Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2025. Vol. 4. No. 3; e-ISSN: 2808-4519

Application of Bounded Collusion for Identity-Based Encryption Using the Identity Based Encryption Algorithm

Theresia Mary[1]*, Octara Pribadi[2], Leony Hoki[3]

[1,3]*Informatics, STMIK TIME, Medan*
[2]*Information System, STMIK TIME, Medan*
theresia.mary.tm@gmail.com[1*]

Abstract

This research aims to design and develop an identity encryption application using the bounded collusion method with the implementation of the Identity Based Encryption (IBE) algorithm. The method combines IBE, bounded collusion, and key generation based on the user's email. The application was developed using Visual Basic. In its implementation, the application can perform text encryption and decryption while limiting the number of decryptions to a maximum of two times per identity, in accordance with the bounded collusion principle. The testing results show that the application effectively protects user identities by generating unique keys based on email and restricting potential collusion attacks between users. Therefore, the implementation of bounded collusion and IBE is proven to enhance the security of identity-based encryption processes.

Keywords: *Bounded Collusion, Identity Based Encryption, Identity Encryption, Visual Basic*

1. Introduction

With the advancement of technology, data security has become increasingly important due to the growing number of cyberattacks and attempts to steal sensitive information in various sectors such as education, healthcare, government, and others [1]. One approach used to address this issue is the development of identity-based encryption applications using cryptographic techniques that can protect user identities from unauthorized access [2]. Identity-based encryption is a concept that uses identity information, such as an email address, as the basis for generating cryptographic keys in the encryption process [3]. The use of the Bounded Collusion method is believed to be capable of overcoming challenges in maintaining the integrity and confidentiality of information [4].

Bounded Collusion is a cryptographic approach that limits the number of entities that can collaborate in an attempt to breach the security system, even if they do not have direct access to the secret key [5]. This limited collusion provides additional protection against joint attacks by multiple users trying to access encrypted data, by setting strict limitation parameters [6]. The risk of unauthorized information disclosure and identity theft continues to rise with the increasing exploitation of data, and conventional encryption is often insufficient to withstand complex attacks [7].

Based on this background, this study adopts the Identity-Based Encryption (IBE) algorithm combined with the Bounded Collusion approach to build a system that limits the maximum number of decryptions that can be performed by users [4]. Although this method involves higher implementation complexity and computational requirements, its advantage in preventing collusion attacks makes it a relevant choice. The application developed using Visual Basic software is expected to provide a practical and secure solution for various organizations to effectively protect user identity information [8]. Therefore, the author is interested in addressing this issue as the topic of the final project, entitled "Application of Bounded Collusion for Identity-Based Encryption Using the Identity-Based Encryption Algorithm."

2. Research Methods

2.1. Encryption and Decryption Process

The encryption process in the bounded collusion implementation begins with user input in the form of a string with a maximum length of 256 characters. Once the system validates the input length, it proceeds to the Identity-Based Encryption (IBE) phase, where identity information, such as an email address, is used as the basis for generating an encryption key. The resulting encrypted data is then reinforced with the bounded collusion mechanism, which limits the possibility of collusion through the use of a public key. The final stage of this

process is the formation of the ciphertext in an encrypted string format (e.g., hexadecimal or Base64), indicating that the data has been successfully secured.

The decryption process begins when the user inputs the ciphertext. The ciphertext is first processed through the bounded collusion stage, using the public key again to ensure protection against collusion attacks. Subsequently, the system performs the Identity-Based Decryption (IBD) process by using the user's identity (i.e., email) to generate the appropriate private key. With this private key, the ciphertext can be reverted to its original form, a string of up to 256 characters. The process concludes with a completion step, indicating that the data has been successfully decrypted and restored. Encryption and Decryption process are illustrated in Fig 1

