**Applications of Polynomial Interpolations**

**Abstract:**

**Interpolation** is a type ofEstimation, a method of constructing a newdata points within the range of adiscrete set of known data points. Apolynomial is a mathematical expression comprising a sum of terms, each term including a variable or variables raised to a power and multiplied by aCoefficient.

Types of Interpolation:

1. Piecewise Constant Interpolation
2. Linear Interpolation
3. Polynomial Interpolation
4. Spline Interpolation

**Polynomial interpolation** is theinterpolation of a givendata set by the polynomial of lowest possible degree that passes through the points of the dataset.

When graphical data contains a gap, but data is available on either side of the gap or at a few specific points within the gap, an estimate of values within the gap can be made by interpolation.

**Basic Concept behind Polynomial Interpolation:** If a set of data contains *n* known points, then there exists exactly one polynomial of degree *n*-1 or smaller that passes through all those points. The interpolation error is proportional to the distance between the data points to the power *n*. Furthermore, the interpolant is a polynomial and thus infinitely differentiable. So, we see that polynomial interpolation overcomes most of the problems of linear interpolation.

Now, we are going to see the Applications of Polynomial Interpolations where Linear Algebra is heavily used.

**Applications:**

Polynomials can be used to approximate complicated curves, for example, the shapes of letters in**Typography** given a few points. A relevant application is the evaluation of the**natural logarithm** and**trigonometric functions**: pick a few known data points, create alookup table and interpolate between those data points. This results in significantly faster computation. Polynomial interpolation also forms the basis for algorithms in **numerical quadrature** and **numerical ordinary differential equations** and **Secure Multi Party Computation**, **Secret Sharing** schemes.

Polynomial interpolation is also essential to perform sub-quadratic multiplication and squaring such as **Karatsuba multiplication a**nd **Toom–Cook multiplication**, where an interpolation through points on a polynomial which defines the product yields the product itself. For example, given *a* = *f*(*x*) = *a*0*x*0 + *a*1*x*1 + ... and *b* = *g*(*x*) = *b*0*x*0 + *b*1*x*1 + ..., the product *ab* is equivalent to *W*(*x*) = *f*(*x*)*g*(*x*). Finding points along *W*(*x*) by substituting *x* for small values in *f*(*x*) and *g*(*x*) yields points on the curve. Interpolation based on those points will yield the terms of *W*(*x*) and subsequently the product *ab*. In the case of Karatsuba multiplication this technique is substantially faster than quadratic multiplication, even for modest-sized inputs. This is especially true when implemented in parallel hardware. Interpolation Problems can arise in Heat Transfer Estimation, approximation of temperature points where there’s no data reading and when finding derivatives from experimental values.

**Shamir’s Secret Sharing Scheme:**

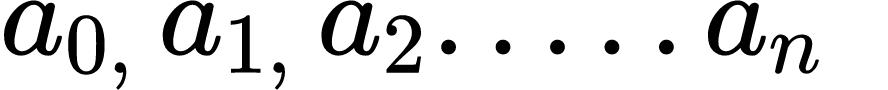
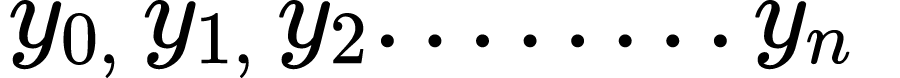
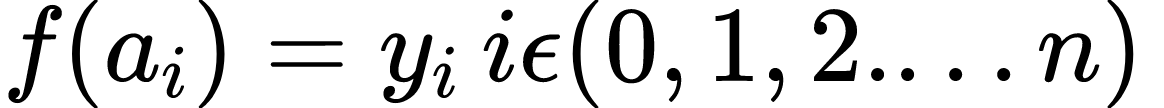
**Introduction:**

Secret sharing is a technique for protecting sensitive data, such as cryptographic keys. It is used to distribute a secret value to several parts or shares that must be combined to access the original value. These shares can then be given to separate parties that protect them using standard means, e.g., memorizing, stored on a computer or in a safe.

