Journal of educational equipment: Applied research, Volume 1, Issue 320 (September 2024)
ISSN 1859 - 0810

A new approach to improving the efficiency of end-to-end encryption systems based on the Pareto principle

Le Ba Cuong and Do Van Thuc

Academy of Cryptography Techniques

Received: 05/8/2024; Accepted: 05/9/2024; Published: 10/9/2024

Abstract: *This paper presents a new method for determining cryptographic parameters for end-to-end encryption using a multi-objective optimization approach. It is crucial to select appropriate parameters that balance security and performance for systems with different security needs. To overcome this challenge, the proposed approach focuses on creating a detailed mathematical model for the end-to-end encryption problem using multi-objective optimization. We evaluated the optimization model based on security, performance, and global objectives. The experimental model was tested using a custom-developed end-to-end encryption application with multiple data files of varying sizes and cryptographic parameters. The evaluation results demonstrate that it can be applied to various cryptographic systems and schemes.*

Keywords: *E2EE, multi-objective optimization, Pareto optimization.*

I. INTRODUCTION

In the fast-evolving field of secure communication and data privacy, the challenge of End-to-End Encryption (E2EE) has gained critical importance. E2EE safeguards sensitive data by encrypting it at the source and ensuring that only the intended recipient can decrypt it, maintaining confidentiality, integrity, and security throughout the communication process. However, as the digital landscape becomes more intricate, E2EE faces numerous challenges, including the need for stronger cryptographic measures, concerns over computational efficiency, and resilience against future threats. Addressing the optimization of encryption has become a crucial task that demands thorough investigation and solutions.

When securely exchanging or storing files online, using end-to-end encryption (E2EE) provides the highest security. A new optimization method using the Pareto principle focuses on safety and performance [1, 2]. Experimental results confirm its effectiveness, and further improvements are planned.

The interdisciplinary approach in the investigation of Multi-Objective Optimization for the End-to-End Encryption Problem aims to redefine data privacy and secure communication. The paper proposes a new method for selecting cryptographic primitives, striking a balance between security and performance. The effectiveness of the approach is demonstrated through experiments with Duplicati, showing that it can be applied to various cryptographic systems and cipher suites.

The rest of the paper is organized as follows: Part II - Synthesis and analysis of related studies; Part III - IV. Background knowledge and Development of the multi-objective optimization for e2ee; Part V - Experimentation; Part V - Conclusion and development direction.

II. RELATED WORKS

In the field of cybersecurity, there's ongoing research on balancing end-to-end encryption's security benefits with its impact on system functionality. Various methods such as homomorphic encryption [3], secure multiparty computation [4], and property-preserving encryption [5] are used. Studies explore combining multi-objective optimization and cryptography to balance security, efficiency, and scalability in cryptographic systems [6-8]. Similarly, in backup storage, optimization strategies are being developed to meet multiple goals like data redundancy, recovery time, and resource utilization [9]. These investigations often use techniques from evolutionary algorithms and mathematical optimization to find trade-offs between competing objectives, contributing to more robust and efficient systems.

III. BACKGROUND KNOWLEDGE

A. Multi-object Optimization

Multi-object optimization, or multi-criteria optimization, addresses the challenge of simultaneously optimizing multiple conflicting objectives. It seeks to identify a set of solutions that represent a trade-off among the conflicting objectives, forming what is known as the Pareto front. This approach is crucial in decision-making processes where stakeholders need to explore diverse solutions that balance different criteria.

