ARSMS: A Hybrid Secured SMS Protocol for Smart Home using AES and RC4

Jasim M. Saadoon† and Imad J. Mohammed††

University of Baghdad, College of Science, Iraq

Summary

In the 21st century, people are more engrossed in how they can control their homes and secure them from intruders. Such realities have been made possible by the introduction of Short Message Services (SMS) which are used entirely to control different household equipment. These technologies are referred to as smart homes. Many people around the world use this technology because of their essential characteristics such as; automatic to carry tasks, they are safe, they are reliable and knowledgeable when it comes to the home environment that is fully supported by GSM or networks. However, one of the most significant problems experienced with this technology is its security. GSM technology is widely used over the years, but a lot of complaints have been raised due to its lack of confidentiality. Current studies have come up with a hybrid system that is composed of secured SMS and implementation of the smart home system.

Based on the efforts to secure Smart homes, this study proposes two models that should be used in smart homes. The two models of the proposed framework include; smart home (Arduino and ARSMS protocol using C language) model and controls (Android and ARSMS protocol using Java language) model connected via GSM. ARSMS is a hybrid protocol recommended to secure the exchange SMS messages in which RC4 and AES are integrated for message confidentiality (SMS encryption). The Implementation results of this technique shows that the proposed protocol can provide secure communication in IoT environment regarding secrecy and randomness with some overhead in space and time. Furthermore, the execution result of this technology shows that it is hard to hack into these systems thereby making the information passed through these systems to be more secure than the traditional GSM.

Keywords:

AES, Arduino, GSM, RC4, Smart Home, SMS.

1. Introduction

In the recent past, there has been an increasing concerned among users of the smart home idea on its reliability and safety against theft of information. Such concerns are arising because of the importance of smart home to many people. Generally, the smart home is connected with numerous devices such as; home diversion comforts, security systems, lighting and access control systems. Furthermore, Smart home mechanization framework is joined into shrewd homes to give solace, accommodation, and security to homeowners [1],[2]. Moreover, smart homes system also has the ability to speak to and report the status of the associated devices in a natural, easy to understand interface enabling the client to communicate and control different devices with the touch of buttons. Some of the significant communication technologies utilized by the current smart home systems include Bluetooth, GSM, WiMAX, ZigBee and Wireless LAN (Wi-Fi) [2].

However, over the past years, the smart home system has seen a lot of development in most of its components [2]. For instances; a massive number of cell phones are at present utilized everywhere throughout the world to perform particular tasks that are aimed at securing the information shared between people and the devices [3],[4]. Such moves are made to energize the use of smart home ideas. More importantly, some of these innovations have made it possible for an individual to operate the appliances of home from the workplace or other outside areas.

With increasing need to bring comfort in the home, smart homes system intends to use specific apparatus proficiently and viable to ensure that such things are available to the clients. Therefore, this study proposed a smart home system that uses Arduino through SMS protocols. The most significant concern using Arduino is the constrained memory accessible. The main differences between a general-purpose computer and the Arduino microcontroller are the large memory available to the general computers. Arduino UNO has just 32 KB of flash memory, 2 KB of SRAM and 1 KB of EEPROM. This means that the Arduino has 100,000 times less memory as compared to the low-end personal computer [4]. Such limitations were taken into consideration when choosing the encryption algorithms.

The other sections of the paper will be as follows; section 2 will present background about the applied algorithms and hardware used in the project while section 3 will introduce the related works. More so, section 4 will present the proposed protocol regarding the smart home structure, materials, and design. Section 5 will demonstrate the implementation of the proposed protocol while section 6 will be an in-depth discussion of the results found by this study. Lastly, the conclusion part will summarize the paper findings.

