NEW CLASSES OF BINARY RANDOM SEQUENCES FOR CRYPTOGRAPHY

By

PRASHANTH BUSIREDDYGARI

Bachelor of Engineering in Electronics & Communication Engineering
Visvesvaraya Technological University
Bengaluru, Karnataka
2008 - 2012

Master of Science in Electrical & Computer Engineering
Oklahoma State University
Stillwater, Oklahoma
2014 - 2015

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 2019

NEW CLASSES OF BINARY RANDOM SEQUENCES FOR CRYPTOGRAPHY

Dissertation Approved:

Dr. Subhash Kak

Dissertation Advisor

Dr. Rama Ramakumar

Dr. Daniel Grischkowsky

Dr. Jinkyu Lee

Name: PRASHANTH BUSIREDDYGARI

Date of Degree: May, 2019

Title of Study: NEW CLASSES OF BINARY RANDOM SEQUENCES FOR CRYPTOGRAPHY

Major Field: Electrical Engineering

Abstract: In the vision for the 5G wireless communications advancement that yield new security prerequisites and challenges we propose a catalog of three new classes of pseudorandom random sequence generators. This dissertation starts with a review on the requirements of 5G wireless networking systems and the most recent development of the wireless security services applied to 5G, such as private-keys generation, key protection, and flexible authentication. This dissertation proposes new complexity theory-based, number-theoretic approaches to generate lightweight pseudorandom sequences, which protect the private information using spread spectrum techniques. For the class of new pseudorandom sequences, we obtain the generalization. Authentication issues of communicating parties in the basic model of Piggy Bank cryptography is considered and a flexible authentication using a certified authority is proposed.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

LIST OF ABBREVIATIONS

1G	1st Generation wireless communication and network systems
2G	2nd Generation wireless communication and network systems
3G	3rd Generation wireless communication and network systems
4G	4th Generation wireless communication and network systems
5G	5th Generation wireless communication and network systems
AWGN	Additive White Gaussian Noise
BDM	Bit Division Multiplex
BEAST	Browser Exploit Against SSL/TLS attack
CA	Certification Authority
CBC	Cipher Block Chaining
CDMA	Code-Division Multiple Access
CSI	Channel State Information
CSPRNG	Cryptographically Secure Pseduo Random Number Generator
DDoS	Distributed Denial of Service
DoS	Denial of Service
D2D	Device-to-device
DL	Double-lock cryptography
DFA	Deterministic Finite Automaton
DFT	Discrete Fourier Transform
EAP	Extensible Authentication Protocol
FDMA	Frequency Division Multiple Access
FSM	Finite State Machine
GE	Gate Equivalent
GMW	Gorden-Mills-Welch sequences
HG	Helleseth-Gong sequences
IDMA	Interleave Division Multiple Access
IoT	Internet of Things
IV	Initialization Vector
KW	Kasami-Welch sequences
LDS-CDMA	Low Density Spreading - Code Division Multiple Access
LFSR	Linear Feedback Shift Register
MAC	Message Authentication Code

MD5	Message Digest 5
MIMO	Multi-Input Multi-Output systems
MSACC	Mean Square Aperiodic Auto Correlation
MSAAC	Mean Square Aperiodic Cross Correlation
MIM	Man-in-the-Middle attack
mmWave	Millimeter Wave
MUSA	Multi-user Shared Access
NOMA	Non-orthogonal Multiple Access
NFA	Nondeterministic Finite Automaton
NFV	Network Functions Visualization
NIST	National Institute of Standards and Technology
OFDMA	Orthogonal Frequency Division Multiple Access
OMA	Orthogonal Multiple Access
OTP	One-Time Pad
PB	Piggy Bank protocol
PDMA	Pattern Division Multiple Access
PLS	Physical Layer Security
PN	Pseudo Noise or Pseudo Random
PRNG	Pseudo Random Number Generator
PSK	Pre-Shared Key
RAM	Random Access Memory
RFID	Radio Frequency Identification
RMS	Root-Mean-Square
RSA	Rivest-Shamir-Adleman public-key cryptosystem
SAMA	Successive interference cancellation Amenable Multiple Access
SCMA	Sparse Code Multiple Access
SDMA	Spatial Division Multiple Access
SDN	Software Defined Network
SKID	Secret- Key Identification protocol
SHA	Secure Hash Algorithm
TDMA	Time Division Multiple Access
TLS	Transport Layer Security
UDN	Ultra-Dense Network
V-BLAST	Vertical Bell Laboratories Layered Space Time
WEP	Wired Equivalent Privacy
WSN	Wireless Sensor Networks

CHAPTER 1

INTRODUCTION

5th generation wireless communication and network systems (5G) are envisioned the next generation telecommunications to serve heterogeneous collection of scenarios: from mobile to Internet of Things (IoT) communications [Panwar et al. 2016]. The focus of 5G research and development ranges from high capacity data transfers to high density broadband users with low latency and low energy consumption and guaranteeing a high level security and privacy [Zhang and et al, 2017].

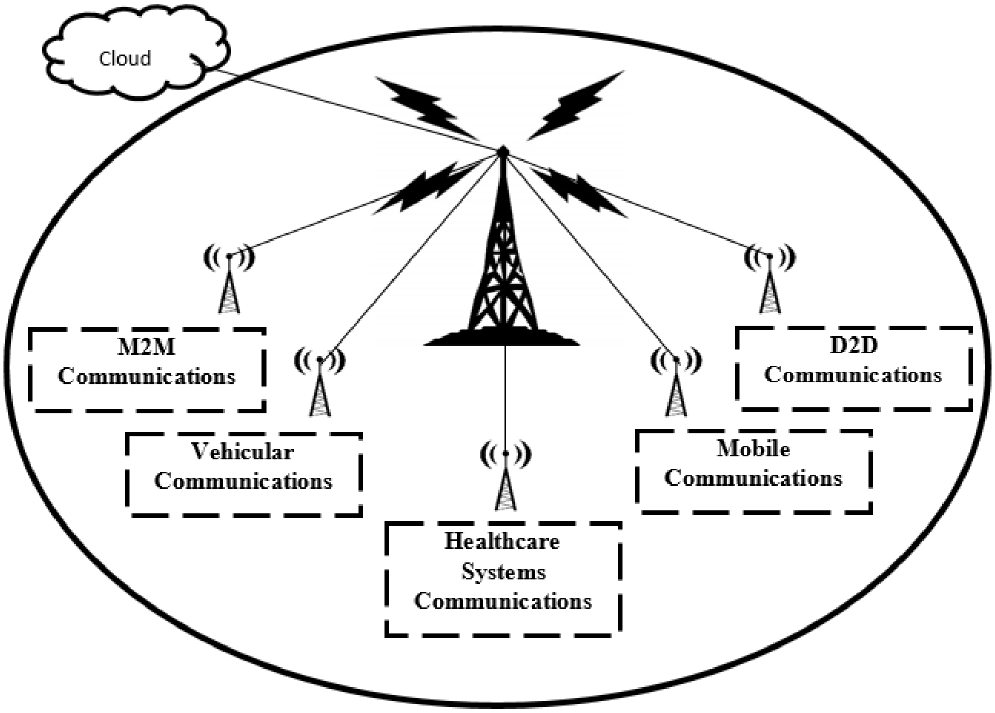


Fig. 1. 5G Wireless systems architecture

Fig. 1 represents in a schematic way a generic 5G wireless systems architecture. Some of the breakthrough features of 5G wireless systems present: 100% coverage, 99.99% availability, 1000x bandwidth allocation per unit area, 1 to 10 Gbps data rate to the user, 10 to 100x device connection, 1ms latency, 90% network energy usage reduction, and an average 10-year battery life for low powered devices (IoT) [Edward and Zoraida, 2018]. Recent technological advancements in millimeter waves (mmWave) [Qiao and et al, 2015], network slicing [Bordel and et al, 2018], software defined networks (SDN) [Dabbagn and et al, 2015], network function visualizations [Wang and et al, 2017], device-to-device (D2D) communications [Qiao and et al, 2015], and pseudorandom number generators (PRNGs) [Adem and et al, 2015] have been applied for 5G commercialization.

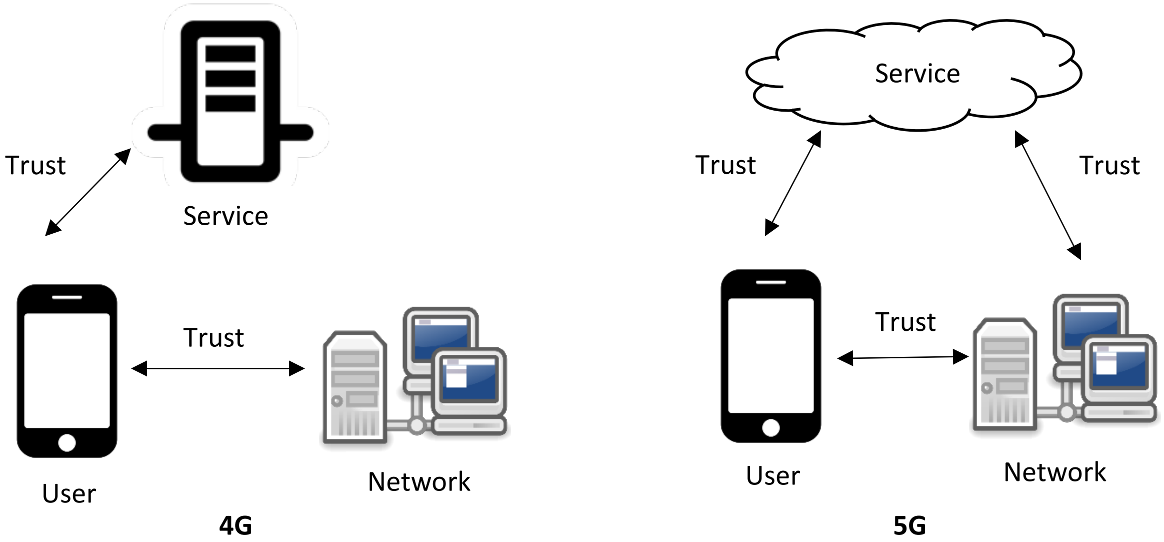


Fig. 2. Comparison of trust models in 4G and 5G wireless networks

5G wireless systems provide communications to several kinds of smart devices and new applications that connect user community. Fig. 2. shows a comparison of 4G and 5G networks. The new technologies, and the new networking paradigms, together with new use case scenarios in 5G wireless systems offer risks to users' security at MAC and physical layers due to the current broadcast nature and resource-constrained hardware and software environments [Bordel and et al, 2018] [Vij and Jain, 2015] , where the term resource-constrained describes an integrated development environment that has reduced design space [Orue and et al, 2017].

Majority of heterogeneous collection of 5G scenarios - IoT communications and device-to-device communications: device-to-vehicle communications; vehicle-to-infrastructure communications – have resource constrained microcontrollers and/or other embedded devices, so use of complex algorithms to generate pseudorandom (PN) sequences that are used as private-keys is not a viable solution. Data encryption standards in current 4G wireless networks incorporate complex PRNGs that cause signal delays and drifts while sharing the private-key [Orhanou and Hajji, 2016]. Block ciphers and stream ciphers [Berbain and et al, 2008] [Hell and et al, 2007] [Babbage and Dodd, 2008] [Canniere and Preneel, 2008] whose computational costs are unaffordable for embedded wireless devices, including IoT, as are the proposed security solutions to generate PN sequences [Bordel and et al, 2018].

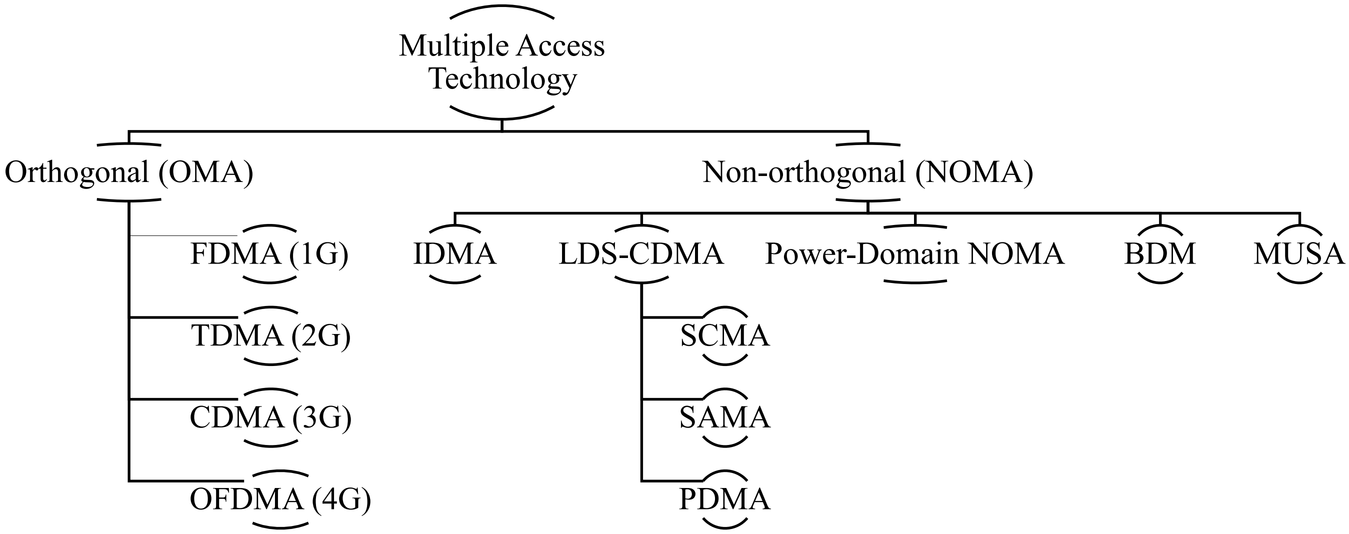
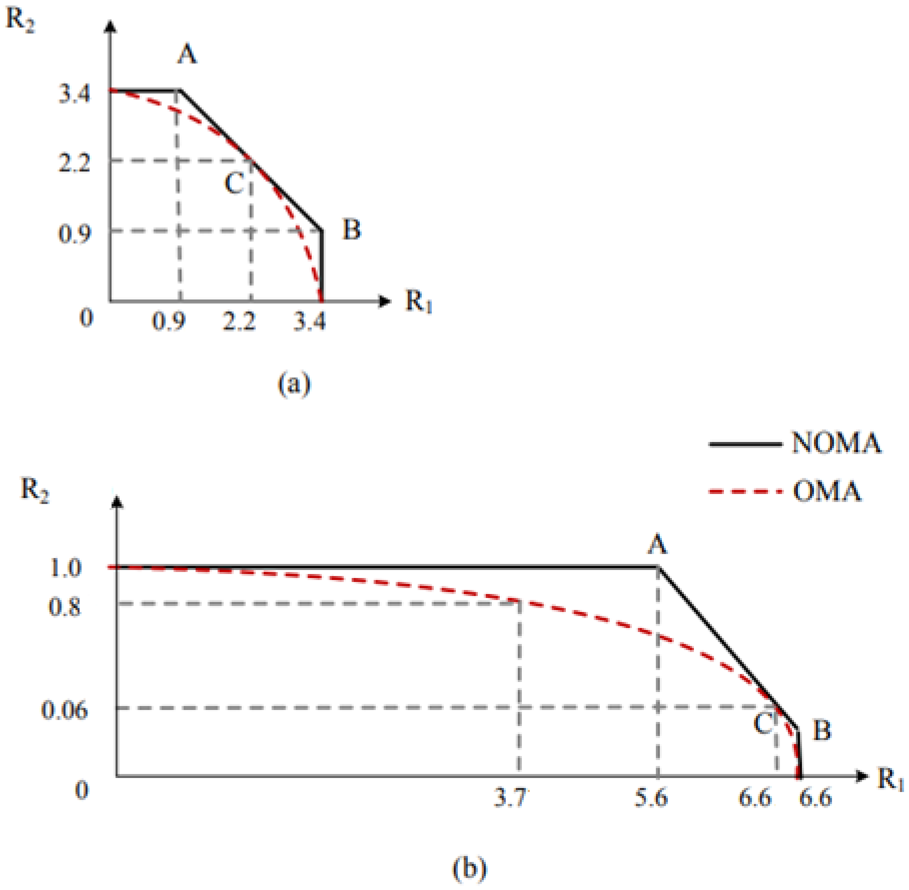


Fig. 3. Revolution of wireless communication systems [Zhou and et al, 2016].

Non-orthogonal multiple access, NOMA, is the basis of 5G and it investigates the support for users at large capacity with existing orthogonal resources: time, frequency, or code-domain. Fig. 3. Above illustrates the evolution of wireless communications. The concept of NOMA designed in a way it should allocate non-orthogonal resources to all wireless users at an expense of increased complexity at the receiver end. The complexity increases due to the separation process of non-orthogonal signals. In the recent past, [Choi, 2016] [Du et al. 2016] [Zhou and et al, 2016] and [Dai and et al, 2018] investigates several solutions for NOMA. The classification of NOMA schemes is based on: (a) power-domain; (b) code-domain.



(a)

(b)

Fig. 4. OMA and NOMA uplink channel capacity: (a) Symmetric channel; (b) Asymmetric channel [Dai and et al, 2018]

Power-domain NOMA allocates multiple users with multiple levels of power signals based on their communication channel quality, meanwhile sharing the same orthogonal resources among multiple users. Based on the user's power-difference measure, successive interference cancellation (SIC) is evaluated and distinguishing of different users is made. Fig. 4 and Fig. 5 illustrates the contrast between uplink and downlink channel capacity of OMA and NOMA.

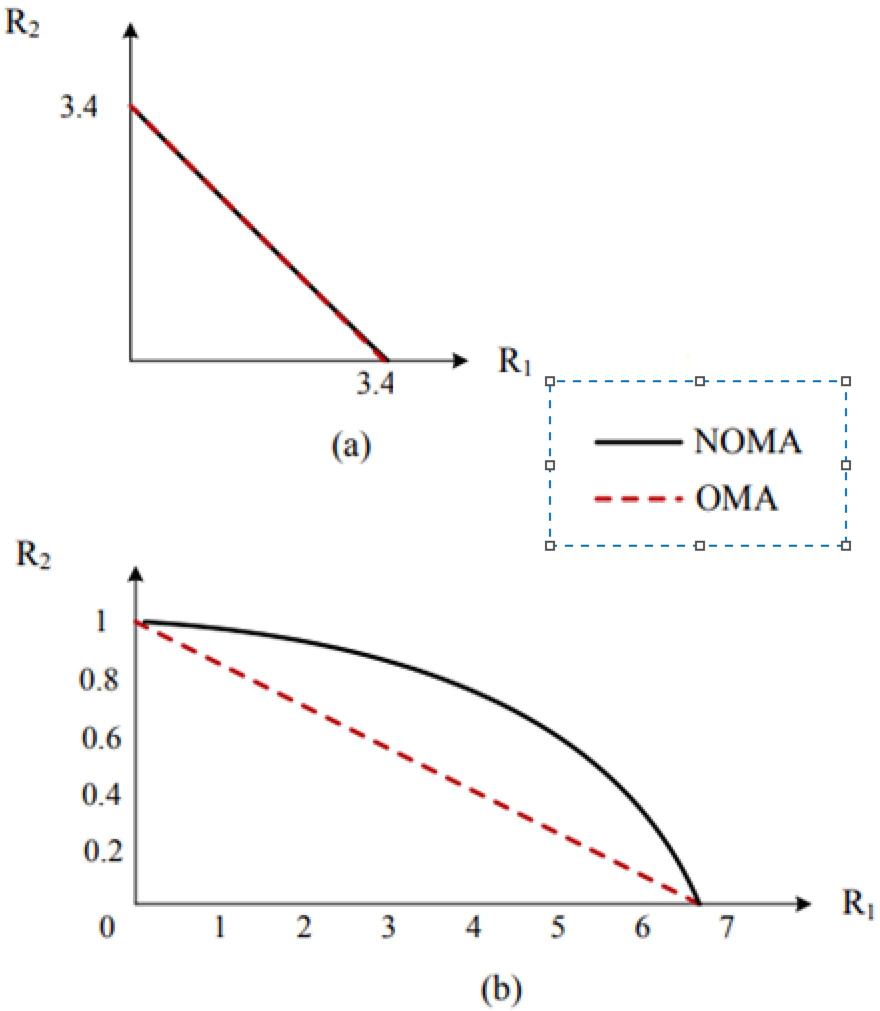


Fig. 5. OMA and NOMA downlink channel capacity: (a) Symmetric channel; (b) Asymmetric channel [Dai and et al, 2018]

Code-domain NOMA came into existence with the success of CDMA, where various users share same orthogonal resources; however, use exclusive device specific PN sequences as spreading sequences. In code-domain NOMA, compared to CDMA, the general use of spreading sequences is limited to binary PN sequences with minimal correlation properties.

It is important in power-domain NOMA, to have error-free channel state information (CSI) to achieve the efficient system performance, otherwise there will be significant performance decrease because of channel estimation miscalculations and [Rusek and et al, 2013] latency factors [Larsson and et al, 2014]. There has been an exhaustive investigation done by [Zhao, 2015] [Fu and Tao, 2012] [Samardzija, 2001] to enhance the NOMA performance using conventional V-BLAST using spread spectrum sequences to distinguish communicating users. Using various readily available PNRGs and orthogonal code generators, having variable sequence lengths and statistical properties, the complexity of at the receiver will be less; however, the system performance will be reduced, mainly, mainly in terms of low latency. The classical CDMA systems determine the accuracy of modulation and demodulation in terms of a correlation scores.

In [Novosel and Sisul, 2014], the authors show the comparison of RMS delay spreads against different PN sequences. The RMS delay spreads for different channel models using Kasami sequences, for a sequence length 991, is summarized in the below Table 1.

Table 1. RMS delay spreads for different channel models [Novosel and Sisul, 2014]

Channel Model	**RMS Delay (milliseconds)**
Urban	1,100
High traffic urban	2,400
High terrain	4,000
Rural	100

Lightweight security solutions are in demand for 5G wireless systems to minimize delays and resource allocation. Lightweight cryptography is based on cryptographic primitives customized for the constrained IDEs [Mouha, 2015]. As an instance, in hardware design, we consider hardware memory, chip size, energy consumptions, computational delay, bandwidth allocation to evaluate lightweight properties and efficiency of wireless communication. The design complexity to implement a PRNG depends on the total number of logic gates required, whereas in software programming, the developers consider program execution time, RAM utilization and code size. All the developers follow a minimalist approach, using less resources, without taking security risks in account to design a communication systems, specifically, IoT systems that are prone to various cyber-attacks – interference, spectrum scanning, packet sniffing and so on – since they are unattended pervasive wireless systems and are also heavily resource constrained [Orue and et al, 2017].

It is a challenge to implement cryptographically secure PRNGs (CSPRNGs) in the resource-constrained IoT devices. Orue and et al has found majority of PRNGs in his reviewed literature are susceptible to cybersecurity attacks and are cryptographically weak [Peinado and et al, 2014] [Mabin and et al, 2017] [Tian and et al, 2017] [Banik and Maitra, 2013]. Secure communications and robust network services to embedded technologies – IoT, radiofrequency identification (RFID), wireless sensor networks (WSN) – are established by strengthening physical-layer security that involves message encryption and authentication. For encryption, generation of symmetric key using lightweight PRNGs is a viable option.

Since 2009, when 4G was launched, several different security solutions have been proposed for IoT devices, which includes block ciphers [Shirai and et al, 2017], stream ciphers [Bernstein, 2008], hash functions [Aumasson and et al, 2010], traditional PRNGs (based on LFSRs) [Rahman and et al, 2017], and lightweight PRNGs [Katagi and Moriai, 2008]. The key negotiation phase mandates the aforementioned list of security mechanisms:

1 Extensible Authentication Protocol – Transport Layer security (EAP-TLS) [Pawlowski et al. 2014]

2 EAP-Pre-Shared Key (EAP-PSK) [Hernandez and et al, 2002]

3 EAP-Message Digest (EAP-MD5) [Marin and et al, 2015].

In this dissertation we focus on lightweight CSPRNGs, as it is our main focus. In [Fathi and et al, 2015], the authentication delays in various 4G wireless systems were investigated. Fig. 6 to Fig. 10 summarize the results of their analysis. As the authentication message increases, the average authentication delay increases, making the 4-way handshake employed by 802.11i incur the highest delay. As a result, the security offered by 802.11i tradeoffs with increased authentication delay.

