APPLICATIONS OF SECURE MULTIPARTY COMPUTATION IN SECRET SHARING FOR TACTICAL ENVIRONMENTS

MAHRUD SAYRAFI AND SIMON WOO

Abstract. Secret sharing schemes are methods using which a secret can be distributed between a set of N players in a way that at least K players ($K \leq N$) have to combine their pieces of the secret in order to reconstruct the original secret, but no information about the secret can be gained otherwise.

Secure multiparty computation algorithms are methods that intend to answer question of how can we compute a function while keeping the inputs secret from all parties.

It is well known that many secret sharing methods can be used to construct secure and efficient multiparty computation algorithms and recent papers and academic endeavours in both fields focus on introducing more complex (albeit computationally faster) methods to achieve that goal and to improve security of MPC algorithms against mallicious or fake parties. However, MPC algorithms can also be used to improve secret sharing methods.

This paper introduces a simple method of applying a rudimentary MPC method to secure Shamir's secret sharing method against a range of attacks in an attempt to attract interest in applying modern MPC methods to create better secret sharing methods.

Contents

1. Introduction 2
1.1. The Application Scenario 2
2. Preliminary Cryptographic Protocols 4
2.1. Vandermonde Matrix 4
2.2. Shamir's Secret Sharing Scheme 4
2.3. Secure Multiparty Computation 6
3. Secure Multiparty Reconstruction 6
4. Contributions of this Paper and Conclusion 7
5. Discussion and Future Works 7
6. Acknowledgement 8
Appendix A. Autonomous Information Unit 8
References 8

Date: September 19, 2013.
Key words and phrases. Secret Sharing, Secure Multiparty Computation.

1. Introduction

In this section we provide few real world analogies that can help readers of different disciplines understand the goals and applications of this paper.

1.1. **The Application Scenario.** Suppose you are a bank owner with N managers who need to access the vault, however, since one can never be too cautious, you want to grant them access to the vault if and only if at least $K \leq N$ of them are present. The first solution that comes to mind is to make a few copies of the key and break them into small pieces and distrtibute them between your managers; this method is very complicated due to the fact that the required number of keys increases exponentially to a point where it is impractical to use this method.

To a cryptologist's eye, however, this puzzle looks as simple as the following high school pre-calculus problem:

Problem 1.1. *Interpolation Problem*
Imagine that there is an unknown function $f(x)$ for which someone supplies you with its values at K distinct points $x_0 < x_1 < ... < x_k$ – i.e., $(x_0, y_0), (x_1, y_1), ..., (x_k, y_k)$ are given. Construct a degree $K - 1$ function $f(x)$ that passes through these points.

It should be easy to imagine that we can consider the coefficients of the function as our key. Now we can generate as many points on the polynomial as needed and distribute them between the managers and rest assured that any K distinct points are sufficient to reconstruct the key while $K - 1$ points will not reveal any information about the secret.

In cryptology, these systems are referred to as Secret Sharing System because the goal is to share a secret, namely the key, between a group of participants. In fact the method described above is the backbone of a system described in section 2.2.

The problem that this paper intends to answer is, how do you reassemble these pieces of the secret securely? Or more precisely, who reassembles them? Can we trust a single person to do that? What if a thief disguised as a bank manager was chosen to reassemble the pieces and once he has access to the pieces, he took copies of them? Even outside the analogy, at the end of the day one CPU is processing the reconstruction algorithms, so whoever has access to that CPU can plant a backdoor to steal the pieces. Once we simplified the problem, we realized that all that is needed to do is to somehow keep the individual shares secret as well; but how is it possible to compute an algorithm over secret inputs?

Suppose you are attending your high school reunion and while having dinner with two old friends you wonder what is the average salary of you three, but nobody is willing to reveal his own salary. Here we have the same problem: how do we compute a simple function over a set of inputs that need to be kept secret.

Just as before, a cryptologist sees this enigma as a simple problem that only requires knowledge of middle school algebra – once again, mathematics to the rescue! Here we will explain a basic protocol for this problem and in section 2.3 a general solution is provided.

Average Salary Protocol

Suppose three participants X, Y, and Z with three secret numbers x, y, and z want to calculate the average of their secrets. The simplest method of doing so is as follows:

(1) Z and Y choose on a random number r and keep it secret from X. This number doesn't have to be calculated by both of them in a multiparty setting; i.e., the older participant can choose a random number and tell the younger one.
(2) Z privately sends $z + r$ to X.
(3) Y privately sends $y - r$ to X.
(4) Now X knows the following numbers: x, $z + r$, $y - r$. It is evident that X can simply add those numbers and since $+r$ and $-r$ cancel each other, X will achieve $S = x + y + z$.
(5) X divides S by three to find the average and anounces the result.

