Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Website Design For Solving And Securing Confidential Files With The Asmuth Bloom Secret Sharing Protocol

Edward Fernando[1*], Edi[2], Octara Pribadi[3]

[1,3]*Informatics Engineering, STMIK TIME Medan, Indonesia*
[2]*Information Systems, STMIK TIME Medan, Indonesia*
edwardf1702@gmail.com[1*], edi@stmik-time.ac.id[2], octarapribadi@gmail.com[3]

Abstract

Conventional cryptographic algorithms cannot be applied to break a plaintext (message) into several ciphertexts because conventional cryptographic algorithms can only produce one ciphertext from a plaintext (message). In this case, a cryptographic protocol can be applied, namely the Asmuth-Bloom secret sharing protocol. The working process of the Asmuth-Bloom secret sharing protocol is divided into two parts, namely the shadow formation process and the shadow merging process. The shadow formation process will start from the process of filling in the message and the collection of d values along with the values of m, n and prime numbers p. After that, the process is continued by converting the message to ASCII Code. After that, each value will be processed according to the input d value so that a collection of shares is obtained. Meanwhile, the shadow merging process will start from filling in m shares. After that, the process is continued with the calculation of each character using the Chinese Remainder Theorem. The process will end with the merging of each character produced so that the original message is obtained. The results of this study are in the form of a website-based application that provides an interface to carry out the process of breaking text files into n shadow files and merging m shadow files into the original text file.

Keywords: *cryptography, secret sharing, shadow, message, Asmuth Bloom method*

1. Introduction

Data security is a crucial component that requires careful consideration during data storage procedures. In certain contexts, access to confidential data may be permitted only if multiple authorized entities collaboratively contribute to the retrieval of that sensitive information [1]. Conventional or traditional cryptographic algorithms have the ability to generate ciphertext derived from plaintext alone [2]. To generate ciphertext from plaintext, one must use methodologies beyond the scope of traditional cryptographic algorithms [3]. To do this, secret sharing protocols can be implemented.

A secret sharing scheme allows a secret to be shared among n participants such that only k of them can reconstruct the message, but any (k–1) cannot obtain any information about the secret, where $k \leq n$ [4]. In the cryptography literature, many secret sharing algorithms can be found, one of which is the Asmuth Bloom secret sharing scheme. The Asmuth-Bloom Secret Sharing Algorithm is used to facilitate data security by generating shadows to be distributed among many individuals and then combining these shadows to reconstruct the original message [5]. This algorithm utilizes prime numbers along with random numbers to enhance its security measures. Furthermore, the implementation of this algorithm requires the use of 'n' different sequences of numbers, which must comply with certain criteria [3]. The process of generating the ciphertext in the Asmuth-Bloom algorithm is relatively easy, achieved through the implementation of modulo addition operations. In contrast, the reconstruction of the original message is much more complicated, requiring the application of the Chinese Remainder theorem [6]. This is the advantage of the Asmuth-Bloom method. The Chinese Remainder Theorem (CRT) is a mathematical framework that facilitates the simplification of extensive modular exponentiation [7]. The Chinese Remainder Theorem (CRT) has several advantages, such as being able to improve security, having low computational complexity, allowing the replacement of large integer computations with several similar computations on small integers and helping to simplify large modular exponentiation. The efficacy of the Chinese Remainder Theorem lies in its ability to reconstruct integers in a specified range of values, utilizing the remainders obtained from coprime-number pairs [8].

2. Literature Review

2.1. Cryptography

Cryptography is a scientific discipline that examines the clandestine encoding of information through mathematical techniques. To maintain data confidentiality through cryptography, the original data, referred to as plaintext, is converted into an encoded format known as ciphertext, with the original data being retrievable solely through the use of a specific key. This process effectively reduces the risk of data exploitation by unauthorized entities. Cryptography is classified into three distinct categories based on the key utilization: symmetric

cryptography, asymmetric cryptography, and hybrid cryptography. When a single key is used, the cryptographic method is designated as symmetric. Symmetric cryptography is often preferred due to its fast processing power, resulting in minimal consumption of computing resources. Well-known algorithms used in symmetric cryptography include Rivest Cipher 4 (RC4), Triple Data Encryption Standard (DES), Blowfish, and Advanced Encryption Standard (AES). Conversely, in asymmetric cryptography, two different keys are used in the encryption and decryption processes. One key is made publicly available, while the other is kept secret. Examples of algorithms related to asymmetric cryptography include the Rivest-Shamir-Adleman (RSA) and Diffie-Hellman algorithms. Finally, hybrid cryptography represents an integrative approach that uses multiple algorithms to leverage the strengths of each. This algorithmic combination leverages the fast encryption capabilities of symmetric algorithms along with the secure key exchange characteristics of asymmetric algorithms.

