Design, Analysis and Evaluation of a Lightweight Symmetric Key Stream Generator for Smart Meter and Autonomous Vehicle Security

NAGARAJ HEDIYAL (nagaraj_hediyal@yahoo.com)
REVA University
B P Divakar
REVA University

Research Article

Keywords: Encryption, Smart Grid, Advance Metering Infrastructure, Autonomous Vehicle, Symmetric, Security, Statistical, Tests, Attacks

Posted Date: May 5th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1618395/v1

Abstract

The security of the smart grid and autonomous vehicles is the cause for concern as they function with the aid of heterogeneous networks formed by information & communication technologies. These technologies employ different standards and protocols to operate, resulting in security issues. The issue of security can be mitigated by applying cryptographic solutions. The basic requirements of security solutions must be lightweight in terms of hardware complexity, cost, least computation time, memory, and other resources. This paper proposes a symmetric key stream generator that satisfies the requirements of being a lightweight solution. An experimental setup developed around a popular Arduino nano-platform was used to implement the proposed symmetric key generator. Also, a statistical test suite was developed to conduct statistical tests on the output bit stream of the proposed symmetric key generator. The proposed symmetric key generator and its output bit stream were subjected to mathematical, experimental, statistical, security, and comparative analysis to validate its suitability for smart grid and autonomous vehicle security. The results have proved that the proposed symmetric-key generator is indeed suitable for smart grid and autonomous vehicle security.

1. Introduction

The penetration of information and communication technologies (ICTs) into the smart grid has helped the grid to enhance its performance, reliability, efficiency, and safety. However, ICTs generate vast amounts of data that need to be secured; otherwise, it can be tampered with, thus resulting in an unstable grid and the economy [1]. On the other hand, autonomous vehicle (AV) and connected vehicle technologies have improved the mobility of various sections of humans, like differently abled, children, and elderly populations. These technologies also have improved the fuel efficiency of vehicles, and reduced accidents too, all due to integrated ICTs. To ensure safe drive, these vehicles use several sensors and communication devices. The amount of data generated in real time through these devices is very huge [2]. Despite, potential benefits of ICTs, there are significant challenges in smart grid (SG) and autonomous vehicle (AV) network as the data exchanged in real time is very huge and most sensitive too. This data is vulnerable to cyber-attacks leading to compromise on security. To ensure data security, cybersecurity policies have been employed across these communication networks. There are many cyber security solutions and most of them call for multiple algorithms to establish secure communication channels before exchanging the data safely and securely. The time required to setup a secured channel, and to exchange data is very important and critical in ensuring security. This time must be very short in order to secure both the smart grid and the autonomous vehicles. Unfortunately, most of this time is lost in adhering to the key-management policies; which involve key generation, exchange, storage, erase, and replacement of keys of the cryptographic algorithm while setting up the secured channel leaving little time for data exchange and operations. This paper addresses this issue and simplifies the key-management procedure without compromising on its security properties with respect to the communication channels.

The main objectives of this work are:

i. To develop randomly and uniquely computed digital signature-based authentication and symmetric key stream generator for encryption/decryption.
ii. To provide critical analysis of selected schemes, classical algorithms and proposed symmetric key generator.
iii. To develop an experimental test bench and statistical test suite to evaluate the proposed symmetric key stream generator.

Rest of this article is organized as follows: In section II, the related work on Smart Grid and AV/CV is presented. In section III, symmetric key generator for smart grid and autonomous vehicles is presented. Section IV describes the proposed key stream generator design followed by a discussion on secured communication process using the cryptographic algorithm in Section V. Results and analysis are presented in section VI while discussion and evaluation are presented in section VII followed by conclusion in section VIII.

2. Related Work

The information & communication technologies employed in smart grid and autonomous vehicle control and operations generate huge amount of data resulting in cyber-attacks to destabilize their operations [1]-[2]. Yuancheng Li et al., [1] has presented Light weight quantum encryption to secure the transmission of power data in smart grid. The presented encryption involves key management where prior distribution of the keys is a must, along with a “one-time pad” (OTP) for authentication and confidentiality. The key distribution is very complicated and moreover, quantum cryptography is still evolving [1]. Otisitswe Kebotogetse et al.,[3] has presented the review of various key management techniques proposed by previous researchers to secure the advance metering infrastructure (AMI) in smart grid. The key management techniques reviewed in [3] are multithreaded and complicated. The smart grid is heterogeneous in nature and has many interconnected networks such as Home Area Network (HAN), Neighbourhood Area Network (NAN), and Wide Area Network (WAN). Cheng et al, [4] have presented lattice-based public key encryption for mutual authentication and key exchange between the smart meter and the NAN to secure the meter data. The lattice is a regularly spaced grid of points stretching out to infinity. The authentication scheme presented in [4] ensures confidentiality, but the time required seems to be large. Dominik Engel et al., [5] have presented a Wavelet-Based Multiresolution Smart Meter Privacy to protect power data. The proposed scheme requires hierarchical keying schemes to protect meter data. Hesham Aly El Zouka, et al., [6] utilized a Time Granularity-based Privacy Protection for Cloud Metering systems”, i.e., Asymmetric key encryption scheme. Amritha et al., [7] have utilized Nth degree Truncated Polynomial Ring Units (NTRU) based light-weight public key algorithms to provide end-to-end security of Supervisory Control and Data Acquisition (SCADA) systems in smart grids. The proposed algorithm performed better than Rivest, Shamir, Adleman (RSA) public key cryptography and elliptic curve cryptography as they consume huge amounts of time in the key generation and authentication process. Jia-Lun Tsai et al., [8] have presented Anonymous Key Distribution Scheme for Smart Grid security. In this presentation, identity-based signature and encryption schemes are employed to secure the smart grid. In

the proposed method, the smart meter is authenticated by a service provider without the third-party involvement as in public key cryptography. Ref [9], [10] and [11] have dealt with the security and authentication process through dynamic, fast encryption schemes for smart grid applications. On the other hand, autonomous/connected vehicles operation is also supported by ICTs. The intelligent traffic systems (ITS) interact with AV/CV over ICTs leading to security issues. The data communication could be between vehicle-to-vehicle (V2V), vehicle-to infrastructure (V2I) and other road identities. To protect the vehicular data, encryption is a must. Abdullahi Chowdhury et al, [12] have presented a detailed survey on the security issues related to connected vehicles and autonomous vehicle technologies. In [13] Sahand Murad et al., has dealt the security of the vehicular data shared over cloud by means of fragmentation and encrypting. The researchers have indicated that encryption with fragmented data saves resources and time. Joonsang Yoo et al., [14] have proposed code-based light weight cryptography to generate the authentication code to start the vehicle by the authorized driver only. Since the proposed authentication encryption involves only a multiplication process, the time required to authenticate the user is very short indicating that computation time is short. Mahdi Dibaei et al., [15] have detailed the various types of attacks on AV/CV and defences. Jamal Raiyn [16] has proposed biometric data for message authentication and communication in AV networks. The paper deals with security of the data acquired from various sensors on board and off board of the AV while in move. The work uses an iris-based driver authentication schedule. In [17] a survey of AV evolution, current devices in use have been detailed. The work indicates future directions, how exactly AV technology is evolving, issues concerned about safety and what is to be expected.Barry Sheehan et al., [18] have presented the classification of cyber risk involved in both AV/CV technologies. The work provides an insight into the security aspects that need to be taken care of in both AV and CV transportation systems. In [20], the AV application for farming is described along with their security concerns with respect to the user. In [21], the detection of cyber-attacks on AV/CV is described. The detection method uses machine learning techniques to identify and classify the type of attack on AV/CV. Machine learning techniques employ Decision Tree and Naive Bayes algorithms in detecting, identifying and classifying the type of attack on AV/CV. In [22] and [23] a review of security of AV and CV has been dealt.

The reviewed various authentication and cyber security schemes for SG, AV/CV above, involves at least two algorithms with time constraint. The authentication process uses one algorithm and the other for encryption /decryption. Most of the algorithms are expensive in terms of resources, time, and cost. Hence, there is a need for a cryptographic algorithm suitable for SG, AV/CV applications with least computation time, low cost, limited resources to meet the security aspects.

3. Symmetric Key Generator For Sm/av

Most of the security solutions proposed in the past for SG and AV applications involve at least two algorithms, one for authentication and the other for encryption/decryption process. Usually, Diffie-Hellman public key infrastructure (PKI) for authentication and symmetric key cipher using multilevel process to secure the data. It has also been noted, that traditional data encryption standard (DES),

advanced encryption standard (AES), Triple DES schemes were proposed to be used in these applications [10]. However, these are not economical and efficient due to high speed and light weight requirements.

A. RESEARCH PROBLEM

The ICT's have a number of wired and wireless communication technologies and are vulnerable to security threats, especially the wireless ones. The security requirements of these applications are:

i. Authentication of communication entities to ensure that the data exchange is in fact between legitimate entities.
ii. The data must be secured with a cryptographic scheme/algorithm.
iii. The time, cost, and resources, must be lightweight.