This system is secure against passive attackers, meaning that X, Y, and Z are trusted and have established secure channels between each other; i.e., X cannot eavesdrop on Y and Z to steal r.

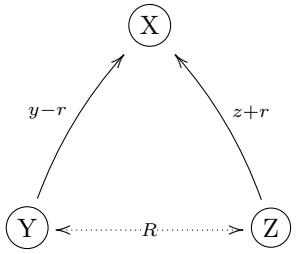

FIGURE 1. Multiparty Addition Between Three Members

Back to the original problem, now empowered by the knowledge of Secret Sharing Systems and Secure Multiparty Computation, how can we achieve our goal, namely to reconstruct the secret without risking the pieces falling into wrong hands?

In section 2 we will describe the general format of the tools that we need in details and in section 3 we will explain the Secure Multiparty Reconstruction algorithm that is the main contribution of this paper.

Further, we will show the outline of possible lines for future work.

2. Preliminary Cryptographic Protocols

In this section we present the preliminary cryptographic protocols used in this paper.

2.1. **Vandermonde Matrix.** A Vandermonde matrix, named after Alexandre-Théophile Vandermonde, is a matrix in which each row is a geometric progression with distinct bases $a_0, a_1, \cdots, a_{n-1}$. For instance the following is a $m \times n$ Vandermonde matrix:

$$V = \begin{pmatrix} a_0^0 & a_0^1 & a_0^2 & \cdots & a_0^{n-1} \\ a_1^0 & a_1^1 & a_1^2 & \cdots & a_1^{n-1} \\ a_2^0 & a_2^1 & a_2^2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m-2}^0 & a_{m-2}^1 & a_{m-2}^2 & \cdots & a_{m-2}^{n-1} \\ a_{m-1}^0 & a_{m-1}^1 & a_{m-1}^2 & \cdots & a_{m-1}^{n-1} \end{pmatrix}$$

An interesting property of this matrix is that the determinant of a square Vandermonde matrix (i.e., $m = n$) can be written as:

$$|V| = \prod_{0 \le i < j < n} (a_i - a_j)$$

Hence, as long as all bases are distinct the determinant is nonzero, thus the matrix is invertible.

2.2. **Shamir's Secret Sharing Scheme.** In [1], Adi Shamir proposed a secret sharing scheme based on polynomial interpolation (see figure 2.2) that soon became one of the most well known schemes in the field.

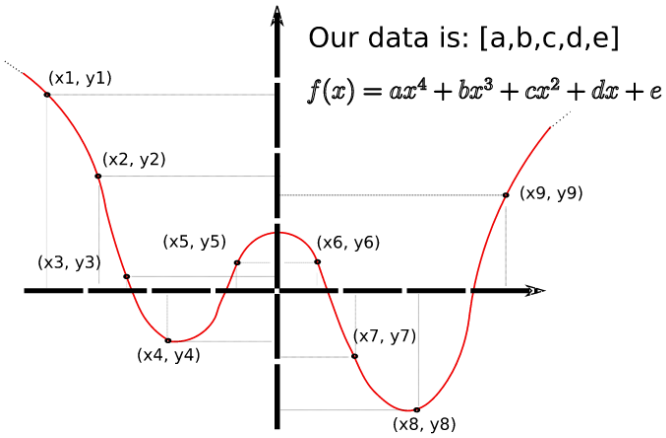

Figure 2. Any $k = 5$ of the $n = 9$ points here are sufficient to interpolate the degree-4 polynomial $f(x) = ax^4 + bx^3 + cx^2 + dx + e$.

In this section we demonstrate the matrix representation of this method:
We define matrix $V_{n\times k}$ as a Vandermonde matrix with distinct bases $x_0, x_1, ..., x_{n-1}$:

$$V = \begin{pmatrix} x_0^0 & x_0^1 & x_0^2 & \cdots & x_0^{k-1} \\ x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^{k-1} \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-2}^0 & x_{n-2}^1 & x_{n-2}^2 & \cdots & x_{n-2}^{k-1} \\ x_{n-1}^0 & x_{n-1}^1 & x_{n-1}^2 & \cdots & x_{n-1}^{k-1} \end{pmatrix}$$

Then, in order to divide a secret vector $S = \begin{pmatrix} s_0 & s_1 & s_2 & \cdots & s_{k-1} \end{pmatrix}$ into n pieces such that any k can reconstruct s, we simply find the vector $D = VS$:

$$D = VS = \begin{pmatrix} x_0^0 & x_0^1 & x_0^2 & \cdots & x_0^{k-1} \\ x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^{k-1} \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-2}^0 & x_{n-2}^1 & x_{n-2}^2 & \cdots & x_{n-2}^{k-1} \\ x_{n-1}^0 & x_{n-1}^1 & x_{n-1}^2 & \cdots & x_{n-1}^{k-1} \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{k-1} \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} \end{pmatrix}$$

