Secret Sharing Based on BCH Error Correction Code

Artak, Khemchyan

National polytechnic university of Armenia (NPUA)
Yerevan, Armenia
e-mail: artak-khemchyan@mail.ru

ABSTRACT

New secret sharing scheme based on Error-Correcting Code of BCH is described in this paper. The new method has some advantages which are described as well. The comparison results of the new method with Shamir's one via testing with different parameters are also summarized.

Keywords

Secret sharing, threshold scheme, error correction codes, BCH code

1. INTRODUCTION

Recent decades have been marked by a sharp increase in the level of productivity computing systems. Such systems have been joined the extra bandwidth computer networks, which can transfer information with 100 or 1000 Mbps. The computer network users' dependency from information technology security level is also increasing along complication of information processing methods. In the era of development of modern computer technology, information security solution is mainly based on cryptographic methods of information encryption, data integrity testing, validation and secret keys generation, etc. [1]. Appearance of distributed systems for data processing and technically very complex computing systems development led to changes in the information environment and applying methods. Related to that threshold distributed algorithms become actual instead of local calculations. Distributed systems are characterized by the presence of several participants in the system. Threshold schemes are able to design a system in which the system functions are implemented in cooperation between several participants.

2. THRESHOLD SCHEMES

There are threshold secret sharing schemes, which allow the possibility of sharing confidential information so it will be possible to work with information only with certain minimum number of participants available. Data is split into several parts and stored in different places. Figure 1 shows secret sharing according to the 5/3 scheme.

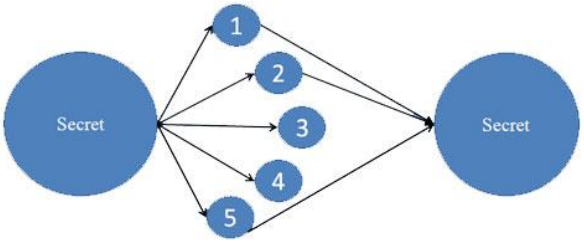

Fig. 1. The 5/3 secret sharing scheme

The more popular threshold secret sharing scheme is the Shamir secret sharing scheme, but the Shamir scheme works quite slowly, because it is based on polynomial operations and the performance is low when the threshold value is high.

Below is given the formula of Shamir for components calculation, where k- is the threshold value, M- a secret, p- is a prime number, greater than secret(M in this case). It is understood that the calculation process of components is complicated by increasing the threshold.

$$k_1 = F(1) = (a_{k-1} \times 1^{k-1} + a_{k-2} \times 1^{k-2} + \cdots + a_1 \times 1 + M) \bmod p$$
$$k_2 = F(2) = (a_{k-1} \times 2^{k-1} + a_{k-2} \times 2^{k-2} + \cdots + a_1 \times 2 + M) \bmod p$$
$$\cdots$$
$$k_i = F(i) = (a_{k-1} \times i^{k-1} + a_{k-2} \times i^{k-2} + \cdots + a_1 \times i + M) \bmod p$$
$$\cdots$$
$$k_n = F(n) = (a_{k-1} \times n^{k-1} + a_{k-2} \times n^{k-2} + \cdots + a_1 \times n + M) \bmod p$$

A similar complex operations are used in the restoration of secret as well. So, as mentioned above a new secret sharing method is proposed that is based on Error-Correcting Codes of BCH. Since the error-correcting codes mostly use logical operations which can be performed faster by a computer, this method works faster than the Shamir method.

3. THE SHARING METHOD

Below is described how the BCH(21,31,5) code example works [2]. After encoding the 21 bit initial information with BCH(21,31,5) code, the result is 31 encoded codeword allowing to correct any 2 error bits. Each 21 bit of the secret file, which should be shared, is encoding to the corresponding codeword. The codewords are considered as a two-dimensional array, which is presented in Figure 2. This is already an encoded file with BCH(21,31,5) code, and it is ready for sharing.

