = 4$

Figure 15 shows the number of sequences to generate to obtain a set with $x_{max/Welsh} = 4$. In this case, the order of magnitude of the computational complexity is much higher than for the previously studied case. In particular, 15 attempts are required to find two correct sequences. The number of attempts soars to 17 282 for a set of 10 selected sequences and skyrockets to 9.8 million to obtain only 17 low-correlated codes.

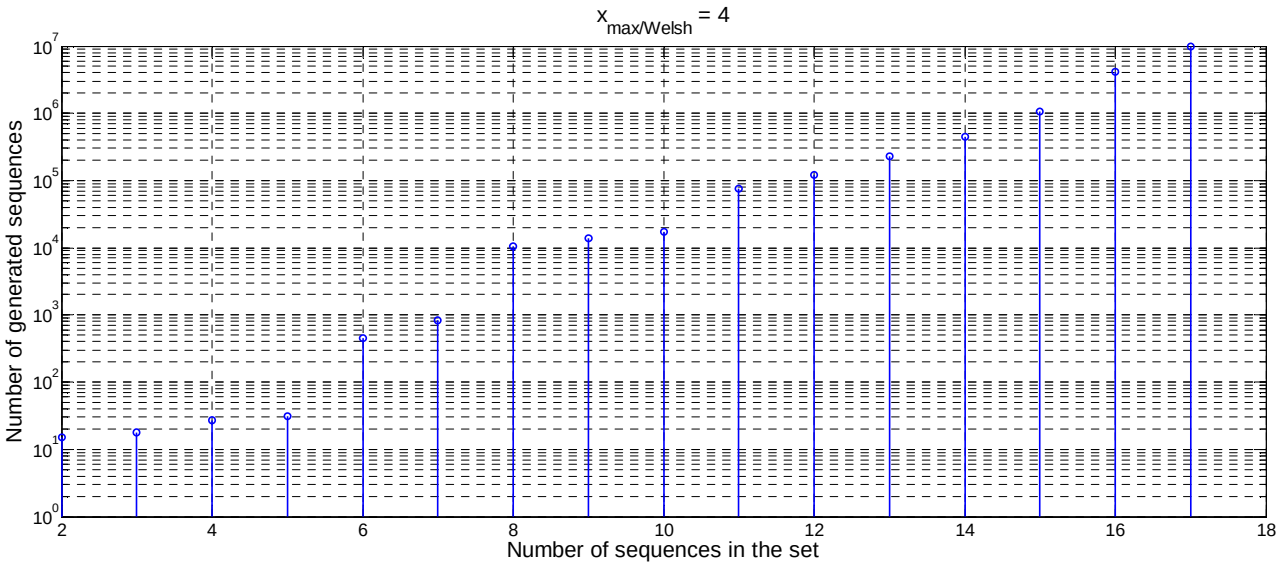

Figure 15: Number of generated sequences to obtain a set with $x_{max/Welsh} = 4$ – Case of PRNG – logarithmic scale for y-axis axis

The behaviour of Figure 16 for the highest (even limited) number of sequences in the set indicates possible interval for the value of the scale parameter λ in the case $x_{max/Welsh} = 4$. The final number of selected sequences being much smaller, the tendency is less clear than above. However it can reasonably be assumed that λ = 0.94, especially considering the 4 last values. From this hypothesis, table 5 conjectures the complexity for larger sets. In practice, with such exponential increase, only a few ten codes could be generated even with the highest computing technology.

Figure 16: Evolution of the scale parameter λ in function of the size of the set for $x_{max/Welsh} = 4$ – Case of PRNG

Table 5: Measurement and *extrapolation* of computational complexity for $x_{max/Welsh} = 4$ (λ=0.94) - Case PRNG

Size of the set	17	*20*	*22*	*24*	*26*
Number of generation	$9.81*10^6$	*$1.46*10^8$*	*$9.58 * 10^8$*	*$6.28*10^9$*	*$4.11*10^{10}$*
Time required	≈ 3.5 days	*20.2 days*	*144 days*	*2.9 years*	*20.6 years*

