Literature review on Data Encryption, its algorithms, and the future of Data Encryption

Kevin Niland, *Computing in Software Development (Honours), GMIT*

***Abstract*—Data encryption is integral to many different areas in the computing world. Encryption is used to help protect sensitive data by changing it into a form only readable by a person or entity who has the decryption key. Colossal amounts of personal information and data is managed and stored in the cloud/servers. With computers becoming increasingly more and more powerful - ways of measuring this include Moore's Law, which suggests that the number of transistors in a microchip/integrated circuit doubles almost every two years, and Million Instructions Per Second (MIPS), which was an older measure of a computer's speed and power. MIPS roughly measured the number of machine instructions that a computer could execute or perfrom in one second (in 1951, a UNIVAC I processor had a known MIPS of 0.002 at 2.25 MHz. In 2017, an AMD Ryzen 7 1800X processor had a known MIPS of 304,510 at 3.6 GHz) - so too comes the risk that peoples personal information can be accessed by an unauthorized party. As a result, several different algorithms and methods have emerged to prevent this. This paper will look at three different encryption algorithms, the latter two of these algorithms are in use today: the Data Encryption Standard (DES) [1], the Rivest–Shamir–Adleman (RSA) algorithm [12], and the Advanced Encryption Standard (AES) [2], which came about 23 years after the RSA algorithm and 26 years after DES. In this paper, each of these algorithms are discussed at length in regards to how they came about, and their advantages and disadvantages. This paper will also look at the future of data encryption and various methods that could feasibly be proposed as possible replacements for current algorithms and techniques.**

I. Introduction

Data encryption is a technique/form of security where sensitive data is encoded/encrypted and can only be decrypted by a person/entity with the corresponding decryption key. Encrypted data (cipher text) appears indecipherable/unreadable to a person or entity that tries to access it without permission. Decrypted data (plain text) appears readable to a person or entity accessing it with permission. A block cipher transforms plain text blocks of a fixed length n_b to cipher text blocks of the same length under the influence of a cipher key k [13]. A Feistel cipher is a symmetric structure used in the construction of block ciphers. Figure 1 shows the encryption process:

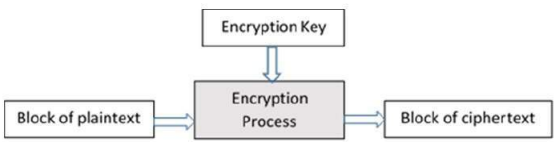

Fig. 1. Block Cipher

The two main types of data encryption that exist are: symmetric encryption and asymmetric encryption (also known as public-key encryption). Asymmetric encryption came about after symmetric encryption, "as an answer for secure communications with an emphasis on applications that could not have been satisfactorily handled by cryptographic techniques at the time" [17].

A. History of Data Encryption

Data encryption, in one form or another, has existed for almost 3000 years. Circa 500 B.C., Spartans used a device called a scytale to send secret messages during a battle [11]. Circa 60 B.C., Julius Caesar invented a substitution cipher that shifted characters by three places. A becomes D, B becomes E, C becomes F and so on [11]. In 1553, Italian cryptologist Giovan Battista Bellaso proposed the first cipher to use a proper encryption key - it would use an agreed-upon keyword that the recipient needs to know if he or she wants to decode the message. In 1795, Thomas Jefferson invented Jefferson's Wheel Cipher which was later used in the United States Army from 1923 to 1942. However, interestingly, the army never knew about Jefferson's invention. They simply re-invented it and called it the 'M-64' [11]. In 1854, Charles Wheatstone invented the Playfair Cipher, which encrypted pairs of letters instead of single ones and was therefore harder to crack.

B. Current Status of Data Encryption

The above mentioned methods have been proved to be insecure and impractical in terms of modern computing since cryptanalysts - also known as eavesdroppers - could exploit simple features of the cipher text to easily discover the plain text and the decryption key, allowing them to easily decipher any future messages using that same system [5]. As stated, modern computing technology has made it necessary to use far more complex encryption algorithms that are harder to break/crack by cryptanalysts. In parallel, however, cryptanalysts have adopted and developed this technology to improve their ability to break cryptosystems [10], [3], [21].

II. Symmetric Encryption

Symmetric encryption is a form of data encryption where the encryption key and the corresponding decryption key are the same. The persons or entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. Symmetric encryption method differs from asymmetric encryption as in asymmetric encryption, a pair of keys (one public and one private) is used to encrypt and decrypt messages. To discuss symmetric

encryption, this paper will look at the first standardized cipher which was the Data Encryption System (DES).