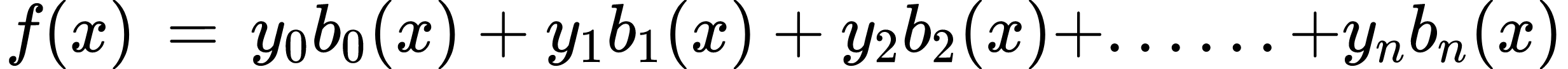
Secret sharing is used in modern cryptography to lower the risks associated with compromised data. Sharing a secret spreads the risk of compromising value across several parties. Standard security assumptions of secret sharing schemes state that if an adversary gains access to any number of shares lower than some defined threshold, it gains no information of the secret value. The first secret sharing schemes were proposed by Shamir and Blakley. This work gives the standard definition of a **(k, n) threshold secret sharing scheme** and its properties. We continue by exploring polynomial evaluations (Lagrange Interpolation Technique) as the mathematical background for Shamir’s scheme. After describing Shamir’s scheme, we prove its security and present algorithms for performing operations with shares.

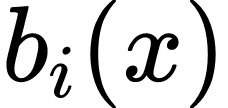
Before diving into the Concept of Secret Sharing Scheme, we need to understand the **Lagrange Interpolation Scheme.**

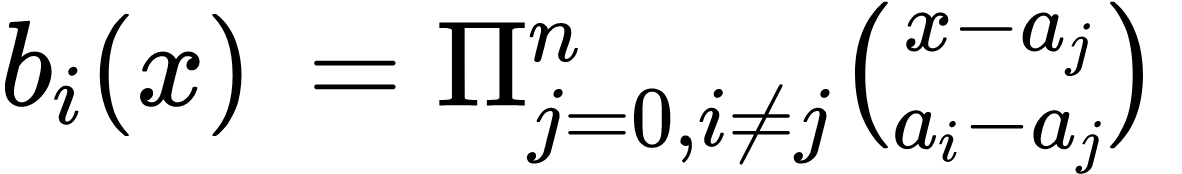
**Lagrange Interpolation Theorem:**

Let R be a field and  and  ∈ R so that all values  are distinct. Then there exists only one polynomial f over R so that degree of f ≤ n and 

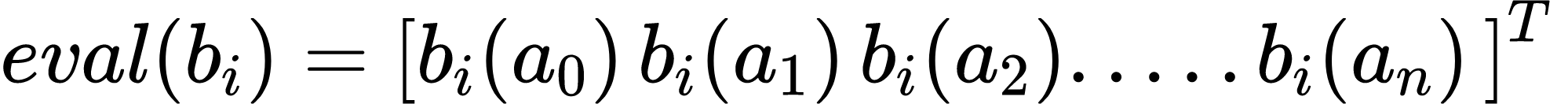
The Lagrange interpolation polynomial can be computed as the sum



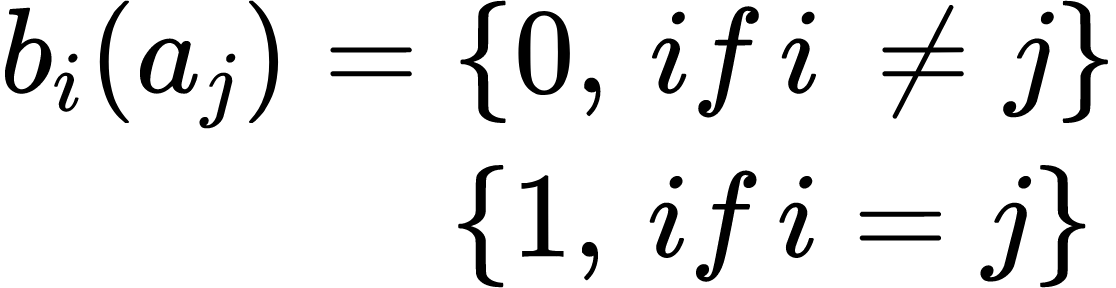
where the base polynomials  is defined as



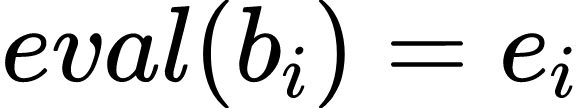
As one could expect, the Lagrange interpolation polynomial has a useful property, its base polynomials correspond to our vectors :