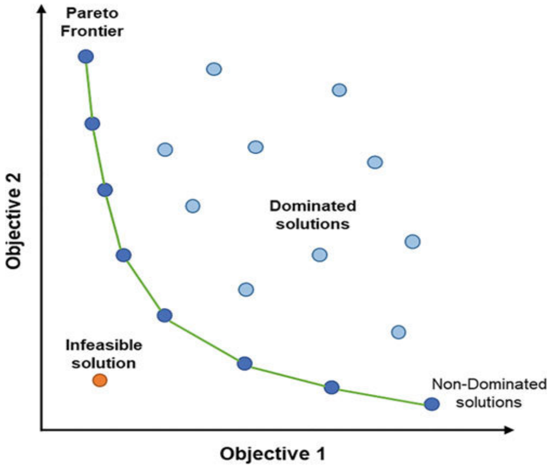

Fig. 1: Multi-objective optimization

Multi-objective optimization is a framework for handling optimization problems with conflicting objectives. It involves identifying Pareto optimal solutions and exploring trade-offs among competing objectives. In vector optimization, multiple objectives are represented as a vector, with the goal of optimizing the vector. This approach facilitates decision-making by providing diverse solutions with different trade-offs among objectives.

B. Pareto multi-objective optimization method

Pareto optimization is a key method in multi-objective optimization. In this method, calling X* the solution to be found, then X* must have the following properties:

• *X** must be at the point where all possible solutions to the problem are satisfied constraints *X** ∈ *D*.

• Every possible alternative other than X∈ D that has one objective better $(f_i(X) \geq f_i(X^*)$ must also have at least one other objective worse $(f_j(X)<f_j(X^*))$ with $i \neq j$.

*X** is also called the efficient solution. That is, efficient solutions are those satisfying: $\nexists X \in D$ which is possible: for any *i* where $f_i(X) >= f_i(X^*)$, and there must exist a j where $f_j(X)>f_j(X^*)$. Overall, there is no single X that can outperform X*.

C. End to End Encryption

Key management:

End to End Encryption is a popular encryption model. In this paper, based on [9], we build our E2EE diagram and the experimental program, described as follows.

Derive the key from the password

$$K_{0_i} = PBKDF2(PRF, Pwd_i, Salt, c, dkLen) \qquad (1)$$

where,

- PRF is a pseudorandom function
- Pwd_i is the password of the user i
- Salt is a sequence of bits
- c is the number of iterations desired
- dkLen is the desired bit-length of the derived key

Encrypt the private key and save it at the client.

$$K_{pri_i}' = Encrypt_{K_{0_i}}^{AES-l_{AES},\ m_{AES}}(K_{pri_i}) \qquad (2)$$

where,

- m_{AES} is the encryption mode of AES
- l_{AES} is the key length of AES.

Encrypt files:

Each file is encrypted and authenticated with its own, randomly generated encryption key, authentication key, and IV.

Randomly generate K_1, K_2, IV_1:

$$(K_1, K_2, IV_1) = (Hdrbg(l_1),\ Hdrbg(l_2),\ Hdrbg(l_3)) \qquad (3)$$

where,

- $l_1 = l_2 = 32$ (because the length of K_1 is equal to the length of K_2, and equal to 32 bytes).
- $l_3 = 16$ (because the length of IV_1 is equal 16 bytes).

Get all public keys of users in the group and encrypt K_1, K_2, IV_1.

$$(K_1', K_2', IV_1')_i = Encrypt_{Kpub_i}^{RSA-l_{RSA},\ m_{RSA}}(K_1, K_2, IV_1) \qquad (4)$$

where,

- m_{RSA} is the encryption mode of RSA
- l_{RSA} is the key length of RSA.

Encrypt data D of the file D'

$$D' = Encrypt_{K_1, IV_1}^{AES-l_{AES},\ m_{AES}}(D) \qquad (5)$$

Generate message authentication code H of the file:

$$H = HMAC_{K_2}^{SHA\,256}(D') \qquad (6)$$

Create encryption file according to the structure: Header contains $UserID_i$, $(K_1', K_2', IV_1')_i$ and *H*; The body part contains cipher of data D'. The structure of the encryption file has been carried out according to the regulations of the Cipher Committee, but due to confidentiality, we do not present it here.

Verify and decrypt files:

For each encryption file placed in the configured code folder to synchronize between client software and social networks.

Read the encryption file, separate the Header and code data D'. Analyze and retrieve the ciphertext $(K_1', K_2', IV_1')_i$ corresponding to the user's UserID i.

