A Novel Technique to Secure Inter-Process Communication

A.E.M. Eljialy[1] and Sultan Ahmad[2*]
[1]Department of Information Systems, College of Computer Engineering and Sciences,
Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia
[2]Department of Computer Science, College of Computer Engineering and Sciences,
Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia
[*]Corresponding Author: Sultan Ahmad(*s.alisher@psau.edu.sa*)

Summary

Interprocess communication (IPC) is the interface for communication between different programs. Communication occurs through message transfer. We gave a model for security of these messages. We applied XOR cipher and AES cipher on these methods and checked their credibility. Same size for message and key was used to make it difficult finding key that is traveling with the encrypted message. Moreover, we proposed a method to deal with DDoS attacks and save the system from going offline. At the end we tested our system on basis of speed, security and integration with the system. As a result, we got state-of-the-art system on basis of security.

Keywords:
IPC, XOR cipher, AES cipher, message security, encryption.

1. INTRODUCTION

Interprocess communication (IPC) is an interface that helps different programs communicate with each other. It helps the system to handle more user requests at same time. Programs need to communicate with each other because a user request may involve different programs. To communicate with different programs there is massage transfer involves. Our research deals with security of these messages. Speed and security both are the concerns when we deal with messages encryption as it takes time. For security different encryption techniques are used in the past. Encryption is a process of encoding the plain text on basis of certain rules. These certain rules are used both for encryption and decryption of text in most of the cases. Encrypted text can only be decoded by these certain rules. Different encryption techniques are used on basis of need i.e if security is main concern or not or if speed is main concern. Security and speed has a tradeoff with each other. Mostly highly secured encryption techniques are computationally more costly and hence slow. As one of the technique used in paper (XOR) is computationally cheap but is fast but we proposed a method to increase it's security.

This paper shows different methods for message encryption in interprocess communication. We use XOR and AES cipher methods to encrypt the messages.

We showed that these methods can be very useful in message encryption. At the end we run different test using both of these encryption methods i.e speed and security. Moreover, we also checked our method's integration with hardware. We also proposed a method to save IPC from DDoS attacks. While using XOR and AES we also kept in mind their disadvantages as if any letter of XOR encryption is decrypted whole message can be decrypted. We also deal with these cons and as a result we got state-of-the-art system with no breaches during testing.

2. RELATED WORKS

Researchers have work on reliable Inter-process communication and they focused in enhancement of middleware software. Researchers proposes the design and implementation of composite software architecture which realize kernel [1] high-performance and reliable mobile distributed IPC mechanism. Nevertheless, software architecture has been measured in a geographically distributed system Cloud The purpose of this study is to give standardization model of Inter-process [2] communication that increases manufacturing organization operational performance. In this respect, this study has proposed two model which as follows: (i) Holistic model (ii)Quantifying model. On other hand, BRIAN N. BERSHAD[3] studies has focused on share memory multiprocessor. This research proposes a solution to the problem of share memory multiprocessor, through move the communication out of the kernel and supporting them at the user level. This study has described an approach to encounter the kernel [4] operations via transferring traditional operating system out of the kernel, also it divides the responsibilities for both kernel and share memory. This research has focuses on a hypnosis that the internet should be based on inter-process communication that would be achieved only through a protocol that gave a communication between protocols and managed the distributed intercrosses communication. The study has specified the complete operation of distributed inter-process communication' layer. This paper describes the [5] messages synchronization and performance improvement that support message passing protocols. This mechanism

Manuscript received September 5, 2022
Manuscript revised September 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.9.100

provides a transfer of messages perfectly without intermediate buffer. An experimental study has provided an algorithm that adapts the communication between different protocols. Whereas, there are variety of protocols need to understand each other, in this regards, this will represent a conflict. Meanwhile, this research has proceed an experimental study to eradicate this problem by genetically engineer classifier system that different protocol layers that resolves the incompatibility conflict. This paper has discussed several classes of inter-process communication, and it has also [6][7] stated an algorithm which is scholar has claimed robustness, based on a formalism uses of classes of interprocess communication. Moreover, it wasn't like more conventional formalism. Nonetheless, a research which had been using Linda model of coordination and communication. [8][9] It discusses the features of Linda model, whereas, scholars have significantly highlighted Linda's implementation problems . Further on, a research discusses using of inter-process communication in network system, although, this approach hasn't been implemented by the scholars of [10] manuscript. However, scholars have noted in this research work, no need for monitoring message. Moreover, scholar has claimed

