DOI: 10.15849/IJASCA.211128.03

Int. J. Advance Soft Compu. Appl, Vol. 13, No. 3, November 2021
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Adaptive Multi-Applications Cryptographic System

Basil Al-Kasasbeh

Faculty of Computer Studies, Arab Open University, Saudi Arabia
e-mail: bkasasbah@arabou.edu.sa

Abstract

Cryptography is the core method utilized to protect the communications between different applications, terminals, and agents distributed worldwide and connected via the internet. Yet, with the distribution of the low-energy and low-storage devices, in the Internet-of-Things (IoT), the cryptography protocols cannot be implemented because of the power constraints or because the implementation is beyond the time constraints that hindered their usability of these protocols in real-time critical applications. To solve this problem, an Adaptive Multi-Application Cryptography System is proposed in this paper. The proposed system consists of the requirements identifier and the implementer, implemented on the application and transportation layer. The requirement identifier examines the header of the data, determines the underlying application and its type. The requirements are then identified and encoded according to four options: high, moderate, low, and no security requirements. The inputs are processed, and ciphertext is produced based on the identified requirements and the suitable cryptography algorithm. The results showed that the proposed system reduces the delay by 97% relative to the utilized algorithms' upper-bound delay.

Keywords: *Cryptography, symmetric key encryption, block cipher, delay and performance, quantum computing.*

1 Introduction

The growth in internet applications leads to the distribution of Real-time applications (RTA) and telecommunication services like IP-Telephony and VoIP, which grow as a successful businesses in the world [1]. Besides, healthcare applications and agriculture applications are developed as time-critical applications using the Internet-of-Things (IoT) [2]. Quality of Service (QoS) is considered as a major issue in such applications, more specifically, the delay parameters, which consists of many types of delays, such as packetization delay (sampling, coder-decoder (codec), compression and encryption), and end-to-end delay (processing, queuing, serialization and propagation delays) [3]. Accordingly, real-time applications (RTA) are distributed worldwide and connected via the internet, and the recently developed IoT require high-performance communications and are harmed badly with packet delay. Thus, high-performance security protocols and systems for securing RTA communications are in demand. While the best way to secure communication over the internet is cryptography, the encryption and decryption processes require more processing time than the other transmission processes, such as routing, encapsulation, and decapsulation, turn harms the Quality of Service (QoS) [4]. Accordingly, the efforts should be directed towards building applications that can deal with different QoS levels, both basic and costume qualities that may involve preserving confidentiality, making it more complicated and may result in higher delay. These applications required lightweight encryption and decryption processes, yet each with identified requirements balance performance and confidentiality [5, 6].

Many applications are required to convey the bulk of information among the users in a secure way. The best way to secure information is by using cryptography through the encryption and decryption processes [7]. Encryption is the process of transform messages into a form unreadable for everyone except the intended receiver. Encrypted data should be decrypted first, and then it can be read by the receiver. A classical encryption algorithm hides the actual message. For example, letters of the message are substituted or transposed to different letters, letter pairs, and many letters, as illustrated in Fig. 1 [8].

Cryptography is the science of encrypting and decrypting data. Depend on complex mathematics, cryptography offers many effective information security services include confidentiality, authentication, integrity, and non-repudiation. Cryptography protocols and programs simplify the encryption process and allow users to secure their data without carrying out the complex mathematics themselves. Modern cryptography relies on cryptographic keys, typically a short string of text, for encoding and decoding messages in combination with cryptographic techniques. Depend on the type of keys, the algorithms are classified as either symmetric or asymmetric key cryptography. Both symmetric and asymmetric key cryptography offer data confidentiality. Asymmetric key encryption is sometimes called public-key encryption. Digital signatures, one of the by-products of public-key cryptography, allow the verification of authenticity, integrity, and non-repudiation [9].