In the proposed work, the communication scenario between the smart meter and the data concentrator is considered as one example and the other is between the remote and the autonomous vehicle. Both scenarios use WiFi networks to communicate. The communication scenario between smart meter and data concentrator is depicted in Figure 1.

Similarly, the communication scenario between the WiFi network and the autonomous vehicle is shown in Figure 2.

In the practical scenario, the smart meter data is shared every 15 minutes once, based on the requests by the respective data concentrators. Whereas, in case of AV/CV the data is requested by the vehicle from wifi network frequently to reach the destination. If an adversary is monitoring the communication in both the scenarios, data may be compromised. As a result, smart grid may become unstable and insecure. In case of AV/CV, the vehicles may be driven anywhere other than their respective destination posing a threat to the life of the user. Hence, cryptographic solutions are required.

B. CHALLENGES

The existing security solutions do employ cryptography to secure SG, autonomous vehicles but almost all of them possess complicated authentication (Public Key Infrastructure) scheme and symmetric keys for data security with prior session key distribution. The existing symmetric key schemes in use require storage space to store the session keys in a system.

The challenges in both applications are:

i. Encryption scheme must be very efficient and reliable with low overhead.
ii. Security algorithm must be light weight in storage and computational resources.
iii. The cyber security solution must be resistant to most of the threats.
iv. The solution must have a very low or no key storage and prior distribution of keys.
v. The time, and cost, must be low.

Taking into account the detailed research problem and the challenges above, a symmetric key stream generator has been proposed.

4. Proposed Key Stream Generator-design

Cyber security in both SG and AV/CV applications demands an algorithm that must not only provide data security but also ensure the authentication of communication entities within the shortest possible time. The schemes devised must be, simple, strong, and resource friendly. The proposed work consists of two parts namely, a key stream generator and key generation.

A. STRUCTURE OF KEY STREAM GENERATOR

Key stream generator designs are mostly based on shift registers as they possess good statistical properties. Moreover, they are simple to implement and faster in speed. The proposed key stream generator consists of three linear feedback shift registers (LFSR) with each having four primitive polynomials. The three shift registers of degree 19, 23 and 29 are selected for the algorithm. The polynomials for the selected three LFSRs are listed in Table -1.

Table-1: List of LFSR and Corresponding Polynomials

Sl No	Degree of LFSR	Polynomial
1	19-bit	$x^{19} + x^5 + x^2 + x + 1$
		$x^{19} + x^{17} + x^{15} + x^{14} + x^{13} + x^{12} + x^6 + x + 1$
		$x^{19} + x^{17} + x^{15} + x^{14} + x^{13} + x^9 + x^8 + x^4 + x^2 + x + 1$
		$x^{19} + x^{16} + x^{13} + x^{11} + x^{10} + x^9 + x^4 + x + 1$
2	23-bit	$x^{23} + x^5 + 1$
		$x^{23} + x^{17} + x^{11} + x^9 + x^8 + x^5 + x^4 + x + 1$
		$x^{23} + x^5 + x^4 + x + 1$
		$x^{23} + x^{12} + x^5 + x^4 + 1$
3	29-bit	$x^{29} + x^2 + 1$
		$x^{29} + x^{20} + x^{11} + x^2 + 1$
		$x^{29} + x^{12} + x^7 + x^2 + 1$
		$x^{29} + x^{21} + x^5 + x^2 + 1$

The structure of the proposed keystream generator with single and multiple polynomials are illustrated in Figure 3 and Figure 4 respectively.

To increase confusion and diffusion, the key stream generator works on a clock-controlled shift register combination. The clock to the shift registers is controlled by the feedback bits. The feedback bits E1, E2 & E3 control clocks to the respective shift registers while generating key streams encryption/decryption.

To generate the key stream, all LFSRs require to be fed with a non-zero value known as an initial vector value or an initial key bit sequence or key seed bits to generate a key stream. The stream of bits generated from such a generator is known as a pseudo-random bit sequence (PRBS). The total session key bits required for the proposed key stream generator is:

29-bits for 29-bit shift register +

23-bits for 23-bit shift register +

19-bits for 19-bit shift register +

2-bits for 29-bit LFSR mux +

2-bits for 23-bit LFSR mux +

2-bits for 19-bit LFSR mux

= 77 key bits

Thus, a session key of a 77-bit length is required to run the proposed symmetric key generator to encrypt or decrypt the data.

B. KEY GENERATION

This is an important phase of any crypto algorithm as the whole stream generator output depends on the key seed bits. The key generation is divided into three phases as:

i. BASIC KEY SEED GENERATION

The basic key seed is generated from two parameters, namely date and time. The formats of the chosen data and time are; DD:MM: YYYY (4 bytes) and HH:MM: SS (3 bytes). The length of the basic key seed is 32-bit.

The following rules are applicable to obtain basic key seed:

1. Get the date and put in binary coded decimal (BCD) format

For example:

Let the date be 16.10.2021

BCD form

0001 0110, 0001 0000, 0010 0000 0010 0001 =32 bits

2. Form 3 x 8 and 2 x 2 bits group as shown

0001 0110, 0001 0000, 0010 0000 00**10** 00**01** =32 bits

3. Get the time of the day

For example:

Let the Time of the day be 10:20:45

BCD Form

0001 0000, 0010 0000, 0100 0101 = 24 bits

4. Form 2 x 8 and 2 x 2 bits groups as shown

0001 0000, 0010 0000, 01**00** 01**01** = 24 bits

5. Exclusive OR the output of step 2 and 4 as shown

0001 0110, 0001 0000, 0010 0000 00**10** 00**01**

⊕

0001 0000, 0010 0000, ———— 0100 01**01**

————————————————————

0000 0110 0011 0000 0010 0000 011001**00**

d31————————————————d0

The basic key seed is represented as d31-d0.

ii. SESSION KEY GENERATION

Figure 5 depicts the structure of the session key generator derived from the main key stream generator.

The basic structure of the 3-LFSRs, is the same as in Figures 3 & 4. However, one of the LFSRs outputs is selected through a multiplexer to generate session keys. The key seed generator does not use any clock-controlled technique; instead, any one of the selected shift registers is clocked regularly. The basic key

seed bits d0-d1 (Marked in blue extreme right) and d4-d5 (Marked in blue) are used to select LFSR and feedback polynomial for the seed bits generation as shown in Table 2.

Table-2: Selection of LFSR and feedback polynomial

Sl No	d1	d0		d5	d4	Polynomial
1	0	0	19-bit	0	0	$x^{19} + x^5 + x^2 + x + 1$
				0	1	$x^{19} + x^{17} + x^{15} + x^{14} + x^{13} + x^{12} + x^6 + x + 1$
				1	0	$x^{19} + x^{17} + x^{15} + x^{14} + x^{13} + x^9 + x^8 + x^4 + x^2 + x + 1$
				1	1	$x^{19} + x^{16} + x^{13} + x^{11} + x^{10} + x^9 + x^4 + x + 1$
2	0	1	23-bit	0	0	$x^{23} + x^5 + 1$
				0	1	$x^{23} + x^{17} + x^{11} + x^9 + x^8 + x^5 + x^4 + x + 1$
				1	0	$x^{23} + x^5 + x^4 + x + 1$
				1	1	$x^{23} + x^{12} + x^5 + x^4 + 1$
3	1	0	29-bit	0	0	$x^{29} + x^2 + 1$
				0	1	$x^{29} + x^{20} + x^{11} + x^2 + 1$
				1	0	$x^{29} + x^{12} + x^7 + x^2 + 1$
				1	1	$x^{29} + x^{21} + x^5 + x^2 + 1$
4	1	1				Special case

Further, the bits d4-d31 will be considered as initial vector bits for 29-bit LFSR if selected, or d10-d31 will be considered as initial vector bits for 23-bit LFSR if selected, or d17-d31 will be considered as initial vector bits for 19-bit LFSR if selected. Run the session key generator (Fig-5) for about 128 clock cycles with appropriate basic key bits to flush out the known bits. A 77-bit session key is generated by re-running the session key generator for 77 clock cycles. This session key is designated as ks77-ks0.

iii. KEY STREAM GENERATION

The session key bits of 77-bits are divided into six fields as shown in Table 3.

Table-3: Session key bit's allocation

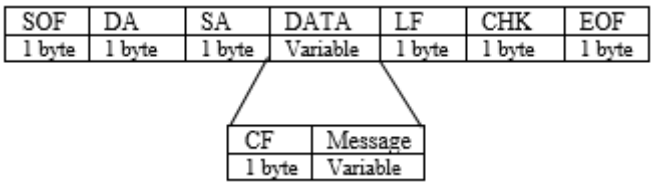
19-bit	23-bit	29-bit	Mux2	Mux1	Mux0
ks76-ks58	ks57-ks35	ks34-ks6	ks5-ks4	ks3-ks2	ks1-ks0
19	23	29	2	2	2
Initial vector bits			Feedback selection bits		

Accordingly, the session key is loaded in to the key stream generator depicted in Figure 4. Run the key stream generator by enabling the clock to either encrypt or decrypt the information bit by bit. The encryption/decryption process is depicted in Figure 6.

