Three Part Hybrid Encryption Schema

H. GENÇOĞLU, T. YERLİKAYA

Abstract—**Hybrid encryption schemes are consist of two parts. First part for encryting the text with symmetric algorithm called DEM (Data Encryption Mechanism) and the second one is for encrypting the symmetric algorithm key with asymmetric algorithm called KEM (Key Encryption Mechanism). If we think to expand the KEM packet with e-sign, message digest and extra security data for validation and authentication, how will be the encryption of KEM packet and also the performance KEM mechanism. Our study try to answer that question and also offer a new hybrid encryption mechanism for information security.**

Index Terms—**Cryptography, Hybrid Cryptography, Data Security**

I. Introduction

IT IS NEVER safe to do business or personal correspondence on the Internet unless it is possible to protect the information. Information security is provided by eliminating threats such as being listened by another one, change of information, imitation of identity. The basic tool used for securing information security is cryptography. Cryptography is a science that analyzes information security and makes the understandable one incomprehensible. Topics such as confidentiality, reliability, data integrity, authentication, authenticity, and irrefutability are important areas of work for cryptography. [1-4]

Encryption techniques can be used to prevent unauthorized access to data content during data security. By using symmetric or asymmetric algorithms, the encrypted data will be meaningless even if it is captured by unauthorized persons. However, by using asymmetric algorithms and processing the data with the keys of the sender, it is possible to obtain unique values. If these values are related to the data and the sender, the sender of the data will be guaranteed if it is added to the data, which is a digitally signing process.

Nowadays, communication security is not only to secure the data content. It is now important for communication security to associate the sender with the data. It should also be proved that the received data is received without any modification on the recipient's side.

Complex security solutions are generally provided by hybrid encryption mechanisms. Symmetric algorithms only have the ability to encrypt data. They are fast and safe. Key management of asymmetric algorithms is successful. Symmetric algorithms are much slower when compared to symmetric algorithms in encryption processes. Hybrid mechanisms are built to complement the missing aspects of symmetric and asymmetric algorithms.

II. Encryption Algorithms

A. Asymmetric Encryption Algorithms

Asymmetric algorithms use two different keys, one for encryption and the other for decryption. These keys mathematically depend on each other, but one cannot get a key from the other. The decryption key is known only by the recipient, but the encryption key can be known by anyone. The fact that the keys are interlinked and one cannot get a key from the other makes it possible to use asymmetric algorithms rather than encryption. Today, asymmetric algorithms are used for digital signatures. [5]

B. Symmetric Encryption Algorithms

Symmetric algorithms use a single key for encryption and decryption. This key must be kept confidential by both the sender and the recipient, otherwise unauthorized persons can easily decrypt the encrypted data and access the actual data. They have a widespread use, because they perform encryption and decryption fast.[6]

C. Hash Algorithms

Hash algorithms are used to create a strict representation of the given data. Calculated hash value is unique. The smallest change in the actual data substantially changes the hash value. Since they are one-way functions, the actual data cannot be obtained. Because of these features, Hash algorithms play a big role in data integrity processes and in authentication. [14 - 15]

III. Previous Works

One of the important works on this subject was made by Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa and Victor Shoup. In this study, the hybrid mechanism is composed of two modules. It is called DEM Data Encryption Mechanism for the Hybrid Mechanism's encryption of the data itself, and KEM-Key Encapsulation Mechanism for the mechanism managing encryption keys. In principle, the data is

HAKAN GENÇOĞLU, is with Department of Computer Engineering İstanbul Sabahattin Zaim University, Istanbul, Turkey, (e-mail: hakan.gencoglu@izu.edu.tr).

https://orcid.org/0000-0003-2968-1615

TARIK YERLİKAYA, is with Department of Computer Engineering Trakya University , Edirne, Turkey, (e-mail: tarikyer@trakya.edu.tr.).

Manuscript received September 07, 2019; accepted October 08, 2019.
DOI: 10.17694/bajece.616893

encrypted with DEM and the keys are encrypted with KEM. [7]

In the tag-KEM / DEM architecture, when a KEM is generated, first a random key is selected and then this key is encrypted with a tag value. Signing procedures are not available in this architecture.

