ISSN 2320 - 2602

Volume 3, No.2, February 2014

International Journal of Advances in Computer Science and Technology

Available Online at http://warse.org/pdfs/2014/ijacst07322014.pdf

Generalization Of Secret Sharing Scheme Using Python

Siyaram Gupta[1], Madhu Sharma[2]

[1]Master of Technology, Dehradun Institute of Technology, Dehradun, India, siyaram421@gmail.com

[2]Assistant Professor, Dehradun Institute of Technology, Dehradun, India,madhuashishsharma@gmail.com

ABSTRACT

Sometimes it is important for the owner of the information to secure it from unauthorized parties. There are many applications such as medical image security, EPR hiding [1] and records that must need to be shared or distributed among a group of participants. Secret Sharing scheme proposed by Adi Shamir in 1979 [2] meets these requirements. According to his well-known (k, n) threshold scheme at-least k participant must gather to regenerate the secret. The scheme proposed by Shamir is based on Lagrange interpolation polynomial, we use polynomial to construct shares for participants [3]. His scheme ensures both confidentiality and integrity of information. This scheme is useful and implementable in many aspects. Many scholars use this scheme to hide or secure the information and secrets. We are implementing the secret sharing scheme using Python. Python is a user-friendly and open source language. It is easy to generate and use polynomial using python.

Key words : Cryptography, Secret sharing, Polynomials, Python.

1. INTRODUCTION

Security and protection of information is a major issue in almost every field using insecure channels for transmission of data among different parties. Such concerns led us to use cryptographic techniques to maintain the confidentiality of information, it also assure that information should not be tempered in any way during its transmission. In cryptography, secret sharing is a scheme through which a secret or information is distributed among a number of parties. The major issue is that how to distribute or share secrets in such a way that only authorized person can access. Shamir's scheme divides the information into n pieces so that each participant gets their share. Each participant also has a unique value for their identification as valid participant. The secret is revealed if and only if k participants must gather k-1 participants do not allow recovering the secret. In general, in this scheme there are n players and one dealer, the dealer is responsible for distribution of shares but only when some predefined conditions are fulfilled. The dealer use such scheme by distributing the share in such a way that the particular group of 't' or more members which has the share can together recover the secret but no group which has less than 't' members can able to access the secret. The scheme is named as Shamir's secret sharing scheme or a (t, n) - threshold scheme.

Processing and handling information by computers and sharing them over high-speed network infrastructure have become a common practice since wide deployment of low cost computing and networking hardware. Currently, student/teacher text files and records are stored on disks of college/university database systems for fast and reliable storage and retrieval. Protection of the integrity and confidentiality of department information is an issue in the management of student/teacher personal & professional records. Confidentiality states that unauthorized parties should not be granted to access those records during transmission. Integrity, on the other hand, implies that information such as results should not be modified in any way during transmission.

These schemes use polynomial interpolation for shares. There are k points in the 2-D plane (X_i, Y_i), ……… (X_k, Y_k) with unique X_i's values, there is one polynomial h(x) of degree (k-1). Modular arithmetic is more suitable than real arithmetic for such purpose. Pick an integer prime number P in this way that is bigger than the secret S and the number of members' n. The (k, n) threshold scheme has some properties such as the size of each share does not exceed the secret size. Share S_i can be added or deleted dynamically once k is kept fixed. Change in S_i pieces does not affect the original secret anymore. These properties of the scheme enhance security since exposed pieces by security holes can't be accumulated until the values of all are from same edition of the polynomial. We can get a hierarchical scheme as S is determined by number of shares depends on importance. This is a hierarchical scheme and each share has its own important and values.

2. RELATED WORK

To keep the secret confidential and provide privacy of information, Shamir (1979) [2] and Blakley [4] first developed the concept of secret sharing scheme. There are lots of applications and areas that implements the concept of secret sharing such as the concept is used for hiding medical images and patients records etc.

