Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-021-00613-1

ORIGINAL RESEARCH

Group security using ECC

Purna Chandra Sethi[1] · Neelima Sahu[2] · Prafulla Kumar Behera[2]

Received: 26 September 2020 / Accepted: 11 January 2021

Abstract Nowadays security is main issue during transmission of data. Among many cryptographic methods, ECC is the public key asymmetric cryptosystem which provides faster computation over smaller size in comparison to other asymmetric key cryptosystems. In this paper, we have proposed a group security algorithm using the ECC cryptography algorithm. The group security is applied to ECC in terms of m-gram selection called ECC m-gram selection. Due to the group security implementation in terms of common grams, processing speed will be faster in comparison to individual item security. We have also made the comparison study between the traditional ECC algorithm with the proposed group security algorithm using generalized frequent-common gram selection for depicting lesser time requirements to achieve better security for the whole process.

Keywords GFGS · m-gram · ECC · EMS · Group security

1 Introduction

The use of computers as well as computer-related activities over the internet growing exponentially day by day. With the growth of internet users, the threats, as well as the attacks to personal and organization information, is also growing exponentially. According to a survey done by ITU at the end of 2019, 53.6% of global populations are using the Internet which is less than 1% in 1999 for their day to day activities. These increasing populations towards Internet-related activities attracted the cybercriminals for imparting various crimes. To protect personal data as well as organization information, large numbers of researchers are motivated towards research in information security that not only includes data security but also network security. According to industrial Ethernet book, the global revenue for protecting from industrial cyber security threats and attacks are becoming doubled between 2013 and 2019 (approximately 600 million dollar to 1200 million dollars). To provide information security, various symmetric keys, as well as asymmetric key cryptography algorithms, are used by researchers. In case of symmetric key cryptosystem, a common private key is shared among the sender and receiver with proper security whereas, in case of asymmetric key cryptosystem, a public key is used by the sender for encryption and a private key owned the receiver for decryption. Due to two independent keys implementation for cryptography algorithm implementation, asymmetric key cryptography algorithms are becoming more popular with respect to private key cryptography algorithms [1].

There are several asymmetric key cryptography algorithms present which differ by their functionality. The algorithms are defined according to the numbers of bits used for each key. If a cryptography algorithm uses less number of key bits then and incurs longer time for

✉ Purna Chandra Sethi
purna.sethi@gmail.com

Neelima Sahu
sahu.neelima45@gmail.com

Prafulla Kumar Behera
p_behera@hotmail.com

[1] Department of Computer Science, Rama Devi Women's University, Bhubaneswar, Odisha, India

[2] Department of Computer Science and Applications, Utkal University, Bhubaneswar, Odisha, India

Published online: 28 February 2021

Table 1 Differentiation of key size according to experimental complexity [4]

Symmetric scheme (key size in bits)	ECC-based scheme (size of n in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15,360

decryption, then it is becomes more secure in comparison to other cryptography algorithms that involve more number of key and requires nearly same or more time for decryption. Each cryptography algorithm has its pros and cons and each has its application field [1]. To provide enhance the security features, many researchers proposed different hybrid cryptography algorithms [2]. In contrast, we have proposed a technique called group security in which the cryptography algorithm is applied to a group of items instead of individual ones. The algorithm applied to a set of suspicious elements that undergoes security algorithm implementation. The suspicious elements are generated using generalized frequent-common gram selection (GFGS). The suspicious elements are the set of elements which are frequently occurring among the items to be stored [3]. Since ECC algorithm provides better security with less key size in comparison to various famous cryptography techniques namely RSA, AES, DES, etc. [4], are used in this research work.

The whole research paper organization is specified as: Sect. 2 deals with the survey of literature containing a brief insight into the research work made by different researchers various popular cryptography algorithms for achieving information security. Section 3 focuses on a brief overview of the GFGS algorithm which is used in our research work. Section 4 provides a basic idea about ECC and its implementation steps. The proposed work is explained in Sect. 5 of this paper. Section 6 contains the proposed algorithm and its description. The experimental result is provided in Sect. 7 followed by performance analysis of the proposed method in Sect. 8 of the paper. The conclusion is specified in Sect. 9 followed by future scope in Sect. 10 of this paper.