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

2. Background

2.1 RC4 Strem Cipher Encryption Algorithm

RCA Stream Cipher Encryption Algorithm is one of the mostly used stream cipher encryption algorithms in the world today. This algorithm is designed by Ron River. Furthermore, RC4 algorithm always employs the variable length key of between one to two hundred fifty-six bytes to introduce a table encoding bytes. RC4 works in two stages. The first stage is the main setup stage and the second phase is the encryption stage [5],[6]. The main setup stage is the most challenging phase where the encryption key variable is generated from the base key which consists of two components keys scheduling algorithm and pseudo-random number generation algorithm. Once these keys are produced, they are moved to the second stage where they are used to create the encrypted message by XORing on the key with the plaintext. The flowchart for RC4 algorithm Key generation components is shown in figure 1.

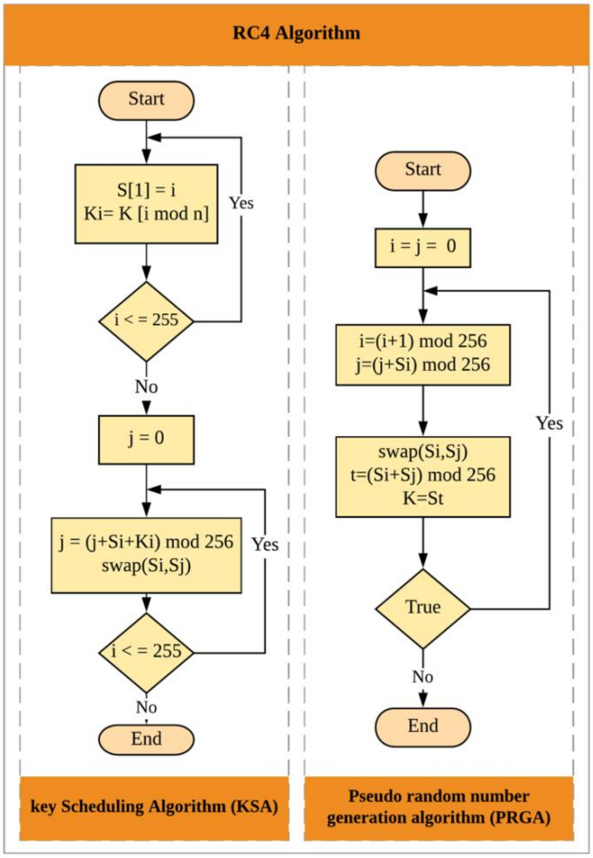


Fig. 1 Flowchart of RC4 Algorithm

2.2 Advanced Encryption Standard (AES)

The AES is a crucial asymmetric block cipher used by many organizations such as government, health organizations among other institutions to protect classified and sensitive information. The National Institute of Standards (NIST) affirmed AES as Federal Records Processing standards which suggests to use it in all sensitive characterized facts [5]. The AES represents the maximum popular symmetric encryption algorithm. The EAS is one of the famous and extensively used algorithms worldwide [6]. The wide variety of rounds in AES is variable and is predicated upon the duration of the critical size. AES uses 10 of 128-bit keys together with 12 rounds of 192-bit keys as well as 14 rounds of 256-bit keys. Each of these rounds makes use of a change 128-bit round key, which is computed from the authentic AES key [5],[6].

3. Related Work

Different scholars have carried different studies using SMS applications to monitor and help people to control various home appliances when they are at work or away from home. One of such researchers Abbas M.AL Bakry and RajaaD. Resan [7]. These scholars developed a smart door lock that could be controlled entirely by a smartphone. The unique data contained in the smartphone was encrypted and decrypted using RC4 Cipher stream. The primary purpose of their study was to improve security based on individual smartphone data by developing a smart door lock using RC4. This door was able to communicate with the remote or the smartphone through Bluetooth [7] Moreover, these scholars used the phone number as the encryption key. However, one of the main limitations of this study was using the phone number as a public key of RC4 without modification. As a result, making it easier for people who are aware of the cell phone number to access the house easily.

Furthermore, a home automation system supported entirely by a Wi-Fi technology other have been developed in the past. This system composed of web server which was the central system core that controlled everything, hard interface module consisting of Arduino PCB which provided supplied sensors to the actuators of the home automation system. This system was used widely for scalability and flexibility than the commercially home automation system. When the users are connected to the internet, they will be able to access the served web through the internet web browser [8].

