A Generalized Ideal Secret Sharing Scheme

Tanujay Saha

Dept. of Department of Electronics and Electrical Communication Engineering,
Indian Institute of Technology,
Kharagpur, West Bengal, INDIA-721302
tanujay.saha@gmail.com

Abstract. Sharing a secret efficiently amongst a group of participants is not easy since there is always an adversary / eavesdropper trying to retrieve the secret. In secret sharing schemes, every participant is given a unique share. When the desired group of participants come together and provide their shares, the secret is obtained. For other combinations of shares, a garbage value is returned. A threshold secret sharing scheme was proposed by Shamir and Blakeley independently. In this (n,t) threshold secret sharing scheme, the secret can be obtained when at least t out of n participants contribute their shares. This paper proposes a novel algorithm to reveal the secret only to the subsets of participants belonging to the access structure. This scheme implements totally generalized ideal secret sharing. Unlike threshold secret sharing schemes, this scheme reveals the secret only to the authorized sets of participants, not any arbitrary set of users with cardinality more than or equal to t. Since any access structure can be realized with this scheme, this scheme can be exploited to implement various access priorities and access control mechanisms. A major advantage of this scheme over the existing ones is that the shares being distributed to the participants is totally independent of the secret being shared. Hence, no restrictions are imposed on the scheme and it finds a wider use in real world applications.

Keywords: Generalized Secret Sharing, Access Structures

1 Introduction

In a secret sharing scheme a dealer has a secret key. There is a finite set P of participants and a set γ of subsets of P. A secret sharing scheme with γ as the access structure is a method which the dealer can use to distribute shares to each participant so that a subset of participants can determine the key if and only if that subset is in γ. The share of a participant is the information sent by the dealer in private to the participant. A secret sharing scheme is ideal if any subset of participants who can use their shares to determine any information about the key can in fact actually determine the key, if the set of possible shares is the same as the set of possible keys.

The threshold secret sharing schemes proposed in [1] and [2] are now being used widely in numerous real world applications. Some of the applications which require the frequent use of secret sharing schemes are cloud computing, banking sectors and sensor networks [3]. For the past ten years, threshold implementations are also used for masking to combat side channel attacks [4]. Unfortunately, with the advent of cloud computing, internet of things and big data, many desired applications cannot be implemented by just threshold implementations. These require more generalized secret sharing schemes which can implement any access structure with more restrictions. Many works on generalized secret sharing schemes exist in the literature but most of these are infeasible in the practical scenario. This is due to the requirement of an enormous number of shares for every user. As proposed in [5], monotone access structures can be theoretically implemented but in the worst case, each of the n users has to hold on to 2^n shares. In such a scenario, the key management system becomes too convoluted and practically unrealizable.

It is proven in [6] that there exists no threshold secret sharing scheme for arbitrary monotone functions. It has been shown that a fully generalized secret sharing scheme for any arbitrary γ does not exist without 2^n space and time complexity. In [2], multi-level and compartmented secret sharing schemes are implemented which are more general than threshold secret sharing schemes. Secret sharing schemes have also been generated from lattices [7]. Each of these schemes are suitable for specific applications but none of them is fully generalized such that any arbitrary γ can be implemented.

In this paper, we propose an algorithm to realize fully generalized secret sharing scheme for arbitrary access structures. This implies that a group of participants, say α, can access the secret if and only if $\alpha \epsilon \gamma$. All other groups of participants, including all subsets and supersets of α, will get a garbage value if they input their shares in the algorithm. If γ is a monotone access structure, then all the supersets of the authorized sets present in γ will also be able to gain access.

The paper first outlines the famous threshold secret sharing scheme. Then we propose our generalized secret sharing scheme followed by its security proof. We describe a possible attacker model and how to prevent the adversary from benefiting from it. The next section outlines the secure parameters to be used for the implementation of the algorithm followed by a section stating some of the possible future works related to this scheme. In the end, we conclude the paper.

