Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 5-B, pp. 1845–1854
DOI : 10.47974/JDMSC-2361

EHC-BGV hybrid homomorphic encryption algorithmic sequences for processing authenticated role based secure keys

Santosh Kumar Henge [†]
Chanakya Kumar Hinge *
Deepthi Kothapeta [§]
Department of Computer Science and Engineering
School of Computer Science and Artificial Intelligence
SR University
Warangal 506371
Telangana
India

Sanjeev Kumar Mandal [‡]
Ravleen Singh [@]
Department of Computer Science & Information Technology
Jain (Deemed-to-be University)
Bangalore 560069
Karnataka
India

Avinash Bhagat [#]
Amit Sharma [$]
Department of Computer Application
School of Computer Application
Lovely Professional University
Jalandhar 144402
Punjab
India

[†] *E-mail:* **hingesanthosh@gmail.com**
* *E-mail:* **hingechanakya7@gmail.com (Corresponding Author)**
[§] *E-mail:* **deepthivaishu18@gmail.com**
[‡] *E-mail:* **sanjeev.mandal93@gmail.com**
[@] *E-mail:* **ravleen1234@gmail.com**
[#] *E-mail:* **avinash.bhagat@gmail.com**
[$] *E-mail:* **amit.25076@lpu.co.in**

© Taru Publications

Anantharamaiah Vengala [^]
Department of Artificial Intelligence & Data Science
Koneru Lakshmaiah Education Foundation
Vaddeswaram 520002
Andhra Pradesh
India

Aditya Upadhyay [+]
Department of Computer Application
Centre for Distance and Online Education
Manipal University Jaipur
Jaipur
Rajasthan
India

Abstract

The algorithm is designed to show the step-by-step procedure of the proposed methodology which experimented with four test cases: first one initiated with creation of multiple tenants in cloud architecture; second one with integration of uploading a file on cloud server on real-time; third one initiated end user request for accessing a file and last one processed self-key mutation in case of an attack. It integrated with key mapping parity rules and 14 multi-authentications, dependable, and non-dependable parameters which tuned with initiated variables based parameters which initiated and implemented at sender and receiver side for restricting unauthorized access. The methodology tested for 161 end-users, 10 multi-tenants, 8 head-tenants and 3 enterprise levels, and achieved 95 percentages of data transmission and access control rate.

Subject Classification: *Primary 68M25, Secondary 68P25, 68P27.*

Keywords: *Ciphertext (CT), Policy attribute encryption (PAE), DAC (Discretionary access control), Mandatory access control (MAC), Role based access control (RBAC), Task based access control (TBAC), Attribute based access control (ABAC).*

1. Introduction

Alternative, non-invasive, contactless, and scalable authentication procedures must be adopted in the current context, where user authentication serves as the first line of defense against developing attacks and can be regarded as a defining part of any security infrastructure [1]. A perspective on encryption that enables computing on encrypted data is

[^] *E-mail:* **ar.vengala@gmail.com**
[+] *E-mail:* **adityaupadhyay144@gmail.com**

HHE. Combining private, encrypted data in a cloud environment and using it for computations without jeopardizing data privacy are examples of potential uses [2]. Hence is the need for Homomorphic Cryptosystem [3-5]. HE algorithm schemes are used to perform the operations and computations on the encrypted data in a secure manner. Decryption process is not required as is required traditionally to perform the evaluation of values stored on the cloud [6-10]. This allows data integration, privacy and protection. The secret key (SK) and public key (PK) are generated at the client's terminal and the data variables x and y are encrypted using the PK [11-14].

This as it supports static and dynamic data both [3]. Hence, this hybrid approach is found to be more security [15].

2. Related Work

This section included the study and analysis of EHC, BGV and Hybrid EHC-BGV homomorphic algorithms and related innovations and methodologies. It also articulated basic key concepts and terms relevant to the research work like HHE algorithm [16], virtualization, cloud models etc are discussed in this chapter.