3.2.3.3.4 Results for CSC

3.2.3.3.4.1 $x_{max/Welsh}$ = 6 and $x_{max/Welsh}$ = 7

Figure 17 shows the number of sequences to generate in order to obtain a set with $x_{max/Welsh}$ = 6 (right) or 7 (left). It can be observed that obtaining such set is quite fast since set of 10 000 sequences can easily be build. However, this method produces the set slower than the PRNG method. Indeed, for $x_{max/Welsh}$ = 7 near 420 000 attempts are required to produce the set. For $x_{max/Welsh}$ = 6, this number increases up to 660 000 attempts whereas less than 12 000 were necessary using PRNG method. This difference is due to the intrinsic nature of CSC generation which does not produce a-priori low-correlated sequences contrary to PRNG. This result is thus in accordance with previous observations made in §3.2.3.2.3.

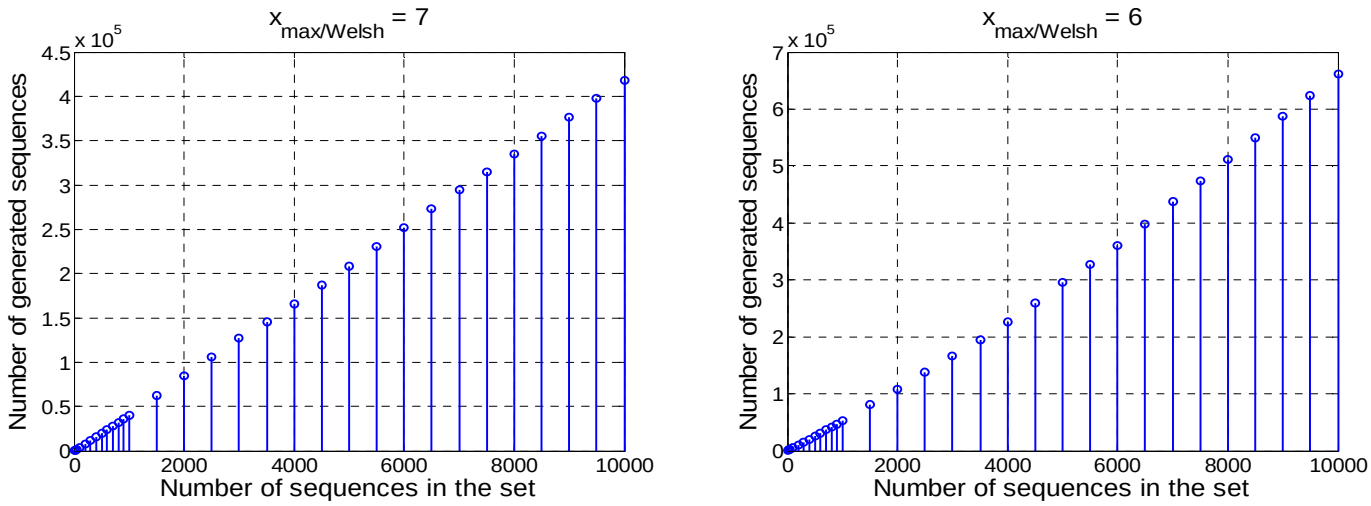

Figure 17: Number of generated sequences to obtain a set with $x_{max/Welsh}$ = 7 (left) and $x_{max/Welsh}$ = 6 (right) - case of CSC – linear scale for y-axis

The value of the scale parameter λ for $x_{max/Welsh}$ = 6 can be assessed by considering the asymptotic behaviour of Figure 18 for x > 5000 which indicates $\lambda = 1.1*10^{-3}$, what is more than for PRNG method ($\lambda = 3.1*10^{-4}$). With such value of the scale paramter, Table 6 conjectures the computational complexity for the generation of larger sets. Sets with more than 15 000 sequences seem to correspond to a practical limit in the case of CSC generation method.