The receiving process has the privilege stop the flow of messages. A technical report has discussed light Wight communication, the main objective of this technical report is to develop the latency of message passing system. It has been targeted data level and targeting the logging and flashback tools. Their approach has been claimed in manuscript that eliminated the use hub and substitute it with message passing system. Furthermore, this research [11] has been based on a technical implementation of inter-process communication for robots and performance and evaluation have been carried out. In 2002, a research has generated any mechanism that called PSYNC. This mechanism has been [12] used to order messages and it has stated an efficient implementation on unreliable network. This ordering of messages has been used to emphasize that the conversation is highly significant for messages ordering. The outcome of this research, it has distinguished between policy and mechanism, also it has carried out how conversation in the communication system can be done in less cost wise. PYSNC is in a low level of communication abstraction in a distributed environment. In case of android system, a research a manuscript has combined both microkernel and regular kernel (personal computer). In this respect, the flexibility and reliability has been addressed. Further on, it discusses a security mechanism and it does elaborates the attacker how approach the kernel components binder. Meanwhile, this attack relies on how attacker can get across [13] kernel binder and passes data through kernel binder. But it shows how it is so easy to get through kernel binder. Basically, this manuscript has stated that the defending methods to this type of attacks are so difficult and represent real challenge to be eradicated. Whereas, a lot of data have been sent via inter-process communication you can verified weather are normal and falls into suspicious activities. With the rapid development of android system, a manuscript discusses the different attacking methodologies under android system platform. Moreover, this research has elaborate the binder components of android system and it has elaborated [14] as well. Nonetheless, it has described the exchange of messages in android platform, which proposes different hacking techniques in a level of the kernel. In this regards, the research reach beyond that, whereas, it has been proven that data have been extracted from any process and particularly system calls. Within the need to optimize and run some analysis on the performance of interprocess communication, a research has been evaluating the performance and optimization of inter-process communication. Meanwhile, it compares [15] the binder with traditional inter-process communication. This research yields a result that stated binder is much efficient in case of small data transmission. Whereas, their refutation was based on, whenever there are much more concurrent processes the efficiency of binder goes down. Conference' paper has proposes new technique which is called "prison". This technique has been made to solve the injection of process or data directed to another process. This research emphasizes the continuous communication between processes that using interprocess communication technique, it makes the system vulnerable to malware containment. Also it has stated that malware uses the trusted processes therefore, malware would travel and made a malicious actions. In this basis, this paper [16] has introduced PRISON as a technique that monitors the processes interactions and prevent Malware. Scholars has advised that PRISON would be better used to monitor an online system for tracking the processes that contain a malicious processes interactions. A model has been proposes by Xiao[17].

Peng which used V inter-process communication programming interface. This paper has worked on the enhancement of distributed environment functionality. Thus, they implement semaphore interface [18] technique under Linux operating system kernel to decrease processes congestion. The scholar of this paper has praised their model compared to similar systems. Scholars has emphasized the fast calling of application compare to other system who have use the traditional techniques of inter-process communication. Broker is a programming language that has been used by a scholar in their empirical research, in order to develop of a complex operations in robotics system. This thought behind representing a robotic system contains many of processes, where information needed to be exchange between these processes. In this context, this research [19] has carried out to describe the use of inter-process communication, however, this processes exchange their information in a robotics system.

3. MOTIVATION

IPC (Inter-Process Communication) with message queues allows for the use of unique keys to identify processes. This provides for access-level security. However, the use of keys poses a problem. If an offending process tries several keys, it will eventually read a message in the message queue and the offending process itself is able to check automatically when it succeeds in reading a message from the queue.

