Secret Sharing Schemes Threshold Determination

Armindo Guerra Jr.

Department of Informatics and Statistics
Federal University of Santa Catarina
Florianopolis, Brazil
Email: armindogjr@gmail.com

Ricardo Felipe Custodio

Department of Informatics and Statistics
Federal University of Santa Catarina
Florianopolis, Brazil
Email: custodio@inf.ufsc.br

***Abstract*—In threshold secret sharing schemes, the threshold and the total number of shareholders are public information. We believe that such information should be secret and the threshold value must be determined before the secret reconstruction. Thus, this paper propose a method to do it. We propose also, an analysis of tailored access structure that has different subsets to rebuild the secret, each subset with his threshold. Furthermore, in our investigation we have seen that is possible that there are malicious access structures where a previleged subgroup of shareholders, in smaller number than the threshold, can reconstruct the secret. Finally, we propose a method to choose a polynomial that does not generate malicious access structures.**

Keywords–Secret sharing; threshold; information

I. INTRODUCTION

Secret sharing schemes have been used to protect sensitive data through its division into pieces and subsequent distribution to various different custodians, with the possibility that a portion of these shares can be used to recover the original data. The first known constructions that allow to share a secret were proposed independently by Shamir [1] and Blakley [2]. Shamir's scheme uses polynomial interpolation to recover the secret, while Blakley's scheme is based on the geometric intersection of hyperplanes. Many other schemes have been proposed since then and we can highlight the schemes of [3] [4], which propose general constructions and Mignotte's scheme [5], which is based on the Chinese remainder theorem. A good summary of secret sharing schemes can be found in Beimel's survey [6].

According to [7], secret sharing scheme allows a Dealer (D) to protect a secret S among a set of n shareholders. The access structure of the scheme is the set of subsets of the shareholders that are able to reconstruct the secret using their shares. A special case called (t, n)-threshold access structure consists of subsets containing at least t shareholders, where $1 < t \leq n$. In this case, any t or more shares out of n shareholders can recover the secret.

In (t, n)-threshold secret sharing schemes, t and n are public information, and, furthermore, a possible reconstructor (R) needs to know how many t shares are needed to reconstruct the secret. In this scenario, an attacker knows how many shareholders need to be coerced in order to reconstruct the secret by himself. This paper proposes that the value of t do not must be published. Therefore, our main contribution is to show that it is possible to reconstruct the secret without first knowing the threshold.

A secret sharing scheme begins with a secret and derives from it certain number of shares. The secret must be reconstruct only by certain predetermined subsets of shareholders. Let $\mathcal{V}$ be the set of n shareholders and $\mathcal{B} = 2^{\mathcal{V}}$ be the power set of $\mathcal{V}$. The set $\mathcal{B}$ is partitioned into 2 sets, $\mathcal{B} = \overline{\Gamma} \cup \Gamma$, where $\overline{\Gamma}$ is the complement of Γ. We call Γ the access structure; it contais all the subsets of $\mathcal{B}$ with cardinality greater than or equal to t. Thus, any element $x \in \Gamma$ can be used to rebuild the secret and any $x \in \overline{\Gamma}$ cannot do it. The D is the entity responsible for spliting the secret into n shares and for delivering each share to the shareholders. The R is the entity responsible for collecting shares and performs the reconstruction of the secret.

Due to our method, to reconstruct the secret without first knowing the threshold, we detected that is possible more than one access structure to reconstruct the same secret. Different subsets of shareholders will have different thresholds. Furthermore, we realize that is possible that there are malicious access structures where a previleged subgroup of shareholders can reconstruct the secret. thus, for an access structure be reliable it is necessary avoid such previleged subgroups of shareholders. Hence, we also propose a method to do it.

This document is organized as a follows. In Section II, we present a method to determine the threshold before recontructing the secret and its implications. In Section IV, we present some considerations about the propose methods. Finally in Section V, we present our conclusions and future works.

II. RECONSTRUCTING THE SECRET WITHOUT t AND n

In threshold secret sharing schemes, we are interested in building an (t, n)-access structure, where t is the threshold and n is the total number of shareholders. With at least t shares, we can determine the secret S. Normally, the secret S is the independent term of the interpolating polynomial and both t and n are public values.

In this work, however, we argue that these values should be secret. In fact, R do not need to know t nor n in order to determine the secret S. Thus, R needs a way of knowing he has used the minimum number of shares necessary to obtain the correct interpolating polynomial and so the secret S.

A. Threshold determination method

We suppose that D divides a secret S into n shares using a polynomial whose degree is $t-1$. Now, let's suppose that R wants to obtain the secret S from the w shares. As R does not kown t, then in principle it cannot reconstruct S. Nevertheless,

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5

we supose also that R perform an interpolation of w shares and d is the degree of interpoling polynomial. It is easy to check that if $d < w - 1$, then $t = d$. Thus, R has more shares than necessary to reconstruct S. However, if $d \geq (w - 1)$ R cannot determine t. Therefore, R just get many shares until the degree of the polynomial is $d = w - 1$.