2 Secret Sharing

Let S be a finite domain of secrets. A secret sharing scheme realizing an access structure γ is a scheme in which the input of the dealer is a secret $s \epsilon S$ such that the following two requirements hold:

- **Reconstruction requirement:** The secret s can be reconstructed by any authorized set. That is, for any set $G \epsilon \gamma$ ($G = \{i_1, ..., i_{|G|}\}$), there exists a reconstruction function $h_G : S_{i_1} \times ... \times S_{i_{|G|}} \Rightarrow S$ such that for every secret s and every random input r,

$$\text{if } \pi(s,r) = \langle s_1, s_2, ..., s_n \rangle \text{ then } h_G\{s_{i_1}, ..., s_{i_{|G|}}\} = s$$

- **Security Requirement:** Every unauthorized set of participants cannot reveal any partial information about the secret s. Stating this explicitly: for any $B \notin \gamma$, for every two secrets $a_1, a_2 \epsilon S$, and for every vector of possible pieces $\{s_i\}_{i \epsilon B}$:

$$\text{Pr } [\Lambda_{P_i \epsilon B} \; \pi_i(a_i, r) = s_i] = \text{Pr } [\Lambda_{P_i \epsilon B} \; \pi_i(a_2, r) = s_i]$$

where the probabilities are taken over the random input of the dealer.

3 Threshold Secret Sharing Scheme

The (n, t) threshold secret sharing scheme states that the secret S is divided into n shares and each of the n participants receive one share each.

- The secret S can be reconstructed if at least t participants come together and use their shares.
- The secret S cannot be reconstructed if less than t participants come together and combine their shares.

A simple model is discussed over here for the purpose of introduction of the scheme. A polynomial,

$$y(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + ... + a_1x + S, \; a_i \; \epsilon \; \mathbb{R}$$

is constructed where S is the secret. The shares of the secret given to each user are the points (x_i, y_i) which lie on the $y(x) = 0$ curve in the real plane. To reconstruct the secret from the shares, the coefficients of the polynomial have to be calculated. This can be done by solving the system of linear equations and obtaining S. The system of linear equations can be obtained by substituting at least t pairs of (x_i, y_i) in the above polynomial and assigning $y(x) = 0$. At least t such shares are required to do this as this is a polynomial of degree $(t-1)$ with t unknown coefficients. If less than t shares are available, then the system of equations for solving for a_i, $i \; \epsilon \; \{1, 2, ..., (t-1)\}$ and S will have infinite solutions. Hence, this is an efficient implementation of the (n, t) threshold secret sharing scheme. Algorithms for solving a system of linear equations using interpolation techniques show that this computation can be done in $O(nlog^2n)$ complexity.

4 Generalized Secret Sharing Scheme

In this section, we discuss the proposed generalized secret sharing scheme. First, we discuss how the dealer distributes the shares among the participants followed by the method to retrieve the secret. The next part of this section outlines the security proof of this scheme. The last part demonstrates the implementation of monotone access structures using this scheme.

4.1 Distributing the shares

Let the participants be $\{A, B, C, D, E, F, G\}$ and γ be the access structure. The steps for the generalized secret sharing scheme are stated below:

1. *Share Distribution Mechanism :* Every user is given a unique prime number as a share. In the aforementioned example, the primes distributed to $A, B, ..., G$ are $p_A, p_B, ..., p_G$ respectively.

2. *Characterizing each subset in γ :* Every subset i belonging to γ is assigned a characteristic number c_i such that,

$$c_i = \prod_{\forall j \epsilon i} p_j$$

   As every subset i consists of a unique set of participants, every c_i is a product of a unique set of prime numbers. This ensures that c_i is exclusive to i.

3. *Polynomial Construction:* When a subset i of participants contribute their shares, they retrieve the secret S if $i \ \epsilon \ \gamma$. Otherwise a garbage value is obtained. To execute this scheme, we construct the following polynomial $y(x)$ Let k be the number of subsets of participants present in γ.

$$y(x) = (x - c_1)(x - c_2)...(x - c_k) + S$$

   Unlike threshold secret sharing schemes, this polynomial is expanded and made public to all the participants.

4.2 Retrieving the secret

The steps to retrieve the secret are described below:

1. The set of participants (let it be denoted by α) contribute their individual shares and compute r.

$$r = \prod_{i=1}^{n} p_{\alpha_i}$$

   where n is the number of participants in α, α_i denotes the i^{th} participant in α and p_{α_i} denotes the prime number assigned to the participant α_i.

2. $y(r)$ is computed by the participants. Every participant had received a polynomial $y(x)$. $y(r)$ can be easily computed by putting $x = r$ in the polynomial. If $\alpha \ \epsilon \ \gamma$, $y(r) = S$. Otherwise $y(r)$ returns a garbage value.

