Educational Administration: Theory and Practice
2024,30 (1), 4488 - 4492
ISSN: 2148-2403
https://kuey.net/

Research Article

Ubiquitous Hand-Held Device's Sigma LFSR Driven Security Model

Atanu Datta[1*], Somsubhra Gupta[2], Subhranil Som[3]

[1*] Swami Vivekanand University, Barrackpore -700121, WB, INDIA (e-mail: atanudat@gmail.com).
[2] Swami Vivekanand University, Barrackpore -700121, WB, INDIA (corresponding author phone: 9836545777; e-mail: gsomsubhra@gmail.com).
[3] Bhairab Ganguly College, West Bengal State University, Kolkata, WB, INDIA (e-mail: subhranil.som@gmail.com).

Citation: Atanu Datta et al (2024) Ubiquitous Hand-Held Devices Sigma LFSR Driven Security Model , *Educational Administration: Theory and Practice, 30(1), 4488 - 4492*
Doi: 10.53555/kuey.v30i1.7969

ARTICLE INFO

ABSTRACT

Mobile data security is crucial in modern communication, especially in financial transactions. This paper introduces a data security mechanism for the banking and financial sectors based on the AES algorithm. Hackers often exploit various techniques to intercept, alter, or steal information during these transactions. The primary objective of this work is to identify effective encryption methods to secure data from such attacks during financial transfers. By integrating the AES algorithm into an Object-Oriented Model, with SIGMA LFSR in the pipeline, the proposed system enhances message confidentiality and aims to improve trust in mobile-based online banking, particularly on client-side applications for handheld devices. With mobile phones becoming an integral part of daily life, security concerns surrounding the storage of sensitive credentials on these devices have increased. This research focuses on applying cryptographic techniques to bolster the security of mobile devices.

Index Terms— Device Affordability, Digital Divide, Inclusive Education, Mental Health, NOODLE, Nutrition, OBE, Virtual Learning.

I. INTRODUCTION

The transfer of financial data, whether from banking, digital wallets, online shopping, or government tax payments, involves utilizing information technology to exchange services and data between citizens, businesses, and other stakeholders within financial institutions. A critical challenge in developing applications for financial transactions is verifying the authenticity of electronic documents and packaging them efficiently. Additional unresolved concerns include ensuring the security, integrity, uniqueness, and traceability of these documents while preventing unauthorized duplication and tampering. India's National E-Transaction Plan [6] seeks to establish a framework for the sustainable growth of e-transactions both within and outside the country, with a particular focus on Indian financial institutions. Some key client-side applications in the financial sector include income tax submissions, banking services, provident fund status, passport and visa information, verification of voter ID/national citizen cards, trade license agreements, PAN, and Aadhar card verification, among others. Cybercriminals may alter critical information for their own benefit, making it crucial to use secure cryptographic methods to prevent unauthorized access when transmitting data over the internet. Cryptographic techniques are classified into three main categories: symmetric cryptosystems, asymmetric cryptosystems, and digital signatures. In symmetric cryptosystems, both the sender and receiver share the same key, ensuring privacy and confidentiality. Asymmetric cryptosystems, on the other hand, use distinct keys for encryption and decryption, primarily aiding in key exchange and authentication. Digital signatures, which provide irreversible encryption, ensure data integrity. Encryption scrambles data in a way that is dependent on the key size, offering a high level of protection. Various algorithms are employed, such as Triple DES (Data Encryption Standard), RSA, and AES (Advanced Encryption Standard). Symmetric cryptosystems typically use DES and AES, while RSA is commonly used in asymmetric cryptosystems.

Cryptography [1], the practice of encoding and decoding information using mathematical techniques, secures sensitive data, whether stored or transmitted over unsecured networks like the internet, ensuring only the intended recipient can access it. In mobile financial transactions, the goal is to facilitate smooth

communication among different parts of the transaction process, which is often achieved using an asymmetric cryptosystem.