Fujisaki-Okamoto's KEM-DEM architecture is an improved version of the "Tag-KEM / DEM" architecture. But this architecture does not have signing procedures either.

Another study was conducted by Kerim YILDIRIM and H. Engin DEMİRAY. The previous hybrid mechanisms have been improved in their study called “SİMETRİK VE ASİMETRİK ŞİFRELEME YÖNTEMLERİNE METOTLAR: ÇIRPILMIŞ VE BİRLEŞİK AKM-VKM”. [8]

Three algorithms are mentioned in this study. In the "Scrabbled KEM-DEM" architecture, KEM and DEM constructions were mixed using the receiver's open key and sent to the user in a single locale. It is stated that the purpose of this mixing process is to make sure that the attacker does not know if it works with KEM or DEM. In the cascaded KEM-DEM architecture, which is the other algorithm, the shuffle algorithm was considered to be safer to do with the random key instead of the receiver's open key. However, in both cases, it will be seen that there is no difference between the DEM keys if they are thought to be encrypted and stored in the KEM. In the "Combined KEM-DEM" structure, the server was first logged on and secure communication was established with the "Scrabbled KEM-DEM" architecture. The logon key encryption specified by the server in the logon process was performed as if it were in the other two systems.

Although this work enhances security measures, signing and verification procedures are incomplete. Plain or encrypted text is not signed in the system, only the key that generated the key which is used to encrypt the text is signed. However, the signing of the text instead of the key will be more meaningful. In addition, first a key is generated in the system with a string expression, then the encryption key is generated using this key, and encryption is performed. This process will increase the work load on the system and increase the working time of the system. If it is desired that the encryption key is independent of the sender, the system can generate the key without any string entry and encryption can be performed.

In both cases, data security is provided by encrypting the plain text and obtaining a summary to guarantee that it has not changed. Signaling by the sender to verify the sender of the message is also an important security parameter and it should be used for authentication purposes if the attack techniques are considered to be diverse today.

Another study, "A Novel Idea on Multimedia Encryption using Hybrid Crypto Approach", describes that multimedia files can be used safely without leaving the hybrid algorithm architecture.[9]

In the study named “Dik eşleştirme arayış yöntemi ile hibrit veri sıkıştırma ve optiksel kriptografi ", the processed signal is secured with the hybrid mechanism and transmission is proposed.[10]

In the study "A Password-Protected Secret Sharing Based on Kurosawa-Desmedt Hybrid Encryption", the hybrid mechanism has been used to eliminate defects in the secret sharing schemes. [11]

In another study, implementation solution was presented for hybrid mechanism cloud storage systems. RSA is used as the asymmetric algorithm and AES is used as the symmetric algorithm in the work named "The hybrid encryption algorithm of lightweight data in cloud storage".[12]

An idea is proposed for the hybrid architecture to work on the FPGA in the study called "Implementing a hybrid crypto-coding algorithm for an image on FPGA".[13]

As it is seen, the vast majority of studies on hybrid encryption algorithms focus on the implementation of the architecture or performance enhancement rather than the development of the architecture.

IV. Encrypted Communication

The encrypted communication between two points is made using the keys of the point. There will be 4 keys at each point. These keys are; symmetric algorithm key for encryption of the data, public and private keys of asymmetric algorithm for signing, validation, and security packet encryption, and public key belonging to the receiver.

Create 3 packages during the communication. The first package, the message package, contains encrypted data. The second package, the security package, contains signing and authentication values. The third package is the key package that contains the keys of encrypting message package and the security package.

While there are two parts in classical hybrid cryptography architectures, we have created the architecture in three parts in our study. We designed the security package as a separate package and increased the security parameters to make the security of the message stronger. Because increasing the security parameters increases the packet size, we encrypted the security package with a symmetric algorithm. We have completed the hybrid architecture by encrypting two symmetric keys using asymmetric algorithm.

A. Package –Message (Data) Package

The plain data goes in two processes:

- Encryption process
- Sign the message by the sender

The plain data is encrypted using the symmetric algorithm with the randomly generated key. The encrypted message goes to shuffle algorithm and produces two outputs:

1. Shuffled encrypted message
2. Coefficients after the shuffling operation

The purpose of re-mixing the encrypted message is to make the prediction more difficult against the known text attack against the symmetric algorithms. To make this process meaningful, the coefficients must be sent to the receiver and the mapping must be reversed to determine the original locations of the data blocks. One of the inputs to the security package is these coefficients.

