CSA E0 312: Secure Computation	26th August 2015
[Lecture 5]	
Instructor: Arpita Patra	*Submitted by: Marilyn George*

1 Recap: Security Definition for MPC

In the last few lectures, we have seen several aspects of MPC like the models of computation, networks, distrust and adversary. Next, we have seen the security definition of MPC in Real World / Ideal World based paradigm. We have seen two definitions in the same paradigm:

- Indistinguishability of the adversary's view in the real and ideal worlds.
- Indistinguishability of the joint distribution of the output of the honest parties and the view of adversary in the real and ideal worlds.

Today we will see the second definition is stronger and is required to prove security of MPC protocols for any function (randomized /deterministic). Whereas for the MPC protocols for deterministic functions, the first definition is enough. Below we recall the real world /ideal world based security definition. We assume the adversary $\mathcal{A}$ is semi-honest.

1.1 Ideal World MPC

We model the ideal world MPC, as seen earlier, with the help of a Trusted Third Party (TTP). We assume there are n parties and the adversary $\mathcal{A}$ corrupts t out of n parties:

1. The setup consists of a TTP and n parties, $P_1, \ldots, P_n$.
2. Each party P_i sends its input x_i to the TTP.
3. The TTP performs the computation $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$, and sends the output y_i to the respective party P_i.

Since all the communication in the ideal world happens over secure channels, the above protocol satisfies our intuition of security that no party receives any other information other than his/her own input and the output of the computation. Additionally, since the TTP performs the computation, the output is correct.

1.2 Real World MPC

In the real world, the parties communicate among themselves according to a proposed interactive protocol. The protocol is such that it computes the expected output $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$. We prove security of the proposed protocol executed in the real world by proving that the view of the adversary in the real world is indistinguishable from its view in the ideal world, where security is clearly satisfied.

1.3 Comparison of Real World with Ideal World

On fixing the inputs of all the parties, say $x_1, x_2, .., x_n$ we argue that the real world view of the adversary contains no more information than his ideal world view as follows:

- The view of P_i on the given input in the ideal world is denoted by $\mathsf{View}_i^{\mathsf{Ideal}}(x_1, \ldots, x_n)$. Hence the adversary's view is denoted as $\{\mathsf{View}_i^{\mathsf{Ideal}}(x_1, \ldots, x_n)\}_{P_i \in C}$, where C is the set of all the corrupt parties. This view, consisting of $\{x_i, y_i\}_{P_i \in C}$ is referred to as the 'Allowed Values', since the values that $\mathcal{A}$ knows in the ideal world are allowed to be leaked.
- Similarly, the view of the adversary in the real world is denoted by $\{\mathsf{View}_i^{\mathsf{Real}}(x_1, \ldots, x_n)\}_{P_i \in C}$, and is referred to as the 'Leaked Values'. The leaked values in the above example consist of $\{x_i, y_i, r_i, t_i\}_{P_i \in C}$ where r_i denotes the randomness used by party P_i and t_i denotes the protocol transcript seen by P_i.
- We say that our protocol is secure if the leaked values contain *no more information* than the allowed values. This is held to be true if the leaked values can be *efficiently* computed from the allowed values.

We go about proving this using an algorithm called the Simulator which we will denote by SIM. Given only the allowed values i.e. in the ideal world, SIM is expected to interact with the adversary to simulate the adversary's real world view. Hence, SIM will interact with the adversary on behalf of all the honest parties according to the real world protocol. If it can simulate a view close enough to the real world such that the adversary $\mathcal{A}$ cannot distinguish between the real and ideal worlds, this implies that the leaked values can be *efficiently* computed from the allowed values. Then the protocol is considered secure. SIM is sometimes referred to as the *ideal adversary* as it receives the inputs and outputs of the adversary (i.e. the allowed values) in the ideal world.

