Secret Sharing and Cryptography

Benjamin Kuykendall

Columbia Splash

4 November 2017

1 Introduction

1.1 Motivation

In layman's terms, if Alice shares a secret with Bob, both of them know the secret. Because of this, sharing secrets can be pretty dangerous: Bob could share the secret with Charlie, without any input from Alice. Let's try to come up with something better.

Imagine we give half of the secret to Alice and the other half to Bob. This way, Bob cannot tell Charlie the whole secret because he doesn't even know it. But if Alice and Bob jointly decide to reconstruct their secret, they can easily combine their halves. Let's try to analyze this scheme formally. Think of our secret as a binary message of some fixed length

$$m = m_1 m_2 \dots m_t \text{ where each } m_i \text{ is a bit 0 or 1}$$

Now we define two procedures: a Share procedure that takes the secret and splits it into Alice's and Bob's parts, and a $\mathsf{Reconstruct}$ procedure that should recreate the original message using these shares. Here, both procedures are rather trivial.

$$\mathsf{Share}(m) = (m_1 \dots m_{t/2},\ m_{t/2+1} \dots m_t)$$
$$\mathsf{Reconstruct}(a, b) = a_1 \dots a_n b_1 \dots b_n$$

For the scheme to be correct, we simply mean that the reconstructed message is equal to the original secret. Clearly this scheme is correct. But is it secure? For a secret sharing scheme to be secure, we want it to be impossible to "figure out" the secret from only one of the shares. But what exactly does figuring out a secret mean?

Let's think of a concrete case. Say Alice and Bob want to share access to a bank account; the secret will be the username and password. If they are the same length, then our scheme would give Alice the username and Bob the password. This might seem okay at first, but it

would be a problem if we have any prior knowledge about usernames. For example, my bank uses my email address as the username. So if Bob has this outside information, he actually knows the whole secret!

There are easy ways to fix this specific problem: say give each person half of the username and half of the password. But we want to solve this problem more generally. Cryptographic secret sharing will assure us that no matter what outside information one player has, they cannot combine it with their share to compromise the secret.

1.2 Secret Sharing

Here is a simple scheme that does exactly that. First, let us define the XOR operation on bits. This operation means exclusive or, and it comes from considering bits as the values True and False. Then we say a XOR b is true when exactly one of a and b are true. Using bits 1 for True and 0 for False, we can consider XOR as an operation on bits:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Now we are ready to define our scheme.

$\mathsf{Share}(m) = (r_1, \ldots, r_t,\ m_1 \oplus r_i, \ldots, m_t \oplus r_i)$ for random bits r_i

$\mathsf{Reconstruct}(a, b) = a_1 \oplus b_1,\ \ldots,\ a_t \oplus b_t.$

Correctness is easy. Looking at our truth table, we have $b \oplus (a \oplus b) = a$

a	b	$a \oplus b$	$b \oplus (a \oplus b)$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

So each bit of the reconstructed secret is $r_i \oplus (m_i \oplus r_i) = m_i$.

For security, we have the following property: individually, Alice's and Bob's keys are completely random, regardless of the message. This is a foundational idea of information theory, and it should be very intuitive that you cannot figure anything out from a completely random value. To demonstrate this, let's play a game. I am thinking of a number. If I flip a coin, and tell you whether it falls heads or tails, does this help you guess my number?

Now lets do some analysis to show that Alice's and Bob's shares really are random.

For Alice's share, this is obvious: her share is a string of random coin tosses.

For Bob, we can work case by case. Consider the one bit message :

$$\text{If } m = 0 \text{ then } \begin{cases} \text{with probability } 1/2, & r = 0 \text{ so } s = m \oplus r = 0 \oplus 0 = 0 \\ \text{with probability } 1/2, & r = 1 \text{ so } s = m \oplus r = 0 \oplus 1 = 1 \end{cases}$$

$$\text{If } m = 1 \text{ then } \begin{cases} \text{with probability } 1/2, & r = 0 \text{ so } s = m \oplus r = 1 \oplus 0 = 1 \\ \text{with probability } 1/2, & r = 1 \text{ so } s = m \oplus r = 1 \oplus 1 = 0 \end{cases}$$