II. GAP ANALYSIS

Pervasive and widely used computing devices are frequently subject to misuse. Users often perform activities unrelated to their primary tasks, which creates vulnerabilities, particularly in online payment gateways. Another critical concern is the digital divide, as merchants and those in rural areas face challenges in managing the associated costs. This makes securing their work and data more expensive. As a result, security has become a central issue in all areas during this transitional phase of adapting to these emerging ubiquitous devices, especially in comparison to older technologies. Hence, there is an urgent need for more robust and secure devices.

III. IMPORTANCE OF OBJECT ORIENTATION

Simplicity: Software objects represent real-world entities, which helps simplify and clarify the program's structure. A key component is that an object within the program generates the secret key.
Modularity: Each object operates independently, with its internal functions separated from other system components.
Modifiability: Making adjustments to data representation or procedures is easy in an object-oriented system. Changes within a class do not affect other parts of the program since access to the class is restricted to its methods. Frequent updates are needed due to evolving financial transaction policies from banks or the government, as depicted in Figure 1.
Extensibility: New features or changes to operational environments can be addressed by introducing new objects or modifying existing ones. Governments frequently implement new policies to foster societal progress.
Maintainability: Objects can be managed individually, making it easier to identify and fix issues.
Reusability: Objects can be reused in various programs.

Cryptography: Cryptography involves applying mathematical techniques to secure and decipher information. It ensures sensitive data can be stored securely or transmitted across insecure networks, such as the internet, while restricting access to authorized recipients. Cryptanalysis, on the other hand, aims to analyze and break secure communications. Classical cryptanalysis combines analytical reasoning, mathematical tools, pattern recognition, patience, and perseverance. Cryptology encompasses both cryptography and cryptanalysis.

Figure 1: Block Diagram

3.1 The Encryption Key and Its Expansion

When using a 128-bit key, it is arranged in a 4x4 byte matrix. The first word from the key occupies the first column of the matrix, and subsequent words fill the remaining columns. The four-column words are expanded into a schedule of 44 words. Each round utilizes four words from this key schedule. The diagram illustrates how the 128-bit key is arranged and expanded into the key schedule of 44 4-byte words.

The Overall Structure of AES

The overall architecture of AES encryption and decryption is depicted in Figure 1. The number of rounds shown in Figure 2 corresponds to a 128-bit encryption key (as previously mentioned, a 192-bit key requires 12 rounds, while a 256-bit key requires 14 rounds). Before any round-based encryption process begins, the input state array is XORed with the initial four words from the key schedule. A similar approach is applied during decryption, where the ciphertext state array is XORed with the final four words of the key schedule.
Each round of encryption consists of the following four steps:
1. Byte substitution

2. Row shifting
3. Column mixing
4. Adding the round key

In the last step, the result of the previous three steps is XORed with four words from the key schedule.

For decryption, each round includes following four steps:
1.Inverse row shifting
2. Inverse byte substitution
3. Adding the round key
4. Inverse column mixing

In the third step, the result of the previous two steps is XORed with four words from the key schedule.
The final encryption round omits the "Mix columns" step, while the last decryption round excludes the "Inverse mix columns" step.

