Hybrid Cryptosystem of Spritz Symmetric Algorithm and RSA Dual Modulus Algorithm in Digital File Security

Sarwedi Harahap[1*], Budi Triandi[2], Dedy Hartama[3]
[1]Master of Computer Science, Faculty of Engineering and Computer Science, Universitas Potensi Utama, Medan. Indonesia
[2]Department of Computer Science, Faculty of Engineering and Computer Science, Universitas Potensi Utama, Medan. Indonesia
[3]Department of Computer Science, Faculty of Engineering and Computer Science, Sekolah Tinggi Tunas Bangsa, Pematang Siantar. Indonesia
[*]Corresponding author: Wedhyharahap95@gmail.com

Abstract - Amid the rapid development of the digital era, data protection has become very important for individuals and organizations. Digital data often contains sensitive and high-value information that needs to be protected from illegal access and cyber threats. Cryptography can be classified into two types based on the keys used, namely symmetric cryptography and asymmetric cryptography. Spritz is an asymmetric stream cipher algorithm with encryption and decryption speed but only guarantees the security of keys 128 bits long. Dual Modulus RSA is a variant of the RSA algorithm that offers a higher level of protection compared to Spritz because of the difficulty of factoring large prime numbers during key generation. Still, the Dual Modulus RSA algorithm requires a long time in the process of encrypting and decrypting messages. So this research offers a hybrid cryptography system where the spritz algorithm is used for the message encryption process and the spritz key that is generated is encrypted using a Dual modulus RSA public key so that the key used has double security where the key generated is bigger and more random than the key previous spritz. The Dual Modulus RSA algorithm is used to decrypt the cipher key which will then produce a key spritz. This research shows that using a hybrid cryptography system can prevent the possibility of data theft because the keys used are double-secured and the encryption process is relatively faster.

Keywords: ***Cryptography; Hybrid; Spritz; Dual Modulus RSA; Python.***

1. INTRODUCTION

In the ever-growing digital era, data security is a top priority for individuals and organizations. Digital data often contains sensitive and high-value information that must be protected from unauthorized access and cyber threats [1]. Cryptography is a science that studies the art or method of guaranteeing a message so that the message cannot be interpreted or cannot be understood by other parties who do not have authority over it[2]. There are four objectives of cryptography which is also an aspect of information security, namely confidentiality, data integrity, authentication, and non-repudiation [3]. Cryptography can be classified into two types based on the keys used, namely symmetric cryptography and asymmetric cryptography. Symmetric cryptography is a cryptographic algorithm that uses the same key in the encryption and decryption process [4]. Communicating entities must exchange keys so they can be used in the decryption process. The secret key used by the sender and receiver can be a random series of letters and numbers. Examples of symmetric algorithms: Spritz algorithm, Rabbit Stream, RC4, TwoFish, Rijndael, etc [5]. Asymmetric cryptography is an algorithm that uses different keys for the encryption and decryption processes [6]. Asymmetric encryption algorithms include RSA, ECC, DSA, ElGamal, etc [7].

In asymmetric cryptography, one of the keys is published, and knowing the public key will provide an opportunity for unauthorized people to decipher all the keys, although this requires a long process [8]. The advantage of using asymmetric algorithms is that they provide greater scalability than symmetric algorithms and guarantee confidentiality and authentication, but asymmetric algorithms work much slower than symmetric ones. Using symmetric algorithms can often be easily cracked by cryptanalysts because the security of symmetric algorithms only depends on the confidentiality of the key. If the key used can be guessed or known by an irresponsible party, then all messages can be easily decrypted [9].

Spritz is a stream cipher algorithm designed by Ronald Rivest and Jacob Schuldt as a successor to RC4, which is an algorithm that encrypts messages one character at a time using a short time-dependent encryption transformation[10]. The process of initializing the Spritz Algorithm uses two substitutions Key Scheduling Algorithm (KSA) and Pseudo-Random Generation Algorithm (PRGA)[11]. The main advantage of the alma Spritz is high encryption and decryption speed, but there is also a weakness of this spritz, namely that it only guarantees a key of 128 bits [12].