Besides, there are applications that have been designed based on the existing Arduino and Android system [9],[10],[11]. For instances, there is an interface card developed to enable communication to take place between the server, the remote users, Arduino and the various home gadgets [9]. The application is lodged on both the web server, the Android smartphone as well as the Arduino. The android on the smart card is tasked with the role of giving commands to the Arduino. In conjunctions with these properties, an interface card is used to relay update signal between the Arduino and the actuator sensors.

4. The Propsed Protocol (ARSMS)

ARSMS, an Arduino secured SMS protocol is designed to provide the connection between the smart home and Android mobile users. The following sections demonstrate the structure and materials of ARSMS-based on smart home;

4.1 Smart Home Structure and Scenario

Figure 2 shows the Smart Home Structure Integrated to ARSMS Protocol. The central processor of the smart home network is Arduino UNO, connected to a GSM SIM900a Module. This Microcontroller connects domestic appliances and runs a home automation software that supervises domestic gadgets. The relay module joined to the I/O Arduino is tasked with the role of relaying the information to Arduino to either turn on or off the home electronic gadget. The primary purpose of this technology is to allow users to monitor the status of home network connected to the home appliances using Android mobile phone. Homeowners communicate with Arduino using SMS protocol, and Arduino translates the received SMS control messages to actions on smart home devices. Besides, when an abnormal state occurs, the Arduino can notify the home users by sending SMS.

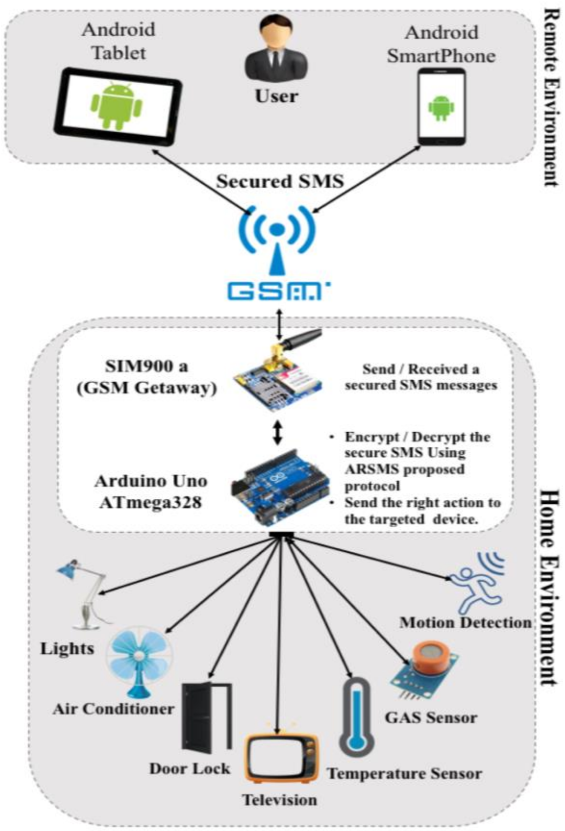


Fig. 2 Smart Home Structure Integrated to ARSMS Protocol

4.2 Materials

This section provides an overview of the tested smart home components and specifications of each.

1. Arduino UNO

Arduino UNO (Figure 3) is a popular device uses ATmega328 Microprocessor. The Arduino UNO characteristics are listed in Table 1.

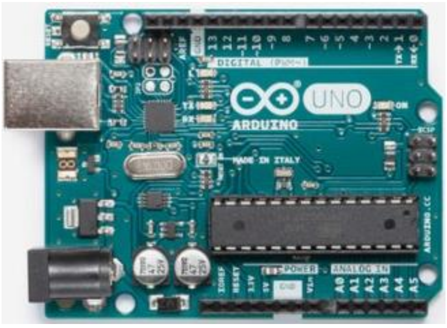


Fig. 3 Arduino Uno

Table 1: Arduino UNO Characteristics [12]