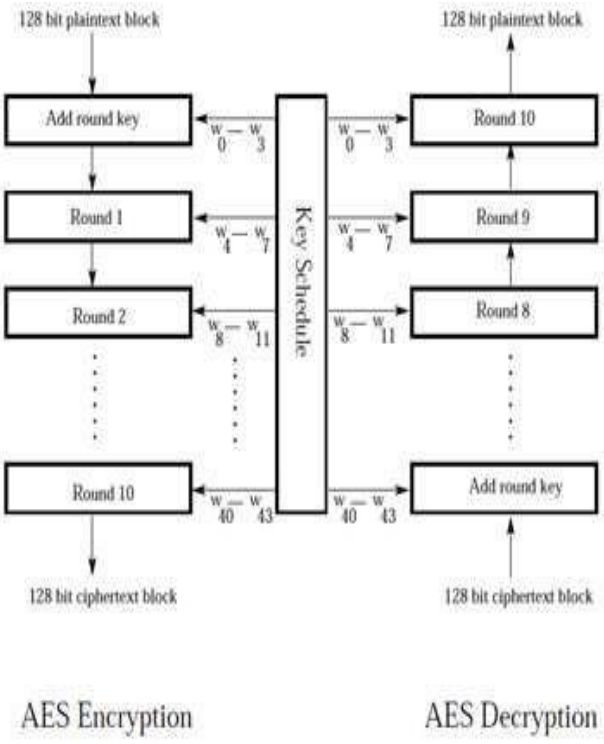

Figure 2: Rounds of AES

3.2The Four Steps in Each Round of Processing

STEP 1: Generate Sub Bytes for byte-by-byte substitution during the forward process.
STEP 2: Shift Rows for shifting the rows of the state array during the forward process. The corresponding transformation during decryption is called Inv Shift Rows for Inverse Shift-Row Transformation.
STEP 3: Mix Columns for mixing the bytes in each column separately during the forward process.
STEP 4: Add Round Key for adding the round key to the output of the previous step during the forward process.

This involves a byte-by-byte substitution, where the substitution byte for each input byte is found using the same lookup table.
The size of the lookup table is 16 × 16.

The Shift Rows Step

- This involves shifting the rows of the state array as follows:
  - The first row is not shifted.
  - The second row is circularly shifted one byte to the left.
  - The third row is circularly shifted two bytes to the left.
  - The fourth row is circularly shifted three bytes to the left.

The Mix Columns Step

- This step replaces each byte in a column with a function of all the bytes in that same column.
- Specifically, each byte in a column is replaced by:
    - Two times the value of that byte,
    - Plus three times the next byte,
    - Plus the following byte,
    - Plus the byte after that.
    - Here, 'next' refers to the byte in the row directly below, with this wrapping being circular within the same column.

Adding the Round Key

- The 128 bits of the state array are bitwise XORed with the 128 bits of the round key.
- The AES Key Expansion algorithm is used to derive the 128-bit round key from the original 128-bit encryption key.

4. SIGMA LFSR model

LFSRs are frequently used as pseudorandom pattern generators to generate a random number 1s and 0s. Each output of the LFSR is multiplexed with an ASIC input and, when the device is placed in the LFSR (test) mode, the random, high-toggle-rate patterns produced are extremely good for generating high-fault coverage.

Generating random numbers **on a deterministic machine like a computer is complicated — this is where** linear-feedback shift registers (LFSR) come in handy, and you can try them out with our **LFSR calculator**. With a little bit of math and some computer science knowledge, you will learn everything you need about this special type of register.

4.1. Pipelining SIGMA LFSR model

Next, control will be redirected from AES to a SIGMA-LFSR channel through a pipeline, which will block unauthorized infiltration or invasion. This ensures that the system's files and data are protected through the SIGMA-LFSR module with a proper authentication scheme. The aim is to provide seamless access for the authentic user while simultaneously creating a protective shield around the system's jobs, files, and data using the SIGMA-LFSR module to guard against infiltrators and intruders.