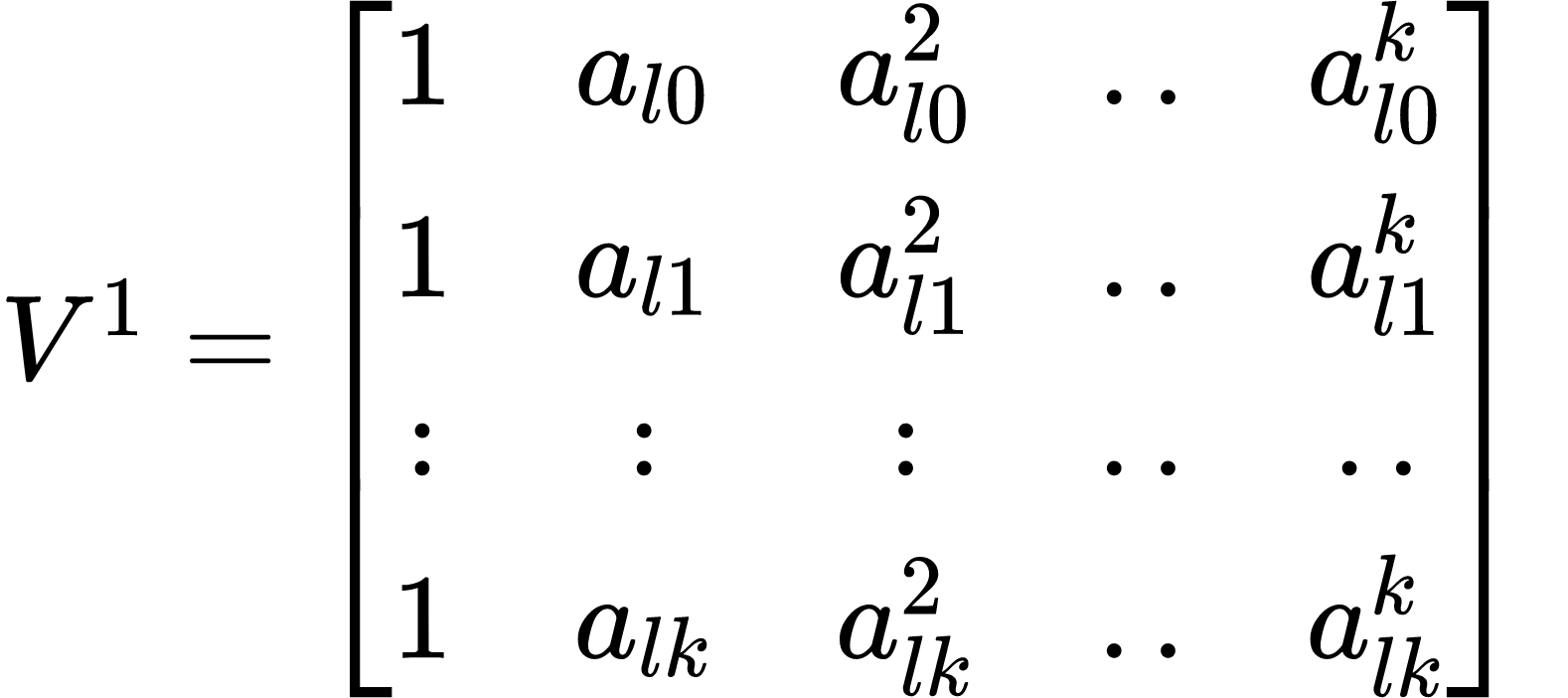
Since,

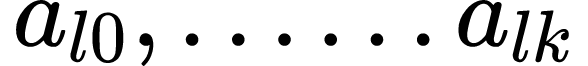
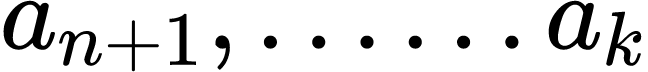
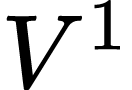


