Personal Data Storage and Management Using Error Correction Codes and Elliptic Curves Proxy Signature Algorithm

https://doi.org/10.31713/MCIT.2019.23

Artur Joshi
National University of Water and Environmental Engineering
Institute of Automatics, Cybernetics and Computer Engineering
Rivne, Ukraine

Abstract— **Recently personal data storage and management have become one of the most important issues in the field of information technologies. 2018 was the year when well-known GDPR was issued which stated general personal data protection regulations. It's very important to study methods and tools which can enhance the security of information systems processing personal data. Distributed data storages are widely used for fault tolerance as well as cryptography is used for access control.**

Keywords—distributed applications, error correction codes, proxy signature, elliptic curves, access control.

I. Introduction

Key principles of data protection regulation include integrity and confidentiality – personal data must be protected against unauthorized processing or corruption; authenticity – corrupted personal data must be corrected or erased [1].

Information stored and transmitted in systems is often can be naturally or deliberately corrupted. The most dangerous threats include information destruction, disclosure, interception and substitution. Traditional way to protect information from disclosure is data encryption. Countering unauthorized access and data substitution is achieved by authentication which can be evaluated by imitability. The communication channel may be affected by interference from the transmitter or attacker. The protection against this attack is data encoding which can be evaluated by error correction ability rate.

The main goals of information security are confidentiality, integrity, authenticity and accessibility. Cryptographic methods are implemented by cryptographic transformations, special key data to be able to hide and restore the content.

Recently distributed applications are widely used for data storage systems [3]. The most specific feature of distributed applications is the possibility of partial break down. A partial break down occurs when an error occurs in one of the application components. This failure may disrupt some components, but other components are supposed to work as expected.

II. Data Storage Using Error Correction Codes

It is known that Reed-Solomon codes can be used to provide error correction in RAID systems. RAID (Redundant Array of Independent Disks) is a data virtualization technology that integrates multiple disks into a logical element for redundancy and performance improvements. The main problem of data storage is the following. Suppose there are n storage devices $D_1, D_2, \dots, D_n$, each containing k bytes. Such devices are commonly referred as storage devices. Also m devices $C_1, C_2, \dots, C_n$ are added to the system each containing k bytes as well. Such devices are called checksum devices. The values contained in the checksum devices must be calculated based on the values stored in data storage devices. The main problem is to specify an algorithm for calculating checksum values in the way that if any m devices from the set $D_1, D_2, \dots, D_n, C_1, C_2, \dots, C_n$ are failed, their content can be restored from the data of devices which remain working.

Error correction codes were discovered decades ago. However, the technology of sharing data across many storage devices is quite new. The first prototype of such systems was RAID technology in which small hard drives were grouped into an array providing relatively large capacity, data transfer and reliability. Since then, the technology has become used for developing network file systems that provide high reliability and information transfer speed. Such systems are known as RAID-like systems.

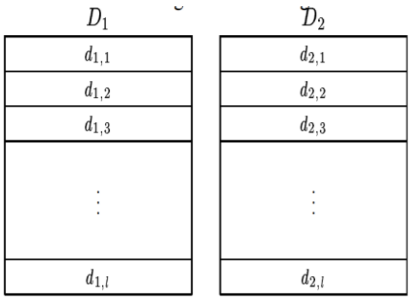

Figure 1. Data storage devices

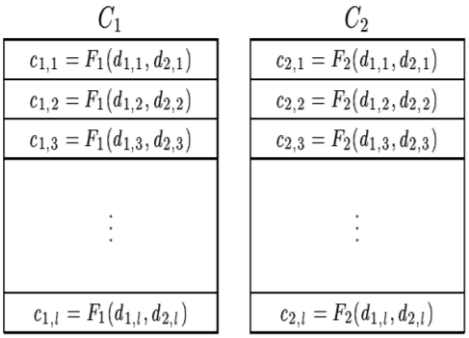

Figure 2. Checksum devices

The problem of data storage on distributed systems is one of the most important for all RAID-like systems. If data storage is provided by a set of *n* devices, that a chances that at least one fails is quite high. Hence fault tolerance becomes one of the most important requirements for RAID-like systems [4].

That's why transformations F_i defined in a following way:

$$c_i = F_i(d_1, d_2, \ldots, d_n) = \sum_{j=1}^{n} d_j f_{i,j} \quad (1)$$

where $f_{i,j} = j^{i-1}$ and then (1) looks like following:

$$\begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{m-1} & \cdots & n^{m-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \quad (2)$$

and

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \cdots & n^{m-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}. \quad (3)$$

Based on (3) it's possible to restore data from storage or checksum devices.

In distributed systems Reed-Solomon encoding in RAID systems is different from that used in RAID controllers. Such systems have two basic operations: checkpoints creation and recovery. For checkpoints creation it's assumed that the storage devices already contain some information, but checksum devices have not been used. There are two main approaches that can be used for checksum devices initialization.