2 Literature overview

The current day to day activities starting from entertainment to business is highly dependent on the internet. All these activities are made over the network. According to a survey done by Hootsuite in January 2020, out of 7.76 billion of the total population, 5.18 unique mobile phone users are there which nearly 67% of the total population is. Among the mobile phone users, 4.54 billion are using the internet which is nearly 59% of the total population and approximately 3.81 billion are active social media users which are nearly 49% of the total population. During this Covid-19 pandemic, the online and digital activities over the internet are increased significantly. Online video conferencing, on classes for education, social media, online video calls, online transaction, etc. were became highly adopted by almost all users. Due to all these activities, the information is becoming public and hence the chance of accessing information increased significantly. Hence to secure these data, cryptography algorithm implementation is highly demanded. Cryptography algorithms are mainly used to achieve confidentiality in terms of hiding the information from unauthorized access, integrity in terms of protection from unauthorized change, authentication in terms of ownership for accessing the data and non-repudiation in terms of restricting the users from deny in future from sending or receiving message. There is large number of security algorithms present which differ by the scalability, encryption, and decryption speed and security feature. Since the number of internet users increases exponentially, the maintenance, as well as processing time for cryptography algorithms, are comparatively high. So in [9], the authors proposed an efficient method for managing the traffic incurred during networking considering bandwidth on demand approach in a run time environment for faster processing and performing better utilization of bandwidth. Paper [10] is the extension of [9] which not only provides faster data transmission but also provides security to information using double ECC algorithm.

The efficiency of cryptography algorithm depends on many factors out of which length key plays a vital role. The key is an important factor for cryptography which is required for encryption and decryption. But as the key size increases, it becomes more difficult to break the algorithm. Hence, the cryptography algorithm that involves lesser key size and providing better security is becoming more popular. Out of the cryptography algorithms, ECC provides faster key generation, faster key agreement along with better information security through reduced key size in comparison to the other cryptography algorithms. The basic comparison between RSA and ECC according to experimental complexity during the implementation of different cryptography method is depicted in Table 1 [4, 8]:

Elliptic curve cryptosystem is an asymmetric key cryptosystem that is described in finite field. The elliptic curve algebraic structure is primarily used for various implementations. As compared to other cryptosystem ECC provides us strong security with smaller bits that refer to