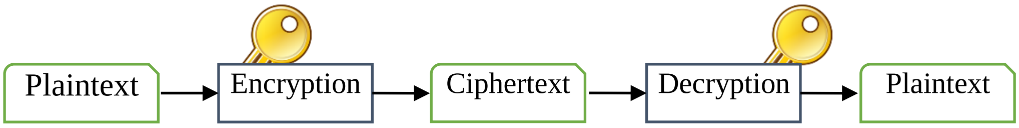

Fig. 1: Cryptography encryption/decryption process

The symmetric algorithms, specifically stream ciphers, are among the most vital fundamental techniques for converting a block of data at high speed [10, 11]. Asymmetric cipher is used public and private keys uses as a pair. The public key is for the public, and the private key is only known by the owner. The public key is very efficient for authentication and encryption [12]. Quantum computing is a type of computing that uses quantum phenomena (photons and their polarization, a quantized characteristic) to implement complex computational tasks. Thus, quantum computers are developed to solve complex computational problems by which cryptography is grounded. Quantum computer encodes the information in qubits, which is the unit for quantum cryptography. Quantum cryptography is a new technique for securing computer network communication channels. Quantum cryptography is secure as it depends on the inalienable quantum mechanics laws [13]. Existing standard cryptography systems use advanced techniques to generate key pairs, which are very hard to inverse engineer, as illustrated in Figure 1. Quantum cryptography avoids any mathematical technique and uses the principles of quantum physics. Quantum cryptography implements a new algorithm for creating and exchanging cryptography keys, making it impossible for third-party entities to get those keys. Keys created in this way will automatically destroy themselves if read by a third-party interferer [14]. It is almost impossible to interrupt the encryption algorithm without knowing the exact key value because of the internal key generation with the entered key's reference. Accordingly, encryption and decryption can be implemented for general applications to send confidential data and sending the internal key to the sender using a quantum key exchange channel, which is secure to the receiver [15].

Because the quantum security is ensured as only the key is transmitted without any possibility for information leak, the problem left is with the security level of the encryption algorithm and the performance of these algorithms. Different data, based on the application form in which these data is generated or emitted, requires different security levels. For example, email data might be of high risk and require a high level of security. The performance in encryption/decryption of the email data might not be critical, as such data is delay insensitive. In contrast, Voice Over IP (VOIP) data in the non-secure channel required low-level security, demanding a high-performance algorithm [16].

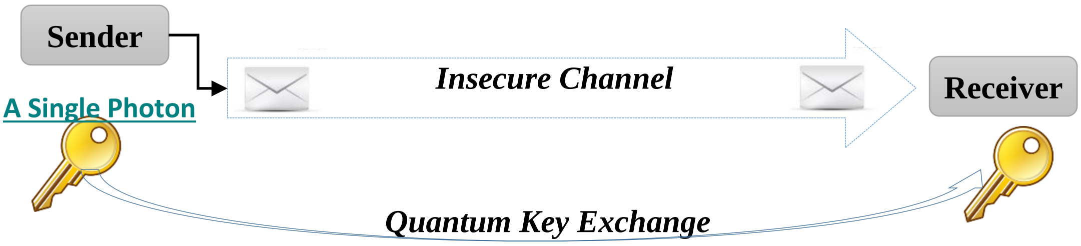

Fig. 2: Quantum cryptography

Accordingly, this research proposed a system to lessen the encryption delay while maintaining the confidentiality of traditional and quantum computing. The proposed system uses symmetric encryption algorithms that allow the sender to choose among different algorithms depends on the type of data to be encrypted and transmitted. Using the proposed system, the encryption process is made adaptive, such as the type of encryption algorithm utilized depending on the application and the data to be encrypted and transmitted.

2 Related Work

Existing algorithms for encryption vary in terms of the input style (block vs. stream), the key length, the implemented operations, and the number of rounds. In the literature, combining cryptography algorithms resulted in a new algorithm to improve the security or reduce the complexity of the original algorithm. In this context, Srikantaswamy and Phaneendra [17] proposed an encryption algorithm by extending the Caesar cipher and the columnar transposition cipher algorithms. The proposed encryption algorithm used random number generation to assign a key for the encryption and decryption processes. Kester [18] combined vigenere cipher and columnar transposition cipher algorithms to improve the security level and overcome the algorithms' limitations. The columnar transposition is used as the key generator, and the vigenere cipher is used for encryption and decryption. Omolara, Oludare [19] proposed a system of cryptography, which involves multiple applications of columnar transposition, alongside other forms of Caesar cipher techniques aiming at increasing the security of the columnar transposition cipher technique. Singh, Maakar [20] extends the data encryption standard (DES) to overcome its limitation to brute force attacks. To enhance the security of the DES algorithm, the transposition algorithm is implemented before the DES algorithm. Accordingly, breaking into such a technique required first breaking the original DES algorithm and then the transposition algorithm [21].

