



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI ȘI
PROTECȚIEI SOCIALE
AMPOSDRU



Fondul Social European
POS DRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
CERCETĂRII
TINERETULUI
ȘI SPORTULUI

OIPOSDRU



Investește în oameni!

FONDUL SOCIAL EUROPEAN

Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007 – 2013

eng. Kinga MÁRTON

PhD Thesis

**Contributions to the Generation and Testing of
Random Number Sequences Intended for
Cryptographic Applications**

(Extended abstract)

PhD Advisor
Prof. dr. eng. Iosif IGNAT

**TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
COMPUTER SCIENCE DEPARTMENT**

2011

1 Introduction

The security of cryptographic systems is strongly related to randomness forasmuch as almost every aspect of cryptography depends on the accessibility of random number generators that exhibit high statistical quality, are available, affordable, provide high throughput and at the same time are strong in the sense that can withstand analysis.

The major importance of high performance and high quality randomness generators in cryptography attracts an increasing interest from the research community, but unfortunately this tendency is only vaguely reflected in the community of security application developers, thus many cryptographic systems, in lack of a thorough analysis of the randomness source and of the quality of produced sequences, continue to be compromised due to the utilization of inadequate randomness generators. The result is the compromise of the whole system that fails to provide the desired level of security. The loss and the associated costs and efforts that are necessary for recovering from the security break can be inestimable, and constitute the central motivation of the research work in the direction of choosing, designing and carefully integrating high quality random number generators in cryptographic systems.

In this context, the presented research work aims to fulfill the stringent need to accentuate the crucial importance of random number sequences in cryptography and to provide a set of efficient and practical techniques for the generation and testing of random number sequences intended for cryptographic applications. These three objectives constitute the main goal of the thesis.

The organization of this thesis reflects the major objectives and is structured as follows. Chapter 2 introduces the background on randomness presenting the many practical applications of random numbers in cryptography, followed by the major contributions of the thesis to the domain of random number generation - in Chapter 3, and testing in Chapter 4. Chapter 5 presents final conclusions and further perspectives.

2 Literature review

Chapter 2 provides a brief analysis and description of the fundamental concepts of the thesis highlighting the many practical applications of random numbers in cryptography, starting by a short introduction in the vast domain of randomness and types of random number generators (TRNG – true random number generators, PRNG – pseudorandom number generators and URNG – unpredictable random number generators) followed by a more detailed presentation of the many roles randomness plays in cryptographic techniques and protocols in the form of:

- *cryptographic keys* which determine the transformation of plaintext into ciphertext and vice versa in both symmetric and asymmetric techniques,
- *initialization vectors* used in symmetric stream and block ciphers in order to ensure that the ciphers produce a unique output even under the same encryption key, thus avoiding the laborious work of rekeying,
- *nonces and challenges* used to ensure message freshness, principal liveness, mutual or unilateral authentication and demonstrations of knowledge of a secret while revealing no information about the secret,
- *cryptographic salt* used generally as one of the inputs for the key derivation functions,

- *padding strings* which in symmetric block ciphers fill the last block of plaintext in order to bring the message to a length multiple of the block size and in asymmetric techniques serves to turn deterministic public encryption schemes into randomized algorithms,
- *blinding factors* used in blind signature schemes for ensuring that the message signing process is performed without exposing the message content to the signer.

The importance of each of these roles emphasizes the high degree to which cryptography relies on randomness in providing the security requirements it is designed to deliver and carries forth the significance of choosing and integrating suitable random number generators as security flaws in the generator can easily compromise the security of the whole system.

1 Contributions to the generation of random number sequences

Each of the five proposed methods for generating randomness can be classified in the category of unpredictable random number generators (URNG), which represent a suitable solution when sources of true randomness are not available, too expensive or the nature of the application employing the generator requires a higher throughput and practicality than available TRNGs (True Random Number Generators) might provide, but at the same time, the desired level of irreproducibility and unpredictability can not be met by pseudorandom generators. Hence, URNGs are practical approximations of TRNGs and are generally based on the unpredictability inherent to human computer interaction and on the nondeterminism introduced by the complexity of the underlying phenomenon.