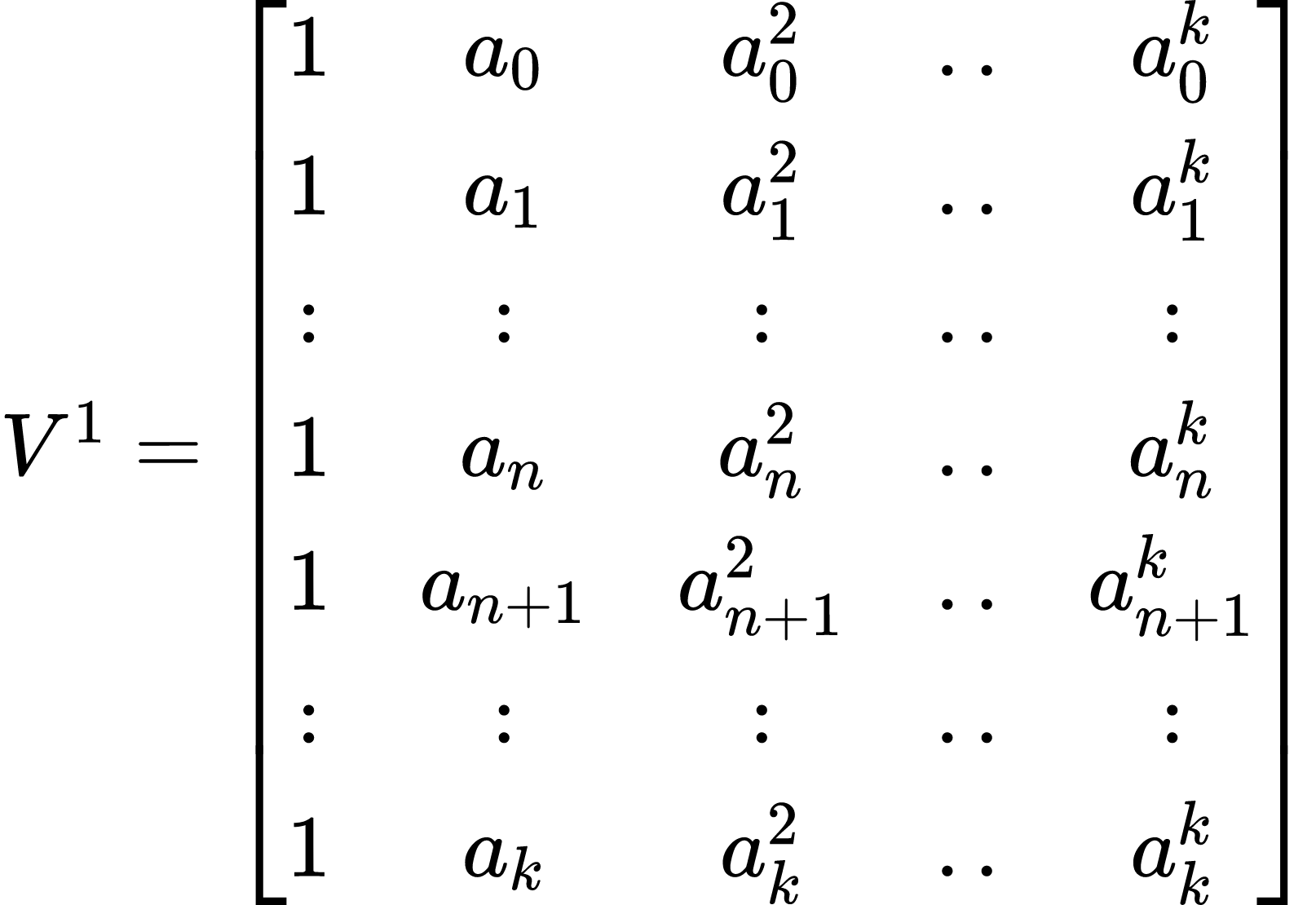
We see that,

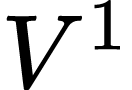
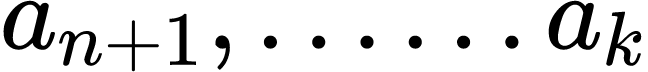