Dual Modulus RSA is a variant of the RSA algorithm that aims to increase security through the use of two public keys and two private keys[13]. This concept is based on the basic principles of RSA, which involves factors of two large prime numbers to generate encryption and decryption keys. By introducing two key pairs, Dual RSA offers a higher level of security compared to Spritz, as it requires solving two independent factorization problems to solve the system [14].

The Hybrid Cryptosystem scheme is the process of generating a symmetric key and the key will be encrypted with an asymmetric key and uses the Spritz Algorithm and Dual RSA [15]. Spritz carries out the text security process, and then the key from Spritz will be secured using Dual RSA to strengthen the text document digital file security system. A hybrid Cryptosystem is used because it utilizes data processing speed with a symmetric algorithm and easy key transfer using an asymmetric algorithm to increase the speed of data transaction processing, faster processing, and safer because of the layered security of the key [16].

2. METHODS

In this research, the author uses a varied text data set. The data is in the form of text files with the extensions .txt and .docx which are saved in ASCII format. Data sources were obtained from various scientific publications, such as journals, books, and conference proceedings.

As a first step, this research conducted an in-depth study of the Spritz and Dual Modulus RSA algorithms. A comprehensive understanding of these two algorithms is the basis for designing stronger hybrid encryption schemes. At this stage, the researcher first carried out a manual calculation process for the two algorithms, where later the two algorithms were hybridized.

A. Spritz algorithm

In this research, the Spritz algorithm is used to encode the text that will be sent, which will later become ciphertext. The process in this spritz algorithm is:

1. Key Scheduling Algorithm (KSA)

The Key Scheduling Algorithm or key scheduling algorithm is used to create an S-Box table (S array) and permutation tables in the S array. The required array length is 256 starting from index 0 to 255. The goal of KSA is to have a permutation array process with as many values as 256 times initialized with variables i and j with integer type.

The KSA formula is:

```
for i = 0 to N – 1 S[i] = i
then i i, j = 0
for i = 0 to N – 1
j = ( j + S[i] + K[i mod Key.length]) mod N
swap ( S[ i ], S[ j ] )
j = j next i
```

where N is the size of the array to be mutated, e.g. 0 - 255.

2. Pseudo-Random Generation Algorithm (PRGA)

Pseudo-Random Generation Algorithm (PRGA) or randomization is carried out to obtain a new key with a number of plain elements. The W value is a new variable added to the Spritz algorithm according to the RC4 algorithm. The values of the variables i, j, k, and z start from 0 and will change according to the results of each iteration. This process involves the S array values that have been allowed in the KSA process.

The PRGA formula is:

```
For i = 0 to plain.length
i = (i + w) mod N
j = (k + S [j + S [i]]) mod N k = (i + k + S [j]) mod N swap S [i], S [j]
z = (S [j + S [i + S [z + k]]]) mod N
output z next i where w is an integer value relatively prime with N and the values i, j, k, z start from 0.
```

3. Encryption and Decryption.

Encryption

Each byte of plaintext is XORed with a byte from the keystream to produce the ciphertext.

B. AlgorithmDual Modulus RSA

Dual RSA is a variation of the RSA encryption algorithm that uses two different pairs of keys (two pairs of public and private keys). Following are some of the advantages and disadvantages of Dual RSA.

Stages of the Dual Modulus RSA algorithm:

Key generation
Choose a prime number p1,p2,q1,q2
Calculate value ni = pi * qi
Calculate the quotient □(ni) = (pi-1)*(qi-1)
Determine the number e1 and e2 with the provision of : gcd (□(n1),e1 = 1 and gcd (□(n2),e2 = 1
Determine a random number d1 dan d2 with the provision of: d1.e1 mod □(n1) = 1 and d2.e2 mod □(n2)=1

Public key (n1, n2, e1, e2)
Private key (d1,d2)

Encryption

C = 〖〖(m〗^e1 mod n1)〗^n1 mod n2

Decryption

m = 〖〖(c〗^d2 mod n2)〗^d1 mod n1

Hybrid Cryptography

Hybrid Cryptography is a cryptographic technique that combines the advantages of two types of cryptographic methods, namely symmetric cryptography and asymmetric cryptography. In symmetric cryptography, the encryption and decryption processes use the same key, which makes it very efficient and fast in processing large amounts of data. However, challenges the main thing about symmetric cryptography is how to secure the key distribution. Meanwhile, asymmetric cryptography uses a pair of keys, namely a public key for encryption and a private key for decryption, which provides a higher level of security. However, the encryption and decryption process with asymmetric cryptography tends to be slower than with symmetric cryptography. To overcome the shortcomings of each of these techniques, hybrid cryptography combines these two methods in one more secure and efficient system.