5. Secured Communication

Generally, data communication employs a frame-based protocol to ensure error and loss-free data exchange between the source and destination. In the proposed work, a variable length data frame is designed for this purpose. The structure is as depicted in Table-4.

Table-4: Data Frame Structure



SOF: Start of the frame and is a fixed known byte

DA: Destination address (Recipient)

SA: Source Address (Sender)

DATA: Data field (Control and Information)

CF: Control field to indicate command or message and is 1 byte. It can have 256 values and defined as required.

Message: Actual information to be exchanged

LF: Length of the data

CHK: Checksum of the data

EOF: End of frame and is a fixed known byte.

In the above frame, the message field is filled with encrypted/ciphered data. The length of the frame varies as the message length varies. The secured communication between the entities consists of mutual

authentication, encryption at the sender end and decryption at the receiver end.

A. AUTHENTICATION PROCESS

Each and every element in the smart grid is given a unique identification number (Id) and is not public. This identification code is represented by an 8-bit digital value. The cluster of elements, which regularly communicate with each other are all shared with their Id's in prior, not only for the purpose of authentication, but also to communicate with others. The authentication process requires the creation of a digital signature of any entity who wishes to communicate with the other. For example, if Alice wants to communicate with Bob, Alice creates its digital signature using its unique Id and sends it to Bob. Then Bob, decrypts the digital signature to get the Id by the same mechanism Alice used to create it. Then Bob compares the stored Id of Alice with a decrypted digital signature and confirms the request is indeed came from Alice. To create a digital signature and to complete the process of authentication, the following procedure must be followed:

1. Divide the computed basic seed key bits of 32 into 4 x 8 bits groups and compute checksum (chk_7-chk_0)/hash value.
2. Now, encrypt the Id of the entity, (that wishes to communicate) with the checksum/hash value computed in (1) above.
3. This encrypted Id forms a digital signature.
4. Append this digital signature along with date and time in the data field.
5. Send a request for communication to the recipient along with a digital signature and other data.
6. Recipient will compute the base seed key form the received date and time stamp as described in section 4B(i).
7. Computes checksum and decrypts the digital signature to retrieve the Id.
8. If Id is valid, it accepts the communication requests and sends reply with 'ready' message.
9. If Id is not valid, it discards the frame and activates the flag of alarm.

Algorithm 1 *Digital Signature for Authentication*

Initialization: chk= [chk7, chk6….chk0] = [0];

Id= [Id7, Id6…. Id0]; ds= [ds7, ds6….ds0] = [0];

begin:

Get basic key seed of 32-bits;

Divide them as 4 x 8 groups of (d31-d24), (d23-d16), (d15-d8) and (d7-d0).

Compute checksum/ hash value as:

chk=[chk7-chk0] = [(d31-d24) ⊕ (d23-d16) ⊕ (d15-d8) ⊕ (d7-d0)];

Compute digital signature by encrypting Id with computed checksum/hash value as:

Read Id;

Compute

[ds7, ds6….ds0] = **[(chk7, chk6….chk0)** ⊕ (**Id7, Id6…. Id0)];**

ds= [ds7, ds6….ds0] => digital signature

Figure 7 indicates the one-way authentication process, i.e., Bob authenticates Alice. The authentication process can be either one-way or both ways as the need arises. The example of Alice and Bob is considered in this figure to demonstrate the authentication process. If authentication of Bob is also a requirement, same procedure has to be followed from Bob to Alice. On completion of the authentication phase, the encryption process will be initiated from the sender end.

B. ENCRYPTION/DECRYPTION PROCESS

On completion of authentication and accepting the communication request, information is exchanged between the sender and receiver over an established secured channel.

i. ALGORITHM FOR ENCRYPTION/DECRYPTION

Since, the proposed key stream generator is symmetric, both encryption, and decryption functions utilise common/same key. The encryption and decryption algorithm are outlined as below:

Algorithm 2 *Encryption and decryption*

Initialization: ks= [ks76.ks75…. ks0] = 0; Session key set to zero

Session key generation:

Configure session key generator according to the generated basic key seed ‘d’=d31-d0; Table-2 and the Figure 5.

Run the configured generator for 128 clock cycles; To flush out initial bits

Run the session key stream generator for 77 clock cycles and collect

ks= ks76-ks0; 77-bits of session key

Configure the key stream generator shown in Figure 5 with session key bits allocation as shown in Table-3.

Run key stream generator:

while (p-k) <0 **do**

// where ‘p’ represents the plain text bits & ‘k’ key bits while encrypting.

//for decryption function is (c-k), where ‘c’ represents ciphered bits and ‘k’ key bits while decrypting

for (i=0; i<n; i++) **do**

// where n= number of key bits required to encrypt/decrypt the plain text/cipher text bit by bit.

{

Get E1, E2 & E3

if (E1=1) {clk-2=1}; //Clock to LFSR-2

else {clk-2=0}; // No Clock to LFSR-2

if (E2=1) {clk-3=1]; //Clock to LFSR-3

else {clk-3=0}; //No Clock to LFSR-3

if (E3=1) {clk-1=1}; //Clock to LFSR-1

else {clk-1=0}; //No Clock to LFSR-1

if (Out-3 =1)

{Ki= Out-2};

else

{Ki= Out-1};

Ci= Pi ⊕ Ki; // For Encryption

Or

Pi= Ci ⊕ Ki;// For Decryption

}

}

1. **ENCRYPTION/DECRYPTION PROCESS**

The encryption and decryption process are very straightforward in the proposed work. The data encryption process is defined as:

$C = Ek(P);$ (1)

Where C= Ciphered text

Ek= Encryption function

P= Plain text

$Ci = Pi \oplus Ki;$ (2)

Where i=n, equal to the number of bits to be encrypted.

The data decryption is defined as:

Decrypt the encrypted data (ciphered data) with key stream as:

$P = Dk (C)$ (3)

Where Dk is decryption function

C ciphered/encrypted data

$P = Dk (Ek(P))$ (4)

$Pi = Ci \oplus Ki;$ (5)

Where i=n, equal to the number of bits to be decrypted.

According to the encryption algorithm described in the previous section the data encryption process is as follows:

a. The symmetric key stream generator depicted in Figure 5 generates the main encryption key bits equal to that of the plain text bit by bit.
b. The plain text denoted by Pi is read bit by bit to encrypt, where ‘i' represents the number of bits in plain text.
c. Encryption: $Ci = Pi \oplus Ki;$ as in (2).
d. Then, the encrypted data is transmitted to the receiver as per the data frame shown in Table-4.

The decryption process is as follows:

a. Receive the data frame,
b. Retrieve the encrypted data from the data frame.
c. Read the encrypted data bit by bit.
d. Decryption: Pi= Ci $\oplus$ Ki; as in (5).

6. Results And Analysis

Figure 8 shows the experimental setup, in which the proposed key stream was implemented.

The hardware setup consists of a popular Arduino nano-board with a built-in reset and operating power circuit. The 16 x 2 liquid crystal display is provided to display the progress of the key stream generator process starting from basic seed generation to the main key stream bit generation. The Arduino integrated development environment (IDE) is utilized to implement the key stream generator. The compiled code was ported on to the Arduino nano development board. The experimental setup also has a current sensor to estimate the current consumption as it is one of the prime parameters to decide the resource constraint of the algorithm. The power consumption is also an important parameter to be considered in hardware design. Screen shots of the experimental process of key stream generation are shown in Figure 9.

The experimental setup was used to generate three keystreams of 1MB each. The summary of hardware resource utilization is shown in Table-5.

Table-5: Resources Utilization-Results

Attribute	Specifications	Utilization	Remark
Microcontroller	ATmega 328		8-bit
Frequency	16Mhz		
Flash memory	32KB	8.2KB	25%
SRAM	2KB	235 Bytes	11.45%
EEPROM	1KB	Nil	
Power	5V	155mA@5V	0.775mW
Digital I/O	16	8	50%
Analog I/O	6	2	33.33%
Time of execution		<5sec (approx.)	1 MB

The amount of effort involved in terms of code execution; resource utilization has been great as can be understood from the Table-5.

The suitability of the proposed key stream generator for the security of smart meters and autonomous vehicles requires an in-depth analysis. The objective of the proposed key stream generator is to be lightweight in terms of cost, complexity, ease of implementation, time of execution and so on. Therefore, the analysis of the proposed key stream generator is divided into various phases as follows:

A. MATHEMATICAL ANALYSIS

The mathematical analysis of the proposed key stream generator is to determine the values of some of the attributes like length of the LFSR, its period and linear complexity. These parameters are required to understand the generator's capability to provide the secrecy of the information. The attributes of the key stream generator are as follows:

i. MAXIMUM LENGTH OF LFSR

The properties of the key stream generator can only be revealed if it satisfies certain conditions of algebra. The definitions of algebra, which indicate the necessary conditions to consider the key stream generator for the cryptographic applications are as follows.