1.4 Definition 1: Indistinguishability of the Adversary's View

With respect to this definition, we require that the adversary's view in the real world $\{\mathsf{View}_i^{\mathsf{Real}}(x_1, \ldots, x_n)\}_{P_i \in C}$, be indistinguishable from his view in the ideal world, $\{\mathsf{View}_i^{\mathsf{Ideal}}(x_1, \ldots, x_n)\}_{P_i \in C}$. We note that $\{\mathsf{View}_i^{\mathsf{Ideal}}(x_1, \ldots, x_n)\}_{P_i \in C}$ is the view created with the help of SIM, which interacts with the adversary on behalf of all the honest parties and simulates the real world protocol. We can define security as:

$$\{\mathsf{View}_i^{\mathsf{Real}}(x_1, \ldots, x_n)\}_{P_i \in C} \approx \{\mathsf{View}_i^{\mathsf{Ideal}}(x_1, \ldots, x_n)\}_{P_i \in C}$$

Both $\{\mathsf{View}_i^{\mathsf{Ideal}}(x_1, \ldots, x_n)\}_{P_i \in C}$ and $\{\mathsf{View}_i^{\mathsf{Real}}(x_1, \ldots, x_n)\}_{P_i \in C}$ are random variables, so the above statement refers to the indistinguishability of the distributions.$\{\mathsf{View}_i^{\mathsf{Ideal}}(x_1, \ldots, x_n)\}_{P_i \in C}$ is a random variable over the random coins of SIM and the adversary[1], and $\{\mathsf{View}_i^{\mathsf{Real}}(x_1, \ldots, x_n)\}_{P_i \in C}$

[1]When the adversary is semi-honest we can assume that the simulator initiates the adversary with the inputs of the corrupted parties, fixes the adversary's randomness and simulates the adversary's view in the real world. The adversary therefore is the part of the simulator and the simulation is nothing but a mental game. This is fine as a semi-honest $\mathcal{A}$ always follows protocol steps and participates in the protocol with the inputs it is given. An malicious adversary $\mathcal{A}$ on the other hand cannot be thought of as a part of the simulator. This is because the malicious adversary behaves arbitrarily and it must be treated as an entity outside the simulator.

is a random variable over the random coins of all the participating parties in the real world protocol.

1.5 Definition 2: Indistinguishability of joint distributions of Output and View

In this definition, we require that the joint distribution of the outputs of the honest parties and the adversary's view in the real world be indistinguishable from the corresponding distribution in the ideal world. This definition is stronger than Definition 1, and it accounts for randomized functions, as we see in the next section. We formalize the definition with the help of additional notation for the output of the honest parties in the real world - $\{\mathsf{Output}_i^{\mathsf{Real}}\}_{P_i \in H}$, and the ideal world - $\{\mathsf{Output}_i^{\mathsf{Ideal}}\}_{P_i \in H}$; where H is the set of all honest parties. The security notion is then defined as:

$$\left[\{\mathsf{View}_i^{\mathsf{Ideal}}\}_{P_i \in C}, \{\mathsf{Output}_i^{\mathsf{Ideal}}\}_{P_i \in H}\right] \approx \left[\{\mathsf{View}_i^{\mathsf{Real}}\}_{P_i \in C}, \{\mathsf{Output}_i^{\mathsf{Real}}\}_{P_i \in H}\right]$$

2 Definition 1 vs. Definition 2

Our intuition is that Definition 2 is required for randomized functions, since both the outputs of the honest parties and the view of the adversary are functions of the randomness picked by the participants in the protocol. Note that the outputs of the honest parties are also a random variable and induces a probability distribution over the randomness used by the parties since the considered function is a randomized function. For a deterministic function, the outputs of the honest parties are determined from the inputs of the parties and is independent of the randomness used by the parties in the protocol. Hence, we infer that the view distribution of the adversary and the output distribution of the honest parties are correlated for randomized functions. We now see an example, a protocol to evaluate a randomized function that is clearly insecure. But we will be able to prove the protocol secure according to Definition 1 but it cannot be proven secure according to Definition 2.

