Proc. of the International Conference on Electrical, Computer,
Communications and Mechatronics Engineering (ICECCME) 7-8
October 2021, Mauritius

# A novel secure approach of selecting the key or nonce by Guessing in Cryptography

1st R S

***Abstract*—The world of Cryptography and Encryption is evolving rapidly with new technologies being upgraded or invented. The area of Blockchain, Quantum computing, normal sharing of data over the network all uses the techniques of encrypting the data using some sort of keys. As the Machines and technology are becoming more and more powerful, legacy way of handling the communication needs a secure approach. This paper propose an idea of guessing the nonce or key, in such a way that it would evolve with the machines and technology. The Idea of Proposed adaptive algorithm works on challenges and using different challenges, the key is guess. There are three variations of this approach Direct, Separation and Pooling. The Messages are encrypted from the pooled of solved challenge keys and using the proposed techniques, nonce or key is selected mutually. Using Poission distribution, it is found that the identification of key becomes more robust as the level of communication increases. This Methodology allows the flexibility to change or make the challenge more complex depending upon their needs, which indeed increase the probability of guessing. This can also solve the problem of Denial of service(DOS) and spamming. In future of advance computations and development, this approach will be helpful in making the system secure.**

***Index Terms*—Cryptography;Security;Algorithm**

## I. Introduction

With evolution of data encryption and encoding, security of data is guaranteed by an algorithm which do this change over communication channel. Complete process if breakdown, comes to the point where majority of algorithms work with Prime numbers [1], [2], due to factoring property of prime number. Major algorithms like RSA [3], [4], Diffie-Helman [5], [6] are used in majority of present application. The application of Blockchain [7]–[9], highly used this type of encryption technique and different products like Hyperledger [10], [11], Ethereum [12], [13] rely with an idea of key exchange using above techniques. With the Quantum theory of particles, theoretically it can be proved that compromising of key can be done easily within targeted time. As urge to get Quantum supremacy [14], [15] will make things more and more loose from legacy type of security mechanism. There is a requisite of some adaptive methodology which can evolve with complexity of problem and compromising of key should be not possible. Depending upon application the methodology should be adjusted and should provide opportunity for developer to make complexity as per requirement of the application.

## II. Proposed Methodology

The proposed algorithm is random based key selection mechanism which uses bruteforce guessing mechanism to guess the common key or nonce, for further safe communication by encrypting it. Any key based encryption technique can be used to encrypt the data, but the selection of key is adaptive and can be made complex depending upon the application. Three different variation of algorithm has been described in this paper

- Direct approach
- Pooling of hashes-Seperation of Message and Hashes
- Complex direct approach (variation of Direct approach)

Algorithm- Direct/Complex direct approach

1) Alice throws Challenge
2) Bob solve challenge and select one of random key
3) Shares hash of message along with Encrypted message using random key. For complex approach the sharing is done at N level of communication.
4) Alice decrypt message using all the key from the shared challenge.
5) Alice match the hash with the decrypted Messages Hash. If match found then the key is guessed correctly.

Alice may speedup the computation by storing the decrypted messages with all the potential keys and finally select the valid one when it is found at the N communication step. Though the space complexity would be increased but the computation would be faster.

Algorithm- Pooling of hashes

1) Alice throws Challenge
2) Bob solve challenge and select one of random key
3) At N level of communication, Alice decrypt using the keys from the challenge and shares pool of hashed deccrypted Message
4) At N+I Level of communication, Bob select the valid hashed Message and returns back.
5) Alice compare and identify the valid key.

Below is complete detail explaination of the working of different algorithm with an Example.