Definition 1: A prime number is a positive integer with exactly two positive divisors. If p is a prime then its only two divisors are necessarily 1 and p itself, since every number is divisible by 1 and itself.

The first ten primes are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

Definition 2: The LFSR is maximal-length if and only if the corresponding feedback polynomial is primitive. This means that the following conditions are necessary (but not sufficient):

- The number of taps is even.
- The set of taps is set wise co-prime; i.e., there must be no divisor other than 1 common to all taps.

Definition 3: Let a and n be integers greater than one. If a^n-1 is prime, then a is 2 and n is prime.

Theorem 1: Let LFSR-1, LFSR-2 and LFSR-3 have a primitive feedback polynomial and a nonzero initial state.

If 2^{29}-1, 2^{23}-1 and 2^{19}-1 are prime (Definition-3) then the length of each LFSR is 2^{29}-1, 2^{23}-1 and 2^{19}-1 respectively.

ii. PERIOD OF KEY STREAM GENERATOR

The key stream generator must have a large period to be considered for cryptographic applications. The proposed key stream generator is designed using linear feedback shift registers with a nonlinear output function represented by a 2:1 multiplexer as shown in Figure 3 & 4. The clock to the shift registers is controlled by means of feedback bit generated using selective polynomials which would introduce higher confusion & diffusion in the bit stream output of the key stream generator.

Theorem 2:

If (l,m,n,)=1, then its period is $(2^{l}-1)(2^{m}-1)(2^{n}-1)$

Thus, basic generator has a period of:

Period= $(2^{L1}-1) \times (2^{L2}-1) \times (2^{L3}-1)$ (6)

i.e., Period= $(2^{29}-1) \times (2^{23}-1) \times (2^{19}-1)$ 2^{71}

To increase the level of complexity for attackers on the key stream generator, multiple feedback polynomials have been included in the design. Thus, the period increases by 2^{12} and avoids the issue of being insecure. Therefore, the period of the proposed key stream is approximately = $[(2^{29}-1) \times (2^{23}-1) \times (2^{19}-1)] \times 2^{12} = 2^{83}$.

iii. LINEAR COMPLEXITY

This is another important property that the keystream generator has to possess for it to be considered suitable for cryptographic applications. The linear complexity must be large but does not guarantee the randomness properties of the sequence under consideration.

Definition-4: Let Z = z0, z1, z2, z3, . . . be a finite or infinite sequence. Then the linear complexity LC(Z) of Z is the length of the shortest LFSR which generates it.

The proposed key generator has the multiplexer at the output of the key stream generator and forms the nonlinear combining function. The algebraic normal from (ANF) of the multiplexer is:

f (x1, x2, x3) = x1 x2 x2 x3 x3 (7)

Therefore, the LC of the basic element is:

LC= L1x L2 + L2 L3 + L3 (8)

i.e. LC = 29 x 23 + 23 x 19 + 19

LC= 1123= 2^{10} (approximately)

Since the proposed key stream generator has 4 primitive polynomials at the feedback of each LFSR, that increases the LC by a factor of 4 roughly. Overall complexity increases by a factor of 2^{12} approximately.

B. EXPERIMENTAL ANALYSIS

Experimental analysis is an important tool to investigate key stream generator performance in terms of resources utilization, execution time, ease of implementation, and so on. The analysis is divided into categories as follows:

i. IMPLEMENTATION ANALYSIS

The firmware code written in the high language ‘C’ is about 25% of the total programming memory of 32KB.The SRAM is about 11.45%. The code optimization technique was not adopted while writing the firmware. The power consumption of the experimental setup was about 775mW. This is extremely good for any stream generator to run. The resources utilization would have been much better with a code optimization technique in place, which could be a next attempt to improve the performance of the key stream generator for the smart meter and autonomous vehicle security. On the whole, hardware resource utilization is quite satisfactory. The results of the implementation in terms of hardware resources utilization are shown in Figure 10.

ii. STATITSTICAL ANALYSIS

The National Institute of Standards and Technology (NIST) has prescribed certain statistical properties, which would reveal the weakness of the key stream generator. Statistical tests have been formulated to test a specific **null hypothesis,** denoted by **H0**. The null hypothesis for conducting tests on the output bit stream of the proposed key stream generator is that it is random. An alternative hypothesis (**Ha**) to this null hypothesis is the bit stream not being random. The null hypothesis defines a critical value that can be used to compare with the determined statistic value of the bit stream to decide whether the bit stream is random or non-random. If the test statistic value exceeds the critical value, the null hypothesis for randomness is rejected. Otherwise, the null hypothesis is not rejected. The statistical test results are probabilistic and hence there is an attachment of called level of significance. It is denoted by ‘α’ and it is fixed at 0.05 in this paper. It is the probability that the test will indicate that the bitstream is not random, even though really it is. A theoretical reference distribution of the possible probability values for each test varies. The important statistical tests are frequency test (Monobit test), Serial test, Poker test, Runs test, and Autocorrelation tests. Each test mentioned here determines whether the key stream bits possess a certain attribute that a truly random sequence would exhibit; the conclusion of each test is not definite, but rather probabilistic. Based on the outcome of the test attribute, the sequence may be declared as random or non-random. To conduct these tests on the key stream generator, at least each block size must be >10,000 bits. Therefore, the keystream generator output of size 1Mb was chosen for experimental analysis. The whole data file was divided into number blocks of different sizes while conducting the minimum tests as per NIST standard.

To conduct statistical tests, a test suite was developed and the screen shots of the same are shown in Figure 11.

Table-6: Statistical Test Results

Size of the input file		1MB (1024kb)				
Test	Block Size	No of Blocks	Seq1	Seq2	Seq3	Inf
Frequency	10k	100	93	91	90	R
			7	8	10	NR
	25k	50	47	45	49	R
			3	5	1	NR
	50k	20	16	18	19	R
			4	2	1	NR
	100k	10	9	8	8	R
			1	2	2	NR
Serial (di-bit)	10k	100	89	97	94	R
			11	3	6	NR
	25k	50	44	47	49	R
			6	3	1	NR
	50k	20	16	19	16	R
			4	1	4	NR
	100k	10	8	9	6	R
			2	1	4	NR
Poker’s	10k	100	92	98	95	R
			8	2	5	NR
	25k	50	48	46	44	R
			2	4	6	NR
	50k	20	19	18	16	R
			1	2	4	NR
	100k	10	8	8	9	R
			2	2	1	NR
Runs	10k	100	89	92	98	R
			11	8	2	NR
	25k	50	47	47	45	R

			3	3	5	NR
	50k	20	17	18	15	R
			3	2	5	NR
	100k	10	9	8	9	R
			1	2	1	NR
Autocorrelation	10k	100	88	95	96	R
			12	5	4	NR
	25k	50	43	47	49	R
			7	3	1	NR
	50k	20	18	16	19	R
			2	4	1	NR
	100k	10	9	8	6	R
			1	2	4	NR

Note 1k=1024 bits

Note: R→ Random, NR→ Non-Random, Inf→ Inference.

Three test sequences of size 1MB each were generated from the hardware setup and subjected to statistical tests. Table-6 above, summarizes the details of the tests carried out on the output of the proposed key stream generator and their results. Graphical representation of the results is shown in Figure 12.

Table-7 shows the significance value, expected probability value for each test along with the result of probability i.e., satisfied or not.

Table-7: Statistical Test Probability Value-Results

Significance value α	0.05	
Name of the test	Expected Probability Value	Results of all three sequences
Frequency	3.8415	Satisfied
Serial (di-bit)	5.9915	Satisfied
Poker's	14.071	Satisfied
Runs	9.4877	Satisfied
Autocorrelation	1.6449	Satisfied

The statistical test suite was developed with a python programming language and used on Windows platform. The time taken by the suite to complete each test with different block sizes is summarized in Table-8 and graphical representation in Figure 13.

Table-8: Time taken for each test-Results

Size of the input file	1MB (1024kb)			
Name of the test /Block size	10k	25k	50k	100k
Frequency	2.5ms	4.8ms	7.8ms	10.5ms
Serial (di-bit)	4.3ms	9.2ms	15.7ms	23.5ms
Poker's	11.4ms	20.6ms	34.8ms	50.2ms
Runs	18.7ms	37.2ms	52.8ms	68ms
Autocorrelation	26.8ms	49.5ms	67.2ms	82.4ms

The statistical test results indicate that the proposed key stream generator produces a different key bit stream with different basic key seed. Also, this suggests that the generated key stream bits almost satisfy the statistical criteria for being random. The analysis is pretty simple that the proposed key stream generator produces a random bitstream indeed.