Microcontroller	ATmega328P
Operating Voltage	5 V
Input Voltages	7-12 V
Digital I/O Pin	14 Pins
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

2. SIM900A GPRS/GSM Module

SIM900A module (Figure 3) is widely used in GSM protocol communication. This module is designed with control sparing procedure, so the present utilization is as low as 1.5 mA in sleep mode. AT commands are evolved for customers to use the protocol without difficulty and it is integrated with the TCP/IP protocol for data transfer applications. The module contains RS232-primarily based serial port for connection, a SIM card holder, an antenna for sending and receiving indicators to the SIM, and LED as standing for an incoming call, power, and sign [13].

Fig. 4 SIM900a GPRS/GSM module

3. Relay Module

Relay Module is practically applicable to the main switch relay in a project that requires eight channels with electronic circuit microcontroller. The work of this module is to either turn on or off other electronic gadgets that are controlled by DC high-voltage or 240 VAC electrical devices [13],[14]. RM54OC Relay Module shown in Figure 5.

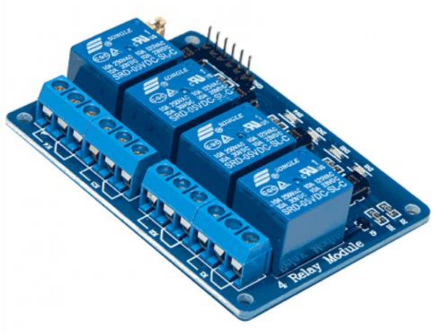


Fig. 5 RM54OC Relay Module

4. Sensors

Table 2 shows sensors that used to test the proposed prototype.

Table 2I: Smart Home Sensors [14]

No.	Sensor Name	Descriptions
1	DHT11	Low-cost digital temperature and humidity sensor.
2	MQ3	Gas Sensor module: useful for gas leakage detecting.
3	HC-SR04	Ultrasonic ranging module: provides non contact measurement function.

4.3 Hardware Design

The system consists of an Arduino microcontroller as a controller, SIM900a as SMS gateway, relays as outputs and three sensors for gas detection, temperature and motion detection. Figure 6 shows the circuit design of the smart home.

5. Implementation of the Proposed Protocol

The proposed protocol (ARSMS) is designed as the hybrid symmetric encryption algorithm (AES-RC4) supported by timestamp to achieve crucial dynamic generation across GSM. The output stream of RC4 is fed into AES which in turn encrypt the smart home control command as plaintext to ciphertext.

The received message is divided into two packets; the first packet is the encrypted IoT commands which will be decrypted by AES, and the second packet is the attached timestamp sent by the sender to generate the dynamic key by using RC4). The receiver first encrypts the timestamp by RC4 using a shared key, then AES will apply to decrypt the encrypted SMS. Figure 7 shows the working flow of ARSMS in both sender and receiver sides (Android/Adriano).

6. Results and Discussion

Space and time limitations of Arduino are considered during the validity of our proposed prototype. The performance evaluation of ARSMS is classified into three factors; time complexity security consideration and space complexity and as follows:

6.1 Space complexity

It is noticed from the space complexity evaluation that the proposed protocol needs 76% of Flash Memory, 90% of EEPROM and 93% of SRAM). Making it applicable without problems using Arduino UNO. However, the proposed protocol can be more practical if Arduino Mega is used since it supports (128 KB of Flash, 4 KB of

EEPROM and 8 KB of SRAM). Table 3 and Figure 8 shows the summary of space (available and used) in Arduino Uno (ATmega328) When implementing the three algorithms of the proposed framework; native RC4 algorithm, native AES and ARSMS proposed protocol.