Asymmetric cryptography is used to secure the exchange of symmetric keys between the sender and receiver. Once symmetric keys have been securely exchanged, symmetric cryptography is used to encrypt the data, allowing for fast encryption and decryption. Thus, hybrid cryptography provides an optimal solution by maintaining the security of key exchange through asymmetric cryptography, while still utilizing the speed of symmetric cryptography for data processing.

Hybrid cryptography steps :

1. The plaintext is encrypted using a symmetric private key to produce ciphertext
2. The Spritz key is encrypted with an asymmetric public key to produce a cipher key
3. So the ciphertext and cipher key will be sent to the recipient
4. After the recipient gets the ciphertext and cipher key, the decryption process will be carried out
5. The cipher key is decrypted with an asymmetric private key so that it produces an asymmetric key
6. The ciphertext is decrypted using the asymmetric key at point five

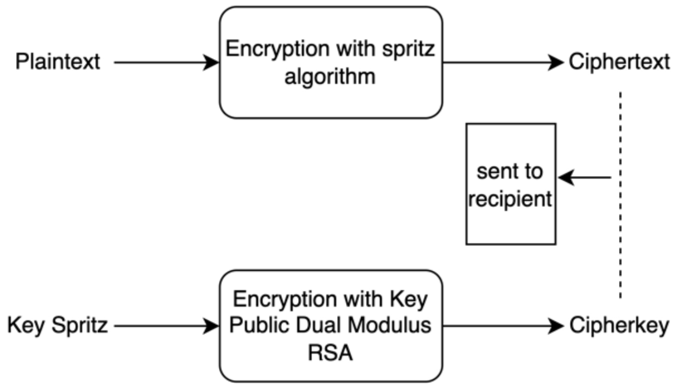

Plaintext
Encryption with spritz algorithm
Ciphertext
sent to recipient
Key Spritz
Encryption with Key Public Dual Modulus RSA
Cipherkey

Figure 1. cryptographic hybrid encryption process

3. RESULTS AND DISCUSSION

The aim of this research is to minimize attacks on the key spritz and speed up the encryption and decryption process by combining the spritz algorithm and Dual Modulus RSA. The implementation of this hybrid cryptographic system is carried out using the Python programming language on hardware with the following specifications:

Processor	: 2.9 GHz Intel Core i5
Memory	: 8 GB 1600 MHz, DDR3
SSD	: 500 GB
OS	: Windows

In this research, we will test the hybrid cryptographic system that has been developed. Testing will be carried out through manual calculations to understand the basic mechanisms, as well as through program implementation using the Python language to test system performance and efficiency.

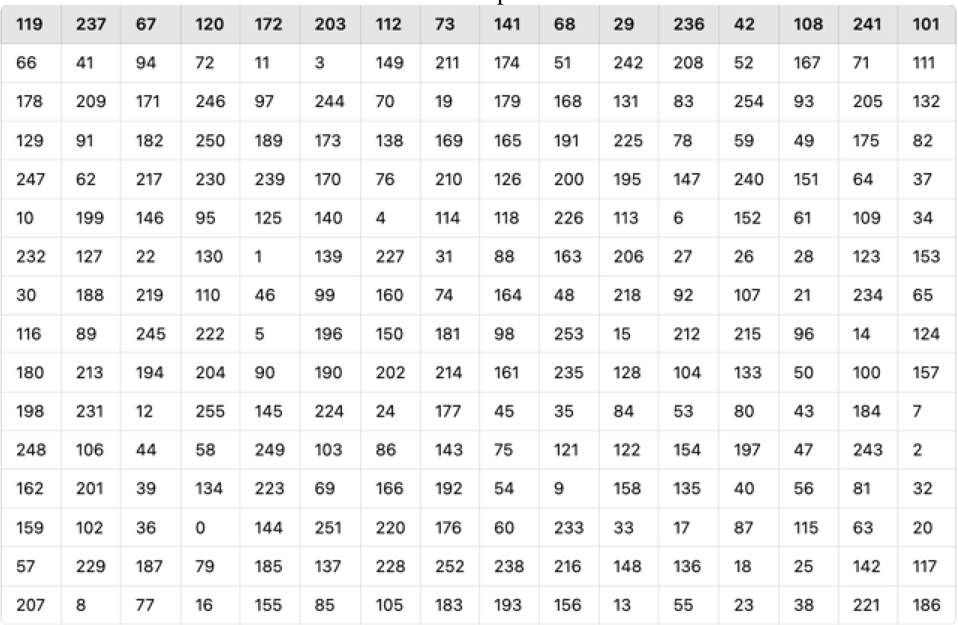
Data Encryption with the Spritz Algorithm In this research, the spritz algorithm is used to carry out the encryption process for plaintext data that has been converted into decimal format. Examples of keys and plaintext used in this research can be seen in Table 1 below:

Table 1. Sample Data

index	0	1	2	3	4	5	6
Plantext(char)	s	a	r	w	e	d	i
Plaintext (Decimal)	115	97	114	119	101	100	105
Key(char)	w	e	d	2	0	2	5
Key (Decimal)	119	101	100	50	48	50	53

1. The first step that must be taken is to calculate the first value of j with the initial value
   i=0
   j=0
   j = (j + S[i] + key[i mod keylength]) mod 256
   = (0 + S[0] + key [0 mod 6] mod 256
   = (0 + s[0] + key[0]) mod 256
   =(0+0+119) mod 256
   = 119 mod 256
   j = 119
   value s[i] = s[0] swap with value s[j] = s[119]
   so s[0] = s[119] and value s[50] = s[0]
   Do the calculation again as above 256 times to produce the KSA table:

Table 2. KSA Spritz Results



119	237	67	120	172	203	112	73	141	68	29	236	42	108	241	101
66	41	94	72	11	3	149	211	174	51	242	208	52	167	71	111
178	209	171	246	97	244	70	19	179	168	131	83	254	93	205	132
129	91	182	250	189	173	138	169	165	191	225	78	59	49	175	82
247	62	217	230	239	170	76	210	126	200	195	147	240	151	64	37
10	199	146	95	125	140	4	114	118	226	113	6	152	61	109	34
232	127	22	130	1	139	227	31	88	163	206	27	26	28	123	153
30	188	219	110	46	99	160	74	164	48	218	92	107	21	234	65
116	89	245	222	5	196	150	181	98	253	15	212	215	96	14	124
180	213	194	204	90	190	202	214	161	235	128	104	133	50	100	157
198	231	12	255	145	224	24	177	45	35	84	53	80	43	184	7
248	106	44	58	249	103	86	143	75	121	122	154	197	47	243	2
162	201	39	134	223	69	166	192	54	9	158	135	40	56	81	32
159	102	36	0	144	251	220	176	60	233	33	17	87	115	63	20
57	229	187	79	185	137	228	252	238	216	148	136	18	25	142	117
207	8	77	16	155	85	105	183	193	156	13	55	23	38	221	186

1. The next step is to PRGA as many as thenumber of plaintext lengths so that it can produce a z-value
2. Initialize initial values

   i = 0
   j = 0
   k = 0
   z = 0
   w = 101 (relatively prime to 256)
   256 mod 101 = 54
   101 mod 54 = 47
   54 mod 47 = 7
   47 mod 7 = 5
   7 mod 5 = 2

5 mod 2 = 1
2 mod 1 = 0
So 101 relatively prime to 256 because GCD = 1
i = i+w mod 256
= 0 + 101 mod 256
i = 101
j = k + S[j + S[i]]
= 0 + s[0 + s[101] mod 256
= 0 + s[0 + 139] mod 256
= s[139] mod 256
= 212 mod 256
= 212
k = i + k + S[j]
= 101 + 0 +S[212] mod 256
= 101 + 0 + 144 mod 256
= 245

S[i] , S[j] = s[i] = s[101] = 139
S[j] = s [212] = 144

Swap s[i] dan s[j] = s[i] = s[101] = 144
S[j] = s[212] = 139

z1 = S[j + S[i + S[z + k]]] mod 256
= s[212 + s[101 + s[0+245]] mod 256
= s[212 + s [101 +s[245] mod 256
= s[212 + s[101+ 85] mod 256
= s[212 + s[186] mod 256
= s[212 + 122] mod 256
= s[334 mod 256
= s[78]
z1 = 64

Carry out the process of determining the z value repeatedly as above throughout the plaintext, the key value for each plaintext character is then obtained after going through the Pseudo-Random Generation Algorithm (PRGA) step, and the values

The z is the key to the sprit which will then be converted to binary along with
The text must be encrypted using XOR.

