Concealing Additional Secrets Using Sharing Approach in Steganography

Marek R. Ogiela, Katarzyna Koptyra
AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering
30 Mickiewicza Ave., 30-059 Krakow, Poland
mogiela@agh.edu.pl, kkoptyra@agh.edu.pl

Abstract. This paper describes a method of concealing additional secret data in fuzzy vault cryptosystem. The hidden information is placed on the second level of the system, what means that it is impossible to reveal higher level secret before decoding all related data from lower level. This property gives an opportunity of using presented technique as a secret sharing system in which information from previous step is used in the next part of the algorithm (no additional keys are required). As the existence of second-level secrets is not obvious for external observer, this idea may be applied for steganography purposes. The format of concealed secret is two-dimensional point (x, y), thus the method presented in this paper is suitable for protecting all data that can be presented as a pair of numbers.

1 Introduction

Secret sharing techniques allow to divide some data into pieces that can be joined later when participants agree to cooperate with each other. In many cases it ought to be done confidentially to prevent unauthorized people from discovering the existence of the secret information, as it could create diverse threats to users being in conspiracy. Sometimes even the mere fact of storing keys or shares may be dangerous and contributes to compromise oneself. Therefore for really important data it is worth to look for solutions based on steganography, which do not require storing any key related to shared secret.

In modern world to conceal the message from malicious third parties one can use various information systems. But the problem is that such non-standard functionalities are very rarely implemented out-of-box. On the other way, creating a new system from scratch is not always an option. So we need to either find a way of using some parts of existing system for our own purposes or modify the system to support our new, hidden function. It is difficult especially in multi-user systems, in which every modification should be transparent and not affecting to participants.

L. Barolli et al. (eds.), *Advances on Broad-Band Wireless Computing, Communication and Applications*, Lecture Notes on Data Engineering and Communications Technologies 2, DOI 10.1007/978-3-319-49106-6_50

This paper is a continuation of previous work [1] that describes a method of concealing many independent secrets in a fuzzy vault scheme (which can serve as multi-user system). Current idea is an extension of that concept giving some participants the possibility of sharing additional secrets with assumption that other functions remain intact for rest of users. Therefore it refers to multi-level steganography [2] as second secret is hidden in such a way that lower-level information is used to reconstruct higher-level data.

2 Multi-secret Fuzzy Vault

This section describes multi-secret fuzzy vault, which is a basis for the presented idea. At the beginning the underlying conception of fuzzy vault scheme is discussed.

Fuzzy vault [3] is a cryptosystem that relies on polynomial reconstruction. It uses a key in form of unordered set for locking and retrieving a secret. The entire vault is consisted of a great number of points, some of which are significant and remaining are chaff. The creation process begins with choosing a polynomial that encodes the secret (e.g. as a free term). Then this formula is evaluated on all elements of the key. As a result, genuine points are obtained. To hide the secret, we need also a number of false points which are placed more or less randomly, but with two constraints. Firstly, they cannot lie on polynomial and secondly, their x coordinates may not be members of the key. After producing both groups of points, the vault is ready and the polynomial can be erased. The recovering stage requires a key that should be identical or very similar to the key used in encoding process. With this set, the user can properly identify genuine points and reconstruct the polynomial. If the discrepancies between these keys are too big, some chaff points will be selected, what cases obtaining a wrong formula and an incorrect secret. Two important properties of fuzzy vault scheme are error tolerance (provided by error correction) and order invariance (provided by the form of the key). As a consequence, this cryptosystem is used willingly in biometrics systems with particular focus on fingerprint-based, like [4].

It turns out that fuzzy vault scheme is suitable for locking many secrets at the same time, as described in [1]. To do this, we need more keys, which are still in form of unordered sets, but all of them have to be disjunctive. Each secret information is encoded in individual polynomial, which is then used together with related key to obtain genuine points. After that chaff points are generated. This time we have two requirements: x coordinate of every false point cannot be a member of any key and also every that point may not be placed on any polynomial. There are many examples of such algorithms that can be easily modified to meet these conditions, for instance [5][6]. Thereafter, the multi-secret fuzzy vault is formed of chaff points and all groups of genuine points. Each secret may be reconstructed with use of the corresponding key or with not exactly the same, but very similar one. The recovering process is identical as in original scheme. It should be noted that important properties of fuzzy vault mentioned earlier – error tolerance and order invariance – are also preserved in multi-secret version of this cryptosystem. What is more, above solution can serve either for single user with many secrets or as multi-user system in which each participant has own secret and key.

