Data Sharing Based on Error-Correcting Codes

Artak Khemchyan, Sevak Harutyunyan
National Polytechnic University of Armenia

Abstract

In parallel of cryptographic and stenographic methods of data protection, secret sharing schemes are also widely distributed. The secret sharing schemes are intended for sharing of secret keys. Secret information is being encrypted and afterwards the key of encryption is shared among participants. The above mentioned version of sharing ensures only data confidentiality. This paper describes the new method which ensures the data confidentiality, integrity and availability at once. It is based on the complete sharing of secret information. Taking into consideration the continuous growth of data amount and poor performance of existing sharing schemes, the new, faster method has been developed. It is based on Error-Correcting Codes. The resulting system is faster than the widespread scheme of Shamir.

1. Introduction

In recent years the security of operations taking place over a computer network becomes very important. It is necessary to protect such actions against "bad" users who may try to misuse the system (execute actions without authorization, read personal mail, or impersonate other users). Secret sharing protocols are used for distributed data storage. Most often such information are secret keys or passwords of users. For example, the chief accountant of the company keeps secret operational information encrypted, and a key length of 64 bits is stored in a safe place known only to him. But what happens if the chief accountant leaves or the hiding place with a key burns down? Then the key is lost, and decryption in modern cryptosystems may take millions of years. It is possible to issue a copy of the key by the deputy chief accountant and the director of the enterprise. But if the deputy wants to take the place of the chief accountant, will use its copy of the key and substitute critical information? Or sell it to competitors? It is possible to propose the following solution: split into four key parts of 16 bits, and give one part of the general director, the other - his deputy, and the third - the deputy chief accountant, and the fourth - the spouse (wife) of the chief accountant. But what if the deputies agreed displace their superiors and take advantage of its key parts? Then, in order to recover the key, an attacker will need to pick up just 32 bits, which will require $2^{32} \approx 4.3*10^9$ operations instead of $2^{64} \approx 18.5*10^{18}$ in the selection of 64 bits. Attackers can recover the key for the foreseeable future. It would be wise to divide the 64-bit key K so that each got 64-bit. How to do it? General Director and the deputies will receive a random string of 64 bits, S_1, S_2, and S3 accordingly and the chief accountant the string of bits as follows:

$$S_4 = K - S_1 - S_2 - S_3 \pmod{2^{64}}$$

Then each of them will have a random string of bits by which the key can be restored only with the brute force of 64-bit number. Even combining all three parts, it is impossible to get any information about the key, and it is impossible to reduce the number of searched bits. However, when combining all four parts, the key can be uniquely calculated.

$$S_1 + S_2 + S_3 + S_4 \equiv K \pmod{2^{64}}$$

We have just described a very simple secret sharing scheme with the one group of participants allowed, consisting of 4 subscribers.

2. Sharing protocols

Secret sharing refers to methods for distributing a secret amongst a group of participants, each of whom is allocated a share of the secret. Those protocols are designed to solve the problem of storing information so those groups of people, who are allowed to know the secret, could restore it, and those groups that are not allowed to know the secret, will not be able to recover even by brute force.
The (k, n) threshold secret sharing scheme (where $k \leq n$) called such a scheme, where the secret is shared between n participants, and the access group is a group which consists of not less than k participants.

Secret sharing threshold scheme was initially proposed independently by Shamir [1] and Blakley [2]. The scheme by Shamir relies on the standard Lagrange polynomial interpolation, whereas the scheme by Blakley is based on the geometric idea that uses the concept of intersecting hyper planes. The most common threshold scheme is a Shamir's scheme, therefore, further experimental data and comparisons will be made for this scheme.

3. Shamir's threshold scheme

The main idea on which Shamir's scheme is based, is that k of points is required for interpolation of a polynomial of a level of k−1. If the smaller quantity

of points is known, then interpolation will be impossible.

Preparatory stage: The dealer selects in a random way coefficients of S_1, S_2..., $Sk_{-1} \in Zp$ also makes a confidential polynomial

$$S(x) = S_{k-1}X^{k-1} + S_{k-2}X^{k-2} + \ldots + S_1X + M \bmod p$$

Where M – secret to be shared, and remaining coefficients – arbitrary elements of a field (the dealer keeps coefficients of polynomial in the secret). Obviously, S(0) = M.

Further the dealer selects n of different unclassified nonzero elements r_1, r_2..., r_n from Zp, puts each of which in compliance to one participant of the scheme.