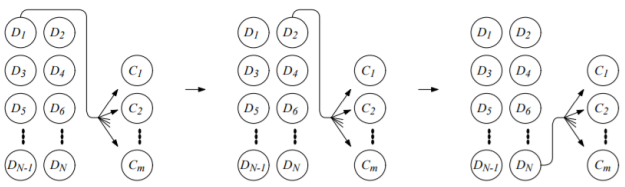

Figure 3. Translation algorithm

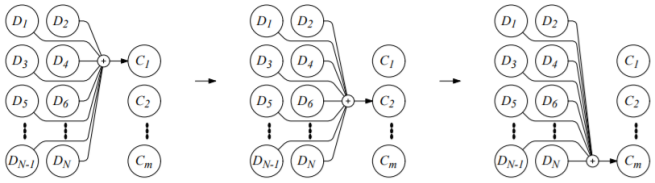

Figure 4. Aggregation algorithm

So, RAID controller systems consist of one central processor that manages the storage devices. Checkpoint system is a distributed system in which each information storage device is managed by a dedicated processor and the processors are connected via communication network [4].

III. Proxy signature access control

Computer processing and information security technologies are one of the most rapidly changing industries in recent years. In Ukraine demand for information security methods began to emerge in the second half of the 1980s. Over time, there was an urgent need of use cryptographic and technical security techniques in the private sector. At present, a great amount of confidential information is transmitted over computer systems by regular communication networks. The term of secrecy may vary from hours to many decades. Therefore, knowledge of cryptographic, technical and complex security methods and their applicability in software and hardware is extremely important to ensure the confidentiality and integrity of confidential information. Cryptographic security methods are considered to be one of the most reliable and effective to achieve this goal.

The main goals of information security are confidentiality, integrity, authenticity and its accessibility. Cryptographic information security is a type of information security which is implemented through cryptographic transformations, special key data for the purpose of hiding and restoring the content of information, authentication, authorship, preventing of unauthorized use and more [2].

Digital signature of a message is a number that depends on the message itself and some secret parameter – key. The digital signature must be easily verified and executed without access to the secret key. Digital signature allows to solve following problems: message source authentication, message integrity confirmation, makes it impossible to refuse a sign of the specific message. Accordingly, two algorithms are required to implement a digital signature scheme: an algorithm for calculating a digital signature and signature verification algorithm. The basic requirements for these algorithms are the inability to obtain a signature without the use of a secret key and the guarantee of the ability to verify the signature without knowledge of secret information.

The reliability of the digital signature scheme is determined by the complexity of the following tasks: signature forgery - location of the signature value under a given message by a person who does not own the secret key; create a signed message - find at least one message with the correct digital signature value; message substitution - select two different messages with the same digital signature value. The main difficulty in implementing a digital signature is to create a public key infrastructure. The main problem is the need for

additional open information, which is used to verify the digital signature and depends on the secret key of the author of the signature. This information is usually referred to as a public key digital signature. In order to prevent the falsification of this information by the persons who wish to act on behalf of the true owner of the signature and the secret key, an infrastructure is created, consisting of the public key certification centers and provides the possibility of timely confirmation of the identity of the given public key to the declared owner.

It is suggested to use a trusted digital signature to sign requests for information resources or, in this example, personal data. In this case, *M* is not actually a document, but a request for an information resource. Trustee *A* is the owner of the personal data and trusted party *B* is a third party or an external service that uses personal data for processing or storage. In this case, each participant of the information exchange will be able to check the permission of the owner of the personal data to use it by an external service within the limits set by *w*.

Considering the advantages of cryptosystems based on elliptic curves, the following is a protocol of trusted digital signature using the features of the group of points of the elliptic curve. The following is a protocol of trusted digital signature using elliptic curves. It uses *n* - a large prime number that is commonly known to all participants in the information exchange. Each participant in the exchange also has a pair of keys *(d, H)* - open and closed, which are related by the following relation: $H = d \cdot G$, where *G* is the well-known generator of a group of points of the elliptic curve. w describes the list of rights and access granted to the trustee. *h* is a cryptographic hash function.

Principal A's actions:

Generates a random number $k \in (1, n)$. Calculates the value of the point $P = k \cdot G = (x_P, y_P)$, calculates a proxy signature $s = d_A + k \cdot w \cdot r$. The value (P, s, w) is sent by trustee A to trustee B.

Trustee's actions B:

Checks the identification condition: $s \cdot G = H_A + P \cdot w \cdot r$. Creates a signing key: $l = d_B + s$. Relevant public key to check: $u = l \cdot G = H_B + H_A + P \cdot w \cdot r$

Signing the message (request for information resource):

M is the message, *h (M)* is its hash. $sign = l \cdot h(M) + d_B$

The quantities $(M, sign, P, w)$are passed to the verifier.

Verifier actions:

Checks the validity of a trusted digital signature of well-known values H_B, H_A, P, w:

$$sign \cdot G = h(M) \cdot [H_B + H_A + P \cdot w \cdot r] + H_B$$

Proposed trusted digital signature protocol based on elliptical curves that can be used to sign requests for information resources.

REFERENCES

[1] «Regulation of the European parliament and of the council of on the protection natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)»

[2] Diffie W., Hellman M.E. New directions in cryptography. *IEEE Transactions on Information Theory Volume 22 Issue 6*. 1976. pp. 644-654.

[3] Neuman, B. Scale in Distributed Systems. *IEEE Computer Society Press*. 1994. pp. 1-28.

[4] James S. Plank. A tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like systems. Knoxville: University of Tennessee, 2007. 19p.