Nan Li et al. extended the Diffie Hellman protocol. It proves its weakness to the man-in-the-middle and impersonation attacks in practice as it has no entity authentication mechanism. Accordingly, various authentication techniques have been reviewed to be combined with the Diffie Hellman and compared based on its computational efficiency. Then, an improved key exchange schema based on hash function is given, which improves the security and practicality of the Diffie-Hellman protocol [22, 23].

Surveys and comparison studies for block cipher algorithms were presented in the literature. Surya and Diviya [24] compare the symmetric block cipher algorithms based on the security requirements, which are privacy, integrity, authentication, non-repudiation, and access control. Albermany and Radihamade [25] presented a survey on symmetric algorithms such as DES, AES, Triple DES, and Blowfish, and asymmetric algorithms public-key algorithms that use two different keys, such as RSA [26]. Mandal [27] presented a comparison between four of the most commonly used symmetric key algorithms: DES, 3DES, AES, and blowfish. The comparison was made based on the rounds block length, key length, encryption and decryption time, CPU processing time, throughput, and power consumption. These results show that blowfish is the best among the compared algorithms. AES is better than 3DES and DES in terms of throughput & decryption time [28].

Rejani and Krishnan [29] performed considering security, throughput, speed, encryption/decryption, power consumption, and other factors. It is presented that the blowfish algorithm has better performance than other symmetric algorithms like DES and 3DES, AES having better performance. The memory requirement is smaller than asymmetric encryption algorithms, and symmetric key algorithms run quicker than asymmetric key algorithms. Additionally, symmetric key encryption offers more security than asymmetric key encryption. Rani and Kaur [30] reviewed various cryptography algorithms for network security, some related work already done by various authors, existing work problems, and some proposals for proposed work. To protect the intended data from hacking, cryptography is performed [31]. Hossain, Hossain [32] discussed the basic characteristics (e.g., key length and block size) of the symmetric algorithms; these are AES, DES, 3DES, BLOWFISH, RC4, the asymmetric algorithms, are RSA, DSA, Diffie-Hellman, EI-Gamal, Paillier, the hashing functions, these are MD5, MD6, SHA, SHA256. Besides, an empirical evaluation of the AES, DES, BLOWFISH, DES, RC4, and RSA was conducted, and the time is analyzed for encryption and decryption with different file sizes [33].

Various conclusions about the existing algorithms for symmetric and asymmetric cryptography algorithms have been reported in the literature. Singh and Shende [34] concluded that each algorithm has its advantages and shortcomings and that all algorithms are useful for real-time encryption. Every algorithm is unique in its way, which may be suitable for a specific application(s). Every day a new encryption method is evolving. The encryption algorithms will continuously work out with a high-security rate, as for the symmetric and asymmetric techniques, each with its loopholes [35]. Kashyap and Madan [36] concluded that cryptography algorithms' throughput depends on the encryption time, CPU time, and packet size. The throughput of the encryption scheme is calculated as plaintext size per second. As such, as the throughput increases, the power consumption is decreased and vice versa. According to the conducted experiments, AES has proved the best algorithm in performance and security but has a high power consumption [37].

Quantum cryptography was created by Bennett, Brassard [38]. Quantum coding was initially presented by Gottesman and Chuang [39] in 1983. Bennett and Brassard [40] used quantum coding and public-key cryptographic techniques to yield numerous unforgeable subway tokens schemes. Several others contributed to quantum cryptography and quantum key distribution. Even though quantum computing is not that feasible, quantum cryptography is reachable over shorter distances [41]. In quantum computing, implementing the cryptography algorithms in a perfect security environment was discussed by Morimae and Koshiba [42]. Aaronson, Cojocaru [43] proves that quantum computing should consider the same security risks faced in classical computing. Accordingly, the algorithms that have been used in classical computing should be implemented according to the quantum settings and based on their characteristics and advantages, as have been reviewed before. Similar arguments were made by Mantri, Demarie [44] about quantum computing and classical computing cryptography [45]. Accordingly, quantum computing is placed on top of the classical cryptography algorithms that have been discussed in this section. As such, a framework for adaptive security for such a case is proposed to overcome the security, limitation, and performance limitations discussed in this section.

3 The Proposed System

The proposed adaptive multi-application cryptography system is developed to determine the security requirement for each application. The requirements are identified based on the desired security level and the desired performance. The system consists of two components, as illustrated in Fig. 3; these are the requirements identifier and the cryptosystem, implemented on the application and transportation layer. The inputs are processed based on the data's application source and produce a ciphertext based on the identified requirements.