Sharing of the secret: The dealer calculates values of a polynomial $c_1 = S(r_1)$, $c_2 = S(r_2)$,…, $c_n = S(r_n)$, part of each user Ai is a couple of numbers (r_i, c_i), i =1,2..., n. Afterword parts are distributed to participants of the scheme.

Recovery of Secret: To recover a secret of M, it is necessary to use the interpolation formula of Lagrange: if it is required to construct a polynomial of S (x) a level (k-1) which in case of x_1, x_2..., x_k accepts according to value y_1, y_2..., y_k, this polynomial will be:

$$S(x) = \sum_{j=0}^{k-1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

As the polynomial should be selected from the scheme of sharing of a secret so that S(0) = M, from a formula of Lagrange follows that

$$M = \sum_{i=0}^{k=1} c_i S_i \text{, где } \quad S = \prod_{i \neq j} \frac{r_j}{r_j - r_i}$$

From described above, it becomes clear that for big values of a threshold, sharing and restoring calculations becomes slowly. Figure 1 shows it.

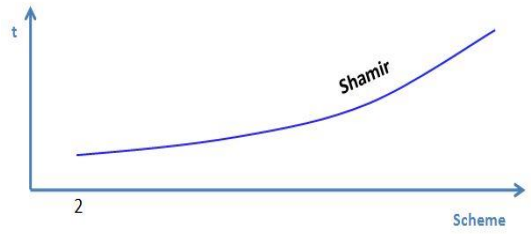

Figure 1. Performance graph of shamir's scheme. Time dependency from threshold value

4. Data Sharing

Threshold secret sharing schemes were originally designed for the distribution of secret keys. This means that data security is ensured as follows:

- encrypt secret information
- share secret key between participants according to (k, n) scheme

In this case, we ensure the privacy of information and only authorized subset of participants can recover the encryption key, and then decrypt information (Fig.2).

But this will not solve the problems of data integrity and availability. Imagine the situation when the encrypted information has been damaged. In the case of restoring the encryption key, however, they will not be able to fully decrypt the secret data. It turns out that this version of sharing does not provide data integrity.

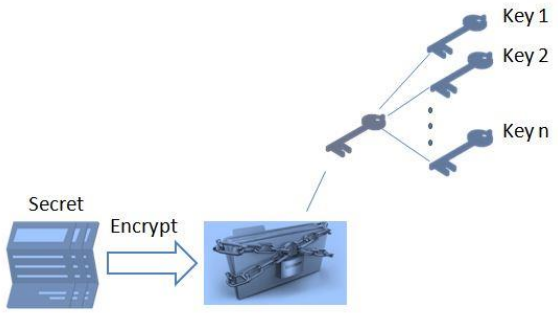

Figure 2. Secret data encryption and key distribution scheme

Now imagine that confidential information is not available some time due to different reasons (server is not available, the file was deleted, etc.). To solve these problems, it is offered to share all data by (k, n) threshold scheme with no encryption (Fig.3). It ensures data confidentiality, integrity and availability.

- Confidentiality - data can be recovered by the members of the authorized group (k or more participants in the group) and cannot be recovered by unauthorized group (less than k members in the group).
- Integrity - imagine that numbers of components have been damaged. In this case it is possible to restore damaged components of confidential data (if you have undamaged k number of component).
- Availability - imagine that one or more components are not available at the moment. In this case also it is possible to recover confidential data using the remaining components.

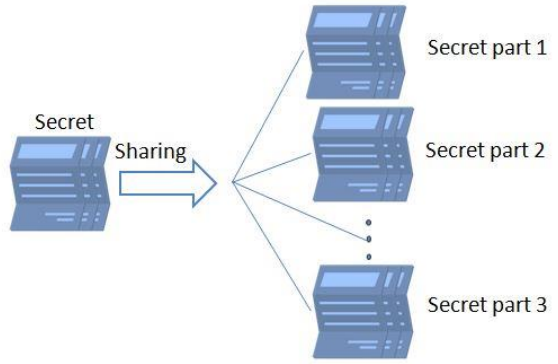

Figure 3. Secret data full sharing scheme

We can say that this method of sharing provides complete data security (confidentiality, integrity, availability). Fig. 4 presents a performance graph (time dependency from threshold value) of encryption of 10mb file with AES-256 [3] encryption (decryption) algorithm and a performance of sharing (restoring) of encryption (secret) key [1] with Shamir threshold scheme (n = 5, k = 2,3,4,5). Following configuration of PC is used to obtain the result of experiment (CPU - Intel Core i3 2.00 GHz, AM - 2GB).