We also must handle the cases where k 6 = n. We will reduce these cases to the (k + 1) × (k + 1) case observed before. First, we consider the case where n > k. If we choose k + 1 different values l 0, . . ., l k ∈ {0, . . ., n}, then we obtain a virtual matrix V 0 by choosing rows l 0, . . ., l k from the original matrix V:

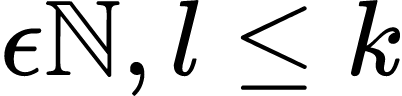
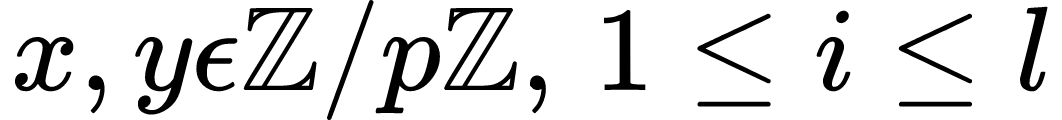
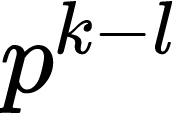
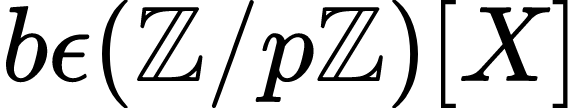
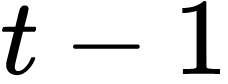
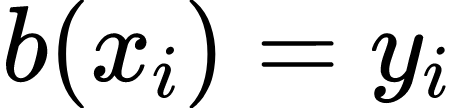
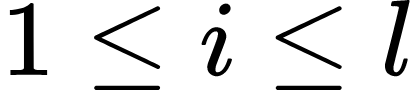


The square matrix V 0 is invertible as its determinant is nonzero, because it corresponds to the evaluation map at [ ] and by showing that we have reached the previously observed and proved case. In the third case when n < k we generate k − n values  so that all values  are distinct. We use these new positions to add rows to the matrix V and get the virtual matrix 

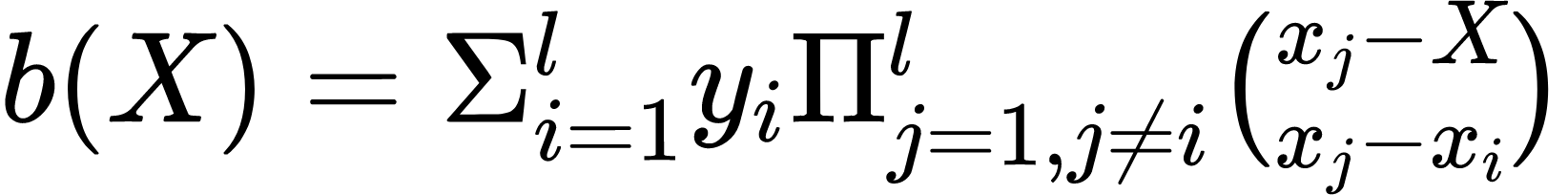


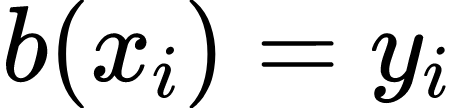
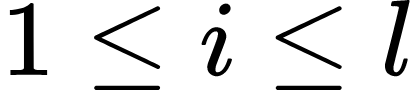
This matrix is an invertible (k + 1) × (k + 1) square matrix that can replace V in the first observed case. Note that if n < k the reconstruction of the polynomial is not unique and is determined by the choice of values  . This gives us a guarantee that it is not possible to uniquely reconstruct the polynomial if there are not enough pairs of positions and evaluations available. Here, we will use this property to prove the privacy of the following secret sharing scheme.