The proposed system identifies the type of application, security required, and delay sensitivity in the application layer. Accordingly, the applications are categorized into two types, real-time and on demands. The security level can be non-cryptographic, low-level cryptography, moderate and advanced cryptography. Finally, the delay sensitivity can be true or false. Each data packet is identified by a 4-bit code representing the specification and requirements as summarized in Table 1.

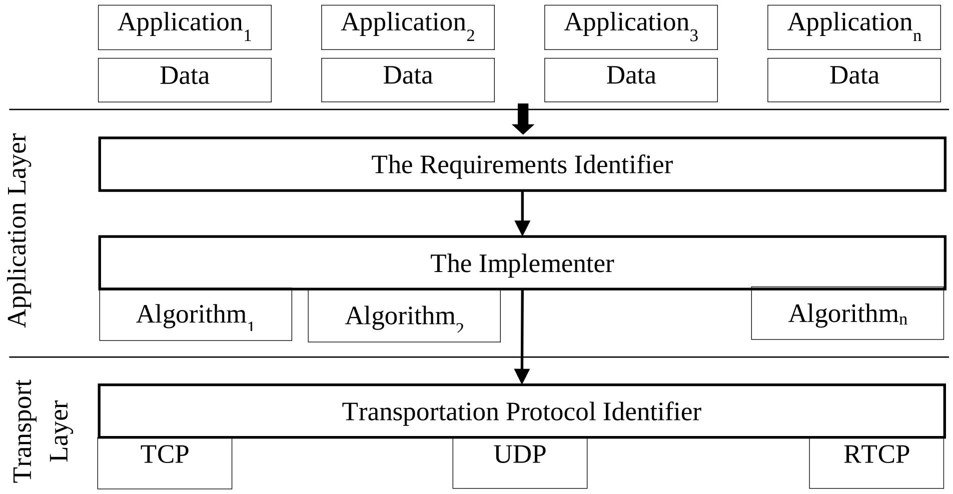

Fig. 3: The proposed system

Table 1: Identification Code Explanation

Bit-Order	Usability	Options
1st bit	Application Type	On-demands (0), Real-time (1)
2nd & 3rd	Security Requirements	No-Cryptography (00), Low-Level (01), Moderate (01), Advanced (11)
4th bit	Delay Sensitivity	Insensitive (0), Sensitive (1)

3.1 The Requirement Identifier

The proposed system encodes the requirements into a special field that can recognize the type of an application and the networking requirements based on the application layer protocol for this particular application using 4-bit XYZD, as given in Fig. 4. The 4-bit code is amended to the file header to identify the requirements, as have been discussed before. Based on this code, the sender and received implementer module can determine the cryptography algorithm by which the data will be encrypted/decrypted.

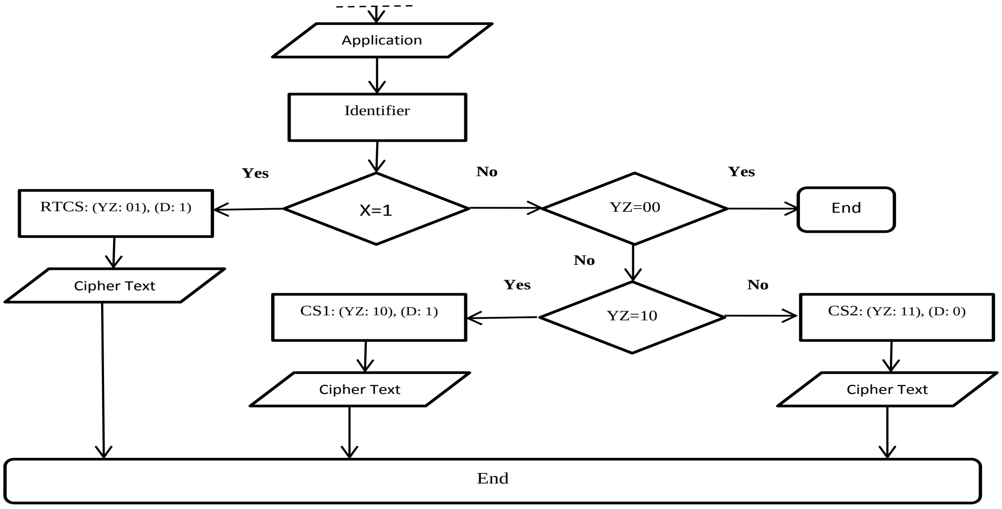

Fig.4: The 4-Bit code analysis processed

3.2 The Implementer

The implementer encrypts/decrypts the data based on the identified requirements and uses one of the selected cryptography algorithms: Data Encryption Standard (DES), Advanced Encryption Standard (AES) Blowfish, RC4, and DEA-RTA are among the best and commonly utilized algorithms for encryption [29, 30, 32, 33]. According to the identified requirements, each of these algorithms' utilization is identified based on their characteristics, as given in Table 2.