Theorem 1: Let t and n be the threshold secret sharing scheme parameters, where t is the threshold and n is the number of total shares. Let w be any number of points able to participate of the reconstruction of the secret. It is possible to determine t if $w > t$.

Proof: To determine the threshold we interpolate the w shares and verify if the degree d of the interpolating polynomial is less than $w - 1$. If so, $t = d + 1$. If not, the w shares are not enough to recover the secret S. In this case, we must increment w and repeat the reasoning until the degree of the interpolator polynomial is equal to $w - 2$. This only happens when $w > t$. ■

As R does not know t nor n, two approachs are possible. The first R has no cost to picking up shares arbitrarily from Γ. In this case it is possible to offer to R all n shares. then, R interpolates n shares and easily finds the value of t. This can be done according Algorithm 1. The second R has significative costs to picking up shares. Thus, it is interesting to search for the value of t starting with the minimum possible number of shares. This search can be done according Algorithm 2.

We know that the threshold t certainly assumes a value between 2 and n. Therefore the search begin with 2 shares, because are needed at least two shares to interpolate a polynomial of degree equal to 1. Nevertheless, the methodology proposed by this paper always use one more share than traditionally required. Thus, the search starts with 3, as can be seen in Algorithm 2. It is important to observe that the proposed method should not be used in unanimous secret sharing schemes, when $t = n$, because a redundant share is always necessary. We consider redundant share one more share than the required in tradicional threshold secret sharing schemes.

Algorithm 1: checkThreshold( )

```
Input: shareList (share = (x_i, P(x_i))), 1 < i < w
Output: threshold t or −1
1  shareList = [ ];
2  w ← shareList.length();
3  polynomial ← interpolate(shareList) ;
   // interpolate() is a method which
   returns a interpoling polynomial
4  degree ← getDegree(polynomial) ;
   // getDegree() is a method which
   returns a degree of polynomial
5  if degree ≤ w − 2 then
6  |  t ← degree + 1
7  else
8  |  t ← −1
9  return t
10
```

In the first strategy, when there is no cost to R shares picks up shares, only the Algorithm 1 is necessary.

Algorithm 2: Search threshold

```
Input: AllSharesList
Output: threshold t or −1
1  t ← −1;
2  AllSharesList =[ ];
3  for i ← 0 until 2 do
4  |  shareList.append(AllSharesList.extractShare( )) ;
   |  // extractShare() is a method which
   |  picks up and delete individually
   |  shares from AllSharesList[ ]
5  repeat
6  |  t ← checkThreshold (shareList);
7  |  if t ≠ −1 then
8  |  |  return t;
9  |  else
10 |  |  shareList.append(AllSharesList.extractShare( )) ;
   |  |  // One more share
11 until AllSharesList.length() = 0 ;
12 return t
```

B. False positives

The method presented in Section II-A, shows how to retrieve the secret S without knowing the threshold t. However, care must be taken with some consequences of using redundant to do it. As Algorithm 2 increases the number of shares picked up from the access structure in an ascending order, polynomials formed by the elements of the same access structure and whose degree is smaller than $t - 1$ could be found. In this case the present method would return a false value of t. We call those false positives polynomial.

Definition: False positives polynomial are polynomial formed by the shares of the access structure whose degree is smaller than that of the polynomial interpolating.

As example, if we offer for Algorithm 1 the shares $\mathcal{A} = \{p1, \ldots, p8\}$, where $\mathcal{A} \subseteq \Gamma$, the result is polynomial $f(x) = 103/384x^6 - 3605/384x^5 + 24205/192x^4 - 154439/192x^3 + 308949/128x^2 - 361631/128x + 99$, each degree is 6. As we can see in Figure 1. However, when have significative costs to picking up shares the Algorithm 2 is used. In this case a polynomial $g(x)$ with degree is 4 will be found. Also we can see in Figure 1. In this case $g(x)$ is a false positive polynomial. It is a problem because a different polynomial can be to result in a diffente secret.

