OPEN ACCESS Freely available online

ISSN: 2090-2697
Journal of
Forensic Biomechanics

Research Article

The HRF Model Implemented Using Dynamic High-Order Hill Matrix Keys and the Rail Fence Model

Colin Chibaya[1*], Chipo Katsande[2]

[1]Department of Computer Science and Information Technology, Sol Plaatje University, Kimberley, South Africa; [2]Department of Computer Science and Information Technology, Manicaland State University of Applied Sciences, Mutare, Zimbabwe

ABSTRACT

The Hill model is a compelling asymmetric algorithm which depends on matrices as keys to achieve data security. Research done on the Hill model, often, restricted to the use of low order matrix keys because of the complexity of operations with high order matrices. This, however, expose the Hill model to basic brute force attacks. This article investigates the use of high order dynamically generated matrix keys selected at run-time. In addition, the original Hill model only uses the 26 alphabetic characters. We extend the character set supported by the Hill model to the 256 ASCII characters. On the other hand, the rail fence model is also a compelling transposition algorithm which generally supports a low character set. This article investigates the combination of the improved Hill model with the rail fence model towards a hybrid product Hill-Rail Fence (HRF) model. Ideally, product ciphers depict enhanced data security more than the sum of the individual securities of the component ciphers. To further complicate the product, encryption is completed over more than one round. We evaluate the computational performance of the HRF model against the original Hill model in terms of the execution time, CPU usage, memory demands, number of running threads, as well as the number of loaded classes. The simulated results portray an increased processing time in the HRF model with every increase in the order of the matrix key. Also, higher order matrices are highly complex to brute force attack as guessing the decryption inverse matrices is close to impossible. Although degraded performances are noted, the time required to, potentially, break the HRF model that uses high order matrices exponentially increases. Benchmarking the HRF model with the original hill model yielded that, although the HRF model takes more execution time and consumes more CPU time and memory, the statistical significance of the performance differences reported are negligible. It is, thus, worth enhancing the hill model to the HRF model.

Keywords: Hill cipher; Rail-fence model; Dynamic matrix key; Product cipher

INTRODUCTION

Owing to the rapid advancement of network communication and technology, data security is becoming a more pressing issue to handle. Cryptography is important for providing data security in such scenarios. In this context, cryptography is about the exchange of secret information through public channels. Most cryptography algorithms endeavour to achieve one or more of these five objectives: confidentiality, integrity, accountability, authenticity, and availability. Thus cryptography is an essential part of any effective information security system. It has become a basic building requirement in most computer security systems [1-4].

Cryptographic algorithms are classified into symmetric and asymmetric. Symmetric algorithms arise when two parties share a common secret key for both encryption and decryption. On the other hand, asymmetric algorithms use different encryption and decryption keys [5,6]. Often, asymmetric algorithms comprise a public and a private key. Public keys are unconditionally shared with all parties intending to share information. However, a private key remains known and useful only to the creator of the keys [7].

Correspondence to: Colin Chibaya, Department of Computer Science and Information Technology, Sol Plaatje University, Kimberley, South Africa, E-mail: colin.chibaya@spu.ac.za

Received: 23-Nov-2022, Manuscript No. JFB-22-18969; **Editor assigned:** 25-Nov-2022, Pre QC No. JFB-22-18969 (PQ); **Reviewed:** 12-Dec-2022, QC No. JFB-22-18969; **Revised:** 19-Dec-2022, Manuscript No. JFB-22-18969 (R); **Published:** 27-Dec-2022, DOI:10.35248/2090-2697.22.13.411

Citation: Chibaya C, Katsande C (2022) The HRF Model Implemented Using Dynamic High-Order Hill Matrix Keys and the Rail Fence Model. J Forensic Biomech.13:411

The Hill model is an asymmetric algorithm which uses linear algebra and modulo arithmetic on matrices [8]. Matrices are used to transform blocks of plaintext into blocks of cipher text. While a matrix key is used for encryption, its inverse is used for decryption, hence an asymmetric model [9]. Mostly, the Hill model users opt for manageable matrix keys of second, third, or fourth order [10]. The higher the matrix order, the harder it is to attack the cipher because, generally, finding the inverse of high order matrices is cumbersome. On the other hand, use of low order matrix keys has been brute force attacked [11,12]. In this article, we propose a mechanism for generating and using high order matrix keys created at run time. Selection of the pair of matrices to use as the encryption and decryption keys, at a time, is system-handled from a pool of randomly generated high order candidate matrix key pairs. The traditional Hill model uses a character set with the cardinality of 26 (alphabetic characters). Several works have attempted to increase this character set. We investigate the use on the ASCII character set which further complicates brute force attack.