2.1 Randomized Function and Definition 1

Consider a two party computation between P_1 and P_2 of the the simple randomized function, $f(\cdot, \cdot) = (r, \cdot)$, where r is a random bit, i.e. the function computes a random bit, and requires no input from either party. r is the output of the computation to P_1, and P_2 does not get anything.

- In the ideal world, the TTP generates a random bit r and sends it to P_1.
- In the real world protocol, P_1 samples a random bit r, outputs it and sends it to P_2.

Now assume that P_2 is the corrupted party. Let us design a simulator for this case to prove security against an adversary $\mathcal{A}$ that has corrupted P_2. Our simulator works as follows: SIM, on being placed in the ideal world, samples a random bit r', and sends it to P_2, in order to simulate the real world protocol.

$$\{\mathsf{View}_i^{\mathsf{Ideal}}\}_{P_i \in C} \approx \{\mathsf{View}_i^{\mathsf{Real}}\}_{P_i \in C}$$
$$\{r' : r' \text{ is random}\} \approx \{r : r \text{ is random}\}$$

It is easy to see that when P_2 is corrupted, then there is a simulator that can create an ideal view of the adversary that is perfectly indistinguishable from the adversary's real world view. When P_1 is corrupted, it is very easy to prove security. So with regard to Definition 1, the above protocol is secure as P_2 receives a randomly sampled bit in both cases, and his views are indistinguishable.

However, we can see trivially that the protocol is insecure, since P_2 receives the bit computed by P_1, but it receives no information in the ideal world. Now we will see that the insecure protocol cannot be proved secure according to Definition 2.

2.2 Randomized Function and Definition 2

Consider the joint distribution of the output of the honest party and the view of the adversary both in real and ideal world:

$$\Big[\{\mathsf{View}_i^{\mathsf{Ideal}}\}_{P_i\in C}, \{\mathsf{Output}_i^{\mathsf{Ideal}}\}_{P_i\in H}\Big] \quad \Big[\{\mathsf{View}_i^{\mathsf{Real}}\}_{P_i\in C}, \{\mathsf{Output}_i^{\mathsf{Real}}\}_{P_i\in H}\Big]$$
$$\Big[(r', r) : r, r' \text{ independent and random}\Big] \not\approx \Big[(r, r) : r \text{ is random}\Big]$$

Now, it is easy to see that the protocol is insecure according to Definition 2. However in the next section, we see that Definition 1 is indeed sufficient to model security, provided we modify our view of randomized functions.

2.3 Definition 1 is sufficient

For deterministic functions, the output is fixed once the inputs are fixed. In this case then the distributions of output of honest parties and view of the adversary are not correlated, and they can be considered separately. In fact the output can only be one element, and there is no distribution of outputs.

$$\Big[\{\mathsf{View}_i^{\mathsf{Ideal}}\}_{P_i\in C}, \{\mathsf{Output}_i^{\mathsf{Ideal}}\}_{P_i\in H}\Big]_{x_1,\ldots,x_n,k} \approx \Big[\{\mathsf{View}_i^{\mathsf{Real}}\}_{P_i\in C}, \{\mathsf{Output}_i^{\mathsf{Real}}\}_{P_i\in H}\Big]_{x_1,\ldots,x_n,k}$$
$$\Big[\{\mathsf{View}_i^{\mathsf{Ideal}}\}_{P_i\in C}\Big]_{x_1,\ldots,x_n,k} \approx \Big[\{\mathsf{View}_i^{\mathsf{Real}}\}_{P_i\in C}\Big]_{x_1,\ldots,x_n,k}$$

Hence both the above are equivalent, where k is the security parameter. When considering randomized functions, we can model them as deterministic functions where each party inputs some randomness, in addition to their actual inputs. For example, we could compute $g(x_1, x_2; r)$ as $f((x_1, r_1), (x_2, r_2))$ where the randomness $r = r_1 + r_2$. Then we know that Definition 1 will suffice to model all functions.