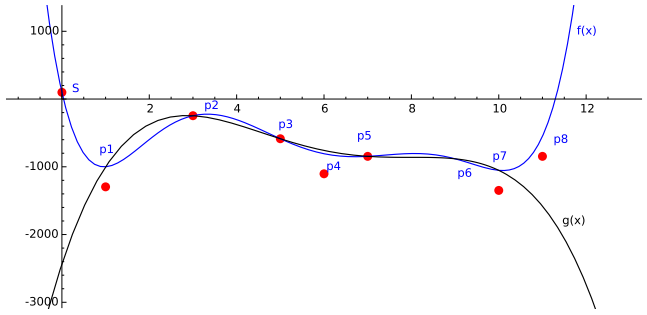

Figure 1. False positive

Knowing the possibility of false positives existence to use the method proposed by this work is necessary to generate a tailored access structure with no false positives. As the method to find false positives polynomial we propose the Algorithm 3. The objetive of this algorithm is to recieve the list of the shares from the access structure as the algorithm entry and as a result to give back all possible false positives polynomials.

Algorithm 3: Search of curves

Input: shareList
Output: A list possibleCurves, which return all possible polynomial formed using a redundant share

```
1 possibleCurves = [ ];
2 listLength ← shareList.length();
3 for j ← 3 until listLength do
4     combinationsList ← combinations(shareList,j) ;
      // combinations() is a method which
      returns the combinations of
      shareList's shares taken j by j
5     for i ← 0 until combinationsList.cardinality() do
          // combinationsList.cardinality()
             is a method whith returns the
             combinations of share of
             shareList taken j by j
6         q ← interpolate(combinationsList[i]);
7         if q.getDegree() = j − 2 then
8             possibleCurves.append(q);
9 return possibleCurves;
```

C. Generation of access structure

In the literature there are some ways to generate an access structure for a threshold secret sharing scheme. Iftene [8], for example, choose a interpolating polynomial with his coefficients over $\mathbb{F}[x]$ and choose the shares $s_1, \ldots, s_n$ as $s_i = P(x_i)$, for all $1 \leq i \leq n$, where $x_1, \ldots, x_n$ are pairwise distinct public values. In this study we consider that the access structure is formed by all ordered pairs values (x_i, s_i).

However, it is necessary to generate an access structure whose shares do not yeld false positive polynomials. A pratical way is to choose a random polynomial and test whether the access structure generated by it has or not false positives. If it has, we choose another random polynomial and perform the test again until we find a polynomial that has no false positives.

After choosing the interpolating polynomial, one way to test it if there are false positives is to give the shares generated by the interesting polynomial as the entry to the Algorithm 3. Basically the Algorithm 3 tests all combinations shares from the access structure and shows only the polynomial formed obeying the requirement of redundant share.

III. Consequences

A. Hidden access structures (HAS)

In our investigations about the tradition threshold secret sharing scheme, we discovered that there may be malicious access structures, where a subgroup of privileged shareholders in smaller number than the threshold can reconstruct the secret. We call this a hidden access structures (HAS). Let us supose that the Dealer choose a polynomial of degree 6 and generate from it an access structure. As noted in the example of Figure 2, there is a sub-access structure whose shares can interpolate the polynomial of degree 3 and whose independent term is the same than. In this case with less shares correctly assembled it is possible recontruct the secret.

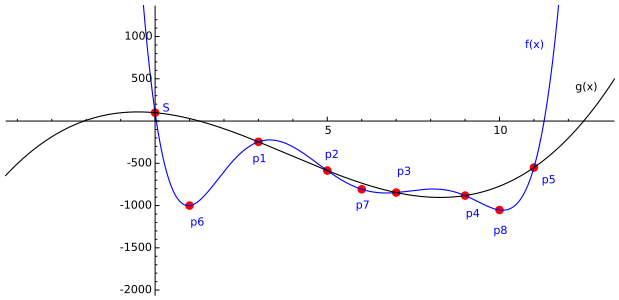

Figure 2. HAS

HAS can be created involuntarily in the moment of generation of access structure or the Dealer may be malicious and do it to take advantage in the future. In both cases we need to avoid a HAS, because it contradicts the fact that sets with cardinality smaller than t are not able to obtain information about the secret. We suggest to test the chosen polynomial and access structure generated by it.

Definition Hidden access structures (HAS) are subsets of the access structure with cardinality lower than t and that with it is possible to reconstruct the secret with it.

B. Hierarchy

It is possible to get a hierarchical scheme using tuples of shares [1]. For example, if we give the company president three shares, each vice-president two shares, and each executive one share, then a $(3, n)$-threshold scheme enables checks to be signed either by any three executives, or by any two executives one of whom is a vice-president, or by the president alone. We call this a weighted threshold secret sharing scheme.

As we have seen, hidden access structures must be in general avoided. On the other hand, in some situations to have a sub-access structure such that can be useful. As we can see in Section III-A, one of the consequences that to work with the redundant share is the possibility of having more than one curve with the same independent term. This means that shares from the curve of degree less can be obtain more important status in the secret sharing scheme. We can to compare this situation with military hierarchy. For example, the shares from the curve of degree less can be generals and another shares are soldiers. Important to say that the generals can reconstruct the secret only if work together.