The rail fence model, on the other hand, is a transposition algorithm that rearranges the characters of the plaintext [13].The characters of the plaintext are written diagonally downwards on successive "rails" of an imaginary fence. The characters of the plaintext are then read off row by row from top to down [14]. The positions of the characters of the plaintext are interchanged using a rail fence key which is just the number of rails [15].

Product ciphers are compelling as they are constructed by combining independent ciphers. The basic operations completed when ciphers are combined may include permutations, transpositions, translations, linear transformations, arithmetic operations, modular multiplication, or simple substitutions [16]. Combining two or more models together results in a cipher that is more secure than the individual component ciphers [17,18]. In this case we dynamically incorporate principles of the rail fence transposition model principles into the Hill model or *vice versa*. The sequencing of the combination of the two models is also system-handled.

Statement of the problem

We can rephrase the problem addressed in this study to an investigation of the development of a hybrid product model (HRF) by integrating the Hill model that uses high order dynamically generated matrix keys and the rail fence model. Four sub questions drive this work as follows:

1. How do we implement a Hill model that uses dynamically generated high order matrix keys, supporting the ASCII character set?
2. How do we implement the Rail Fence model that, also, supports the ASCII character set?
3. How do we integrate these two models into a product HRF model?
4. What are the relative performances of the HRF model against the original Hill model?

Successful answers to these questions may bring about new content to the body of knowledge that may be useful to new entrants in the field of cryptography. In addition, businesses may consider the use of this cheaper, simpler, and affordable alternative. More importantly, new avenues for research are connoted in this study.

Overview

The rest of the article proceeds as follows: Section 2 presents work related to attempts that have been made in the past to improve security in particular ciphers through dynamism, hybridization, and productization. In section 3, we bestow the methods embraced in this study, emphasizing the computational design of the Hill model, Rail Fence model, and the HRF model. The results related to the relative performances of the models under study are presented in section 4, along with the discussions. We conclude the work in section 5, highlighting the key conclusions, the main contributions, and direction for future work.

Related work

Polyalphanumeric ciphers such as the Hill model are preferred in cryptography for their strength [19]. That polygraphicness, and use of linear transformation stands out. Traditionally, the Hill model assigns a number to each letter. For example, a=0, b=1,..., z=25. Flipping this view to the use of ASCII codes is innovative. Basically, characters can be represented by their ASCII codes.

While picking a matrix key is easy, finding that matrix whose inverse can be the key to decrypt ciphertext back to plaintext is a daunting task. The mathematics for finding k-1 is complex and hard [20]. What mathematics can we use to find k-1 in equation 1 below? Precisely, the equation has two unknowns.

P=k-1C mod size of (character set)

Such complexity even increases when the matrix key is of high order [21]. In addition, operating in mod 256 (the size of the character set) further hardens the process of brute force attacking this inverse matrix. Development of substantial adaptations of the Hill model around its matrix key is a common area of study [22,23]. Most works attempt to avoid linearity during matrix transformation by devising different methods of generating and selecting matrix keys. For example, random matrices that use random permutations have yielded plausible results. However, low order matrix keys were prevalently used [24,25]. Although determining matrix keys that are invertible is hard and penalize the use of low order matrix keys in favour of high order matrix keys for the strength of the model [26,27].

Similar attempts to twick the matrix key are observed in the work of who suggested the use of a different matrix key for each block. However, these keys remained of low order and easy to attack [28]. Introduction of an Affine Hill cipher variant was more outstanding. However, its use of random numbers together with recursive Hashed Message Authentication Code (HMAC) complicated the scheme [29].

Suggestions to opt for high order matrix keys are not new because it is confirmed that the higher the order of the matrix key, the stronger the security aspect thereof. For that, our HRF model can use matrix keys of any desired order. Robustness and further strength arise from the proposed use of pools of candidate matrix keys from which a pair is selected in every run.