Table 3: Amount of Memory available in Arduino when using native RC4, native AES and ARSMS proposed protocol in KB

Type of Memories	Total space in Arduino Uno	Required Memory in KB For			
		RC4	AES	Sketch of Sensors	ARSMS Protocol
Flash	32	4.44 (13.88%)	7.384 (23.08%)	12.62 (39.44%)	24.444 (76.39%)
EEPROM	1	0.2 (20.00%)	0.255 (2a5.50%)	0.452 (45.20%)	0.907 (90.70%)
SRAM	2	0.4 (20.00%)	0.575 (28.75%)	0.9 (45.00%)	1.875 (93.75%)

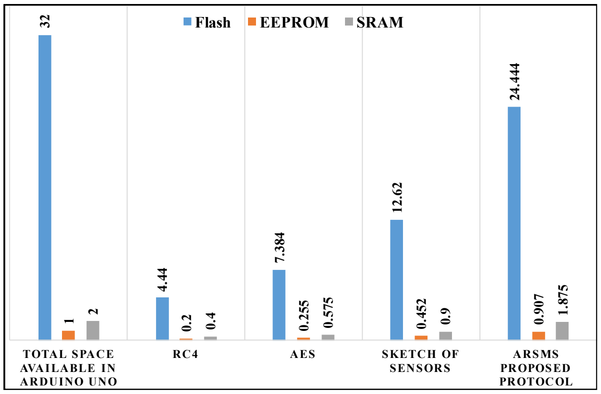


Fig. 5 Amount of Memory available in Arduino when using native RC4, native AES and ARSMS proposed protocol in KB

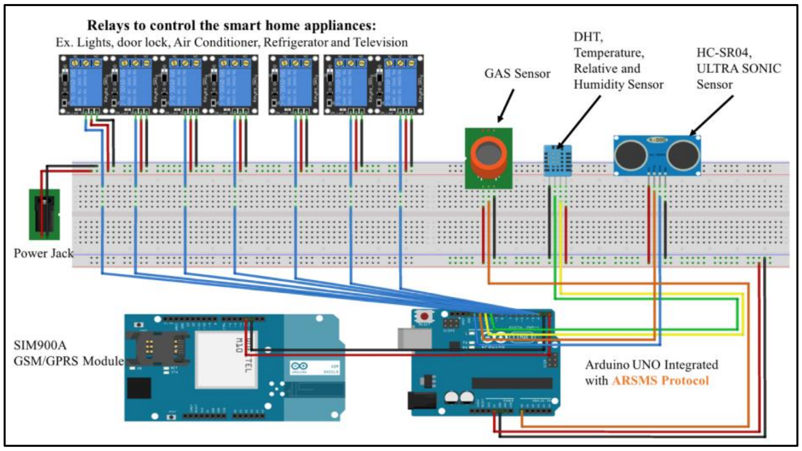
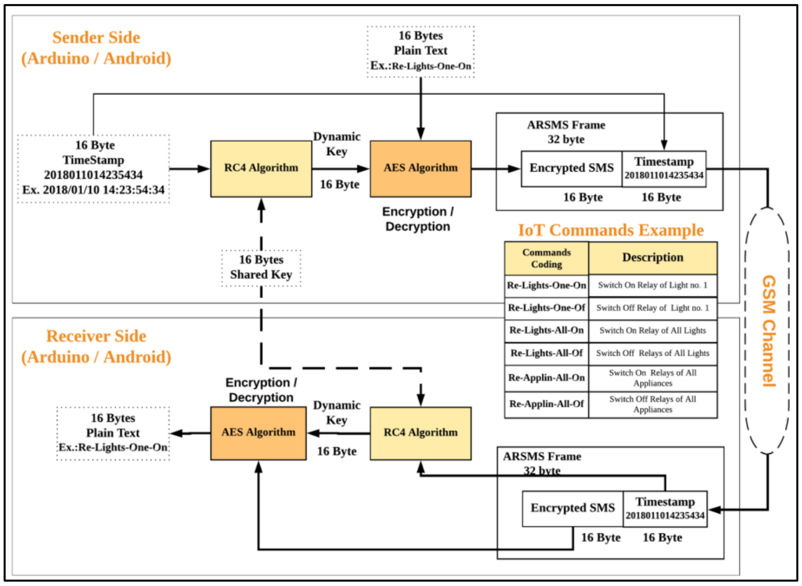