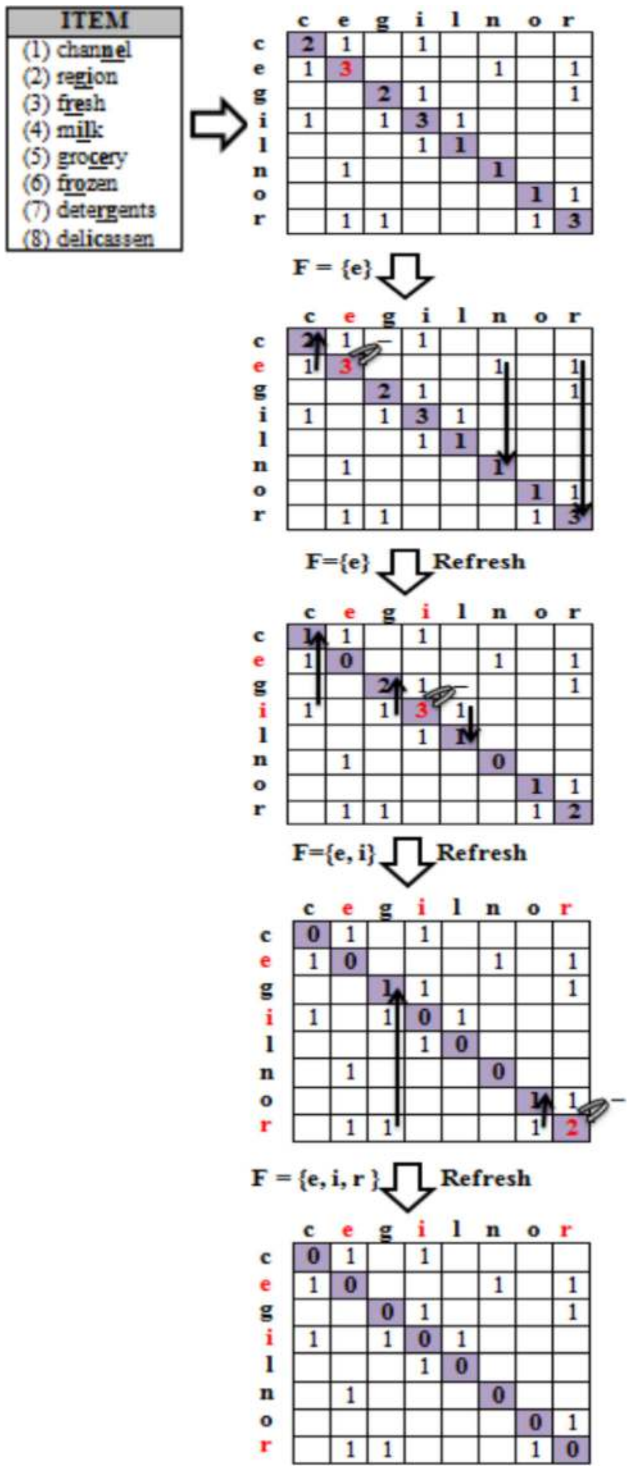

Fig. 1 Example of GFGS algorithm

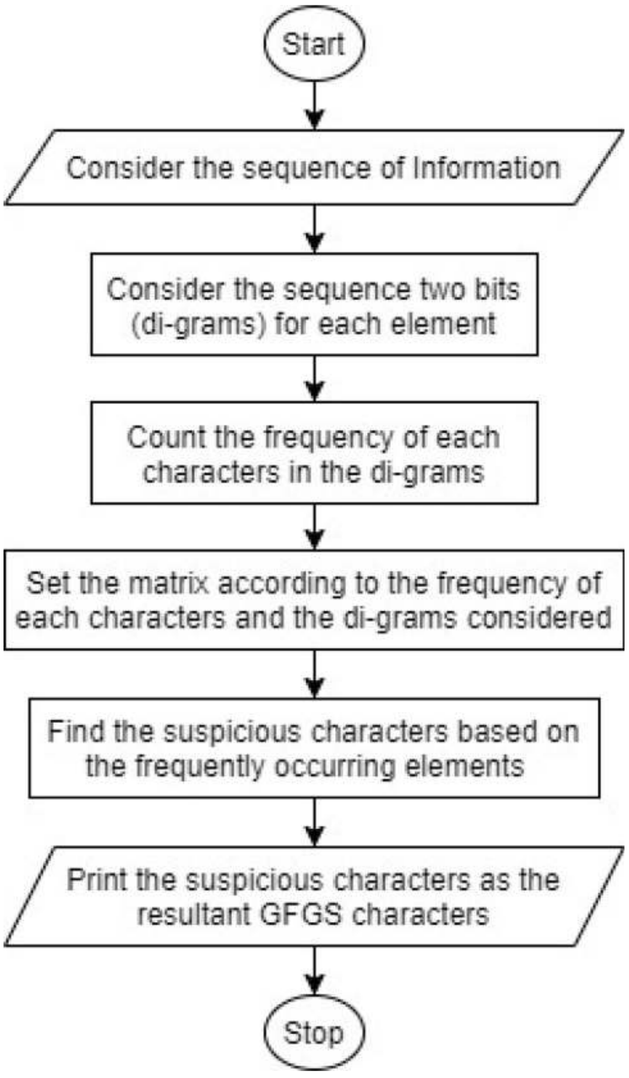
Start
Consider the sequence of Information
Consider the sequence two bits (di-grams) for each element
Count the frequency of each characters in the di-grams
Set the matrix according to the frequency of each characters and the di-grams considered
Find the suspicious characters based on the frequently occurring elements
Print the suspicious characters as the resultant GFGS characters
Stop

Fig. 2 Flowchart for GFGS character set selection

faster performance to reduce storage and transmission requirements. It is a non-trivial method i.e. it involves a combination of ECC cryptography along with ECDH key exchange and symmetric encryption algorithm for implementation. Here, ECDH shared a symmetric secret key that helps both encryption and decryption. Security provided by a public key of 256 bit is compared to the security provided by the public key of RSA having 3072 bits.

In this paper, we are encrypting messages by applying the ECC algorithm. We are also applying an m-gram method to provide minimum searching time and faster processing speed. Previously data are encrypted individually and also decrypted individually which takes comparatively more time. Here we are applying the pattern matching process. So that words having a common element will be accessed at a time which will mainly reduce our searching time. By searching the common element we can access all the words that refer to that particular common element. That also will require less external memory. For the frequently used data, the no. of collision point can be calculated from the frequency of that particular data that how many times a particular data is used.