3 Concealing Additional Shared Secrets

The construction of multi-secret fuzzy vault gives an opportunity of hiding additional, shared secrets. The idea is based on the fact that one vault can have inside many secrets encoded in various polynomials. The points in which the polynomials intersect are places where additional information can possibly be concealed. So the format of shared data is a pair (x, y). The number of secrets we can embed is dependent on degree of polynomials in the vault. To be more precise, for degree n we can hide up to n secrets (the assumption is that n is equal for all formulas). Because second level secrets are embedded in intersection points, the polynomials used in this process should not be selected totally randomly, but with specific algorithm. Such method not only has to generate formulas encoding lower level secrets, but also requires that selected points belong to both polynomials. Algorithm 1 depicts it in more detailed way.

Algorithm 1. Polynomial generation.
Input: n – degree of polynomials (a number), (x_{s1}, y_{s1}), (x_{s2}, y_{s2}), …, (x_{sn}, y_{sn}) – second level secrets (2D points), S – first level secret (a number)
Output: $(a_n, a_{n-1}, \ldots, a_1, a_0)$ – coefficients of polynomial

1. Create general formula with unknown coefficients
   $w(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + S$
2. Make a system of equations (n equations of degree n)
   $a_n x_{s1}^{\ n} + a_{n-1}x_{s1}^{\ n-1} + \ldots + a_1 x_{s1} + S = y_{s1}$
   $a_n x_{s2}^{\ n} + a_{n-1}x_{s2}^{\ n-1} + \ldots + a_1 x_{s2} + S = y_{s2}$
   …
   $a_n x_{sn}^{\ n} + a_{n-1}x_{sn}^{\ n-1} + \ldots + a_1 x_{sn} + S = y_{sn}$
3. Solve the system from point 2.
4. return $(a_n, a_{n-1}, \ldots, a_1, S)$

The explanation of Algorithm 1 is as follows.

First we create a general formula which encode first level secret. Then we have to fit its remaining coefficients to conceal shared information from second level. To do this we create the system of n equations to find missing n unknowns. After solving this system we obtain all coefficients needed to form a polynomial of degree n that embeds both secrets.

Next stages of hiding phase (genuine and chaff points generation) are the same as described in [1]. Later in this paper is presented an example showing how to conceal and reveal secrets for Alice and Bob.

Below is explained how to restore a secret shared between two users. It requires reconstruction of both polynomials first. During this operation two first level secrets are also decoded as they are stored in one of the coefficients. The whole process is shown in Algorithm 2.

Algorithm 2. Recovering 2nd level secret.
Input: w_A, w_B – polynomials
Output: (x_{s1}, y_{s1}), (x_{s2}, y_{s2}), …, (x_{sn}, y_{sn}) – shared secrets (2D points)

1. Create an equation
   $w_A(x) = w_B(x)$
   $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = b_n x^n + b_{n-1} x^{n-1} + \ldots + b_1 x + b_0$
2. Solve the equation from point 1.
3. return $(x_{s1}, y_{s1}), (x_{s2}, y_{s2}), \ldots, (x_{sn}, y_{sn})$

It should be underlined that reconstruction of shared secret requires cooperation between participants. With only one polynomial it is impossible to compute points of intersection. The users have to decode their first level secrets first, as they take part in higher level information reconstruction. It means that revealing process does not need any additional key as it uses only data from lower level. This is an important aspect in security and will be discussed wider in Conclusions section.

4 Example

This section depicts an easy example which shows how presented idea can works in practice. Suppose that we use multi-secret fuzzy vault cryptosystem with all polynomials of degree 2. It means that it is possible to hide two additional shared secrets in form of 2D points. So we select them as follows: (1, 8) and (-1, 4). These will be shared between Alice and Bob on 2nd level. The participants have also their 1st level secrets, which are: 5 for Alice and 10 for Bob.

Hiding phase. This stage starts from choosing the polynomials for users. Because degree is equal to two, the formula is:

$$w(x) = ax^2 + bx + c \tag{1}$$

Now it is time to find *a*, *b* and *c* from formula (1) in such a way that the polynomial encodes 1st level secret and also can be considered as a share (used for recovering 2nd level secrets) at the same time. For Alice we can write:

$$\begin{aligned} a + b + 5 &= 8 \\ a - b + 5 &= 4 \end{aligned} \tag{2}$$

The solution is a = 1 and b = 2. Thus we receive Alice's polynomial

$$w_A(x) = x^2 + 2x + 5 \tag{3}$$

For Bob we have:

$$\begin{aligned} a + b + 10 &= 8 \\ a - b + 10 &= 4 \end{aligned} \tag{4}$$

The solution is a = -4 and b = 2. Therefore Bob's polynomial is

$$w_B(x) = -4x^2 + 2x + 10 \tag{5}$$

Next steps (like evaluating the formula on key elements) are identical as in [1] and are not presented here.