Data Encryption Standard (DES) is a symmetric block encryption algorithm with a 64-bit key, 56 bits of which make up the independent key, and 8 bits for parity-check or error detection. The encryption and decryption algorithms are similar except, the keys are used in opposite orders. DES algorithm structure is based on the Feistel function that divided the block into two halves and implemented in four phases, expansion, key mixing, substitution, and permutation. DES is implemented in 16 rounds, and the output consists of 64 bits that are the function of the input message and the key. Advanced Encryption Standard (AES) is a symmetric block encryption algorithm with a 128-bit key (AES-128), 192-bit key (AES-192), and 256-bit key (AES256). AES encrypts a 128-bit data length that can be divide into four basic operational blocks. The blocks are considered an array of bytes and organized as a 4x4 matrix called a state. The number of rounds used in encryption is 10, 12, and 14 for the key length of 128-bit,192-bit, and 256-bit, respectively. Blowfish is a symmetric block encryption algorithm with a 32-bit to 448-bit key and block size of 64-bit. Blowfish is based on the fiestel function (MS, 2014), and it is implemented in16 rounds for the encryption process. Rivest Cipher 4 (RC4) was developed by Ron Rivest as a symmetric stream cipher encryption algorithm. The algorithm is mutual for both encoding and decoding [29, 30, 32, 33]. The RC4 produces a pseudorandom stream of bits (keystream) based on bit-wise operations. The operations involved the permutation of the 256-byte key and two 8-bit index-pointers. DEA-RTA is a symmetric stream cipher encryption algorithm that allows users to choose the key length for each packet. The DEA-RTA properties are as follows: The encryption key is flexible in length, the plain text is flexible in size, the encryption process is very simple, the transposition table is simple too, the shifted transposition table is easy to initiate, and complex to regenerate. These properties result in better encryption delays while maintaining confidentiality.

Table 2: Cryptography Algorithm Characteristics and Utilization in the Adaptive Framework

Algorithm	Time Delay		Feature	Used For
	Encryption	Decryption		
AES	*Average (≈15)*	*Average (≈9)*	*Excellent Security*	*CS2*
DES	*High (≈54)*	*High (≈53)*	*Low Security*	*CS1*
Blowfish	*High (≈37)*	*High (≈30)*	*Excellent Security*	*CS2*
RC4	*Low (≈8)*	*Low (≈6)*	*Good Security*	*RTCS*
DEA-RTA	*Very low (≈1)*	*Low (≈2)*	*Excellent Security*	*RTCS*
**Time provided for 1 M.B. file in Millisecond*				

4 Simulation Results

The proposed and compared methods were implemented in NetBeans Integrated Development Environment (IDE) and Java Development Kit (JDK 1.6). The experiments are carried out on a machine of I7 CPU with 8-GB RAM. The proposed system is simulated using the selected cryptography algorithms and input files of different sizes; these are 1 K.B., 3 K.B., 1 M.B., and 3MB. Both encryption and decryption process was implemented within the same conditions. Fig. 5 illustrates the complicated algorithms' encryption time using different file sizes, while the decryption time is illustrated in Fig. 6.