1	2	3	:::	31
v1,1	v1,2	v1,3	:::	v1,31
v2,1	v2,2	v2,3	:::	v2,31
v3,1	v3,2	v3,3	:::	v3,31
v4,1	v4,2	v4,3	:::	v4,31
v5,1	v5,2	v5,3	:::	v5,31
:::	:::	:::	:::	:::
vq,1	vq,2	vq,3	:::	vq,31

Fig. 2.The structure of encoded file

In this example "k" is the length of the codeword (in this case k=31) and "q" depends on the volume of the original

file. Each line of the array represents a BCH(21,31,5) code codeword, which means that any 2 bit error can be corrected in the 31 bits. Based on this characteristic, we can distribute the encoded file by columns: If we do consider each column as a separate component, it is obvious that we can recover the original file in case of the damage or the loss of any 2 components. It results in the 29/31 threshold scheme.
In order to achieve the safety and to have the volume of the components equal to the distributed file volume, we should apply grouping. From each part, 21 columns should be taken (with a certain method) to meet all the requirements. For example, in case of grouping according to Table 1, we will obtain a 4/3 threshold scheme.

Table 1
The 4/3 threshold scheme

Part 1	23	12	19	13	28	29	31	17	15	9	7	10	14	6	4	5	2	25	8	3	1
Part 2	17	22	26	10	20	24	30	4	6	9	1	16	2	18	8	7	3	25	5	28	13
Part 3	27	29	13	23	8	3	1	17	6	25	5	14	4	19	20	10	9	26	2	7	31
Part 4	2	7	5	10	30	18	16	4	12	9	6	14	15	1	17	3	22	24	26	27	8

Table 2
The 5/3 threshold scheme

Part 1	28	20	5	10	8	25	1	14	6	9	12	13	4	15	17	7	23	3	2	29	31
Part 2	21	7	17	26	19	3	24	4	6	9	13	16	5	18	8	2	1	25	10	29	30
Part 3	19	25	5	10	8	13	1	27	30	31	3	14	17	2	20	23	7	26	4	6	9
Part 4	16	15	24	28	31	3	18	4	6	27	12	7	2	1	22	23	5	9	10	30	8
Part 5	14	22	5	10	8	19	1	24	30	9	12	2	15	16	18	3	21	7	2	25	6

Table 3
The 5/4 threshold scheme

Part 1	17	23	14	18	13	3	19	26	16	9	12	8	5	15	6	2	10	1	7	4	29
Part 2	25	15	5	10	23	17	1	4	22	9	12	13	7	16	3	18	6	8	2	28	30
Part 3	15	7	21	17	8	3	27	19	6	23	13	2	16	10	18	4	5	9	24	25	1
Part 4	18	14	16	10	26	21	1	4	15	9	11	13	7	6	5	17	2	3	23	8	28
Part 5	27	29	30	10	8	3	1	4	6	9	11	13	14	15	16	17	22	24	2	7	5

Each column represents a separate part (component). Values in the tables are the numbers of 1-31 bits. Table 1 shows that after merging any 3 columns, 2 bits are missing (not enough). While using the BCH(21,31,5) decoding algorithm we can recover the 2 missing(corrupted) bits [2][3] .It turns out that the secret sharing according to Table 1 results in the 4/3 threshold scheme. The secret file is shared into 4 parts and any 3 is enough for recovering. Table 2 presents the 5/3 threshold scheme table. Table 3 presents the 5/4 threshold scheme table. The same method of secret-sharing is applicable for other error-correcting codes. Further, other codes will also be investigated, and the performance evaluation of each code will be carried out. The system will automatically determine which code is suitable (applicable) for using. This operation should be performed invisibly for the user. Continuing the future investigation, comparison of this method with Shamir's method will be carried out through testing with various parameters.

Let us suppose that a potential opponent can be informed about the table used in distribution, that means that having fewer components will give him some info about the initial confidential information. It contradicts the conditions imposed on threshold schemes, that the combination of the component, less than the threshold value should not provide any info about initial confidential information. Although it is not possible to restore the secret with the above mentioned method, to solve this contradiction and to make this method of the secret sharing more perfect, it is supposed to use substitution. The table that is selected for sharing substitution should be applied before the sharing process. This process is performed in the following steps:

1. count of 2*(m/3) (m is the length of codeword) ID numbers are generated (corresponding column numbers in the table). The set of those IDs is considered as the codeword of sharing and it is presented as $ID_1ID_2ID_3\ldots ID_n$.

2. in already selected table , the columns are shifted with each other (ID_1 with a ID_2, etc). Since the number of generated ID numbers is couple, this process is always possible.