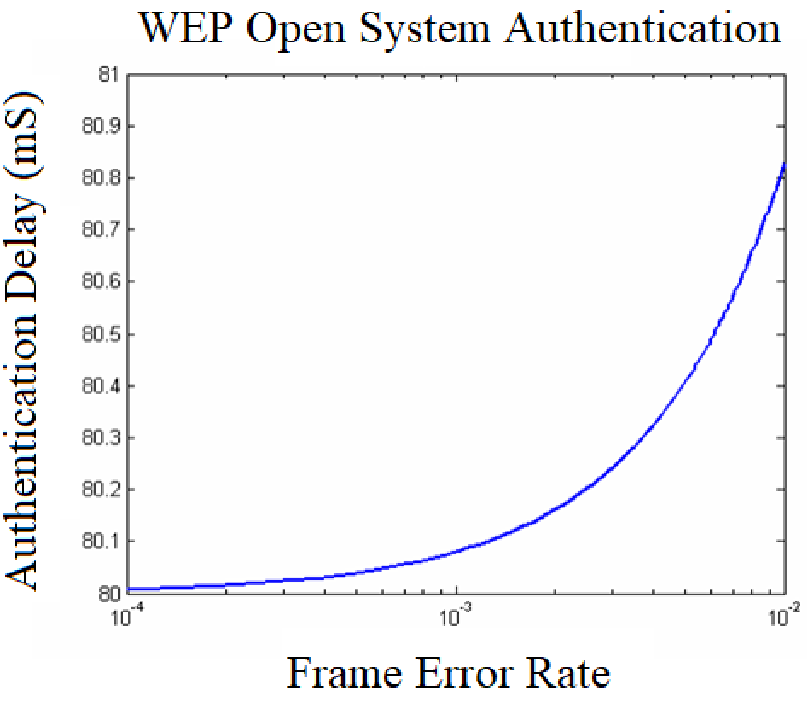


Fig. 6. Authentication delay for WEP open system authentication [Mkubulo, 2014]

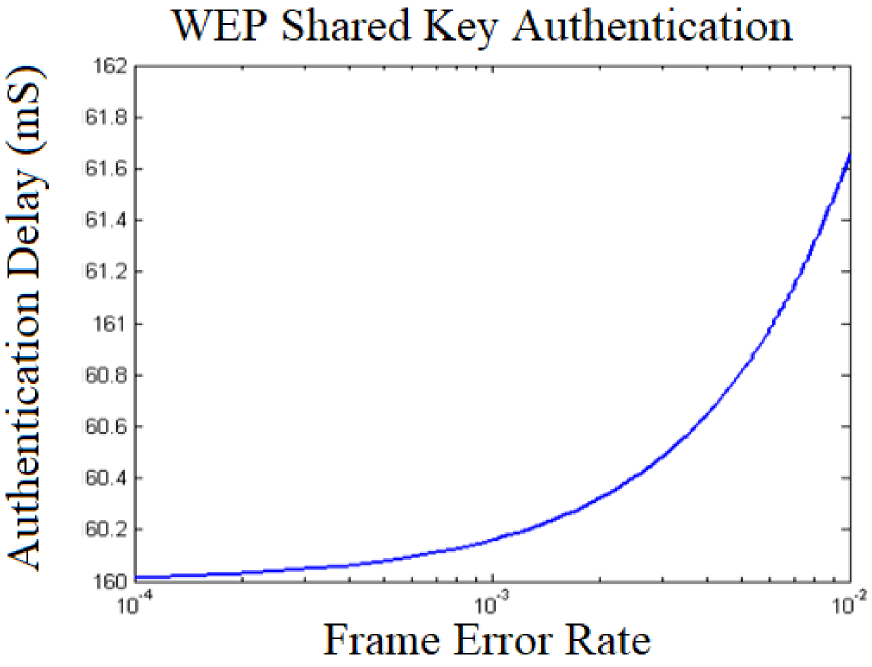


Fig. 7. Authentication delay for WEP shared key authentication [Mkubulo, 2014]

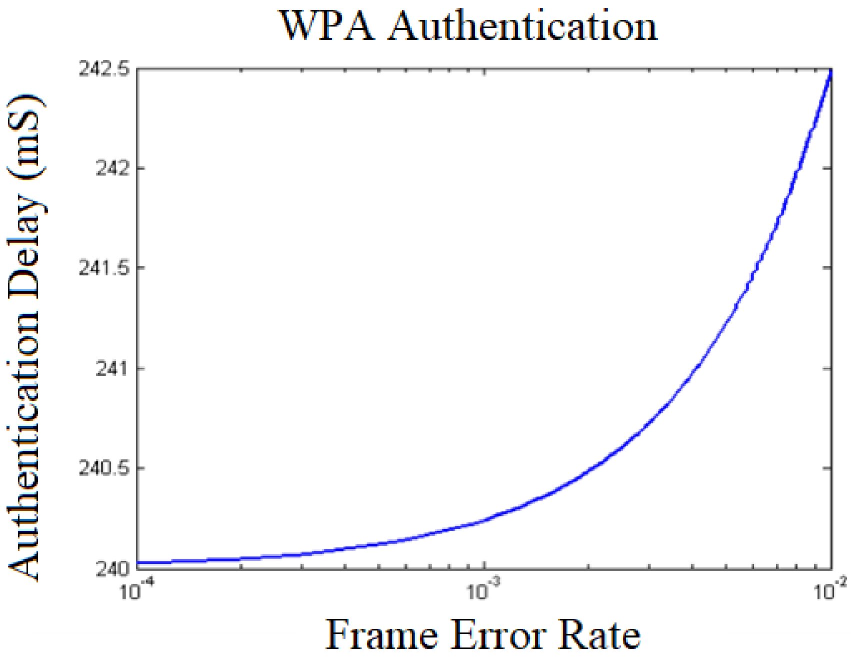


Fig. 8. Authentication delay for WPA authentication [Mkubulo, 2014]

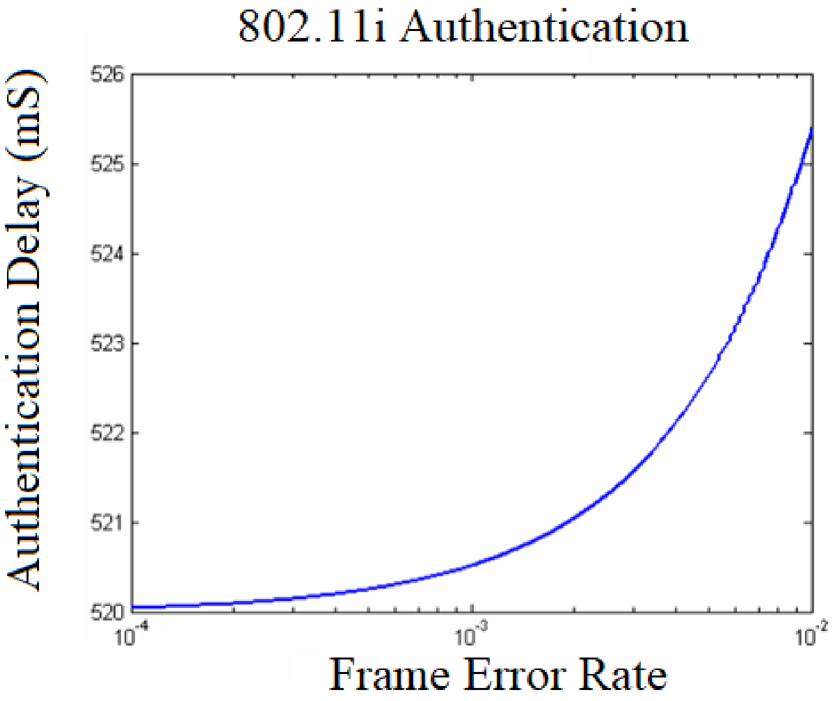


Fig. 9. Authentication delay for 802.11i authentication [Mkubulo, 2014]

Table 2. Stream ciphers performances for low power devices

Cipher	**Security**	**Key size**	**Speed (cycles per byte)**	**Initialization Vector**
HC-128	No	256	8	512
Rabbit	Yes	256	3.7 to 9.7	128
Salsa	No	512	4.24 to 11.84	256
ISSAC	No	256	2.38 to 4.69	N/A
Grain	No	160	?	128
MICKEY	No	160	>8	512
RC4	No	160-4096	7	N/A
BBS	Yes	Variable	4-5	N/A
LFG	No	Variable	6.53	N/A
Trifork	Yes	Variable	5.5	160

The Well-known lightweight stream ciphers are HC-128 [Wu, 2004], Rabbit [Boesgaard and et, 2003], Salsa [Bernstein, 2008], SOSEMANUK [Berbain, 2008], Grain [Hell and et al, 2007], MICKEY [Babbage and Dodd, 2008], RC4 [Prasithsangaree and Krishnamurthy, 2003], BBS [Bucerzan and et al, 2010], and LFG [Schneier, 1996]. Table 2 depicts the variation of stream ciphers performance for IoT applications.

The concept of secure cipher uses one-time pad (OTP), which is acquired by convoluting a PN sequence, typically the same length as the message, with the message [Beaulieu and et al, 2015] [Biham and Dunkelman, 2000]. How the random sequence is shared with the intended communicating party is the focus and challenge in secure communications and the practical solution for that is to use a PNRG. All PRNGs mandate initialization seed(s) –seed state or initialization vector (IV) to output a random sequence. The seed value or vector plays a major role in secure communications. The TLS CBC IV attack or the BEAST attack [Kurokawa et al, 2016] is an example of initialization value being compromised. The IV selection criteria should be two-fold: (1) Unpredictable; (2) At a maximum length to rescue adversary search attacks. This is a major motivation for generating cryptographically secure PN sequences. In this work we use a lightweight PRNG as a core of a stream cipher for emerging 5G network and to serve as a spread spectrum technique.

The most concerned features of 5G include cryptography and physical layer security. Physical layer security is based on resource allocation to cover that [Wang and et al, 2016] introduced security-oriented resource allocation scheme in ultra-dense networks (UDNs). The authors presented several resource dimensions that may be employed to influence secure transmission. The authors recommend major focus should be on the power, time and bandwidth allocation, the relay selection, and beam forming. The authors speculate there exists open issues and future directions in physical layer security regarding interference suppression, jamming, and mobile security, and heterogeneity.

Since 5G networks thrive for low latency, message authentication process must be made efficient than ever before. For example, consider RFID authentication, which has been popularly used for various authorization applications has to several resource limitations. The cryptographic algorithms and authentication mechanisms involved in the RFID application must be strong enough to tackle adversary attacks.

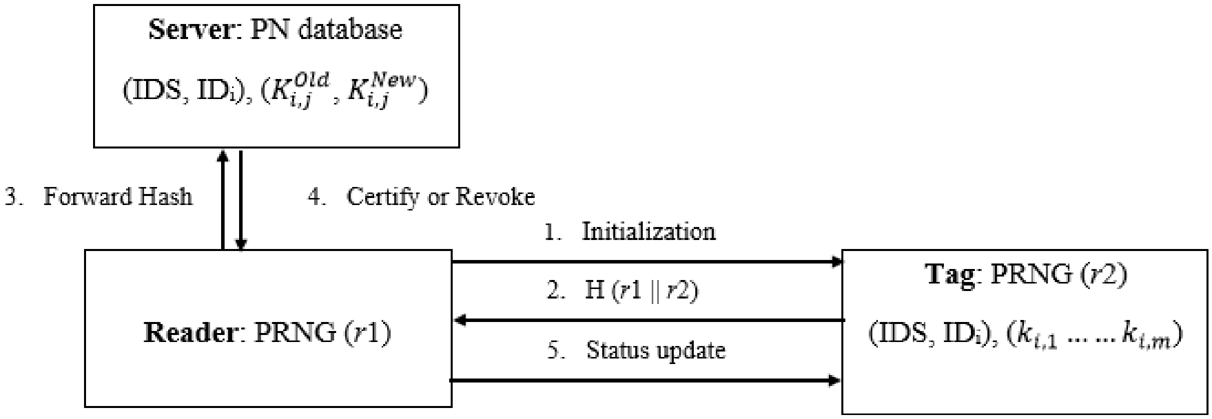


Fig. 10. RFID authentication

Fig. 10 shows the authentication mechanism of a RFID scheme. The reader consists of a PNRG and the server corresponds a hash function of it and stores it as record (PNDB). The server validates a record for every established tag (TDS, $\mathrm{ID_i}$) as $(k_{i,j}^{old}, k_{i,j}^{Now})$. r_1 is the first random instance by the PRNG.

In [Peinado et al. 2014], the authors, analyze the cryptographic security of J3Gen, a certified low cost RFID by the EPCglobal Gen2 for secure communications and general applications. The authors perform a deterministic attack by decimating the output PN sequence generated by the RFID. Fig. 11, presents the cryptanalysis of LFSR popularly used in all RFID authentication applications.

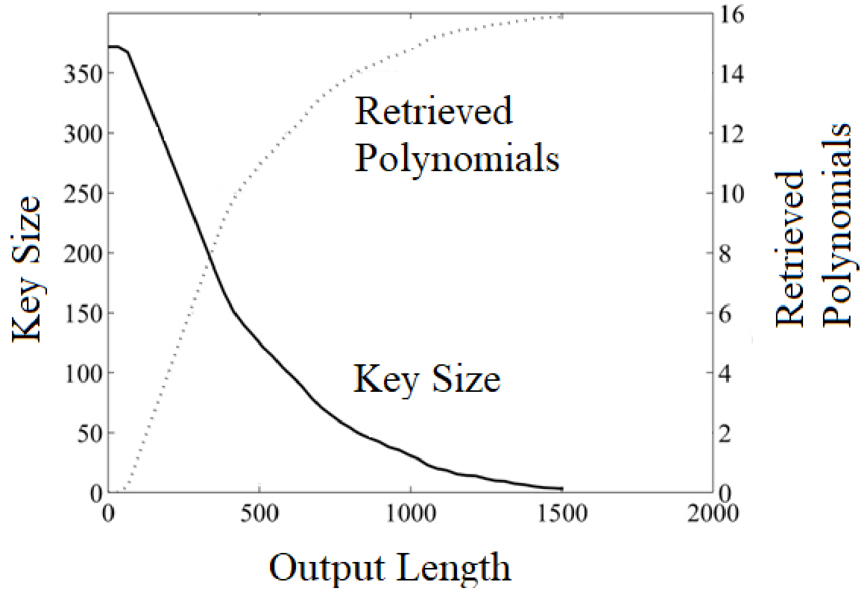


Fig. 11. Cryptanalysis of LFSR binary sequence [Peinado et al. 2014]

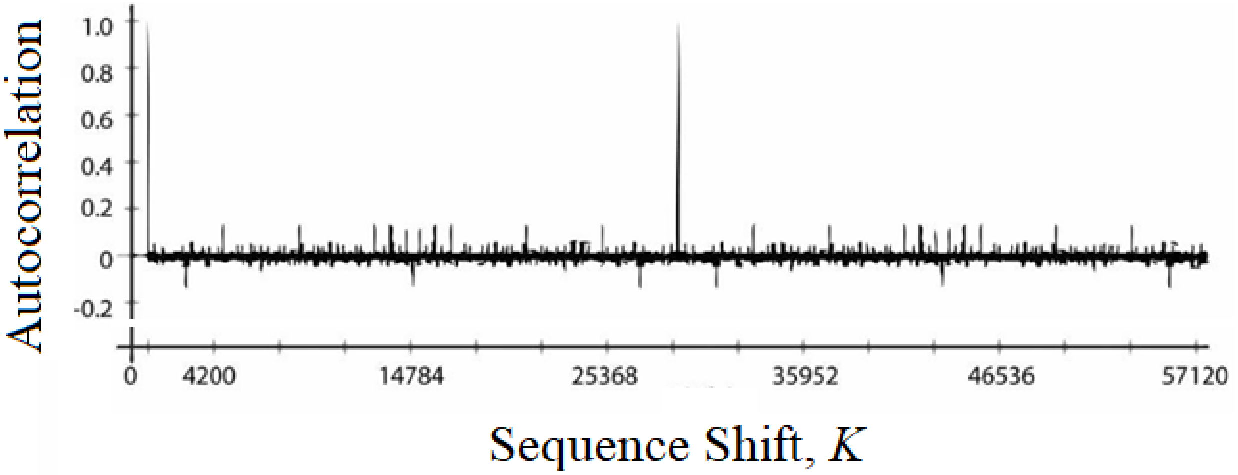


Fig. 12. Autocorrelation graph of the LFSR binary sequence [Peinado et al. 2014]

The autocorrelation of the sequence cryptanalyzed above is shown in the Fig. 12 below.

Considering the analysis described by [Peinado et al. 2014], we investigate an alternative way to generate a PN sequence of maximum length in this dissertation. Lightweight PNRGs, proposed in this dissertation, which are to be embedded into the secure application scheme, offer low latency, low complexity, and high security.

1.1 Research Goal

The specific goal of this dissertation is to identify methods to exploit the rare occurrence of prime numbers and convert that to the randomness of a new class of PN sequence to generate cryptographic keys for communications and network security. The objective is to develop complex theory-based, number theoretic approach to the computational hardening of PN sequences for the enhancement of existing cybersecurity systems.

1.2 Purpose of the Study

Our purpose is to design and test new classes of PRNGs for secure communications and cryptographic applications. Researchers have designed PRNGs by applying different techniques using shift registers and linear functions. No research has been conducted on generating PN sequences using primes sequence so far because of the asymptotic law of distribution of prime numbers. The standard assumption of "finding larger prime numbers is expensive" is false because primes sequences of large lengths can be generated by adding several shifted versions of variable-length prime sequences. This dissertation presents flexible and highly efficient means of generating PN sequences for secure communications and cryptographic applications.

1.3 Significance of the Study

The idea of using the random distribution of primes among integers to generate a new class of PN sequence is novel and adds value to the crypto science. The proposed research ensures new findings in number theory, probability theory, and in the practice of cryptography. This research develops three lightweight and cryptographically secure PRNGs useful for resource-constrained devices that have the potential to be extensively used in 5G wireless systems.

The computational cost of generating new classes of PN sequence is nominal and meets the advanced 5G wireless network systems requirements that demand reduced network energy usage algorithms to generate light-weight pseudorandom sequences.

CHAPTER 2

LITERATURE REVIEW ON PN SEQUENCES

Over the past half century, there has been a lot of research into constructing new families of PN sequences that are suitable for diverse fields of science and technology – mathematics, computer and information sciences, and information engineering.

In [Blum and Micali, 1986], the classical definition for a PN sequence has been mentioned as nonconstructive by [Martin-Lof, 1966], [Kolmogorov, 1965], and [Chaitin, 1966] since the sequences generated are not statistically tested. Various authors such as [Shamir, 1981], [Blum and Micali, 1986], and [Plumstead, 1982] performed a set of analyses on PRNGs to define a cryptographically secure *pseudorandom sequence*, which can be used in the practice of cryptography. The results of their basic analyses motivated others to concentrate on the subject of statistical testing of PN sequences.

The physical basis of randomness is an important problem [Landauer, 1996] and it takes us back to the very nature of information in quantum theory [Kak, 1999], [Kak, 2007]. The question of randomness is also fundamental in cognitive science and decision theory [Aerts, 2009], [Kak, 1996]. From a mathematical perspective, pseudorandomness may be related to the complexity of certain number-theoretic properties [Cassaigne et al. 1999], [Mauduit, 2002]. Specifically, one may seek to define the unexpectedness of some properties of composition of numbers [Chen, 1978], [Kak, 2014], [Nicolas and Robin, 1997] that yield a good randomness measure.

Probability theory and its extensions in stochastic process provide abstract notion of uncertain processes whose realization depends on a computer. PRNGs implemented on computers generate sequence of uniformly distributed independent random variables over the interval of [0, 1]. Such random sequences have applications in programming [Knuth, 1997], cryptography [Luby, 1996] and communications [Golomb and Gong, 2005] and they are useful in simulation experiments. These sequences are also important in areas beyond communications [Miguel and Juan, 2016]. Complex systems such as linguistics, social networks, finance and economic systems are characterized by self-similar or scale–invariant behavior that is random [Watts, 2003], and the behavior of such systems can be modeled by algorithm-generated random sequences that are transformed to correspond to the domain-specific distributions.

Physical-layer security using code division in wireless systems is based on the use of pseudorandom (PN) sequences [Viterbi, 1995]. Such sequences are usually investigated from the frequency and the complexity approaches [Golomb, 1982], [Kolmogorov, 1965]. They are important in cryptography due to their importance in cryptographic keys [Hoffstein et al, 2012], simulation, and in noise analysis in communication systems [Simon et al, 1994]. The applications of PN sequences to the physical-layer security begins with their generation that is usually achieved by using feedback shift registers [Golomb and Gang, 2005], [Golomb, 1982], [L'Ecuyer, 2012].

The applications of PN sequences in cryptography (key generation, multi-party communication), Monte Carlo simulation, e-commerce, computer games, casino machines, radar ranging is a major motivation in their further study. Most extensively used methods of generation of PN sequences in the existing literature are: linear feedback shift registers (LFSR) for single bit generation, series-parallel method for high speed generation, and linear congruence. M-sequences are widely used PN sequences because of the following factors: ideal autocorrelation, balance of

0s and 1s, and run property [Goresky and Klapper, 2009]. Recent advances use m-sequences in ultra-wideband sensing devices since they provide high-time stability and high-speed measurement [Sachs et al, 2007].

Walsh code or Walsh Hadamard produces orthogonal set of sequences; however, these codes do not possess good correlation values as *m*-sequences. m-sequences are basis to form Gold and Kasami sequences. Fig. 13 shows Gold sequence generator.

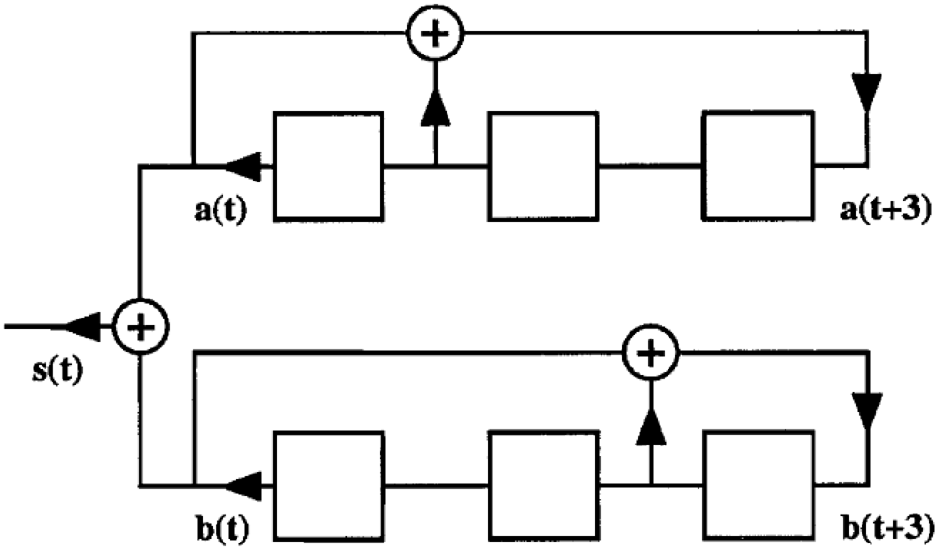


Fig. 13 Gold sequence generator

It is known that use of Gold or Kasami sequence in the place of m-sequence will improve the correlation and random measures [Reynen et al, 2010]. However, families of sequences generated using LFSRs are not the ideal choices for CDMA systems because of user traffic [Leon et al, 2004] [Schindler, 2013].