The digest value is calculated by entering the plain data in the digest algorithm. This value is signed using the sender's secret key, and a message signature is generated which means that the message is approved by the user. The message signature is one of the inputs of the security package.

$\delta_{K1}(m) = y$

$Sh(y) = (z, q)$

$q \to \delta_{K2}$

$z \to Message\ Package$

$\delta_{K1} =$ $K1$ Symmetric encryption algorithm using K1 Key

$\delta_{K2} =$ $K2$ Symmetric encryption algorithm using K2 Key

$Sh =$ Shuffle Algorithm

$K1 =$ Encryption Key (Symmetric Algorithm)

$z =$ Shuffled Encrypted Message

$q =$ Coefficient Array

$y =$ Encrypted Message (Message Package)

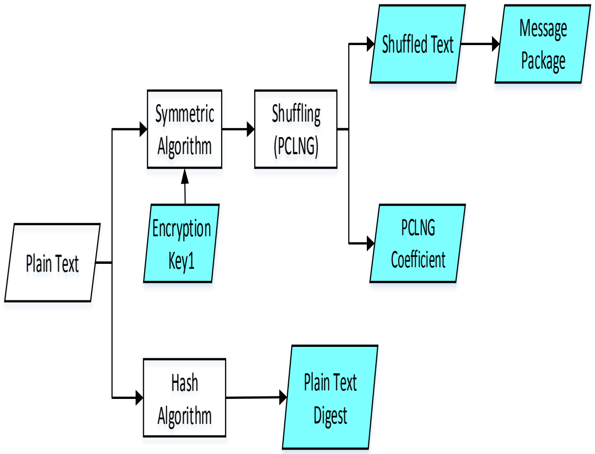

Fig.1. Message Encryption and Message Packet Creation Flowchart

B. Package2 - Security Package

The message signature is one of the inputs of the security package.

The digest of the plain message and the digest of the scrambled message are created in a data packet, first by signing with the device secret key and then by signing with the user secret key. Coefficients of the data packet from shuffle algorithm are also added to the signed data and encrypted with a symmetric algorithm.

A randomly generated key is used for encryption, then this key goes to the package.

$H(m) = h_m$

$H(z) = h_z$

$SG_{SKC}(h_m) = I_{mC}$

$SG_{SKK}(I_{mC}) = I_{mK}$

$SG_{SKC}(h_z) = I_{zC}$

$SG_{SKK}(I_{zC}) = I_{zK}$

$(I_{mC}, I_{mK}) = I_m$

$I_m \to \delta_{K2}$

$q \to \delta_{K2}$

$\delta_{K2}(I_m, q) = \tau$

$\delta_{K2} = K2$ Symmetric encryption algorithm, using K2 Key, pre-security package

$SG_{SK} =$ SK Signature Algorithm using SK Key (Device-Person)

$H =$ Hash Algorithm

$SKC =$ Sender Device’s Secret Key (Asymmetric Algorithm)

$SKK =$ Sender’s Secret Key (Asymmetric Algorithm)

$I_{mC} =$ Signature of the message created by the device

$I_{mK} =$ Signing a device signature for a message created by a person

$I_{zC} =$Signature of the scrambled encrypted message created by the device

$I_{zK} =$ Signing the device signature of the scrambled encrypted message created by the person

$K2 =$ Security Packet Encryption Key (Symmetric Algorithm)