**An Important Lemma**

Let {"id":"g44hsj2ry381605884450630","code":"$n$","font":{"size":12,"family":"Times New Roman","color":"#000000"},"type":"$","ts":1605884450630,"cs":"ff495e8867a1c6ec9f9a8d14b","size":{"width":7.699999999999999,"height":5.6}}, {"id":"eumi36tvr5u1605884199898","code":"$k$","font":{"size":12,"family":"Times New Roman","color":"#000000"},"type":"$","ts":1605884199899,"cs":"3ca1fd64eaacc84991b9e83aa","size":{"width":6.3,"height":9.1}}, Also, let Where are pair-wise distinct. Then there are exactly polynomials of degree With ,

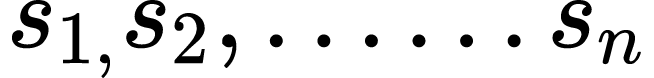
The Lagrange Interpolation Formula yields the polynomial

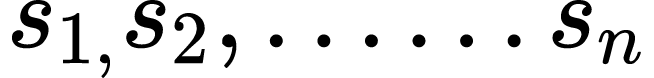
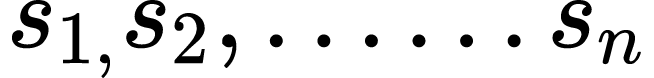


It satisfies ,

This shows that at least one such polynomial exists which satisfies all the l=k values.

**Concept of Secret Sharing:**

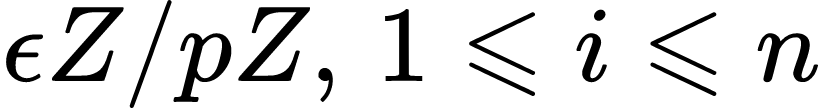
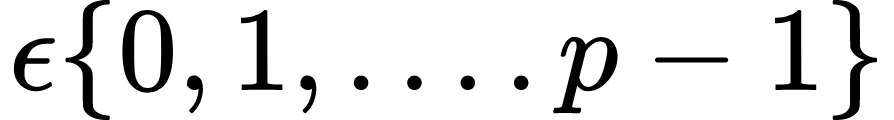
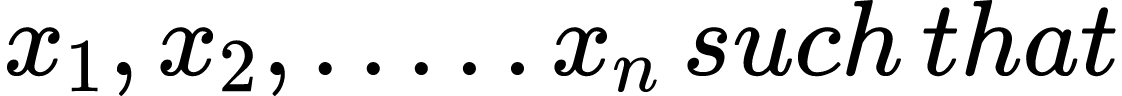
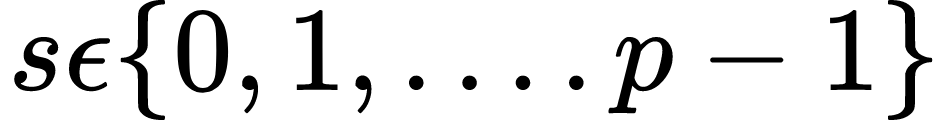
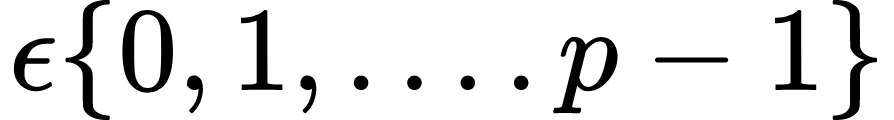
Let the secret data be a value s. An algorithm S defines a k out of n threshold secret sharing scheme, if it computes S(s) = [] and the following conditions hold:

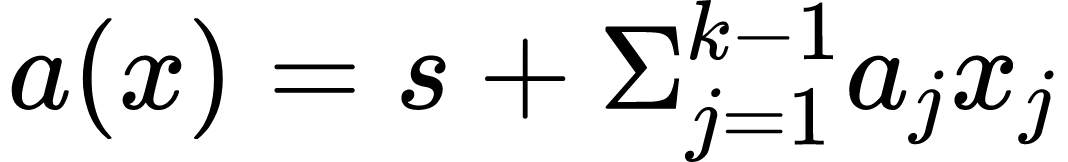
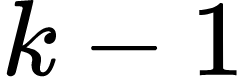
1. **Correctness:** s is uniquely determined by any k shares from [] and there exists an algorithm that efficiently computes s from k shares.
2. **Privacy:** Having access from any k-1 shares from {} should give no information about the value s i.e., the probability distribution of k-1 shares is independent of s.