Several works focused on matrix key for affine and polynomial transformation [30]. However, most of these extensions remained prone to cryptanalytic attacks [31]. Others attempts focused on the use of two matrix leys at a time. This approach was quite innovative and inspiring. Similar perspectives on strengthening the Hill model by using multiple keys were also reported in yielding much stronger product models. In some case, a key bunch matrix scheme has been proposed [32,33]. The technique represented several keys in the form of a matrix, called a key bunch matrix (a matrix of matrices). However, the ciphertext thereof became so difficult to decipher. Leading to information loss although these modifications complicated the key matrix, and the entire encryption process further, the model became more and more expensive in terms of the CPU time and memory demands. The design of the HRF model emphasizes simplicity and affordability.

The Rail Fence model, on the other hand, has also seen a few modifications with time. It has been combined with many other models to bring about more complicated systems to break [34,35]. In this model, the characters of the plaintext are written downwards and diagonally on successive "rails" of an imaginary fence, and then upwards when it reaches the bottom of the rail. The characters of the plaintext are then read off as a sequence of rows. The positions of the characters of the plaintext are interchanged using a rail key. Alone, the Rail Fence model is weak because an attacker may learn the key by merely splitting the code into different number of lines and reading the code into zigzag pattern. Such insecurity is dissolved when the Rail Fence model is combined with other ciphers into product ciphers [36]. In this study, we investigation the combination of a hybrid Hill model and a Rail Fence model.

Hybridization of substitution and permutation ciphers is also quite popular in the literature [37]. However, little research is available on the integration of the modified Hill model and the Rail Fence model. Combining substitution and permutation ciphers produces cipher text that is hard to crack. Although the integration of the Hill and Rail Fence models has been tried before, the combined variants were not designed the same way as ours. One approach considered three phases where the first part encrypted plaintext using the Caesar model [38]. The second phase accepted the output of the first phase as input and then applied the Hill encryption technique. The output of the second phase was then applied to the Rail Fence technique to generate the final cipher text. However, having that stipulated sequence of combination of the three models simplified possible attacks. We propose dynamic generation of the high order Hill matrix keys, use of the ASCII character set in both the Hill and Rail Fence models and propose the integration of the two models using run time generated sequence of operations, into a HRF model [39]. This, to the best of our knowledge, is a creative intervention worth pursuing in the body of knowledge.

METHODOLOGY

To complete this study, we embrace the design science research paradigm which purports spiral development of deductive evidence supported by quantitative outcomes. The HRF model is built based on system-generated high order Hill matrix keys and the ASCII character set. Rail-fence transposition principles are then incorporated in unpredictable sequences. The approach followed in this study is highly experimental, allowing spiral replication of the simulations towards establishing the causal properties of the HRF model.

Development of the HRF model has four stages. The first stage is about the generation of high order matrix keys and their corresponding inverses and throwing these in a pool of candidate keys for different runs. The second stage is about the generation of the Rail Fence model to, also, support ASCII characters. In the third stage, plain text is encrypted using the Hill model or Rail Fence technique based on the dynamically selected sequence. The final phase is about applying the other model that was not used in the second stage.

The Hill model takes three inputs, the order n of the square matrix key required, and parameter m which controls the number of matrices to be generated and placed in the pool for selection, as well as the factor r which indicates the number of encryptions rounds to consider. Specifying the order of the matrix key gives the user the flexibility to go for any order of choice. The matrix key terms would be any value in the range of 0 to 255, since we consider all ASCII characters. The system validates the generated matrix key for invertibility. The first m invertible matrix keys that are generated are placed in a pool for consideration in the current encryption call, where each call comprises r encryption rounds. Therefore, r pairs of matrix keys are picked from the pool per call. For example, if the user chose r=6, the Hill component of the HRF model will dynamically pick six matrix keys and their inverses and encrypting the message over six iterations using a different set of matrices in each round. We assume that the decryption keys are securely communicated.

Hill encryption is achieved by dividing the plaintext into blocks of lengths equal to the dimension of the matrix keys. The message is converted into column vectors before they are multiplied by the picked invertible matrix keys. We indicated that the yielded answers are re-encrypted over the selected number of rounds before the outcome is stored in modulo 256 [40]. The Rail Fence model would be applied thereafter if the selected sequence said so otherwise, the Rail Fence component would be used first before the Hill process. The Rail Fence component takes one main input, the key which determines the number of rows of the rail matrix.