3 GFGS

In general, each information is considered separately during data transmission and processing. Individual information selection and processing involve more searching time as well as processing time. So, instead of processing individual elements, a set of commonly occurring elements called m-grams are identified and all the operations are applied to the common grams. The frequent common grams nominated behave as the representatives for the set of items. We have considered a 2-gram method in which a set of two bits called di-bits is searched for the set of elements [3, 5]. Here, we have applied the ECC algorithm on the set of di-bits instead of individual elements so that group security can be achieved and processing time would be relatively less. The 2-gram selection methodology can be depicted using an example as shown in Fig. 1. It is represented by the wholesalers dataset of UCI repository. The flowchart for finding the GFGS character identification is depicted in Fig. 2 as shown below.

4 ECC

An elliptic curve can be defined using the finite field called a Galois field comprises of points for the equation: “$y^{2-} = x^3 + ax + b$” where “$4a^3 + 27b^2 \neq 0$” with a condition for point of infinity denoted by ’O’.

ECC is an asymmetric key cryptosystem where each user owns one public and private key pair. Let Eq(a,b) is the elliptic curve having parameter a, b, and q, for any prime number q or an integer which can be represented in terms of 2 to the power n. G be a point on the elliptic curve that satisfy for a large value of n. The steps involved in ECC algorithm [6, 11] are depicted in Fig. 3 as:

A. Key generation (at sender)

- Let, Private key of sender (A) is= nA
  Public key of sender (A) is= pA
- pA= nA * G
- Secret key of A is K= nA * pB

- Let private key of receiver (B) is= nB
  Public key of receiver (B) is= pB
- pB= nB * G
- Secret key of b is K = nB * pA

B. Signature generation (at sender side)
The message ’M’ needs to be signed by the sender for authentication. The steps that are followed for signature generation are:

1. The cryptographic hash function ‘e’ is determined by
   e=HASH(m)
2. Select a random integer k ∈ [1, n-1].
3. Compute the signature as: (r, s)
   Where, r = x1 mod n, for the point (x1, y1)=k*G
   s = (k-1)(e+nA*r), for any random number k.
4. Send the signature to the receiver.

C. Encryption algorithm using public key of receiver

Let, A sends a message M to B.

Cipher text: C(M)= { kG , M + kpB}, for any random number k.