A solution is to apply a cypher upon the message that will be sent. The sending process and the receiving process must agree as to the ciphering process and cypher keys[20]. There is no straightforward automatic way for the offending process to do the confirmation check for breaking the message, even if the cypher that was tried at a given moment is the correct one. There are two ways to store cyphers: text and hard-code in the executable itself [21]. Neither approach is bullet-proof. For text-based storage, to the process is granted a level of privilege as to read the cypher text. This means that the user that is executing it has also access to the cypher text. Hardcoded into the executable has the distinct advantage that the user should have execute permissions but does not need to have read permissions. On the other side, for hard-coded cyphers, once the cypher is discovered, the executables for the send and receive processes will need a rebuild. An offending process may send DDoS (denial of service) attacks. A control process can guard against it checking if the number of messages in the message queue is higher than a given threshold and destroying the message queue if needed.

4. METHODOLOGY

For ciphering the text two types of methods were purposed,XOR cipher

1. AES cipher

4.1 XOR cipher

It simply works on bitwise XOR operation on text. Operation that it uses is:

$$A^{\mathrm{L}}0 = A$$

$$A^{\mathrm{L}}A = 0$$

$$A^{\mathrm{L}}(B^{\mathrm{L}}C) = (A^{\mathrm{L}}B)^{\mathrm{L}}C$$

$$(B^{\mathrm{L}}A)^{\mathrm{L}}A = B^{\mathrm{L}}0 = B$$

A simple XOR table is as follow:

Table 1: XOR table

Inputs				Outputs		
	0	0			0	
	0	1			1	
	1	0			1	
	1	1			0	

For encryption process each letter of word is converted into 8 bit binary and a same 8 bit key is used to encrypt each letter of word.

For Example, if we want to encrypt word “xor”, it’s binary is “01111000 01101111 01110010” and we use same key for each letter “11110011”. The results will be as follow:

01111000 01101111 01110010

$^{\mathrm{L}}$11110011 11110011 11110011

= 10001011 10011100 10000001

for decryption process applying XOR operation between same key and encrypted message gives :

10001011 10011100 10000001

$^{\mathrm{L}}$11110011 11110011 11110011

= 01111000 01101111 01110010

4.2. AES cipher

Advanced Encryption Standard (AES) is advanced form of Data Encryption Standard (DES). AES is stronger and faster than DES. Main problem of DES was that it has small key size (56 bits) in double DES key size increasesbut not to that much extent. In AES key sizes also vary between 128 bits and 256 bits. These are some basic properties of AES:

1. Symmetric Key Block Cipher
2. 128 bit data
3. Stronger and Faster than DES

Symmetric key block cipher means same key is used for encryption and decryption which is same as DES. Size of key depends on number of rounds. Number of rounds means how many times you want to encrypt your data. Given table explains the relation:

Table 2: Relation between no. of rounds and size of key used

No. of rounds	Key used
10	128
12	192
14	256

Now coming to the encryption process, message pass through different rounds of encryption where each round uses different encryption key. Also each round consist of four more steps which are:

1. Sub bytes
2. Shift rows
3. Mix columns
4. Add round key

“Add round key” of each round becomes input of next round and for last round it becomes encrypted text.

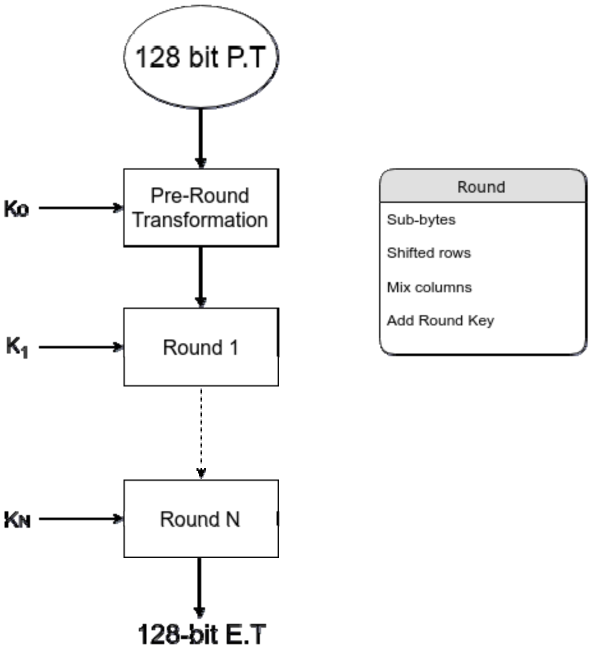

Fig. 1 AES Encryption process

5. EXPERIMENTS

Planning and Configuration

In our tests, we measure the performance of plain-text, XOR cypher and AES cypher. We worked with the following three scenarios:

- Send plain-text message via process 1 and receive plain-text message via process 2
- Encode message with XOR and send via process 1 and receive message and decode via process 2
- Encode message with AES and send via process 1 and receive message and decode via process 2.