Toumanari analyzed the business opportunities in both financial and non-financial sectors using Blockchain technology [17]. Sun [18] proposed in their paper ciphertext-policy attribute-based encryption (CP-ABE) key encapsulation mechanism. This scheme reduces the expense for the encryption operation as in traditional CP-ABE. Sun [18] evaluated some technologies related to cloud security like CP-ABE, KP-ABE (Key policy attribute-based encryption), proxy re-encryption, access control, and multi-tenancy and suggested to include the upgraded and updated technologies and policies to secure the data [19][20]. Dubey and Rai [21] discussed different access methods which provide an efficient way to ensure only authorized users can access the data. Singh et al. [22] designed access rules for the E-health cloud based on the trust degree of the users to provide customized services to the users. Chhabra [23] compared the available access control techniques analyzed their suitability to the cloud computing environment. A multitenant environment requires fair access control between its tenants [24]. Ding, Yan, and Deng [25] proposed a flexible way to control the access using a privacy-preserving data processing scheme.

Hingwe and Bhanu [26] proposed a role-based access control that is integrated with PHE Paillier's algorithm for a multi-tenant database. Yu

et. al. [27] proposed a multi-bit public key encryption protection based on learning parity with noise [28].

3. Methodology

Algorithm for generating the token/keys based on the user-role-type. In proposed methodology, tenants, also known as users, are divided into two categories: authorized tenants and unlawful tenants. An unauthorized user or an attacker does not have legitimate login credentials to access the cloud, while an approved tenant does. A dictionary with information about all the approved tenants is created in order to determine the type of user. The user's name, IP address, and location are required for registration. The post-registration key is in active mode, but it is not enabled. To enable it, the user needs to modify the key from its default value. Using the mutation technique, access to unauthorized users is automatically prevented. In case, an attack is identified using the dictionary technique; the self-key mutation takes place automatically. This is a process of shuffling bits is based on some rules. Consider a scenario where an attacker can guess the password, considering 30 seconds are needed to guess a password or a character. If the attacker succeeds to guess a character, then in these 25 to 30 seconds, the self-key mutation process occurs. This process of self-key mutation repeats itself after every 30 seconds. Hence after every 25 seconds, the password will change and the attacker cannot access the cloud data leaving the cloud protected and secured. The authorized user can request for a new password again to access the file which the attacker failed to access. As per the end user level, the user will receive an email after 3 invalid attempts by the attacker. The email contains the geographical information, IP address and other relevant details of the attacker. The self-key mutation concept is used for randomly changing the position of the key bits to increase the complexity. Cloud security can be based on the proposed model for better protection from attackers.

4. Results and Discussion

The multi-tenant cloud API is designed with two servers; the main server and the web server. The database server or main server is applicable for communication inside the organization or an enterprise; only tokens are generated for internal communication. The database tables created in

MySQL contains the tenant information based on their role. The role is approved by ADMIN only.

4.1 *Test Case Scenarios*

The complete processing is done by integrating the algorithms in the web server interface. The proposed model tested for 161 end-users, 10 multi-tenants, 8 head-tenants and 3 enterprise levels.

Case 1: Creating multiple tenants in cloud architecture

The ADMIN is the first layer of the tenant category who is responsible to approve or reject the tenant's authentication and registration. The login credentials are shared on the email id of the authenticated tenant or the user. The credentials should be modified by the user once the user logins the page, as the credentials shared by the admin on the email id are the default credentials and should be changed for security reasons. Next layer is designed for data owners who are responsible to upload the data on the cloud server. Further it is divided into different user types; insider tenant, trusted tenant, partial tenant, and guest tenant. Depending on the role type, key expiration technique is followed.

Case 2: Uploading a file on cloud server on real-time

Admin1 is responsible to upload the data or files on the cloud environment. Suppose "sample4.txt" file needs to be uploaded. Clicking on "choose file" button navigates to the windows browser; the required file can be uploaded by clicking on "upload" button afterwards.

Case 3: Request for accessing a file.