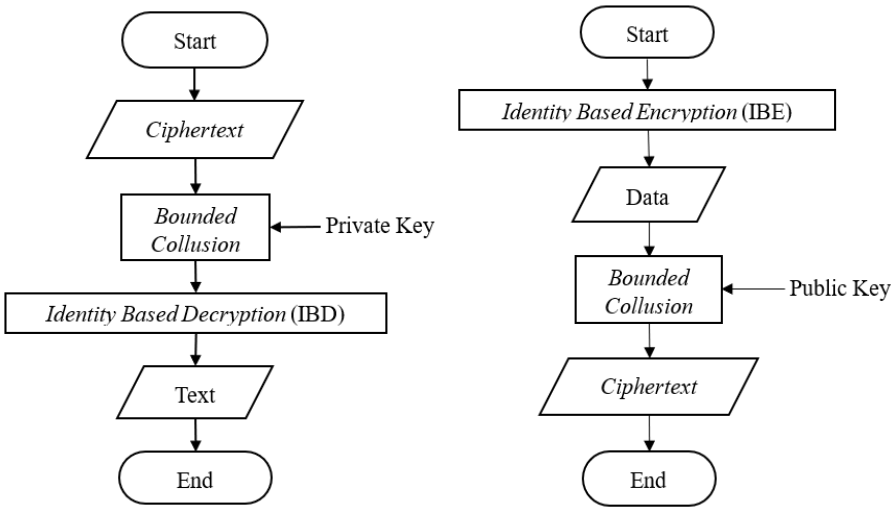

Fig. 1: Encryption and Decryption Process

2.2. System Design

The system is developed using Visual Basic and consists of several user interfaces, each serving specific functions within the encryption and decryption process. The design includes Login Form, Register Form, and Menu Form are illustrated in Fig 2

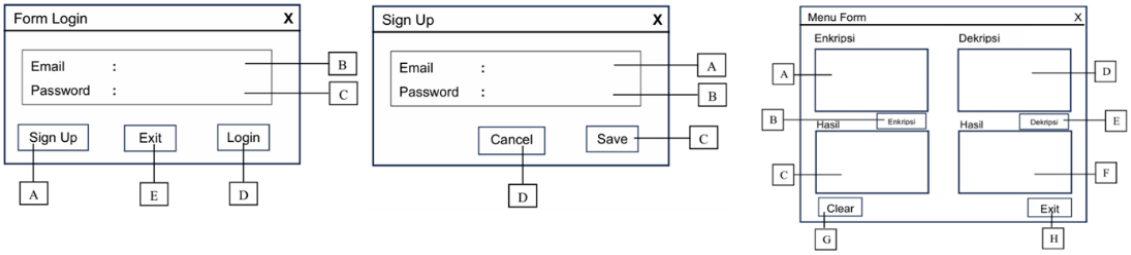

Fig. 2: Login, Register, and Menu Form

Descriptions of each button or function in each form as shown in Table 1.

Table 1: Functions of each button.

Button	Login Form	Register Form	Menu Form
A	Sign Up: Opens the registration form for new users	Email Field: Enter the email to be registered. If already registered, an error message is shown	Plaintext Input: Users enter the text to be encrypted
B	Email Input: Users must enter a registered email	Password Field: Users can enter a password (maximum 8 characters)	Encrypt Button: Triggers the encryption process
C	Password Field: Displays asterisks (‘*’) for security	Save Button: Saves the entered email and password, then redirects to the login form	Encrypted Output: Displays the resulting ciphertext in the textbox
D	Login: Opens the main menu containing encryption and decryption features	Cancel Button: Cancels the registration process and returns to the login form	Ciphertext Input: Field for users to input the encrypted text for decryption
E	Exit: Cancels the login process and closes the application	-	Decrypt Button: Executes the decryption process
F	-	-	Decrypted Output: Displays the plaintext result in the textbox
G	-	-	Clear : Erase all the input inside the textbox

3. Results and Discussion

The results of the research conducted by the author are described as follows

a. Login Form Display
In this interface, users are required to input their registered email and password. The login form consists of three buttons from left ti right as shown in Figure 3:
1. Sign Up – to register a new user, if not yet registered.
2. Exit – used to close the system.
3. Login – allows users to enter the main menu form.

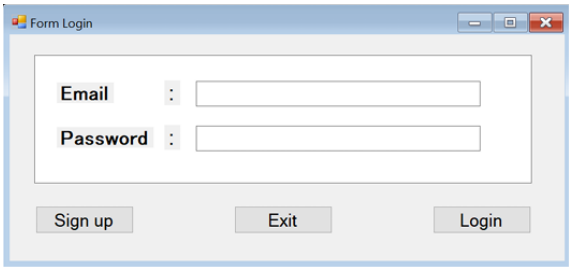

Fig. 3: Login Form Display

If the user does not yet have a registered account, they will be redirected to the registration (Sign Up) form.

b. Sign-Up Form Display
In this form, users enter an email and password which will be registered as a new user by clicking the Save button. The Cancel button redirects the user back to the login form as shown in figure 4. When the email is saved in the system, a cryptographic key is generated using the Identity-Based Encryption (IBE) scheme. This key will be used during both encryption and decryption processes.