C. SECURITY ANALYSIS

Security analysis is another important step in validating the suitability of the keystream generator for the purpose of information security. Validation attributes are with respect to certain well-known attacks such as Eaves dropping, man-in- the middle, etc. The other attributes, which makes the proposed algorithm are also presented here.

i. **Resistivity:** The key stream generator is designed using LFSRs with primitive polynomials. As seen from the mathematical analysis, these LFSR's are of maximum length and period. The period of the key stream generator is 2^{83} which is sufficiently large enough to resist attack.

ii. **Berlekamp-Massey attack:** The linear complexity of the generator is about 2^{25} (2^{11} x 2^{14}). It indicates that the 2^{25} plain text bits are required to perform the Berlekamp-Massey attack.

iii. **Privacy Against Eves dropping:** Since the communication is secured with encryption algorithms, it would be difficult to eaves dropping.

iv. **Protection Against Man-in-the Middle Attack:** As described earlier, authentication of communication entities involves the dynamically determined digital signature. The hash value (chk =[chk7-chk0] is generated from a unique algorithm as described earlier. This is used to encrypt a unique Id=[Id7-Id0] to create a digital signature ds=[ds7-ds0].

The receiver, performs the decryption of the digital signature to get the 'Id' of the sender and compares it with the stored, non-public 'Id' to determine the source of the information is legitimate. The MITM is very difficult due to the fact that the digital signature creation procedure is unique and the Ids are not public.

v. **Forward secrecy:** Since session key generation is unique, dynamic and used only once, forward, secrecy of the information is guaranteed.
vi. **Malicious communication**: The injection of malicious communication is not possible, as the communication is based on a unique data frame of variable length. Since, data is secured using encryption process and modification in the data frame results in wrong decryption of the information resulting in declaration of invalid data frame. Thus, it is safe against malicious communication injection or modification of data frames.
vii. **Session Key Security:** Since session key generation is dynamic and used only once, it would be difficult to capture by any adversary as it is not shared on the link.
viii. **Prior Key Distribution:** There is no prior key distribution of any type. This makes the job of the adversary very difficult to hack into the link.

D. COMPARATIVE ANALYSIS

The related work on the security schemes was an opportunity to examine some of the selected works of previous researchers in detail with respect to their ability to perform in the smart grid and autonomous vehicle environments. The comparative study of those selected works and the proposed work is summarized in Table-9.

Table-9: Comparative Study of schemes

Attribute	Proposed	[1]	[7]	[8]	[9]
A1	NK	Yes	No	No	No
A2	Yes	Yes	Yes	Yes	Yes
A3	Yes	Yes	Yes	Yes	Yes
A4	Yes	Yes	Yes	Yes	Yes
A5	Yes	Yes	No	Yes	Yes
A6	No	Yes	Yes	Yes	Yes
A7	No	No	Yes	Yes	Yes
A1	Protection against quantum attack				
A2	Privacy against eaves dropper				
A3	Protection against man-in-the middle				
A4	Forward secrecy				
A5	Session Key security				
A6	Prior Key distribution				
A7	Session key sharing				
NK	Not known.				

The comparative study reveals that, most of the selected work's dependent on prior distribution of key data, session key sharing and so on. The analysis completed in previous sections indicates that the authentication and encryption process can withstand most of the cryptographic attacks, since no sharing of any kind of information is used to start these processes.

E. COMPARATIVE ANALYSIS OF PROPOSED AND CLASSICAL ALGORITHMS

Since, few of the previous studies recommended the classical algorithms such as data encryption standard (DES) and advanced encryption standard (AES) to use in smart grid applications. The comparative analysis of classical algorithms and proposed work is summarized in Table-10

Table-10: Comparative analysis of classical and proposed algorithms

Parameter	DES	AES	Proposed
Key length	56	128-192-256	77
Key used	same	same	same
Block size	64 bits	128 bits	Single bit
Structure	Feistel	Substitution	Multiplexed LFSR
Scalability	Scalable	No	Scalable
Security	Suitable	Excellent	Suitable
Speed of execution	Low	Low	Fast
Encryption/decryption	Yes	Yes	Yes
Authentication	No	No	Yes

The comparative analysis indicates that the classical algorithms are more complex to implement as they have multiple rounds to execute. In summary, proposed work has the lowest cost of time compared to classical algorithms.

7. Discussion And Evaluation

Previous sections of this paper presented the detailed design, and analysis of the proposed symmetric key stream generator with respect to ease of implementation, time of execution, resources utilization etc. The complete analysis, mathematical, experimental, security, comparative of the proposed keystream generator was done to ascertain its suitability for the smart meter and autonomous vehicle applications. The detailed analysis of the proposed symmetric key stream has the following features:

i. Dynamic and random basic key generation,
ii. Dynamic session key generation and used only once.
iii. One-way and two-way authentication with unique hash value computation.
iv. Communication is based on variable length data frame. Low power, Low cost and High performance.

The above features have the following advantages in terms of physical resources requirements:

1. Key generation centre is not required.
2. Key distribution facilities are not required.
3. No key storage requirement.
4. No prior distribution of any type of key involved.
5. No PKI for authentication.
6. No human resource requirement.

The discussion and evaluation of the proposed symmetric key stream generator indeed suggests that it is suitable for the security of smart meters, autonomous vehicle/connected vehicle applications for the obvious reasons.

8. Conclusion

The paper presents the design, analysis, and evaluation of a symmetric key stream generator, which is truly lightweight in all possible ways. Its simplicity, ease of implementation, resource utilisation, both in terms of hardware & firmware makes it truly suitable for smart meters, autonomous or connected vehicle security. The dynamic, random basic key generation, unique technique to determine hash values, and session key generation are the major achievements of the proposed work. The scheme also uses a dynamically generated secret key for securing the data for every frame or session; thus, the scheme is strong and provides sufficient security to the data. The analysis of cryptographic properties as per NIST standard ensures security against passive and active attacks. Since the proposed scheme is light weight, it is much suited for real time applications where time and security are the major concerns. A few of the key features of the developed symmetric key stream generator are:

a. The cryptographic algorithm does not store any data related to encryption/decryption keys, thus avoiding requirements of high-cost memory.
b. Does not use any PKI-based authentication but still authenticates the users.
c. A randomly generated key seed is used for both authentication and session key generation.
d. The key seed and the session keys are used only once and no repetition.

Declarations

ACKNOWLEDGEMENT

The authors wish to thank REVA University for providing the facilities to carry out the research work. The proposed work has been submitted to the patent authorities for publications.

References

1. Yuancheng Li, Pan Zhang, and Rong Huang “Lightweight Quantum Encryption for Secure Transmission of Power Data in Smart Grid”, DOI 10.1109/ACCESS.2019.2893056, *IEEE Access*.
2. Muktadir Chowdhury, Ashlesh Gawande andLan Wang, “Secure Information Sharing among Autonomous Vehicles in NDN”, *The 2nd ACM/IEEE International Conference on Internet-of-.ings Design and Implementation, Pi.sburgh, PA USA,* April 2017 (IoTDI 2017), 11 pages. DOI: 10.1145/3054977.3054994
3. Otisitswe Kebotogetse , Ravi Samikannu and Abid Yahya, “Review of key management techniques for advanced metering infrastructure” *International Journal of Distributed Sensor Networks 2021, Vol.*

17(8) The Author(s) 2021. DOI: 10.1177/15501477211041541 journals.sagepub.com/home/dsn

4. Chi Cheng, Yue Qin, Rongxing Lu, Tao Jiang, and Tsuyoshi Takagi “Batten Down the Hatches: Securing Neighborhood Area Networks of Smart Grid in the Quantum Era”, *IEEE Transactions on Smart Grid,* Vol. 10, No. 6, November 2019, 1949-3053 _c 2019 IEEE.
5. Dominik Engel and Günther Eibl, “Wavelet-Based Multiresolution Smart Meter Privacy”, *IEEE Transactions on Smart Grid* 10.1109/TSG.2015.2504395, 1949-3053 _c 2015 IEEE.
6. Hesham Aly El Zouka, and Mustafa Mohamed Hosni, “Time Granularity-based Privacy Protection for Cloud Metering Systems”,*Advances in Science, Technology and Engineering Systems Journal* Vol. 5, No. 6, 1278-1285 (2020)
7. Amritha Puliadi Premnath, Ju-Yeon Jo, Yoohwan Kim “Application of NTRU Cryptographic Algorithm for SCADA security” 2014 11th *International Conference on Information Technology: New Generations* 978-1-4799-3187-3/14 $31.00 © 2014 IEEE, DOI 10.1109/ITNG.2014.38.
8. Jia-Lun Tsai and Nai-Wei Lo, “Secure Anonymous Key Distribution Scheme for Smart Grid” *IEEE Transactions on Smart Grid,* Doi: 10.1109/TSG.2015.2440658, 1949-3053 _c 2015 IEEE.
9. Vanga Odelu, Ashok Kumar Das, Mohammad Wazid, and Mauro Conti, “Provably Secure Authenticated Key Agreement Scheme for Smart Grid”, *IEEE Transactions on Smart Grid,* DOI 10.1109/TSG.2016.2602282, 1949-3053 (c) 2016 IEEE.
10. Neetesh Saxena and Santiago Grijalva, “Dynamic Secrets and Secret Keys Based Scheme for Securing Last Mile Smart Grid Wireless Communication”, *IEEE Transactions on Industrial Informatics*, DOI 10.1109/TII.2016.2610950, 1551-3203 (c) 2016 IEEE.
11. Yu Meng, Rao Yao, Liu Jiaji, Zhu Liangliang and Yi Yengxian, “Research on Fast Encryption Method for Smart Energy Management System in Smart Grid”,*2021 IEEE 3rd International Conference on Communications, Information System, and Computer Engineering* (CISCE 2021), 978-0-7381-1215-2/21/# 31.00 © 2021 IEEE.
12. Abdullahi Chowdhury, Gour Karmakar, Joarder Kamruzzaman, Alireza Jolfaei and Rajkumar Das, “Attacks on Self-Driving Cars and Their Countermeasures: A Survey”, *IEEE Access,* Digital Object Identifier 10.1109/ACCESS.2020.3037705Vol 8, 2020
13. Sahand Murad, Asiya Khan, Stavros Shiaeles and Giovanni Masala, “Data Encryption and Fragmentation in Autonomous Vehicles using Raspberry Pi 3”, *2019 IEEE World Congress on Services (SERVICES),* 978-1-7281-3851-0/19/$31.00 ©2019 IEEE, DOI 10.1109/SERVICES.2019.00058
14. Joonsang Yoo and Jeong Hyun Yi, “Code-Based Authentication Scheme for Lightweight Integrity Checking of Smart Vehicles”, *IEEE Access,* Vol 6, 2018, Digital Object Identifier 10.1109/Access.2018.2866626, 2169-3536, 2018 IEEE.
15. Mahdi Dibaei, Xi Zheng, Kun Jiang, Robert Abbas, Shigang Liu, Yuexin Zhang, Yang Xiang, and Shui Yu, “Attacks and defences on intelligent connected vehicles: a survey”, *Digital Communications and Networks (2020)*, doi: https://doi.org/10.1016/j.dcan.2020.04.007.