Now we can distribute each pair (x_i, d_i) between the players (a.k.a. shareholders). To reconstruct the secret, once k players with pairs of (a_i, b_i) have gathered, we can construct the square Vandermonde matrix A as:

$$A = \begin{pmatrix} a_0^0 & a_0^1 & a_0^2 & \cdots & a_0^{k-1} \\ a_1^0 & a_1^1 & a_1^2 & \cdots & a_1^{k-1} \\ a_2^0 & a_2^1 & a_2^2 & \cdots & a_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k-2}^0 & a_{k-2}^1 & a_{k-2}^2 & \cdots & a_{k-2}^{k-1} \\ a_{k-1}^0 & a_{k-1}^1 & a_{k-1}^2 & \cdots & a_{k-1}^{k-1} \end{pmatrix}$$

And the vector B as:

$$\begin{pmatrix} b_0 & b_1 & b_2 & \cdots & b_{k-1} \end{pmatrix}$$

Then, knowing that in a square Vandermonde matrix with distinct bases the determinant is nonzero, we calculate the inverse A^{-1} such that:

$$AA^{-1} = I$$

So for any vector v we have:

$$AvA^{-1} = v$$

Thus, we can reconstruct secret vector S:

$$BA^{-1} = ASA^{-1} = S$$

2.3. **Secure Multiparty Computation.** In [2], Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen described a multiparty Secure Addition Protocol that is a generalization of the Average Salary Protocol described in the introduction:

SECURE ADDITION Protocol

Participants are P_1, P_2, P_3, input for P_i is $x_i \in Z_p$, where p is a fixed prime agreed upon in advance.

(1) Each P_i computes and distributes shares of his secret x_i as described in the text: he chooses $r_{i,1}$, $r_{i,2}$ uniformly at random in Z_p, and sets $r_{i,3} = x_i - r_{i,1} - r_{i,2} \mod p$.
(2) Each P_i sends privately $r_{i,2}$, $r_{i,3}$ to P_1, $r_{i,1}$, $r_{i,3}$ to P_2, and $r_{i,1}$, $r_{i,2}$ to P_3 (note that this involves P_i sending "to himself"). So P_1 for instance, now holds $r_{1,2}$, $r_{1,3}$, $r_{2,2}$, $r_{2,3}$ and $r_{3,2}$, $r_{3,3}$.
(3) Each P_j adds corresponding shares of the three secrets – more precisely, he computes, for $l \neq j$, $s_l = r_{1,l} + r_{2,l} + r_{3,l} \mod p$, and announces s_l to all parties (hence two values are computed and announced).
(4) All parties compute the result $v = s_1 + s_2 + s_3 \mod p$.

3. Secure Multiparty Reconstruction

Here we introduce a method for reconstruction of the shares in a secure multiparty fashion. This method is based on a preprocessing step in which each party generates a session share. This share is different in each session consisting of different parties, thus the scheme is safe against reply attacks; i.e., the information gathered by a malicious party during one session is useless in a session with different members.

Let $P_0, P_1, P_2, ..., P_{k-1}$ to be a set of K parties each with a piece if the form (x_i, d_i) gathered to reconstruct their shares and let the vector $D = \begin{pmatrix} d_0 & d_1 & d_2 & \cdots & d_{k-1} \end{pmatrix}$ be the set of their secrets. First we construct the Vandermonde matrix V with entries equivalent to the x values of the parties; that is, let the first row be a geometric series with base x_0 and so on. Now we publicly calculate V^{-1}, the inverse Vandermonde matrix equivalent to those parties. As observed before, it is clear that $V^{-1}D = s$. Let us define the inverse Vandermonde matrix as:

$$V^{-1} = \begin{pmatrix} v_0 & v_1 & v_2 & \cdots & v_{k-1} \end{pmatrix}$$

Note that each item v_i is a column of the matrix. Thus, we have:

$$V^{-1}D = \begin{pmatrix} v_0 & v_1 & v_2 & \cdots & v_{k-1} \end{pmatrix} \begin{pmatrix} d_0 & d_1 & d_2 & \cdots & d_{k-1} \end{pmatrix} =$$
$$= d_0v_0 + d_1v_1 + d_2v_2 + ... + d_{k-1}v_{k-1} = S$$

What we realize is that party i only requires to know d_i along with v_i, and what follows is simply a sum of columns which can be performed using the secure addition protocol.