1.1 DataFlow entropy collector

Entropy is at the heart of all random number generators, having a significant impact on the level of randomness the generators can provide.

The proposed method is an entropy collector that uses many different types of entropy sources from within a personal computer, and collects the entropy generated by these sources by means of a data flow mechanism expressed by the mouse movements over the areas of the system's interface assigned to different entropy sources. This way a new level of uncertainty is added: data from several unpredictable entropy sources flow inside the entropy pool, according to the user's unpredictable mouse movements.

Entropy may be collected from different hardware components like the keyboard, mouse, microphone, video capture devices (webcam, TV tuner, etc.), network interfaces, screen captures, or known libraries like HAVEGE, PCQNG and also from a preexisting file.

Experimental results show that the resulting data is highly unpredictable; the quality of this data from the randomness point of view is however depending on the ability of the user to efficiently collect entropy from as many sources as possible, to process the raw entropy bits using the available tools (filters, hash functions) and to combine the collected entropy by means of the data flow mechanism provided by the system.

A major advantage of the proposed system is the possibility to pre-evaluate (using simple tests) the quality of individual sources, before final combination, and to postprocess the output in order to provide high quality output sequences. Furthermore, the nondeterministic combination method and the possibility to set individual delays on each entropy source, introduce additional levels of unpredictability, which combined with independent and high quality buffers may provide cryptographically strong output random number sequences.

1.2 The Parallel-HAVEGE and GridHAVEGE generators

The unpredictable generation methods proposed in this section are based on the HAVEGE generator, which gathers entropy produced by external sources in the internal volatile processor states using the memory hierarchy and the branch prediction mechanism, and expand the entropy using a PRNG (Pseudorandom Number Generator). The security of the algorithm relies upon the fact that the internal state of the generator can not be completely determined because this state is not only composed of memory mapped data but of thousands volatile hardware states that are inaccessible even to the user running the application.

To make the algorithm more secure against an adversary trying to guess the next sequence one could skip certain results (a process called *step hiding*) by not returning from the function once the output buffer is full, but instead uses this buffer (and overwrites it) to generate a new set of random numbers. This could be done several times, sacrificing speed for security.

The goal of the proposed parallel and distributed methods is to provide high security level without significant loss in the generator's throughput by enabling HAVEGE to take advantage of the processing power of the latest technologies in parallel computing using multi-core architectures and distributed computing using the Grid infrastructure.

The experimental results show that the parallel and "gridified" versions of HAVEGE are useful if a large amount of unpredictable random numbers is needed. The use of any function that is safer (and therefore slower) than the *ndrand* function could also benefit from a speed boost in real life conditions where the numbers need to be consumed with a higher rate than the generator's output rate.

The more steps performed by the custom harvesting function and the more volatile values are involved in the generation, the more difficult will be for an adversary to predict the output and therefore the greater the security of the algorithm. However, in a serial implementation this increase in security comes at a price: as we increase the number of steps, the speed (throughput) of the generator will decrease proportionally.

A significant advantage of having parallel and distributed implementations of HAVEGE is the ability to increase the security (number of steps) while maintaining a constant throughput by increasing the number of HAVEGE threads up to the number of available cores.

1.3 Unpredictable random number generator based on Hardware Performance Counters

Originally intended for design evaluation and performance analysis, hardware performance counters (HPCs) enable the monitoring of hardware events, yet are noisy by their very nature. The causes of variations in the counter values are so complex that are nearly impossible to determine. Hence, while being a major issue in the process of accurately evaluating software products, the unpredictability exhibited by HPCs offer a high potential for random number generation.

The proposed method introduces a new unpredictable random number generator (URNG) based on HPCs and analyses the feasibility of producing cryptographic quality randomness. For cross-platform compatibility reasons only HPCs associated to *preset events* were considered and through several experiments a subset of these HPCs were identified that can be employed as sources of high unpredictability within the proposed generator.

The statistical quality of the generated randomness was thoroughly tested using the well known statistical test suites: Alphabit and Rabbit batteries of TestU01 and the NIST test suite, the

results showing the high randomness quality of generated sequences, nonetheless carrying forth the major importance of applying multiple statistical test suites in order to increase the confidence in the generator.