However, as these schemes are discussed, there are several drawbacks which are further fix and improved by many researchers. Multi-secret sharing (MSS) [11-12] schemes have been proposed to overcome the drawback that shares

only single secret at a time. C.C. Yang, T.-Y. Chang, M.S. Hwang [6] proposed a new MSS, which is based on the two-variable one-way function [5] in 2004.
To overcome the problem of dishonest dealers and malicious participant scholars proposed the verifiable secret sharing (VSS) schemes. AVSS scheme allows participants to verify the validity of shares of the other participants and her/him. The first realization of VSS was written by Chor et al. [7] in 1985.
Thereafter, Harn [8] presented the verifiable multi-secret sharing (VMSS) scheme in 1995. Some researchers improve this scheme but the cost of computing in it is still high. The YCH scheme is a relatively efficient multi-secret scheme at the present time. But the issue is that the scheme doesn't have the property of verification.
A new literature [9] has a property of verification based on YCH scheme; the scheme is unpractical because of its big-ticket system. To overcome the drawback of the scheme ,later a new practical verifiable multi-secret sharing scheme proposed by Jianjie Zhao ,Jianzhong Zhang ,Rong Zhao [3] which is based on discrete logarithm [10] and intractability.

3. OVERVIEW OF SHAMIR'S SHARING SCHEME AND PYTHON

3.1 Overview of Shamir's Sharing Scheme

Shamir's secret sharing scheme divides the information into n pieces so that each participant gets their share. Each participant also has a unique value for their identification as valid participant. The secret is revealed if and only if k participants from pool of n must gather, k-1 participants do not allow to recovering the secret. The scheme proposed by Shamir is based on Lagrange interpolation polynomial; polynomial concept is used to construct shares for participants.
The scheme divides the secret S into n shares denoted by (S_1, S_2, ... S_2). A (k-1) degree polynomial with a large prime number P is used to compute shares as in (1).
$H(x) = (S+a_1 x+a_2 x^2+\cdots+a_{k-1}x^{k-1}) \bmod P$ (1)

(a_1, a_2, $\cdots$, a_{k-1}) are the coefficient of the polynomial selected randomly from within range (0, P]. The computed shares are as in (2).
$Y_1 = (1, H(1))$, $Y_2 = (2, H(2))$, $Y_3 = (3, H(3))$. . . . . $Y_n = (n, H(n))$ (2)

The shares are pair of two integers. If k of pairs gathers, then only participants are able to reconstruct the secret using Lagrange's interpolation technique. The constant term S of the polynomial is the secret. This scheme is suitable for secret sharing.

3.2 Overview of Python

Python language is used for the implementation of sharing scheme. Python is an interpreter, object-oriented ,high level scripting and programming language. Guido van Rossum was the first who introduce the python language in 1991**.** Python is very easy to learn and use and it's a open source language, it helps a lot at the time of implementation. A generation of polynomials is an easy task in python language. Python is also useful for us to generate random numbers. We are using GMPY for multi-precision in this scheme. It provides a great platform for its users; the manuals and documents provided by python help the beginner. It has almost all libraries and functions a user needs to implement his work. Numpy , gmpy , flint etc are the common one used by us during implementation. It has effective approach to object-oriented programming and high level, efficient data structure. Features like elegant syntax and dynamic typing make it one of the powerful language and because of its interpreted nature, python is used as scripting language , it is also useful in development of applications for most of the platforms. The data types and functions implemented in C or C++ (Cython) Java in the form of Jython are easily used in python. Python extends there features.
Following are the main features and functions that are used during implementation:-

Random Function: Used to generate random numbers such functions are "random.randrange()" and "random.getrandbits ()".

Example: a =random.getrandbits (64)
(Generate a random number of 64 bits.)

GMPY for multi-precision: A C-coded Python extension module that supports multiple-precision arithmetic. Gmpy support for the MPFR and MPC libraries. The gmpy mpz type supports arbitrary precision integers. Gmpy provides a rational type call mpq. Gmpy uses mpf or mpfr type to support multiple-precision reals.

"gmpy.mpz ()" is for multi-precision integer value.

```
Example:    import gmpy
            y=gmpy.mpz (64)
```
(A 64 bit gmpy integer variable)

Numpy Library: Python provide numpy library that is useful for numeric calculations. NumPy (Numeric Python) package provides basic routines for Array creation routines, Linear Algebra, Polynomials, Random numbers and much more.

```
Example:        import numpy as npp
      Array Creation        a = npp.array ([1, 4, 5, 8], float)
      Polynomials       npp.poly ([-1, 1, 1, 10])
      Random Numbers   npp.random.rand(5)
```

There are lots of applications of python such as scripting language for web applications, Libraries like Matplotlib, NumPy, SciPy and gmpy allow Python to be used effectively in scientific computing. Python has also been used in artificial intelligence and natural language processing tasks.