Suppose user "user4" wants to access the uploaded file "sample4. text" by "admin1". Then "user4" first requests file access to the data owner, this request is received and approved by "admin1", the data owner, and then a key is generated using EHC algorithm since the category of "user4" is "Insider" user. For a partial user, BGV algorithm is used to generate the hybrid key. The snip of token/key received on the email id of the user as shown in Figure 1. This token or key is generated using EHC or BGV algorithm based on the category of the user.

Case 4: Self-key mutation in case of an attack

A total of 56 parity rules are applied randomly. These 56 rules are defined for 8-byte bit formation or 64-bit matrix formation. The original

key is mutated randomly 5 times which increases the complexity in guessing the key.

4.2 *Creating a multitenant environment on CloudSim and generating the keys using EHC algorithm*

Authentication and authorization of users is done using the credentials of the users and only after proper verification and validation of the credentials, the user can access the cloud. The bits required for PuK and PK and the time taken by these keys are evaluated.

Table 1

Decryption time variations among EHC algorithm and Hybrid EHC-BGV on 8-byte key

Plain Text Size	EHC(in ms)	Hybrid(in ms)
24 bits	2.1533	0.0525
72 bits	4.1264	0.1582
128 bits	6.2627	0.3305
256 bits	8.6085	0.6308
1024 bits	11.4445	0.8546
2048 bits	14.0209	2.407
4096 bits	16.8254	2.6386
8248 bits	19.4799	3.3513

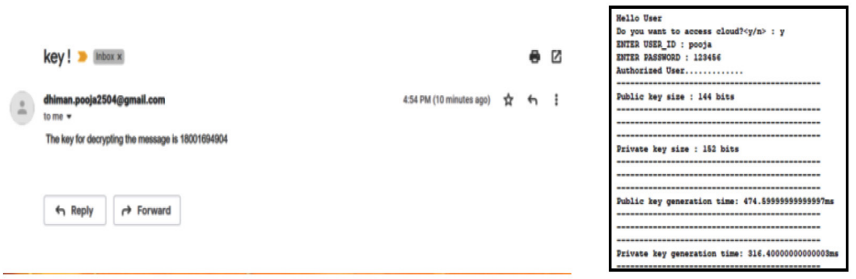


Figure 1

Test case for generating the token key which is received successfully on email id and Test case with EHC keys generation time and size [33]

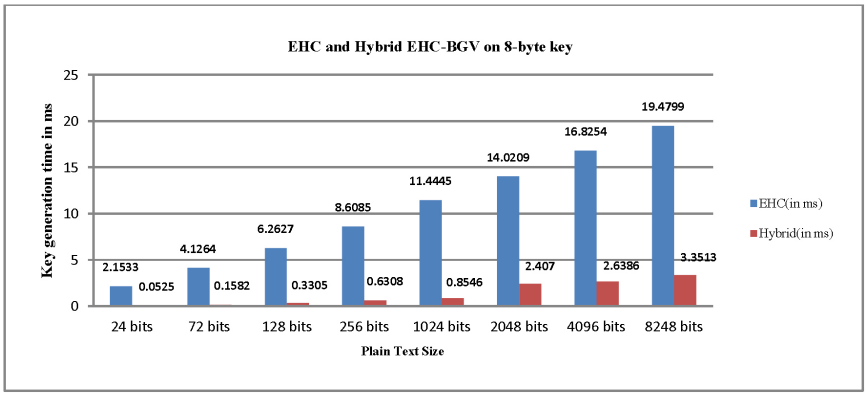


Figure 2

Decryption time variations among EHC algorithm and Hybrid EHC-BGV on 8-byte key

The Table 1 and Figure 2 shows the key generation time for EHC algorithm which is generated by implementing the EHC algorithm, the public key size is 144 bits and private key size is 152 bits and the key generation time for public key is 474.599ms and for private key, it takes 316.400ms (round off to 3 decimal places). The use of hybrid approach that integrates two FHE algorithms that is, EHC and BGC gives better results as compared with EHC algorithm alone. The decryption time in hybrid is very less as compared with EHC algorithm.