Relays to control the smart home appliances:
Ex. Lights, door lock, Air Conditioner, Refrigerator and Television
GAS Sensor
DHT, Temperature, Relative and Humidity Sensor
HC-SR04, ULTRA SONIC Sensor
Power Jack
SIM900A GSM/GPRS Module
Arduino UNO Integrated with ARSMS Protocol

Fig. 6 ARSMS Protocol Integrated to Smart Home Circuit Design


Sender Side (Arduino / Android)
16 Bytes Plain Text Ex.:Re-Lights-One-On
16 Byte TimeStamp 2018011014235434 Ex. 2018/01/10 14:23:54:34
RC4 Algorithm
Dynamic Key
16 Byte
AES Algorithm
Encryption / Decryption
ARSMS Frame 32 byte
Encrypted SMS
Timestamp 2018011014235434
16 Byte
16 Byte
16 Bytes Shared Key
GSM Channel
Receiver Side (Arduino / Android)
Encryption / Decryption
16 Bytes Plain Text Ex.:Re-Lights-One-On
AES Algorithm
Dynamic Key
16 Byte
RC4 Algorithm
ARSMS Frame 32 byte
Encrypted SMS
Timestamp 2018011014235434
16 Byte
16 Byte

IoT Commands Example

Commands Coding	Description
Re-Lights-One-On	Switch On Relay of Light no. 1
Re-Lights-One-Of	Switch Off Relay of Light no. 1
Re-Lights-All-On	Switch On Relay of All Lights
Re-Lights-All-Of	Switch Off Relays of All Lights
Re-Applin-All-On	Switch On Relays of All Appliances
Re-Applin-All-Of	Switch Off Relays of All Appliances

Fig. 7 Workflow of the proposed protocol (ARSMS) Integrated to Arduino/Android via GSM

6.2 Security Evaluation

In this subsection, an analysis of the proposed protocol (ARSMS) compares to native RC4 and AES based on two main parameters randomness tests and cipher secrecy is introduced. 100 random plaintexts and 100 keys have been generated randomly for each input. Cipher secrecy measured by entropy which gave an indicator for the amount of information that exists in a random string. Randomness tests are evaluated by Frequency test, Block test and Runs test to find if the distribution of data is random or not.

The security results show that the cipher of the proposed protocol is more secure and random than RC4 and AES when implementing each alone. ARSMS wins by 10 out of 12 tests (using three cases: plain 128 * key 128, plain 128 * key 192, and plain 128 * key 256). The average gain using ARSMS compared to the implementation of native AES is 48.56% using Frequency test, 11.88% using Block test, -50.41% using Run test and 17.18% using cipher secrecy test. RC4 was already known as fast with limited security power compared to AES and consequently not comparable to ARSMS (as shown in Table 4).

Table 4: Randomness and Secrecy Tests

Sizes	Tests		Native RC4	Native AES	ARSMS Proposed Protocol
Plain 128 * Key 128	Randomness	Freq.	0.3645	0.2509	0.5029
		Block	0.096	0.1468	0.2447
		Run	0.0977	0.0488	0.0222
	Secrecy		0.232	0.2407	0.4609
Plain 128 * Key 192	Randomness	Freq.	0.4267	0.2988	0.5125
		Block	0.0941	0.0746	0.1125
		Run	0.0335	0.0531	0.0669
	Secrecy		0.8875	0.8542	0.9836
Plain 128 * Key 256	Randomness	Freq.	0.4690	0.2164	0.4741
		Block	0.9028	0.8906	0.9047
		Run	0.2271	0.6452	0.4043
	Secrecy		0.6518	0.7006	0.7235
Win out of Total Tests			1/12	1/12	10/12
Average	Randomness	Freq.	0.4201	0.2554	0.4965
		Block	0.3643	0.3707	0.4207
		Run	0.1194	0.2490	0.1645
	Secrecy		0.5904	0.5985	0.7226
First Highest Value					
Second Highest Value					

6.3 Time Complexity

ARSMS takes more execution time to encrypt/decrypt messages compared to native AES. Regarding time overhead, ARSMS implementation gains 0.91%, 0.72% and 0.80% more than native AES in average when it uses key sizes (128, 192, and 256) bits respectively. Table 6 shows the average encryption time for RC4, AES and ARSMS. All tests implemented on Arduino UNO with its specification and limitations as mentioned in Table 1.