4.3 Implementing Monotone Access Structures

Definition: Let $\{P_1, P_2, ...P_n\}$ be the set of participants. A collection $\gamma \subseteq 2^{\{P_1,...,P_n\}}$ is monotone if $B \epsilon \gamma$ and $B \subseteq C$ implies $C \epsilon A$.

Our scheme does not implement monotone access structures implicitly. Only the sets of users present in γ are treated as authorized sets of users. Since this scheme is highly generic, monotone access structures can also be generated using this scheme. To realize a monotone access structure A, we create a new access structure A' such that all the possible supersets of the authorized sets present in A are included in A'. Formally,

$$A' = A\ U\ D$$
$$where\ D = \{\alpha' \mid \alpha \subset \alpha',\ \alpha \epsilon A\}$$

Then, we design our aforementioned secret sharing scheme for the access structure A'.

The performance of this algorithm deteriorates significantly when it is modified to implement monotone functions. The time required for this algorithm is of the order $O(k)$, where k is the degree of the polynomial $y(x)$.

$$k = \textit{degree of y(x)} \quad (1)$$
$$= n(A) \quad (2)$$

When the access structure A is modified to A', the cardinality increases exponentially. Let the number of authorized sets in A be k, the average number of participants in an authorized set be v and the total number of participants be n.

$$\text{If } n(A) = k \times v$$
$$\text{then } n(A') = O(k.2^{n-k})$$

5 Proof of Security

A secret sharing scheme is said to be secure if it fulfils the following comditions:

- A subset of participants (let it be denoted by α) should be able to retrieve the secret deterministically if and only if $\alpha \epsilon \gamma$.
- Any other set of participants should not be able to retrieve the secret deterministically, that is they can retrieve the secret with a negligible probability.
- Proper subsets of α should not be able to retrieve the secret.
- If the secret sharing scheme implements monotone access structures, then a subset of participants β, where $(\beta \neq \alpha)$, can retrieve the secret if $\alpha \subseteq \beta$.

The security proof for this scheme is outlined below. Prior to that we show that,

$$y(x) = (x - c_1)(x - c_2)...(x - c_k) + S \quad (3)$$
$$\Rightarrow y(x) = g(x) + S \quad (4)$$

$$where, \quad g(x) = (x - c_1)(x - c_2)...(x - c_k)$$

Let the set of participants contributing their shares be denoted by β. Let

$$\gamma = \{\alpha_1, \alpha_2, ..., \alpha_k\}$$

where α_i $(i \epsilon \{1, 2, ..., k\})$ is a set of participants allowed to access the secret.

- $\beta = \alpha_i$: This is the case when a set of n participants belonging to γ come together to retrieve the secret. The value r is computed as per the method of retrieval of the secret.

  $$r = \prod_{j=1}^{n} p_{\beta_j}$$

  $$\Rightarrow r = \prod_{j=1}^{n} p_{\alpha_{i,j}}$$

  $$\Rightarrow r = c_i$$

  c_i is a root of the polynomial $g(x)$. Therefore,

  $$g(r) = g(c_i) = 0$$

  $$y(r) = g(r) + S$$
  $$\Rightarrow y(r) = S$$

  Hence, the secret S is successfully retrieved.

- $\beta \neq \alpha_i$: This is the case when a set of n participants not belonging to γ come together to retrieve the secret. The value r is computed as per the protocol.

  $$r = \prod_{j=1}^{n} p_{\beta_j}$$

  $$r \neq \prod_{j=1}^{n} p_{\alpha_{i,j}}$$

  $$\Rightarrow r \neq c_i$$

  $$\Rightarrow g(r) \neq 0$$

  $$y(r) = g(r) + S$$
  $$\Rightarrow y(r) \neq S$$

  Hence, the secret S is not retrieved. As every participant has a unique prime number, it is ensured that c_i can be constructed only when the participants of α_i contribute their shares. This property prevents any other set of participants from retrieving the secret.

It is not possible for any adversary to deterministically decompose $y(x)$ into $g(x) + S$. This is because there are $(k+1)$ variables, namely $c_1, c_2, ..., c_k$ and S. The adversary has access to only k equations. These equations are obtained by equating the algebraic coefficients of $x^{(k-1)}, x^{(k-2)}, ..., x$ and the constant of the polynomial $y(x)$ to their corresponding numerical values.