3 Making Indistinguishability Precise

To make the distinguishability in Definition 1 precise, we first introduce the following standard definitions. k will be denoted as the security parameter. It is a natural number. The set of natural numbers are denoted as $\mathbb{N}$

Definition 1[Negligible Function] A function $\mu(\cdot)$ is negligible, if for every polynomial $p(\cdot)$, there exists an N such that for all $k > N$, $\mu(k) < \frac{1}{p(k)}$. $\diamondsuit$

Definition 2[Probability Ensemble] An infinite series $X = \{X(a,k)\}$, indexed by a string a and a natural number k, such that each $X(a,k)$ is a random variable. $\diamondsuit$

Given a security parameter $k \in \mathbb{N}$, we wish that security should hold for inputs of all lengths as long as k is large enough. Then in our scenario, we define the following probability ensembles, taking $a = (x_1, x_2, .., x_n)$ and k as the security parameter:

- $X(x_1, x_2, ..., x_n, k) = \left\{\{\mathsf{View}_i^{\mathsf{Real}}\}_{P_i \in C}\right\}_{(x_1,x_2,..,x_n,k)}$ in the probability space defined by the randomness of all the participating parties.
- $Y(x_1, x_2, ..., x_n, k) = \left\{\{\mathsf{View}_i^{\mathsf{Ideal}}\}_{P_i \in C}\right\}_{(x_1,x_2,..,x_n,k)}$ in the probability space defined by the randomness of SIM.

We have now the following flavours of indistinguishability with respect to the above probability ensembles.

Definition 3[Computational Indistinguishability of $X = \{X(a,k)\}$ and $Y = \{Y(a,k)\}$] We say that two ensembles X and Y are computational Indistinguishable denoted as $X = \{X(a,k)\} \approx_c Y = \{Y(a,k)\}$ if for every *polynomial-time* distinguisher D there exists a negligible function $\mu(\cdot)$ such that for every a and all large enough values of k:

$$\left|\mathsf{Pr}[D(X(a,k) = 1] - \mathsf{Pr}[D(Y(a,k) = 1]\right| < \mu(k)$$

$\diamondsuit$

The distinguisher D in our case is the real adversary, and the above definition implies he cannot distinguish between the two distributions (views) with more than negligible probability. This can also be written as - If the probability of D guessing the correct distribution in the distinguishing game is denoted by $Adv_D(X,Y)$, then:

$$\left|Adv_D(X,Y)\right| < \frac{1}{2} + \mu(k)$$

i.e the adversary can distinguish with a probability that is at most negligibly better than randomly guessing one of the two distributions. We now discuss statistical indistinguishability.

Definition 4[Statistical Indistinguishability of $X = \{X(a,k)\}$ and $Y = \{Y(a,k)\}$] We say that two ensembles X and Y are computational Indistinguishable denoted as $X = \{X(a,k)\} \approx_s Y = \{Y(a,k)\}$ if for *every* distinguisher D there exists a negligible function $\mu(\cdot)$ such that for every a and all large enough values of k:

$$\left|\mathsf{Pr}[D(X(a,k) = 1] - \mathsf{Pr}[D(Y(a,k) = 1]\right| < \mu(k)$$

$\diamondsuit$

Similarly this can be written as - If the probability of D guessing the correct distribution in the distinguishing game is denoted by $Adv_D(X,Y)$, then:

$$\left|Adv_D(X,Y)\right| < \frac{1}{2} + \mu(k)$$

The difference is that D in this case could be an unbounded powerful adversary. We now discuss perfect indistinguishability.