### *A. Approach one : Direct*

#### *1) PHASE 1 : Challenge:*

The Alice creates a challenge that use 3 bits and select the bits which are having 2 zeros in them so out of 8 combination the valid are 3 in the above case

001, 010 ,100 are the 3 valid cases

At Bob side when challenge is received he also calculate the 3 cases and select any one from it (this is random) and encrypt the message with that key

978-1-6654-1262-9/21/$31.00 © 2021 IEEE

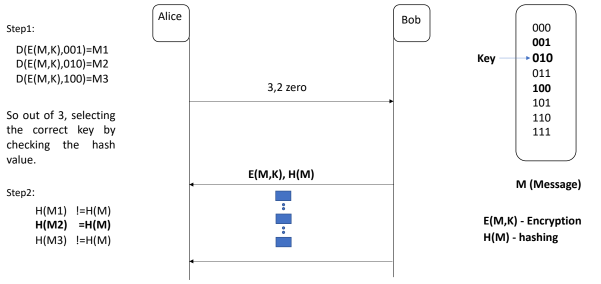


Fig. 1: Variation1: Direct approach for mutually selecting key

E(M,k) ⇒ The message is encrypted with the key which only bob know from the challenge (provided from the alice) , say key selected is 010.
Infact, the Bob can pass the challenge to Alice while eliminating the process of alice sending the challenge. (depending upon the business case.)
H(M) ⇒ this is the Hashed of the message passed along with the encrypted message

### 2) PHASE 2 : Analysis:

*Step1* :
As Alice has the 3 valid case it apply the decrypting logic on the message which is provided by bob and will get the 3 possible messages
D(E(M,k),001) ⇒ decrypting the message with the key 001. That is,

$$\begin{gathered} D(E(M,k),001)=M1 \\ D(E(M,k),010)=M2 \\ D(E(M,k),100)=M3 \end{gathered}$$

*Step2*:
Alice has to guess right message so it will use the hash and apply it to find which is the same as provided by the bob.So, H(M1) and H(M3) will not match. But H(M2) will match with the H(M).
Now Alice know the M2 has been generated using the key 010. So, at last the key is guessed correctly. And hence further communication is done. Here the computation was just 3 comparisons.
**Note:** To make it more complex,Bob can send the hash in second request so the new H(M+A), Where A is the second message now, this will make more complex to guess the right key. As the comparisons will be From [M1, M2, M3] and [A1, A2, A3]. Total 9 comparisons is needed. So, depending upon the requirement the complexity can be easily tuned. Depending upon the challenges the concept can be rotated.

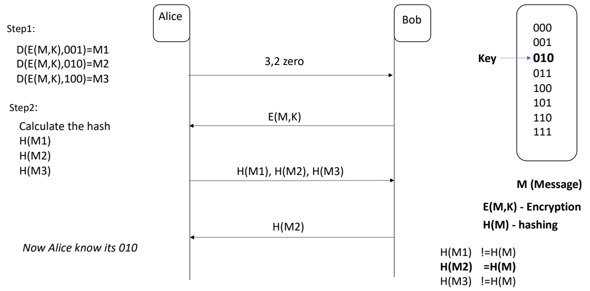


Fig. 2: Variation2: Pooling of hashes-Seperation approach for selecting key

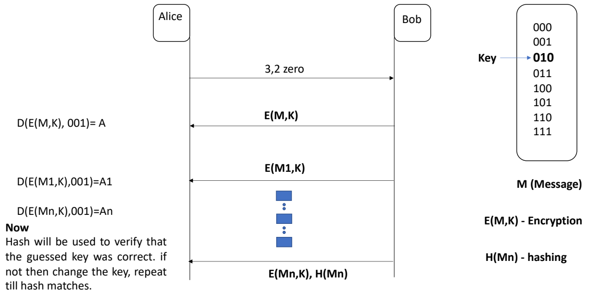


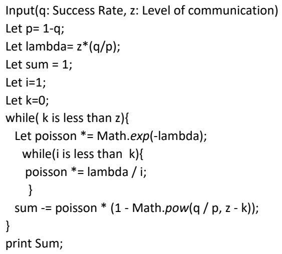
Fig. 3: Variation3: Complex direct approach

### B. Approach Two : Pooling of hashes-Seperation of Message and Hashes

With reference to figure 2, challenge part of of this approach is same. The difference is rather than sending hash from Bob to Alice. Alice sends the cumulative hashes, then bob selects the valid one and sends back to the Alice. Alice has the one to one mapping of messages formed with associated key, She now knows the key and further can use it to for communication.
**Note:** The complexity can be increased or decreased depending upon the cummulative sharing of the hashes, which may be shared in the form of *clusters or groups* at different stages of communications