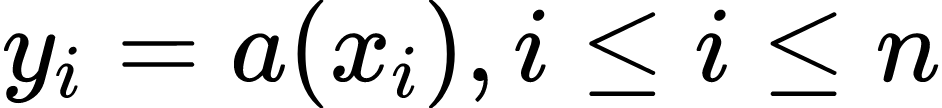
We now describe Shamir’s secret sharing scheme that is based on polynomial evaluations. We start by explaining the infrastructure of secret sharing. The central party is the dealer that performs share computation operations on input secrets and distributes the resulting shares to other parties. When the secret must be reconstructed, the parties give their shares to the dealer, that can then combine the shares and retrieve the secret.

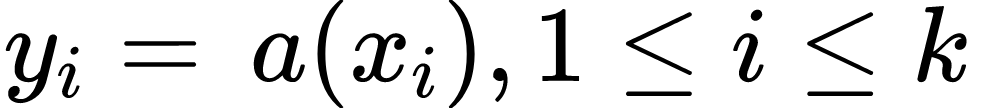
In Shamir’s scheme shares are evaluations of a randomly generated polynomial. The polynomial f is generated in such a way that the evaluation f (0) reveals the secret value. If there are enough evaluations, the parties can reconstruct the polynomial and compute the secret. Algorithm 1 describes how shares are computed in Shamir’s scheme.

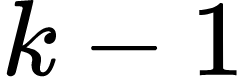
**Algorithm:**

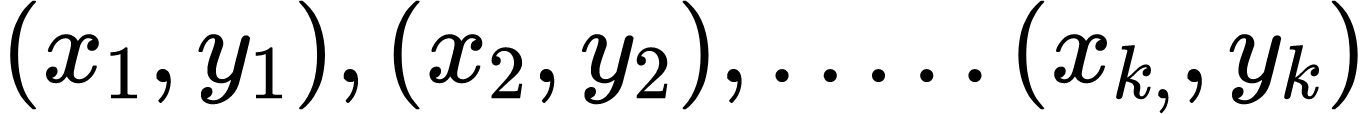
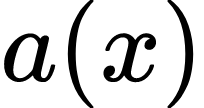
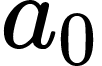
1. The dealer chooses
   * A prime number p, p>=n+1
   * Non-zero elements,  which are pair-wise distinct, i.e., 
2. The elements are the least non-negative elements in the residue class.
3. The dealer publishes these values of 
4. Secret s: 
5. Dealer chooses secretly  and constructs a polynomial

 and degree 

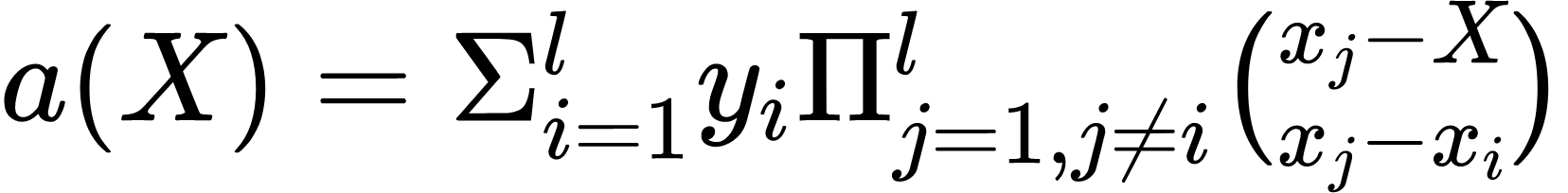
1. Dealer computes shares by 

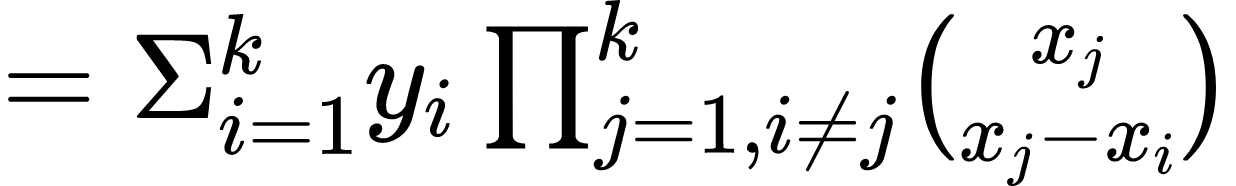
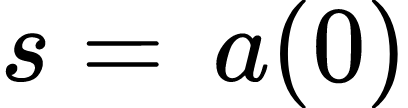
Suppose that k shareholders collaborate. We assume that shares are numbered such that  with the polynomial chosen by the dealer.