The Data Encryption Standard (DES) was first published in 1975, standardized in 1977 and ultimately withdrawn as the standard by the National Institute of Standards and Technology in 2001 [1]. It was the first publicly available cryptographic algorithm that was endorsed by the U.S. government. Figure 2 shows the DES structure.

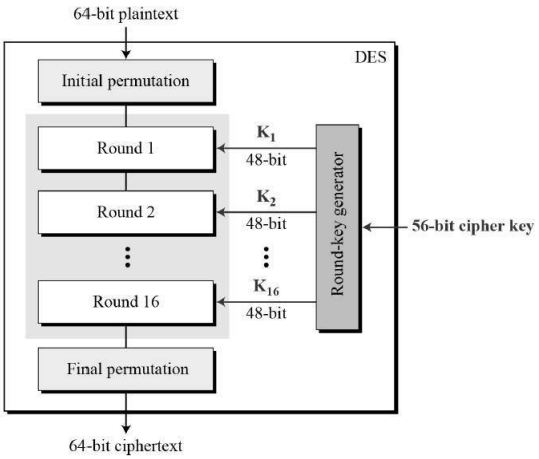

Fig. 2. DES Structure

DES was an outcome of a call for primitives in 1974, which did not result in many serious candidates except for a predecessor of DES, Lucifer, designed by IBM around 1971 [1]. As discussed in Miles E. Smid and Dennis K. Branstad's paper, *"The Data Encryption Standard: Past and Future"*, the DES security controversy [19] forced security questions about how good is good enough and how long is long enough. This controversy came about due to DES' short key length of 56 bits, which was criticized from the beginning. Although this made it too insecure for most current applications (this is evidenced by the fact that it was cracked in 1997 [22]), it was highly influential in the advancement of modern cryptography [8] and in the year 2000, it was replaced by the Advanced Encryption Standard (AES) which was found through a competition open to the public.

The Advanced Data Encryption Standard (AES), originally known as Rijndael, was developed by Vincent Rijmen and Joan Daemen. AES is a subset of the Rijndael block cipher. It was first announced in 2000 by NIST as the surprise winner [2] of its Advanced Data Encryption Standard competition. It subsequently replaced DES in 2001 [18]. Figure 3 shows the AES structure.

Rijndael is a family of ciphers with different key and block sizes. AES and Rijndael differ only in terms of the range of supported values for the block length and cipher key length. In terms of the block length, Rijndael has a minimum of 128 bits and a maximum of 256 bits. AES has a fixed block length

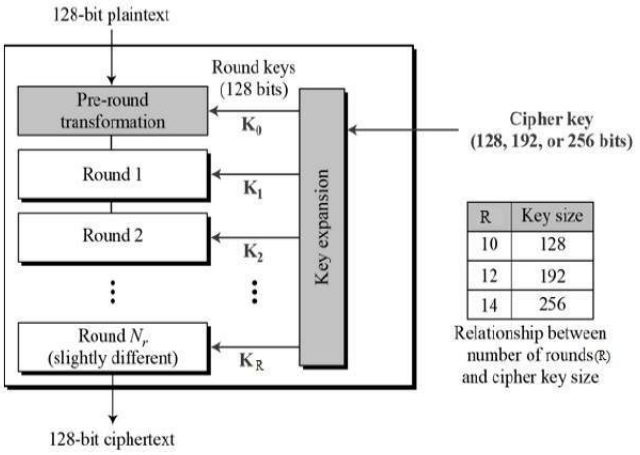

Fig. 3. AES Structure

of 128 bits. In terms of the key length, Rijndael supports any length that is a multiple of 32 bits. AES supports key lengths of 128, 192, or 256 only [14]. Since its inception, it has been adopted by the U.S. government and is now also used worldwide. AES is based on a design principle, known as a substitution–permutation network, which is a series of linked mathematical operations used in block cipher algorithms. Unlike DES, AES doesn't use a Feistel network/cipher. Due to AES being implemented in both hardware and software, it is the most robust and common security protocol. It is also used for a wide array of applications, for example wireless communication and financial transactions. One problem with AES Encryption is that it has slow computation. As described in Rizky Riyaldhi, Rojali, and Aditya Kurniawan's paper, *"Improvement of Advanced Encryption Standard Algorithm With Shift Row and S.Box Modification Mapping in Mix Column"*, their experiments showed that it takes 3.045 milliseconds to compute 1024 bytes of data. This increases by 3 - 4 milliseconds for 2048 bytes of data and so on [15]. Figure 4 shows the computational times published in Rizky Riyaldhi, Rojali, and Aditya Kurniawan's paper.