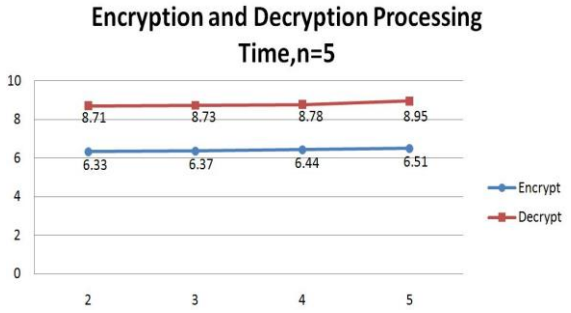

Figure 4. AES 256 encryption (decryption) algorithm and key sharing (restoring) time graph

Fig. 5 shows the time graph of sharing and restoring of the same file (10MB volume) with the Shamir's scheme (n = 5, k = 2, 3, 4, 5).

Fig. 5 clearly shows that the performance is getting slow while increasing file size either the threshold value. While Fig. 4 shows that the increase of threshold brings small impact on performance.

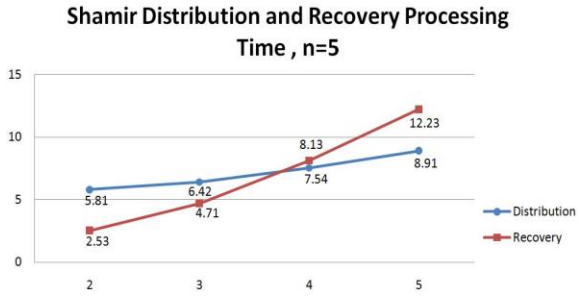

Figure 5. Time graph of Shamir scheme of sharing and recovery

The complete sharing of file is not applicable as distribution schemes are quite slow for sharing of large amount of data. This problem occurs to have a high-speed data distribution over the threshold scheme. A new threshold scheme is offered, which is based on Error-Correcting Codes.

5. Relation to error-correcting codes

Error-correcting Codes designed to find and correct errors, which are occurred during transmission or information storing (keeping). For such purpose, Information is split into k bit sized parts, which result to n-length code words after being encoded. Such code is denoted (n, k, d_{min}), where d_{min} called minimum distance of code and $d_{min}=2t+1$, where t is a maximum number of errors, which code can correct.
A relationship between Shamir's secret sharing scheme and the Reed-Solomon error correction codes was mentioned by McEliece and Sarwate in 1981 [4]. After that several authors have examined the construction of secret sharing schemes based on linear error correcting codes [5, 6, 7, 8, 9, 11, 10 and 12]. Massey used linear codes for secret sharing and pointed out a relationship between the access structure and the minimal codewords of the dual code [8, 9]. Below is described the new secret sharing method which is based on Error-Correcting Codes. It allows the usage of any Error-Correcting Code for construction of secret sharing scheme.

6. The new secret sharing method

Here is described how secret sharing schema is worked which is based on Hamming (7, 4, 1) code. However, described algorithm works for other codes as well. After encoding the 4 bit of initial information with Hamming (7, 4, 1) code, it results 7 bit of encoded codeword, allowing to correct any 1 error bit. Each 4 bits of the secret file, which should be shared, are encoded to the corresponding codeword. The codewords are represented as a two-dimensional array, which is shown in Fig.6. This is already an encoded file with Hamming (7, 4, 1) code, and it's ready for sharing.

In this example **n** is the length of the codeword (in this case n=7) and **q** depends of the volume of original file. Each line of the array represents a Hamming (7, 4, 1) codeword, which means that any 1 bit error can be corrected in the 7 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 1 component. It results the (6, 7) threshold scheme.

1	2	3	:::	7
v1,1	v1,2	v1,3	:::	v1,7
v2,1	v2,2	v2,3	:::	v2,7
v3,1	v3,2	v3,3	:::	v3,7
v4,1	v4,2	v4,3	:::	v4,7
v5,1	v5,2	v5,3	:::	v5,7
:::	:::	:::	:::	:::
vq,1	vq,2	vq,3	:::	vq,7

Figure 6. The structure of encoded file

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, 4 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 (2, 4) threshold scheme.