In recent years, cryptographers have paid an increasing attention to digital systems based on chaos theory [Dabal and Pelka, 2012] and have used chaotic PN signals to carry information. In chaos-based PN sequence generation, a chaotic dynamical system is used to produce real-valued (non-binary) chaotic sequences [Guo and Wang, 2010]. Although chaotic PN sequences have good statistical properties and high-data rates and are preferred for generating cryptographic keys, the complexity to synchronize sequences of communicating parties makes them unsuitable in most of cryptographic applications [Youssef, 2009]. Fig. 14 shows an overview of classification of various PN sequences.

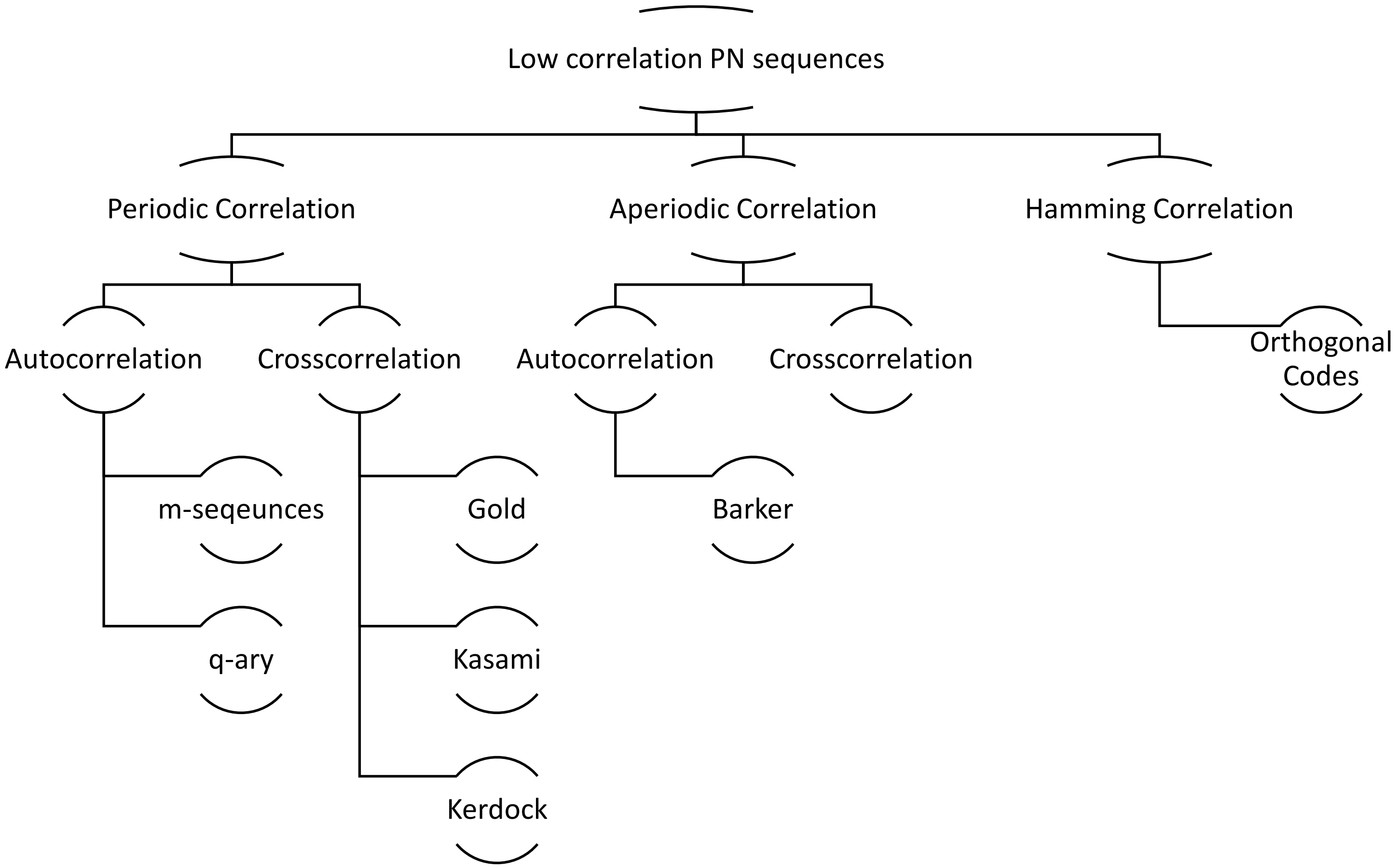


Fig. 14. Overview of PN sequences correlation values

2.1. Strategies for Testing Cryptographic Secure PRNGs

According to the German Federal Office for Information Security, the design of cryptographically secure PRNGs must meet the following criteria [Schindler, 2013]:

- Maximum period
- Should pass statistical tests and correlation tests
- Effort in guessing the sequence should be complex – from given subsequence or previous sequence

For theory and practice, a statistical test suite for PRNGs (hardware or software) has been developed by NIST with a package of 15 tests. These tests include are mentioned in the later chapters. But, [Wang, 2015] highlights the limitations of NIST SP800-22 due to Type II errors and the need for autocorrelation testing using robust tools if the sequences are used in cryptographic applications. The infamous RANDU failure is an evident example that randomness tests could only test the statistical properties and not all the randomness properties . The Mersenne Twister, which passed numerous tests including diehard tests, is not cryptographically secure. The method of [Schindler, 2013] is the default PRNG in most of all the programming software systems including Python, Matlab, MS Excel, Ruby, Stata, etc.

The existing literature strongly suggests that correlation testing determines the suitability of PN sequences for cryptographic applications [Dillon and Dobbertin, 2004] [Wang, 2015]. Correlation properties determine whether or not any selected subsequence contains information about the next element(s) in the sequence. PN sequences with low correlation and maximum period are significant in signal synchronization, secure communications, and cryptography [Helleseth and Kumar, 1999]. [Golomb and Gong, 2005] and [Helleseth and Kumar, 1998] studied signal design and the necessity to employ low-correlation PN sequences in various applications. Kasami-Welch (KW) sequences, KW-like sequences [Dillon and Dobbertin, 2004], Helleseth-Gong (HG) [Helleseth and Gong, 2002] sequences are of considerable interest in DS communication systems due to their low autocorrelation properties. The correlation characteristics of good PN sequences should be as near to purely random sequence as possible, and this requires that the autocorrelation function be nearly two-valued. Families of sequences with ideal autocorrelation property inlcude; m-sequences, Jacobi sequences, Gordon-Mills-Welch (GMW) sequences, Dillon sequences and Lin sequences.

In CDMA communication systems, PN sequences with low crosscorrelation are used to minimize interference, since every user is allocated with a unique chip sequence. Gold and larger Kasami sequences have high crosscorrelation compared to KW and HG sequences. Jacobi sequences and GMW sequences crosscorrelation values are higher when compared to KW-like sequences. In practice, due to data modulation, the correlation sequences tend to be aperiodic rather than being periodic. Hence, it is necessary to investigate aperiodic correlation properties of PN sequences.

Computational hardness is crucial to design and develop a CSPRNG. High speed Congruential PRNGs offers integer factorization [Panneton et al, 2006], PRNGs constructed by [Goldreich and Rosen, 2003] and [Micali-Schnorr, 1991] are based on RSA inversion problem, Chaotic-based PN sequence generators challenges adversaries with lattice problems, [Blum and Micali, 1986] and [Blum et al, 1996] proposed PRNGs which deals with discrete logarithm problem.

2.2. Conclusion

The need for research on finding new families of pseudorandom sequences is a part of the process of keeping one step ahead of eavesdroppers and they are important in the design of cryptographic keys that cannot be easily guessed by intruders. PRNGs have attractive feature of provable security generates sequences with minimal correlation values. No work has been reported generating PN sequences using the set of infinite primes numbers. Some previous studies focused on generating PN sequences using a fuzzy logic which is easy for adversaries to emulate.

CHAPTER 3

CRYPTOGRAPHIC SECURE HASH FUNCTION

Generating randomness is the foundation of many cryptographic protocols. It is crucial for all encryption schemes, especially the primitives such as hash functions and message authentication code. This dissertation presents two new class of PN sequences, which are designed to strengthen existing hashing functions for message authentication.

3.1. Initialization Vector

Cryptographic primitives require unpredictable fixed-size pseudorandom sequences as input, Initialization Vector (IV), prior to hash being generated. Numerous protocols make IVs public after the hash generation to establish accountability.

Wired Equivalent Privacy (WEP-40) encryption algorithm used 24-bit IV. Multiple usage of IVs having the same key was allowed in WEP-40, which compromised the encryption algorithm [Bristov et al, 2001]. This sudden trend gave way to WEP depreciation, as packet injection could allow WEP to be cracked in short intervals of time. However, in cipher-block chaining, CBC mode, the IV has to be unpredictable during the time of encryption, in addition to being distinctive. As shown by SSL 2.0, it is unsafe to use the IV of a previous message as the IV of the next message can be very insecure at times. The TLS CBC IV attack or the BEAST attack [Kurokawa et al, 2016] is an example of this method, where if an attacker has an idea about the IV of a previous block of a cipher text, he can verify that guess about the plaintext of some other block that had the encryption with the same key before.

3.1.1. Structure of Hash Function Using Initialization Vector

A hash function is a one-way function with variable inputs and fixed output. The variable input is divided into several variable sized *L* blocks, of *X* bits.

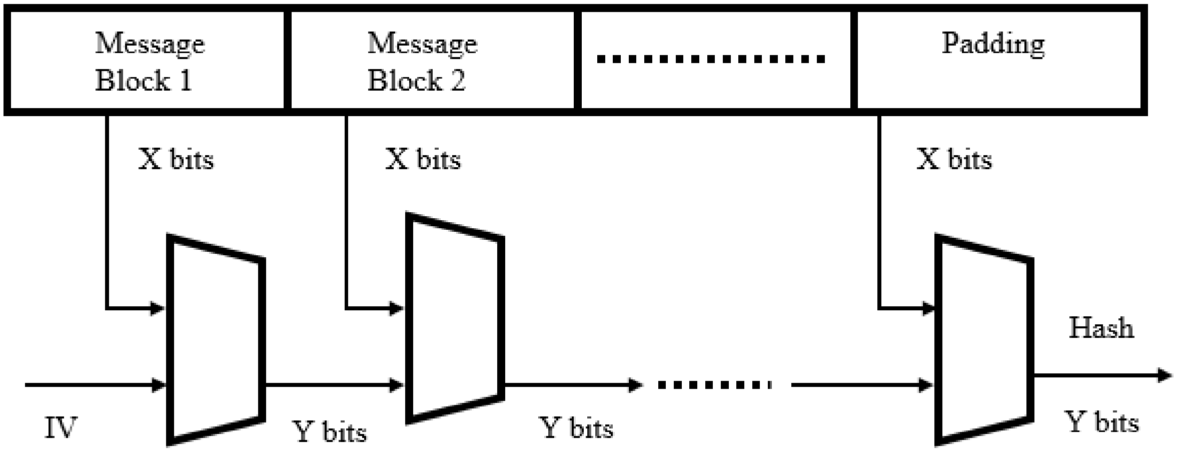


Fig. 15. The structure of hash function using Initialization Vector

At stage-1, a unique *n*-bit sequence called the Initialization Vector (IV) is supplied to initiate the hash function. The new classes of PN sequences investigated in the chapter 3 can be used as IVs to generate hash functions. Depending on the desired hash function output size, one could possibly use portions of the PN sequences proposed in the later chapter.

Since the message block size, *X*-bit, and the *Y*-bit input (or the IV) are different, a compression function *f* that processes the two inputs and produce a fixed *Y*-bit output is needed. Normally, the message block size greater than *Y*-bit. The function *f* process the two inputs multiple rounds to produce an output, which is fed as input to the next stages.

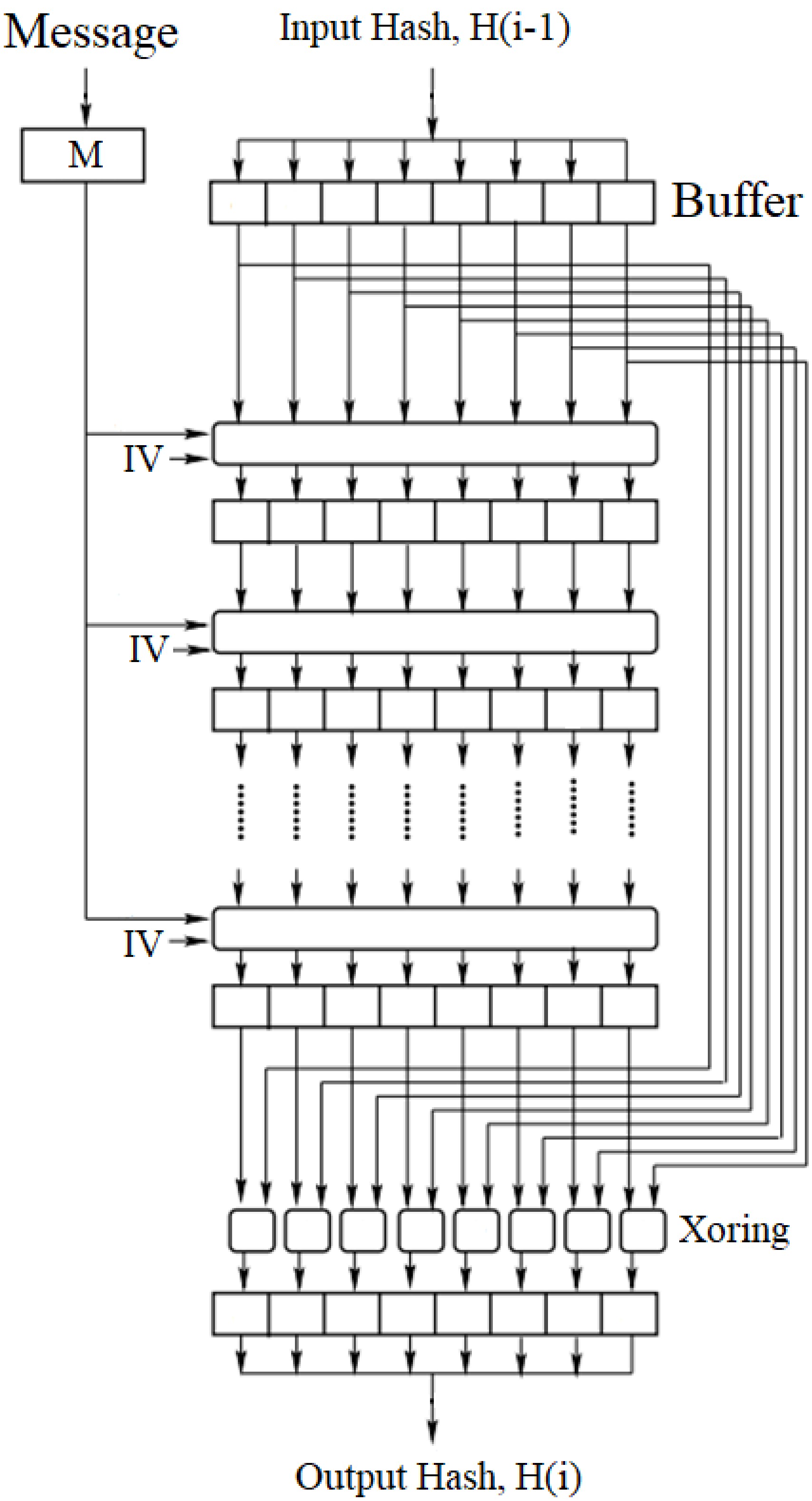


Fig. 16. The structure of compression function, f.

3.2. Hashing based on modular addition of PN sequences

Adding PN sequences to the *n*-bit input prior to the first stage of the hashing will increase the complexity of the resultant hash code. Such an addition not only increases the complexity but also counters cipher attacks such as BEAST attack and packet injection.

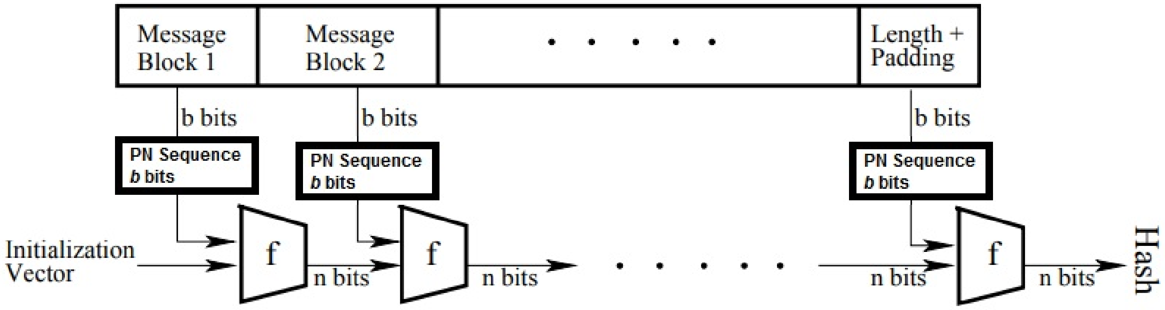


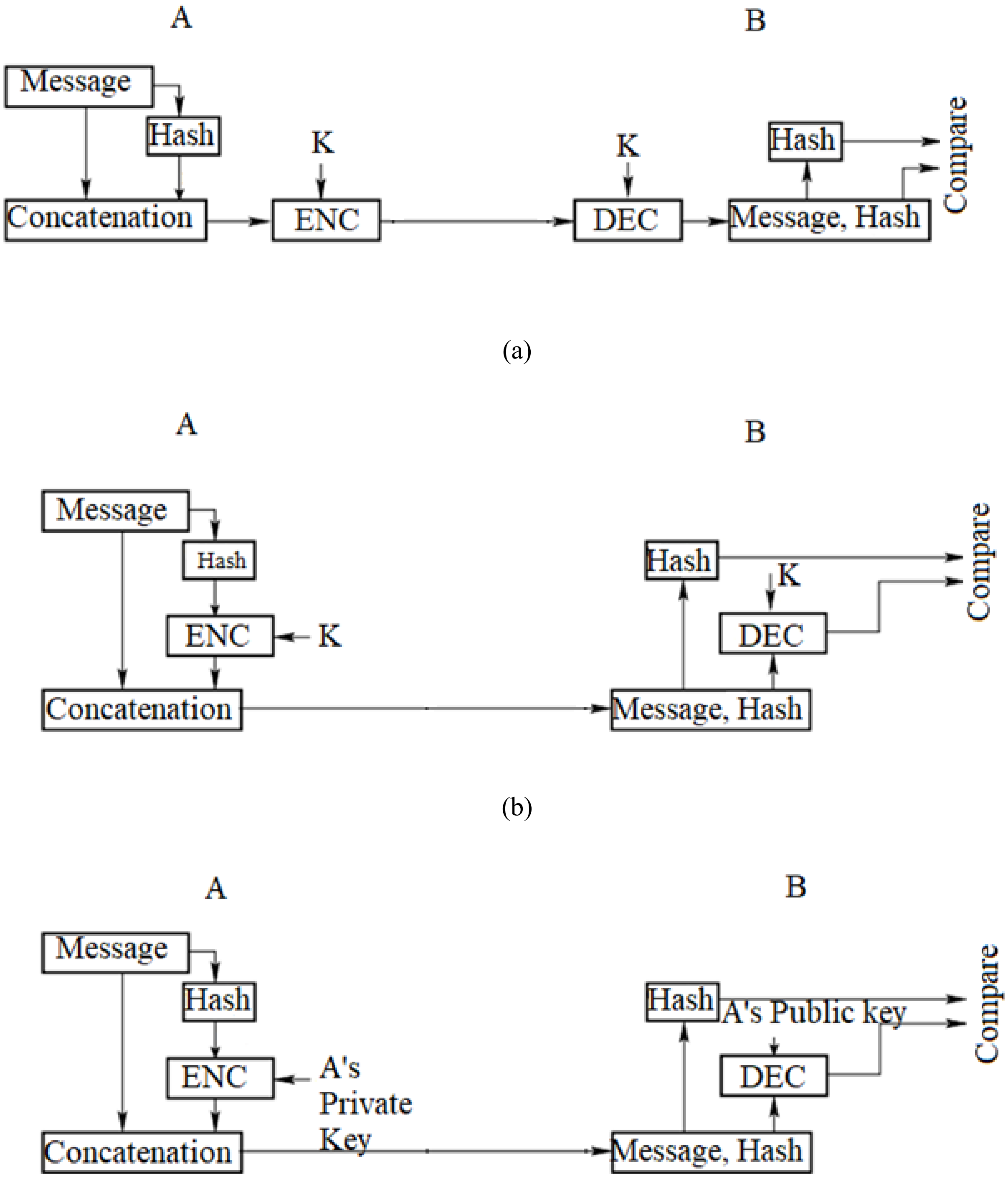
Fig. 17. The structure of hash function using modular addition of PN sequences

3.3. Multiple Ways of Hashing for Authentication

Fig. 18 shows six unique and multiple ways of using hashing for authentication in a communication channel. These techniques cover various ways of protecting the hash of a given message. This prevents the risk of message tampering from an adversary, in which case no authentication can be achieved at the receiving end.

- Fig. 18(a) shows a symmetric-key encryption based scheme, where before the message is transmitted, its hashcode is concatenated along with it and the full composite message is encrypted at once. The receiver on the other end separates the income hashcode after decrypting the message, and then compares it with the hashcode calculated from the received message. Thus the hashcode is fully authenticated and confidential.

- Fig. 18(b) shows a scheme where is message authentication of utmost importance but the confidentiality is not. Only the hashcode is encrypted here before sending the message. The receiver with the knowledge of secret key knows how to authentic hashcode of the message and thus the receiver can verify the authenticity of the message.
- Fig. 18(c) shows a scheme with a variation from the scheme shown in Fig. 18(b), in that it is a public-key encryption version. Confidentiality is not important here since the sender encrypts the hashcode of the message using his own private-key; however, the receiver utilizes the sender's public-key and recovers the hashcode using that, and he tests the authenticity of the message as it comes from the alleged sender.
- When both confidentiality and authenticity are needed, a symmetric-key algorithm can be added to the approach shown in figure 18(c). Fig. 18(d) shows this very commonly used approach.
- Fig. 18(e) gives a unique take on the situation, where neither the message nor the hashcode is encrypted. While sending the message, the sender appends a shared secret S to the message before calculating its hashcode. The receiver knows this too, and he adds the same secret to the original message before checking for authentication of the hashcode of the message. This method prevents any adversary from altering the message, even when they obtain the original message and the hashcode.
- Lastly, Fig. 18(f) shows an extended usage of the scheme from Fig. 18(e), where symmetric-key based confidentiality is appended to transmit the message between the sender and the receiver.