Bytes of Data	Average Times (ms)
1024	3.045
2048	6.570
3072	9.806
4096	13.851
5120	17.284

Fig. 4. AES Computational Times

In the same paper, however, they proposed a method to improve the AES algorithm through the use of Shift Row and S. Box modification for Mix Column transformation. Shift Row is a circular process that shifts bytes in each row of a matrix by a certain offset, which is determined by the encryption algorithm [20]. For Rizky Riyaldhi, Rojali, and Aditya Kurniawan's paper, the shift row process started from the 2nd row to the 4th row, as shown in Figure 5.

The result of this showed that the computational time of AES was reduced by 3 milliseconds, as shown in Figure 6.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
value	0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11

Fig. 5. Shift Row

Encryption				Decryption		
Bytes of Data	Before	After	Percentage Optimization	Before	After	Percentage Optimization
1024	3.045	0.463	84.795%	4.121	3.893	5.542%
2048	6.570	0.920	85.997%	8.260	7.604	7.931%
3072	9.860	1.351	86.223%	12.480	10.511	15.775%
4096	13.851	1.851	86.636%	16.440	13.785	16.151%
5120	17.284	2.236	87.063%	19.977	15.976	20.026%
Average All:			86.143%	Average All:		13.085%

Fig. 6. AES Improvement

III. Asymmetric Encryption

Asymmetric encryption, also known as public-key encryption, is a form of data encryption where the encryption key (the public key) and the corresponding decryption key (the private key) are different. This paper will look at a specific asymmetric encryption algorithm, the Rivest–Shamir–Adleman algorithm. The Rivest-Shamir-Adleman (RSA) algorithm, was designed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978. The asymmetry of RSA is based on the practical difficulty of the factorization of the product of two large prime numbers, named "prime factorisation" [16]. RSA is based on number theory, which is a block cipher system. It is one of the most widely known public key cryptosystems and is used for key exchange, digital signatures, or encryption of blocks of data. RSA uses a variable size encryption block and a variable size key. The most notable advantage of the RSA algorithm is that since it's a public-key cipher, it makes it easier to to solve the fundamental problem of cryptography - that is, how to safely distribute keys. Public cryptography solved this dilemma by having two different keys - one key for encryption and one key for decryption. To generate the public keys (for encryption) and private keys (for decryption), it uses two prime numbers. Figure 7 shows the basic operation of the Rivest-Shamir-Adleman (RSA) algorithm:

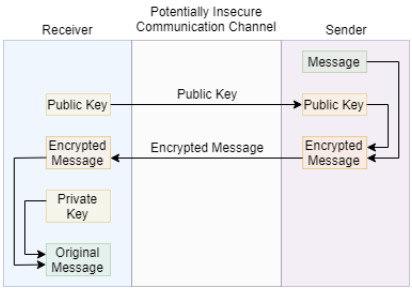

Fig. 7. RSA Process

The Sender encrypts the message using the Receiver's public key. Once the message has been successfully transmitted to the Receiver, the Receiver can then decrypt the message using their own private key. When RSA was introduced, it implemented two important principles: public-key encryption and digital signatures [12]. The RSA operation(s) can be broken down into three broad steps: key generation, encryption, and decryption. While RSA has its advantages, it has many flaws and is subsequently not suitable or preferred for commercial use. As discussed in Gurpreet Singh and Supriya's article, "*A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security*", if two values p and q (where p and q are distinct prime numbers such that $p \neq q$) are too small when designing the key, then the encryption process becomes too weak and one would be able to decrypt the data by using random probability theory and side channel attacks. Conversely, if p and q are too big when designing the key, the encryption process consumes more time and the performance gets degraded in comparison to DES [18].

IV. Future of Data Encryption and Possible Replacements of Existing Encryption Algorithms

There are several different techniques that could possibly be employed in the future to further enhance our ability to encrypt and decrypt our data. Some of these have been and still are being extensively researched and tested. Some examples include, but are not limited to: Elliptic Curve Cryptography (ECC) [7], Quantum Computation [4], and a data encryption algorithm which is based on the location of mobile users which has been given the name, "Location-Dependent Data Encryption Algorithm (LDEA)" [9] has also been proposed.