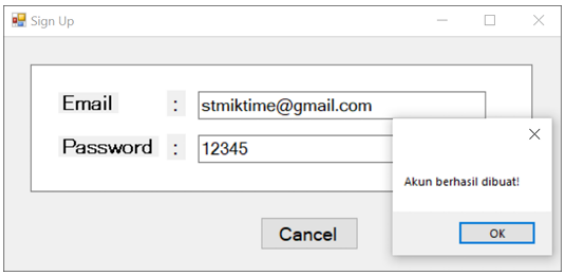

Fig. 4: Sign-Up Form Display

c. Menu Form Display
In this form, the user can enter a message to be encrypted or a ciphertext to be decrypted, as shown in the figure 5.

Fig. 5: Encryption and Decryption Results

The encryption process is initiated when the user clicks the Encrypt button. The system then generates a key based on the user's identity and encrypts the message into ciphertext. Similarly, during decryption, the same key is retrieved to convert the ciphertext back into plaintext. The key retrieval mechanism as shown in Figure 6.

Fig. 6: User Key Generation

If different emails are used, the resulting ciphertext will also be different even if the original message is the same. This is in accordance with the principles of Identity-Based Encryption, as illustrated in the figure 7.

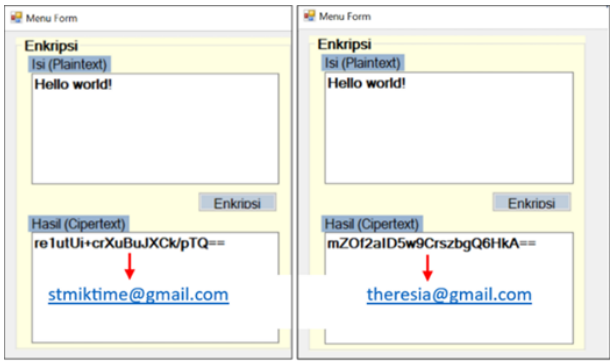

Fig. 7: Ciphertext Results for Different Users

In this experiment, each user has a different key because the email addresses are different, as shown in figure 8.

Fig. 8: Key per User

In the next experiment, the user attempts to decrypt the same ciphertext three times. On the third attempt, the system rejects the decryption process. This behavior occurs because the Bounded Collusion approach has been applied—limiting the number of decryption attempts to enhance security against collusion attacks. The enforcement of this limitation is illustrated in figure 9.

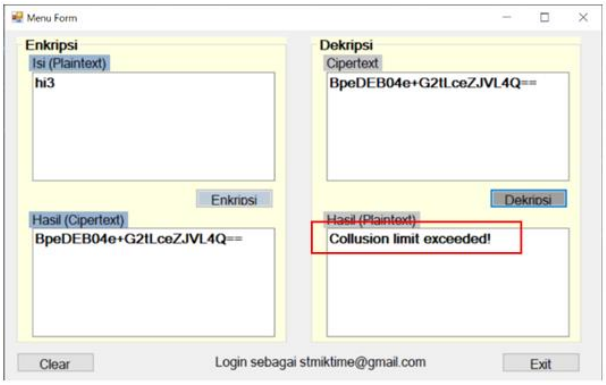

Fig. 9: Bounded Collusion Limit Enforcement

4. Conclusion

Based on the study, it can be concluded that the developed application successfully implements identity-based encryption (IBE) using the Bounded Collusion (BC) method, effectively enabling email-based encryption and decryption while serving as a logical safeguard against collusion attacks. The application operates reliably in Visual Basic with a simple interface, input validation, and basic security features. Future work should focus on integrating a database system to enhance data storage security, and the IBE approach shows promising potential for extension to cloud-based data security solutions.