$m =$ Plain Text

$h_m =$ Message Digest

$h_z =$ Summary of scrambled encrypted text

$q =$ Coefficient Array

$\tau =$ Security Package

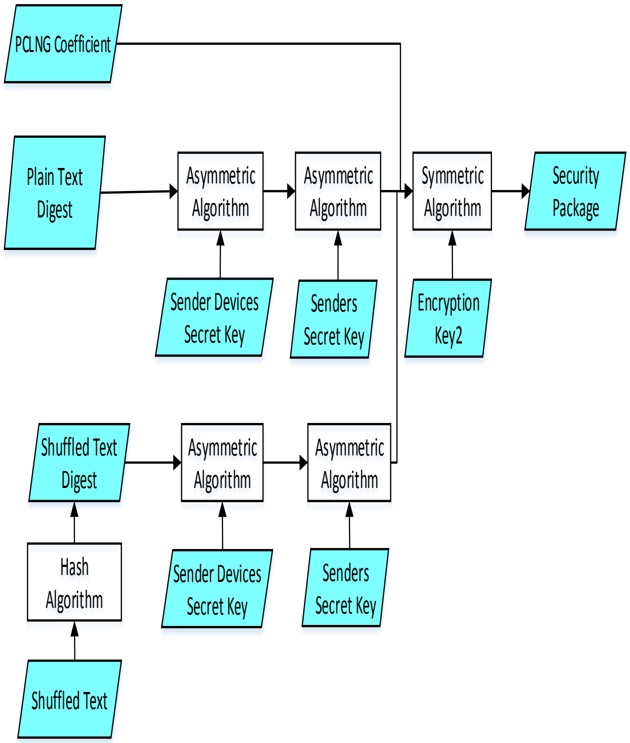

Fig.2. Signing and encryption of security package entries by device and person and creation of security package

C. Package3 - Key Package

Asymmetric algorithm is used in the last packet. Encryption keys for the first two packages in the encrypted state are moved. Encryption keys of symmetric algorithms used in message package and security package are encrypted by Asymmetric Algorithm to create Packet3.

$\gamma_{PK}(K1, K2) = Ap$

γ= Asymmetric encryption algorithm

$K1$ = Message Encryption Key (Symmetric Algorithm)

$K2$ = Security Packet Encryption Key (Symmetric Algorithm)

PK = Receiver's Public Key (Asymmetric Algorithm)

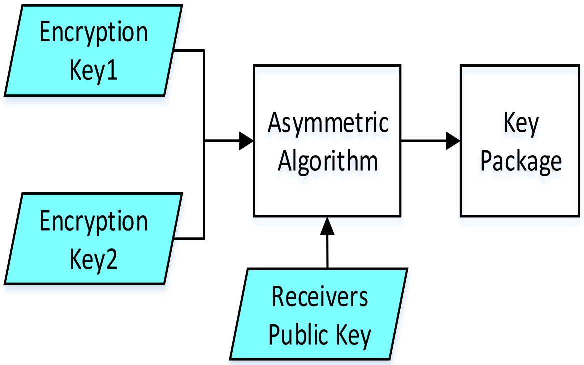

Fig.3. Key Package Creation Process

$Vp\ (Mp, Gp, Ap)$

Ap = Key Package

Gp = Security Package

Mp = Message Package

The data packet is formed by combining the message package security package and the password package.

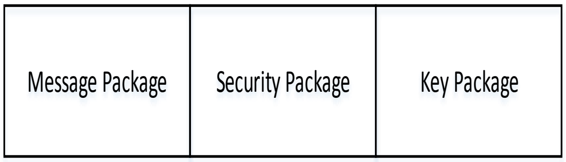

Fig.4. Combination of Password Pack and Packages

After receiving the recipient message, the keys needed to display the message content are encrypted in the key package. In this case, the key package must only be opened by the receiver. For this, the key package must be encrypted with the receiver's public key. Public keys must be mutually shared for the encryption to take place.

V. Decryption Process

When the data package arrives at the recipient, it is sufficient for the open message to reverse the order of the transactions in order to be able to access the signature information. The same key in the symmetric algorithm and the second key in the asymmetric algorithm must be used.

The packages are separated from each other in the very beginning. Each package follows its own sequence of operations within the architecture. First the security package is opened, and the shuffle coefficients and signatures are obtained, then signatures are checked to obtain hash values. Finally, the Message Pack is opened and a plain text is obtained. The shuffle algorithms for the plain text are compared with the hash values obtained from the security packet. If the values obtained in this comparison are the same, it is guaranteed that the message is not changed.

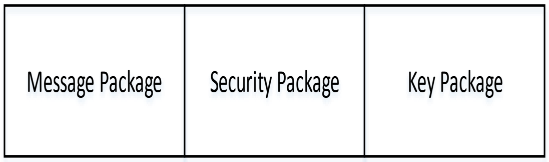

Fig.5. Combination of Password Pack and Packages

A. Step 1. Unpacking the Key Package

The key package is encrypted with the recipient's public key. The keys of the symmetric encryption algorithm used for message encryption and security packet encryption are obtained by unpacking the receiver's secret key.