5. Conclusion

The algorithm is made to demonstrate the step-by-step process of the suggested methodology, which was tested with three test cases: the first involved creating multiple tenants in a cloud architecture; the second involved integrating real-time file uploading to a cloud server; the third involved initiating end user requests to access files; and the final one handled self-key mutation in the event of an attack. In order to prevent unwanted access, it was merged with key mapping parity rules, 14 multi-authentication, dependable, and non-dependable parameters tuned using initiated variables based parameters that were initiated and executed at the sender and recipient sides.

References

[1] G. Crihan, L. Dumitriu, and M. V. Crăciun, "Preliminary Experiments of a Real-World Authentication Mechanism Based on Facial Recognition and Fully Homomorphic Encryption," *Appl. Sci.*, vol. 14, no. 2, article 718 (Jan. 2024), doi: 10.3390/app14020718.

[2] E. Golimblevskaia, E. Shiriaev, and N. Kucherov, "Survey Software Implementations of Homomorphic Encryption Methods," in Advances in Automation II, RusAutoCon 2020, Lecture Notes in Electrical Engineering, vol. 729, A. A. Radionov and V. R. Gasiyarov, Eds., Cham, Switzerland: Springer, pp. (2021) [page numbers if known], do i: 10.1007/978-3-030-71119-1_59.

[3] R. Kanagavalli and S. Vagdevi, "Secured Data Storage in Cloud Using Homomorphic Encryption," *Int. J. Cloud Comput. Serv. Archit.*, vol. 9, no. 4, pp. 1–11 (2019), doi: 10.5121/ijccsa.2019.9401.

[4] I. Ahmad and A. Khandekar, "Homomorphic Encryption Method Applied to Cloud Computing," [Journal Name?], vol. 4, no. 15, pp. 1519–1530 (2014).

[5] B. Chen and N. A. Zhao, "Fully Homomorphic Encryption Application in Cloud Computing," (2014), [publication details not provided].

[6] W. Ding, Z. Yan, and R. H. Deng, "Privacy-Preserving Data Processing with Flexible Access Control," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 2, pp. 363–376 (2020), doi: 10.1109/TDSC.2017.2786247.

[7] S. V. S. Prasad and K. Kumanan, "Homomorphic Encryption Using Enhanced BGV Encryption Scheme for Cloud Security," *Int. J. Eng. Comput. Sci.*, vol. 7, no. 3, pp. 23785–23789 (2018), doi: 10.18535/ijecs/v7i3.22.

[8] P. V. Parmar, S. B. Padhar, S. N. Patel, N. I. Bhatt, and R. H. Jhaveri, "Survey of Various Homomorphic Encryption Algorithms and Schemes," *Int. J. Comput. Appl.*, vol. 91, no. 8, pp. 26–32 (2014), doi: 10.5120/15902-5081.

[9] L. Chen, Z. Tong, W. Liu, and C. Gao, "Non-interactive Exponential Homomorphic Encryption Algorithm," in Proc. 2012 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2012, pp. 224–227 (2012), doi: 10.1109/CyberC.2012.44.

[10] U. Shwetha, R. M. H., and M. June, "Securing the Data in Cloud Using Algebra Homomorphic Encryption Scheme Based on Up-

dated ElGamal (AHEE)," [Journal/Proc. info missing], vol. 6, no. 3, pp. 287–292 (2017).

[11] H. M. Al-Mashadi and A. A. Khalaf, "Hybrid Homomorphic Cryptosystem for Secure Transfer of Color Image on Public Cloud," *J. Theor. Appl. Inf. Technol.,* vol. 96, no. 19, pp. 6474–6486 (2018).

[12] G. Vnkv, S. Rao, and G. Uma, "An Efficient Secure Message Transmission in GJCST-E Network, Web Security," *GJCST-E Network Web Secur.,* vol. 13, no. 9 (2013).

[13] A. El-Yahyaoui and M. D. ElKettani, "Fully Homomorphic Encryption: State of the Art and Comparison," *Int. J. Comput. Sci. Inf. Secur.,* vol. 14, no. 4, pp. 159–168 (2016), doi: 10.6084/M9.FIGSHARE.3362338.