Table 3. Plaintext Encryption Process

P	Z		Ciphertext
01110011	01000000	Xor	00110011
01100001	00110010		01010011
01110010	01001000		00111010
01110111	00100111		01010000
01100101	01010011		00110110
01100100	11101011		10001111
01101001	11000011		10101010

Spritz Key Encryption with DM RSA

The hybrid uses the Spritz key for the encryption process to produce a cipher key. The cipher key will later be sent to the recipient as part of the data security process. This process is designed to ensure stronger security in protecting critical information. The main purpose of using this method is to provide an additional layer of security for the key. This aims to minimize the risk of leaking keys and data that will be sent during the communication process. The encryption process and results can be seen in Table 4 below.

Table 4. Spritz key encryption process

Key	Encryption
	e1 = 44599
	e2 = 291371
	n1 = 109093
	n2 = 541109
	$c = (m^{e1}\ mod\ n1)^{e2}\ mod\ n2$
		Cipherkey

64	$c = (4^{44599}\ mod\ 109093)^{291371}\ mo$	186035
50	$c = (50^{44599}\ mod\ 109093)^{291371}\ m$	319291
72	$c = (72^{44599}\ mod\ 109093)^{291371}\ m$	496422
39	$c = (39^{44599}\ mod\ 109093)^{291371}\ m$	477140
83	$c = (83^{44599}\ mod\ 109093)^{291371}\ m$	70405
235	$c = (235^{44599}\ mod\ 109093)^{291371}\ r$	175284
195	$c = (195^{44599}\ mod\ 109093)^{291371}\ r$	102603

Cipher key Decryption Process

Once the message is received from the sender, the recipient will start the decryption process. This process is an important step to ensure that the message received can be returned to its original form and understood by the recipient. Before the ciphertext can be decrypted, the recipient must first decrypt the cipher key that has been sent. This cipher key is encrypted using the Dual Modulus RSA algorithm, so the recipient must use the appropriate private key to open the cipher key. The decryption result of the cipher key is the Spritz private key, which will later be used to decrypt the ciphertext. This private key spritz is an important element in the decryption process, as shown in Table 3.5 With this key, the recipient can access the contents of the encrypted message and recover the plaintext.

Table 5. Cipherkey decryption

Cipher key	Decryption
	d1 = 71971
	d2 = 3491
	n1 = 109093
	n2 = 541109
	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$
		Key Spritz

186035	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$	64
319291	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$	50
496422	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$	72

477140	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$	39
70405	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$	83
175284	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$	235
102603	$P = (c^{d2}\ mod\ n2)^{d1}\ mod\ n1$	195

Ciphertext Decryption with the Spritz Algorithm

At this stage, the ciphertext received by the recipient will be processed using the spritz algorithm to carry out decryption. This process aims to return the ciphertext to its original form so that the contents of the previously encrypted message can be read and understood by the recipient.