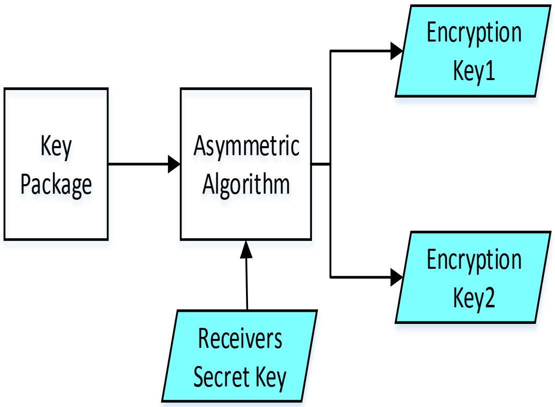

Fig.6. Unpacking the Key Package and Obtaining the Encryption Keys

$\gamma_{SKK}(Ap) = (K1, K2)$

γ= Asymmetric encryption algorithm

$K1$ = Message Encryption Key (Symmetric Algorithm)

$K2$ = Security Packet Encryption Key (Symmetric Algorithm)

SKK = User's Secret Key (Asymmetric Algorithm)

Encryption Key2 will be used to decrypt security package, Encryption Key1 will be used to obtain the clear message.

B. Step 2: Uncpacking Security Package

The security packet obtained from the data packet is decrypted using the Encryption Key2 obtained by decrypting the key packet. The obtained data are a hash of the message signed by the user and the device and the coefficients used in the process of shuffling the encrypted message. This hash value will be compared with a hash value of the message obtained in the next step, and the device and the person to whom the message is sent will be verified.

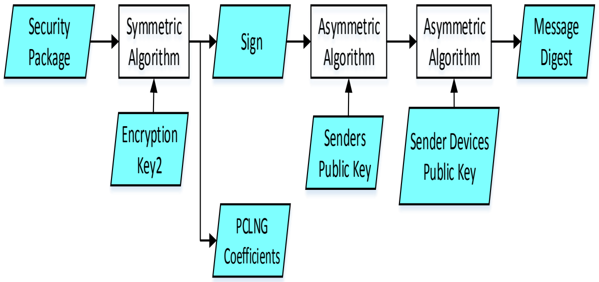

Fig.7. Unpacking of Security Package and Retreiving Hash Value and AKDU Coefficients

$\delta_{K2}(\tau) = (I_K, q)$
$SG_{SKK}(I_K) = I_C$
$SG_{SKC}(I_C) = (h_m, h_z)$
$H(m) = h_m$
$H(z) = h_z$
$I_K \rightarrow \delta_{K2}$
$q \rightarrow \delta_{K2}$

$\delta_{K2} = K2$ Symmetric encryption algorithm using K2 Key
$SG_{SK} = SK$ Signature Algorithm using Key K (Device-Person)
$H =$ Hash Algorithm
I_C =Signature of the message created by the device
$I_K =$ Signing a device signature created by a person
$K2 =$ Security Packet Encryption Key (Symmetric Algorithm)
$m =$ Plain Text
$h_m =$ Message Digest
$h_z =$ Hash of scrambled encrypted text
q =Coefficient Array
$\tau =$ Security Package

The signature is calculated by processing the digest value of the message data first with the device's secret key then sender's secret key by the asymmetric algorithm. The signature verification process is also performed by asymmetric algorithm using first the device's secret key then the sender's secret key. As a result of this process, a hash value of the signature data should be obtained.

C. Step 3: Unpacking Message Package

The string that was passed in the shuffling process after the message package is encrypted. Therefore, the cryptographic message must first be obtained by reversing the shuffling process, and the resulting cryptographic message must be decrypted with the Encryption Key 1 coming from the key packet with the symmetric algorithm. The coefficients that will be processed to recover the shuffling process come from the security package.