6 An Attacker Model

We show that an attacker can guess a range of values in which S lies. However, we can reinforce the security of the scheme by maximizing this range so that the adversary cannot guess the secret effectively.

The attacker knows that $y(x) = g(x) + S$ and $g(x)$ has k roots. He computes the values δ_1 and δ_2 such that

$$\begin{aligned} z_1(x) &= y(x) + \delta_1 \\ z_2(x) &= y(x) + \delta_2 \end{aligned}$$

so that δ_1 and δ_2 are the minimum and maximum values respectively for which $z_1(x)$ and $z_2(x)$ have exactly k roots. The secret $S \ \epsilon \ (-\delta_2, -\delta_1)$.

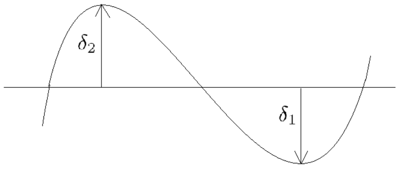

Fig. 1: The values of δ_1 and δ_2 for $y(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

To make this attacker model inefficient for the adversary, the function $y(x)$ should be modified appropriately to maximize the values of δ_1 and δ_2.

7 Parameters of Implementation

The polynomial $y(x)$ cannot be computationally decomposed to $g(x) + S$ deterministically. Still, to ensure maximum security of this scheme we choose large prime numbers to be distributed among the participants. Prime factorization being a well-known hard problem, makes the decomposition of $y(x)$ into $g(x) + S$ virtually impossible. The security can be further increased by choosing the primes to be of the same order so that no information of any kind is leaked to the adversary.

To ensure the efficient performance of the scheme, it is desirable to restrict the number of authorized sets from becoming too high. The time required to compute $y(x)$ for a given value of x is $O(k)$, where k is the degree of $y(x)$. The space complexity for storing all the coefficients of $y(x)$ is $\theta(k)$. As shown earlier,

$$k = n(\gamma)$$

The scheme becomes computationally infeasible if the $k = O(2^n)$. This is not a problem in real world applications where $k << 2^n$ but for optimal theoretical security and efficiency we would restrict k to be at most a subexponential function of n.

8 Future Work

Future work on this scheme may include making it more robust and effective when the number of authorized sets increase. Scope of research lies in finding new adversary models and measures to prevent them. Security bounds can formally be defined by applying intricate complexity theory on this scheme. Optimizations can be done on this scheme to make it more effective for specific applications.

9 Conclusion

This novel scheme is one of the first schemes to realize (strict) generalized ideal secret sharing. The previous schemes have restricted themselves to the realization of monotone access structures. This secret sharing scheme can be utilized effectively in implementing access control and authentication of groups in an environment with a large number of participants. This includes applications in cloud computing, banking sectors and internet of things. This scheme is lightweight, hence it can also be incorporated into resource constrained environments such as embedded devices.

References

1. A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979. [Online]. Available: http://doi.acm.org/10.1145/359168.359176
2. G. R. Blakley and G. Kabatianskii, "Linear algebra approach to secret sharing schemes," in *Error Control, Cryptology, and Speech Compression - Workshop on Information Protection, Moscow, Russia, December 6-9, 1993, Selected Papers*, 1993, pp. 33–40. [Online]. Available: http://dx.doi.org/10.1007/3-540-58265-7_5
3. H. Chan, A. Perrig, and D. X. Song, "Random key predistribution schemes for sensor networks," in *2003 IEEE Symposium on Security and Privacy (S&P 2003), 11-14 May 2003, Berkeley, CA, USA*, 2003, p. 197. [Online]. Available: http://dx.doi.org/10.1109/SECPRI.2003.1199337
4. S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, 2006, pp. 529–545. [Online]. Available: http://dx.doi.org/10.1007/11935308_38

5. 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, pp. 56–64, 1989.
6. J. C. Benaloh and J. Leichter, “Generalized secret sharing and monotone functions,” in *Advances in Cryptology - CRYPTO ’88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, 1988, pp. 27–35. [Online]. Available: http://dx.doi.org/10.1007/0-387-34799-2_3
7. H. YAMAMOTO, “Proposal of a lattice-based visual secret sharing scheme for color and gray-scale images,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 81, no. 6, pp. 1262–1269, 1998.