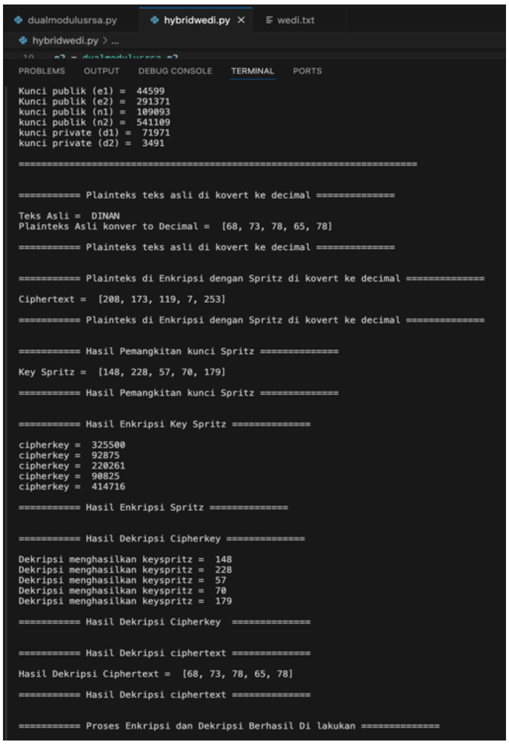
Table 6. Decryption ciphertext

Ciphertext	Z		Plainteks	
00110011	01000000	Xor	01110011	s
01010011	00110010		01100001	a
00111010	01001000		01110010	r
01010000	00100111		01110111	w
00110110	01010011		01100101	e
10001111	11101011		01100100	d
10101010	11000011		01101001	i

Hybrid Program Results of the Spritz Algorithm and Dual Modulus RSA

At this stage, the results of the implementation and testing of the algorithms carried out using the Python programming language version 3.9.6 are explained. This implementation process aims to test the effectiveness of the algorithm used in securing messages and ensuring data integrity. Testing is carried out using certain test data, where one example of the main parameters is the key and plaintext used.

Key	: wed2025
Plaintext	: DINAN



```
Kunci publik (e1) =  44599
Kunci publik (e2) =  291371
kunci publik (n1) =  109093
kunci publik (n2) =  541109
kunci private (d1) =  71971
kunci private (d2) =  3491

======================================================================


=========== Plainteks teks asli di kovert ke decimal ==============

Teks Asli =  DINAN
Plainteks Asli konver to Decimal =  [68, 73, 78, 65, 78]

=========== Plainteks teks asli di kovert ke decimal ==============


=========== Plainteks di Enkripsi dengan Spritz di kovert ke decimal ==============

Ciphertext =  [208, 173, 119, 7, 253]

=========== Plainteks di Enkripsi dengan Spritz di kovert ke decimal ==============


=========== Hasil Pemangkitan kunci Spritz ==============

Key Spritz =  [148, 228, 57, 70, 179]

=========== Hasil Pemangkitan kunci Spritz ==============


=========== Hasil Enkripsi Key Spritz ==============

cipherkey =  325500
cipherkey =  92875
cipherkey =  220261
cipherkey =  90825
cipherkey =  414716

=========== Hasil Enkripsi Spritz ==============


=========== Hasil Dekripsi Cipherkey ==============

Dekripsi menghasilkan keyspritz =  148
Dekripsi menghasilkan keyspritz =  228
Dekripsi menghasilkan keyspritz =  57
Dekripsi menghasilkan keyspritz =  70
Dekripsi menghasilkan keyspritz =  179

=========== Hasil Dekripsi Cipherkey  ==============


=========== Hasil Dekripsi ciphertext ==============

Hasil Dekripsi Ciphertext =  [68, 73, 78, 65, 78]

=========== Hasil Dekripsi ciphertext ==============


=========== Proses Enkripsi dan Dekripsi Berhasil Di lakukan ==============
```

Figure 2. Program Encryption and Decryption Results

4. CONCLUSION

The results of analysis and testing prove that the applied approach can increase security and efficiency in the data encryption and decryption process. The encryption results of the key spritz by utilizing Dual Modulus RSA produce keys with a high level of randomness, making it very difficult for attackers to reconstruct the original message without knowing the corresponding decryption key Manual implementation and using Python of the spritz and Dual Modulus RSA hybrid algorithm produces the same encryption and decryption output without adding or subtracting the meaning of the message sent or received. combining the two algorithms produces a new key that is larger and more random, thereby minimizing the occurrence of key leaks.

ACKNOWLEDGMENT

The authors would like to thank the Master of Computer Science Program for its support and sponsorship. And we also thank the research collaboration of Universitas Potensi Utama and Sekolah Tinggi Tunas Bangsa for supporting this research.

REFERENCES

[1] D. Sarumaha, M. Mardiah, S. Amir, A. Gafur, and I. Zega, "Penerapan Algoritma Enhanced Dual Rivest Shamir Adleman untuk Pengamanan Data," Digital Transformation Technology, vol. 4, no. 1, pp. 34–41, Mar. 2024, doi: 10.47709/digitech.v4i1.3686.