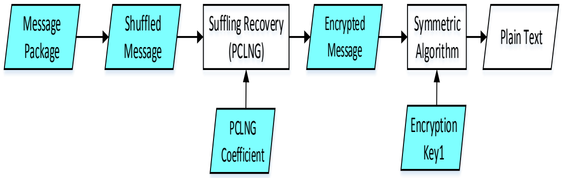

Fig.8. Unpacking Message Pack and Obtaining Open Message

$Sh(z)_q = y$
$\delta_{K1}(y) = m$

$\delta_{K1} = K1$ Symmetric encryption algorithm using Key K1
$Sh =$ Shuffle Algorithm
$K1 =$ Message Encryption Key (Symmetric Algorithm)
$z =$ Shuffled Encrypted Text
$q =$ Coefficient Array
$y =$ Encrypted Text (Message Packet)

D. Step 4: Verification

At the end of the second phase, the hash value was obtained. This summary value consisted of a message digest mixed with a plain message digest. Because it is not possible to obtain the original data from the summary value, the plain text is again processed to the shuffling and hashing operations to obtain the hash value which is verified by comparing the summary value with the data packet.

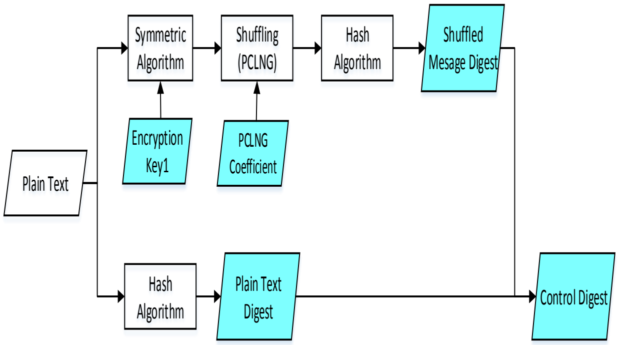

Fig.9. Confirmation of Value Compared with Security Package and Open Messaging Values

The control hash value is compared with the summary from the data packet. If the results are correct, the device and sender are verified.

VI. Prime Coefficient Linear Number Generator-PCLNG

A. Shuffling Process:

As A[n]={ $a_1, a_2, a_3, \dots, a_n$ } and 1 ≤ x ≤ n are, algorithm is like this:

Select p,q primes as2 ≤ p ≤ n and 2 ≤ q ≤ n.

1. Define Sh(x) function
   Sh:[1,n] → [1,n]
   Sh(x)=px+q mod n
2. $(T[x] = A[Sh(x)])_{x=1}^{n}$ place all values of array A to temporary array T using mapping function.
3. $(A[x] = T[x])_{x=1}^{n}$ transfer the temporary T array values to array A.

Using this algorithm we have shuffled array A.

Main question is if Sh:[1,n] → [1,n] Sh(x)=px+q mod n function is 1 → 1 ? .Is $Sh(x_1) = Sh(x_2)$, for $x_1 \neq x_2$.

Proof:

Select two elements as $x_1 \neq x_2$ in different places.

We assume that px_1+ q=px_2 + q mod n px_1+ q=px_2 + q + kn

$px_1=px_2 + kn$

$px_1 - px_2 = kn$

$p(x_1 - x_2) = kn$ is found.

i. İf k=0, because of $p \neq 0$, $x_1 - x_2 = 0 \Rightarrow x_1 = x_2$ is found, and it is against our approval

ii. If $x_1 - x_2 = \frac{nk}{p}$,because of gcd(p,n)=1, p∤n is found. That means p cannot divide n. We assume that p divide k (p|k): $\frac{k}{p} = k'$. At this time $x_1 - x_2 = nk' \Rightarrow x_1 = nk' + x_2$ is found. That means $x_1 > n$ and that situation is against $x_1 \leq n$.

Therefore Sh(x) = px+q mod n function is 1→1 function. And for all $x_1 \neq x_2$, becomes $Sh(x_1) \neq Sh(x_2)$.

B. Recovery Process:

To recover the original array after shuffling process, we must use reverse of the Sh(x)=px+q function.

Reverse of Sh(x)=px+q mod n function is $Sh' = \frac{x-q}{p}$. Because of this function's being linear the reverse of p and q to mod n can be used instead of them. Lets call p' and q' to the reverse of p and q. İf $p' = \frac{1}{p}$ mod n and $q' = -q$ mod n are calculated $Sh'(x)$ becomes, $Sh'(x) = (x + q')p'$ mod n.