References

[1] Dewan Teknologi dan Komunikasi Nasional, “Pengembangan keamanan siber nasional,” Policy Paper, 2018.
[2] J. S. Pasaribu, “Penerapan Algoritma Hill Chiper Dalam Pengamanan Data Dengan Teknik Enkripsi Dan Dekripsi,” in Seminar Nasional Telekomunikasi dan Informatika, 2016..
[3] W. Wardiana and P. Informatika-LIPI, “Pencegah Pembajakan Perangkat Lunak dengan Menggunakan Teknik Identity-Based Encryption dan Obfuscation,” Pencegah Pembajakan Perangkat Lunak dengan Menggunakan Teknik Identity-Based Encryption dan Obfuscation, 2009.
[4] G. Itkis, E. Shen, M. Varia, D. Wilson, and A. Yerukhimovich, “Bounded-collusion attribute-based encryption from minimal assumptions,” in Lecture Notes in Computer Science, vol. 10401, pp. 61–89, 2017, doi: 10.1007/978-3-662-54388-7_3..
[5] R. Garg, R. Goyal, G. Lu, and B. Waters, “Dynamic Collusion Bounded Functional Encryption from Identity-Based Encryption,” in Lecture Notes in Computer Science, vol. 13276, pp. 655–685, 2022, doi: 10.1007/978-3-031-07085-3_25.
[6] M. Azhari, D. I. Mulyana, F. J. Perwitosari, and F. Ali, “Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES),” Jurnal Pendidikan Sains dan Komputer, vol. 2, no. 1, 2022, doi: 10.47709/jpsk.v2i01.1390.
[7] A. K. Harsa, “Keamanan Data dengan Menggunakan Algoritma Rivest Code 4 (RC4) dan Steganografi pada Citra Digital,” Informatika Mulawarman, Feb. 2014.

[8] P. Ananth and V. Vaikuntanathan, “Optimal Bounded-Collusion Secure Functional Encryption,” in Lecture Notes in Computer Science, vol. 11478, pp. 489–518, 2019, doi: 10.1007/978-3-030-36030-6_8.
[9] S. D. Nurcahya, “Implementasi Aplikasi Kriptografi Metode Kode Geser Berbasis Java,” *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 4, pp. 694–697, 2022, doi: 10.32672/jnkti.v5i4.4690.
[10] N. A. Nanda, S. M. S. Silalahi, D. F. Nasution, M. Sari, and I. Gunawan, “Kriptografi dan Penerapannya Dalam Sistem Keamanan Data,” *J. Media Inform.*, vol. 4, no. 2, pp. 90–93, 2023, doi: 10.55338/jumin.v4i2.428.
[11] Univ_medanarea@uma.ac.id, “Perbedaan Simetris dan Asimetris,” *WordPress*. [Online]. Available: https://p2mbim.uma.ac.id/2023/01/07/perbedaan-simetris-dan-asimetris/
[12] P. Liu, “Public-key encryption secure against related randomness attacks for improved end-to-end security of cloud/Edge computing,” *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2967457.
[13] Y. Anshori, A. Y. E. Dodu, and D. M. P. Wedananta, “Implementasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) pada Tanda Tangan Digital,” *Techno.Com*, vol. 18, no. 2, 2019, doi: 10.33633/tc.v18i2.2166.
[14] E. A. Adeniyi, P. B. Falola, M. S. Maashi, M. Aljebreen, and S. Bharany, “Secure Sensitive Data Sharing Using RSA and ElGamal Cryptographic Algorithms with Hash Functions,” *Inf.*, vol. 13, no. 10, 2022, doi: 10.3390/info13100442.
[15] Y. Yan, “The Overview of Elliptic Curve Cryptography (ECC),” in *J. Phys.: Conf. Ser.*, vol. 2386, no. 1, 2022, doi: 10.1088/1742-6596/2386/1/012019.
[16] R. M. Simbolon, “Perancangan Perangkat Lunak Enkripsi Pesan dengan Metode Paillier Cryptosystem,” *Pelita Inform. Budi Darma*, 2013.
[17] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003, doi: 10.1137/S0097539701398521.
[18] A. Lewko, A. Sanais, and B. Waters, “Revocation systems with very small private keys,” in *Proc. IEEE Symp. Security and Privacy*, 2010, doi: 10.1109/SP.2010.23.
[19] A. Lewko and B. Waters, “Decentralizing attribute-based encryption,” in *Lect. Notes Comput. Sci.*, vol. 6632, pp. 568–588, 2011, doi: 10.1007/978-3-642-20465-4_31.
[20] D. J. Bernstein, “ChaCha, a variant of Salsa20,” *Work. Rec. SASC*, 2008.