A. *Elliptic Curve Cryptography (ECC)*

Elliptic-curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. In Nicholas G. McDonald's research review, "*Past, Present, and Future Methods of Cryptography and Data Encryption*" [11], he discussed in brief Elliptic Curve Cryptography. As he stated, Elliptic Curve Cryptography has technically already been invented but he considered it to be a future technique of cryptography because its advantages and disadvantages are not yet fully understood. Figure 8 shows an example of an Elliptic Curve.

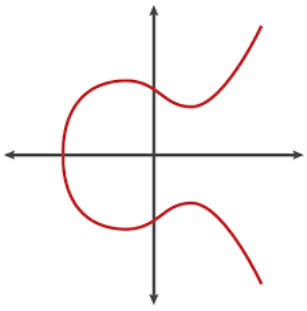

Fig. 8. An example of an Elliptic Curve

In terms of encryption, ECC is an approach that utilizes the complex nature of elliptic curves in finite fields. A finite field (also called a Galois field, so named in honor of Évariste Galois, a French mathematician) in mathematics is simply a field that contains a finite number of elements. While RSA encryption and ECC typically use the same types of

algorithms, the difference is that the numbers used are chosen from a finite field defined within an elliptic curve expression.

B. *Quantum Cryptography*

Integer factorization is thought to be computationally infeasible with an ordinary computer for large integers if the integers are the product of few prime numbers. Integer factorization is the breaking down of a larger, positive integer into a product of smaller integers. In Nicholas G. McDonald's paper, he also briefly discussed the possibility of using Quantum Computation as a future method of data encryption [11]. Quantum Computation is performed in a quantum computer or processor, which is a processor that makes use of quantum mechanical phenomena, such as quantum superposition and quantum entanglement. Quantum superposition is a fundamental principle of quantum mechanics. It states that any two (or more) quantum states can be added together i.e. superimposed, and the result will be another valid quantum state. Using quantum logical qubit state, which is quantum superposition of the "basis states" - 0 and 1 as an example - a superposition is where a qubit can be in the state 0 and 1 at the same time, contrary to a classical bit which can only be in the state corresponding to state 0 or the state corresponding to state 1. Depending on the quantum design, each qubit can store a set number values simultaneously. Quantum computing is still in its early stages and, as a result, quantum processors manufactured today are very small and do not have the computational size that transistor processors have. The fear is that a successful and practical quantum computer would threaten the world's financial system by breaking every encryption system known and in use - public-key cryptography relies on computers being slow to compute discrete logarithms and prime factorizations. Figure 9 shows a new quantum-cryptography scheme, Random Oblivious Transfer (ROT), that would allow for secure anonymous transactions.

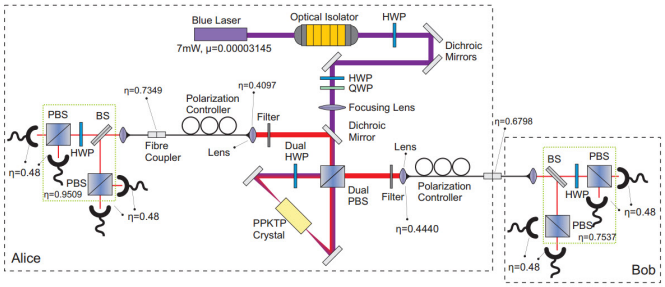

Fig. 9. Random Oblivious Transfer (ROT)

C. *Location-Dependent Data Encryption Algorithm (LDEA)*

As proposed in Hsien-Chou Liao and Yun-Hsiang Chao's 2008 study, "*A New Data Encryption Algorithm Based on the Location of Mobile Users*" [9], most of the data encryption technology today is location-independent. That is, encrypted data can be decrypted anywhere. Today's encryption technology is unable to restrict the location at which data is decrypted. Hsien-Chou Liao and Yun-Hsiang Chao proposed the idea of a location-dependent approach, named Location-Dependent Data Encryption Algorithm (LDEA), as shown in Figure 10.