For longer messages, it works exactly the same way

$$\text{If } m = 00 \text{ then } \begin{cases} \text{with probability } 1/4, & r = 00 \text{ so } s = m \oplus r = 00 \oplus 00 = 00 \\ \text{with probability } 1/4, & r = 01 \text{ so } s = m \oplus r = 00 \oplus 01 = 01 \\ \text{with probability } 1/4, & r = 10 \text{ so } s = m \oplus r = 00 \oplus 10 = 10 \\ \text{with probability } 1/4, & r = 11 \text{ so } s = m \oplus r = 00 \oplus 11 = 11 \end{cases}$$

$\vdots$

1.3 Formal Definition

Let's write the guarantee formally: for any secret m, m', player i, and possible share s_i^*

$$\Pr[\mathsf{Share}(m) \to s_i = s_i^*] = \Pr[\mathsf{Share}(m') \to s_i = s_i^*].$$

Remember this is the correct notion that "without the other share s_i tells us no information about the message".

We can generalize this to more than two players very easily. The idea is we want to share a secret with n people such that any k of them can reconstruct the secret, but any fewer have no information. Define the n-out-of-k secret sharing problem as follows. Now, Share will generate n shares.

$$\mathsf{Share}(m) \to (s_1, \ldots, s_n)$$
$$\mathsf{Reconstruct}(s_{i_1}, \ldots, s_{i_k}) \to x$$

Correctness means that it is always the case that $m = x$.

Security means that for any set of $k-1$ players $i_1, \ldots, i_{k-1}$ and possible shares $s_{i_1}, \ldots, s_{i_k}$ we have the following quantity is independent of m

$$\Pr[\mathsf{Share}(m) \to s_{i_j}^* = s_{i_j} \text{ for } j = 1, 2, \ldots, k-1].$$

To construct a scheme like this, we need to use some new mathematical tools.

2 Algebra

2.1 Modular Arithmetic

We are going to define modular arithmetic. This idea is this: addition and multiplication have a lot of nice properties over the real numbers. But, the real numbers are infinite, which introduces a lot of difficulties in information theory and computer science. Is there a way we can preserve these nice properties over a finite set? It turns out for $0, 1, 2, \ldots, p-1$, where p is a prime number, we can.

For any integer $0 \leq i \leq p-1$ define the remainder of $n \bmod p$ to be the smallest non-negative r such that for some whole number s we have $n = sp + r$. I am sure you have all seen this before: if you are doing long division, and stop at the decimal point, then the remainder is the part left over. For example, the remainder of $25 \bmod 7$ is 4 as $4 + 7 \times 3 = 25$.

To define addition and multiplication $\bmod p$, we do the operations normally, and then take the remainder mod p. These are operations that take integers less than p to other integers less than p.

$$a +_p b = (a + b) \bmod p$$
$$a \times_p b = (a \times b) \bmod p$$

For example in $\mathbb{Z}_5$ we have $1 +_5 1 = 2 \bmod 5 = 2$, $3 +_5 2 = 6 \bmod 5 = 0$, $4 \times_5 3 = 12 \bmod 5 = 2$.

Now let's think about the inverse of these operations. To define subtraction we can do the same thing

$$a -_p b = (a - b) \bmod p$$

Although $a - b$ will be negative, $a -_b b \bmod p$ is always positive. For example, what is $1 - 3 \bmod 5$? We also can verify that modular addition and subtraction are in fact inverses

$$r = (a +_p b) \Rightarrow r + np = a + b$$
$$s = (a +_p b) - b \Rightarrow s + mp = r - b$$

Putting these together

$$s + (m - n)p = a + b - b = a.$$

To define division, note the following definition for real numbers

$$a/b = a \times \frac{1}{b} = ab^{-1}.$$

Then we know b^{-1} is the unique number such that

$$1 = b^{-1}b.$$

It turns out this is the notion of division we want to apply in the modular setting. But how can we be sure that such a number exists? Let's work through an example: write out the times table for $p = 5$.

$\times_p$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Let's look at the 1s in this table. For each number other than 0, there is another number such that their product is 1.

$$1 \times_p 1 = 1 \qquad 3 \times_p 2 = 1 \qquad 4 \times_p 4 = 1$$

These numbers are multiplicative inverses. Now, did we get lucky that each number had a multiplicative inverse? No. It turns out that whenever p is prime, there are inverses like this. We will skip over the proof of this statement.

An aside: a set with addition, multiplication, subtraction, and division that work like this is called a field. There are lots of interesting fields in mathematics, and not all of them look like the integers. For example, the complex numbers are a field. But looking to our secret sharing applications, the finite field of integers mod p (often called $\mathbb{Z}_p$) is exactly what we need.

2.2 Polynomial Interpolation

From here on out, drop the subscripts, and let $a + b$ denote modular addition and so forth.