16. *Jamal Raiyn, "*Data and Cyber Security in Autonomous Vehicle Networks", *Transport and Telecommunication, 2018,* volume 19, no. 4, 325–334, DOI 10.2478/ttj-2018-0027.
17. Fabio Arena_, Giovanni Pau, and Mario Collotta,"A survey on driverless vehicles: from their diffusion to security Features", *Journal of Internet Services and Information Security (JISIS)*, volume: 8, number: 3 (August 2018), pp. 1-19.
18. Barry Sheehan, Finbarr Murphy, Martin Mullins, Cian Ryan," Connected and autonomous vehicles: A cyber-risk classification Framework", *Transportation Research Part A,* https://doi.org/10.1016/j.tra.2018.06.033
19. Belén Martínez-Rodríguez, Sonia Bilbao-Arechabala and Fernando Jorge-Hernandez, "Security Architecture for Swarms of Autonomous Vehicles in Smart Farming", *Appllied Science. 2021*, 11, 4341. https://doi.org/10.3390/app11104341
20. Qiyi He 1, Xiaolin Meng, Rong Qu and Ruijie Xi, "Machine Learning-Based Detection for Cyber Security Attacks on Connected and Autonomous Vehicles", *Mathematics 2020*, 8, 1311; doi:10.3390/math8081311
21. Ahmer Khan Jadoon , Licheng Wang ,Tong Li, and Muhammad Azam Zia, "Lightweight Cryptographic Techniques for Automotive Cybersecurity", *Hindawi, Wireless Communications and Mobile Computing* Volume 2018, Article ID 1640167, 15 pages https://doi.org/10.1155/2018/1640167
22. Mustafa Saed Kevin Daimi and Samar Bayan, "A Survey of Autonomous Vehicle Technology and Security", VEHICULAR 2019 : *The Eighth International Conference on Advances in Vehicular Systems, Technologies and Applications,* Copyright (c) IARIA, 2019. ISBN: 978-1-61208-720-7 39
23. Manar Abu Talib, Sohail Abbas, Qassim Nasir and Mohamad Fouzi Mowakeh, "Systematic literature review on Internet-of-Vehicles communication Security", *Internet of Vehicles: Architectures, Applications and Challenges - Research Article, International Journal of Distributed Sensor Networks* 2018, Vol. 14(12), DOI: 10.1177/1550147718815054.
24. Yosef Ashibani, Qusay H. Mahmoud "Cyber physical systems security: Analysis, challenges and solutions", *ELSEVIER, Computers and Security,* 0167-4048/© 2017 Elsevier Ltd.

Figures

Figure 1

Communication between SM & DC over WiFi Network

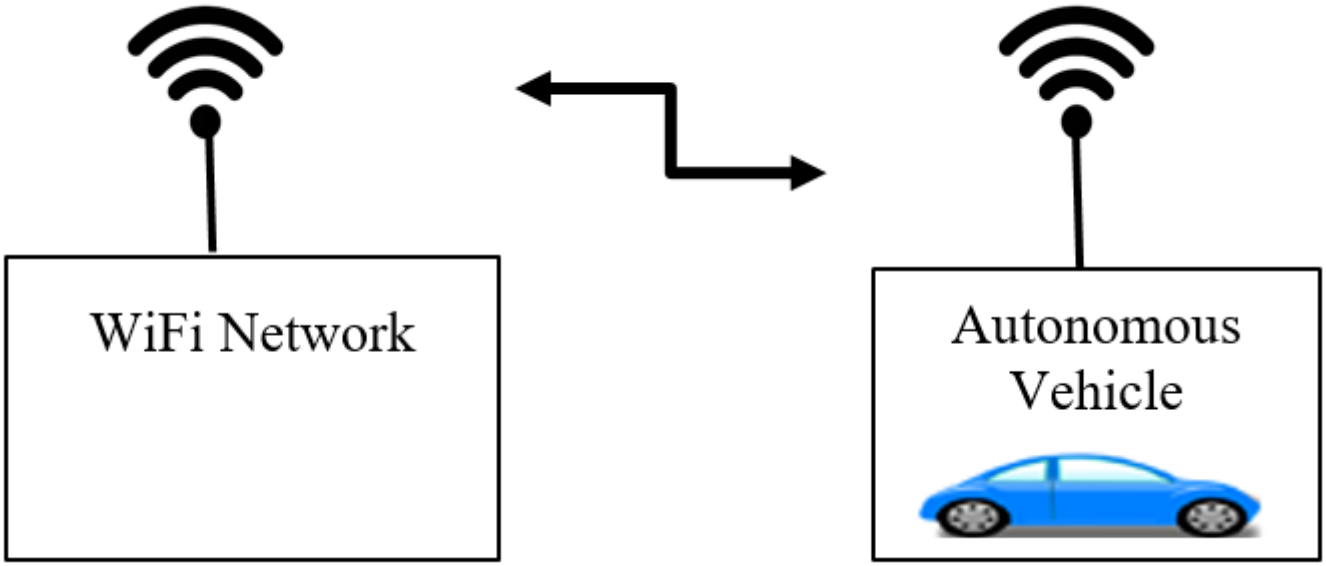


Figure 2

Communication between AV/CV and Wi-Fi NW.

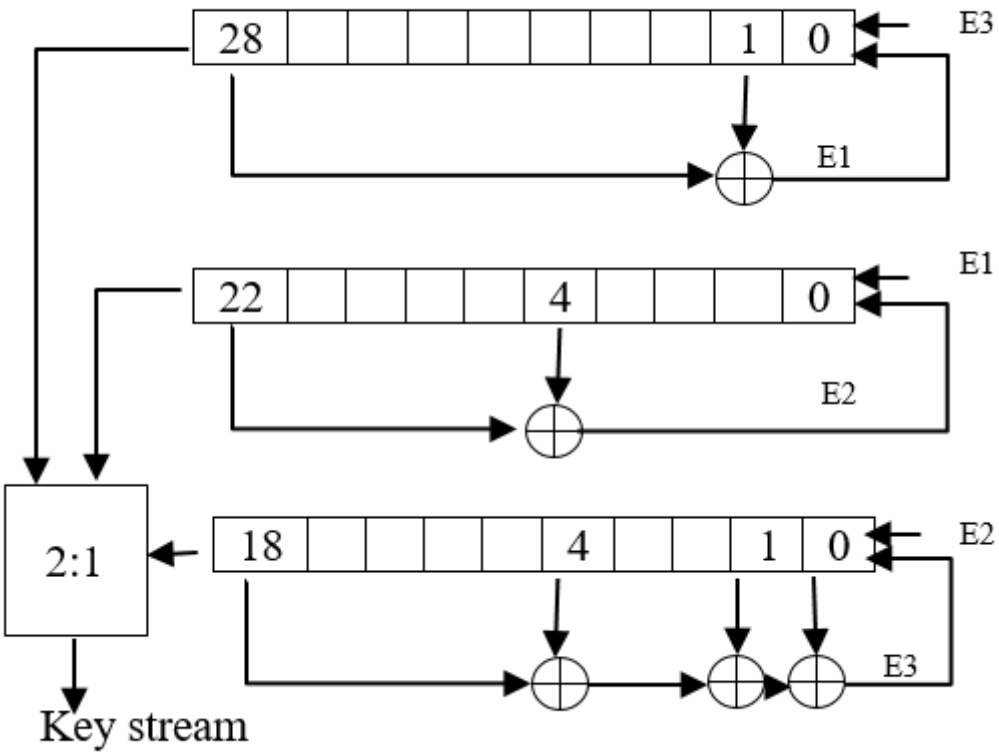


Figure 3

Structure of Keystream Generator with single polynomial for each of the LFSR.