Table 1. The (2, 4) threshold scheme

Part 1	3	6	2	7
Part 2	1	6	7	5
Part 3	5	4	2	6
Part 4	7	5	3	4

Table 2. The (3, 4) threshold scheme

Part 1	7	6	2	7
Part 2	4	6	5	6
Part 3	5	1	5	7
Part 4	7	6	3	6

Table 3. The (2, 3) threshold scheme

Part 1	5	2	5	4
Part 2	4	7	5	3
Part 3	6	2	7	3

Each row represents a separate part (component). Values in the tables are the numbers of 1-7 bits. Table 1 shows that after merging any 2 rows, 1 bit is missing (not enough). While using the Hamming (7, 4, 1) decoding algorithm we can recover the 1 missing (corrupted) bit. It turns out that the secret sharing according to Table 1 results the (2, 4) threshold scheme. The secret file is shared into 4 parts and any 2 is enough for recovering. Table 2 presents the (3, 4) threshold scheme. Table 3 presents the (2, 3) threshold scheme. The same method of secret-sharing is applicable for other error-correcting codes.

Table 4 presents (3, 4) threshold scheme for BCH (31, 21, 2) code [13]. Table 5 for (3, 5) and Table 6 for (4, 5) threshold scheme.

Table 4. The (3, 4) 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 4. The (3, 5) 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 6. The (4, 5) 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

Now, let us suppose that a potential attacker 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:

- count of n (n=m+1 or n=m+2 (n must be even number)) (m is the length of codeword) ID numbers are generated. The set of those IDs is considered as the password of sharing and it is presented as $ID_1ID_2ID_3\ldots ID_n$.
- 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 attacker will not know which bit of codeword is in which bit of which file. After this all, it is necessary that the restoration side 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 so 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 logical that sharing should be done with the same parameters that were used for the entire file sharing (a new ECC method). This method also allows to keep the table number, used in sharing. The foregoing is illustrated in Fig. 7, the structure of the distributed component-file header.

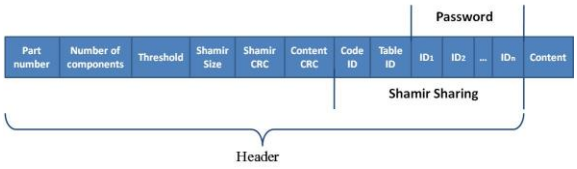

Figure 7. The structure of component-files

1. Content CRC - This field contains the CRC32 [14] value of the distributed part (without header). Designed for detecting possible changes in this part.
2. Table ID - ID of the table of the selected code, which is used in sharing. Each code has many tables for different threshold schemes.
3. Code ID - ID of selected code, which is used in sharing process.
4. $ID_1ID_2...ID_n$ - the generated ID numbers that make up the sharing password together.
5. Shamir CRC - CRC32 value of the "Shamir sector" segment and designed to detect the possible errors.
6. Shamir Size - the number of bytes in the "Shamir sector" segment. Designed for the program to read that part correctly.
7. Part Number - the sequential number of part.
8. Number of components – the total number of components.
9. Threshold - the threshold value of sharing. Designed for restoring the "Shamir sector" segment.
10. Content - the distributed sector of confidential information (with the new ECC 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 8 shows a diagram of the sharing algorithm.

Fig. 10 present a time graph of file sharing(restoring) of 10MB with the new scheme (n=5, k=2,3,4,5). Here we can see that the increase of threshold value doesn't affect the performance of sharing (restoring). As a comparison with Fig. 5, we can say that the new scheme is faster and expedient to apply for complete distribution of files.

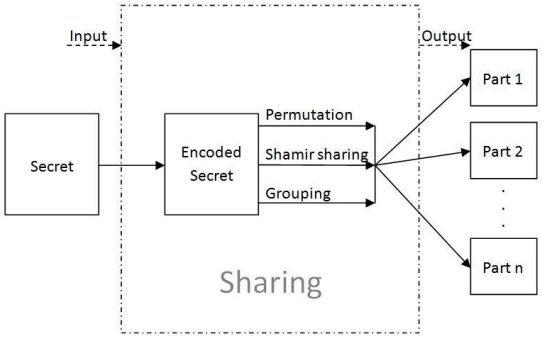

Figure 8. Secret sharing process diagram

Figure 9 shows a diagram of the restoration algorithm.