A
B
Message
Hash
Concatenation
K
ENC
K
DEC
Hash
Message, Hash
Compare

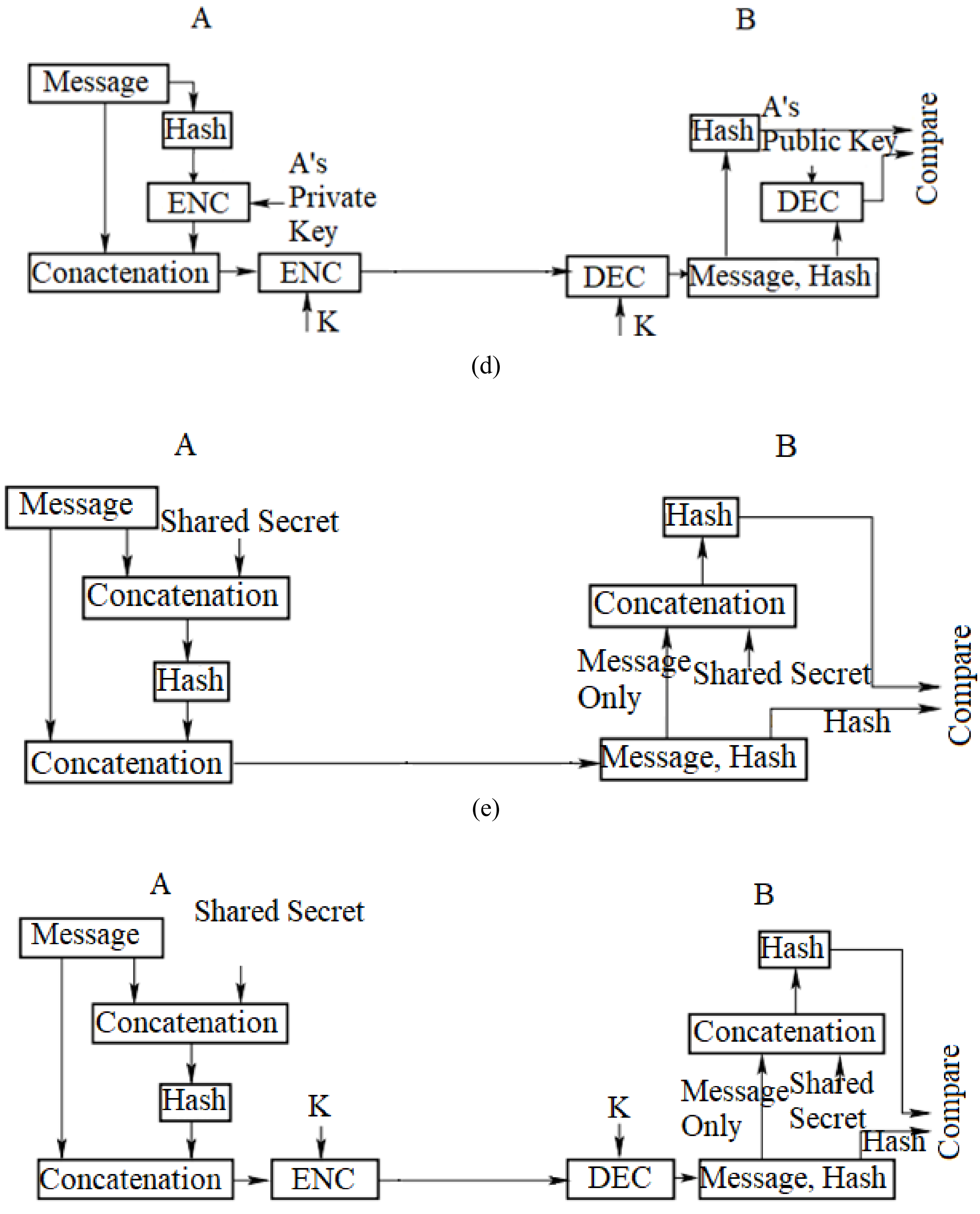
(a)

A
B
Message
Hash
ENC
K
Concatenation
Hash
K
DEC
Message, Hash
Compare

(b)

A
B
Message
Hash
ENC
A's Private Key
Concatenation
Hash
A's Public key
DEC
Message, Hash
Compare

(c)



(d)

(e)

(f)

Fig. 18. Multiple approaches of hashing.

CHAPTER 4

NEW CLASS OF PSEUDORANDOM SEQUENCES

This chapter proposes the use of PN D-sequences that have gone through an additional random mapping for the design of cryptographic keys. These sequences are generated by starting with inverse prime expansions in base 3 and then replacing 2 in the expansion with either the string 01 or 10 based on the preceding bit, which represents a general mapping. We show that the resulting pseudorandom sequences have excellent autocorrelation properties. Such a method can be extended to inverse prime expansions to any base.

The second section investigates the generalized binary primes sequence and proposes its use to enhance commonly used PN sequences in physical-layer security applications. It is shown that the bitwise addition of the generalized binary primes sequence to a PN sequence results in a sequence that is cryptographically stronger. The properties of the proposed sum sequences are compared to other established binary sequences. The problem of guessing the sequence by the attacker is shown to be the order of $O(N^N)$. The complexity properties of the generalized binary primes sequence are examined.

4.1. Recursive D-Sequences

This section investigates the use of a new class of PN D-sequences [Kak, 1981], [Kak, 1985] that are generated by starting with inverse expansions in radix 3 and then replacing 2 in the expansion with either the string 01 or 10 based on the preceding bit and the proposed idea can be extended to sequences in any base. These random number generators are efficient, although by their very nature they must be periodic. Their use is ubiquitous in information systems [Garay et al. 2000] [Rocha et al. 2011].

D-sequences have finite periodic length which is generated by the decimal expansion of the inverse prime number q in a modulo r division. The sequence a_i is outputted according to the rule:

$$a_i = \left[r^i \bmod q\right] \bmod r$$

where q is a prime number. These sequences are strings of bits generated in a very convenient and efficient manner. Our goal is to perform another mapping

$$p_i = f(a_i | a_{i-1})$$

where the output binary bit b_i depends on the previous output value. The function $f(.)$ can take a variety of forms depending upon our objectives. Here we consider an elegant function where it chooses from one of two substrings based on the previous digit produced by the mapping. We would like to do this in a manner that increases the cryptographic strength of the b-string.

As is well-known, inverse prime expansions (D- or decimal sequences) possess good autocorrelation properties and so they are excellent candidates for pseudorandom sequences. But they have an obvious structural redundancy that is a disadvantage: for a maximum length decimal sequence, the bit sequence in the first half of the period is the complement of the second half of the period. This dissertation addresses this limitation by generating binary sequences by starting with ternary sequences and then replacing the 2s by 0s and 1s in a manner that the frequencies of the bits are not changed. Such a process can be applied to decimal expansions in any base and, therefore, the ternary representation should be seen only as an example of the method.

4.1.1. Unbalanced Ternary D- Sequences

If the base selected for the D-sequence is r = 3, then the sequence generated will be a ternary sequence of 2s, 1s and 0s. Ternary pseudo-random D-sequences are generated using a formula similar to (1) as below:

$$a_i = \left[3^i \bmod q\right] \bmod 3$$

where q is a prime number. The maximum length (q-1) sequences are generated when 3 is a primitive root of q.

The autocorrelation function $r_{i,j}(\tau)$, for the unbalanced ternary D-sequence is calculated using the below formula:

$$r_{i,j}(\tau) = \frac{1}{N}\sum_{i=1}^{N=1} a(k) * a(k+\tau)$$

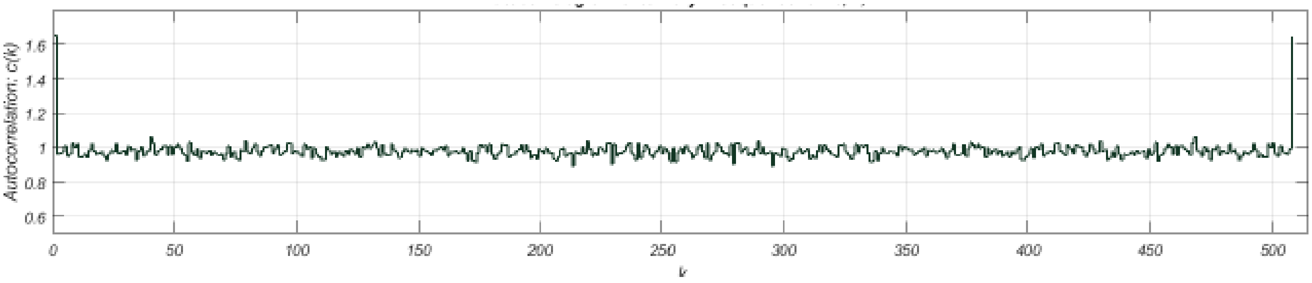


Fig. 19. Unbalanced ternary D-sequence autocorrelation for prime number 509

Fig. 19. presents the autocorrelation function of an unbalanced ternary PN D-sequence considering a prime number 509. As we know from decimal sequence theory since the period is 508, each of the three digits 0, 1, and 2 is almost equally likely. For all values of a_i, $a_i \in \{0, 1, 2\}$. Since the digits have equal probability, the peak value of the autocorrelation function is $\frac{1}{3}\sum_{i=0}^{2} i^2 = 1.667$.

It is clear that the D-sequences are not cryptographically strong, but that is not the property that is essential in generating cryptographic keys. In any event, our objective is to strengthen these sequences by means of an additional mapping (*b_i*).

4.1.2. Balanced Ternary D- Sequences

Due to the computational convenience, balanced ternary sequences (where the 2s have been replaced by -1s) are preferred when compared to unbalanced ternary sequences. Also, we can find better autocorrelation properties when the ai sequence normalized to stream of bits containing 0s, 1s and -1s (where 2s are represented as -1s). Fig. 20. presents the autocorrelation function of an balanced ternary PN D-sequence considering a prime number 509.

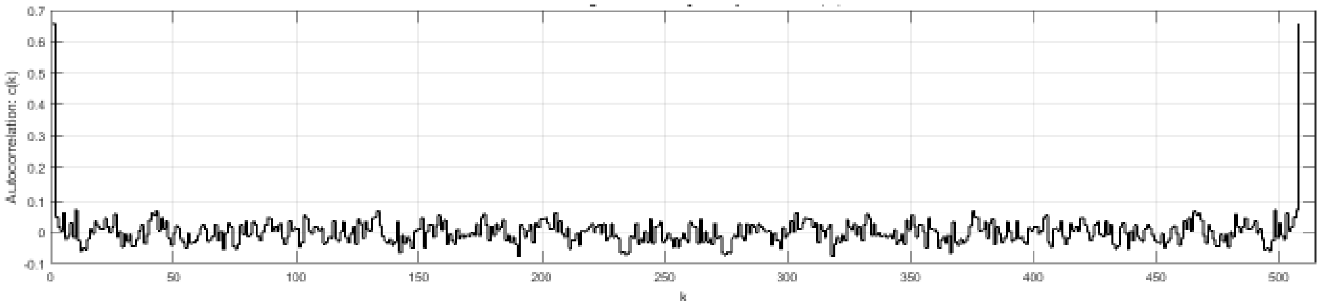
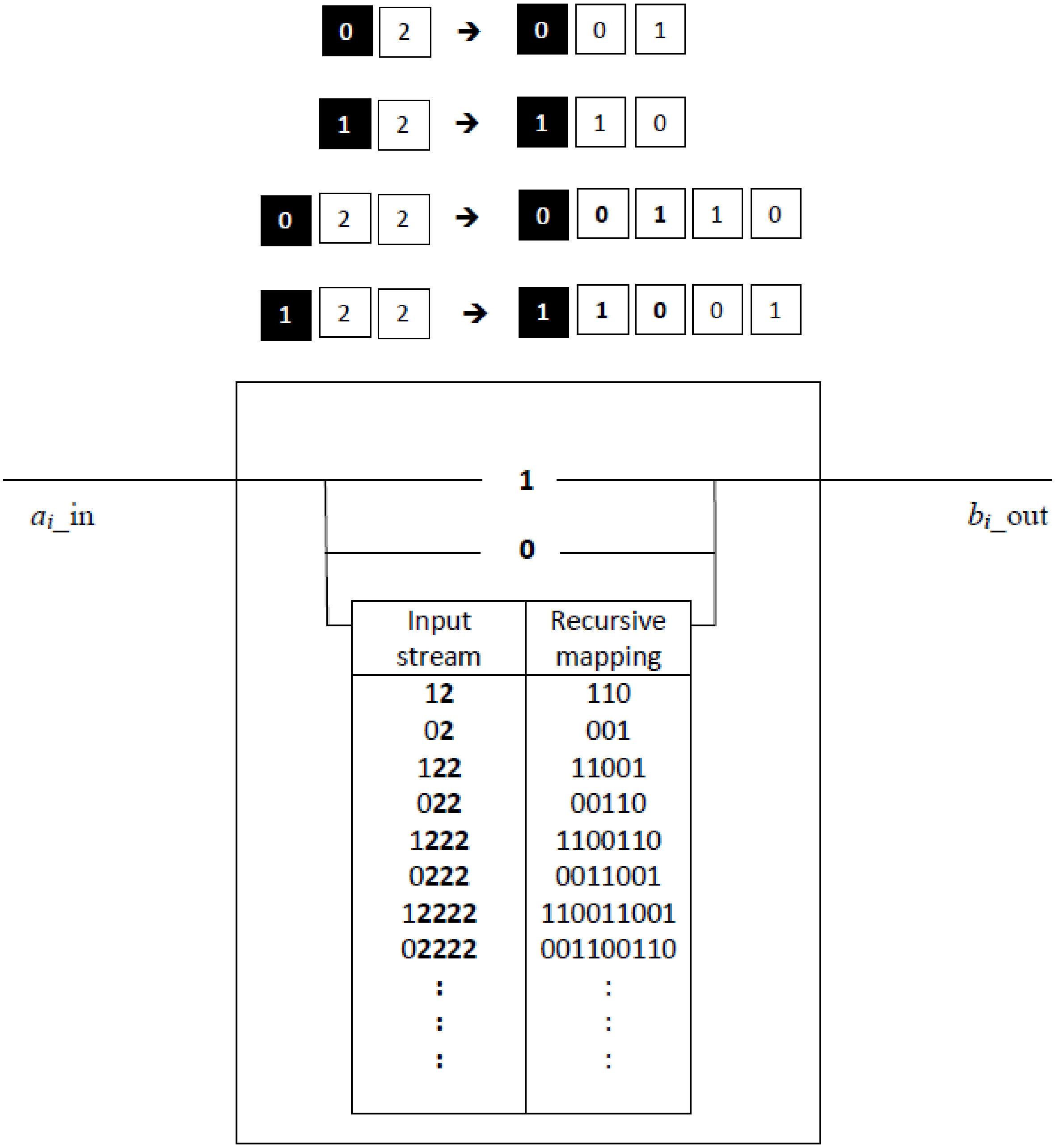


Fig. 20. Balanced ternary D-sequence autocorrelation for prime number 509

Although the autocorrelation function looks better its peak has been reduced to 0.6 due to $\frac{1}{3}\sum_{i=-1}^{1} i^2 = 0.667$. To overcome this limitation, we propose to replace the 2s of the ternary sequence by 01 or 10 based on the preceding bit.

4.1.3. Unbalanced Ternary P- Sequences

Here in this approach, we will replace encountered 2 in a_i bit stream with ‘01’ if it’s preceded value in the bit stream is 0, or ‘10’ if its preceded value in bit stream is 1. When substrings of several 2s are encountered, then the above scheme of mapping can be used recursively as shown in Fig. 21.



Input stream	Recursive mapping
12	110
02	001
122	11001
022	00110
1222	1100110
0222	0011001
12222	110011001
02222	001100110
:	:
:	:
:	:

Fig. 21. Bit stream mapping

By this approach, one can generate a variable sized bit stream for a_i by challenging adversaries to guess the periodic length of the underlying ternary D-sequence.

Illustration: For a prime number 7, a_i = [0 2 0 1 2 1] with a periodic length of 6 and its normalized ternary sequence will be p_i = [0 0 1 0 1 1 0 1] with a periodic length of 8. Likewise, the a -sequence [1 0 2 2 1 1 0] will be transformed to the p -sequence [1 0 0 1 2 1 1 0] which in the next step to [1 0 0 1 1 0 1 1 0].

Fig. 22, presents the autocorrelation properties for this ternary PN D-sequence with the peak autocorrelation of 0.5 because of the unbalanced characteristics of the sequence.

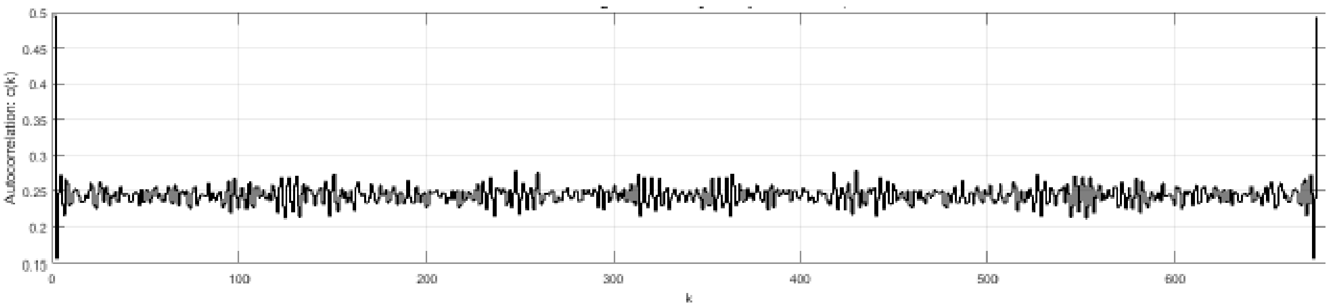


Fig. 22. Unbalanced ternary P-sequence autocorrelation for prime number 691

Generating a predictable pseudorandom sequence such as a decimal sequence, or exposing even a few bits of random sequence in each of several digital signatures, may suffice to obtain the private key since such sequences are not cryptographically strong. Clearly, the additional nonlinearity introduced in the sequence will make the task cryptographically harder.

4.1.4. Balanced Ternary P- Sequences

The method of the use of ternary b-sequences will provide no opportunity for adversaries to estimate the periodic length of P-sequences. Variable periodic length is advantageous in dealing with adversaries.

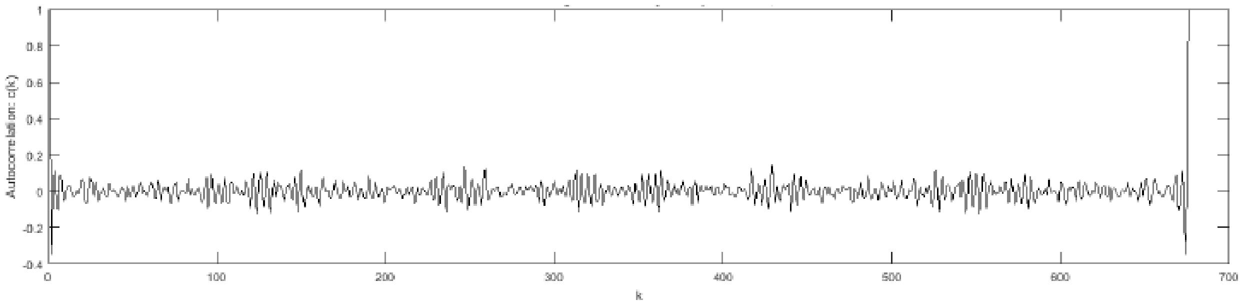


Fig. 23. Balanced ternary D-sequence autocorrelation for prime number 691

The p_i sequence is also represented in the balanced form where 0s have been mapped into -1s. Fig. 24 presents the autocorrelation properties for a new class of ternary pseudo-random D-sequence when the recursive bit stream is balanced with better autocorrelation.

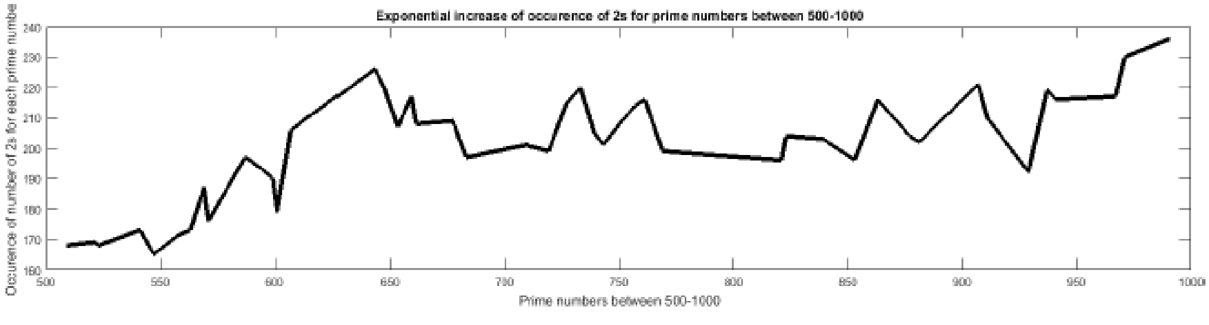
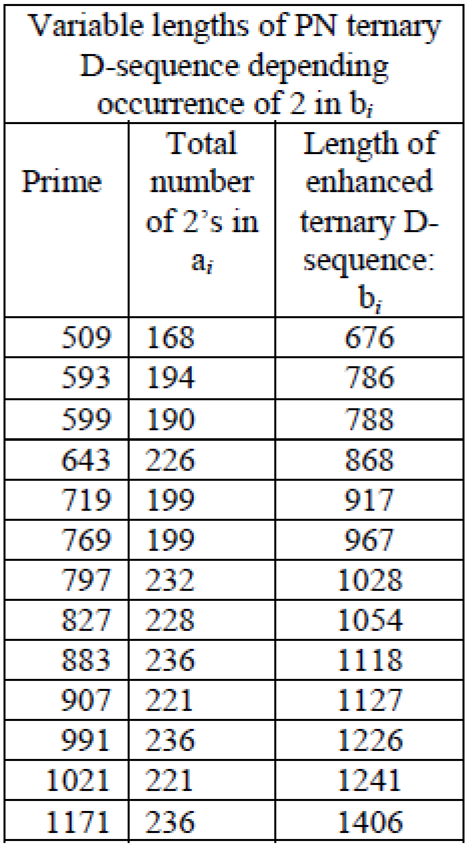


Fig. 24. Increase of 2s in p_i for each prime number between 500-1000

Table 3. Calculated lengths of enhanced ternary D-sequences



Variable lengths of PN ternary D-sequence depending occurrence of 2 in b_i		
Prime	Total number of 2's in a_i	Length of enhanced ternary D-sequence: b_i
509	168	676
593	194	786
599	190	788
643	226	868
719	199	917
769	199	967
797	232	1028
827	228	1054
883	236	1118
907	221	1127
991	236	1226
1021	221	1241
1171	236	1406

The increase is not systematic since the patterns of consecutive 2s will be encountered in an unexpected way and, furthermore, not all sequences are maximum length as the primes are increased. This is at the basis of the increased cryptographic strength of the *p*-sequences.

4.1.5. Conclusion

We proposed the use of ternary pseudo-random D-sequences, together with a mapping to take the 2s into 0s and 1s that leads to a new class of binary random sequences that can be good candidates for cryptographic keys. The major advantage of the proposed class of sequences is the flexibility in the choice of the period. This method of pseudorandom sequence generation need not be limited to ternary expansions and can be used for expansion to any radix with a subsequent mapping to binary.

4.2. Binary Primes Sequences

This section deals with the idea of using the random distribution of primes [Busireddygari and Kak. 2017] among integers to generate a binary primes sequence and use it to strengthen PN sequences in physical-layer security applications. To overcome the problem of the rareness of prime numbers several shifted versions of the binary primes sequence are introduced and added together to generate the generalized binary primes sequence, B sequence, in which the number of 0s and 1s are approximately balanced. We show that the proposed sequences have excellent correlation properties and are cryptographically strong. The section is organized as follows. Section 4.2.1 presents the B sequence construction and its characteristic properties. Section 4.2.2 proposes the computational hardening of B sequence by adding it to other PN sequences. Section 4.2.3 discusses the quality parameters of B sequence and its comparison with other PN sequences prominent in the literature. Section 4.2.4 provides a conclusion with different perspectives.

4.2.1. The B Sequence

The binary primes sequence is defined as:

$$b(k) = \begin{cases} 1, k = prime \\ 0, k \neq prime \end{cases}$$

The primes sequence is unbalanced with the distribution of 0s and 1s. According to the prime number theorem [Hardy and Wright, 1954], the number of prime numbers less than an integer *N* is calculated using the below formula

$$\pi(N) \sim \frac{N}{\ln N}$$

When we examine first 5000 natural numbers, the number of primes less than 5000 is 669 and hence the primes sequence has 669 1s and 4331 0s. We balance the primes sequence by adding multiple shifted versions of the original primes sequence with itself.