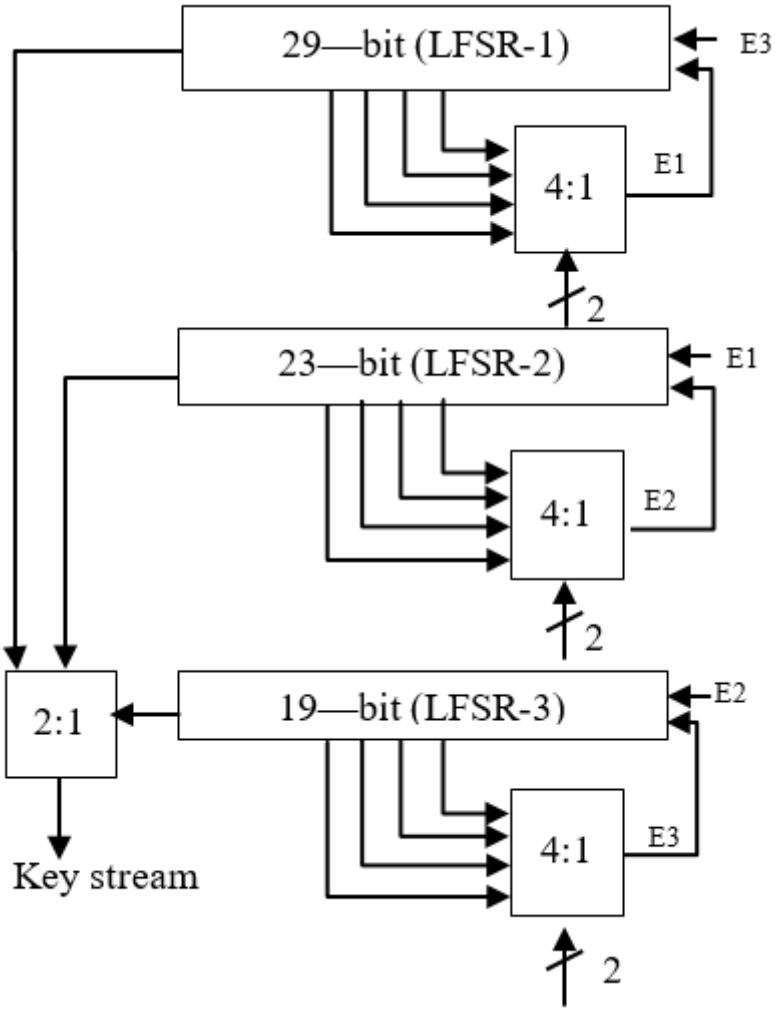


Figure 4

Key stream generator with multiple feedback polynomials

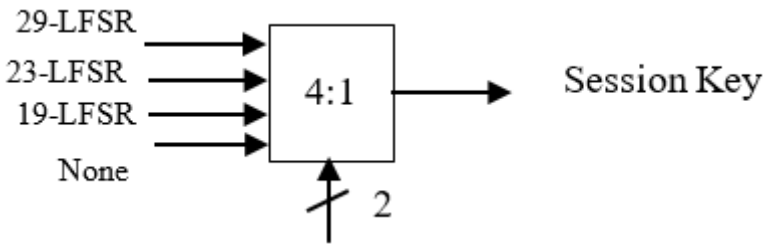


Figure 5

Structure of Session Key Generator

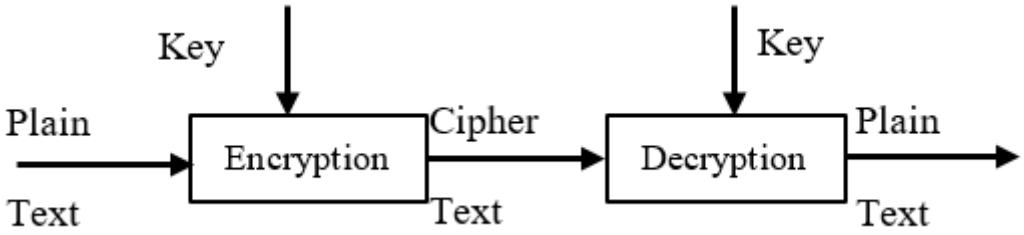


Figure 6

Encryption/Decryption Block Diagram

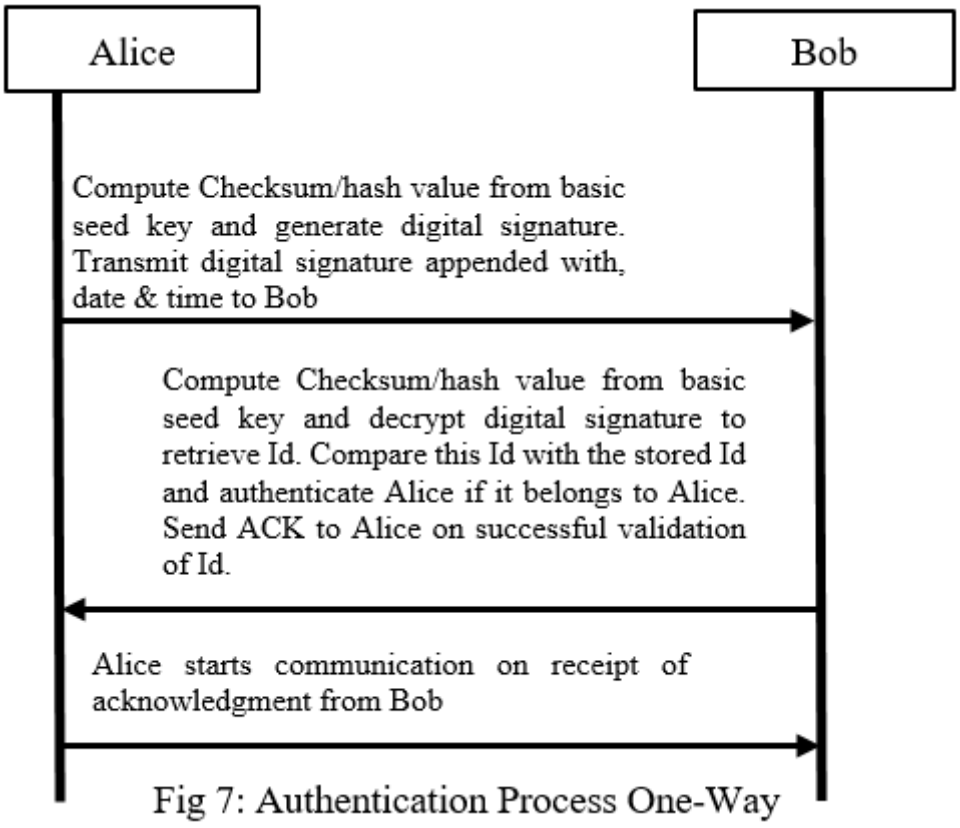


Figure 7

Authentication Process One-Way

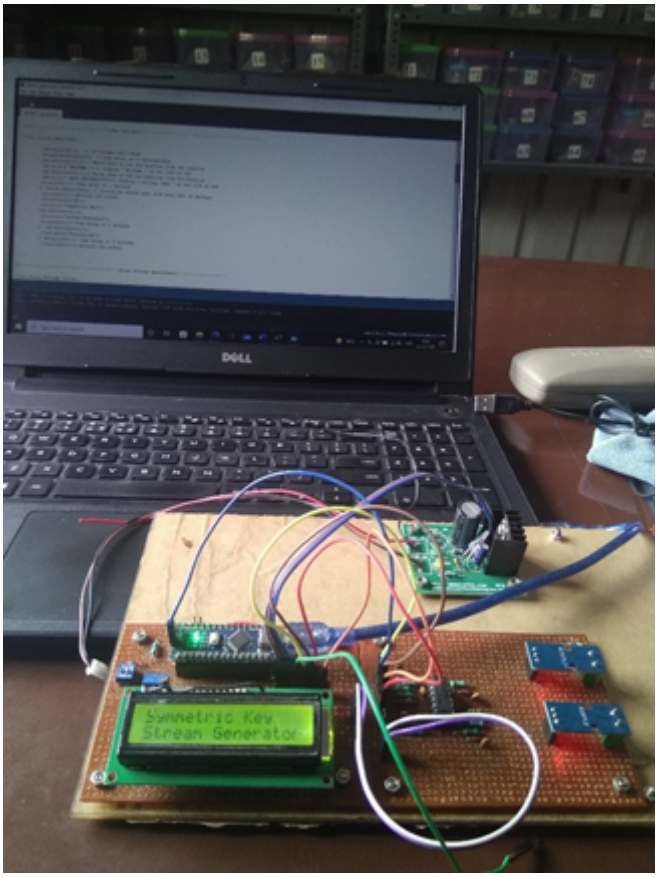


Figure 8

Hardware Setup

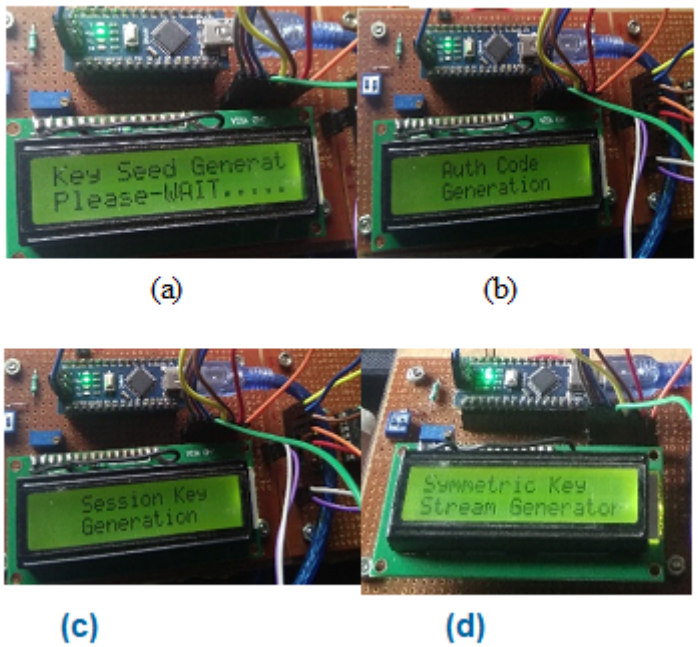


Figure 9

Screen shots of Key Stream Generation Stages

(a) Key Seed Generation

(b) Authentication/Hash Value Generation