2.2. Secret Sharing

This secret-sharing protocol operates on an (m, n)-threshold model; that is, information relating to a secret is disseminated in such a way that any m of n individuals (where $m \leq n$) have sufficient information to ascertain the secret, while a set of m-1 individuals is incapable of doing so. In any secret-sharing framework, there is a designated group of individuals whose aggregate information is sufficient to decipher the secret.
In certain implementations of the secret-sharing paradigm, each participant is allocated a portion of the secret after its generation. Conversely, in alternative implementations, the actual secret remains obfuscated from participants, although access to retrieve the secret is granted (e.g., through entry to a secure location or authorization to execute a specific process).
The variables included in the (m, n)-threshold framework fulfill different functions, as described in detail below:
1. The variable m indicates the number of parts required for the message to be understood.
2. The variable n indicates the total number of fragments or components of the message.
In addition, the variables inherent in the (m, n) threshold scheme must comply with the following conditions:

$$m \leq n \tag{1}$$

The operational mechanics of the (m, n) threshold scheme can be described as follows:
1. The message is partitioned into n segments, referred to as shadows or parts.
2. These segments are allocated to n individuals, each receiving a distinct part that is different from the others.
3. The value of m is set so that m message segments are required for the reconstruction of the original secret message.

2.3. Chinese Remainder Theorem Method

In the first century, the Chinese mathematician Sun Tzu made the following statement:
"Identify an integer which, when divided by 5, gives a remainder of 3; when divided by 7, gives a remainder of 5; and when divided by 11, gives a remainder of 7."
Sun Tzu's investigation can be articulated as a linear congruence system:

$$z \equiv 3 \pmod 5$$
$$z \equiv 5 \pmod 7$$
$$z \equiv 7 \pmod{11}$$

The Chinese Remainder Theorem will then be used to obtain solutions to linear congruence systems analogous to the one mentioned above:
Theorem 2.1: Let n1, n2, …, nn be positive integers for which GCD(ni, nj) = 1 for $i \neq j$. Then the linear congruence system:

$$z \equiv x_k \pmod{n_k} \tag{2}$$

has a unique solution modulo $n = n_1 . n_2 . \ldots . n_n$.
The procedure for solving a linear congruent system using the Chinese Remainder Theorem can be categorized into two distinct stages:
1. An attempt to find the inverse modulo value using the Extended Euclidean algorithm.
2. An attempt to obtain a solution to the linear congruent system using the derived inverse modulo value.
In addition, the Chinese Remainder Theorem utilizes the basic mathematical methodology used to solve systems of linear equations, specifically, systems characterized by substitution of equations.
In general, the Chinese Remainder Theorem operates as follows:
1. The initial linear congruence of the variable z, expressed as $z \equiv b1 \pmod{n1}$, is transformed into the equation $z = b1 + n1.k1$.
2. Substituting this primary linear congruence formulation into the next linear congruence can be articulated as follows:

$$z \equiv b_2 \pmod{n_2}$$
$$b_1 + n_1.k_1 \equiv b_2 \pmod{n_2}$$
$$n_1.k_1 \equiv b_2 - b_1 \pmod{n_2}$$

If the resulting expression (b2 - b1) (mod n2) is considered invalid, it must be reformulated into a valid configuration. A valid configuration of a linear congruence must satisfy the condition that the remainder modulo must be less than the modulus value. In this context, the expression (b2 - b1) must be less than n2. If this condition does not hold, then the expression (b2 - b1) must be taken modulo n2. The result of this modulo operation replaces the initial value (b2 - b1). As a result, the process can be described as follows:

if $((b_2 - b_1) < n_2)$ *then*
$\quad i = (b_2 - b_1)$
else,
$\quad i = (b_2 - b_1) \bmod n_2$

$$n_1.k_1 \equiv i \pmod{n_2}$$
$$k_1 \equiv i \pmod{n_2} . \mathbf{n_1^{-1} \pmod{n_2}}$$