Like we have polynomials over the real numbers, we can define polynomials over any field using our new definitions of addition and multiplication. A degree d polynomial is

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d$$

where exponentiation is simply repeated multiplication.

Our goal today is to show the following: given a set of points $(x_1, y_1), \ldots (x_k, y_k)$ there is a unique polynomial of degree at most $k - 1$ such that $p(x_i) = y_i$ for all of the i. If you think about this visually, this should match your geometric intuition. It is a plain fact of Euclidean geometry that any two points determine a line. And you might know that any three points determine a parabola. We want to extend this in two ways. First in degree: this should hold

for any k. Second in generality, instead of polynomials in the real numbers, we want this to hold for polynomials over any field.

To prove there is a unique polynomial requires two directions: first that there is at least one polynomial, and then that there is at most one. We will start with the "at most" direction.

2.2.1 Uniqueness

Assume there are two polynomials p and q of degree $k-1$ such that for all i

$$p(x_i) = q(x_i) = y_i$$

Then we can define another degree $k-1$ polynomial

$$r(x) = p(x) - q(x).$$

W know $r(x_i) = y_i - y_i = 0$ for each i. So we can factor out $(x - x_i)$ leaving some R

$$r(x) = (x - x_1)(x - x_2)\dots(x - x_k)R$$

Now distribute again. We get a leading term

$$r(x) = x^k R + \dots$$

However, we know r is a degree $k-1$ polynomial, so it cannot have a x^k term. Thus $R = 0$. Then

$$p(x) - q(x) = r(x) = (x - x_1)(x - x_2)\dots(x - x_k)0 = 0$$

Or in other words $p = q$.

2.2.2 Existence

We will construct a polynomial such that $p(x_i) = y_i$. First define

$$h_{a,b}(x) = \frac{x - a}{b - a}.$$

This is a degree 1 polynomial; it is well-defined when $a \neq b$ since $(b-a)$ will have an inverse. Now note

$$h_{a,b}(a) = \frac{a - a}{b - a} = 0 \qquad h_{a,b}(b) = \frac{b - a}{b - a} = 1$$

So alone this is not very interesting, but what if have many of them?

$$g_1 = \left(\frac{x - x_2}{x_i - x_2}\right)\dots\left(\frac{x - x_k}{x_i - x_k}\right)$$

and g_i similarly: taking the product of the h_{x_i,x_j} for all $j \neq i$. Then we have

$$g_i(x_i) = h_{x_i,x_1}(x_i)h_{x_i,x_2}(x_i)\dots h_{x_i,x_k}(x_i) = 1 \times 1 \times \dots \times 1 = 1$$
$$g_i(x_j) = h_{x_i,x_1}(x_j)h_{x_i,x_2}(x_j)\dots h_{x_i,x_j}(x_j)\dots h_{x_i,x_k}(x_j) = 1 \times 1 \times \dots \times 0 \times \dots \times 1 = 0$$

So we have g_i is 1 on x_i and 0 on the remaining x_j. Now define

$$p(x) = y_1 g_1(x) + y_2 g_2(x) + \dots + y_k g_k(x)$$

We have

$$p(x_i) = y_1 0 + y_2 0 + \dots + y_i 1 + \dots + y_k 0 = y_i$$

So p is the polynomial we wanted!

2.2.3 Example

Let's find the quadratic such that

i	x_i	y_i
1	1	1
2	2	4
3	3	3

I think you all already know the solution to this, but let's verify that our method spits out the correct answer as well. Start with the g

$$g_1 = \left(\frac{x-x_2}{x_1-x_2}\right)\left(\frac{x-x_3}{x_1-x_3}\right) = \left(\frac{x-2}{1-2}\right)\left(\frac{x-3}{1-3}\right)$$
$$g_2 = \left(\frac{x-x_1}{x_2-x_1}\right)\left(\frac{x-x_3}{x_2-x_3}\right) = \left(\frac{x-1}{2-1}\right)\left(\frac{x-3}{2-3}\right)$$
$$g_3 = \left(\frac{x-x_1}{x_3-x_1}\right)\left(\frac{x-x_2}{x_3-x_2}\right) = \left(\frac{x-1}{3-1}\right)\left(\frac{x-2}{3-2}\right)$$