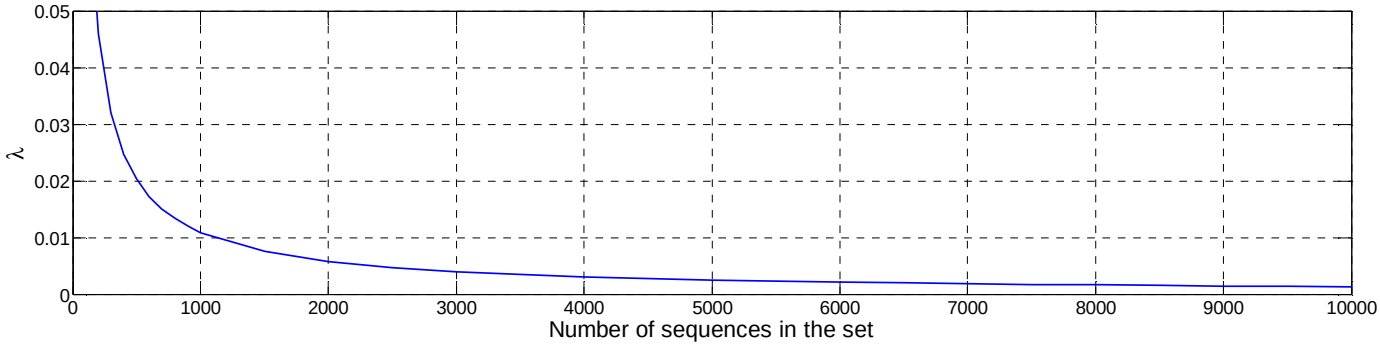

Figure 18: Evolution of the scale parameter λ in function of the size of the set for $x_{max/Welsh}$ = 6 – Case of CSC

Table 6: Measurement and *extrapolation* of computational complexity for $x_{max/Welsh}$ = 6 ($\lambda = 1.1*10^{-3}$) - Case CSC

Size of the set	10000	*15 000*	*18 000*
Number of generation	$6.60 * 10^5$	*$1.46 * 10^7$*	*$3.97 * 10^8$*
Time required	≈ 6 days	*4.2 years*	*140 years*

3.2.3.3.4.2 $x_{max/Welsh} = 5$

Figure 19 shows the number of sequences to generate to obtain a set with $x_{max/Welsh}$ =5. Similarly to PRNG technique, this figure reveals the high computational complexity of generating set with such correlation level. In particular, 12 542 generations are necessary to obtain a set of 100 sequences. This number rises to $8.57*10^5$ for a set of 500 sequences. Finally, almost $30*10^6$ runs must be done to obtain a set of 900 sequences. One can also note that all these milestones are much higher than for the PRNG method.

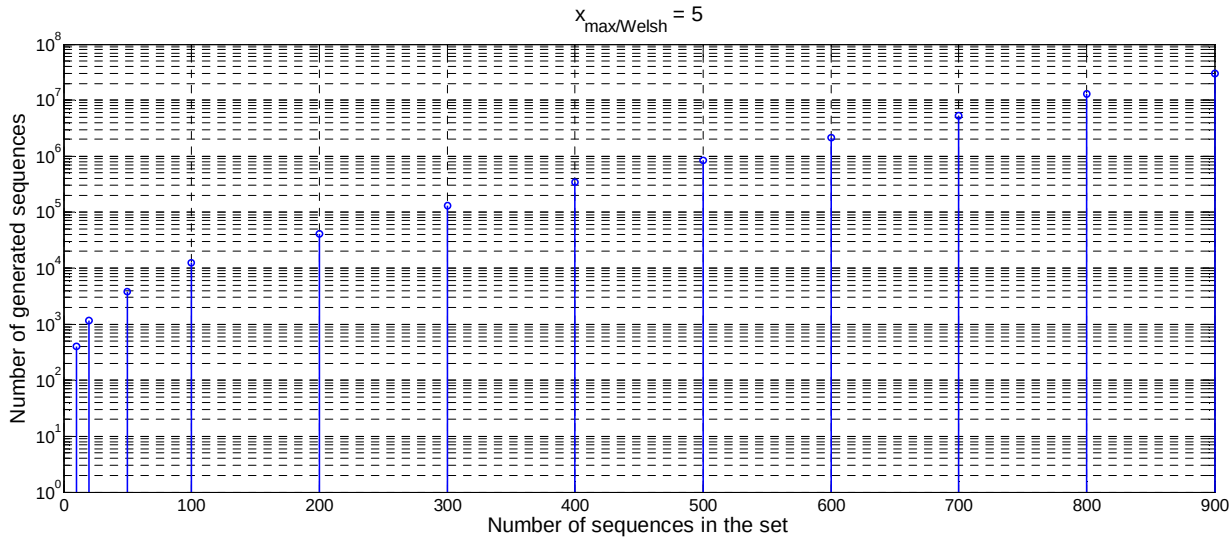

Figure 19: Number of generated sequences to obtain a set with $x_{max/Welsh} = 5$ - Case of CSC – logarithmic scale for y-axis