The place of each element of the array is calculated in $Sh(x) = px + q$ mod n function, and their new place is found. To recover the original array we must calculate $Sh'(x) = (x + q')p'$ mod n for all places in the shuffled array and find their original places.

VII. Conclusion

Encryption and decryption times at different data sizes are measured by developing an application for performance evaluation.

RSA is used as the asymmetric algorithm in signing operations and ciphers, and AES-256 is chosen as the symmetric algorithm. The P and Q prime numbers required for the RSA algorithm are selected to be 402 digits - 1334 bits and 401 digits - 1331 bits.

The numbers E and D selected using P and Q are 802 digits - 2662 bits and 801 digits - 2658 bits.

The values measured by the application are given in the table.

TABLE 1
THE VALUES MEASURED BY THE APPLICATION

Data (Byte)	Encryption (Sec)	Decryption (Sec)
8	2,9021564	1,6582876
16	2,3145948	1,8892842
24	2,2512891	1,7815471
32	2,2989085	1,7368394
40	2,3057441	1,7242867
80	2,3866955	1,7844664
160	2,3197742	1,6806094
320	2,3422487	1,6720422
640	2,2734561	1,6412479
1280	2,361179	1,6358062
4000	2,2584238	1,5706789
12000	2,4123994	1,7610501
24000	2,6840385	1,8357856
48000	3,7703456	2,1218189
72000	5,5372788	2,7813307
96000	8,3044108	3,7025864
144000	16,4070262	7,4168919

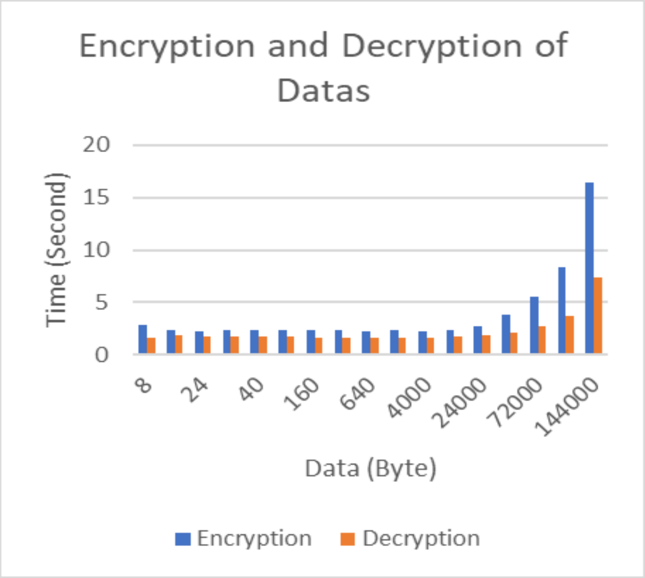

Fig.10. Encryption and Decryption Time of Datas

Certificate authorities do not prefer 1024 bit keys in signing and strategic applications anymore. Nowadays, 2048 or 4096 bit keys have begun to be preferred for longer security choice

Absolute security is ensured in the transmission of thumbnails and textual data as well as problems during transmission of audio and video data. Our next work will be on ensuring faster delivery of larger data.

REFERENCES

[1] H. Kodaz “Veri İletiminde Güvenlik İçin Şifreleme”, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2002

[2] T. Yerlikaya, E. Buluş, N. “Buluş, Asimetrik Şifreleme Algoritmalarında Anahtar Değişim Sistemleri”, Akademik Bilişim 2006 (Ab2006), 9-11 Şubat 2006, Denizli

[3] M. Krishnamurthy, E.S. Seagren, R. Alder, A.W. Bayles, J. Burke, S. Carter, E. Faskha, “Basics of Cryptography and Enryption, How to Cheat at Securing Linux”, 2008, 249- 270.

[4] T. Stapko, “Security Protocols and Algorithms, Practical Embedded Security”, 2008, 49-66.