Then

$$p = g_1 y_1 + g_2 y_2 + g_3 y_3$$

$$= \left(\frac{x-2}{1-2}\right)\left(\frac{x-3}{1-3}\right)1 + \left(\frac{x-1}{2-1}\right)\left(\frac{x-3}{2-3}\right)4 + \left(\frac{x-1}{3-1}\right)\left(\frac{x-2}{3-2}\right)9$$

$$= \frac{x^2-5x+6}{-1\times-2} - \frac{4(x^2-4x+3)}{1\times-1} + \frac{9(x^2-3x+2)}{2\times1}$$

$$= \frac{1}{2}\Big[(x^2-5x+6) - 8(x^2-4x+3) + 9(x^2-3x+2)\Big]$$

$$= \frac{1}{2}\Big[(1+9-8)x^2 + (-5-32-27)x + (6-24+18)\Big] = x^2.$$

And as we already alluded to, this same process works fine mod p.

3 Shamir's Secret Sharing

After all of this, we can finally get back to secret sharing. Remember, out goal is to create n-out-of-k secret sharing, so that any any n shares can determine the secret. This should sound similar to the polynomial problem we just solved above. Here is the construction. Let our secret be an element $m \in \mathbb{Z}_p$ for some $p > k$. Then

$\mathsf{Share}(m)$:

Choose random $a_1,\ a_2,\ \ldots, a_{n-1}$ uniformly from $\mathbb{Z}_p$.
Let $p(x) = m + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$.
Output shares $(p(1),\ p(2),\ \ldots, p(k))$.

$\mathsf{Reconstruct}(s_{i_1}, s_{i_2}, \ldots s_{i_n})$:

Reconstruct the polynomial such that $q(s_{i-1}) = i$.
Output $q(0)$.

Correctness follows by our theorem on polynomials: as the degree of the polynomial is $n-1$ and we have n points, we know that q is unique. Thus we have $p = q$ so $q(0) = p(0) = m$.

Security follows from the same idea. Compute

$$\Pr[\mathsf{Share}(m) \to (s_1, \ldots, s_n) \ : \ s^*_{i_1} = s_{i_1} \ldots s^*_{i_{n-1}} = s_{i_{n-1}}]$$

By definition this is just

$$\Pr[p \ : \ p(i_1) = s_{i_1} \ldots p(i_{n-1}) = s_{i_{n-1}}].$$

This is just the number of polynomials such that $p(0) = m, p(i_1) = s_{i_1}, \ldots, p(i_{n-1}) = s_{i_{n-1}}$ divided by the total number of choices for p. The denominator is easy. We choose $n-1$ coefficients a_i so

$$\#\text{ choices for } p = \underbrace{p \times p \times \cdots \times p}_{n-1 \text{ times}} \text{ is } p^{n-1}.$$

What about polynomials that agree on the $n-1$ shares? We can count them by adding another point, say $k+1$. There is exactly 1 one polynomial that agrees on

$$q(0) = m,\ p(i_1) = s_{i_1},\ \ldots,\ p(i_{n-1}) = s_{i_{n-1}},\ p(k+1) = y.$$

Then there are p choices for y. And since each polynomial that agrees with the first n points must have some value in $k+1$, these are all of them! So

$$\#\text{ choices for } p \text{ such that } p(0) = m, p(i_1) = s_{i_1}, \ldots, p(i_{n-1}) = s_{i_{n-1}} \text{ is } p.$$

Conclude $\Pr = p/p^{n-1} = 1/p^{n-2}$. Because we never used anything specific to these values of i or s, we know this holds for any s, proving the security of our scheme.

4 General Secret Sharing

Another type of secret sharing we can consider considers more complicated access structures. Instead of two (or k) people equally sharing a bank account as we did before, think about a corporation's bank account. To take money out, a majority of the 10 board members along with either the president or vice president of the company must agree. How could we implement a scheme like this using our existing n-out-of-k secret sharing?

4.1 General Definition

Let an access structure by a predicate $\mathsf{Authorized}(i_1, \ldots, i_m)$ that tells whether or not these people are allowed to reconstruct the secret. Then define correctness as "any authorized set of shares can reconstruct the secret" and security as "the shares of any unauthorized set of people are distributed the same for every message".

4.2 Two Level Access Structure

Let us solve the secret sharing problem for our specific example of a "two level" access structure. The trick is to compose different shared secrets using our existing n-out-of-k algorithm.

First, generate shares s_1, s_2 of m using 1-out-of-2 secret sharing.

Do 6-out-of-10 secret sharing of s_1, giving a share to each board member.

Do 1-out-of-2 secret sharing of s_2, giving a share to the president and vice president.