As example Figure 2 shows two polynomials, one of degree 6 and another of degree 3, both with the same independent term S. The shares p_1, p_2, p_3, p_4, p_5 are generals, so only with those shares it is already possible to reconstruct the secret. Thus, such shares should be considered hierarchically more important than the other shares.

Although, the Shamir's weighted threshold secret sharing scheme is more dynamic and flexible, our scheme gives this extra possibility while being simple already to apply.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5

C. Multi-secret

Although the use of a redundant share to determine the threshold implies the possibility of polynomial behave as false positive, since under the Dealer control, we can use this fact in our favor. In some situations more than one secret it is necessary.

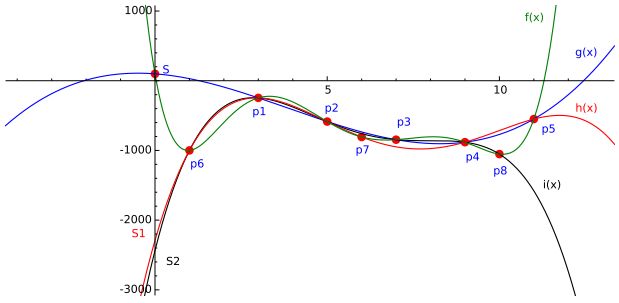

Figure 3. All curves

As we can see in Figure 3, we have 4 different possible polynomials, 2 with the same independent term and 2 with differents independent terms.

IV. Evaluation

According to theorem 5.1 of [9], computing an interpolating polynomial at shares can be performed with $O(n^2)$ operations in $\mathbb{F}[x]$. When using the Lagrange interpolation, for example, it is necessary to compute precisely $7n^2 - 7n$ operations. The Algorithm 1 to perform interpolation polynomial. In general, when only one round of interpolation is necessary, Lagrange interpolation is used.

The Algorithm 2 can use more than once Algorithm II-A, because the number of shares taken from the access structure is increased. It is possible to perform the interpolation polynomial in the Newton form and use the method of divided differences to construct the coefficients. One example is Neville's algorithm. The cost also is $O(n^2)$ operations. However, you only need to do $O(n)$ extra work if an extra share is added to the data set, while for the other methods, like Lagrange for example, you have to redo the whole computation.

It is easy to show that cost of the Algorithm 3 is $O(n^2 \times n!)$. The cost of the Algorithm 3 is significant hight to several shares. Thus, the next step of this work is to find more efficient algorithm. In spite of the cost be hight, using it we avoid the attacker to learn how much shares is necessary to bribe to break the system.

Although the proposed method is different from the traditional method, the security of the scheme is still based on the polynomial interpolation. In other words, with less than t shares, nobody can infer anything about the secret.

V. Conclusions and Future Works

One of the most significant findings to emerge from this study, is a method to determine the threshold value. In this way, the values of t and n do not need to be public information. This study has shown also a brief analysis of the possibility of construction an access structure that has different subsets to rebuild the secret. In other words, different subsets of shareholders will have different thresholds to rebuild the secret. Furthermore, in our investigations we have seen that is possible that there malicious access structures (HAS) where a previleged subgroup of shareholders in smaller numbers than the threshold can reconstruct the secret. In reliable secret sharing schemes HAS must be avoided. Thereby, our last propose is a method to do it.

Some intersting possible future works are presented next:

- In Section II we present a method to generate an access structure whose shares do not form false positive polynomials We belive thar is a more efficient method can be propose;
- To calculate the possibility of the false positive polynomials occurrence.

References

[1] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, Nov. 1979, pp. 612–613. [Online]. Available: http://doi.acm.org/10.1145/359168.359176

[2] G. R. Blakley, "Safeguarding cryptographic keys," Managing Requirements Knowledge, International Workshop on, vol. 0, 1979, p. 313.

[3] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol. 72, no. 9, 1989, pp. 56–64.

[4] J. Benaloh and J. Leichter, "Generalized secret sharing and monotone functions," in Advances in Cryptology CRYPTO 88. Springer, 1990, pp. 27–35.

[5] M. Mignotte, "How to share a secret," in Cryptography. Springer, 1983, pp. 371–375.

[6] A. Beimel, "Secret-sharing schemes: a survey," in Proceedings of the Third international conference on Coding and cryptology, ser. IWCC'11, 2011, pp. 11–46. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017916.2017918

[7] K. Wang, X. Zou, and Y. Sui, "A multiple secret sharing scheme based on matrix projection," in Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International, vol. 1. IEEE, 2009, pp. 400–405.

[8] S. Iftene, "Secret Sharing Schemes with Applications in Security Protocols," Ph.D. dissertation, University of Iasi, Romenia, 2006.

[9] J. Von Zur Gathen and J. Gerhard, Modern computer algebra. Cambridge university press, 2013.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-427-5