[5] H. Kodaz, F. M. Botsali “Simetrik Ve Asimetrik Şifreleme Algoritmalarinin Karşilaştirilmasi”, Selçuk-Teknik Dergisi ISSN 1302-6178 Journal of Selcuk-Technic Cilt 9, Sayı:1-2010 Volume 9, Number:1-2010 10

[6] T. Yerlikaya, E. Buluş, N. Buluş “Asimetrik Şifreleme Algoritmalarinda Anahtar Değişim Sistemleri” AKADEMİK BİLİŞİM 2006 + BilgiTek IV 9-11 Şubat 2006 Pamukkale Üniversitesi Denizli

[7] M. Abe, R. Gennaro, K. Kurosawa, V. Shoup, “Tag-KEM/DEM: A New Framework For Hybrid Encryption And New Analysis Of Kurosawa-Desmedt KEM”, Advances in Cryptology – Eurocrypt 2005, Lncs 3494, Pp. 128–146, 2005.

[8] K. Yıldırım, H. E. Demiray, “Simetrik Ve Asimetrik Şifreleme Yöntemlerine Metotlar: Çırpılmış Ve Birleşik Akm-Vkm”, Gazi Üniv. Müh. Mim. Fak. Der. Cilt 23, No 3, 539-548, 2008

[9] S. C. Iyer, R.R. Sedamkar, S. Gupta, “A Novel Idea On Multimedia Encryption Using Hybrid Crypto Approach”, 7th International Conference On Communication, Computing And Virtualization 2016, Procedia Computer Science 79 (2016) 293 – 298.

[10] E. Atar, O. K. Ersoy, L. Özyılmaz, “Dik Eşleştirme Arayış Yöntemi İle Hibrit Veri Sıkıştırma Ve Optiksel Kriptografi”, Journal Of The Faculty Of Engineering And Architecture Of Gazi University 32:1 (2017) 139-147

[11] T. Arai, S. Obana, “A Password-Protected Secret Sharing Based On Kurosawa-Desmedt Hybrid Encryption”, Fourth International Symposium On Computing And Networking (CANDAR) CANDAR Computing And Networking (CANDAR), 2016 Fourth International Symposium On. :597-603 Nov, 2016

[12] L. Chengliang, Y. Ning, R. Malekian, W. Ruchuan, “The Hybrid Encryption Algorithm Of Lightweight Data in Cloud Storage”,2nd International Symposium On Agent, Multi-Agent Systems And Robotics (ISAMSR) Agent, Multi-Agent Systems And Robotics (ISAMSR), 2016 2nd International Symposium On. :160-166 Aug, 2016

[13] B.V. Srividya, S. Akhila, “Implementing A Hybrid Crypto-Coding Algorithm For An Image On FPGA”, Information And Communication Technology For Intelligent Systems, ICTIS 2017. (Smart Innovation, Systems And Technologies, 2018, 84:72-84)

[14] F. Yavuzer-Aslan, M. T. Sakallı, B. Aslan, “Önemli Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi”, Akademik Bilişim’12 - XIV. Akademik Bilişim Konferansı Bildirileri 1 - 3 Şubat 2012 Uşak Üniversitesi 49

[15] M Özbek, “Adli Bilişim Uygulamalarında Orijinal Delil Üzerindeki Hash Sorunları”, 1st International Symposium on Digital Forensics and Security (ISDFS’13), 20-21 May 2013, Elazığ, Turkey

[16] http://primes.utm.edu/curios/index.php?start=301&stop=1000

BIOGRAPHIES

HAKAN GENÇOĞLU Bursa, Türkiye, in 1978. He received the B.S. in Mathematics from İstanbul University in 2002 and M.S. degrees in Computer Engineering from the İstanbul Aydın University, İstanbul, in 2012 and the Ph.D. degree in Computer Engineering from Trakya University, Turkey, in 2018. Research topics are Asymmetric Encryption Algorithms performance and applications.

BIOGRAPHIES

TARIK YERLİKAYA İpsala, Edirne, in 1977. He received the B.S. Electronics and Communication Engineering from Yıldız Technical University in 1999, M.S. and Ph.D. degrees in computer engineering from the Trakya University in 1999 and 2006.

From 1999 to 2007, he was a Research Assistant with the Trakya University Computer Engirneering Department. He is Assistant Professor since 2007. His research interests focus Encryption Algorithms and their Cryptanalysis