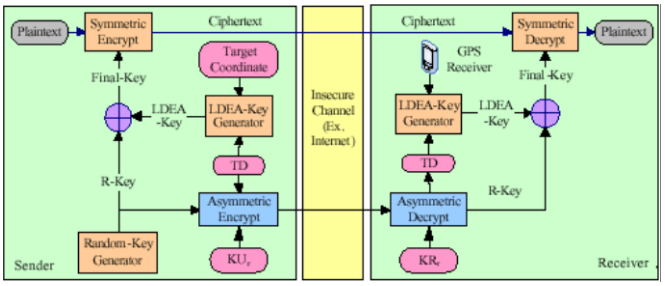

Fig. 10. Location-Dependent Data Encryption Algorithm (LDEA)

For this approach to work, a target latitude and longitude co-ordinate is first determined. The co-ordinate is then incorporated, with a random key, for data encryption. As a result, the receiver can only decrypt the cipher text when the co-ordinate acquired from the GPS receiver is matched with the target co-ordinate. A number of drawbacks to modern technology, such as GPS, renders LDEA difficult to implement well. The location of a mobile user is difficult to exactly match with the target co-ordinate due to current GPS technology being inaccurate and inconsistent.

In Hatem Hamad and Souhir Elkourd's research paper, "*Data encryption using the dynamic location and speed of mobile node*", they proposed a similar idea to that of the one proposed by Hsien-Chou Liao and Yun-Hsiang Chao in their paper. Figure 11 shows the generation process of the secret key proposed in Hatem Hamad and Souhir Elkourd's research paper.

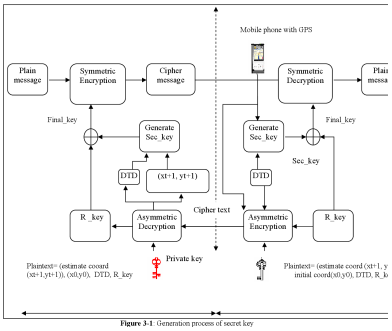

Fig. 11. Generation process of secret key

They proposed a method of key security where the receiver Mobile Node (MN) registers some co-ordinates and speed during travel and thus can map the course of movement [6]. Through this path, one can then predict the next co-ordinate expected after a certain amount of time (since the GPS receiver is inaccurate and inconsistent) depending on how many satellite signals are received, as mentioned in Hsien-Chou Liao and Yun-Hsiang Chao's paper. However, as stated above, Hsien-Chou Liao and Yun-Hsiang Chao's protocol isn't strong enough due to them using the static location (the longitude and latitude co-ordinates) of the mobile node. To overcome the inaccuracy and inconsistency of a GPS receiver,

Hsien-Chou Liao and Yun-Hsiang Chao use a static Tolerance Distance (TD). In contrast, Hatem Hamad and Souhir Elkourd applied a dynamic location of the mobile node and dynamic TD which greatly improves the security of the protocol [6].

V. Conclusion

This paper reviewed data encryption in regards to three main areas: the history of data encryption, the current state of data encryption, where this paper looked at three specific forms of data encryption algorithms - the Data Encryption Standard (DES), the Rivest–Shamir–Adleman algorithm (RSA), and DES' successor the Advanced Data Encryption Standard (AES). From this, it is clear that data encryption is a highly important aspect for today's technology and is widely researched in the area of Computer Science. Along with this, it is also evident from this paper that the field of data encryption will continue to progress and grow exponentially with new methods of data encryption, such as Elliptic Curve Cryptography and Location-Dependent Data Encryption Algorithm (LDEA), are being continually researched and tested. As Quantum Computing continues to grow in popularity and accessibility, this also opens up the possibility of different avenues of data encryption being explored and potentially implemented in the coming years.

References

[1] Alex Biryukov and Christophe De Cannière. *Data encryption standard (DES)*. Springer-Boston, Massachusetts, 2005. ISBN: 9780387234731.

[2] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag, Berlin, 2002. ISBN: 9783642076466.

[3] Niels Ferguson et al. *Improved Cryptanalysis of Rijndael*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 213–230. ISBN: 9783540447061. DOI: https://doi.org/10.1007/3-540-44706-7_15.

[4] K.A.G Fisher et al. "Quantum computing on encrypted data". In: *Nature Communications* 5.3074 (2014), pp. 1–7. DOI: https://doi.org/10.1038/ncomms4074.