The asymptotic part of Figure 20 indicates the order of magnitude of the scale parameter λ. The exponential behaviour becomes significant for x > 200. In this case, the value of λ trends to $1.8*10^{-2}$. This value is superior to the scale parameter observed with the PRNG method and confirms previous observations (CSC method is much slower than PRNG method). Under the hypothesis $\lambda = 1.8*10^{-2}$, Table 7 conjectures the order of magnitude of complexity for larger sets. The slight difference between the scale parameters of the 2 methods being amplified by the exponential law, sets larger than 1100 codes are barely achievable in a reasonable amount of time, even with the highest computing technology.

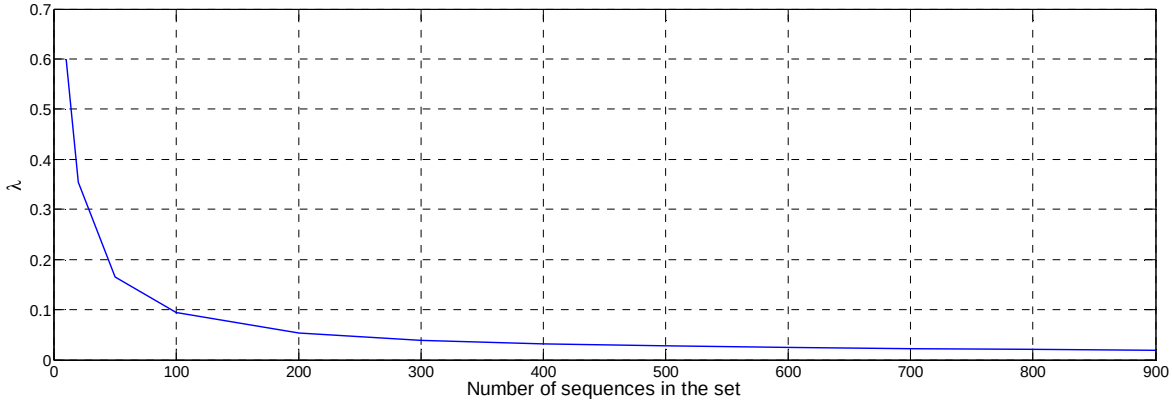

Figure 20: Evolution of the scale parameter λ in function of the size of the set for $x_{max/Welsh} = 5$ - Case of CSC

Table 7: Measurement and *extrapolation* of computational complexity for $x_{max/Welsh} = 5$ ($\lambda=1.8*10^{-2}$) - Case CSC

Size of the set	900	*1100*	*1200*
Number of generation	$2.96*10^7$	*$3.97*10^8$*	*$2.43*10^9$*
Time required	≈ 7 days	*8.5 years*	*56 years*

3.2.3.3.4.3 $x_{max/Welsh} = 4$

Figure 21 shows the number of sequences to generate to obtain a set with $x_{max/Welsh} = 4$. In this case, the increase of computational complexity is even more dramatic than in the previously studied case. More specifically, 503 attempts are necessary to find two correct sequences. The number of trials soars to 2.1 million for a set of 10 sequences and to more than 81 million to obtain 14 low-correlated codes. The number of generations is much higher than PRNG method.