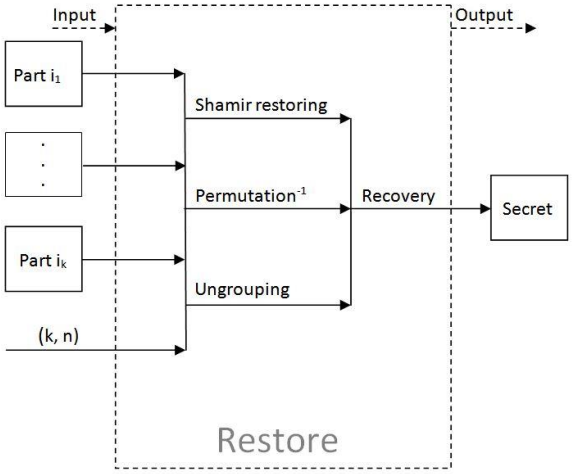

Figure 9. Secret restoring process diagram

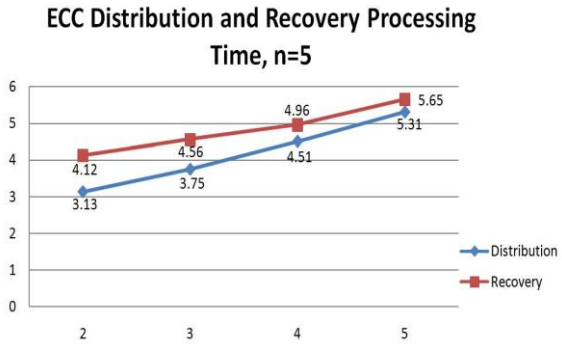

Figure 10. Time graph of the new sharing (restoring) scheme

It should be noted that the new scheme described above is ideal (the volume of components are equal to the secret) and perfect (it is not possible to know anything about secret with less then k components).

7. Conclusion

So secret sharing ensures required level of data protection. It was shown that the complete data distribution not only ensures their confidentiality, but also integrity and availability. For this purpose new

secret sharing scheme has been created, which is based on Error-Correcting Codes. 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 scheme 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.

8. References

[1] A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.

[2] Blakley, G. R., Safeguarding cryptographic keys. In: AFIPS conference proceedings, vol. 48, 313 - 317, 1979.

[3] Joan Daemen, Vincent Rijmen, "The Design of Rijndael: AES – The Advanced Encryption Standard." Springer, 2002. ISBN 3-540-42580-2

[4] McEliece, R. J., Sarwate, D. V.: On sharing secrets and Reed-Solomon codes, Comm. ACM, 24, 1981, 583–584.

[5] Anderson, R. J., Ding, C., Helleseth, T., Kløve, T.: How to build robust shared control systems, Designs, Codes and Cryptography, 15, 1998, 111– 124.

[6] Ding, C., Kohel, D., Ling, S.: Secret sharing with a class of ternary codes, Theoretical Computer Science, 246, 2000, 285–298.

[7] Karnin, E. D., Greene, J. W., Hellman, M. E.: On secret sharing systems, IEEE Trans. Information Theory, 29, 1983, 35–41.

[8] Massey, J. L.: Minimal codewords and secret sharing, Proc. 6th Joint Swedish-Russian Workshop on Information Theory, August 22-27, 1993, 276–279.

[9] Massey, J. L.: Some applications of coding theory, Cryptography, Codes and Ciphers: Cryptography and Coding IV, Formara Ltd, Esses, England, 1995, 33–47.

[10] McEliece, R. J., Sarwate, D. V.: On sharing secrets and Reed-Solomon codes, Comm. ACM, 24, 1981, 583–584.

[11] Okada, K., Kurosawa, K.: MDS secret sharing scheme secure against cheaters, IEEE Trans. Inform. Theory, 46(3), 2000, 1078–1081.

[12] Pieprzyk, J., Zhang, X. M.: Ideal Threshold Schemes from MDS Codes, Information Security and Cryptology - Proc. of ICISC 2002, LNCS 2587, Springer Verlag, Berlin, 2003, 269–279.

[13] Bose, R. C., Ray-Chaudhuri, D. K. , "On A Class of Error Correcting Binary Group Codes", Information and Control 3, 68-79 (1960).

[14] Koopman, Philip (July 2002). "32-Bit Cyclic Redundancy Codes for Internet Applications". The International Conference on Dependable Systems and Networks: 459 468.