(c) Session Key Generation and

(d) Main Key Stream Generation

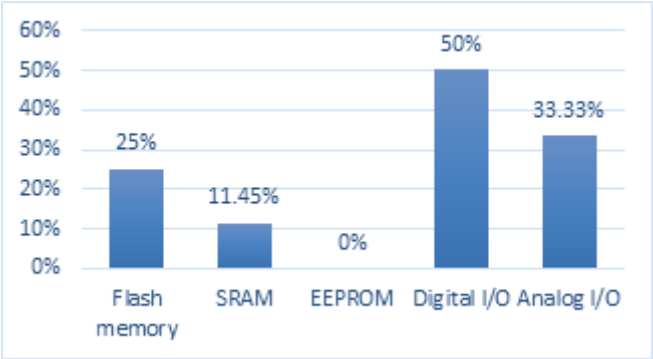


Figure 10

Hardware resources utilization

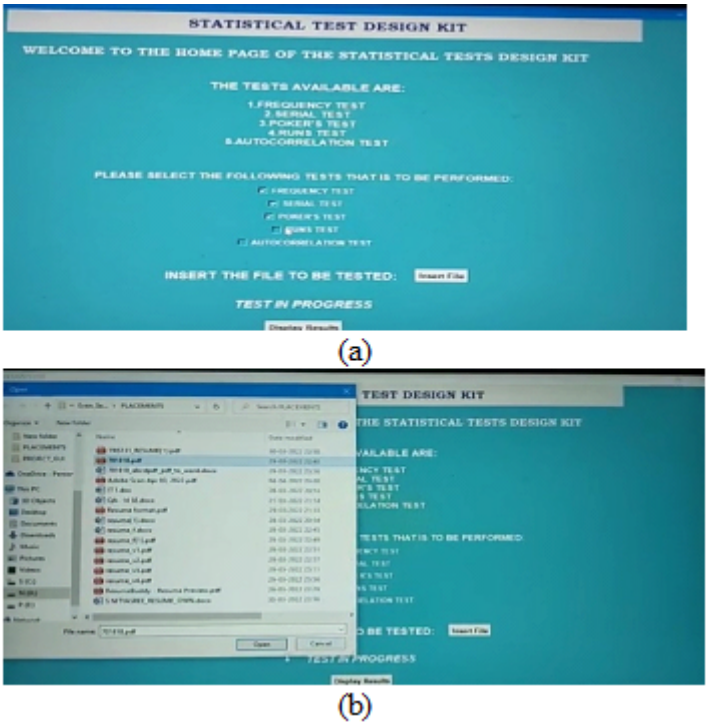


Figure 11

Screen shots of Statistical Test Suite

(a) Main Screen with list of tests

(b) File Selection

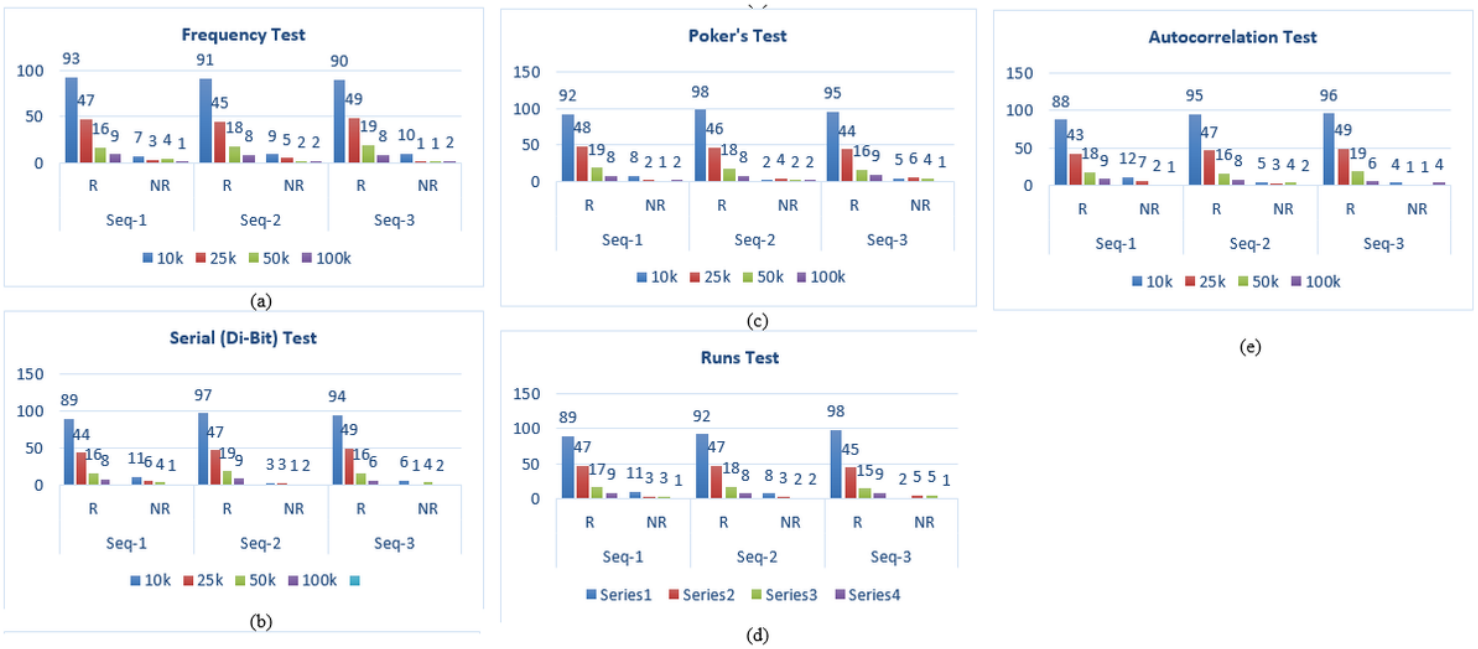


Figure 12

Statistical Test Distribution-Results

(a) Frequency Test

(b) Serial (Di-Bit) Test

(c) Poker's Test

(d) Autocorrelation Test and

(e) Runs Test.

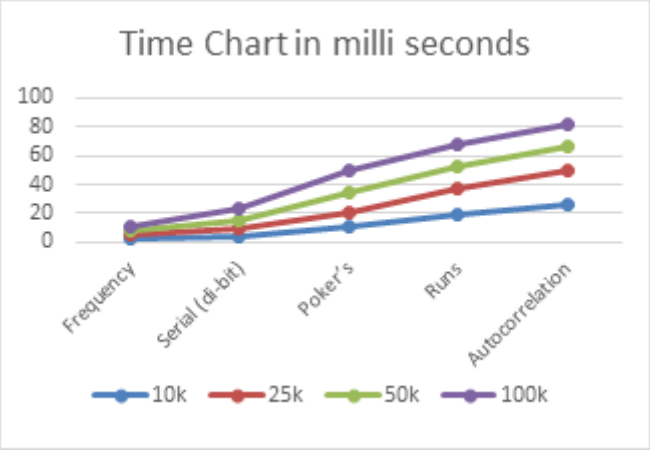


Figure 13

Time spent on each test