D. Decryption algorithm using private key receiver

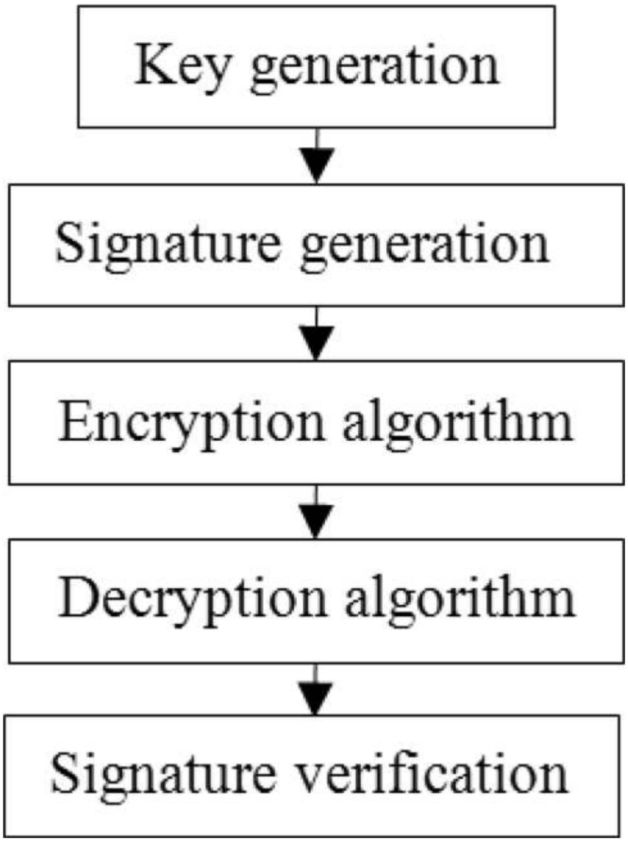

Fig. 3 Steps of ECC algorithm

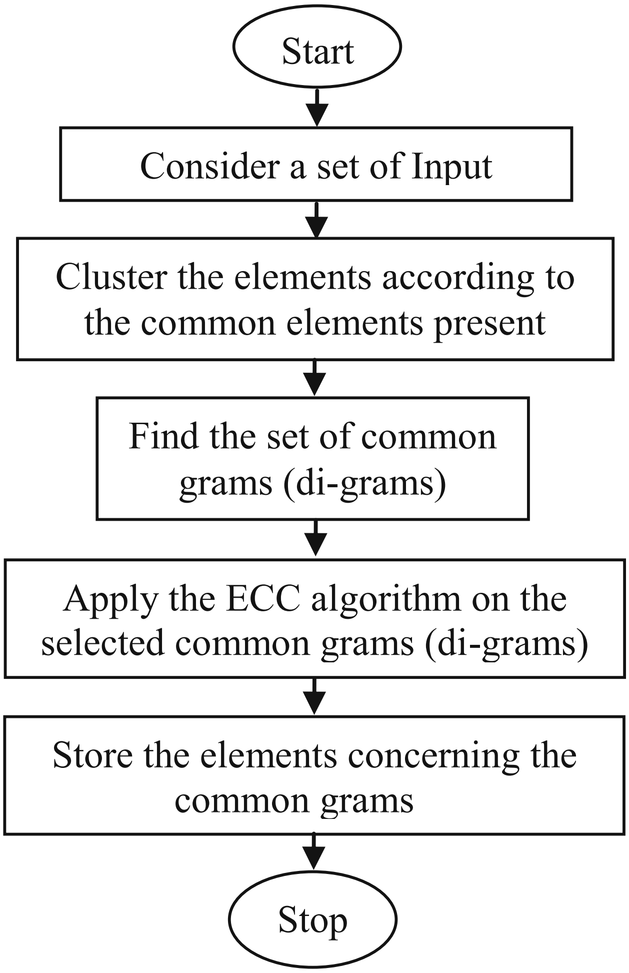
Start
Consider a set of Input
Cluster the elements according to the common elements present
Find the set of common grams (di-grams)
Apply the ECC algorithm on the selected common grams (di-grams)
Store the elements concerning the common grams
Stop

Fig. 4 Flow diagram of the proposed model

E. Signature verification (at receiver side)
For sender's signature authenticate, the receiver need to identify its public key 'PA'.

```
1. The receiver essentially ha to verify (r, s) pair for authentication within the range 1 and n-1.
2. The hash function 'e' is determined by the receiver similar to that of signature generation.
3. Determine w=(s-1) mod n
4. Determine u1=(e*w) mod n
             u2=(r*w) mod n
5. Determine (x1,y1)=u1*G + u2*PA
6. If (x1=r mod n) Then
      Set valid signature
   Else
      Set invalid signature
```

5 Proposed work

In comparison to other cryptography algorithms, ECC involves reduced key length for encryption as well as decryption. Adopting the same security features of ECC, we have tried to reduce the processing time. To reduce the processing time, we have applied ECC using generalized frequent-common gram (GFGS) method.

GFGS algorithm involves searching for suspicious elements for a set of times. These suspicious elements are called as common grams. Hence, the frequently occurring items are called as the common grams among the set of elements. We have adopted 2-gram techniques in which set of di-bits (double bits occurring sequentially) are finally obtained for the set of items. The set of di-bits undergoes ECC implementation to provide group security [7, 9].

After identifying the common grams in the first layer, the elements are concerning the common gram in the second layer. Finally, ECC is applied in the third layer of the proposed model on the common grams instead of the elements itself. The whole process is depicted in Fig. 4 (present at the last page of the paper).

The proposed model operation can be represented using the following flow diagram depicted in Fig. 5.

In brief, from earlier ECC algorithm implementation, it is observed that time taken for encryption and decryption of individual items is comparatively more as compared to our paper. By implementation of group encryption on common grams, group security could be achieved bearing the same flavor of ECC algorithm in reduced time processing time. As the security algorithm is applied to common grams, less external memory is also needed as compared to the normal ECC algorithm (Fig. 6).