The parts marked in bold can be calculated using the Extended Euclidean algorithm. Suppose n1-1 (mod n2) = y, then:

$k_1 \equiv i \,.\, y \pmod{n_2}$

If equation i . y (mod n2) obtained is invalid, so it must be changed to a valid form. So, the process can be written as follows:

If $(i \,.\, y < n_2)$ *then*

$g = i \,.\, y$

else,

$g = i \,.\, y \pmod{n_2}$

$k_1 \equiv g \pmod{n_2}$; or :

$k_1 = g + n_2.k_2$

3. Substitute the obtained result into the initial linear congruential relationship framework articulated as follows:

$z = b_1 + n_1.k_1$

$z = b_1 + n_1.(g + n_2.k_2)$

$z = b_1 + n_1.g + n_1.n_2.k_2$

$z = (b_1 + n_1.g) + (n_1.n_2).k_2$

4. Systematically repeat the procedure described in steps (2) and (3) for all remaining linear congruences until the terminal linear congruence (denoted as n), which has been derived:

$$z = (b_{n-1} + n_{n-1}.g) + (n_{n-1}.n_n).k_n \quad (3)$$

5. The result that satisfies the linear congruential system is represented by $(b_{n-1} + n_{n-1}.g)$.

2.4. Asmuth Bloom Method

The Asmuth-Bloom algorithm uses modular arithmetic, prime numbers, and stochastic elements to enhance its security measures. Furthermore, it requires the use of the Chinese Remainder theorem during the message recombination process. Broadly classified, the Asmuth-Bloom algorithm can be segmented into three procedural stages:

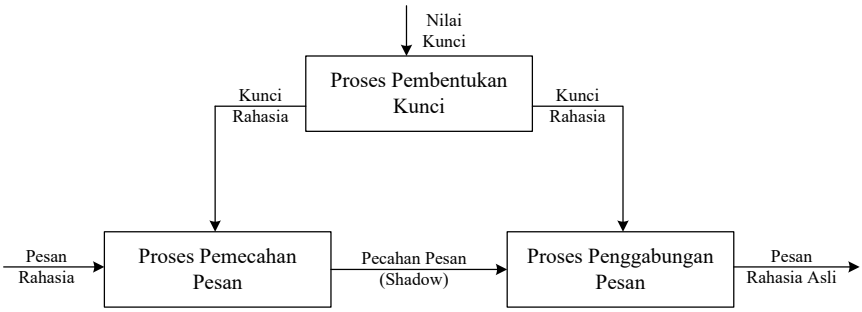

Fig. 1: Schematic of the Asmuth-Bloom Algorithm Work Process

1. Key Generation Process

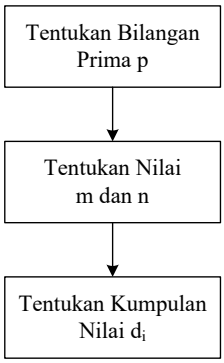

Fig. 2: Schematic of the Key Formation Process of the Asmuth-Bloom Algorithm

a. Identify the prime number p, where p exceeds the ASCII Code value corresponding to Message M.
b. Specification of m and n values, where $m \leq n$.
c. Definition of n numerical pieces less than p, in particular:

$$d_1, d_2, d_3, \ldots, d_n \quad (4)$$

such that:

1) The array of values d is arranged in ascending order, so that $< d_{i+1}$.
2) Each value must be coprime relative to every value in the remaining set.
3) $d_1 * d_2 * \ldots * d_m < p * d_{n-m+2} * d_{n-m+3} * \ldots * d_n$.