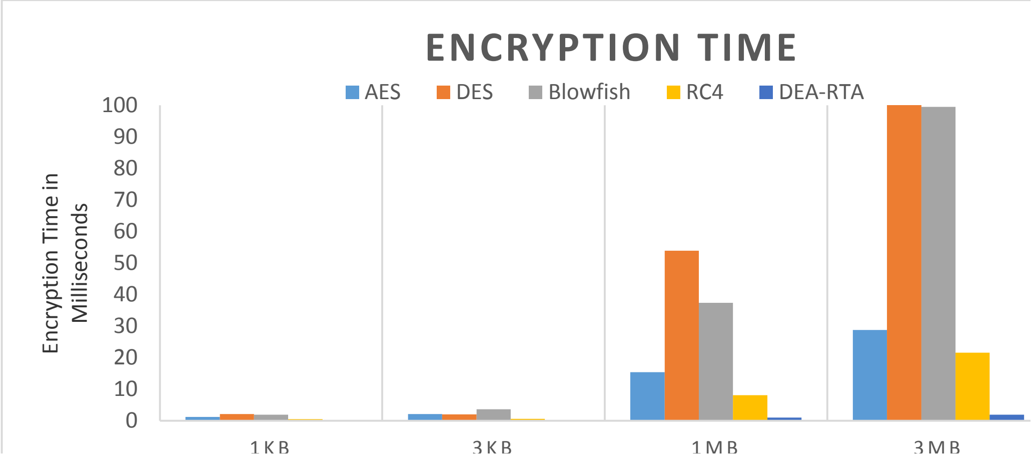

Fig. 5: Comparison of the Encryption Algorithms based on the Encryption Time

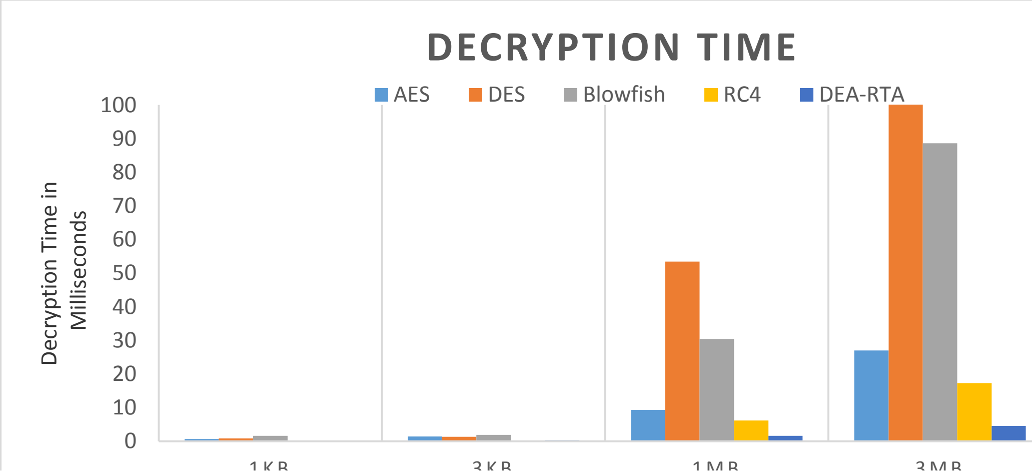

Fig. 6: Comparison of the Encryption Algorithms based on the Decryption Time

Accordingly, the proposed system delay minimization can be related to differences between the lower and upper bound of the encryption and decryption time, as given in Table 3. An application with low confidentiality requirements and high sensitivity to delay is said to improve the delay by 97% using the proposed system.

Table 3: System Performance-based on Different Cryptography Algorithms (Milliseconds)

	AES	*DES*	*Blowfish*	*RC4*	*DEA-RTA*	*Upper-Bound*	*Lower-Bound*	*Upper-Bound Reduction*
1KB	1.8281	2.9007	3.5455	0.4892	0.1877	3.5455	0.1877	94.71%
3KB	3.4445	3.2327	5.5496	0.6374	0.3666	5.5496	0.3666	93.39%
1MB	24.6312	107.2775	67.8103	14.2206	2.5724	107.2775	2.5724	97.60%
3MB	55.7931	261.8749	188.1286	38.8591	6.4221	261.8749	6.4221	97.55%

5 Conclusion

In this paper, a system for cryptography is proposed using requirement identifiers and cryptography algorithms that are used adaptively based on the encrypted data. The proposed adaptive multi-application cryptography system is developed to determine the security requirement based on the desired security level and the desired performance. The proposed system is simulated using the selected cryptography algorithms and input files of different sizes; these are 1 K.B., 3 K.B., 1 M.B., and 3MB. The proposed system delay minimization linked to differences between the lower bound and the upper bound of the encryption and decryption time showed that the proposed system improves the delay by 97%. Accordingly, the proposed framework for adaptive security overcomes the security limitation and performance limitation that faces the current individual implementation of the cryptography algorithms.

ACKNOWLEDGEMENTS

The author would like to thank Arab Open University and Oracle-KSA for supporting this study.