Furthermore, the summary results (Table 6) show that the developed ARSMS protocol wins concerning secrecy and randomness tests except for the run test, and there is a very little processing overhead less than 1% of average encryption time.

Table 5: Average Encryption Time in Millisecond

Plain Size	Key Size	Average Encryption Time			Overhead between native AES and ARSMS Protocol
		RC	AES	ARSMS Proposed Protocol	
128	128	5.109	16.968	17.01	0.27%
	192	5.092	18.327	18.38	0.30%
	256	5.249	24.844	24.87	0.12%
256	128	5.12	17.951	18.04	0.50%
	192	5.13	17.058	17.18	0.73%
	256	5.179	21.745	21.91	0.75%
512	128	5.097	17.247	17.59	1.95%
	192	5.105	17.033	17.23	1.15%
	256	5.131	22.084	22.44	1.60%
Average	128	5.108	17.39	17.55	0.91%
	192	5.109	17.47	17.6	0.72%
	256	5.186	22.89	23.08	0.80%

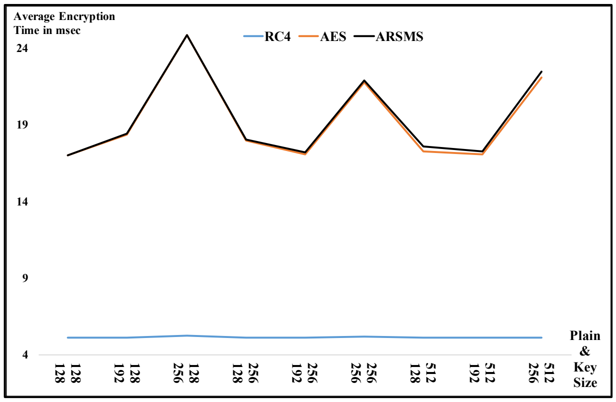


Fig. 6 Average Encryption Time in Millisecond

Table 6: Results Summary

1. Randomness and Secrecy			
Test	Native AES	ARSMS Proposed	Gain
Freq.	0.2554	0.4965	48.56%
Block	0.3707	0.4207	11.88%
Run	0.249	0.1645	-51.41%
Secrecy	0.5985	0.7226	17.18%
2. Space Complexity			
Type of Memory	Native AES	ARSMS Proposed	Total Space Used
Flash	7.384	24.444	76%
EEPROM	0.255	0.907	91%
SRAM	0.575	1.875	94%
3. Average Encryption Time			
Key Size	Native AES	ARSMS Proposed	Time Overhead
128	17.39	17.55	0.91%
192	17.47	17.6	0.72%
256	22.89	23.08	0.80%

7. Conclusion

From this study, a unique mechanism for securing SMS messages have been demonstrated through developing a new hybrid protocol called (ARSMS). It uses symmetric algorithms for encryption/decryption. Taken into consideration that SMS is for exchanging shorter information, the SMS protocol does not provide any security. The proposed protocol ARSMS uses RC4 for crucial dynamic generation using a timestamp, and AES for plaintext encryption on Arduino UNO to control smart home devices remotely. The performance evaluation shows that ARSMS outperforms native RC4 and competitive to the native AES regarding secrecy and randomness, while there is unnoticed overhead in encryption time.

References

[1] Ahmad, Arbab Waheed, Naeem Jan, Saeed Iqbal, and Chankil Lee. "Implementation of ZigBee-GSM based home security monitoring and remote control system." In Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on, pp. 1-4. IEEE, 2011.