2. Shadow Generation Process

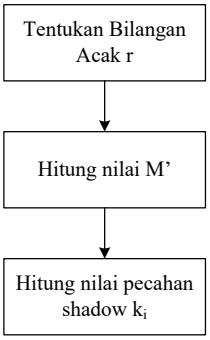

Fig. 3: Schematic of the Message Breaking Process of the Asmuth-Bloom Algorithm

a. The definition of the random variable r is mandated.
b. The calculation of the M' value is carried out using the following formula:
$M' = M + rp$ (5)
c. The message fragment (shadow) is:
$k_i = M' \bmod d_i$ (6)

3. Shadow Merging Process

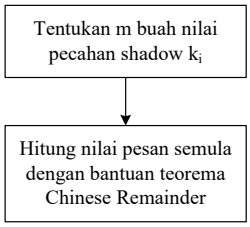

Fig. 4: Schematic of the Message Merging Process of the Asmuth-Bloom Algorithm

a. Determine the m ki values that you want to combine:
For example:
$k_1 = M' \bmod d_1$
$k_2 = M' \bmod d_2$
…
$k_m = M' \bmod d_m$
where the values of ki and di are known.
b. To find the value of M', the Chinese Remainder theorem is used.

3. Research Method

The Secret Sharing algorithm process has similarities with the operational framework of conventional cryptographic algorithms and can be described as three main components, namely as follows:

1. The key generation process, which is designed to generate keys to be used in the shadow generation and merging procedures. This key generation process can be visually represented in the form of a flowchart, as illustrated in Figure 5:

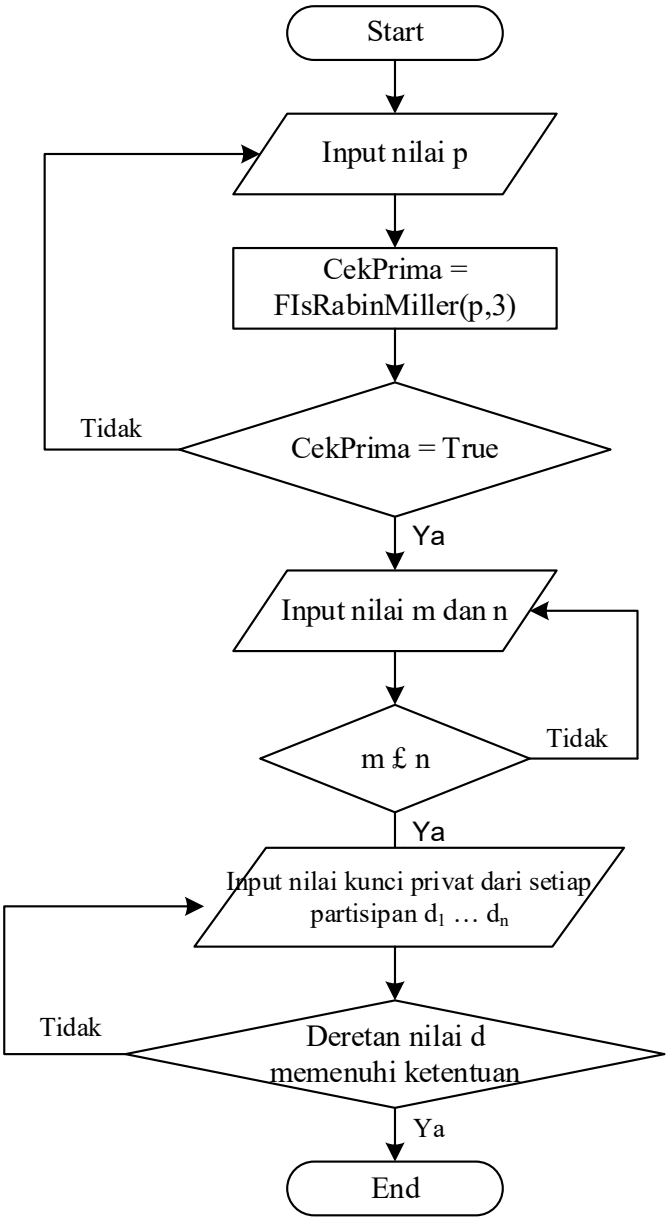

Fig. 5: Key Formation Process of the Asmuth-Bloom Secret Sharing Algorithm

2. The shadow formation process (message fragmentation), which is responsible for generating a set of shadows derived from the message. This procedure is analogous to the encryption process found in traditional cryptographic algorithms. The shadow formation process of this secret sharing algorithm can also be depicted in the form of a flowchart, as represented in Figure 6:

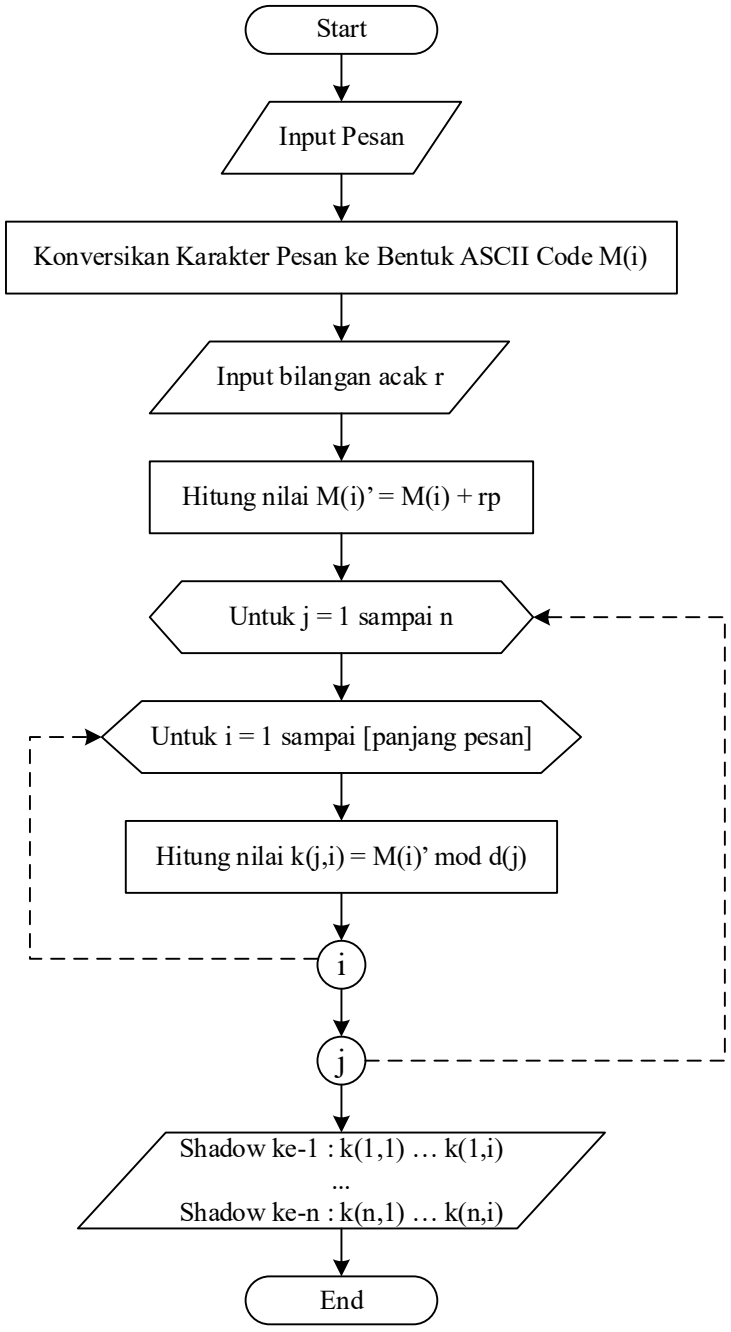

Fig. 6: Shadow Formation Process of the Asmuth-Bloom Secret Sharing Algorithm

3. The shadow merging process, which serves to reconstruct the original message using a certain number of shadows (according to the conditions set during the key generation process). This procedure mirrors the decryption process used in conventional cryptographic algorithms. The shadow merging process associated with the Asmuth-Bloom Secret Sharing algorithm can be depicted in a flowchart, as illustrated in Figure 7:

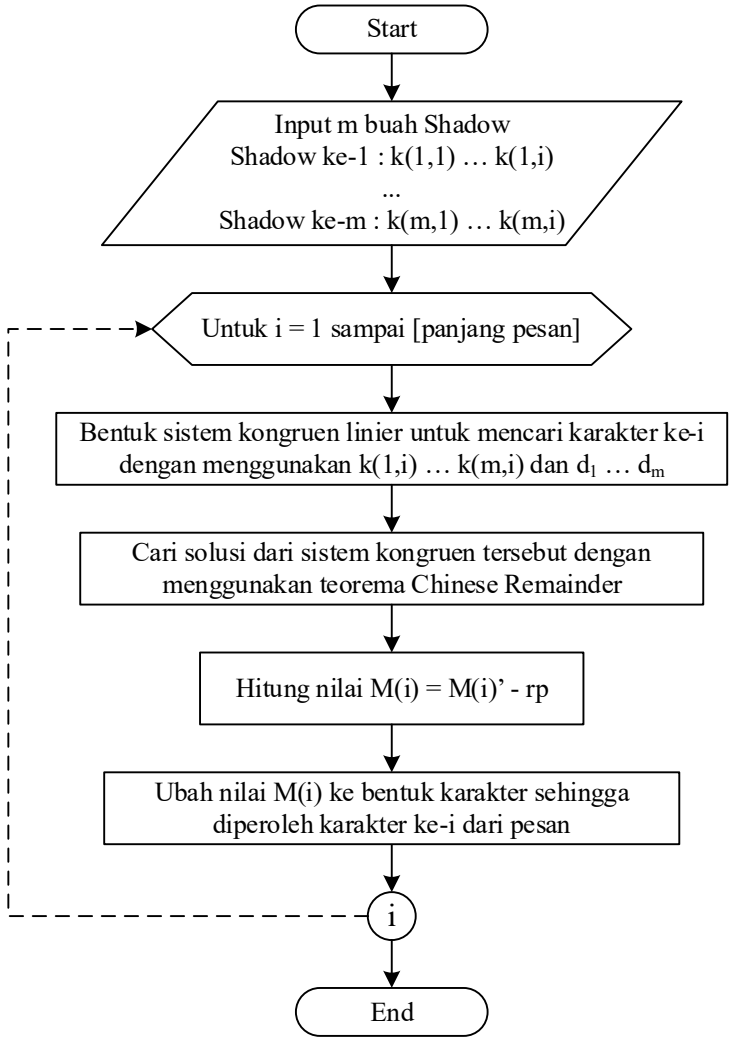

Fig. 7: Shadow Merging Process of the Asmuth-Bloom Secret Sharing Algorithm

4. Result and Discussion

The visual representation of this software is depicted as follows:

1. 'Main' form view:
   This view is the software's main interface, facilitating the interconnection of the various forms included in the software. The 'Main' form view can be seen in Figure 8 below:

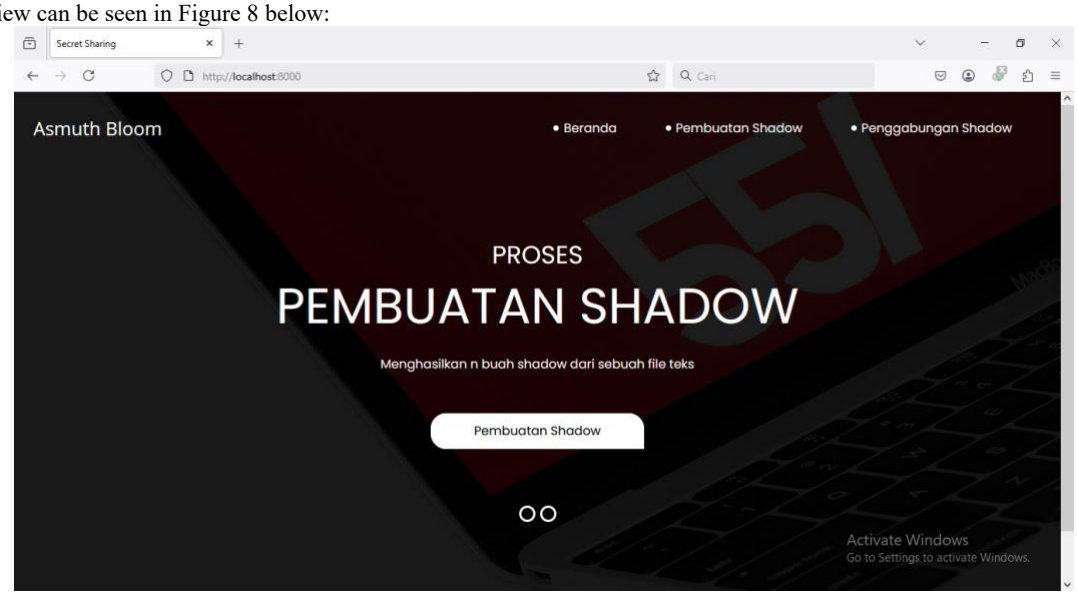

Fig. 8: Main Form View

2. The ‘Shadow Creation Application’ form display:
   This view is used to create shadow files that will be used in the shadow merging process. The 'Shadow Creation Application' form can be seen in Figure 9 below:

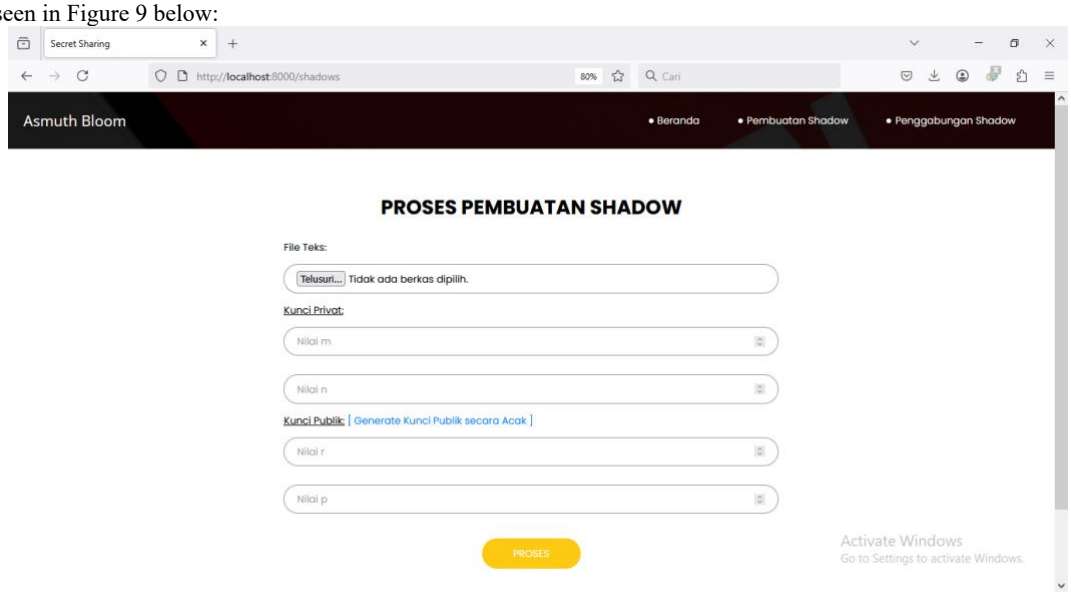

Fig. 9: Shadow Creation Application Form Display

3. The ‘Shadow Merge Application’ form display:
   This display functions to combine the shadow files produced in the shadow creation process to obtain the original text file. The display of the 'Shadow Merger Application' form can be seen in Figure 10 below:

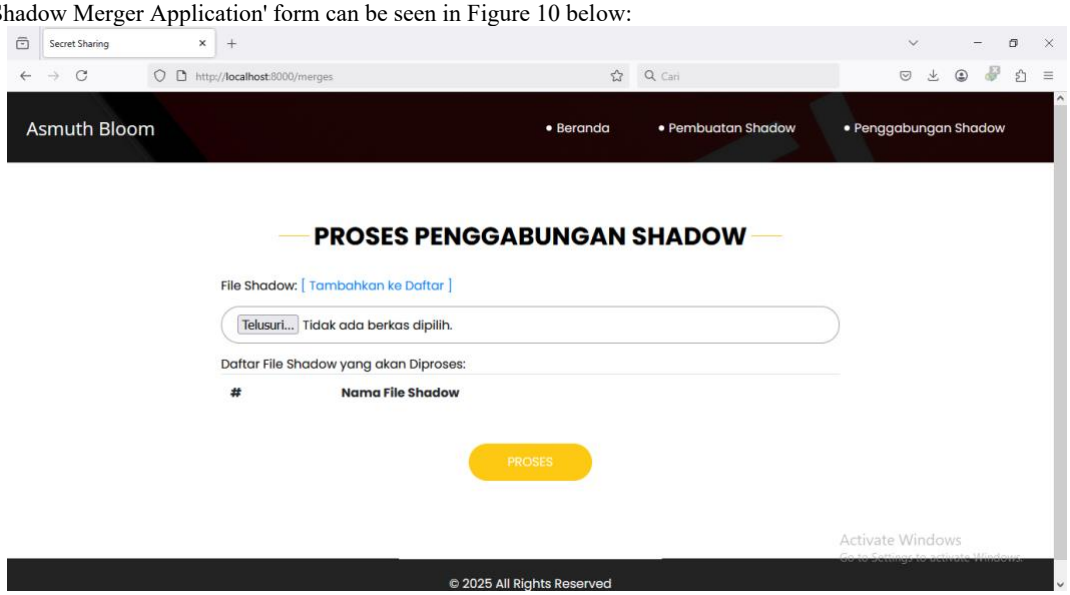

Fig.10: Shadow Merge Application Form Display

5. Conclusion

After completing this final assignment, the author outlined several conclusions, namely as follows:

1. This software includes a shadow creation application capable of unpacking a text file with the extension* txt into n shadow files carrying the extension* shr, thus allowing the division of a secret file into several shadow file fragments.
2. The Asmuth-Bloom algorithm can be used to produce n different ciphertexts from a message and only m ciphertexts are needed to recover the original message using the Chinese Remainder algorithm.
3. The shadow merging process has a longer execution time than the shadow creation process because the shadow creation process only uses modular addition and multiplication arithmetic operations, while the shadow merging process uses the Chinese Remainder theorem and the Extended Euclidean algorithm, which have complicated and long work steps.

Acknowledgement

The author would like to express sincere gratitude to Edi, Octara Pribadi for their valuable assistance and support in the preparation and completion of this journal. Their contributions, insights, and encouragement have been instrumental in ensuring the quality of this work.

References

[1] W. S. Raharjo, A. Rachmat C. dan P. Nadirio A., "Implementasi Secure Multi-Party Computation Menggunakan Metode Shamir Secret Sharing pada Pengamanan Dokumen Digital Rahasia," *JUISI,* vol. 04, no. 01, pp. 1-9, 2018.

[2] Khairani dan M. Z. Siambaton, "Pengamanan Data Teks Menggunakan Algoritma Kriptografi Elgamal dan XOR dari Serangan Hacker," *SUDO: Jurnal Teknik Informatika,* vol. 2, no. 4, pp. 1-12, 2023.

[3] Sugianto, Jimmy, A. Suwandhi, Wilianto dan Benny, "Analisis Metode Secret Sharing Asmuth-Bloom Dan Visual Cryptography," *Jurnal Minfo Polgan,* vol. 13, no. 1, pp. 213-218, 2024.

[4] R. S. Sinaga, S. Purba dan R. M. Siburian, "Aplikasi Pemahaman dan Penerapan Fast (k,n) Threshold Secret Sharing Scheme," *Jurnal Teknologi Informasi dan Indutri,* vol. 3, no. 1, pp. 76-83, 2023.

[5] E. A. Tarigan, "Perancangan Aplikasi Terenkripsi Dengan Menerapkan Metode Secret Sharing Asmuth Bloom," *Jurnal Riset Komputer (JURIKOM),* vol. 5, no. 6, pp. 648-652, 2018.

[6] I. S. Beno, "Analisis Teorema Sisa Cinadalan Dekripsi Data Text Terenkripsi RSA," *KOMBROF Jurnal Teknologi Informasi Papua(KJTIP),* vol. 1, no. 1, pp. 40-44, 2023.

[7] Z. Panjaitan, K. Ibnutama dan M. Suryanata, "Penggunaan Chinese Reminder Theorem (CRT) pada Algoritma RSA," *Sains dan Komputer (SAINTIKOM),* vol. 18, no. 1, pp. 41-46, 2019.

[8] C. O. Purba, "Implementasi Metode Chinese Remainder Theorem Untuk Menyisipkan Citra Digital Kedalam File Video," *Jurnal Sains Dan Teknologi Informasi,* vol. 1, no. 3, pp. 67-72, 2022.