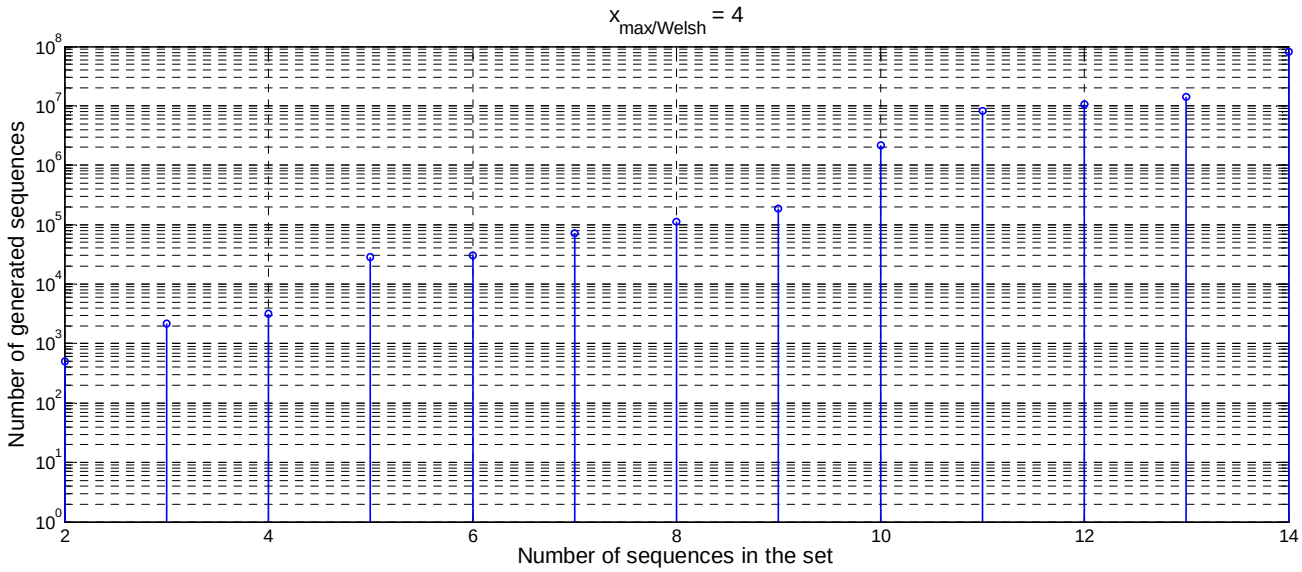

Figure 21: Number of sequences to be generated to obtain a set with $x_{max/Welsh} = 4$ - Case of CSC – logarithmic scale for y-axis

The right part of Figure 22 indicate possible of the scale parameter λ with the number of sequences in the set for $x_{max/Welsh} = 4$. As observed for PRGN, the tendency is less clear than for greater values of $x_{max/Welsh} = 5$ because the number x of selected remains small. However, λ seems to be close to λ = 1.25, especially when considering a number of sequences in the set higher than 8. One more time, λ is higher than for PRNG method. Under the hypothesis λ = 1.25, Table 8 conjectures the complexity for larger sets. With such exponential increase, sets larger than 19 codes can hardly be generated in a reasonnable amount of time, even with the highest computing technology.

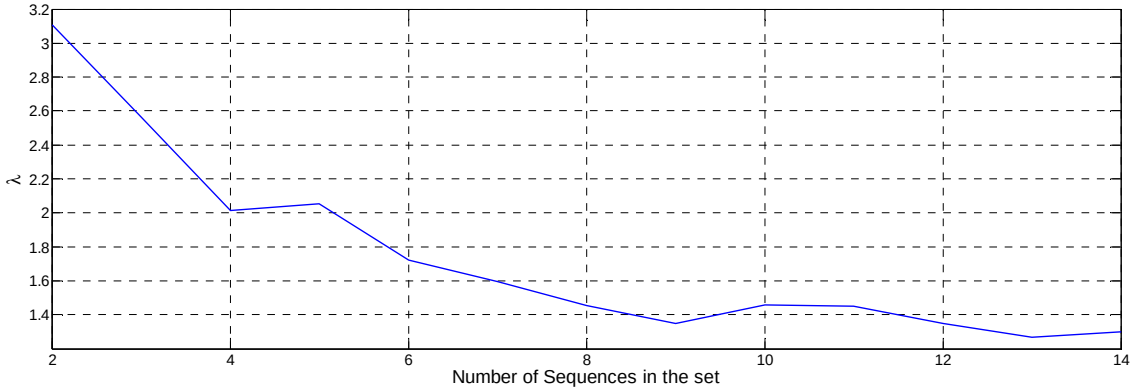

Figure 22: Evolution of the scale parameter λ in function of the size of the set for $x_{max/Welsh} = 4$ - Case of CSC