Definition 5[Perfect Indistinguishability of $X = \{X(a,k)\}$ and $Y = \{Y(a,k)\}$] We say that two ensembles X and Y are computational Indistinguishable denoted as $X = \{X(a,k)\} \approx_s Y = \{Y(a,k)\}$ if for *every* distinguisher D and for every a and all large enough values of k:

$$\left|\mathsf{Pr}[D(X(a,k) = 1] - \mathsf{Pr}[D(Y(a,k) = 1]\right| = 0$$

$\diamondsuit$

This can be written as - If the probability of D guessing the correct distribution in the distinguishing game is denoted by $Adv_D(X,Y)$, then:

$$\left|Adv_D(X,Y)\right| = \frac{1}{2}$$

In this case, even unbounded powerful D cannot do any better than randomly guessing one of the two distributions. Both statistical and perfect indistinguishability belong to the information theoretic world of unbounded powerful adversaries.

4 Scope of the Security Definition considered so far

Our security definition applies for the following aspects of MPC:

1. Networks: Complete and Synchronous. A synchronous network, as seen earlier, is one that has a common clock, in terms of rounds.
2. Distrust: Centralized
3. Adversary: Threshold/Non-threshold, Polynomially bounded/Unbounded powerful, Semi-honest, Static, Rushing.

A *rushing* adversary can wait to compute or send his values in a round till he receives values from the other parties. We usually assume that the adversary is rushing.

5 Importance of Real world/Ideal World Definition Paradigm

We repeat again the advantages of using an Real world/Ideal World based definition over other defintiions:

- The definition paradigm allows us to use one security definition for all the computations required. For eg:
  1. Sum: $(x_1 + x_2 + .. + x_n) = f(x_1, x_2, .., x_n)$
  2. Oblivious Transfer: $(-, x_b) = f((x_1, x_2), b)$
  3. Byzantine Agreement: $(y, y, ..., y) = f(x_1, x_2, .., x_n)$ such that $y = \frac{\text{majority}(x_1, x_2, .., x_n)}{\text{default value}}$

- It is easy to tweak the ideal world scenario and weaken or strengthen security. Then we need to come up with a real world protocol that needs only to achieve what the ideal world achieves.

However, coming up with the right ideal world is tricky and requires skill, as we will see in the later part of this course, where we will deal with malicious adversaries.

6 Information Theoretic MPC with Semi-honest Adversary and Honest Majority

We are now ready to see a generic MPC protocol with honest majority and information theoretic security that can compute any PPT function f. We will assume an adversary that is semi-honest, threshold (with threshold t), static and unbounded powerful. The model of computation is arithmetic circuit. The first protocol for arithmetic circuits with information theoretic security was given by Michael Ben-Or, Shafi Goldwasser and Avi Wigderson in 'Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation' (Extended Abstract) at STOC 1988 [BGW88] for the following scenario (henceforth will be referred as BGM protocol).

– Networks: Complete and Synchronous.

– Distrust Model: Centralized

– Adversary: Threshold (t), Unbounded powerful, Semi-honest, Static.

The prime tool that is used to build BGW protocol is secret sharing. So we now focus on the secret sharing.

6.1 (n, t) Secret Sharing Scheme

Secret sharing existed before the birth of MPC, and it has been used for many other purposes [Shamir 1979, Blackley 1979]. In a secret sharing scheme there is a dealer who has a of a secret s. The dealer wants to share a secret among n parties $P_1, P_2, .., P_n$ using shares, say $s_1, s_2, \ldots, s_n$. We assume that the dealer can be one of the n parties nad there is an adversary $\mathcal{A}$ who can corrupt at most t parties out of the n parties. Any (n, t) secret sharing scheme has two phases:

Sharing Phase: The dealer deals the secret among the n parties in such a way that less than $(t + 1)$ parties should not get any information about the secret.

Reconstruction Phase: The parties come together with their shares to reconstruct the secret.

An (n, t) secret sharing scheme should satisfy the following two properties: **(a)** If at most t parties combine their shares then together they should have no information about the secret; **(b)** If more than t parties combine their shares then together they should have complete knowledge about the secret.