Secure Multiparty Reconstruction Protocol

N parties have shared a secret among themselved using Shamir's secret sharing method. Suppose k members with pieces (x_i, d_i) have gathered to reconstruct the secret, then they must do the following steps:

(1) The parties publicly generate the inverse Vandermonde matrix V^{-1} equivalent to their x values and announce it. Let $V^{-1} = \begin{pmatrix} v_0 & v_1 & v_2 & \cdots & v_{k-1} \end{pmatrix}$
(2) Party i calculates his session share:
$$d_i * v_i = \begin{pmatrix} d_i v_{i,0} & d_i v_{i,1} & d_i v_{i,2} & \cdots & d_i v_{i,k-1} \end{pmatrix} =$$
$$= \begin{pmatrix} s_{0,i} & s_{1,i} & s_{2,i} & \cdots & s_{k-1,i} \end{pmatrix}$$
(3) Now, the group performs the Secure Addition protocol K times, each time finding: $s_j = \sum_{i=0}^{k-1} s_{j,i}$
(4) Finally, we compose the secret vector out of the resulting sums: $S = \begin{pmatrix} s_0 & s_1 & s_2 & \cdots & s_{k-1} \end{pmatrix}$

4. Contributions of this Paper and Conclusion

In this paper we introduced a method of reconstructing secret shares in Shamir's system that protects the shares from being revealed if any of the participants have corrupted or fake shares. Additionally, this system allows us to use the same secret shares for future purposes, and secures the system against protocol hijacking attacks.
Further, this method will be used in the AIU technology to add another layer of security to the current infrastructure.

5. Discussion and Future Works

The next step in this path is to prevent the individual shares from being revealed even if all participants are authentic. So far some progress has been made toward this goal by looking for methods of calculating individual columns of the inverse Vandermonde matrix in a multiparty setting.

In [3] Althaus and Leake shown a method of such computation [3] in a finite-field:

Theorem 5.1. *Let $GF(q)$ denote a Galois Field (Finite Field) with $q = p^m$. If n is relatively prime to q and the roots $a_0, a_1, \cdots, a_n$ of $x^{n+1} \equiv x$ are in $GF(q)$ (such as when $n = q - 1$), the inverse of the Vandermonde matrix can be calculated as:*

$$(1)\ V^{-1} = \begin{pmatrix} 1 & a_0 & a_0^2 & \cdots & a_0^{n-1} \\ 1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{n-1} & a_{n-1}^2 & \cdots & a_{n-1}^{n-1} \end{pmatrix}^{-1} = n^{-1} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ a_0^{-1} & a_1^{-1} & \cdots & a_{n-1}^{-1} \\ a_0^{-2} & a_1^{-2} & \cdots & a_{n-1}^{-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_0^{-(n-1)} & a_1^{-(n-1)} & \cdots & a_{n-1}^{-(n-1)} \end{pmatrix}$$

Based on the theorem above and the proof provided in section 3, each participant can multiply his own share with the equivalent column in the inverse Vandermonde matrix, and then all participants perform the Secure Addition protocol to find the final secret. This way, even the x value in each individual's share can be kept secret.

Is it possible to create a cryptosystem in which the encryption and decryption must be performed in a multiparty setting? Such a system could be used to create a network protocol in which the data is useless, unless a sufficient number of trusted users are present. How about a multi party pseudo random number generator?

6. Acknowledgement

This work was made possible by a Summer Undergraduate Research Fellowship from California Institute of Technology and Jet Propulsion Laboratory in Pasadena, California. The author thanks mentor and co-mentor Simon Woo and Edward Chow for their guidances in the duration of this project.

Appendix A. Autonomous Information Unit

The AIU technology [4] was designed by researchers at the Jet Propulsion Laboratory to provide an infrastructure to protect data between multiple units, where units can be servers, computers, or handheld devices. An AIU provides a mechanism to establish the required trust between the AIUs before they engage in reconstructing the information that has been distributed between them.

References

[1] Shamir, Adi; "How to share a secret." Commun. ACM, 22(11):612-613, November 1979.
[2] Cramer, Ronald, Damgård, Ivan, and Nielsen, Jesper Buus; "Secure Multiparty Computation and Secret Sharing – An Information Theoretic Appoach." Book Draft.
[3] Althaus, H. and Leake, R.; "Inverse of a finite-field Vandermonde matrix (Corresp.)" Information Theory, IEEE Transactions on, 15(1):173-173 1969.
[4] Chow, E. T., Woo, S., James, M., and Paloulian, G.; "Autonomous Information Unit for Fine-Grain Data Access Control and Information Protection in a Net-Centric System." California Institute of Technology and NASA's Jet Propulsion Laboratory. April 2012.