References

[1] Mazurczyk, W., *VoIP steganography and its detection—a survey.* ACM Computing Surveys (CSUR), 2013. **46**(2): p. 1-21.

[2] Asghari, P., A.M. Rahmani, and H.H.S. Javadi, *Internet of Things applications: A systematic review.* Computer Networks, 2019. **148**: p. 241-261.

[3] Alkhatib, A.A., et al., *A novel method for localising a randomly distributed wireless sensor network.* International Journal of System Assurance Engineering and Management, 2018. **9**(2): p. 354-361.

[4] Kawadia, V. and P. Kumar, *Principles and protocols for power control in wireless ad hoc networks.* IEEE journal on selected areas in communications, 2005. **23**(1): p. 76-88.

[5] Wachter, S., *Normative challenges of identification in the Internet of Things: Privacy, profiling, discrimination, and the GDPR.* Computer law & security review, 2018. **34**(3): p. 436-449.

[6] Hnaif, A.A., et al., *Multiprocessing scalable string matching algorithm for network intrusion detection system.* International Journal of High Performance Systems Architecture, 2018. **8**(3): p. 159-168.

[7] Hnaif, A.A. and M.A. Alia, *Mobile payment method based on public-key cryptography.* International Journal of computer networks & communications, 2015. **7**(2): p. 81.

[8] Yi, X., R. Paulet, and E. Bertino, *Homomorphic encryption*, in *Homomorphic Encryption and Applications*. 2014, Springer. p. 27-46.

[9] Joshi, M.R. and R.A. Karkade, *Network security with cryptography*. International Journal of Computer Science and Mobile Computing, 2015. **4**(1): p. 201-204.

[10] Bokhari, M.U., S. Alam, and F.S. Masoodi, *Cryptanalysis techniques for stream cipher: a survey*. International Journal of Computer Applications, 2012. **60**(9).

[11] Sharma, P. and R. Purohit, *Performance Evaluation of Symmetric Block Cipher RC6 with ECB and CBC Operation Modes*, in *International Conference on Intelligent Data Communication Technologies and Internet of Things*. 2018, Springer: Coimbatore, India. p. 134-140.

[12] Khan, A.G., S. Basharat, and M.U. Riaz, *Analysis of asymmetric cryptography in information security based on computational study to ensure confidentiality during information exchange.* International Journal of Scientific & Engineering Research, 2018. **9**(10): p. 992-999.

[13] Padamvathi, V., B.V. Vardhan, and A. Krishna. *Quantum cryptography and quantum key distribution protocols: a survey*. in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. 2016. IEEE.

[14] Tan, X., *Introduction to Quantum Cryptography*. 2013.

[15] Pandey, K.K., V. Rangari, and S. Kumar, *An Enhanced Symmetric Key Cryptography Algorithm to Improve Data Security.* International Journal of Computer Applications, 2013. **74**: p. 0975 – 8887.

[16] Fayed, M.A., *A security coprocessor for next generation IP telephony: architecture, abstraction, and strategies*. 2007: University of Victoria.

[17] Srikantaswamy, S. and D.H. Phaneendra, *Improved Caesar Cipher with Random Number Generation Technique and Multistage Encryption.* International Journal on Cryptography and Information Security (IJCIS), 2012. **2**(4): p. 39-49.

[18] Kester, Q.-A., *A Hybrid Cryptosystem based on Vigenere cipher and Columnar Transposition cipher.* International Journal of Advanced Technology and Engineering Research (IJATER), 2013. **3**(11): p. 141-147.

[19] Omolara, O., A. Oludare, and S. Abdulahi, *Developing a modified Hybrid Caesar cipher and Vigenere cipher for secure Data Communication.* Computer Engineering and Intelligent Systems, 2014. **5**(5).

[20] Singh, S., S.K. Maakar, and S. Kumar, *A performance analysis of DES and RSA cryptography.* International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2013. **2**(3): p. 418-423.

[21] Bala, T. and Y. Kumar, *Asymmetric Algorithms and Symmetric Algorithms: A Review* International Journal of Computer Application (ICAET), 2015: p. 1-4.

[22] Li, N., *Research on Diffie-Hellman Key Exchange Protocol*, in *2nd International Conference on Computer Engineering and Technology*. 2010: Chengdu, China p. 634-637.

[23] Devi, S. and R. Makani, *Generation of N-party Man-In-Middle Attack for Diffie–Hellman Key Exchange Protocol: A Review.* International Journal of Computer Science and Information Technologies, 2015. **6**(5): p. 4281-4285.