6 Proposed algorithm

Let 'M' be the message that is to be sent to the receiver.

```
Initialize, Private key of the sender (A) as nA
            Private receiver (B) as nB
Step 1: Consider G as a point on elliptic curve and
         determine the magnitude of G(1,1).
Step 2: Calculate public key of A as pA=  nA * G
        and the public key of B as pB=  nB * G
Step 3: Calculate the private key of A as KA= nA * pB
        and private key of B as Kb=  nB* pA
Step 4: Calculate e=HASH(M)  for signing the message.
         Consider a randomly selected number k and
         compute the signature pair(r, s).
         Let, (X1,Y1)=k*G
              r=X1 mod n
              s=(k-1)(e+nA*r)
         Return (r, s)
Step 5: Calculate the cipher text that has to be sent to the
         receiver C(M), the length of message L.
Step 6: Initialize encryption time=0.
         For i=0 : L
             Calculate C(M)=(KG, M+K*pB)
         Return cipher text C(M) as two points on the elliptic
         curve along with the time for encryption.
Step 7: Decrypt the cipher text at the receiver's end by using
         the secret key.
Step 8: Initialize the decryption time=0
         For j=0 : No. of Rows in cipher text
              M′ = (M+K*pB)-[nB(kG)]
                  = (M+K*{nB*G})-[nB(kG)]
         Return the original message M and time taken for
         the decryption of the message.
Step 9: Verify the signature along with cipher text using the
         sender's public key as:
           i.   Determine HASH(M′ )
          ii.   Determine w=(s-1) mod n
         iii.   Determine u1=(e*w) mod n
          iv.   Determine u2= (r*w) mod n
           v.   Determine (X1,Y1)= (u1*G,u2*pA)
Step 10: Return (X1== r mod n)
          If this returns 0 then the signature is authenticated,
          otherwise, corrupted.
```

7 Experimental result

We have used MatLab-2015 tool for implementation of the proposed algorithm within Intel core Celeron(R), 1.61 GHz speed processor, 4-GB RAM and with Windows operating system (64-bit). The msnbc dataset of the UCI repository is used for the whole process implementation. We have considered 24 numbers of elements that undergo the GFGS algorithm producing 09 number of 2-grams which is shown in Table 2.

These 2-grams undergo ECC implementation to represent group security. The time needed for encryption and decryption for each element along with the identified 2-grams is depicted in Table 3 given below.

Now we can aggregate the time needed for encryption and decryption using ECC for the set of elements having a single common gram to show the comparison among the traditional approach and the proposed approach. This is depicted in Table 4 given below.

The contents of Table 4 showing the comparison between the traditional ECC and the proposed group security using ECC can be represented using the bar chart as shown in Fig. 5.

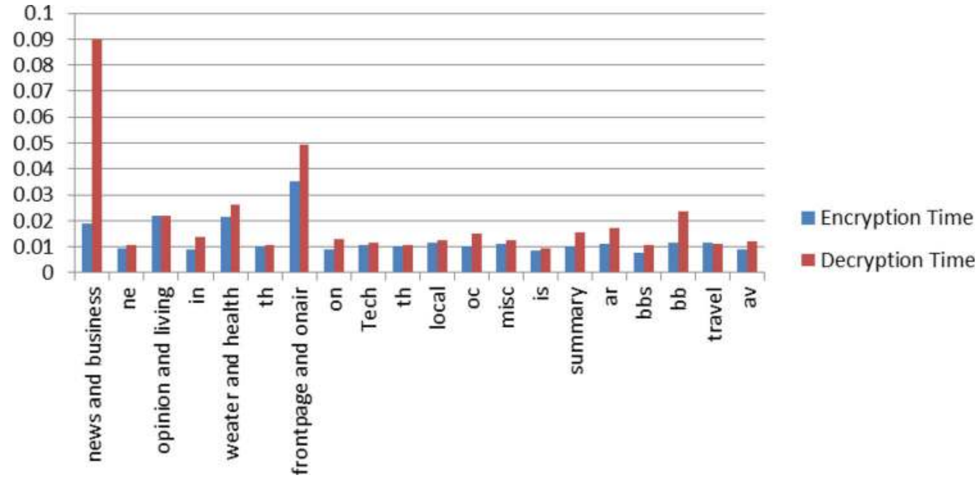

Fig. 5 Bar chart depicting the encryption and decryption time between the group elements and 2-grams