In more detail, for the encoding and decoding processes with XOR, a truly random 32B seed is hard-coded in the executable and the sender and receiver processes generate an arbitrarily large key using that seed. For AES, a key is generated and stored into a file. Sender and receiver processes must have access to the file in order to encrypt/decrypt the message. For random number generation and AES, libsodium 1.19 was used.

Results

With message sizes up to 8KB, all processes were able to send or receive each message within less than a millisecond. IPC isn't really targeted to high bandwidth communication. For the purposes of high bandwidth, shared memory is a better option. It is known that, for long strings of data, XOR outperforms AES in terms of speed. With hardware support to AES, this advantage of XOR diminishes or vanishes. One detail with XOR encryption is that it must be implemented carefully. A security measure is to restrict key sizes to at least the same size of the message itself. This measure renders an attacker unable to map symbols to the underlying alphabet. One point of advantage with AES is that, as it is implemented in hardware in recent microprocessors, it does not suffer of the issue of cache collisions that allow eavesdropping private data, thus strengthening security.

With our tests with DDoS, we set a threshold for a control process to destroy the message queue if the number of messages is higher than a given value. The control process was successful in combating the DDoS attacks. An interesting observation is that past 1092 messages sent sequentially to the queue, the OS itself (Debian 9 64-bit) revoked new messages. We don't know exactly why, but it is probably due to a security measure per the OS.

For random number generation and AES, libsodium 1.19 was used. Our implementations are, as far as we know, state-of-the-art in terms of security and we did not found breaches during security/integrity tests.

Table 3: Comparison between message security approaches

	XOR	AES	SHA-2
# ways	Two-way	Two-way	One-way
Key size	Up to 256GB	Up to 256-bit	Up to 512-bit
Application	En/de-cryption	En/de-cryption	Integrity check
Speed	Very high	Medium*	High*
Security	High	High	High
Key exchange	None	Using files	N/A
Hardware-level	No	Yes	Yes

Table 3 shows comparison between message security approaches, Table 4(Last page) shows speed comparison of sending message in different ways, Table 5(last page) shows Comparison between IPC implementations.

6. CONCLUSION

Interprocess communication needs security but not at cost of speed. Method describe in the paper promises both speed and security, on top of that it also saves IPC from DDos attacks and saves system from freezing. Time for execution of each command remains under 1ms, moreover system is compatible with most of the operating systems also comparison between different security approaches was discussed . For future work can be done remove disadvantages of IPC implementation of this method depending on different types of data transfer during IPC that are discussed in Table 5.

ACKNOWLEDGMENTS

The authors would like to thank the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University, Alkharj, Saudi Arabia for the assistance.

References

[1] S. Bagchi & Susmit, "The software architecture for efficient distributed interprocess communication in mobile distributed systems," Journal of grid computing, vol. 12, no. 4, pp. 615–635, 2014.

[2] J. Villalba-Diez and J. Ordieres-Mer´e, "Improving manufacturing performance by standardization of interprocess communication," IEEE Transactions on Engineering Management, vol. 62, no. 3, pp. 351–360, 2015.

[3] B. N. Bershad, T. E. Anderson, E. D. Lazowska, and H. M. Levy, "Userlevel interprocess communication for shared memory multiprocessors," ACM Transactions on Computer Systems (TOCS), vol. 9, no. 2, pp. 175–198, 1991.

[4] J. Day, I. Matta, and K. Mattar, "Networking is IPC: a guiding principle to a better internet," in Proceedings of the 2008 ACM CoNEXT Conference. ACM, 2008, p. 67.

[5] L. Lamport, "On Interprocess Communication - Parts I: Basic Formalism," Distributed Computing, vol. 1, no. 2, pp. 77–101, 1986.

[6] J. Gu, S. S. Lumetta, R. Kumar, and Y. Sun, "MOPED: Orchestratinginterprocess message data on CMPs," in 2011 IEEE 17th International Symposium on High Performance Computer Architecture. IEEE, 2011, pp. 111–120.