Furthermore, these experiments also emphasize issues that have to be considered such as: the choice of the sampled events, the decrease in randomness quality when concatenating the outputs of concurrent threads sampling the same event counter and the memory requirements for mixing the output files of concurrent threads.

The throughput of the proposed generator is approximately 150 KB/s, a value comparable with the throughput exhibited by the state of the art HAVEG generator. Therefore all aspects regarding the unpredictability, throughput and availability of the randomness source and the quality of the generated sequences prove the proposed unpredictable random number generator's suitability for integration in security systems for providing cryptographic randomness.

1.4 A parallel unpredictable random number generator

The goal of the proposed method is to introduce a parallel random number generator that unpredictably combines pseudorandom number sequences. The proposed generator takes advantage of the high statistical quality and throughput provided by several of the employed pseudorandom number generators and provides the desired level of unpredictability and irreproducibility, based on the nondeterministic mouse movement driven combination technique. Furthermore, the parallel execution of selected generators introduces a significant improvement in both the throughput and quality of the generated sequences.

Results of the statistical testing process, using the well-known NIST test suite, prove the high quality of the produced randomness, which combined with a high throughput and unpredictability, enable the generator to be considered as a suitable solution for providing cryptographically secure randomness.

2 Contributions to the testing of random number sequences

The contributions presented in this chapter include the performance enhancement and optimization of several well-known statistical testing suites. Unfortunately the implementations of the most popular batteries of test suites are not focused on efficiency and high performance and do not benefit from the processing power offered by today's multi-core processors and tend to become bottlenecks in the processing of large volumes of data generated by various random number generators. Hence there is a stringent need for providing highly efficient statistical tests and our research efforts briefly presented in the following intend to fill this need.

2.1 SeqENT and ParENT statistical test suites

The optimization steps applied to the ENT statistical test suite, originally developed by John Walker, are oriented in two main directions, namely the development of a sequential version, called SeqENT, which offers superior performance to the one shown by the original '98 version, and the development of a parallel application, called ParENT, the utilization of which reduced the processing time significantly compared to the '08 sequential version.

The improvement steps taken to overcome the limitations of the original ENT versions include the optimization of the read operation, the possibility to process very large files, the removal of redundant operations and the introduction of parallelism.

SeqENT provides an execution time reduced by approximately 12 times compared to the original '98 ENT version, and ParENT shows an acceleration of approximately 10 times compared to the '08 ENT version, and 20 times compared to the original '98 ENT version.

2.2 SeqNIST and ParNIST statistical test suites

The NIST STS became a „de facto” standard in the process of statistically evaluating random number generators intended for cryptographic applications shortly after its publication by the National Institute of Standards and Technology (NIST).

The applied optimization steps to the original NIST statistical test suite include the paradigm shift towards byte processing mode, the possibility to process large volumes of input data (up to 1 GB) and the most significant improvement stage – the parallelization. The resulting improvements are: a high efficiency sequential version - SeqNIST, where only the first two improvement steps are used and for which execution time was reduced up to approximately fourteen times on the average, and a high performance parallel version - ParNIST, for which the results show a significant performance improvement, the execution time being reduced up to 103 times, and on average by approximately 54 times.

2.3 Visual inspection of random number sequences

The human visual system is highly trained for detecting patterns in the surrounding world, a valuable feature that can be applied in examining more abstract information as well.

The proposed FileSeer tool is designed to take advantage of the enormous power and subtlety of the human visual perception and employ it for the visual inspection of the quality of random number sequences. To this end FileSeer provides a framework for representing the examined information as black and white, grayscale and color images which enable the spotting of repeated patterns and includes the visualization of certain statistical properties of the random sequence such as its balance, entropy and histogram.

In the process of testing random number sequences, the visual representations can help humans comprehend what randomness is and how it looks like by changing the process of understanding from being a cognitive task to being a perceptual task. But at the same time it is important to note that by representing a sequence of numbers graphically, the human visual system is only capable of determining the degree to which the representation satisfies visual randomness but is unable to tell the difference between real randomness and visual patternlessness, hence the proposed inspection tool is designed to complete the wide range of statistical tests by providing an additional dimension to the understanding of randomness and of the usefulness of testing the quality of random sequences by assigning the abstract data a perceptual representation.