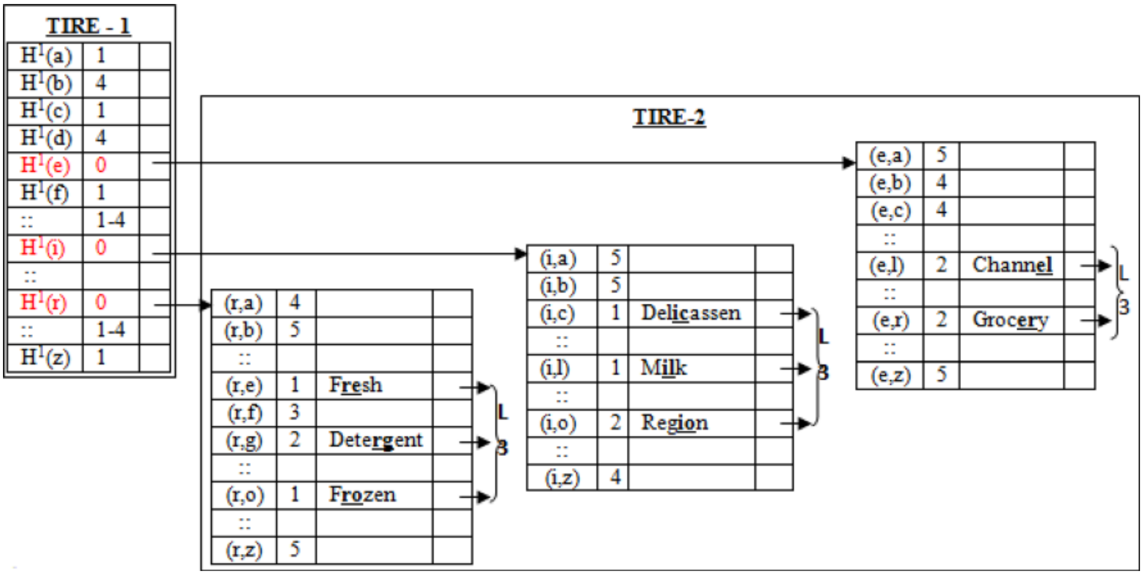

Fig. 6 The proposed model showing tire-1 for common gram selection, tire-2 for storage of elements with respect to the common grams selected and tire-2 (L3) for ECC implementation on the common grams for group security

Table 2 The identified 2-grams for the set of items

Items	2-gram
News, business	ne
Opinion, living	in
Weather, health	th
Frontpage, onair	on
Local	oc
Tech	th
Misc	is
Summary	ar
Bbs	bb
Travel	av

8 Performance of the proposed algorithm

The performance of the suggested algorithm is specified by the comparison between different time incurred for various processes. Table 5 depicts the total time needed for each element encryption, decryption, the sum of total time incurred for encryption as well as decryption, along with the total time for encryption, decryption, and the sum of encryption as well as the decryption for the generated 2-grams. The values are retrieved from Table 4.

By considering the Table 5 contents, the efficiency of the proposed approach can be specified.

- The encryption efficiency would be = 0.0212/ 0.0184 = 1.1521. So, the encryption efficiency would increase approximately by 15.21%.

Table 3 Encryption and decryption time for individual and the obtained 2-grams

Info	Enc. time (in ms)	Dec. time (in ms)	Total time (enc + dec) in ms
News	0.0095	0.0795	0.089
Business	0.0094	0.0107	0.0201
Opinion	0.0096	0.0126	0.0222
Living	0.0123	0.0138	0.0261
Weather	0.0102	0.0117	0.0219
Health	0.0113	0.0144	0.0257
Frontpage	0.0231	0.0336	0.0567
Onair	0.0121	0.0159	0.028
Tech	0.0106	0.0114	0.022
Local	0.0117	0.0124	0.0241
Misc	0.0111	0.0124	0.0235
Summary	0.0104	0.0154	0.0258
bbs	0.0077	0.0106	0.0183
Travel	0.0117	0.0112	0.0229
ne	0.0093	0.0108	0.0201
in	0.0089	0.0138	0.0227
th	0.0104	0.0108	0.0212
on	0.0088	0.013	0.0218
ec	0.0102	0.0148	0.025
ca	0.0086	0.0102	0.0188
is	0.0087	0.0095	0.0182
ar	0.0111	0.0172	0.0283
bb	0.0114	0.0236	0.035
av	0.0091	0.012	0.0211