[7] Eljialy, A.E.M. and Ahmad, S., 2019, November. Errors Detection Mechanism in Big Data. In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 323-328). IEEE.

[8] D. Gelernter& David, “Generative communication in Linda,” ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 7, no. 1, pp. 80–112, 1985.

[9] Sultan, Ahmad, Sudan Jha, Abubaker EM Eljialy, and Shakir Khan. "A Systematic Review on e-Wastage Frameworks." International Journal of Advanced Computer Science and Applications 12, no. 12 (2021).

[10] D. C. Walden and David, “Systems for Interprocess Communication in a Resource Sharing Computer Network,” 1970.

[11] D. Moore, E. Olson, and A. Huang, “Lightweight communications and marshalling for low-latency interprocess communication,” 2009.

[12] L. L. Peterson, N. C. Buchholz, and R. D. Schlichting, “Preserving and using context information in interprocess communication,” ACM Transactions on Computer Systems (TOCS), vol. 7, no. 3, pp. 217–246, 1989.

[13] N. Artenstein and I. Revivo, “Man in the binder: He who controls ipc, controls the droid,” Black Hat, p. 81, 2014.

[14] M. Salehi, F. Daryabar, and M. H. Tadayon, “Welcome to Binder: A kernel level attack model for the Binder in Android operating system,” in 2016 8th International Symposium on Telecommunications (IST). IEEE, 2016, pp. 156–161.

[15] C. Yuan, Y. Yue, X. Li, and L. Feng, “Performance analysis and optimization of inter process communication in Android,” in 2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC). IEEE, 2016, pp. 297–300.

[16] B. Caillat, B. Gilbert, R. Kemmerer, C. Kruegel, and G. Vigna, “Prison: Tracking process interactions to contain malware,” in 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015, pp. 1282–1291.

[17] P. Xiao, Y. Li, and W. Deng, “A model of distributed interprocess communication system,” in 2009 Second International Workshop on Knowledge Discovery and Data Mining. IEEE, 2009, pp. 276–279.

[18] T. Tabata, K. Fukutomi, and H. Taniguchi, “Proposal of Instant Synchronous Interprocess Communication,” in 2008 Third International Conference on Convergence and Hybrid Information Technology, vol. 2. IEEE, 2008, pp. 146–149.

[19] Uddin, Mohammed Yousuf, Ahmad Sultan, and Mohammad Mazhar Afzal. "Disposable Virtual Machines and Challenges to Digital Forensics Investigation." International Journal of Advanced Computer Science and Applications 12, no. 2 (2021).

[20] Ahmad S, Jha S, Alam A, Alharbi M, Nazeer J. Analysis of Intrusion Detection Approaches for Network Traffic Anomalies with Comparative Analysis on Botnets (2008–2020). Security and Communication Networks. 2022 May 12;2022.

[21] Ahmed MA, Eljialy AE, Ahmad S. Memory test and repair technique for SoC based devices. IEICE Electronics Express. 2021:18-20210092.

Table 4: Speed Comparison

	Send/plain-text	Send/AES encryption	Send/XOR encryption	Recv/plaintext	Recv/XOR decryption	Recv/AES decryption
Msg ¡= 128B	< 1ms	<1ms	< 1ms	< 1ms	< 1ms	< 1ms
Msg ¡= 1024B	< 1ms	<1ms	< 1ms	< 1ms	< 1ms	< 1m
Msg ¡= 8192B	< 1ms	<1ms	< 1ms	< 1ms	< 1ms	< 1ms

Table 5: Comparison between IPC implementations

Implementation	Security advantages	Security disadvantages	Implemented in
File	Ease of access control; sandardized in most operating systems	The file must be given read access; storage medium may pose risk to data integrity and security; permanent storage of data allows stealing past messages	Most operating systems
Signal	Does not need to read any data from the sending process	Attacker with control of the signaling process poses risk to system security	Most operating systems
Socket	Performing implementations of security protocols for sockets are popular	Easy eavesdropping of transmitted data; Exposition to external offenses; Control of message boundaries is not strict; Messaging queues must be implemented	Most operating systems
Message queue	Strict delimitation of message boundaries; OS is given control of the queue; Does not need to expose encryption keys	Messages must be encrypted for security	Most operating systems