The number of shifters, L, required to generate the binary primes sequence is given by

$$\frac{N}{2} * \frac{\ln N}{N} = \frac{1}{2} \ln N$$

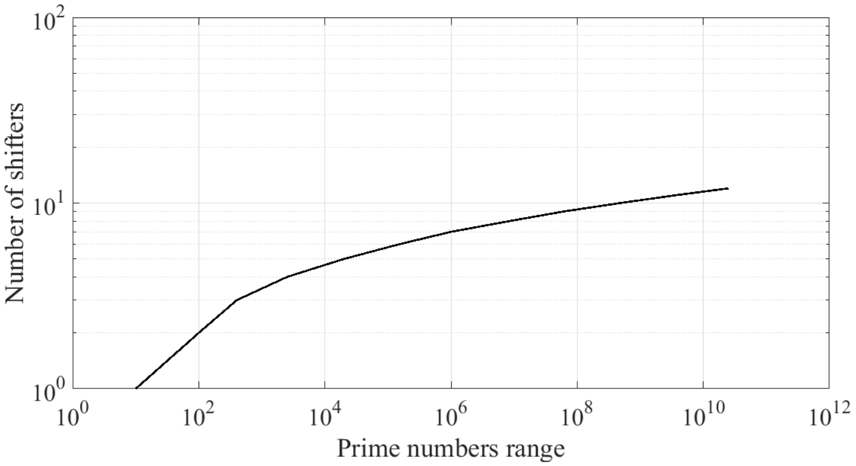


Fig. 25. The number of shifters required to generate a B sequence of given length

The generalized binary primes sequence is called as B sequence or $B_{N,L}(k)$, which is the sum of multiple binary primes sequences of shifts units $a_0, a_1, \cdots a_L$. $a_0 = 0$ always remains as the unshifted version of primes sequence, $b(k)$.

$$B_{N,L}(k) = \sum_{i=0}^{L} b(k - a_i)$$

The B sequence has a very high degree of uniformity with a well-balanced distribution, and non-regular occurrence of binary values.

Table 4 and Table 5 shows the efficient utilization of one or two shifters to produce a balanced B sequence. Although, a shifter can have variable shifts in practical implementations. The shifters employed here are right shifters to shift the prime sequence, $b(k)$, right by one unit.

Table 4. B sequence for prime number 17 (L = 1)

Sequence		1s	0s
$b(k)$	0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0 1	7	10
$b(k-1)$	0 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0	6	11
$B_{N,L}(k)$	0 1 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1	11	6

Table 5. B sequence for prime number 17 ($L=2$)

Sequence		1s	0s
$b(k)$	0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0 1	7	10
$b(k-1)$	0 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0	6	11
$b(k-2)$	0 0 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0	6	11
$B_{N,L}(k)$	0 1 0 0 0 1 0 1 1 0 1 1 0 1 1 0 1	9	8

4.2.1.1. Autocorrelation of B Sequence

The autocorrelation function $r_{i,j}(\tau)$, for the B sequence is calculated using the below formula:

$$r_{i,j}(\tau) = \frac{1}{N} \sum_{i=1}^{N=1} B_{N,L}(k) * B_{N,L}(k + \tau)$$

where $B_{N,L}(k)$ denotes the non-delayed version of maximum length B sequence, and $B_{N,L}(k + \tau)$ denotes the delayed version of generalized binary primes sequence by τ units.

The correlation measures are used to test the randomness and study the statistical properties of the random sequences. The autocorrelation values of the B sequence are low with noise like properties making it suitable for cryptography applications.

Fig. 26. depicts the autocorrelation of the B sequence for prime number 991 employing 3 shifters units with $a_1 = 11$, $a_2 = 55$, $a_3 = 222$. The absolute non-zero peak value of $r_{i,j}(\tau)$ is observed to be 0.3133 for the given prime number.

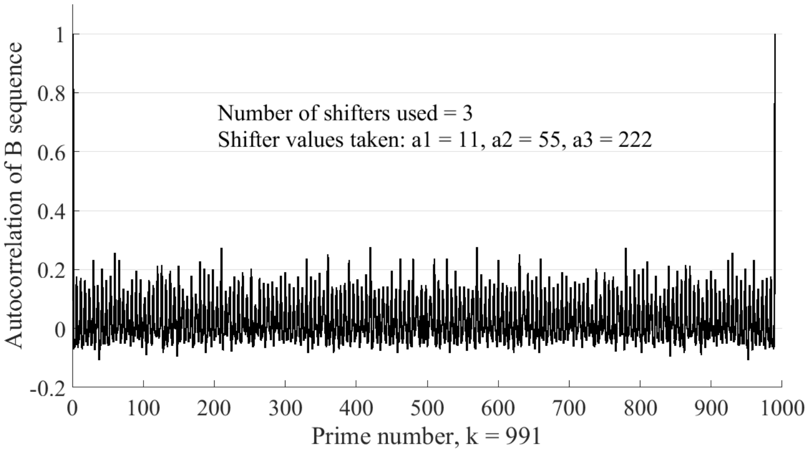


Fig. 26. The autocorrelation graph of the B sequence for prime number 991

4.2.1.2. Randomness Measure for B Sequence

The randomness accuracy for the B sequence, is evaluated using the below formula [Kak, 1970]:

$$R_N(B) = 1 - \frac{\sum_{l=1}^{N-1} |r_{i,j}(l)|}{N-1}$$

The randomness of the B sequence increases with period shown in Fig. 27. The peak randomness measure for the B sequence, $B_{N,L}(k) = B_{487,4}(0,10,50,100,150) = 0.9949$.

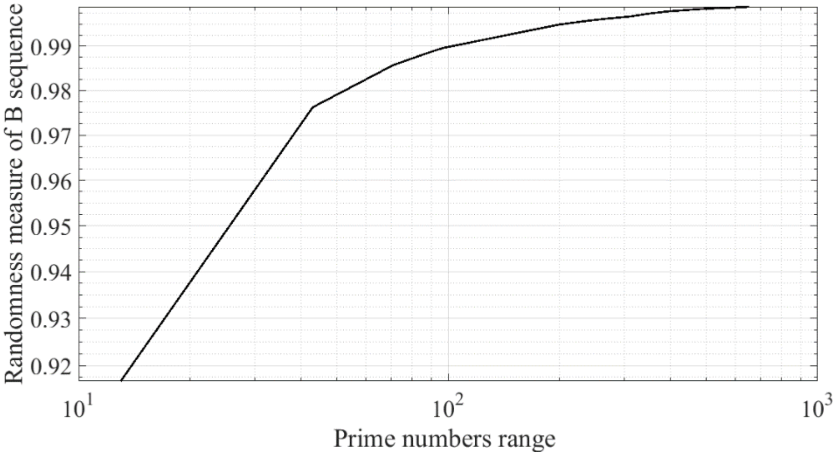


Fig. 27. The B sequence accuracy in terms of randomness

4.2.2. Adding the B Sequence to Other PN Sequences

We show that the addition of B sequence to other PN sequences leads to the computational hardening of the resultant sequence. This is consistent with such similar strengthening that has been observed elsewhere [Brown and Solomon, 1979].

4.2.2.1. The Binary S Sequence

Our first candidate sequences is the binary decimal sequence, S which may be generated by the following expression [Kak, 1985]:

$$s(i) = (2^i \bmod q) \bmod 2$$

where q is a prime number. It is known that the S sequences possess good mutual distance property [Kak, 1981], [Kak, 1985]:

$$C(j) \leq \frac{1}{3}, j \neq 0$$

$$2^m > q, k = q - 1, m = j + 1; j < q$$

where d_j is the hamming distance between the binary maximum length S sequence and its j^{th} cyclic shifts.

The autocorrelation function $C(j)$ for a S sequence, in the symmetric form is given by

$$d_j \geq \frac{k}{m}, j \neq 0, j < k$$

where S sequences have good autocorrelation properties, although they are weak for cryptographic applications.

When we add the B sequence to a PN sequence, the resultant sequence shall be called as the P sequence, $P(k)$. Below we add it to the binary decimal sequence, S:

$$P(k) = B_{N,L}(k) + s(k)$$

Upon computing the autocorrelation function we are most interested in observing the average of the off-peak magnitudes.

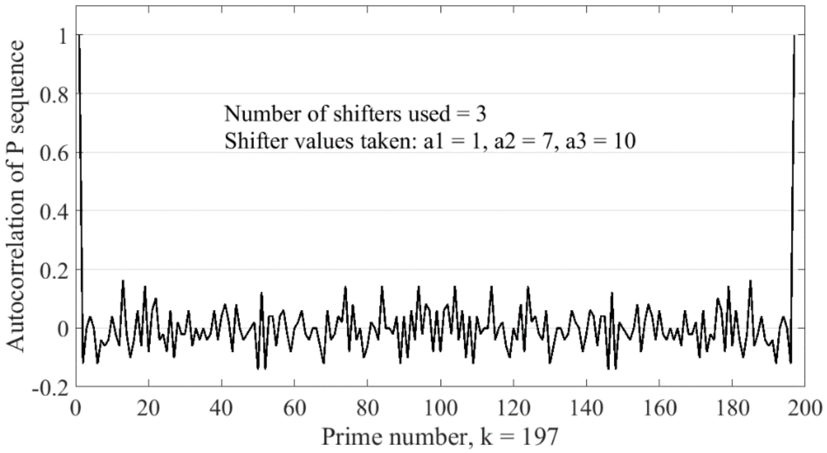


Fig. 28. The autocorrelation of P sequence for prime number 197

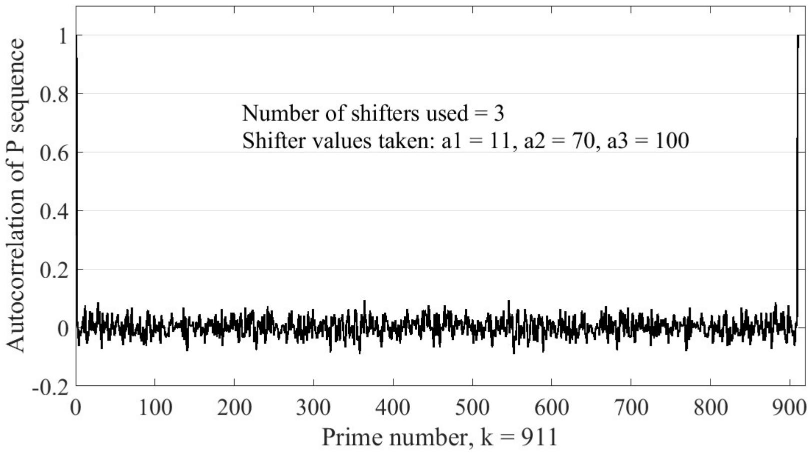


Fig. 29. The autocorrelation of P sequence for prime number 911

Fig. 28. presents the autocorrelation graph of P sequence for the prime number 199. $B_{N,L}(k) = B_{197,3}(0,1,7,10) = b(k)+b(k-1)+b(k-7)+b(k-10)$ and in Fig. 29. presents the autocorrelation graph of P sequence for the prime number 911. $B_{N,L}(k) \ = B_{911,3}(0,11,70,100) = b(k) + b(k - 11) + b(k - 70) + b(k - 100)$. It is evident that in both graphs the autocorrelations magnitudes are exceptional.

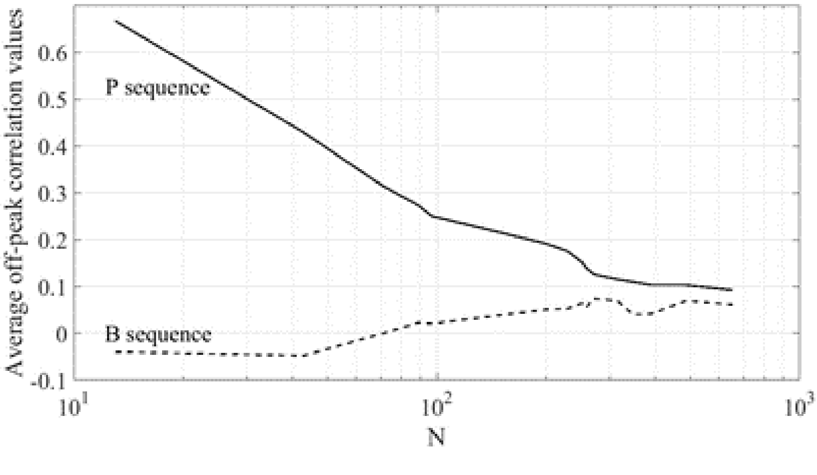


Fig. 30. Comparison of average off-peak curves for the B sequence and the P sequence

Fig. 30. depicts the graph of variation of the average off-peak magnitudes of the B sequence and the P sequence for $N = 50$ to $N = 650$. As expected, the randomness of the P sequence increases when measured in terms of the average autocorrelation off-peaks. Fig. 30. shows that the off-peak values of the B sequence and the P sequence converge at certain point as the prime number range increases.

4.2.2.2. The Valence Sequence

The valency, $v(k)$, of a number $k = P_1^{t_1} P_2^{t_2} P_3^{t_3} \cdots P_k^{t_k}$ is defined as [Kak, 2012], [Kolmogorov, 1965]

$$v(k) = t_1 + t_2 + t_3 + \cdots + t_k$$

when $v(k) = 1$, the number is prime. Note that

$$\delta(k) = \frac{d(k)}{2}$$

where the number of divisors, $d(k)$, for a given number equals $(t_1 + 1) * (t_2 + 1) * (t_3 + 1) * \cdots * (t_k + 1)$.

The binary sequence of $v(k)$ is given by the function $l(\text{k})$ is defined as [Kak, 2012]

$$l(\text{k}) = (-1)^{\delta(k)}$$

We add the B sequence to the valence sequence and the resultant PN sequence is the P sequence, $P(k)$.

$$P(k) = B_{N,L}(k) + v(k)$$

Fig. 31. presents the autocorrelation graph of the resultant sequence for the sum of the B and V sequences for the prime number 911. $B_{N,L}(k) = B_{911,3}(0,11,55,222) = b(k)+b(k-11)+b(k-55)+ b(k-222)$.

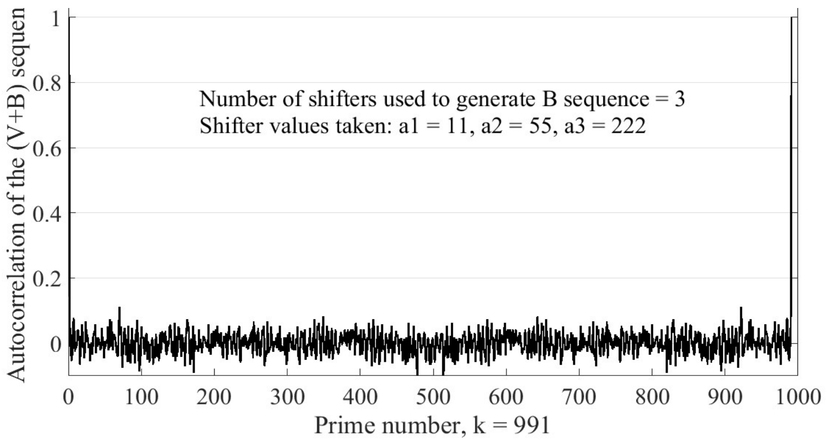


Fig. 31. The autocorrelation of the generalized binary primes sequence for $N = 991$

4.2.3. Results and Discussion

The Mean Square Aperiodic Autocorrelation (MSAAC) and Mean Square Aperiodic Crosscorrelation (MSACC) are the most popular and widely accepted performance metrics to analyze the correlation properties and randomness of PN sequences.

The aperiodic correlation function of a sequence is defined in equation 5.

The MSAAC measure for a code set with M sequences of maximum length N is given by:

$$MSAAC = \frac{1}{M}\sum_{i=1}^{M}\sum_{\tau=1-N;\,\tau\neq 0}^{N-1} |r_{i,j}(\tau)|^2$$

Similarly, the MSACC measure is defined as:

$$MSACC = \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{j=1;j\neq i}^{M}\sum_{\tau=1-N}^{N-1} |\, r_{i,j}(\tau)|^2$$

The sequences with less MSAAC measure corresponds to less correlation between the bits with in a given sample of the sequence, and the sequences with less MSACC correspondingly implies less correlation between samples of different sequences [Mauduit, 2002].

Table 6 presents the comparison of MSAAC and MSACC measures of various sequences described in the paper and other classes that are established in the literature. For all the sequences, the codes length has been taken as 199 bits.

Table 6. Correlation measures for various PN sequences

Sequence	MSAAC	MSACC
B	0.418	0.645
S	0.581	0.783
Valence	0.495	0.801
P (B+S)	0.362	0.590
P (B+V)	0.406	0.480
Gold	0.846	0.928
Small Kasami	0.547	0.766
Large Kasami	0.732	0.601

Table 6 shows that the B sequences have superior MSAAC and MSACC measures because there are no significant peaks in the autocorrelation function. The case B+V was superior to B+S. These measures show that the proposed sequences are superior to Gold, Small Kasami, and Large Kasami sequences.

We now consider a computational complexity view of the B sequence to see its use as future prime-numbers-based PRNG [Aiello et al, 1998], [Chugunkov et al, 2016]. From the attacker's point of view, the deduction of the B sequence offers several challenges. The generation of B sequence involves selecting variable *N*, *L* and shifter values. Considering the a_i values as arbitrary values, the effort of the attacker to compute the similar B sequence becomes extremely hard as the value of *N* increases exponentially.

The three unknown variables utilized in the process of generation of B sequence are

1) The range of prime numbers chosen
2) The number of shifter components (dependent on the chosen prime number)
3) The variable shifters values

If the B sequence is computed using a prime integer, *N*, then the computational complexity for an attacker is $\frac{1}{2} * N^{\frac{\ln N+N}{\ln N}}$. This complexity is based on the following considerations:

1) The number of primes less than an integer *N* is given by $\frac{N}{\ln N}$
2) The number of shifters required used to generate a balanced B sequence is given by $\frac{1}{2}\ln N$
3) Variable shift values considered to the compute the B sequence is given by $N^{\frac{N}{\ln N}}$

Thus the total number of possibilities for an attacker to guess the B sequence is represented by $\frac{N^{\frac{N+\ln N}{\ln N}}}{2}$. An attempt to guess all the above-mentioned computing factors is thus prohibitively difficult.

The proposed B sequences may be used for other cryptographic applications. The very low autocorrelation and crosscorrelation measures demonstrate the strength of the sequences in code-division applications. They can be used by both sending and receiving parties to constitute a symmetric encryption method.

4.2.4. NIST Suite Test

The NIST Test Suite focus on 15 tests to determine statistical randomness of sequences generated using either software algorithms or hardware PNRGs. The test suite requires arbitrarily long binary sequences as input to test various randomness properties that may exist in a sequence. The following are tests the test suite include:

1) The Entropy Test
2) The Matrix Test
3) The Cumulative Sums Test
4) The Frequency Test in a given Block
5) The Linear Complexity Test
6) The Mono-bit Test
7) The Non-overlapping Template Test
8) The Overlapping Template Test
9) The Random Excursions Test
10) The Random Excursions Variant Test
11) The Runs Test
12) The Spectral Test
13) The Serial Test
14) The Longest Run
15) Maurer's Test

1. The Entropy Test:

The goal of entropy test is to examine the frequency of all probable overlapping M-bit substrings of the sequence across the whole sequence. The focus of entropy test is to collate the frequency of overlapping blocks against the expected result for a given binary random sequence.

2. The Matrix Test:

The aim of matrix test is to rank disjoint sub-matrices of a random sequence. The focus of this test is to evaluate the linear dependency among fixed-length sub-sequences of the whole sequence. This test is included in the DIEHARD tests also.

3. The Cumulative Sums Test:

The aim of cumulative sums test is to estimate the maximum excursion of the random walk calculated by the cumulative sum of normalized binary digits (1 ➔ +1, 0 ➔ -1) in a sequence. For an ideal random sequence, the random walk excursions should be zero.

4. The Frequency Test within a given Block:

The frequency test within a given block test determines is the distribution of 1s within M-bit blocks. The focus in this test is to verify whether the occurrence of 1s in an M-bit block is close to M/2, which is the ideal case for the assumption of randomness. When the given block size, M, equal to 1 the test digress to 1.

5. The Linear Complexity Test:

The aim of the linear complexity test is find out the length of a LFSR that generates the sequence. This test examines whether or not the binary sequence has maximum length to determine the randomness.

6.. The Mono-bit Test:

The mono-bit test, otherwise called as frequency test, determines the distribution of 0s and 1s for the given whole sequence. The focus of the frequency test is to verify whether the occurrence of 1s and 0s in a given binary sequence is balanced or not. The test evaluates the occurrence of ones approximately to 0.5, i.e., there should be a balanced distribution of number of 1s and 0s. This test is the most important and all other following tests be dependent on passing this test.

7. The Non-overlapping Template Test:

The goal of non-overlapping template test is to detect the number of occurrences of same bit streams in the entire sequence. This test detects RNGs that produce occurrences of aperiodic bit stream pattern of sequence. For this test, a non-overlapping M-bit window is set to search for a particular M-bit stream. If the bit stream is not detected, the non-overlapping window slides one bit position towards the right. If the bit stream is detected, the matching window will be reset to one bit towards the right after the bit stream pattern is detected, and then continues the search.

8. The Overlapping Template Test:

The difference between the overlapping and the non-overlapping template test is that when the bit stream pattern is detected, the matching window slides one bit towards the left before completing the search over the entire binary sequence.

9. The Random Excursions Test:

The purpose of random excursions test is to find out the number of cycles possessing Q visits in a cumulative sum random walk.

10. The Random Excursions Variant Test:

The goal in the random excursions variant test is investigate the total number of times a particular state anticipated in a cumulative sum random walk.

11. The Runs Test:

The aim of runs test is find out the total number of runs - an uninterrupted string of identical bits, Q - in the binary sequence. The focus of runs test defines whether the number of runs of 1s and 0s of variable lengths is occurring too often or less.

12. The Spectral Test:

The goal of spectral test is to find out the peak DFT values of the sequence. The interest in finding out the peak values is to detect periodic properties in the binary sequence, which gives an in-depth knowledge on the deviation from the ideal randomness.

13. The Serial Test:

The goal of serial test is to find out the frequency of all probable overlapping M-bit patterns across the complete binary sequence. When M = 1, the serial test is same as the frequency test.

14. The Longest Run:

This test is crucial because a non-uniformity in the expected length of the longest run of 1s impacts the uniformity of the expected length of the longest run of 0s.

15. Maurer's Test:

The Maurer’s test finds out the number of bits between matching pattern or bit stream. The significance of this test is to determine whether the binary sequence can be compressed with no impact on the loss of information.

Table 7. NIST suite test results for generalized binary primes (B+S) sequence

Test Name	Times	P-Value	Proportion
Frequency	1	0.87452	0.9159
Block frequency	1	0.000941	0.9357
Cumulative Sums	3	0.635892	0.9467
Runs	1	0.624553	0.9241
Longest run	1	0.465923	0.9782
Matrix	1	0.822562	0.9453
Spectral	1	0.000000	0.9810
Non-overlapping template	198	0.000126	0.9810
Overlapping template	1	0.932653	0.9810
Maurer’s	1	0.621556	0.9476
Entropy	1	0.002547	0.9147
Random excursions variant	2	0.957463	0.9914
Serial	10	0.589671	0.8924
Linear complexity	28	0.547254	0.9945

Table 7 presents various statistical measures of generalized binary primes sequence. The NIST, ENT and Diehard statistical test suites were recommended to validate random measures of PRNGs. Specifically, NIST suite test the cryptographic strength of PRNGs.

4.2.5. Conclusion

In this paper, the B sequence was used for computational hardening of weak PN sequences. We showed that the autocorrelation measures of the resulting sequence are excellent when the component sequences are added together. The computational cost of adding the B sequence to other PN sequences is nominal because it can be pre-computed and stored in the application, and the user simply selects the shift value to be used. The shift value may also be used to represent a cryptographic key that can be agreed upon and shared between two users by using an appropriate key-exchange protocol.

Concluding, the efficiency of the resultant PN sequence is high and therefore it can be used for a variety of physical-layer security applications.

CHAPTER 5

BINARY TABLEAU SEQUENCES

This chapter proposes a new class of random sequences called binary primes tableau (*BPT*) sequences [Busireddygari and Kak. 2017] that have potential applications in cryptography and communications. The *BPT* sequence of rank *p* is obtained from numbers arranged in a tableau with *p* columns where primes are marked off until each column has at least one prime and where the column entries are added modulo 2. We also examine the dual to the *BPT* sequence obtained by adding the rows of the tableau. It is shown that *BPT* sequences have excellent autocorrelation properties. In the general case, the starting number of the tableau sequence may be arbitrary, which increases the complexity of the sequence.

In a recent paper, we showed that binary primes sequence can be used for computational hardening of PN sequences [Reddy and Kak, 2016]. Here we go beyond that idea and generate PN sequences by folding the binary primes sequences using *p* columns.