Decrypt K1, K2, IV1 with a key file:

Derive the key K_{0_i} from the user password:

$$K_{0_i} = PBKDF2(PRF, Pwd_i, Salt, c, dkLen) \tag{7}$$

Decrypt private key:

$$K_{pri_i} = Decrypt_{K_{0_i}}^{AES-l_{AES}, \ m_{AES}}(K_{pri_i}')) \tag{8}$$

Decrypt the message key:

$$(K_1, K_2, IV_1) = Decrypt_{K_{pri_i}}^{RSA-l_{RSA}, \ m_{RSA}}((K_1', K_2', IV_1')_i) \tag{9}$$

File validation:

Calculate $H' = HMAC_{K_2}^{SHA\,256}(D')$

and compare H' with H to verify.

If authentication is successful (H = H'), decrypt the file data:

$$D = Decrypt_{K_1, IV_1}^{AES-l_{AES}, \ m_{AES}}(D') \tag{10}$$

IV. DEVELOPMENT OF THE MULTI-OBJECTIVE OPTIMIZATION FOR E2EE

A. The overall mathematical model

Definition 1 – The security objective function

The security objective function - denoted f_1, is the function used to evaluate the end-to-end encryption model of the file encryption system. This function will be proportional to the key length, depending on the encryption mode. Each configuration of the model includes 4 parameters: the encryption mode and the key length of the cryptography using public keys and the cryptography using symmetric keys. This function can be evaluated according to the following Equation (11):

$$f_1 = a_1 \times \frac{l_{RSA}}{\max\limits_{l_{RSA} \in L_{RSA}} (l_{RSA})} + a_2 \times \frac{s(m_{RSA})}{\max\limits_{m_{RSA} \in M_{RSA}} (s(m_{RSA}))} + a_3 \times \frac{l_{AES}}{\max\limits_{l_{AES} \in L_{AES}} (l_{AES})} + a_4 \times \frac{s(m_{AES})}{\max\limits_{m_{AES} \in M_{AES}} (s(m_{AES}))} \tag{11}$$

where,

- a_i, $i = 14$, is the weight of the parameters of encryption models, such as the key length of RSA, AES, mode of RSA, AES
- $\sum_{i=1}^{4} a_i = 1$.
- m_{RSA} is the encryption mode of RSA
- l_{RSA} is the key length of RSA
- M_{RSA} is the set of RSA encryption modes
- L_{RSA} is the set of RSA key lengths
- m_{AES} is the encryption mode of AES
- l_{AES} is the key length of AES
- M_{AES} is the set of AES encryption modes
- L_{AES} is the set of AES key lengths.

Definition 2 – The performance objective function

The performance objective function - denoted f_2, is the function used to evaluate the speed of the system using the proposed encryption model. This function is inversely proportional to the key length and dependent on the encryption mode of the cryptography. This function can be evaluated according to the Equation (12):

$$f_2 = b_1 \times \frac{\min\limits_{l_{RSA} \in L_{RSA}} (l_{RSA})}{l_{RSA}} + b_2 \times \frac{\min\limits_{m_{RSA} \in M_{RSA}} (t(m_{RSA}))}{t(m_{RSA})} + b_3 \times \frac{\min\limits_{l_{AES} \in L_{AES}} (l_{AES})}{l_{AES}} + b_4 \times \frac{\min\limits_{m_{AES} \in M_{AES}} (t(m_{AES}))}{t(m_{AES})} \tag{12}$$

where,

- b_i, $i = 14$, is the weight of the parameters of encryption models, such as the key length of RSA, AES, mode of RSA, AES
- $\sum_{i=1}^{4} b_i = 1$
- m_{RSA} is the encryption mode of RSA
- l_{RSA} is the key length of RSA
- M_{RSA} is the set of RSA encryption modes
- L_{RSA} is the set of RSA key lengths
- m_{AES} is the encryption mode of AES
- l_{AES} is the key length of AES
- M_{AES} is the set of AES encryption modes
- L_{AES} is the set of AES key lengths.