Table 4 Comparison among the traditional approach and the proposed method

Data	Enc. Time (in ms)	Dec. time (in ms)
News and business	0.0189	0.0902
Ne	0.009	0.011
Opinion and living	0.0219	0.0219
In	0.009	0.014
Weather and health	0.0215	0.0261
Th	0.0104	0.0108
Frontpage and onair	0.0352	0.0495
On	0.009	0.013
Tech	0.0106	0.0114
Th	0.01	0.011
Local	0.0117	0.0124
Oc	0.0102	0.0148
Misc	0.0111	0.0124
Is	0.0087	0.0095
Summary	0.0104	0.0154
Ar	0.0111	0.0172
Bbs	0.0077	0.0106
Bbs	0.0114	0.0236
Travel	0.0117	0.0112
Av	0.0091	0.012

Table 5 Comparison matrix between traditional ECC and proposed group security approach

Traditional ECC approach			Proposed group security approach		
Total enc. time for each item	Total dec time for each item	Sum of enc. and dec. total time	Total enc. time for 2-grams	Total dec. time for 2-grams	Sum of enc. and dec. total time for 2-grams
0.0212	0.0907	0.1119	0.0184	0.0412	0.0596

- The decryption efficiency would be = 0.0907/0.0412 = 2.2014. So, the decryption efficiency would increase approximately by 220.14%.
- The proposed cryptography algorithm efficiency would be = 0.1119/0.0596 = 1.8775. So, the proposed method efficiency would increase approximately by 87.75%.

9 Conclusion

Based on the above implementation and review of performance the following conclusions could be made:

- The efficiency of the proposed method is completely dependent on the dataset. So, it will differ from one dataset to another.
- The efficiency of the proposed method is more in comparison to the existing methods.
- Each common gram behaves as the representative for the group elements. So, if more number of elements is stored in each group, then the time of encryption as well as decryption will reduce significantly.
- In the worst condition, i.e. if there would be no common grams present, the processing time will comparatively reduce due to the cryptography algorithm implementation on the common gram instead of individual elements.

10 Future scope

Future applications will demand a more secure and faster cryptography method. Though this group security will provide a faster and secure approach, a reduced key size algorithm containing the same flavour for implementation is highly demanded. So, we will focus on two aspects such as reduced key size and lower on-chip memory space for the implementation of security algorithms in the future. Also, we will focus on optimizing the worst situation where no common grams are present so that successful data transmission can be done by using less external memory, without any unauthorized access.

References

1. Sethi PC, Behera PK (2015) Methods of network security and improving the quality of service—a survey. Int J Adv Res Comput Sci Softw Eng 5(7):1098–1106
2. Dubai MJ, Mahesh TR, Ghosh PA (2011) Design of new security algorithm: using hybrid Cryptography architecture. In: 2011 3rd international conference on electronics computer technology, Kanyakumari, India, pp 99–101. https://doi.org/10.1109/ICECTECH.2011.5941965
3. Sheu T-F, Huang N-F, Lee H-P (2010) In-depth packet inspection using a hierarchical pattern matching algorithm. IEEE Trans Depend Secure Comput 7(2):175–188
4. William S (2006) Cryptography and network security-principle and practices, elliptic curve cryptography, 4th edn. PHI Publisher, pp 310–313
5. Sethi PC, Behera PK (2014) Secure packet inspection using hierarchical pattern matching implemented using incremental clustering algorithm, December 22–24, ICHPCA-2014 (IEEE international conference)
6. Chen TS (2004) A specifiable verifier group-oriented threshold signature scheme based on the elliptic curve cryptosystem. J Comput Stand Interface 27:33–38
7. Qureshi MA, Park J, Kim S (2020) SALE: smartly allocating low-cost many-bit ECC for mitigating read and write errors in STT-RAM caches. In: IEEE transactions on very large scale integration (VLSI) systems, vol 28, no 6, pp 1357–1370
8. Sethi PC, Behera PK (2016) RSA cryptography algorithm using linear congruence class. Int J Adv Res 4(5):1335–1347
9. Sethi PC, Behera PK (2017) Network traffic management using dynamic bandwidth on demand. Int J Comput Sci Inf Secur 15(6):369–375
10. Sethi PC, Sahu N, Behera PK (2020) Group security using ECC, ICMTCI-4.0. Springer International Conference, Dt-27/08/2020–28/08/2020 **(accepted for publication)**
11. Durairaj M, Muthuramalingam K (2018) A new authentication scheme with elliptical curve cryptography for internet of things (IoT) environments. Int J Eng Technol 7(2.26):119–124. https://doi.org/10.14419/ijet.v7i2.26.14364