Upon completed encryption, decryption is merely the reverse. The last-in-first-out approach is used. If encryption started with the Hill model, then the Rail Fence, decryption would start with the Rail Fence and then the Hill model. Keys are also used in the reverse order.

We administered several experiments to test the performance of the HRF model against the original Hill model in term of CPU

usage, memory usage, loaded class files and number of live threads. Potentially, these measures sufficiently unearth the relative validity of the HRF model.

The key data collected in this study is simulated performances. The experiments are repeatedly run to achieve centrally placed performances. Thus, central tendencies and any dispersions in the assessed performances are reported. At the end, we compare the performance of the HRF model to those of the original Hill model with the goal of verifying the validity of the HRF alternative. To achieve this, the Java Virtual Machine Monitor was used to profile and monitor how resources are used.

RESULTS AND DISCUSSION

The main intention was to measure the performances of the HRF model against that of the original Hill model in terms of CPU time and memory demands. Successful generation of run time matrix keys of high order was observed. For reproducibility and as proof of concept, we limited the order to 10. Both the Hill and Rail Fence supported the ASCII character. Invertibility tests on each candidate matrix key allowed the generation of the associated inverse matrix keys for decryption. Ideally, pairs of the set of matrix keys to use over the selected rounds per call were successfully established.

The Hill model and the Rail Fence model were each independently tested for functionality when the ASCII character set was used. The HRF model was similarly evaluated with possible run combinations such as: Hill first, then the Rail fence, or *vice versa*. Such a process is denoted as E (H (E(RF, rail Key),matrix key)) or *vice versa*.

Figure 1 compares the Heap memory usages of the original Hill model and those of the HRF model. Visually, although the Hill model outperforms the HRF model. It is appealing to notice that both models converge at some point. Precisely, the Hybrid HRF model utilized about 65 M (65153 kilobytes) where the original Hill model alone used 27 M (27551 kilobytes). The hybrid HRF model also used non heap memory of 15 M (15057 kilobytes) where the original Hill model used 14 M (14 320 Kbytes). We can explain the demand for higher memory requirements by the HRF model than the original Hill model as related to the HRF comprising two models in one. That alone implies more computations. However, the observed difference in memory demands is insignificant. The HRF model only used about 18% of the total memory allocated by JVM, which is not much. Rather, both models can be regarded as effective and efficient.

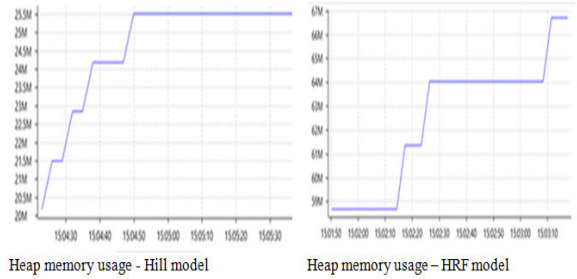

Figure 1: Comparison of Heap memory usage.

On the other hand, Figure 2 compares the CPU time required to run each model. While the Hill alone is heavier early in simulation, the performances of the HRF model depend on the sequence of events in each call. These differences are respectively very minor because the hardware used comprised four core processors.

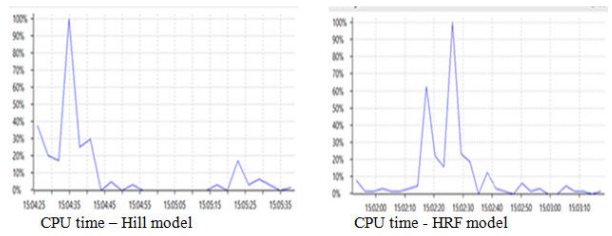

Figure 2: Comparison between the CPU times.

The live threads counts are very close to each other in pattern (Figure 3), suggesting replaceability of one by the other. The same and constant number of threads noticed in both models is because the tests were run on the same machine. A high number of threads imply an increase in power utilization due to the processing power required by each thread. In this case, the thread count for both algorithms is relatively low.