6.1.1 Shamir Sharing: (n, t) Secret Sharing Scheme for Semi-honest Adversaries

We now see a specific instantiation of an (n, t) Secret Sharing Scheme known as Shamir Sharing.

Sharing Phase: The dealer chooses a random polynomial f of degree *at most* t over F_p such that $p > n$ and the constant term is the secret, s. Note that we consider the degree of the polynomial as at most t, as the random choice could assign 0 as the coefficient of highest-degree term. The n parties $P_1, P_2, .., P_n$ are given $x_1, x_2, .., x_n$, the values of the polynomial evaluated at the *distinct, publicly-known* points $\alpha_1, \alpha_2, .., \alpha_n$. Then x_i is the share of the i^{th} party P_i.

Reconstruction Phase: The reconstruction is done using Lagrange's Interpolation. Since the adversary is semi-honest, all participants will send their shares and the reconstruction can proceed with any $t + 1$ shares.

6.1.2 Properties of Shamir-sharing

1. Any $(t + 1)$ parties have 'complete' information about the secret
2. Any t parties have 'no' information about the secret, i.e. Pr[Secret=s before secret sharing] − Pr[Secret=s after secret sharing] = 0.

Both the above properties can be proved using Lagrange's Interpolation.

6.1.3 Lagrange's Interpolation

Assume that $h(x)$ is a polynomial of degree at most t, and C is a subset of F_p of size $(t+1)$. For simplicity, we take $C = \{1, 2, .., t + 1\}$.

Theorem 1 *$h(x)$ can be written as $h(x) = \sum_{i \in C} h(i) \cdot \delta_i(x)$ where $\delta_i(x) = \prod_{j \in C, j \neq i} \frac{x-j}{i-j}$ where each δ_i is a polynomial of degree t, which evaluates to 1 at $x = i$, and 0 at any other point in C.*

Proof Both the LHS and the RHS evaluate to $h(i)$ for every $i \in C$. Both the LHS and the RHS have degree at most t. Now we consider the difference polynomial, LHS−RHS.

$$h(x) - \sum_{i \in C} h(i) \cdot \delta_i(x)$$

The LHS−RHS evaluates to zero for every $i \in C$, as they are equal.
$\Rightarrow$ Each $i \in C$ is a zero of the polynomial LHS−RHS. However, both LHS and RHS have degree at most t, therefore LHS−RHS also has degree at most t. Since the number of zeroes is $|C| = t + 1$, this implies that the polynomial LHS−RHS has more zeros than its degree.
$\Rightarrow$ LHS-RHS is the zero polynomial. $\Rightarrow$ LHS = RHS. Hence Proved.

6.1.4 Proof of Property 1 of Shamir Sharing

Given Theorem 1 in the context of Shamir Sharing, we have a group of $(t+1)$ parties, analogous to C in the above proof. Let the random polynomial picked in the sharing phase be $h(x)$.

- Each $i \in C$ is equivalent to the publicly-known points $\alpha_1, \alpha_2.., \alpha_{t+1}$ of the $(t+1)$ parties.
- Then each $h(i)$ in the above proof i.e. $h(x)$ evaluated at i; is equivalent to the share of the party, x_i.
- We also know $\delta_i(x)$ are public polynomials, since the points $i \in C$ are public. As a result, $\delta_i(0)$ are public values. These values are denoted by r_i.
- Then the secret $s =$ the constant term of $h(x) =$ the value of $h(x)$ at $0 = h(0)$. This can be written as a linear combination of the $h(i)$s (shares), with the help of the combiners $r_1, r_2..r_{t+1}$ according to the Theorem:

$$h(0) = \sum_{i \in C} h(i) \cdot \delta_i(0)$$

Hence, Shamir Sharing satisfies the property that any $t+1$ parties have complete information about the secret. The set of public combiners $(r_1, r_2, .., r_{t+1})$ is called the *recombination vector*.