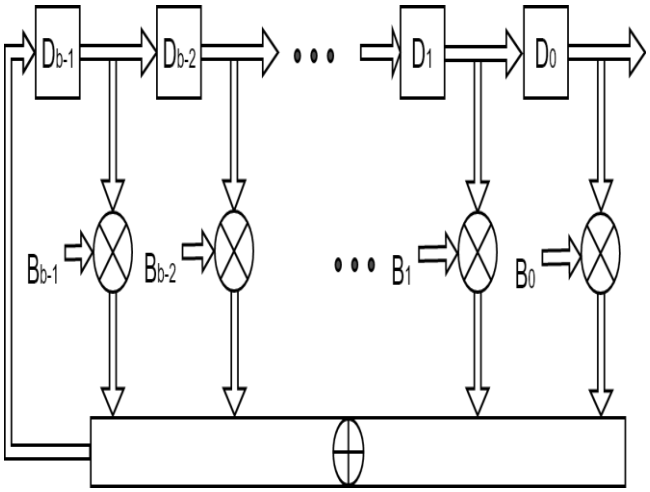

Figure 3: Block Diagram SIGMA-LFSR

IV. CONCLUSIONS

In this paper, we concentrate on modelling the AES secret key cryptographic algorithm using the Object-Oriented Programming (OOP) paradigm. The proposed Object-Oriented model leverages data hiding to protect the secret key. Code reuse is facilitated through the implementation of inheritance. Typically, all secret key cryptographic algorithms use a single secret key at both the sender and receiver ends, which can be easily defined in OOP. In our proposed Object-Oriented model of the AES algorithm, various objects of the sender class are defined such that the secret key of the sender class object remains hidden from third-party objects but is accessible to the receiver class objects. Another advantage of this model is that the roles of sender and receiver classes can be interchanged simply by declaring the objects interchangeably in the Driver Program main(). Object-oriented programming (OOP) offers numerous benefits to both software designers

and users. It addresses various challenges encountered in software development and enhances software quality. The adoption of object-oriented technology promises increased programmer efficiency, superior software quality, and reduced maintenance costs. In this framework, we propose a two-tier security approach incorporating the pipelining of the SIGMA-LFSR model. Through the use of inheritance, redundant code can be eliminated, and the functionality of existing classes such as Sender and Receiver can be extended. This enables the construction of programs using standardized modules, resulting in time savings during development and enhanced productivity. Object-oriented systems are highly scalable, allowing seamless transition from small-scale to large-scale systems. For instance, in our current object-oriented AES model, additional secret keys can be easily incorporated into the Sender class to enable communication with multiple recipients, eliminating the need to rewrite code from scratch.

REFERENCES

[1] Lukasz Lysik, Mobile Security: Threats and Best Practices, Wroclaw University of Economics, Komandorska 118/120, Wroclaw, Poland December 2020
[2] Radi Qayyum, Data Security in Mobile Cloud Computing: A State of the Art Review, Government College Women University Sialkot April, 2020
[3] Xiuqing Lu, Zhenkuan Pan, Hequn Xian, An efficient and secure data sharing scheme for mobile devices in cloud computing, October -2020
[4] Amita GOYAL CHIN, Ugochukwu ETUDO, Mark A. HARRIS, On Mobile Device Security Practices and Training Efficacy: An Empirical Study, May 2016
[5] Loreen M. Powell, Jessica Swartz, Michalina Hendon, Awareness of mobile device security and data privacy tools, Issues in Information Systems, Volume 22, Issue 1, 2021 pp. 1-9,
[6] Rashmi P. Sarode, Subhash Bhalla, Data Security in Mobile Cloud Computing, Jun 2019
[7] Md. Shoriful Islam, Systematic Literature Review: Security Challenges of Mobile Banking and Payments System, Vol. 7, No. 6 (2014), pp. 107- 116
[8] Athidass, G., and K. Alamelu. "Security Issues in Mobile Banking." Shanlax International Journal of Management, vol. 6, no. S1, 2018, pp. 6–10
[9] Trozze A, Kamps J, Akartuna EA, Hetzel FJ, Kleinberg B, Davies T, Johnson SD, Cryptocurrencies and future financial crime. Epub 2022 Jan 5.
[10] Alvarez F, Argente D, Van Patten D, Are cryptocurrencies currencies? Bitcoin as legal tender in El Salvador., Dec 22.
[11] Weinberg C.B., Otten C., Orbach B., McKenzie J., Gil R., Chisholm D.C., Basuroy S. Technological change and managerial challenges in the movie theater industry. J. Cult. Econ. 2021
[12] Mamatzhonovich O.D., Khamidovich O.M., Esonali o'g'li M.Y. Digital Economy: Essence, Features and Stages of Development. Acad. Globe Inderscience Res. 2022 ;3:355–359.
[13] Information Technology—Security Techniques—Guidelines for ybersecurity. ISO; Geneva, Switzerland: 2012. 15. Qayyum Rida, Ejaz Hina, Data Security in Mobile Cloud Computing: A State of the Art Review, International Journal of Modern Education and Computer Science, April 2020