The folding is done by starting with the number zero ordered in a tableau with a prime number of elements in each row. Next, all composite numbers are marked as 0 and the primes themselves are marked as 1. We stop the process once each column has a 1. The columns are added mod 2 to yield a 0 or 1 in each position. If the rows are added we get the dual sequence.

We discuss in this chapter that such tableau sequences have outstanding autocorrelation properties and, therefore, they can be used in communications applications to generate cryptographic keys and in CDMA systems. They can also be used in system simulation applications.

5.1. The Numbers Tableau

Definition 1. *The Numbers Tableau (NT$_P$) of rank p is a table with p columns listed as 0 to p-1, where the entries in the tableau in the k^{th} row are the natural numbers from (k-1)p through (kp-1).* An example of such a tableau is given below.

$$\begin{vmatrix} 0 & \cdots & \cdots & \cdots & p-1 \\ p & \cdots & \cdots & \cdots & 2p-1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (k-1)p & \cdots & \cdots & \cdots & kp-1 \end{vmatrix}$$

NT$_P$ is thus a way of representing the integers starting from 0 in *p* columns.

Example 1. *When p = 3 the NT$_P$ is*

$$\begin{vmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots \end{vmatrix}$$

Definition 2. *The Binary Primes Tableau (BPT$_P$) is a table with p columns listed as 0 to p-1, where the entry is 0 if the number is composite and 1 if it is prime.*

Example 2. *Let p = 5; the BPT$_P$ for this case is given by*

$$\begin{vmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{vmatrix}$$

5.2. The Binary Primes Tableau Sequence

Definition 3. *Complete Binary Primes Tableau (CBPT$_P$) is a finite termination of the BPT$_P$ when there is a 1 in each of the columns of the BPT$_P$.*

There are two options for this termination which lead to what we call sequences of the first kind and the second kind. In the sequences of the first kind, once the prime number in the last row has been recorded, the remaining entries in the last row are put equal to zero (such a termination will be used in the remainder of the dissertation). In the sequences of the second kind, we let the sequence continue until the end of the row.

Example 3. *Let $p = 7$. The CBPT$_P$ of the first kind is constructed as*

$$\begin{vmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \boxed{1} & & & & & \end{vmatrix}$$

The one marked by a square in this matrix fulfills the condition that each column have 1.

Example 4. *Let $p = 7$. The CBPT$_P$ of the second kind is shown as*

$$\begin{vmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \boxed{1} & 0 & 0 & 0 \end{vmatrix}$$

Note that the tableau of the second kind has an additional 1 in the last row.

Definition 4. *Binary Primes Tableau Sequence (BPT): The sequence obtained by adding* $CBPT_P$ *entries column-wise* mod *2, will be called a Binary Primes Tableau sequence.*

Theorem 1. *As p becomes large, the BPT sequence is random from an information complexity point of view.*

Proof. We have p columns in the tableau which has mapped nearly $p \times s \cong N$ integers (we use the adjective nearly since the last row may not be full). Note that, s need not be a prime. This means that each column has 1s that can range for a singleton to multiple values. Therefore, what is being done is to put the primes obtained until almost $p \times s$ has been reached into p partitions where each partition has at least one prime. The number of these primes are approximately $\frac{N}{\log N} \cong m$ by the prime number theorem.

If we look at the problem from the perspective that primes are distinguishable but the partitions are not, then the count of the cases will be given by the Stirling number of the second kind. A Stirling number measures the ways to partition a set of m objects into k non-empty subsets and it is denoted by $S(m, k)$ or $\genfrac\{\}{0pt}{}{m}{k}$. The k subsets here may be compared to the number of columns and m is the number of primes that are being considered until each column is filled.

We know that

$$\sum_{k=0}^{m} \left\{ {m \atop k} \right\} x(x-1)(x-2)\dots(x-k+1) = x^m$$

$$\left\{ {m \atop k} \right\} = \sum_{j=1}^{k} (-1)^{k-j} \frac{j^{m-1}}{(j-1)!\,(k-j)!} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^m$$

Since $k = p$, the value of $\left\{ {m \atop k} \right\}$ increases exponentially and this will be a measure of the difficulty faced by the attacker in detecting the sequence.

If we view the count from the perspective of number of primes and not their value, then the total number of ways these primes can be distributed is $\binom{N-1}{p-1}$, which will also increase exponentially.

The fact that the Stirling numbers increase rapidly highlights the large information complexity aspect of the BPT sequence. □

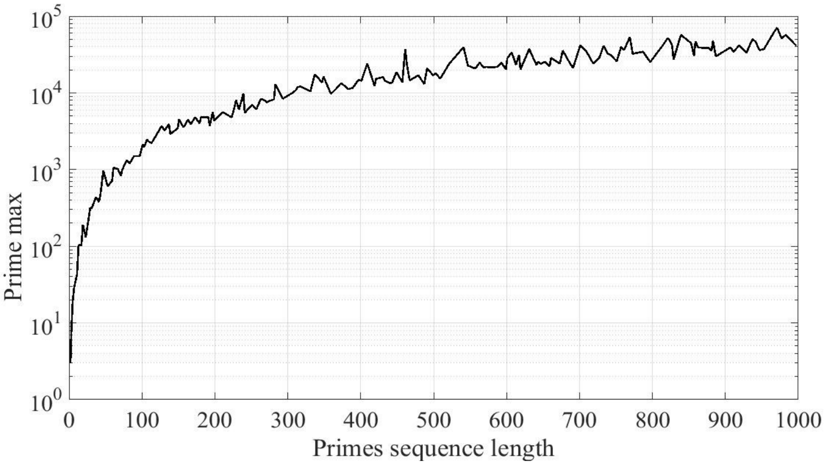


Fig. 32. The variation of $N = p \times s$ with the largest prime

Definition 5. *The Dual Tableau sequence (DT): The dual to the BPT is the sequence that is obtained by adding CBPT$_P$ entries row-wise* mod *2.*

The DT sequence is complementary to the BPT sequence. Therefore, its properties will mirror those of the BPT sequence.

Theorem 2. *As p becomes large, the DT sequence is random from an information complexity point of view.*

Example 5. *Let p = 13. The BPT and DT for p is constructed as*

$$\begin{array}{cl} \begin{vmatrix} 0&0&1&1&0&1&0&1&0&0&0&1&0 \\ 1&0&0&0&1&0&1&0&0&0&1&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&1&0 \\ 0&0&1&0&1&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&1&0&1&0&0&0 \\ 0&0&1&0&0&0&1&0&1&0&0&0&0 \\ 0&1&0&0&0&1&1&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1&0&0&0&1&0&1 \end{vmatrix} & \left.\begin{matrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix}\right\} \textbf{DT Sequence} \\ \underbrace{\begin{matrix} 1&0&1&0&0&1&1&0&0&1&0&1&1 \end{matrix}}_{\textbf{BPT Sequence}} & \end{array}$$

In this example the *DT* sequence, which has a length of 8 is not balanced because the sequence is very short. As *p* increases the sequence becomes more balanced and random.

Let the weight of the sequence be *k*, then the ratio $\frac{k}{p}$ gives a measure of balance. For an even length sequence (as may be true for *DT*), the weight should be half the length for perfect balance. When the length is odd (as for *BPT*), the sequence will be considered perfectly balanced if $k = \frac{p\pm1}{2}$. Thus for *p* = 13, a weight of either 6 or 7 would mean perfect balance.

Definition 6. *The length of the DT sequence will be called as chunk size.*

Fig. 33 is a plot of how the chunk size changes as *n* is increased to 1000.

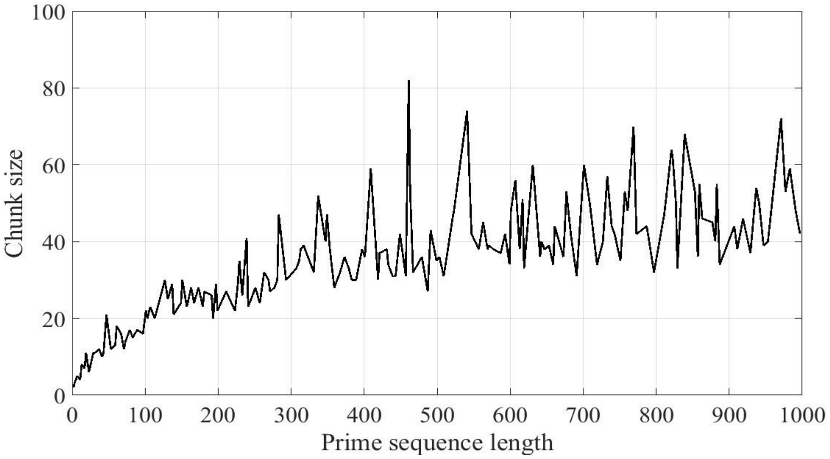


Fig. 33. Chunk size variation

The highly irregular variation in the chunk size reflects the randomness of the tableau sequence.

5.3. Properties of the BPT Sequence and the Dual Sequence

5.3.1. Balance Property

The BPT sequence has a near-balanced distribution of 0s and 1s. The ratio of 1s to the length of the sequence is approximately equal to 0.53 and 0.52 for BPT and DT sequences for the range of primes that have been tested. Table 8 summarizes this distribution. As *p* increases the ratio of 1s and 0s in both BPT and DT become approximately equal.

Table 8. The measure of balance in *BPT* and *DT* sequence

Prime	*BPT*	*DT*
13	0.461	0.125
199	0.532	0.681
461	0.527	0.672
971	0.522	0.465
997	0.534	0.523

The range of the last non-zero prime number in the NT_P to generate the BPT sequence is unpredictable and remains as a challenge for the attacker to find.

The maximum periodic length of the BPT sequence is likely to have only one single cycle based on the complexity structure of the sequences. Therefore, we do not expect any symmetry in the distribution of 0s and 1s.

5.3.2. Correlation Properties

The mean square periodic autocorrelation (MSPAC) and mean square periodic crosscorrelation (MSPCC) are useful to study the randomness properties of random sequences. Sequences with low autocorrelation and crosscorrelation measurements are important from a security point of view [Ziani and Medouri, 2015].

We calculate the periodic correlation function for a sequence by shifting the non-delayed version of sequence with the delayed version cyclically and it is defined as:

$$r_{i,j}(k) = \frac{1}{n}\sum_{i=1}^{n-1} C_i(n) * C_j(n+k)$$

where n is the length of the BPT or DT sequence.

The MSPAC measure provides an estimate of the randomness. The lower the value more random the sequence is.

$$MSPAC = \frac{1}{u}\sum_{i=1}^{u}\sum_{k=1-n;\, k\neq 0}^{n-1} |r_{i,j}(k)|^2$$

We compute the crosscorrelation function of a sequence taking the LCM. of the two sequences of different lengths (u and v). In such computation, we let the shorter sequence run through multiple periods until it match the length of the larger sequence.

The MSPCC measurement is defined as:

$$MSPCC = \frac{1}{u(u-1)} \sum_{i=1}^{u} \sum_{j=1;j\neq i}^{u} \sum_{k=1-v}^{v-1} |\, r_{i,j}(k)|^2$$

For computational convenience, we use the polar version of the BPT sequence (where 0s are converted -1s) to determine the correlation functions.

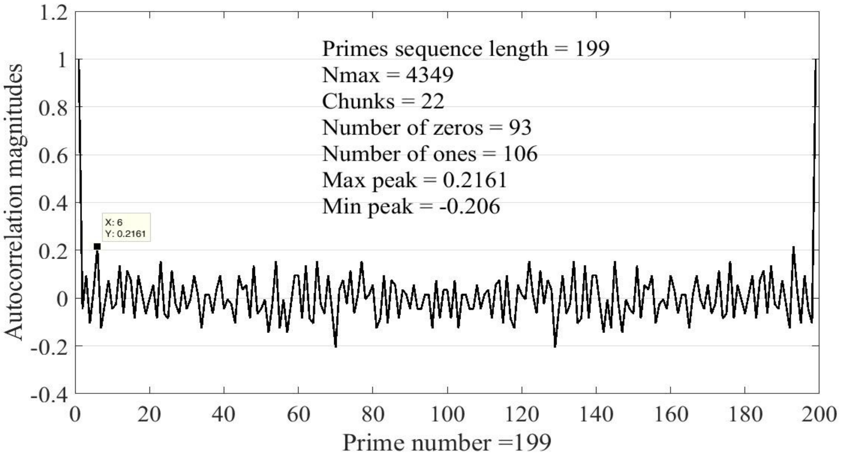


Fig. 34. Autocorrelation function of BPT sequence for prime number 199

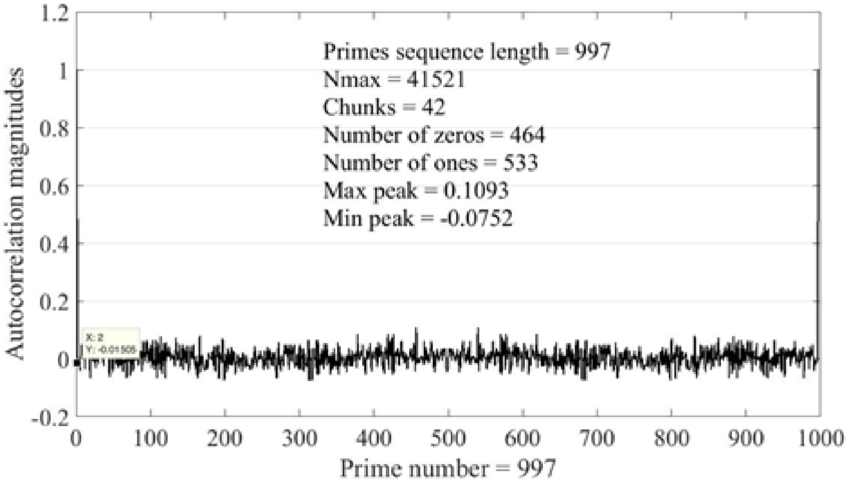


Fig. 35. Autocorrelation function of BPT sequence for prime number 997

Table 9 compares the autocorrelation and crosscorrelation peaks measurements of BPT sequence and other classes that are established in the literature. For all the sequences, the code length is taken to be 199 bits.

From Table 9, we notice BPT has the best MSPAC and MSPCC values compared to other sequences.

Table 9. Correlation measures for various PN sequences

Sequence	MSPAC	MSPCC
BPT	0.116	0.205
Gold	0.846	0.928
Small Kasami	0.547	0.766
Large Kasami	0.732	0.601

5.3.3. NIST Suite Test

Table 10. NIST suite test results for BPT sequence

Test Name	Times	P-Value	Proportion
Frequency	1	0.892999	0.9640
Block frequency	1	0.000923	0.9587
Cumulative Sums	1	0.736222	0.9856
Runs	1	0.520441	0.9990
Longest run	1	0.497330	0.9941
Rank	1	0.723438	0.9112
Spectral	1	0.000000	0.9990
Non-overlapping template	198	0.000120	0.9990
Overlapping template	1	0.989115	0.9990
Maurer’s	1	0.561111	0.9899
Approximate entropy	1	0.001228	0.9558
Random excursions variant	2	0.984622	0.9924
Serial	10	0.482167	0.8991
Linear complexity	28	0.489481	0.9978

5.4. Generalized Tableau Sequences

The theory presented in the previous sections may be generalized. The tableau may be started with any random number but in that case the column that is a multiple of *p* is left out. This is done because the numbers in the column will remain composite.

Definition 7. *The Generalized Number Tableau GNT(g) starts with number g in complete NT_p where the column consisting of multiples of p is cast out.*

The BPT sequence for the GNT(g) may be generated in a manner which is analogous to previous constructions. Note that the length of the final sequence will be $p - 1$.

Example 6. *Let g = 25 and p = 7. The BPT(g) starts with 25 and the column that has multiples of 7 is deleted.*

$$\begin{vmatrix} 25 & 26 & 27 & \boxed{29} & 30 & \boxed{31} \\ 32 & 33 & 34 & 36 & \boxed{37} & 38 \\ 39 & 40 & \boxed{41} & \boxed{43} & 44 & 45 \\ 46 & \boxed{47} & 48 & 50 & 51 & 52 \\ \boxed{53} & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Replacing primes and composite numbers by 1s and 0s in the tableau, we obtain the following tableau.

$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \boxed{1} & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

By adding each column mod 2 will generate the generalized tableau sequence.

It is clear that the above method can be used to generate an infinity of binary tableau sequences. The attacker will now have not only have to determine p but also the starting number g.

5.5. Conclusion

This Chapter proposes a new class of random sequences called binary primes tableau (BPT) sequences that have potential applications in cryptography and communications.

The BPT sequence of rank *p* is obtained from numbers arranged in a tableau with p columns where primes are marked off until each column has at least one prime and where the column entries are added modulo 2. We investigated the dual to the PT sequences obtained by adding the rows of the tableau.

It was shown that PT sequences have excellent autocorrelation properties. When the tableau begins with an arbitrary number we obtain the generalization of the basic BPT sequence.

CHAPTER 6

AUTHENTICATION OF PARTIES USING HASH CODES

Authentication of parties in computer networks is concerned with identity verification to check whether the claimed sender's identity is legitimate or not [Burrows et al. 1990], [Meadows, 2003]. In addition to public key cryptography authentication, there is also the Certification Authority-based authentication which involves the trusted third party's certification in verifying the identity of communicating party's public key on-line or off-line.

Most extensively used message authentication codes (MAC) are obtained from hash functions of the secret message. Automata theory is used to implement MAC because it offers convenience to design types that are suitable for cybersecurity and cryptography, and there exist many ways to model and test security protocols [Alur and Dill, 1994],[Rey, 2007], [Mukherjee et al. 2002], [Oliveria, 2010], [Wolfram, 1986]. In this paper, we model authentication of communicating parties using the input-output automata version [Blundo et al. 2004],[Durgin et al. 2004].

The idea of Piggy Bank (PB) protocol [Busireddygari and Kak. 2017] is based on one-way communication channel while sending secret messages from sender party to recipient party and then from recipient party to sender party involving authentication between the communicating parties. Unlike the standard schemes of symmetric or asymmetric cryptography, the PB protocol allows the sender to send the secret message and its decryption key separately in two different one-way communication channels thereby increasing the complexity of decipherment for adversaries [John 1990], [Kak, 2014].

The PB protocol strengthens cryptographic algorithms like double-lock (DL) cryptography protocol and Secret Key Identification protocol (SKID) which are vulnerable to Man-in-the-Middle (MIM) attack. Although, the use of the PB authentication has been designed to mitigate MIM attack the question of how the communicating parties authenticate each other has not been considered. The basic PB scheme did not discuss mutual authentication between the communicating parties, which is a central concern for next generation networks [Mahdi, 2015].

This paper introduces the use of CA into the basic PB protocol. With this, the scope of the PB protocol is enhanced in terms of security for prevention of impersonation attacks.
Section 4.1 discusses the basic PB protocol. Section 4.2 proposes a strategy to tackle adversary attacks (MIM attacks) using a CA-based authentication of parties in the PB Protocol. Section 4.3 implements CA-based PB protocol in four stages. Section 4.4 describes the security analysis for an attacker to break the PB protocol transformations; it also discusses the computational easiness of the authentication parties. Finally, Section 4.5 provides conclusions.

6.1 The Basic PB Protocol

The PB scheme, which is inspired by Diffie-Hellman and RSA [Rivest et al. 1978], is a protocol to send an encrypted secret message in a locked and sealed piggy bank [Kak, 2014]. The impenetrable piggy bank allows the sender of the secret message to insert the secret by means of a one-way piggy bank (secret) transformation and the decryption key of an encrypted letter into it without unlocking it. Furthermore, the piggy bank allows only the recipient, who has locked it to unlock it.

In addition to the secret message and the decryption key of the encrypted letter, the cryptographic scheme mandates the sender to send a separate message along with the piggy bank to the recipient. Considering the two communicating parties as Alice and Bob, the implementation of the protocol is represented below in Fig. 36. The notation used is as follows:

- *R*, a random key, which is large. The difficulty of guessing this key provides an additional level of security.
- *n*, a composite number with prime factors *p* and *q*. where, $n = p \times q$.
- *e*, an encryption key exponent, which is publicly published in the domain. Recipient party (Alice) chooses *e* in such a way that it is relatively prime to $(p\text{-}1) \times (q\text{-}1)$.
- *S*, secret message, which is randomly generated by the sender party (Bob).
- *K*, another symmetric secret key, which is randomly generated by the same sender party.

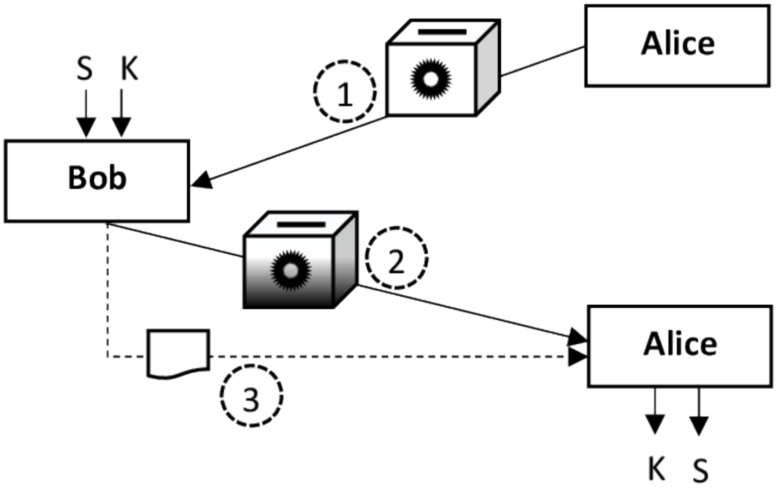


Fig. 36. The basic PB protocol.

The steps of the protocol are as follows:

1) Alice who wishes to communicate with Bob, sends him a transformed value, $f(R) = R^e \bmod n$.

2) Bob upon receiving the piggy bank or the transformed value from Alice, inserts his private message S, into it and then sends the ciphered information $S * R^e + K \bmod n$ to Alice in a one-way communication channel.

3) Bob also sends $f(S) = S^e \bmod n$ in another one-way communication channel to Alice to conveniently recover the secret messages S and K.

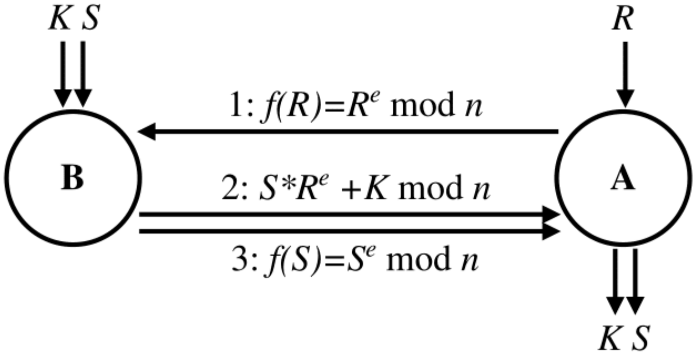


Fig. 37. Transformations in the basic PB protocol.

The basic PB protocol can potentially be implemented with multiple variations according to the cryptosystem specifications [Kak, 2014].

6.2 CA-Based Authentication of Parties in the PB Protocol

The Certification Authority (CA) is a trusted third party entity that issues certificates to the communicating parties by certifying their legitimate ownership of the public key [Burrows et al. 1990]. The CA's certificate notifies the following attributes about the communicating party: a version of the certificate, distinguished user name, encryption-decryption algorithm employed, the name of the certifying authority, validity period, user's public key, and finally signature of the CA. All these details pertaining to the communicating party uniquely distinguish his/her identity.