[2] Begum, Tahmina, Md Shazzat Hossain, Md Bashir Uddin, and Md Shaheen Hasan Chowdhury. "Design and development of activation and monitoring of home automation system via SMS through microcontroller." In Computers and Devices for Communication, 2009. CODEC 2009. 4th International Conference on, pp. 1-3. IEEE, 2009.

[3] Elkamchouchi, H., and Ahmed ElShafee. "Design and prototype implementation of SMS based home automation system." In Electronics Design, Systems and Applications (ICEDSA), 2012 IEEE International Conference on, pp. 162-167. IEEE, 2012.

[4] Padmajothi, V., Ankit Rai, M. Dastagiri Reddy, and N. Renu Kumar. "Cost Effective Home Energy Monitoring System." International Innovative Research Journal of Engineering and Technology 2 (2017): 113-116.

[5] Paar, Christof, and Jan Pelzl. Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media, 2009.

[6] Elminaam, Diaa Salama Abd, Hatem Mohamed Abdual-Kader, and Mohiy Mohamed Hadhoud. "Evaluating the performance of symmetric encryption algorithms." IJ Network Security 10, no. 3 (2010): 216-222.

[7] Bakry, Abbas M. Al, and Rajaa D. Resan. "Smart Phone-Arduino based of Smart Door Lock/unlock using RC4 stream Cipher Implemented in Smart Home." International Journal of Advanced Computer Technology, 5 (2016).

[8] Gunge, Vaishnavi S., and Pratibha S. Yalagi. "Smart Home Automation: A Literature Review." International Journal of Computer Applicatios (2016): 6-10.

[9] Baraka, Kim, Marc Ghobril, Sami Malek, Rouwaida Kanj, and Ayman Kayssi. "Low cost arduino/android-based energy-efficient home automation system with smart task scheduling." In Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on, pp. 296-301. IEEE, 2013.

[10] Javale, Deepali, Mohd Mohsin, Shreerang Nandanwar, and Mayur Shingate. "Home automation and security system using Android ADK." International journal of electronics communication and computer technology (IJECCT) 3, no. 2 (2013): 382-385.

[11] Khan, Muhammad Waseem. "SMS security in Mobile Devices: a survey." International Journal of Advanced Networking and Applications 5, no. 2 (2013): 1873.

[12] Badamasi, Yusuf Abdullahi. "The working principle of an Arduino." In Electronics, computer and computation (icecco), 2014 11th international conference on, pp. 1-4. IEEE, 2014.

[13] ZHAI, Shun, Wei-hong WANG, Kan ZHANG, and Peng LI. "IOT SMS alarm system based on SIM900A [J]." Modern Electronics Technique 5 (2012): 025.

[14] Doukas, Charalampos. Building Internet of Things with the ARDUINO. CreateSpace Independent Publishing Platform, 2012.

Jasim M. Saadoon received his B.Sc. degree in Computer Science from Shit Al-Arab University, Basra, Iraq, in 2012 and he received a higher diploma degree in Computer science from University of Baghdad, Baghdad, Iraq in 2014. He is currently working toward the MSc degree in Computer science with University of Baghdad. His research interest includes Computer Security, IoT, Mobile Development and Computer Networks.

Dr. Imad J. Mohammed born in 1967 in Baghdad-Iraq, is a lecturer in the Dept. of Computer Sciences , college of science, university of Baghdad, Baghdad, Iraq since 2012. Manage of (studies, planning and follow-up) section of college of science. For two years (2016-2017). Also, lecturing of networking course to 4th year degree and Master degree students.

He is graduated with a PhD in Computer Science from USM- Malaysia in 2011 with Thesis titled “AN OPTIMIZED FRAMEWORK FOR HEADER SUPPRESSION OF REAL TIME IPv6 TRAFFIC IN MULTIPROTOCOL LABEL SWITCHING (MPLS) NETWORKS”. Graduated with MSc in computer science from University of Technology – Iraq in 1996. B.Sc in computer science from University of Baghdad, Iraq in 1989. His Interest: Computer networks domain: IPv6, QoS, routing protocols, optimization, WSN, multimedia over IP, IoT and Computer Security.