Definition 3 – The global objective function

The global objective function – denoted f, is calculated by the component objective functions. This function is evaluated by Equation (13):

$$f = w_1 \times f_1 + w_2 \times f_2 \tag{13}$$

where,

- w_1 is the weight of the security function f_1.
- w_2 is the weight of the performance function f_2.

The mathematical model of optimization problem:

$$\begin{cases} f(m_{RSA}, l_{RSA}, m_{AES}, l_{AES}) \rightarrow max \\ (m_{RSA}, l_{RSA}, m_{AES}, l_{AES}) \in Configuration\ space \end{cases} \quad (14)$$

where,

- m_{RSA} is the encryption mode of RSA and l_{RSA} is the key length of RSA.
- m_{AES} is the encryption mode of AES and l_{AES} is the key length of AES.

B. The objective optimization algorithm

To find the optimal set of parameters for the multi-objective approach - tradeoff between safety and performance, the paper performs the algorithm to find f_{max} in Equation (13) as follows:

Input:

- The RSA key length array
- The RSA mode array
- The AES key length array
- The AES mode array

Output:

- The optimization 4-tupe: RSA key length, RSA mode, AES key length, AES mode.

```
Algorithm
Step 1:
Initialize 4 parameter arrays: array of RSA key lengths, array of RSA modes, array of AES key lengths, array of AES modes.
Initialize fmax = 0
Step 2:
Browse for each set of 4 ($l_{RSA}$, $m_{RSA}$, $l_{AES}$, $m_{AES}$) in the solution space as the Cartesian product of 4 initial arrays
    • Calculate $f_1$ according to Equation (10)
    • Calculate $f_2$ according to Equation (11)
    • Calculate f according to Equation (12) with weight set $w_1 = w_2 = 0.5$.
    • Compare f with $f_{max}$; If $f > f_{max}$ then save the current configuration and assign fmax
Step 3:
Select the set of parameters corresponding to fmax. This is a optimization solution.
```

V. EXPERIMENT

A. Data and programs

In this experiment, we established a data set of 500 files with sizes ranging from 1 MB to 500 MB. These files will be employed for end-to-end encryption; the average time is calculated by encrypting each file ten times.

To conduct our experiments, we developed two programs: Program 1 deploys the optimization algorithm described in Section IV.B to find the optimal set of parameters, while Program 2 performs end-to-end encryption on the built data set and calculates the average execution time for each file. Program 2 is used to verify the optimal results obtained by implementing Program 1.

B. Experimental results

In this experiment, we use hash function having a fixed key length 256 bit, so in the result tables, we do not show this hash function.

Experimental results are shows in Table 1, Table 2 and Table 3. Table 1 - illustrates a partial combination of parameters in the end-to-end encryption model, evaluation function values and the optimal parameter tupe. Table 2 – illustrates a portion of the average execution time when end-to-end encrypting some typical files, the objective function value is based on the actual execution time.

In Equation (11), a_i are the estimated weight values to evaluate the impact of key length and cipher mode on complexity. To achieve the results in Table 1, the paper set the weights $a_1=0.4$, $a_2=0.1$, $a_3=0.4$, $a_4=0.1$.

In Equation (12), b_i are the estimated weight values to evaluate the impact of key length and cipher mode on performance. To achieve the results in Table 1, the paper set the weights $b_1=0.45$, $b_2=0.05$, $b_3=0.45$, $b_4=0.05$.

In the paper, we choose the weights *w1=0.5 and w2=0.5* to evaluate the importance of performance and safety equally. In Table 1, the bold line is the global optimal parameter set, corresponding to the largest f value. In Table 2, the bold line shows the experimental results (including the average encryption and decryption time with 500 files) corresponding to the global optimal parameter set. This result shows that the execution speed is not the largest, but it is balanced with the complexity.