6.1.5 Proof of Property 2 of Shamir Sharing

For any secret s from F_p if we sample $f(x)$ of degree at most t randomly such that $f(0) = s$, and consider the following distribution for any C that is a subset of $F_p \setminus \{0\}$ and of size t:

$$\big(\{f(i)\}_{i \in C}\big) \text{is a uniform distribution in } F_p^t$$

where F_p^t is the t^{th} Cartesian power of F_p. Note that C is a subset of $F_p \setminus \{0\}$ since $f(0)$ is the fixed secret. For a fixed secret s, a set of t coefficients from F_p^t defines a unique polynomial of degree at most t and hence a unique element from the above distribution. Similarly a fixed secret and an element from the above distribution form a set of $t+1$ values which uniquely define a polynomial of degree at most t. Hence, we see that the function $f_s : F_p^t \to F_p^t$ (t coefficients to t points for secret s) defined as above is bijective. For every s, the distribution is uniform and independent of s since we sample the polynomial randomly.

In the context of Shamir Sharing, C is a group of t parties who are colluding to try and reconstruct the secret. We now prove that Property 2 holds, and they have 'no' information about the secret. We use the random variables `SH` to denote the t shares of the parties, $\mathcal{S}$ to denote the secret space and `S` to denote the random variable for the secret itself. Then for any secret s_1, the following can be proved. The following argument will be true for any

$s_1 \in \mathcal{S}$.

$$
\begin{aligned}
&\mathsf{Pr}[\mathtt{S} = s_1 \text{ after secret sharing}] \\
&\quad = \mathsf{Pr}[\mathtt{S} = s_1 | \mathtt{SH} = \{f(i)\}_{i \in C}] && \text{A posteriori probability, given } t \text{ shares} \\
&\quad = \frac{\mathsf{Pr}[\mathtt{S} = s_1] \cdot \mathsf{Pr}[\mathtt{SH} = \{f(i)\}_{i \in C} | \mathtt{S} = s_1]}{\mathsf{Pr}[\mathtt{SH} = \{f(i)\}_{i \in C}]} && \text{Bayes' rule} \\
&\quad = \frac{\mathsf{Pr}[\mathtt{S} = s_1] \cdot \frac{1}{|F_p^t|}}{\mathsf{Pr}[\mathtt{SH} = \{f(i)\}_{i \in C}]} && \text{Uniform distribution in } F_p^t \text{ for fixed secret} \\
&\quad = \frac{\mathsf{Pr}[\mathtt{S} = s_1] \cdot \frac{1}{|F_p^t|}}{\sum\limits_{s_j \in \mathcal{S}} \mathsf{Pr}[\mathtt{SH} = \{f(i)\}_{i \in C} | \mathtt{S} = s_j] \cdot \mathsf{Pr}[\mathtt{S} = s_j]} && \text{Properties of conditional probability} \\
&\quad = \frac{\mathsf{Pr}[\mathtt{S} = s_1] \cdot \frac{1}{|F_p^t|}}{\sum\limits_{s_j \in \mathcal{S}} \frac{1}{|F_p^t|} \cdot \mathsf{Pr}[\mathtt{S} = s_j]} && \text{Uniform distribution in } F_p^t \text{ for fixed secret} \\
&\quad = \frac{\mathsf{Pr}[\mathtt{S} = s_1] \cdot \frac{1}{|F_p^t|}}{\frac{1}{|F_p^t|} \cdot \sum\limits_{s_j \in \mathcal{S}} \mathsf{Pr}[\mathtt{S} = s_j]} \\
&\quad = \mathsf{Pr}[\mathtt{S} = s_1] && \text{Denominator sum goes to 1} \\
&\quad = \mathsf{Pr}[\mathtt{S} = s_1 \text{ before secret sharing}] && \text{A priori probability}
\end{aligned}
$$

Hence we have proved Property 2 of Shamir Sharing.

In the next class, we will see some more properties of Shamir Sharing and the secure Arithmetic Circuit Evaluation protocol.