[2] S. M. Naser, "CRYPTOGRAPHY: FROM THE ANCIENT HISTORY TO NOW, IT'S APPLICATIONS AND A NEW COMPLETE NUMERICAL MODEL," 2021. [Online]. Available: www.ea-journals.org

[3] D. Kumar Sharma, N. Chidananda Singh, D. A. Noola, A. Nirmal Doss, and J. Sivakumar, "A review on various cryptographic techniques & algorithms," in Materials Today: Proceedings, Elsevier Ltd, 2021, pp. 104–109. doi: 10.1016/j.matpr.2021.04.583.

[4] Z. Arif and A. Nurokhman, "Analisis Perbandingan Algoritma Kriptografi Simetris Dan Asimetris Dalam Meningkatkan Keamanan Sistem Informasi Comparative Analysis of Symmetric and Asymmetric Cryptographic Algorithms in Improving Information System Security," 2023.

[5] U. Makhsud, "Asymmetric Cryptosystems." [Online]. Available: www.ijeais.org/ijaer

[6] D. Liestyowati, "Public Key Cryptography," in Journal of Physics: Conference Series, Institute of Physics Publishing, 2020. doi: 10.1088/1742-6596/1477/5/052062.

[7] H. Abroshan, "A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms." [Online]. Available: www.ijacsa.thesai.org

[8] B. Halak, Y. Yilmaz, and D. Shiu, "Comparative Analysis of Energy Costs of Asymmetric vs Symmetric Encryption-Based Security Applications," IEEE Access, vol. 10, pp. 76707–76719, 2022, doi: 10.1109/ACCESS.2022.3192970.

[9] X. Dong, D. A. Randolph, and S. K. Rajanna, "Enabling Privacy Preserving Record Linkage Systems Using Asymmetric Key Cryptography."

[10] M. Furqan, R. Kurniawan, A. Halim Hasugian3, I. Zaki, and H. Nst, "http://infor.seaninstitute.org/index.php/infokum/index INFOKUM is licensed under a Creative Commons Attribution-Non Commercial 4.0 International License (CC BY-NC 4.0) Digital Image Security System Using Spritz Algorithm," JURNAL INFOKUM, vol. 10, no. 1, 2021, [Online]. Available: http://infor.seaninstitute.org/index.php/infokum/index

[11] D. Yusvika Yusdartono and H. Muharry Yusdartono, "Instal : Jurnal Komputer The Combination of the Spritz Algorithm and the Bit Plane Complexity Segmentation Technique in Text Security." [Online]. Available: https://journalinstal.cattleyadf.org/index.php/Instal/index

[12] T. Signor et al., "Euclid: Identifying the reddest high-redshift galaxies in the Euclid Deep Fields with gradient-boosted trees," Astron Astrophys, vol. 685, May 2024, doi: 10.1051/0004-6361/202348737.

[13] G. A. Zimbele and S. A. Demilew, "Hidden Real Modulus RSA Cryptosystem," International Journal of Computing, vol. 22, no. 2, pp. 238–247, 2023, doi: 10.47839/ijc.22.2.3094.

[14] A. Datumaya, W. Sumari, I. Annurroni, and A. Ayuningtyas, "The Internet-of-Things-based Fishpond Security System Using NodeMCU ESP32-CAM Microcontroller," vol. 10, no. 1, pp. 51–61, 2025, doi: 10.29207/resti.v9i1.6033.

[15] I. Hidayati, M. A. Budiman, and M. Zarlis, "Analysis of Embedding Locations in the Subband Frequency DCT on Scanned Images," Data Science: Journal of Computing and Applied Informatics, vol. 7, no. 1, pp. 1–14, Jan. 2023, doi: 10.32734/jocai.v7.i1-10359.

[16] D. Sarumaha, M. A. Budiman, and M. Zarlis, "Analysis of Embedding Locations in the Subband Frequency DCT on Scanned Images," Data Science: Journal of Computing and Applied Informatics, vol. 7, no. 1, pp. 1–14, Jan. 2023, doi: 10.32734/jocai.v7.i1-10359.