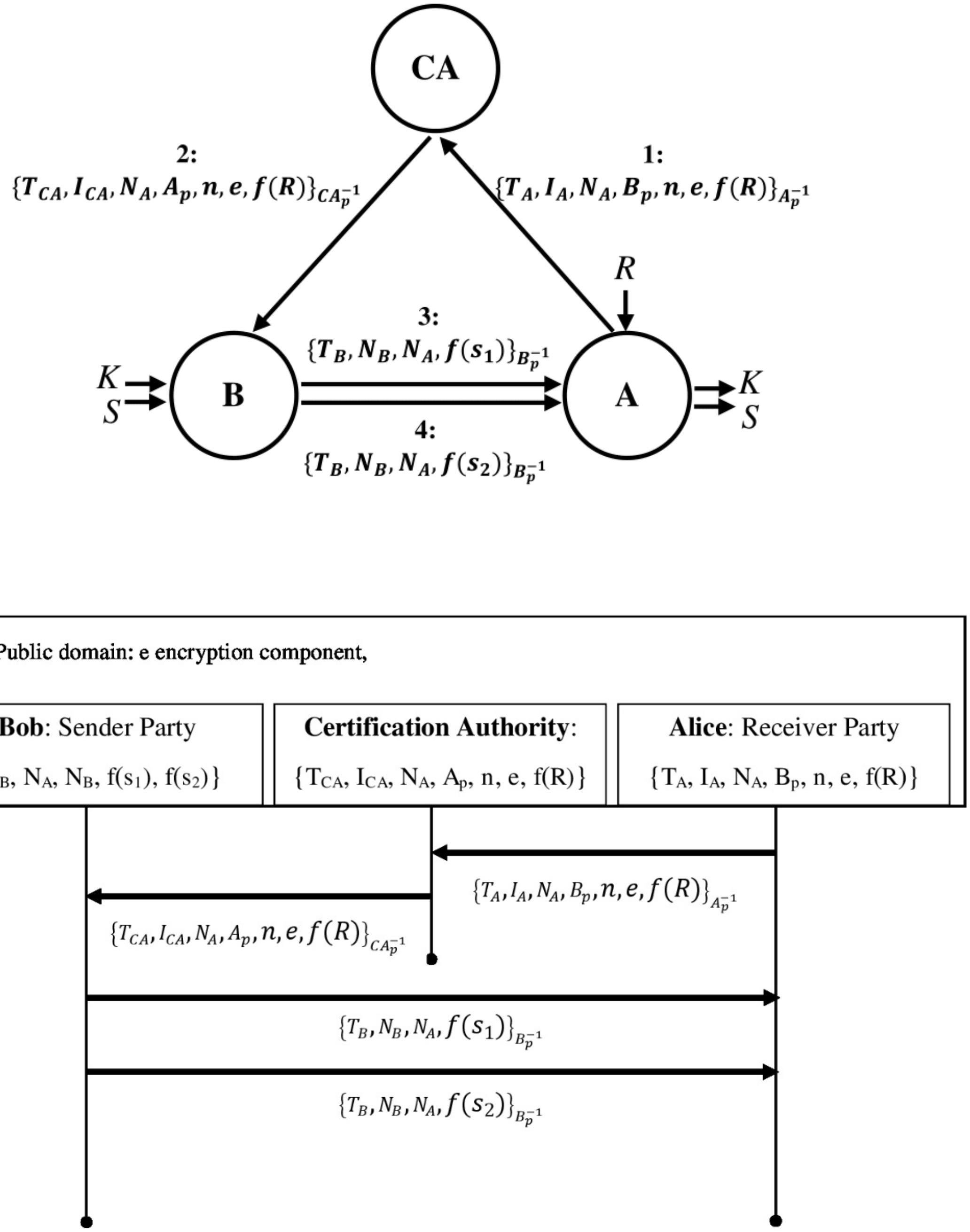

CA
2:
$\{T_{CA}, I_{CA}, N_A, A_p, n, e, f(R)\}_{CA_p^{-1}}$
1:
$\{T_A, I_A, N_A, B_p, n, e, f(R)\}_{A_p^{-1}}$
R
3:
$\{T_B, N_B, N_A, f(s_1)\}_{B_p^{-1}}$
K
S
B
A
K
S
4:
$\{T_B, N_B, N_A, f(s_2)\}_{B_p^{-1}}$
Public domain: e encryption component,
Bob: Sender Party
$\{T_B, N_A, N_B, f(s_1), f(s_2)\}$
Certification Authority:
$\{T_{CA}, I_{CA}, N_A, A_p, n, e, f(R)\}$
Alice: Receiver Party
$\{T_A, I_A, N_A, B_p, n, e, f(R)\}$
$\{T_A, I_A, N_A, B_p, n, e, f(R)\}_{A_p^{-1}}$
$\{T_{CA}, I_{CA}, N_A, A_p, n, e, f(R)\}_{CA_p^{-1}}$
$\{T_B, N_B, N_A, f(s_1)\}_{B_p^{-1}}$
$\{T_B, N_B, N_A, f(s_2)\}_{B_p^{-1}}$

Fig. 38. The PB protocol under the authority of CA.

The PB protocol under the authority of CA will potentially prevent the problem of MIM attack and thereby allow the communicating parties to securely exchange the secret information.

Alice in the CA-based PB protocol encrypts a message (empty piggy bank) using a random key, *R*, and sends the encrypted message to CA, who forwards Alice's encrypted message, *f(R)*, to Bob. Throughout the process, the random key generated by Alice is not shared with either CA or Bob. As a precautionary step, to prevent replay attacks, each message sent between the communicating parties is timestamped.

The authentication of communicating parties (here Alice and Bob) via a CA is as follows:

A. *Alice to CA:* $f(r) = \{T_A, I_A, N_A, B_p, n, e, f(R))\}A_p^{-1}$

- $f(R) = R^e \bmod n$
- T_A, Alice's communication timestamp
- I_A, Alice's identifier (if exists)
- N_A, Alice's secret nonce identifier

Alice initially informs CA that she wishes to communicate with Bob by sending Bob's public key. It is assumed that Alice knows Bob and holds his public key B_p, which is published in the public domain. Also, it is assumed both the communicating parties know the public key of the CA. CA learns Alice's public key, verifies whether her identity is legitimate in its records, and if not found it permanently registers Alice's public key and her identity. CA also registers the timestamp to ensure the message containing the public-key encryption is recent and no replay attacks occur. Finally, CA processes Alice's request to communicate with Bob.

B. *CA to Bob:* $f(c) = \{T_{CA}, I_{CA}, N_A, A_p, n, e, f(R)\}\ CA_p^{-1}$

CA confirms Bob's public key, B_p, which is published in the public domain and his identity. Later, it informs Bob that Alice wishes to communicate with him.

C. *Bob to Alice:* $f(s1) = \{T_B, N_A, N_B, f(s_1)\}\ B_p^{-1}$

where $f(s_1) = S * R^e + K \bmod n$

D. *Bob to Alice:* $f(s2) = \{T_B, N_A, N_B, f(s_2)\}\ B_p^{-1}$

where $f(s_2) = S^e \bmod n$

In these communications, *f(r)* denotes the receiver party, *f(c)* denotes the CA, *f(s1)* denotes the first secret message of the sender party, and *f(s2)* denotes the second secret message of the same sender party. Bob agrees to establish a communication channel with Alice and shares his secret message with her. Bob might or might not know who Alice is but his trust in CA encourages him to share his secret messages with Alice.

6.3 Implementation of the CA-Based Authentication using Input-Output Automata

In this section, we propose finite input-output automata for the CA-based PB protocol using nonce to ensure message integrity in the unsecure one-way communication transmission channel (*T*). This is done so that the automata can model nonce and furthermore can be used to test its resilience against adversary risk models [Doley and Yao, 1983], [Kurkowski and Penczek, 2009], also to account for the nature of asynchronous and distributed cryptosystem [Corin et al. 2004], [Gurgens et al. 2002]. This scheme makes it possible to involve the system components with each other which are operating at different time intervals using one-way communication channel.

The adversary in the unsecure one-way communication channel is assumed to be eavesdropper. It is assumed that the active adversary in the unsecure transmission channel attempts to learn the secret messages sent by Bob to Alice over the transmission channel. However, the adversary remains unsuccessful which is proved using finite input-output automata.

CA-based PB protocol is divided into four stages as shown in Fig. 39 to Fig. 42 that explain Nondeterministic Finite Automaton (NFA), Deterministic Finite Automaton (DFA) and state summaries at each stage. The final regular language expression for complete CA-based PB protocol model using nonce scheme is computed by taking the product of regular language expression at each stage.

A. *Formal definition of a FSM [*Isa et al. 2014*]*

In automata theory, a NFA is declared as a (Q, Σ, Δ, q_0, F).

- Q, all finite set of states in the Stage-I to Stage-IV
- Σ, all finite set of inputs used in designing the automaton
- Δ, transition function which is Q X Σ
- q_0, initial state in the Stage-I, II, III & IV, $q_0 \in Q$
- F, set of accepting states in the Stage-I to Stage-IV,

$F \subseteq Q$

Finite set of all input states a model accepts is the language of the model.

Language, L(Model) = (Q, Σ, Δ, q_0, F)

All finite set of inputs: R, S, K, e, n, A, B, T, +, ^, *, M

- $R \in \{0,1\}^*$: Alice's random key which is astronomically large

- S, K ∈ $\{0,1\}^*$: Bob's secret message which are astronomically large

- e ∈ $\{0,1\}^*$: Encryption exponent shared between Alice and Bob

- n ∈ $\{0,1\}^*$: Product of prime secrets shared between Alice and Bob

- A: Nonce of Alice

- B: Nonce of Bob

- T: Unsecure one-way transmission channel

B. *Stage-I: Alice to CA*

- $f(R) = R^e \bmod n$
- Alice computes and transmits her nonce to CA

In the regular language the expression for Stage-I is

L(Stage-I) = (*R*ˆ*eMnT*)*

{ x ∈ Σ*: Accept only input strings *R*ˆ*eMnT*}

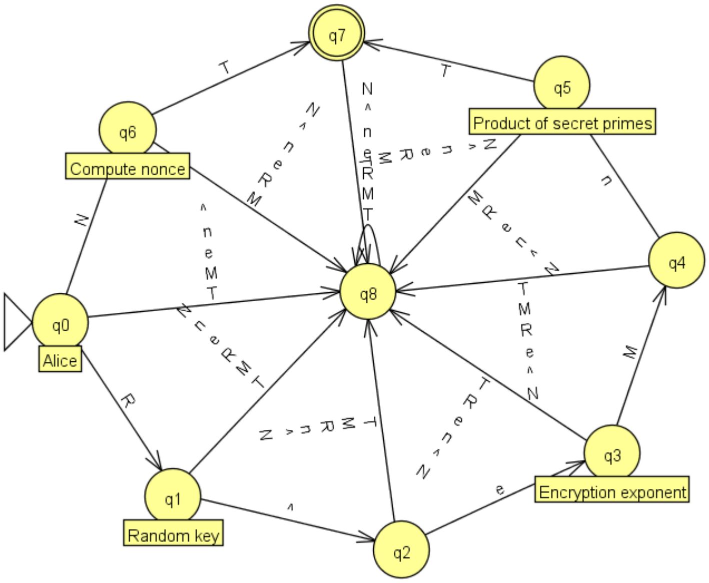


Fig. 39. Stage-I: Deterministic Finite Automaton.

C. *Stage-II: CA to Bob*

- CA verifies Alice’s and Bob’s identity and transmits Alice’s nonce to Bob

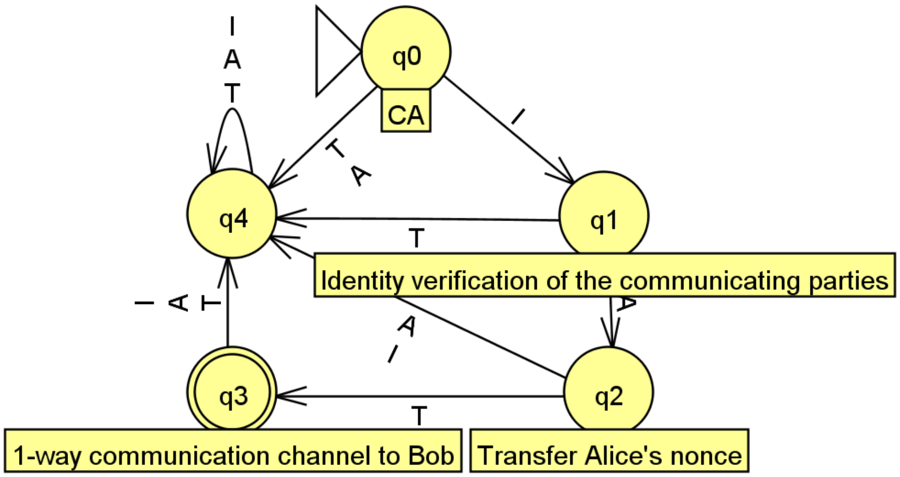

q0
CA
q1
q2
q3
q4
I
A
T
T
A
I
T
Identity verification of the communicating parties
A
A
I
I
A
T
T
1-way communication channel to Bob
Transfer Alice's nonce

Fig. 40. Stage-II: Deterministic Finite Automaton.

In the regular language the expression for Stage-II is

$L(\text{Stage-II}) = (IAT)^*\{\, x \in \Sigma^*\text{: Accept only input strings } IAT\}$

D. *Stage-III: Bob to Alice*

- $f(s_1) = S * R^e + K \bmod n$
- Bob computes and transmits his nonce to Alice

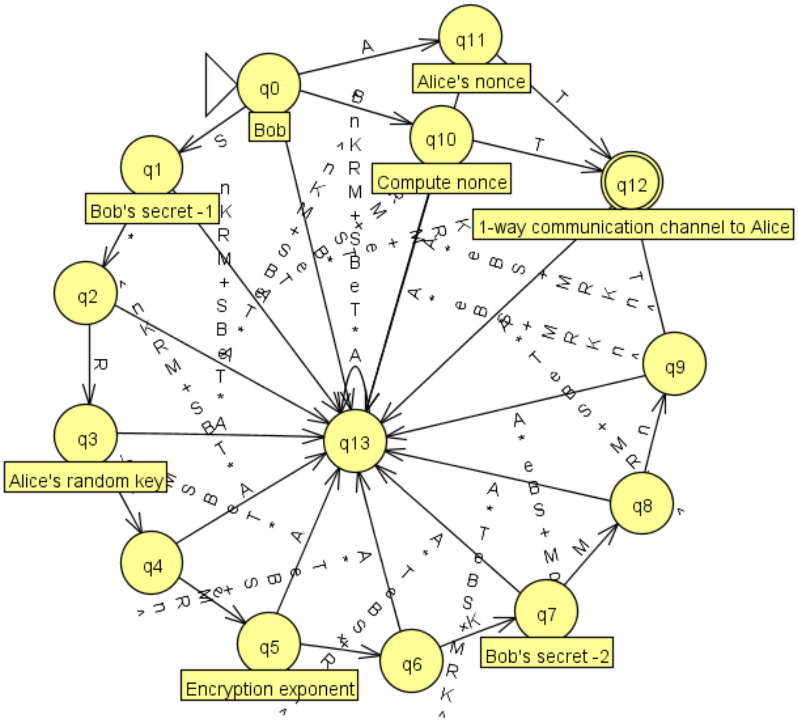


Fig. 41. Stage-III: Deterministic Finite Automaton.

In the regular language the expression for Stage-III is L(Stage-III) = $(S*R\hat{}e+KMnABT)^*$

{ $x \in \Sigma*$: Accept only inputstrings $S*R\hat{}e+KMnABT$}

E. *Stage-IV: Bob to Alice*

- $f(s_2) = S^e \bmod n$
- Bob computes and transmits his nonce to Alice

In the regular language the expression for Stage-IV is

L(Stage-IV) = { $x \in \Sigma*$: Accept only input strings $S\hat{}eMnABT$}

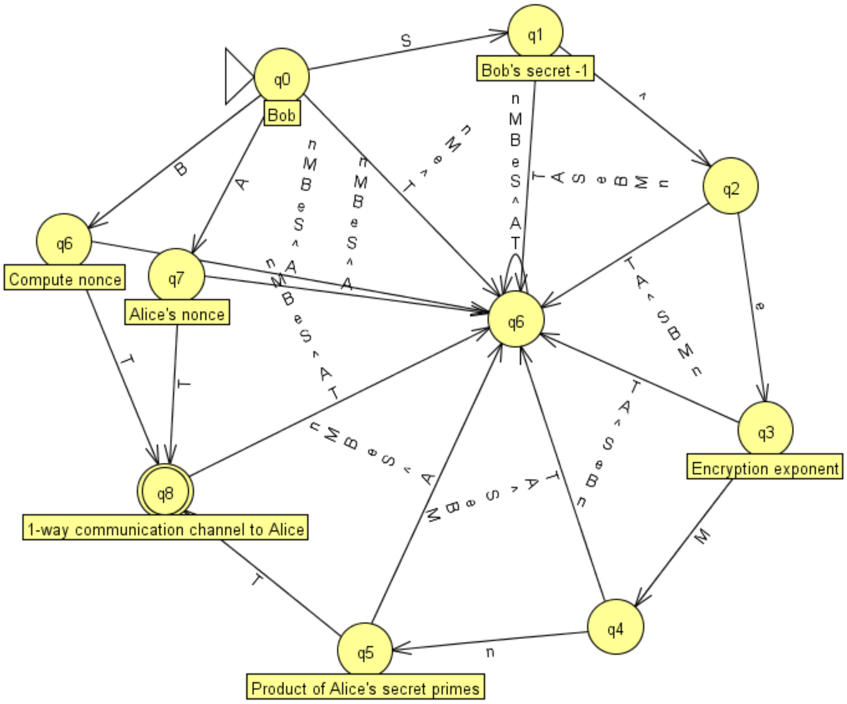


Fig. 42. Stage-IV: Deterministic Finite Automaton.

The final regular language expression for complete CA-based PB protocol model using nonce scheme is

$$L(\text{Model-CA}) = (Q,\Sigma,\Delta,q_0,F) = ((R^\wedge eMnT)^*(IAT)^*(S{*}R^\wedge e+KMnABT)^*(S^\wedge eMnABT)^*)^*$$

{ x ∈ Σ*: Accept only input strings with the pattern $R \wedge eMnT)$ * (IAT) *

$(S{*}R^\wedge e+KMnABT)$ *$(S^\wedge eMnABT)$}

The DFA checks the uniqueness of the regular expression of each stage. They provide the cryptosystem the inputs to compute the encryption formula and then authenticate the parties involved [Bao, 2004], [Gennaro et al. 2013], [Koltuksuz et al. 2010], [Liu et al. 2009], [Lynch, 1999], [Reitzig, 2008]. The regular language expression for the complete CA-based PB protocol is obtained by adding the regular language expression at each stage.

6.4 Security Analysis of the CA-Based PB Protocol

The PB protocol is resilient against the decryption of the secret message for attackers due to the following reasons:

6.4.1 Adaptability to new cryptographic techniques

The PB protocol offers multiple variations depending on the appropriate actions/operations chosen by the sender party [Kak, 2014]. These variations vary depending on the encryption and decryption transformations (exponentiation transformation, *f(R)* can be replaced with any advanced encryption, the additive transformation of *f(s1)* can be replaced by multiplicative transformation, and the secret message, *f(s2)* can be replaced with a nonce/hash). The variations can be configured into the protocol depending on the intended level of security. PB version of public key cryptography offers complete flexibility to implement new cryptographic functions while sending the encryption message.

6.4.2 Resilience to new cryptographic techniques

Cryptographic protocols that use a simple and straightforward encryption, where a communicating party who knows the random key and message can calculate an encrypted message, and another communicating party who knows the encrypted message and its decryption key can deduce the original encrypted message, are prone to user impersonation attacks [Capes-Davis and Neve, 2016]. In addition, in such cryptographic protocols, the communicating parties are not mutually authenticated.

CA in the CA-based PB protocol acts as a "gatekeeper" authorizing the identity of communicating parties and enforcing trust between them. Such an act by the CA not only prevents the root cause of MIM attack but also reduces the chance of other authentication attack techniques.

The idea of PB protocol is resilient against the decryption of the secret message for attackers due to the following reasons:

- The attackers will primarily be challenged to solve the factoring problem within a time limit.
- Authorized communicating parties utilize one-way encryption only once to access secret message.
- By making the whole protocol work under the supervision of the CA, impersonation attacks are less dominant.
- Each message sent between the communicating parties is timestamped to prevent replay attacks.

CA-based PB protocol involves less complexity for the communication parties to encrypt and decrypt the secret message. In addition, minimal communication energy and computational effort would be sufficient for the authorized communicating parties to send messages. Moreover, mutual authentication of communicating parties is guaranteed since the whole protocol works under the supervision of the CA.

6.4.3 Authentication Delay

The authentication delay is the total time taken from the instant the communicating party sends out the request to the time the authentication of communicating party is complete. The PB protocol under the authority of CA will have a variable authentication delay depending on the certificate authorities and their policies. One would consider minimal authentication delay to support real-time applications.

6.5 Conclusion

This chapter describes a version of the basic PB protocol under the supervision of the CA. The CA-based authentication of parties using the concept of piggy banking security simplifies the mutual verification process for the communicating parties involved offering security and resilience against various impersonation attacks, such as MIM attack, replay attacks and commonly possible message attacks like content modification. The modeling of CA-based PB security protocol on finite input-output automata illustrates its resilient security features against adversary threat models.

CHAPTER 7

CONCLUSION

In the vision for the 5G wireless communications advancement that yield new security prerequisites and challenges we propose a catalog of three new classes of pseudorandom random sequence generators. This dissertation starts with a review on the requirements of 5G wireless networking systems and the most recent development of the wireless security services applied to 5G, such as private-keys generation, key protection, and flexible authentication. This dissertation proposes new complexity theory-based, number-theoretic approaches to generate lightweight pseudorandom sequences, which protect the private information using spread spectrum techniques. For the class of new pseudorandom sequences, we obtain the generalization. Authentication issues of communicating parties in the basic model of Piggy Bank cryptography is considered and a flexible authentication using a certified authority is proposed.

REFERENCES

[Adem and et al, 2015] Adem, N., Hamdaoui, B., Yavuz, A.:" Pseudorandom time-hopping anti-jamming technique for mobile cognitive users" IEEE Globecom, 53, (2015), 1-6.

[Aerts, 2009] Aerts, D.:" Quantum structure in cognition." Journal of Mathematical Psychology, 53, (2009), 314-348.

[Aiello et al, 1998] Aiello, W., Rajagopalan, S., Venkatesan, R.:"Design of Practical and Provably Good Random Number Generators"; Journal of Algorithms, Vol. 29, (1998), 358-389.

[Aumasson and et al, 2010] Aumasson, J. P., Henzen, L., Meier, W., Plasencia, M.:" Quark: A lightweight hash". Cryptography Hardware Embedded Systems, vol. 6225, (2010), 1-15.

[Alur and Dill, 1994] Alur, R., Dill, D.: "A theory of timed automata"; Theoretical Computer Science (1994).

[Banik and Maitra, 2013] Banik, S., Maitra, S: "A differential fault attack on MICKEY 2.0" Cryptograph. Hardw. Embedded Syst., (2013), 215–232.

[Babbage and Dodd, 2008] Babbage, S., Dodd, M: "The MICKEY stream ciphers". New Stream Cipher Designs, (2008), 191–209.

[Bao, 2004] Bao, F.: "Cryptanalysis of a partially known cellular automata cryptosystem," IEEE Trans. Comp; 53, (2004), 1493–1497.

[Beaulieu and et al, 2015] Beaulieu, R., Shors, D., Smith, J., Clark, S., Weeks, B., Wingers, L.:"SIMON and SPECK: Block ciphers for the IoT". National Security Agency (NSA), (2010), 585.

[Berbain, 2008] Berbain, C..: "SOSEMANUK, a fast software-oriented stream cipher". New Stream Cipher Designs; vol.4986, (2008), 98–118.

[Bernstein, 2008] Bernstein, D. J..: "The Salsa20 family of stream ciphers". New Stream

Cipher Designs; vol.4986, (2008), 84–97.

[Biham and Dunkelman, 2000] Biham,E. Dunkelman, O.: "Cryptanalysis of the A5/1 GSM cipher". INDOCRYPT, (2000), 43 – 51.

[Blum and Micali, 1986] Blum, M. Micali, S.: "How to generate cryptographically strong sequences of pseudo random bits". IEEE symposium on Foundations of Computer Science, (1986), 112 – 117.

[Blum et al, 1996] Blum, M. Blum, L., Shub, M.: "A Simple Unpredictable Pseudo-Random Number Generator". SIAM Journal on Computing, (1996), 364 – 383.