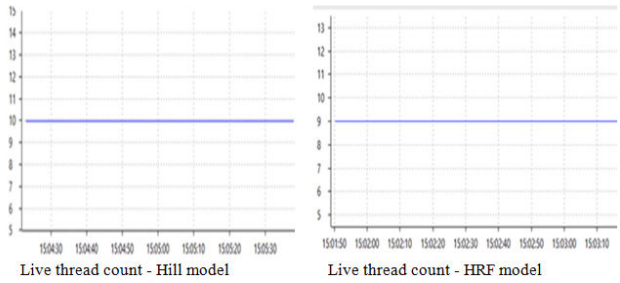

Figure 3: Comparison of live thread count.

In addition, each model exhibited resembling or equivalent loaded files, also suggesting convergence of both models (Figure 4). The HRF model loaded about 1696 class files where the original Hill model loaded 1651. Loaded class files have an impact on memory utilization. Thus, whenever a class file is loaded, it is allocated non-heap memory. The JVM uses non-heap memory to store class level information. In this case, the HRF model loaded about 15 M of non-heap memory while the original Hill model used 14 M. The high number of loaded files of the HRF model explains why non-heap memory consumption of the HRF model is slightly higher. However, on a bigger note, these performances are similar.

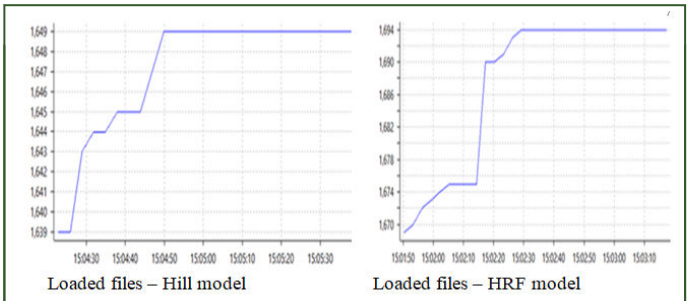

Figure 4: Comparison of the number of loaded files.

CONCLUSION

A HRF model was built and tested. The model depicted relatively acceptable performances benchmarked against the performances of the original Hill cipher. Notably, the future of data security lies in hybridized models mixed in unpredictable sequences. Hopefully, complex models and more sophisticated mechanisms for managing encryption keys will develop further. For example, dynamic selection of encryption keys at run time is compelling and innovative, hence more preferred. Similarly, sole models are easier to break. This study advocates for product hybrid models dynamically mixed at run time.

CONTRIBUTIONS

The HRF models come with three notable worthy benefits as follows:

- The work creates a baseline view upon which further hybrid encryption models and innovative research may be built on. Besides adding content, this creative intervention adds relevant literature to the body of knowledge.
- The work develops new crypto-views based on extended and mixed models in the field of information security. The outcomes of this study can be used to solve real world problems.
- The work presents profound, substantial educational views for novice crypto model developers. This work may be a good starting point for most novices' practitioners.

FUTURE WORKS

We still hope to further hybridize the Hill model by incorporating more independent ciphers to it. Hopefully, the envisaged mix will be hard to predict. The sequence of such a mix will remain dynamic. It would be daunting for one to guess the included models, predict the sequencing of the included models, as well as to prophecy all the keys involved before one can break these models. Hopefully, data confidentiality, data integrity, data authenticity, and accountability may be achieved in one goal.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable

CONSENT FOR PUBLICATION

All authors have read and agreed to the published version of the manuscript. All authors have consented to the publication of this manuscript.

AVAILABILITY OF DATA AND MATERIALS

All the data supporting the reported results is included in the manuscript. There are no additional materials required for this study.