Recovering phase. The strategy for decoding 1st level secret is identical as described in [3] and for this reason is omitted in our example. Below we present how to reveal information from 2nd level. If Alice and Bob want to recover their shared secret, they should reconstruct their polynomials first. Then they have to cooperate and find all points of interception, as shown below in (6).

$$w_A(x) = w_B(x)$$
$$x^2 + 2x + 5 = -4x^2 + 2x + 10 \quad (6)$$

The solution is x = 1 or x = -1. To find y values, we have to evaluate any of the polynomials (w_A or w_B) on computed x values. So y = 8 or y = 4. Finally our points are (1, 8) and (-1, 4).

In order to demonstrate the situation visually, we provide a graph containing all elements from above example (Fig. 1).

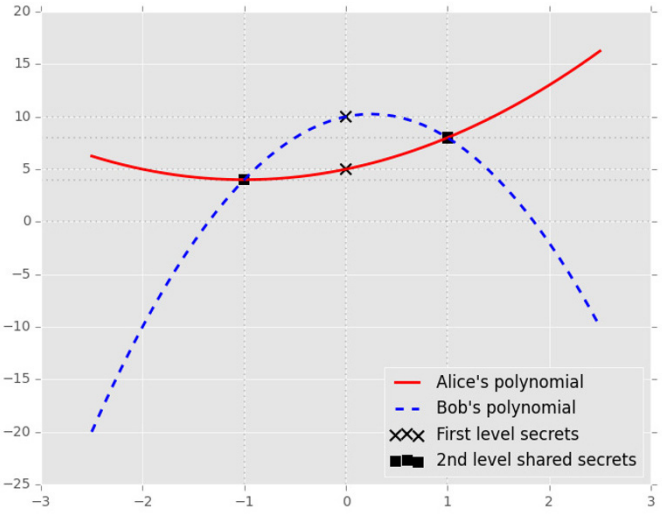

Fig. 1. Generated polynomials with marked 1st and 2nd level secrets.

5 Conclusions

This paper extends the concept of multi-secret fuzzy vault by adding the possibility of concealing additional shared secrets. To recover this data, the participants have to cooperate and reconstruct their polynomials to find points of intersections. In extended method the polynomial generation algorithm is different – the coefficients

are not random, but are selected on the basis of both secrets. Due to this fact all information needed for recovering shared secrets are obtained from reconstructed polynomials. If we consider security, it is an important feature, because no additional keys are required and the users do not have to store suspicious data which may compromise them. In fact, only the key for 1st level secret is necessary, what can be considered as normal situation in multi-secret fuzzy vault cryptosystem.

It should be noted that in presented technique it is impossible to reveal shared information with only one formula. This fact has two consequences. If a participant is not able to reconstruct the polynomial, there is no way to recover the secret. However, it is also true in case of leakage. When an enemy intercepts one formula, he will not be able to reveal shared secret without the second polynomial (guessing identity of the other conspirator and stealing the key). Of course, in general, the existence of second level secret is unknown for unintended third parties and with honest users there is no trace of shared data. First level secrets may serve to deceive an adversary as they are not important for participants (they are visible during joining shares).

Finally, the shared secret is in form of 2D points. It gives an opportunity of hiding specific types of data which are presented as a pair of numbers, like day and month, hour and minutes or geographic coordinates (latitude and longitude). For every pair is known that these two values are related, what cannot be done in systems in which a set of one-dimensional numbers is stored. Therefore the secret may contain, for example, a place where something is hidden for both users and time when they can start.

To sum up, presented construction is able to conceal shared information for two participants in situations in which high level of conspiracy is required.

Acknowledgments. This work was supported by the AGH University of Science and Technology research Grant No 15.11.120.868.

References

1. Koptyra, K., Ogiela, M.R.: Fuzzy vault schemes in multi-secret digital steganography. In: 10th International Conference on Broadband and Wireless Computing, Communication and Applications, BWCCA 2015, Krakow, Poland, November 4-6, 2015. (2015) 183–186
2. Ogiela, M.R., Koptyra, K.: False and multi-secret steganography in digital images. Soft Comput. 19(11) (2015) 3331–3339
3. Juels, A., Sudan, M.: A fuzzy vault scheme. Des. Codes Cryptography 38(2) (2006) 237–257
4. Nandakumar, A.K.J.K., Pankanti, S.: Fingerprint-based fuzzy vault: Implementation and performance. In: IEEE Transactions on Information Forensics and Security. Volume 2. (December 2007) 744–757
5. Hani, M.K., Marsono, M.N., Bakhteri, R.: Biometric encryption based on a fuzzy vault scheme with a fast chaff generation algorithm. Future Generation Comp. Syst. (2013) 800–810
6. Nguyen, T.H., Wang, Y., Nguyen, T.N., Li, R.: A fingerprint fuzzy vault scheme using a fast chaff point generation algorithm. In: Signal Processing, Communication and Computing (ICSPCC), 2013 IEEE International Conference on, IEEE (2013) 1–6