4. PROPOSED SCHEME

Shamir's sharing scheme demonstrates how a secret can be shared so that the secret is recovered only when sufficient no. of shares is available. This scheme gives basic logic of a secret being shared among five people. Out of these five people, if at least three people give their shares the secret can be regenerated.

4.1 Secret sharing using python

Thus in the Shamir's (n, k) we take (5, 3)**:**
We need only quadratic polynomial since it has three coefficients. The three coefficients are the secret to be shared. The five shares are generated by evaluating the coefficients (Y_s) at randomly chosen X_s. Thus (X[i], Y[i]) make up the ith share. Only when at least three such shares are available the original polynomial can be regenerated.

All calculations are done modular taking the mod as mymod = gmpy.mpz (2^{64} - 59), which is the largest 64-bit prime number. The mod has to be a prime as otherwise the modular arithmetic will become very tedious and impractical. It has two functions - quadpolymeval and quadpolymregen for evaluating and regenerating a quadratic polynomial. Poly1 is the original polynomial and Poly2 is the regenerated polynomial. The shares are randomly generated.

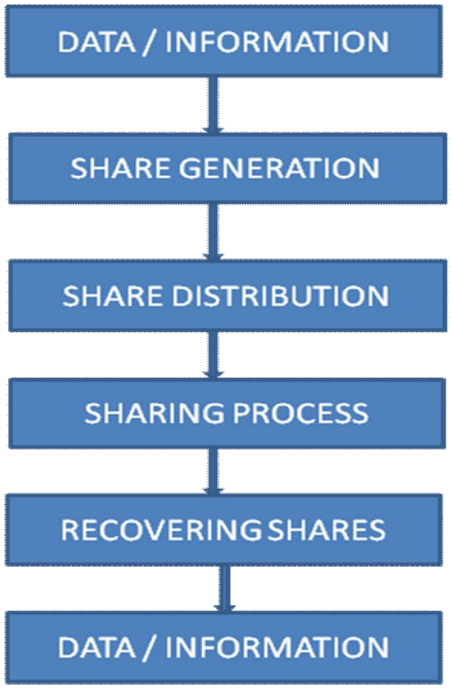

Figure 1 shows Secret Sharing Process

4.2 Implementation of Proposed Scheme

The Scheme has two main phases:

Construction Phase*:*
mymod = gmpy.mpz (2^{64} - 59)

quadpolymeval (coeffs, x, mod)
(This evals quad modular poly defined by coeffs at x, mod is the modulus.)

poly1 = [gmpy.mpz (random.randrange(0, mymod))]
(Poly1 is the evaluated or original quadratic polynomial.)

X_s = [gmpy.mpz(random.randrange(0, mymod))
(Randomly chosen value.)

Y_s = [quadpolymeval(poly1, X_s[0], mymod)
(Generate share for each participants.)

Recovery Phase:
quadpolymregen(X_s, Y_s, mod)
(Finds modular quad polynomial passing through points given by X_s & Y_s).

poly2 = quadpolymregen ([X_s[0], X_s[1], X_s[2]], [Y_s[0], Y_s[1], Y_s[2]], mymod)
(Poly2 is the recovered quadratic polynomial.)

Poly 1 is the original polynomial and Poly 2 is the regenerated polynomial calculated using quadpolyregen function.

```
$ ipython
Python 2.7.5 (default, Jan  10 2014, 09:40:52)

In [1]: execfile("mypolym.py")
In [2]: poly1
Out [2]: [mpz(796391978242292738),
mpz(17069081006830707926L),
mpz(3726351682425429984)]

In [3]: poly2
Out [3]: [mpz(796391978242292738),
mpz(17069081006830707926L),
mpz(3726351682425429984)]
```