[5] Helen F. Gaines. *Cryptanalysis: A Study of Ciphers and Their Solution*. 2014. URL: https://books.google.ie/books?hl=en&lr=&id=Zb2RBQAAQBAJ&oi=fnd&pg=PP1&dq=related:J8ExJoCI0EoJ:scholar.google.com/&ots=ycXrhmfX-2&sig=0I7MAvMu7XzmOWATbZmLwZeuHsM&redir_esc=y#v=onepage&q&f=false.

[6] Hatem Hamad and Souhir Elkourd. "Data encryption using the dynamic location and speed of mobile node". In: *Journal Media and Communication Studies* 2.3 (Apr. 2010), pp. 67–75.

[7] Darrel Hankerson, Alfred Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer Boston MA, 2011, pp. 397–397. ISBN: 9781441959065. DOI: https://doi.org/10.1007/978-1-4419-5906-5. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.394.3037&rep=rep1&type=pdf.

[8] Patrick T. Kenekayoro. "The data encryption standard thirty four years later: An overview". In: *African Journal of Mathematics and Computer Science Research* 3.10 (Oct. 2010), pp. 267–269.

[9] Hsien-Chou Liao and Yun-Hsiang Chao. "A New Data Encryption Algorithm Based on the Location of Mobile Users". In: *Information Technology Journal* 7.1 (2008), pp. 63–69. DOI: https://scialert.net/abstract/?doi=itj.2008.63.69.

[10] Mitsuru Matsui. *Linear Cryptanalysis Method for DES Cipher*. Springer Berlin Heidelberg, 1994, pp. 386–397. ISBN: 9783540482857. DOI: https://doi.org/10.1007/3-540-48285-7_33.

[11] Nicholas G. McDonald. *Past, Present, and Future Methods of Cryptography and Data Encryption*. URL: https://my.eng.utah.edu/~nmcdonal/Tutorials/EncryptionResearchReview.pdf.

[12] Evgeny Milanov. *The RSA Algorithm*. URL: https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf.

[13] Davood Rezaeipour and Mohamad Rushdan Md Said. "The block cipher algorithm-properties, encryption efficiency analysis and security evaluation". In: *Advances and Applications in Mathematical Sciences* 4.2 (Jan. 2010), pp. 129–137.

[14] Vincent Rijmen and Joan Daemen. *AES submission document on Rijndael*. URL: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf#page=1.

[15] Rizky Riyaldhi, Rojali, and Aditya Kurniawan. "Improvement of Advanced Encryption Standard Algorithm With Shift Row and S.Box Modification Mapping in Mix Column". In: *Procedia Computer Science* 116 (2017), pp. 401–407. DOI: https://doi.org/10.1016/j.procs.2017.10.079.

[16] Yaroslav Sergeyev. "The Difficulty of Prime Factorization is a Consequence of the Positional Numeral System". In: *International Journal of Unconventional Computing* 12.5–6 (2016), pp. 453–463.

[17] Gustavus J. Simmons. "Symmetric and Asymmetric Encryption". In: *ACM Computing Surveys (CSUR)* 11.4 (Dec. 1979), pp. 305–330.

[18] Gurpreet Singh and Supriya. "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security". In: *International Journal of Computer Application* 67.19 (2013), pp. 33–38.

[19] Miles E. Smid and Dennis K. Branstad. "Data Encryption Standard: past and future". In: *Proceedings of the IEEE* 76.5 (1988), pp. 550–559. DOI: https://doi.org/10.1109/5.4441.

[20] Harold F. Tipton. *Official (ISC)2 Guide to the CISSP CBK*. CRC Press, 2009, p. 347. ISBN: 9781439809600. DOI: https://books.google.ie/books?id=9gCn86CmsNQC&pg=PA347&dq=shift+row+transformation&hl=en&sa=X&ved=0ahUKEwjwj8HOcbmAhXTTcAKHQDBCskQ6AEIKTAA#v=onepage&q=shiftrowtransformation&f=false.

[21] M.J. Wiener. “Cryptanalysis of short RSA secret exponents”. In: *IEEE Transactions on Information Theory* 36.3 (1990), pp. 553–558. DOI: https://doi.org/10.1109/18.54902.

[22] Paul Zande. *The Day DES Died*. URL: https://www.sans.org/reading-room/whitepapers/vpns/day-des-died-722.

[1] [19] [10] [8] [22] [2] [15] [3] [14] [12] [21] [18] [16] [13] [17] [5] [20] [11] [9] [4] [6] [7]