Table 8: Measurement and *extrapolation* of computational complexity for $x_{max/Welsh} = 4$ (λ = 1.25) - Case of CSC

Size of the set	14	*16*	*18*	*19*
Number of generation	$8.05 * 10^7$	*$4.85*10^8$*	*$5.91*10^9$*	*$2.07*10^{10}$*
Time required	≈ 6 days	*52 days*	*2.0 years*	*7.5 years*

3.2.3.3.5 Conclusion on complexity for generating sets of low-correlated random sequences

The simulations performed in this paragraph have studied the capacities of generating non-deterministic and arbitrary PN sequences of length N with SF = 10*log(M) = 42 dB. This study follows the requirements concerning the design of Tag Signal specified in deliverable D4.1.

We recall that for a set of N sequences of length M, the Welsh bound is close to $\sqrt{M} = 10^{SF/2}$ (see D4.1. and § 3.2.1.2 of this deliverable). This would correspond to the case $x_{max/Welsh} = 1$ in the above study (§3.2.3.3). This also corresponds to a maximum level of correlation side lobe of –21 dB under the main lobe of the correlation function.

This optimal case $x_{max/Welsh}=1$ can be approached when considering deterministic design of PN sequences (e.g. Gold sequences). Unfortunately, such deterministic design makes PN sequences much less resilient to advanced attackers than an arbitrary design (see analyses in deliverables D2.4 and D4.1).

Concerning the design of long-length and random PN sequences, the case $x_{max/Welsh}=1$ is ideal and non-realistic. However, the relevant Welsh bound was used as a metric for the study of the computational complexity of the full random generation of PN sequences with SF = 42 dB. A synthetic conclusion of this study is given in Table 9.

Table 9: Synthesis of the computational complexity for generating arbitrary random sequences with SF = 42 dB

Threshold Value for $x_{max/Welsh}$	Maximum side lobe level	PRNG and CSC capability with current computers (approximately)	Extrapolation of PRNG and CSC capability with deep computing and algorithm optimization
$x_{max/Welsh} \geq 7$	–12,5 dB	PRNG: ≈ 100 000 CSC: ≈ 100 000	PRNG: several millions CSC: several millions
$x_{max/Welsh} \geq 6$	–13,2 dB	PRNG: ≈ 40 000 CSC: ≈ 12 000	PRNG: ≈ 60 000 CSC: ≈ 18 000
$x_{max/Welsh} \geq 5$	–14 dB	PRNG: ≈ 1500 CSC: ≈ 1100	PRNG: ≈ 1 600 CSC: ≈ 1 200
$x_{max/Welsh} \geq 4$	–15 dB	PRNG: ≈ 20 CSC: ≈ 15	PRNG: ≈ 30 CSC: ≈ 20

These simulations reveal that the design of significant sets of non-deterministic random tag signal is perfectly achievable, with some compromises described hereafter:

- The management of the set allocation in the USS schemes is facilitated at high $x_{max/Welsh}$ values and high number of TSs because the risk for collision (i.e. using same TS by to different legitimate links) is very low (~ 1/N).
- The resilience of the TS allocation when facing Eve is facilitated at high $x_{max/Welsh}$ values and high number of TSs because the probability for Eve to anticipate the TS used in the legitimate link (~ 1/N) is very low.
- The synchronization processing and the CIR estimation are more precise at low $x_{max/Welsh}$ values because the threshold in the decision processing shall take into account the maximum sides lobes in order to avoid false alarms. Nevertheless the slight difference (only 1.5 dB) between the case $x_{max/Welsh} \geq 7$ and the case $x_{max/Welsh} \geq 5$ limits the practical impact.

Finally:

- A design of tag signal focussing on sensitivity shall limit the TS sets to 1000 units and take into account maximum sides lobes at –13 dB in the receiving processing
- A design of tag signals focussing on resilience shall use TS sets of more than 1 000 000 units and take into account maximum sides lobes at –12 dB in the receiving processing.