If a part, say X_s[1] and Y_s[1], is repeated, we get error and the polynomial is not regenerated:

```
In [4]: poly3 = quadpolymregen([Xs[0], Xs[1],
Xs[1]], [Ys[0], Ys[1], Ys[1]], mymod)
------------------------------------------------------------------------
--
ZeroDivisionError              Traceback (most recent call last)
<ipython-input-5-dbafdc471c87> in <module> ()
----> 1 poly3 = quadpolymregen([Xs[0], Xs[1],  Xs[1]],
[Ys[0], Ys[1], Ys[1]], mymod)

/home/ash/myresearch/Cryptography/mypolym.py in
quadpolymregen(Xs, Ys, mod)
28                  for j in range(0, 3):
29                      if(j != i):
---> 30      coeffspart[2] *=  gmpy.divm(1, Xs[i] – Xs[j], mod)
31            coeffspart[1] -= Xs[j]
```

```
32          coeffspart[0] *=   - Xs[j]

ZeroDivisionError: not invertible
```

Also, if a part, say $Y_s[3]$ in place of $Y_s[2]$, is repeated, we get error and the polynomial is not regenerated:

```
In [5]: poly4 = quadpolymregen ([Xs[0], Xs[1], Xs[2]], [Ys[0], Ys[1], Ys[3]], mymod)

In [6]: poly4
Out[6]: [mpz (13145891074882953838L), mpz
(543580718118861428), mpz (3960423652794329163)]

In [7]:
```

5. CONCLUSION AND FUTURE SCOPE

In this paper, we adopt the concept of secret sharing first introduced by Adi Shamir. The scheme has the property of sharing and securing the intellectual work of the people. The main aim of such a concept is to ensure and satisfy the security of information and to fix privacy issues. The scheme prevents unintentional disclosure of information to those who are not allowed to access it. Secret sharing scheme is a very powerful concept of sharing. The scheme assures privacy, confidentiality, integrity and authentication of the information. Polynomial coefficients used in this secret sharing scheme provide us to hide the important information. Implementation of this scheme using python makes it more secure and powerful. Further we will enhance this scheme by using cryptographic techniques. By using encryption technique the shares should be more secure.

REFERENCES

[1] **“Medical image security and EPR hiding using Shamir’s secret sharing scheme”**,Mustafa Ulutas, Güzin Ulutas∗, Vasif V. Dept. of Computer Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey.

[2] “A**. Shamir, How to share a secret”**, Communications of the ACM 22 (11) (1979) 612–613.

[3] **“A practical verifiable multi-secret sharing scheme”**. Jianjie Zhao, Jianzhong Zhang, Rong Zhao. College of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710062, People's Republic of Chin b Natural Science Institute, Xi’an University of Technology, Xi’an 710048, People’s Republic of China.

[4] G. Blakley, **“Safeguarding cryptographic keys”**, Proc AFIPS 1979 National Computer Conference, AFIPS Press, New York, 1979, pp. 313–317.

[5] J. He, E. Dawson, **“Multisecret-sharing scheme based on one-way function”**, Electronics Letters 31 (2) (1995) 93–95.

[6] C.-C. Yang, T.-Y. Chang, M.-S. Hwang, **“A (t,n) multi-secret sharing scheme”**, Applied Mathematics and Computation 151 (2004) 483–490.

[7] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, **“Verifiable secret sharing and achieving simultaneity in the presence of faults”**, Proc. 26th IEEE Symp. FOCS, 1985, pp. 251–260.

[8] L. Harn, **“Efficient sharing (broadcasting) of multiple secret”**, Computers and Digital Techniques 142 (3) (1995) 237–240.

[9] J. Shao, Z.-F. Cao, **“A new efficient (t,n) verificable multi-secret sharing (VMSS) based on YCH scheme”**, Applied Mathematics and Computation 168 (2005) 135–140.

[10] R.-J. Hwang, C.-C. Chang, **“An on-line secret sharing scheme for multisecrets”**, Computer Communications 21 (13) (1998) 1170–1176.

[11] H.-Y. Chien, J.-K. Tseng, **“A practical (t,n) multi-secret sharing scheme”,** IEICE Transactions on Fundamentals of Electronics, Communications and Computer 83-A (12) (2000) 2762–2765.

[12] J. He, E. Dawson, **“Multistage secret sharing based on one-way function”**,Electronics Letters 30 (19) (1994) 1591–1592.