After substitution, the opponent cannot know which bit of codeword is in which bit of which file. After all this, it is necessary that the restoration program also should know the substitution password before restoring. Of course, that password should not be stored in components in the open form. It is recommended to share the component within parties and the store distributed version in the files created. Because the volume of the password is not big, it can be shared via the Shamir method and it will not have a significant impact on performance. This process is done only once at the beginning of the sharing process. It is natural that sharing should be done with the same parameters that were used for the entire file sharing (a new method). This method also allows to keep the table number, used in sharing. The foregoing is illustrated in Figure 3, the structure of the distributed component-file header.

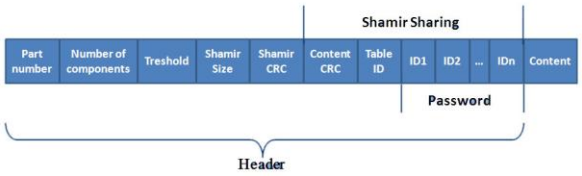

Fig. 3.The structure of component-files

- Content CRC - This field contains the CRC32 value of the distributed part (without head). Designed for detecting possible changes in this part.
- Table ID - ID of the table of the selected code, which is used in sharing. Each code has many tables for different threshold schemes.
- $ID_1ID_2ID_3\ldots ID_n$ - the generated ID numbers that make up the sharing password together.
- Header CRC - CRC32 value of the Shamir Sharing segment and. Designed to detect the possible errors.
- Header Size - the number of bytes in the Shamir Sharing segment. Designed for the program to read that part correctly.
- Part Number - the part Number - The sequential number of part.
- Number of components - the total number of components.
- Threshold - the threshold value of sharing. Designed for restoring the Shamir sharing segment.
- Content - the distributed sector of confidential information (with the ne method).

By this option of sharing, the opponent having less than a threshold number of components can know nothing about the initial secret file. He can only know the volume of the secret file (it is known by all participants).
Figure 4 compares the performance of Shamir's and our (ECC) methods (dependency of the sharing time on the threshold value). Here we can see that for high values of the threshold, Shamir's method is slower than the new method.
Figure 5 shows the graph looks of performance comparison of Shamir methods and ECC, while the number of

components is fixed. This is one of the advantages of this method.

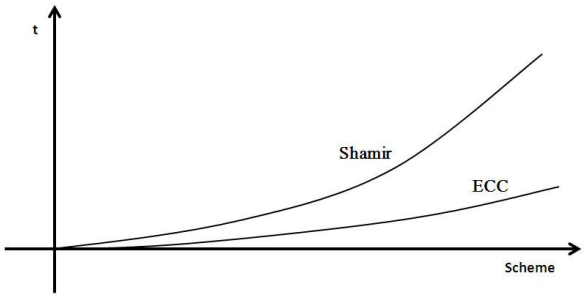

Fig. 4.Performance

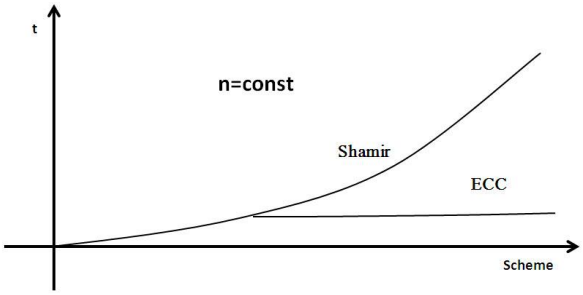

Fig. 5.Performance

4. CONCLUSION

So we have a new secret sharing system using BCH error-correcting code. It is comparatively faster as the error-correcting codes are based on the logical operations, and increasing the threshold on the other hand does not affect the system performance. Unlike Shamir method, this method is applicable not only for sharing of secret keys, but also for a distribution of large amount of confidential information. Continuing the future investigation, comparison of this method with Shamir's method will be carried out through testing various parameters. It is necessary to continue the research to prove that this distribution method is a perfect one. It is necessary to state that it is impossible to recover the secret by less parts than the threshold value, or to get any information about the secret.

REFERENCES

[1] Bruce Schneier. Applied Cryptography - John Wiley & Sons, 1996. Second Edition ISBN 0-471-12845-7 - 784p.

[2] Peterson W.W., Weldon E.J. Error-Correcting Codes - The Massachusetts Institute of Technology ,1972 Second Edition ISBN 0-2621-6039-0 – 572p.

[3] Assmus E.F., Key J.D. Designs and Their codes - Cambridge University Press, 1992. ISBN 0-5214-5839-0 - 364p.