2.4 The problem of counting bits efficiently

The problem of counting the set bits of a bit sequence (commonly known as the “popcount” function) is one of the most interesting problems in cryptography, is associated with several interesting algorithmic problems and is of special interest in the field of testing the quality of random sequences.

After analyzing several different counting methods, the comparative results suggest that the fastest methods are no longer based on the lookup table approach. Instead, by combining the state-of-the-art SSE2 instructions and 128 bit registers with parallel tree-like summation and the CSA (Carry Save Adder) approach, the proposed methods require even less than 0.2 seconds per GB of data, ignoring the time required to bring the processed sequence into the main memory.

These results show a performance improvement of 100 times compared to the NIST implementation of the Frequency (monobit) Test.

3 Conclusions

The main topic of this thesis is integrated within the domain of information security, emphasizing the importance of employing high quality, efficient and practical random number generators in cryptographic applications.

3.1 Major contributions

The list of the major contributions of this thesis is provided in the following.

1. Providing a meaningful glance into the domain of random numbers in cryptography by presenting the different generation methods and essential properties of random number sequences intended for cryptographic applications;
2. Synthesizing and presenting the many roles random number sequences play in cryptography and carrying forth the significance of choosing and integrating suitable random number generators;
3. The development of an entropy collector that uses many different types of entropy sources from within a personal computer, and collects the entropy generated by these sources by means of an unpredictable data flow mechanism expressed in the mouse movements over the areas of the system's interface assigned to different entropy sources;
4. The development and testing of the optimized generation methods *Parallel-HAVEGE* and *GridHAVEGE*, both based on the HAVEGE generator and introduce performance optimizations by using parallel and distributed programming paradigms. Hence the proposed systems allow the achievement of a high security level without significant loss in the throughput, maintaining at the same time the high statistical quality provided by the original generator version;
4. The development and testing of an unpredictable random number generator based on hardware performance counters. The generator exploits the unpredictable and irreproducible variations in the counter values that are due to the noisy nature of these special purpose registers. The cause of variations is very complex and extremely difficult to deduce. The high level of statistical quality is indicated by the result provided by the Rabbit, Alphabit and NIST statistical test suites. The throughput of the generator is comparable to the one provided by the well-known HAVEG unpredictable random number generator.
6. The development and testing of an unpredictable random number generator that involves the parallel and nondeterministic combination method based on the unpredictable mouse movements over the visual grid containing the chosen PRNG arrangement. This method provides a practical solution for employing well-known and newly defined pseudorandom generator variants in a way that mitigates certain security problems the individual generators are exposed to if used on their own. The unpredictable combination method dissolves patterns specific to some generator families and enables the generation of more uniformly distributed and structurless sequences. Furthermore, the parallel execution of selected generators introduces a significant improvement in both the throughput and quality of the generated sequences;

7. The development and testing of the optimized statistical testing suites SeqENT and ParENT, both based on the ENT suite and introduce significant performance amplification: the former within the sequential programming paradigm and the latter through exploiting the computing power of parallel multi-core architectures;
8. The development and testing of the optimized statistical testing suites SeqNIST and ParNIST, both based on the well-known NIST STS and introduce significant performance amplification the former within the sequential programming paradigm and the latter through exploiting the computing power of parallel multi-core architectures;
9. The design and development of a visual inspection tool which takes advantage of the enormous power and subtlety of the human visual perception and employs it for the visual inspection of the quality of random number sequences. The proposed inspection tool is designed to complete the wide range of statistical tests by providing an additional dimension to the understanding of randomness and of the usefulness of testing the quality of random sequences by assigning the abstract data a perceptual representation;
10. The analysis and testing of a wide range of bit counting methods and the development of two new and efficient counting methods based on the combination of SSE2 instructions with parallel tree-like summation and the CSA (Carry Save Adder) method.

Each of these original contributions was validated by experimental testing and by presenting and publishing the methods and associated results within international conferences and journals.

4.1 Further work

The research activity addressed in this thesis can be continued and further developed by extending the proposed methods and approaches and by proposing new and efficient methods for both the generation and testing of random number sequences intended for cryptographic and steganographic applications.