Now, these collaborating parties use Lagrange’s Interpolation to determine the polynomial. Dealer sets k=l, and thus there is exactly one such polynomial of degree.

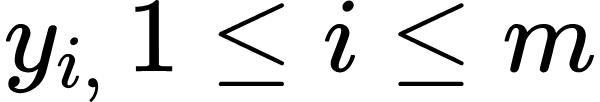
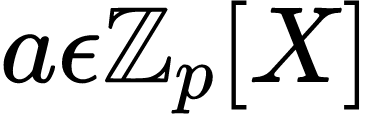
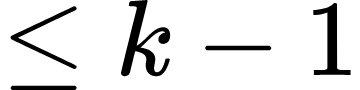
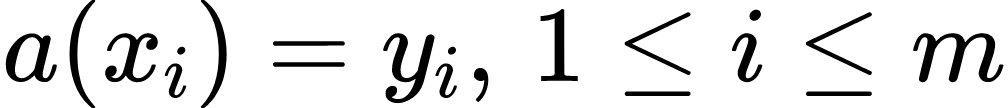
Collaborating parties contains values and by using Lagrange Interpolation Technique, we obtain  Where is the secret of the polynomial.

Thus,

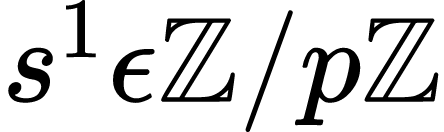
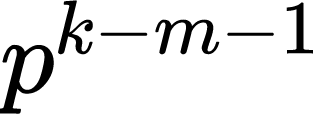
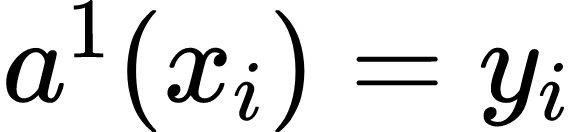
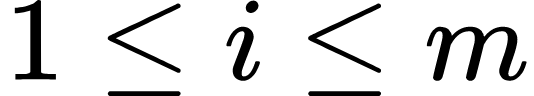


The Secret is 

**Security Analysis**

Suppose that m shareholders want to reconstruct the secret where m<k. Assume that the shares are . The shareholders know that the secret is constant term a(0) of a polynomial of degreethat satisfies .

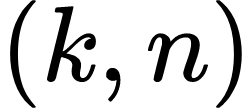
However, we have the following results from Lemma stated above:

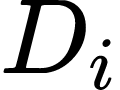
For any , there are exactly polynomials {"id":"xbairfozkrf1605885342433","code":"$,\\,$","font":{"size":12,"family":"Arial"},"type":"$","ts":1605885342433,"cs":"19bb049aceb79149928061063","size":{"width":8.399999999999999,"height":4.199999999999999}}

The assertion follows by using that l=m+1.

This shows that with m<k shareholders all the values of the secret are possible to obtain using Lagrange Interpolation Technique.

**Properties:**

Some of the useful properties of Shamir's threshold Scheme are:

1. **Minimal**: The size of each piece does not exceed the size of the original data.
2. **Extensible**: When {"id":"29h94bfxf3x1605885752838","code":"$k$","font":{"size":12,"family":"Times New Roman","color":"#000000"},"type":"$","ts":1605885752839,"cs":"36d5394760ea31e510f01a5e0","size":{"width":6.3,"height":9.1}}is kept fixed, pieces from secret D can be added or deleted without affecting the other pieces of share.
3. **Dynamic**: Security can be easily enhanced without changing the secret, but by changing the polynomial occasionally (keeping the same free term) and constructing new shares to the participants.
4. **Flexible**: In organizations where hierarchy is important, we can supply each participant with a different number of pieces according to their importance inside the organization. For instance, the president can unlock the safe alone, whereas 3 secretaries are required together to unlock it.