[14] B. S. Al-Attab, "Authentication Technique by Using USB Token in Cloud Computing," no. date, pp. 1–4 (2019).

[15] C. S. Saravanabhavan, K. Anguraju, M. Kannan, and P. Preethi, "Ensuring Efficient Data Storage Using Fully Mature Homomorphic Encryption Technique in the Cloud Environment," *Int. J. Recent Technol. Eng.,* vol. 8, no. 2, pp. 4820–4832 (2019), doi: 10.35940/ijrte.B2472.078219.

[16] S. Y. Ketki and R. R. Ingole, "Blockchain Technology in Cloud Computing: A Systematic Review," *Int. Res. J. Eng. Technol.*, pp. 1–43 (2018).

[17] A. Toumanari, "Efficient Ciphertext-Policy Attribute-Based Encryption Constructions with Outsourced Encryption and Decryption," (2021) [journal/conference info missing].

[18] P. J. Sun, "Security and Privacy Protection in Cloud Computing: Discussions and Challenges," J. Netw. Comput. Appl., vol. 160, p. 102642 (April 2020), doi: 10.1016/j.jnca.2020.102642.

[19] F.-J. Valera-Rodriguez, P. Manzanares-Lopez, and M.-D. Cano, "Empirical Study of Fully Homomorphic Encryption Using Microsoft SEAL," Appl. Sci., vol. 14, no. 10, article 4047 (May 2024), doi: 10.3390/app14104047.

[20] E. Bezuglova and N. Kucherov, "An Overview of Modern Fully Homomorphic Encryption Schemes," in Current Problems in Applied Mathematics and Computer Science and Systems, APAMCS 2022, Lecture Notes in Networks and Systems, vol. 702, A. Alikhanov, P. Lyakhov, and I. Samoylenko, Eds., Cham, Switzerland: Springer,

pp (2023). [page numbers if known], doi: 10.1007/978-3-031-34127-4_29.

[21] S. Dubey and P. K. Rai, "A Review of Cloud Service Security with Various Access Control Methods," *Int. J. Comput. Sci. Mob. Comput.*, vol. 10, no. 3, pp. 39–45 (2021), doi: 10.47760/ijcsmc.2021.v10i03.005.

[22] A. Singh, U. Chandra, S. Kumar, and K. Chatterjee, "A Secure Access Control Model for E-health Cloud," in IEEE Reg. 10 Annu. Int. Conf., TENCON 2019, vol. 2019-Octob, pp. 2329–2334 (2019), doi: 10.1109/TENCON.2019.8929433.

[23] A. Chhabra, "Access Control in Multi Tenant and Diverse Cloud," [publication info not provided].

[24] K. Sethi, P. Bera, A. Chopra, and B. K. Tripathy, "Integration of Role Based Access Control with Homomorphic Cryptosystem for Secure and Controlled Access of Data in Cloud," in Proc. ACM Int. Conf. Proc. Ser., pp. 194–199 (2017), doi: 10.1145/3136825.3136902.

[25] W. Ding, Z. Yan, and R. H. Deng, "Privacy-Preserving Data Processing with Flexible Access Control," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 2, pp. 363–376 (2020), doi: 10.1109/TDSC.2017.2786247. (duplicate of [6]).

[26] K. K. Hingwe and S. Mary Saira Bhanu, "Hierarchical Role-Based Access Control with Homomorphic Encryption for Database as a Service," in Adv. Intell. Syst. Comput., vol. 409, pp. v–vi (2016), doi: 10.1007/978-981-10-0135-2.

[27] Z. Yu, C. Z. Gao, Z. Jing, B. B. Gupta, and Q. Cai, "A Practical Public Key Encryption Scheme Based on Learning Parity with Noise," *IEEE Access*, vol. 6, no. c, pp. 31918–31923 (2018), doi: 10.1109/ACCESS.2018.2840119.

[28] Z. R. Saeed, "Improved Cloud Storage Security of Using Three Layers Cryptography Algorithms," vol. 16, no. 10, pp. 34–39 (2018), [publisher/title info missing].

Received November, 2024