[24] Surya, E. and C. Diviya, *A Survey on Symmetric Key Encryption Algorithms.* International Journal of Computer Science & Communication Networks, 2012. **2**(4): p. 475-477.

[25] Albermany, S. and F. Radihamade, *Survey: block cipher methods.* Int. J. Adv. Res. Technol, 2016. **5**(11): p. 11-22.

[26] !!! INVALID CITATION !!! .

[27] Mandal, P.C., *Evaluation of performance of the Symmetric Key Algorithms: DES, 3DES, AES and Blowfish.* Journal of Global Research in Computer Science, 2012. **3**(8): p. 67-70.

[28] Singh, G., Ashwani Kr Singla, and K.S. Sandha., *Superiority of blowfish algorithm in wireless networks.* International Journal of Computer Applications, 2012. **44**(11): p. 23-26.

[29] Rejani, R. and D.V. Krishnan, *Study of symmetric key cryptography algorithms.* International Journal of Computer Techniques, 2015. **2**(2): p. 45-50.

[30] Rani, S. and H. Kaur, *Technical Review on Symmetric and Asymmetric Cryptography Algorithms.* International Journal of Advanced Research in Computer Science, 2017. **8**(4).

[31] Sonia, R. and K. Harpreet, *Technical Survey on Cryptography Algorithms for Network Security.* International Journal of Advanced Research in Computer Science and Software Engineering, 2016. **6**(9): p. 204-209.

[32] Hossain, M.A., et al., *Performance analysis of different cryptography algorithms.* International Journal of Advanced Research in Computer Science and Software Engineering, 2016. **6**(3): p. 659-665.

[33] Rejani, R. and V.K. Deepu, *Study of Symmetric Key Network Security Algorithms.* International Journal of Computer Techniques, 2015. **2**(2): p. 45-50.

[34] Singh, P. and P. Shende, *Symmetric key cryptography: current trends.* International Journal of Computer Science and Mobile Computing, 2014. **3**(12): p. 410-415.

[35] Singh, P. and P. Shende, *Symmetric Key Cryptography: Current Trends.* International Journal of Computer Science and Mobile Computing, 2014. **3**(12).

[36] Kashyap, S. and E.N. Madan, *A review on: network security and cryptographic algorithm.* International Journal of Advanced Research in Computer Science and Software Engineering, 2015. **5**(4): p. 1414-1419.

[37] Kashyap, S. and E.N. Madan, *A Review on: Network Security and Cryptographic Algorithm.* International Journal of Advanced Research in Computer Science and Software Engineering, 2015. **5**(4).

[38] Bennett, C.H., G. Brassard, and S. Breidbard, *Quantum Cryptography, or Unforgeable Subway Tokens*, in *Advances in Cryptology: Proceedings of CRYPTO '82*. 1982, Plenum. p. 267-275.

[39] Gottesman, D. and I. Chuang, *Quantum Digital Signatures*. 2001, 2001: Quantum Physicse-Print archive.

[40] Bennett, C.H. and G. Brassard, *Quantum Cryptography: Public Key Distribution and Coin Tossing.* Theoretical Computer Science, 1984. **560**(1): p. 175-179.

[41] Kute, S. and C. Desai, *Quantum Cryptography: A Review.* Indian Journal of Science and Technology, 2017. **10**(3): p. 1-5.

[42] Morimae, T. and T. Koshiba, *Impossibility of perfectly-secure one-round delegated quantum computing for classical client.* Quantum Information & Computation 2019. **19**(3): p. 214–221.

[43] Aaronson, S., et al., *Complexity-theoretic limitations on blind delegated quantum computation*, in *The 46th International Colloquium on Automata, Languages and Programming*. 2019: Patras, Greece.

[44] Mantri, A., et al., *Flow ambiguity: A path towards classically driven blind quantum computation.* Physical Review X, 2017. **7**(3): p. 031004.

[45] Badertscher, C., et al. *Security limitations of classical-client delegated quantum computing*. in *International Conference on the Theory and Application of Cryptology and Information Security*. 2020. Daejeon, Korea (Republic of): Springer.

Notes on contributors

Basil Al Kasasbeh is an associate professor at the Faculty of Computer Studies, Arab Open University, Saudi Arabia. His research interests include: Computer Networks, Mobile system and Cyber security.