TABLE 1. ILLUSTRATION OF THE PROPOSED PARETO MULTI-OBJECTIVE OPTIMIZATION RESULTS

	RSA mode	**RSA key lengths**	**AES mode**	**AES key length**	f_1	f_2	f

1	PKCS	512	EBC	128	0.70	2.00	1.35
2	PKCS	512	EBC	192	0.90	1.70	1.30
3	PKCS	512	EBC	256	1.10	1.55	1.33
4	OAEP	512	EBC	128	0.80	1.95	1.38
5	**OAEP**	**512**	**OFB**	**128**	**0.90**	**1.90**	**1.40**
6	OAEP	512	OFB	256	1.30	1.45	1.38
7	OAEP	1024	CTR	256	1.40	1.00	1.20
8	OAEP	2048	OFB	128	1.20	1.23	1.21
9	OAEP	3072	CBC	256	1.78	0.73	1.24
10	OAEP	4096	CTR	256	2.00	0.66	1.33

TABLE 2. ILLUSTRATION OF AVERAGE EXECUTION TIME AND ACTUAL EXXECUTION TIME RESULTS

	RSA mode	RSA key length	AES mode	AES key length	f2	Avg. enc speed	Avg. dec speed
1	PKCS	512	EBC	128	2.00	85.4	128.1
2	PKCS	512	EBC	192	1.70	84.6	126.9
3	PKCS	512	EBC	256	1.55	80.8	121.2
4	OAEP	512	EBC	128	1.95	81.5	122.25
5	**OAEP**	**512**	**OFB**	**128**	**1.90**	**71.1**	**106.65**
6	OAEP	512	OFB	256	1.45	68.4	102.6
7	OAEP	1024	CTR	256	1.00	66.5	99.75
8	OAEP	2048	OFB	128	1.23	64.7	97.05
9	OAEP	3072	CBC	256	0.73	62.9	94.2
10	OAEP	4096	CTR	256	0.66	60.2	90.3

TABLE 3. END-TO-END ENCRYPTION AND DECRYPTION EXECUTION TIME

Order	**File name**	**Size (MB)**	**Avg. enc. time (s)**	**Avg. enc. speed (MB/s)**	**Avg. enc. time (s)**	**Avg. dec. speed (MB/s)**
1	File1	1	0.013	76.9	0.009	115.4
2	File2	2	0.028	71.4	0.019	107.1
3	File3	3	0.042	71.4	0.028	107.1
…	…	…	…	…	…	…
498	File498	498	7.265	68.5	4.843	102.8
499	File499	499	7.285	68.5	4.857	102.7
500	File500	500	7.355	68.0	4.903	102.0
	Average			**71.1**		**106.6**

VI. CONCLUSION

This article introduces a new method for optimizing the E2EE model using the Pareto principle. The method focuses on two main objectives: safety and performance. Experimental results have confirmed the accuracy and effectiveness of this approach. We aim to continue improving this optimization method to support more objectives and apply it to various fields.

REFERENCES

[1] K. Miettinen, "Nonlinear Multiobjective Optimization", *Kluwer Academic Publishers*, 1999.

K. Deb, "Multi-Objective Optimization Using Evolutionary Algorithms", *John Wiley & Sons Ltd.*, 2001.

[2] Craig Gentry. "Fully homomorphic encryption using ideal lattices". *In Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, pages 169–178. ACM, 2009

[3]. Wilko Henecka, et al., "Tasty: Tool for automating secure two-partycomputations". *In Proceedings of the 17th ACM Conference on Computer and Communications Security*, pages 451–462, 2010

[4]. Nathan Chenette, et al., "Practical order-revealing encryption with limited leakage". *In Proceedings ofthe 23rd International Conference on Fast Software Encryption*, pages 474–493. Springer Berlin Heidelberg, 2016

[5]. Hashemi, Seyed et al., "Multi-objective optimization for computer security and privacy". *International Journal of Network Security*. Vol 19. p394-405. 10.6633/IJNS.201703.19(3).08, 2017

[6]. K. Knežević, "Combinatorial Optimization in Cryptography," *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2017, pp. 1324-1330.

[8]. Suri, S., Vijay, R. "A Pareto-optimal evolutionary approach of image encryption using coupled map lattice and DNA". *Neural Comput & Applic 32*, 11859–11873 (2020).

[9]. Nextcloud, "*End-to-End Encryption Design*", Whitepaper 2017.