REFERENCES

1. Nie T, Song C, Zhi X. Performance evaluation of DES and Blowfish algorithms.2010;1-4.
2. Kaur G, Malhotra S. A hybrid approach for data hiding using cryptography schemes. 2013;4(8).
3. Saper N. International cryptography regulation and the global information economy.
4. Annalakshmi M, Padmapriya A. Zigzag Ciphers: A Novel Transposition Method. 2013.
5. Paar C, Pelzl J. Understanding cryptography: a textbook for students and practitioners. 2009.
6. Luckett WM. Cellular automata for dynamic S-boxes in cryptography. 2007.
7. Kareem, S. Hybrid public key encryption algorithms for college of engineering hybrid public key encryption algorithms.2015.
8. Toorani M, Falahati A. A secure variant of the Hill cipher. 2009;313-316.
9. Qasem MH, Qatawneh M. Parallel hill cipher encryption algorithm. Int J Comput Appl.2018;179(19):16-24.
10. Stallings W. Ccryptography and network security: Principles and practice. fifth edition.2010.
11. Huang N. An enhanced hill cipher and its application in software copy protection. J Netw. 2014;9(10):2582.
12. Ismail IA, Amin M, Diab H. How to repair the Hill cipher. J Zhejiang Univ Sci. 2006;7(12):2022-2030.
13. Saini B. Modified caesar cipher and rail fence technique to enhance security. Int J Trend Res Dev. 2015;2(5).
14. Siahaan AP. Rail-fence-cryptography-in-securing-information.2017.
15. Aaref AM. A new cryptography method based on hill and rail fence algorithms. 2017;10(1):39-47.
16. Biryukov A. Encyclopedia of cryptography and security. 2011.
17. Singh A, Nandal A, Malik S. Implementation of caesar cipher with rail fence for enhancing data security. Int J Adv Res Comput Sci Softw Eng. 2012; 2(12):78-82.
18. Dunkelman O, Biham E. Techniques for cryptanalysis of block ciphers.
19. Menezes B. Network security and cryptography, 1st edition, cengage learning.
20. Acharya B, Rath GS, Patra SK, Panigrahy SK. Novel methods of generating self-invertible matrix for hill cipher algorithm.
21. Mahmoud AY, Chefranov AG. A hill cipher modification based on eigenvalues extension with dynamic key size hcm-exdks. Int J Comput Netw Inf Secur. 2014;6(5):57-65.
22. Acharya B, Rath GS, Patra SK, Panigrahy SK. Novel methods of generating self-invertible matrix for hill cipher algorithm.
23. Sastry VU, Janaki V. A modified hill cipher with multiple keys.
24. Saeednia S. How to make the hill cipher secure. Cryptologia. 2000;24(4):353-360.
25. Lin CH, Lee CY, Lee CY. Comments on Saeednia's improved scheme for the Hill cipher. J Chin Inst Eng. 2004;27(5):743-746.
26. Thangarasu N, SelvaKumar AL. Encryption using lester hill cipher algorithm. Int J Innov Res Adv Eng. 2015;2(12):13-17.
27. Siahaan AP. Three-pass protocol concept in Hill Cipher encryption technique.2016.
28. Rangel-Romero Y, Vega-García R, Menchaca-Méndez A, Acoltzi-Cervantes D, Martínez-Ramos L, Mecate-Zambrano M, et al. How to repair the hill cipher. J Zhejiang Univ Sci. 2008;9(2):211-214.
29. Keliher L, Delaney AZ. Cryptanalysis of the toorani-falahati hill ciphers.2013.

30. Murray E. Hill ciphers and modular linear algebra.1999.
31. Siahaan AP. Application of hill cipher algorithm in securing text messages.
32. Klima R, Klima RE, Sigmon N, Sigmon NP. Cryptology: Classical and modern. 2018.
33. Forouzan BA, Mukhopadhyay D. Cryptography and network security. 2015.
34. Thilaka B, Rajalakshmi K. An extension of hill cipher using generalised inverses and mth residue modulo n. Cryptologia. 2005;29(4):367-376.
35. Gupta I, Singh J, Chaudhary R. Cryptanalysis of an extension of the hill cipher. Cryptologia. 2007;31(3):246-253.
36. Sastry VU, Samson C. A generalized hill cipher involving different powers of a key, mixing and substitution. Int J Adv Res Comput. 2012;3(4).
37. Sastry VU, Shirisha K. A novel block cipher involving a key bunch matrix. Int J Adv Res Comput.2012;9(75):8887.
38. Annalakshmi M, Padmapriya A. Zigzag ciphers: A novel transposition method.
39. Ruprah TS. Advance encryption and decryption technique using multiple symmetric algorithm. J Inf Secur Res. 2016;7(2).
40. Singh A, Nandal A, Malik S. Implementation of caesar cipher with rail fence for enhancing data security. Int J Adv Res Comput Sci Softw Eng. 2012; 2(12):78-82.