### C. Approach Three : Complex direct approach

In this approach3, After the challenge part, the message is passed from the Bob to Alice with the bobs random key, At Alice side, the decrypted message is stored, Alice may use any random key from the set for decryption so to avoid the computation cost at the Nth level. Hence this approach can further be optimized. At certain level bob will send the hash of the Nth message to validate the encryption, if the hashing matches then the guessed key by Alice is correct if not then select another key and repeat till the hash matches. This will increase the computation cost, but guarantees that the key cannot be compromised.



```
Input(q: Success Rate, z: Level of communication)
Let p= 1-q;
Let lambda= z*(q/p);
Let sum = 1;
Let i=1;
Let k=0;
while( k is less than z){
  Let poisson *= Math.exp(-lambda);
    while(i is less than  k){
     poisson *= lambda / i;
      }
  sum -= poisson * (1 - Math.pow(q / p, z - k));
}
print Sum;
```

Fig. 4: Psuedo code of attacking probability

## III. Results and Discussion

As the communication level increases the complexity of the algorithm increase which in general decrease the probability of compromization of the key. Applying the Poisson distribution its clear that the moment any one of the communication is missed the chances of identifying the key decreased drastically.

The results from the figure showing code 4 are quite promising, with just 6 level of communication and the success rate of attacker at 30 percent, we get total compromising ratio as 0.13. Further, if we increase the communication to just double that is 12, then the probability of compromising the key reduced to 0.02. In future, using the proposed approach the security can be enhanced by mixing it with different practical application.

## References

[1] H. Riesel, *Prime numbers and computer methods for factorization*. Springer Science & Business Media, 2012, vol. 126.

[2] K. Pavani and P. Sriramya, "Enhancing public key cryptography using rsa, rsa-crt and n-prime rsa with multiple keys," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. IEEE, 2021, pp. 1–6.

[3] B. Kaliski, "The mathematics of the rsa public-key cryptosystem," *RSA Laboratories*, 2006.

[4] S. Naz and S. U.-J. Lee, "Why the new consensus mechanism is needed in blockchain technology?" in *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, 2020, pp. 92–99.

[5] W. Diffie and M. Hellman, "New directions in cryptography, ieee transactions on information theory, v. it-22, n. 6," 1976.

[6] F. Bao, R. H. Deng, and H. Zhu, "Variations of diffie-hellman problem," in *International conference on information and communications security*. Springer, 2003, pp. 301–312.

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[8] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Transactions on Industrial Informatics*, 2020.

[9] T. Meng, Y. Zhao, K. Wolter, and C.-Z. Xu, "On consortium blockchain consistency: A queueing network model approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 6, pp. 1369–1382, 2021.

[10] C. Cachin *et al.*, "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310, no. 4. Chicago, IL, 2016.

[11] H. Javaid, J. Yang, N. Santoso, M. Upadhyay, S. Mohan, C. Hu, and G. Brebner, "Blockchain machine: A network-attached hardware accelerator for hyperledger fabric," *arXiv preprint arXiv:2104.06968*, 2021.

[12] Y. Huang, B. Wang, and Y. Wang, "Research and application of smart contract based on ethereum blockchain," in *Journal of Physics: Conference Series*, vol. 1748, no. 4. IOP Publishing, 2021, p. 042016.

[13] G. Estevam, L. M. Palma, L. R. Silva, J. E. Martina, and M. Vigil, "Accurate and decentralized timestamping using smart contracts on the ethereum blockchain," *Information Processing & Management*, vol. 58, no. 3, p. 102471, 2021.

[14] J. Martinis, "Quantum supremacy using a programmable superconducting processor," *Bulletin of the American Physical Society*, 2021.

[15] C. Portmann and R. Renner, "Security in quantum cryptography," *arXiv preprint arXiv:2102.00021*, 2021.