[Blundo et al. 2004] Blundo, C., Cimato, S., De Prisco, R., and Ferrara, A. L.: "Modeling a certified email protocol using I/O automata"; Electronic Notes in Theoretical Computer Science; (2004).

[Boesgaard and et al, 2003], Boesgaard, M., Vesterager, M., Pedersen, T., Christiansen, J., Scavenius, O.:"Rabbit: A new high-performance stream cipher". Fast Software Encryption, (2003), 307-329.

[Bordel and et al, 2018], Bordel, B., Beatriz, A., Alcarria, R.: An Intra-Slice Security Solution for Emerging 5G Networks Based on Pseudo-Random Number Generators". IEEE Access, (2018), 16149-16165.

[Borisov et al. 2001] Borisov, N., Goldberg, I., and Wagner, D.: "Intercepting Mobile Communications: The Insecurity of 802.11"; ACM, (2001).

[Brown and Solomon, 1979] Brown, M., Solomon, H.:"On Combining Pseudorandom Number Generators"; Ann. Stat. 1, Vol. 7, N0. 3, (1979), 691-695.

[Bucerzanet al. 2010] Bucerzan, D., Craciun, M., andRatiu,C.: "Stream Ciphers Analysis Methods". Int. J. of Computers, Communications & Control, vol.5, (2010), 483-489.

[Burrows et al. 1990] Burrows, M., Abadi, M., and Needham, R.: "A Logic of Authentication"; ACM Transactions on Computer Systems; 8; (1990), 18-36.

[Busireddygari and Kak. 2017] BusiReddyGari, P., and Kak, S.: "Authentication of parties in piggy bank cryptography"; Asilomar Conference on Signals, Systems, and Computers; (2017), 1389-1393.

[Busireddygari and Kak. 2017] BusiReddyGari, P., and Kak, S.: "Binary primes sequence for cryptography and secure communication"; IEEE Conference on Communications and Network Security (CNS); (2017), 570-574.

[Busireddygari and Kak. 2017] BusiReddyGari, P., and Kak, S.: “Authentication of parties in piggy bank cryptography”; Asilomar Conference on Signals, Systems, and Computers; (2017), 1733-1736.

[Canniere and Preneel, 2008] Canniere, C., Preneel, B.:““”Trivium’’" in New Stream Cipher Designs”. (2008), 244-266.

[Capes-Davis and Neve, 2016] Capes-Davis, A., Neve, R. M.: “Authentication: A Standard Problem or a Problem of Standards?”; PLoS Biol 14(6): e1002477; (2016).

[Cassaigne et al. 1999] Cassaigne, J., Ferenczi, s., Mauduit, C., Rivat, J., Sarkozv, A.:”On finite pseudorandom binary sequences III: the Liouville function.” I. Acta Arith, 87, (1999), 367-390.

[Chaitin, 1966] Martin-Lof, P.: “On the length of programs for computing finite binary sequences”. Journal of ACM, (1966), 547 – 569.

[Chen, 1978] Chen, J. R.:” On the representation of a large even integer as the sum of a prime and the product of at most two primes.” II. Sci. Sinica, 16, (1978), 421-430.

[Choi, 2016] Choi, J.:” On the power allocation for a practical multiuser superposition scheme in NOMA systems” IEEE Communications, (2016), 483-491.

[Chugunkov et al, 2016] Chugunkov, I. V., Novikova, O. Y., Perevozchikov, V. A.:” The development and researching of lightweight pseudorandom number generators”; NW Russia Young Researchers in Electrical and Electronic Engineering Conference, IEEE (2016).

[Corin et al. 2004] Corin, R., Etalle, S., Hartel, P. H., Mader, A.: “Timed model checking of security protocols”; ACM Workshop on Formal methods in Security Engineering; (2004).

[Dabal and Pelka, 2012] Dabal, P., Pelka, R.: “FPGA Implementation of Chaotic Pseudo-Random Bit Generators”. Mixed Design of Integrated Circuits and Systems, (2012).

[Dabbagn and et al, 2015], Dabbagn, M., Hu, B., Guizani, M., Rayes, A.: “Software-Defined Networking Security: Pros and Cons”. IEEE Communications, vol. 53, (2015), 73-79.

[Dai and et al, 2015], Dai, L., Wang, B., Yuan, Y., Han, S.: “Nonorthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends”. IEEE Communications, vol. 53, (2015), 74-81.

[Dai and et al, 2018], Dai, L., Wang, B., Ding, Z., Wang, Z., Sheng, C., Hanzo, L.: “A Survey of Non-Orthogonal Multiple Access for 5G”. IEEE Communications, (2018), 1-30.

[Dillon and Dobbertin, 2004] Dillon, J., Dobbertin, H.: “New cyclic difference sets with Singer parameters”. Finite Fields and Their Applications, (2004), 342 – 389.

[Doley and Yao, 1983] Dolev, D., Yao, A.: “On the security of public key protocols”

IEEE Trans. Inf. Theory, vol. 29, no. 2, (1983), 198–208.

[Du et al. 2016] Du, Y., Dong, B., Chen, Z., Fang, J., Wang, X..:“ A fast convergence multiuser detection scheme for uplink SCMA systems” IEEE Communications, vol. 5, (2016), 388-391.

[Durgin et al. 2004] Durgin, N., Lincoln, P., Mitchell, J. C.:“Multiset rewriting and the complexity of bounded security protocols” Journal of Computer Security, vol. 12, (2004).

[Edward and Zoraida. 2018] Edward, Z., Zoraida, F.: “The cost, coverage and rollout implications of 5G infrastructure in Britain” Telecommunications Policy, vol. 42, (2018), 636-652.

[Fu and Tao, 2012] Fu, H., Tao, Y.: “A novel nonlinear precoding detection algorithm for VBLAST in MIMO-MC-CDMA downlink system” Physics Procedia, vol. 24, (2012), 1133-1139.

[Garay et al. 2000] Garay, J., Gennaro, R., Jutla, C., Rabin, T.: “Secure distributed storage and retrieval” Theoretical Computer Science, vol. 243, issue 1-2, (2000), 363-389.

[Gennaro et al. 2013] Gennaro, R., Hazay, C., Sorensen, J. S.: “Automata Evaluation and Text Search Protocols with Simulation Based Security” IACR Cryptology ePrint Archive. [Online]. Available: eprint.iacr.org/2010/484.pdf [Accessed: 15-Sep-2013].

[Goldreich and Rosen, 2005] Goldreich, O., Rosen, V.:”On the Security of Modular Exponentiation with Application to the Construction of Pseudorandom Generators”; Cryptology (2003), 71-93.

[Golomb and Gang, 2005] Golomb, S.W., Gong, G.:”Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar”; Cambridge University Press (2005).

[Golomb, 1982] Golomb, S.W.:” Shift Register Sequences”; Aegean Park Press (1982).

[Goresky and Klapper, 2009] Goresky, M., Klapper, A.:”Algebraic shift register sequences”; IEEE Transactions on Information Theory (2009).

[Gurgens et al. 2002] Gurgens, S., Ochsenschl¨ ager, P., Rudolph, C.: “Role based¨ specification and security analysis of cryptographic protocols using asynchronous product automata” 13th International Workshop on Database and Expert Systems Applications, (2002).

[Hardy and Wright, 1954] Hardy, G.H., Wright, E.M.:”An Introduction to the Theory of Numbers”; Oxford University Press (1954).

[Hell and et al, 2007] Hell, M., Johansson, T., Meier, W.: “Grain: A stream cipher for constrained environments, (2007), 86-93.

[Helleseth and Kumar, 1998] Helleseth, T., Kumar, P. V.: “Sequences with Low Correlation", Handbook of Coding Theory. Elsevier, (1998).

[Helleseth and Kumar, 1999] Helleseth, T., Kumar, P. V.: “Pseudonoise Sequences", The Mobile Communications Handbook, Springer, (1999).

[Helleseth and Gong, 2002] Helleseth, T., Gong, G.: “New nonbinary sequences with ideal two-level autocorrelation”. IEEE Transactions on Information Theory, (2002), 2868–2872.

[Hernandez and et al, 2002] Hernandez, J. L., Pawlowski, M. P., Jara A.J., Skarmeta A.F., Ladid L.: “Towards a Lightweight Authentication and Authorization Framework for Smart Objects”. IEEE J. Sel. Areas Comm. (2015), 690-702.

[Hoffstein et al, 2012] Hoffstein, J., Pipher, J., Silverman, J. H.:” An Introduction to Mathematical Cryptography”; Springer (2012).

[Isa et al. 2014] Isa, M. A. M., Ahmad, M. M., Sani, N. F. M., Hashim, H., Mahmod, R.: “Cryptographic Key Exchange Protocol with Message Authentication Codes (MAC) Using Finite State Machine” Procedia Computer Science, vol. 42, pp.263-270, (2014).

[John 1990] John, R.: “One-Way Functions Are Necessary and Sufficient for Digital Signatures” ACM Symposium on the Theory of Computing, (1990), 387-394.

[Kak, 1970] Kak, S.:” Classification of random binary sequences using Walsh-Fourier analysis”; IEEE Transactions on Electromagnetic Compatibility Magazine, vol.13, pp. 74-77 (1970).

[Kak, 1981] Kak, S.:” Chatterjee, A.: On decimal sequences”; IEEE Transactions On Information Theory, vol. IT-27 (1981).

[Kak, 1985] Kak, S.:” Encryption and Error-Correction Coding Using D Sequences”; IEEE Transactions On Computers, vol. C-34, N0. 9 (1985).

[Kak, 1996] Kak, S.:” The three languages of the brain: quantum, reorganizational, and associative in Learning as Self-Organization.” K. Pribram and J. King (editors) Lawrence Erlbaum Associates, Mahwah, NJ, (1996), 185-219.

[Kak, 1999] Kak, S.:” The initialization problem in quantum computing.” Foundations of Physics, 29, (1999), 267-279.

[Kak, 2007] Kak, S.:” Quantum information and entropy.” Int. Journal of Theo. Phys., 46, (2007), 860- 876.

[Kak, 2012] Kak, S.:” Random Sequences Using the Divisor Pairs Function”; arXiv:1210.4614 (2012).

[Kak, 2014] Kak, S.:” Goldbach partitions and sequences.” Resonance, 19, (2014), 1028-1037.

[Kak, 2014] Kak, S.: “The piggy bank cryptographic trope” Information communications Journal 6, (2014), 22-25.

[Kak, 2014] Kak, S.: “Authentication Using Piggy Bank Approach to Secure Double-Lock Cryptography” arXiv:1411.3645, (2014).

[Katagi and Moriai, 2008] Katagi, M., Moriai, S.: "Lightweight cryptography for the Internet of Things". Sony Corp, (2008), 7-10.

[Kolmogorov, 1965] Kolmogorov, A.:" Three approaches to the quantitative definition of information"; Problems of Information Transmission, pp.1-17 (1965).

[Koltuksuz et al. 2010] Koltuksuz, A., Kulahcioglu, B., Ozkan, M.: "Utilization of timed automata as a verification tool for security protocols" Fourth IEEE International Conference on Secure Software Integration and Reliability Improvement Companion, (2010).

[Knuth, 1997] Knuth, D.:" The Art of Computer Programming." Volume 2: Seminumerical Algorithms. Addison-Wesley (1997)

[Kurokawa et al, 2016] Kurokawa, T., Nojima, R., and Moriai, S.: "On the security of CBC Mode in SSL3.0 and TLS1.0" Journal of Internet Services and Information Security (JISIS), vol 6, (2016), pp. 2-19

[Kurkowski and Penczek, 2009] Kurkowski, M., Penczek, W.: "Timed automata based model checking of timed security protocols" Fundamenta Informaticae, (2009).

[Landauer, 1996] Landauer, R.:" The physical nature of information." Physics Letters A, 217, (1996), 188-193.

[Larsson et al, 2014] Larsson, E. G. Edfors, O., Tufvesson, T. L. Marzetta, L.: "Massive MIMO for next generation wireless systems". IEEE Communications, (2014), 186 – 195.

[L'Ecuyer, 2012] L'Ecuyer, P.:" Random Number Generation"; In: Gentle, J. E., Karl, H. W., Mori, Y. (eds.) Springer, Heidelberg (2012), 35-71.

[Leon et al, 2004] Leon, D. Balkir, S., Hoffman, M. W. Perez, L. C.: “Pseudo-chaotic PN-sequence generator circuits for spread spectrum communications”. IEEE symposium on Foundations of Computer Science, (2004), 543 – 550.

[Liu et al. 2009] Liu, N., Zhu, W., Zhu, Y.: “Security protocol analysis based on rewriting approximation”. Second International Symposium on Electronic Commerce and Security, (2009).

[Luby, 1996] Luby, M.:” Pseudorandomness and Cryptographic Applications”. Princeton University Press (1996)

[Lynch, 1999] Lynch, N.: “I/O automaton models and proofs for shared-key communication systems” 12th IEEE Computer Security Foundations Workshop, (1999).

[Mabin and et al, 2017] Mabin, J., Sekar, G., Balasubramanian, R.: “Distinguishing Attacks on(Ultra-)Lightweight WG Ciphers” 5th International Workshop Lightweight Cryptography for Security and Privacy, (2017), 45-59.

[Mahdi, 2015] Mahdi, A.: “A formal analysis of authentication protocols for mobile devices in next generation networks” Concurrency and Computation: Practice and Experience, (2015), 2938-2953.

[Marin and et al, 2015] Marin, L., Pawlowski, M. P., Jara, A.: “Optimized ECC Implementation for Secure Communication between Heterogeneous IoT Devices”. Sensors, (2015), 21478-99.

[Martin-Lof, 1966] Martin-Lof, P.: “On the definition of random sequences”. Information and Control, (1966), 602 – 619.

[Mauduit, 2002] Mauduit, C.:”Finite and Infinite Pseudorandom Binary Words”; Theoretical Computer Science, (2002), 249-261.

[Meadows, 2003] Meadows, C.: “Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends” IEEE Journal on selected areas in communications, 21, (2003).

[Micali and Schnorr, 1991] Micali, S., Schnorr, C..:”Efficient, Perfect Polynomial Random Number Generators”. Cryptology, (1991), 157-172.

[Miguel and Juan, 2016] Miguel, H. C., Juan, C. G. E.:”Quantum Random Number Generators”; arXiv:1604.0330 (2016).

[Mkubulo, 2015] Mkubulo, D.:” Analysis of Wi-Fi Security Protocols and Authentication Delay”; FSU digital library, (2015), 1-66.

[Mouha, 2015] Mouha, N.:” The Design Space of Lightweight Cryptography”. NIST Lightweight Creyptography Workshop, (2015).

[Mukherjee et al. 2002] Mukherjee, M., Ganguly, N., Chaudhuri, P.: “Cellular automata based authentication (CAA)” Cell. Autom. Lect. Notes Comput. Sci., 2493, pp. 259–269, (2002).

[Nicolas and Robin, 1997] Nicolas, J. L., Robin, G.:”Highly composite numbers of S. Ramanujan”. The Ramanujan Journal, 1, (1997), 119-153.

[Oliveria, 2010] Oliveira, G., Martins, L., Ferreira, G. B., Alt, L. S.: “Secret key specification for a variable-length cryptographic cellular automata model”. Parallel Probl. Solving from Nature, PPSN XI, Lect. Notes Comput. Sci., 6239, (2010), 381-390.

[Orhanou and Hajji, 2016] Orhanou, G. and Hajji, S.:”The new LTE cryptographic algorithms EEA3 and EIA3,"” Applied Mathematics, vol. 7, (2016), 2385-2390.

[Orue and et al, 2017] Orue, A. B., Encinas, L., Martin, A., Montoya, F.:” A lightweight Pseudorandom NumberGenerator for securing the Internet of Things”. IEEE Access, (2017), 1-6.

[Panneton et al. 2006] Panneton, F., L’Ecuyer, P., Matsumoto, M.: “Improved Long-Period Generators Based on Linear Recurrences Modulo 2”; Transactions on Mathematical Software ACM, (2006), 1-16.

[Panwar et al. 2016] Panwar, N., Sharma, S., Singh, A. K.: “A survey on 5G: The next generation of mobile communication”. Physics Communications, vol. 18, (2016), 64-84.

[Pawlowski et al. 2014] Pawlowski, M. P., Jara, A. J., Ogorzalek, M.J.: “Extending Extensible Authentication Protocol over IEEE 802.15.4 networks”. Innovative Mobile and Internet Services in Ubiquitous Computing, (2014), 340-345.

[Peinado et al. 2014] Peinado, A., Munilla, J., Sabater, A.: “EPCGen2 Pseudorandom Number Generators: Analysis of J3Gen”. Sensors, (2016), 6500-6515.

[Plumstead, 1982] Plumstead, J.: “Inferring a sequence generated by a linear congruence”. Proc. 8th Internat. Coll. on Automata, Languages, and Programming, (1981).

[Prasithsangaree and Krishnamurthy, 2003] Prasithsangaree, P. and, Krishnamurthy, P.:” Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs”. Globecom, (2003), 1445-1449.

[Qiao and et al, 2015] Qiao, J., Shen, X., Mark, J., Shen, Q., He, Y., Lei, L.: “Enabling device-to device communications in millimeter-wave 5G cellular networks”, IEEE Communications, vol. 53, (2015), 209-215.

[Rahman and et al, 2017] Rahman, M. S., Basu, A., Kiyomoto, S.:" Decentralized cipher text policy attribute-based encryption: A post-quantum construction", vol. 7, 2017, 1-16.

[Reddy and Kak, 2016] Reddy, B. P., Kak, S.:"The binary primes sequence for computational hardening of pseudorandom sequences." arXiv:1606.00410 (2016)

[Reitzig, 2008] Reitzig, R.: "Modelling and Proving System Security Using Finite Automata and Noninterference" TU Kaiserslautern, (2008).

[Reynena et al, 2010] Reynena, B., Osman, H., Malika, P.: "Using gold sequences to improve the performance of correlation based islanding detection" Elsevier, (2010), 733-738.

[Rey, 2007] Rey, A.:"Message authentication protocol based on cellular automata" Applications of Evolutionary Computing, (2007).

[Rivest et al. 1978] Rivest, R., Shamir, A., Adleman, L.: "A method for obtaining digital signatures and public key cryptosystems"; Comm. ACM, vol.21, (1978), 120-126.

[Rocha et al. 2011] Rocha, F., Abreau, S., and Correia, M.: "The final frontier: confidentiality and privacy in the cloud"; IEEE Computer. Vol. 44, (2011).

[Rusek et al. 2013] Rusek, F., Persson, D., Kiong, B. L., Larsson, E., Marzetta, T. L., Edfors, O. L., Tufvesson, F.: "Scaling up MIMO: Opportunities and challenges with very large arrays". IEEE Signal Process. Mag., (2013), 40-60.

[Sachs et al, 2007] Sachs, J., Herrmann, R., Kmec, M., Helbig, M., Schilling, K.: "Recent Advances and Applications of M-Sequence based Ultra-Wideband Sensors" IEEE International Conference on Ultra-Wideband, (2007).

[Samardzija et al, 2001] Samardzija, D., Wolniansky, P., Ling, J.: "Performance evaluation of the VBLAST algorithm in W-CDMA systems" IEEE Conference on Vehicular Technology, (2001), 723-727.

[Schneier, 1996] Schneier, B.: "Applied Cryptography". J. Wiley & Sons Inc, (second edition), (1996).

[Shamir, 1981] Shamir, A.: "On the generation of cryptographically strong pseudorandom sequences". Proc. 23rd IEEE Symp. on Foundations of Computer Science, (1982), 153-159.

[Shirai and et al, 2017] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: "The 128-bit blockcipher CLEFIA". Fast Software Encryption, vol. 4593, (2017), 181-195.

[Simon et al, 1994] Simon, M. K., Omura, J. K., Scholtz, R. A., Levitt, B. K.:" Spread Spectrum Communications Handbook"; NY: McGraw-Hill (1994).

[Tian and et al, 2017] Tian, F., Zhang, P., Yan, Z.: "A survey on C-RAN security", IEEE Access, vol. 5, (2017), 13372-13386.

[Vij and Jain, 2015] Vij, S., Jain, A.:" 5G: Evolution of a secure mobile technology"; International Conference on Computing, Sustainability and Global Development, (2015), 2192 – 2196.

[Viterbi, 1995] Viterbi, A. J.:" principles of spread spectrum communication"; Addison Wesley Longman Publishing Co., Inc (1995).

[Watts, 2003] Watts, A.:"A dynamic model of network formation. Games and Economic Behavior"; Networks and Groups, (2003), 337-345.

[Wang and et al, 2016] Wang, Y., Miao, J, Jiao, L.:" Safeguarding the ultra-dense networks with the aid of physical layer security: A review and a case study"; IEEE Access, (2016), 9082-9092.

[Wang and et al, 2015] Wang, M., Yan, Z., Niemi, V.:" UAKA-D2D: Universal authentication and key agreement protocol in D2D communications"; Mobile Networks Applications, (2015), 510-525.

[Wang, 2015] Wang, Y.:" On statistical distance based testing of pseudo random sequences and experiments with PHP and Debian OpenSSL"; Computers & Security, (2015), 44-64.

[Wolfram, 1986] Wolfram, S.: "Cryptography with Cellular Automata" Advances in Cryptology: Crypto'85 Proceedings. Lect. Notes Comput. Sci., (1986).

[Wu, 2004] Wu, H.: "The Stream Cipher HC - 128". Lect. Notes Comput. Sci., (1986), 39-47.

[Youssef, 2009] Youssef, M. Zahara, M., Emam, A., Elghany, M.: "Image encryption using pseudo random number and chaotic sequence generators". Radio Science Conference, (2009).

[Zhang and et al, 2017] Zhang, A., Wang, L., Ye, X., and Lin, X.:" Light-weight and robust security aware D2D-assist data transmission protocol for mobile-health systems" Information Forensic security, vol.12, (2017), 662-675.

[Zhao and Li, 2005] Zhao, J., Li, X.:" A novel iterative equalization algorithm for multi-code CDMA system with V-BLAST architecture" Proc. 2005 IEEE Conf. on Consumer Communications and Networking (CCNC), (2005), 211-214.

[Zhou and et al, 2016] Zhou, A., Luo, H., Li, R., Wang, J.:" A dynamic states reduction message passing algorithm for sparse code multiple access" IEEE Wireless Telecommunications Symposium, 4, 12, (2016), 1-5.

[Ziani and Medouri, 2015] Ziani, A., Medouri, A.:” Analysis of Different Pseudo-Random and Orthogonal Spreading Sequences in DS-CDMA.” IJEIT, 4, 12, (2015), 195-207.

VITA

PRASHANTH BUSIREDDYGARI

Candidate for the Degree of Doctor of Philosophy

Dissertation: NEW CLASSES OF BINARY RANDOM SEQUENCES FOR CRYPTOGRAPHY

Major Field: Cryptography & Cybersecurity

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Cryptography and Cybersecurity at Oklahoma State University, Stillwater, Oklahoma in May 2019.

Completed the requirements for the Master of Science in Cryptography and Cybersecurity at Oklahoma State University, Stillwater, Oklahoma in May 2015.

Completed the requirements for the Bachelor of Engineering in Electronics & Communications Engineering at Visvesvaraya Technological University, Belgaum, India in 2012.

Publications:

Busireddygari, Prashanth, Kak Subhash. "Pseudorandom Tableau Sequences." Asilomar Conference on Signals, Systems, and Computers, (Fall 2017): 1733-1736.

Busireddygari, Prashanth, Kak Subhash. "Authentication of parties in piggy bank cryptography." Asilomar Conference on Signals, Systems, and Computers, (Fall 2017): 1389-1393.

Busireddygari, Prashanth, Kak Subhash. "Binary Primes Sequence for Cryptography and Secure Communication." IEEE Conference on Communications and Network Security: The Workshop on Physical-Layer Methods for Wireless Security, (Fall 2017).

Busireddygari, Prashanth. "Coding Side-Information for Implementing Cubic Transformation." Cornell University archive, (Spring 2015): 1506.04650.

Busireddygari, Prashanth. "New Class of Pseudorandom D-sequences to Generate Cryptographic Keys." Cornell University archive, (Spring 2015): 1507.00712.