CSA E0 312: Secure Computation October 15, 2015

[Lecture 14-15]

Instructor: Arpita Patra *Submitted by: Divya Ravi*

1 Introduction

Let us embark on the second part of our journey of secure computation. We need to gear up as we enter into the world where the adversary is more powerful than before; he is no longer semi-honest but *malicious*. In this lecture, we start with a quick recap of the techniques that we have learnt to tackle a semi-honest adversary. We analyze if these techniques suffice to achieve security against malicious adversary as well. We discuss the possible ways in which a malicious adversary can behave in an attempt to breach the security of the protocol. In particular, we shall see specific ways in which the malicious adversary may potentially misbehave in BGW protocol of circuit evaluation. Then, we will see the appropriate measures that need to be incorporated in the BGW protocol to handle each of these malicious actions. We introduce a new primitive known as *Verifiable Secret Sharing*(VSS) which extends secret sharing to the case of corruption by malicious parties. We will also see a formal definition of VSS, and an example of a perfectly secure VSS protocol with $n = 3t + 1$ parties.

2 Recap

We started our journey of multiparty computation(MPC) by exploring the multitude of facets and dimensions that it offers. The area of MPC is rich and vast as it can take various forms depending on parameters such as the model of computation(arithmetic/boolean), type of network(synchronous/asynchronous), characteristic of adversary(semi-honest/malicious) to name a few. We first explored information-theoretic MPC with semi-honest adversary and honest majority. We saw a generic MPC protocol with honest majority(BGW) in which the function to be computed is represented in the form of an arithmetic circuit with addition and multiplication gates. We got exposed to the tool of Shamir-secret sharing which forms the basis of the BGW protocol. Efficiency of the protocol can be improved using the *offline-online paradigm*. We learnt how precomputed data of the offline-phase can be effectively used to make the online phase blazing fast. We studied some interesting techniques such as the Beaver's randomization technique and ways of generating particular types of raw data which are effectively used in evaluating multiplication gates in the online phase. We then saw an important result that it is not possible to design information-theoretic MPC in dishonest majority setting for all functions. This paved the way to the cryptographic world of MPC. An important primitive of cryptographic MPC is Oblivious Transfer(OT). OT can be generically constructed from a CPA-secure public key encryption(PKE) scheme with the property of public-key samplability. We moved on to study the GMW protocol for 2 parties which uses the primitives of OT and additive secret sharing to evaluate boolean circuits. GMW can be extended for multiparty, and certain optimizations of GMW such as

preprocessing of OT and OT extension improve the efficiency of the protocol significantly. The protocols studied so far had round complexity equal to the multiplicative depth of the circuit, this motivated us towards Yao's 2-party constant round protocol which uses cryptographic tools of garbled circuits and OT. There are also certain optimization techniques which can be effectively used in Yao's protocol such as point-and-permute, free XOR and reductions in garbled-circuit size i.e 4-to-3 garbled row reduction and 4-to-2 garbled row reduction. Lastly, we saw the BMR's protocol which extends the Yao's protocol to n players.

3 BGW with malicious adversary

The function to be computed in MPC can be represented in the form of an arithmetic circuit with addition and multiplication gates. Let us recall how the BGW protocol with semi-honest adversary evaluates the circuit in a distributed fashion.

1. **Input Sharing:** Each of the parties secret-share their input using a linear secret sharing scheme such as (n, t) Shamir sharing. To share a secret s, the dealer chooses a random polynomial of degree atmost t with constant term s. The polynomial is evaluated at n publicly known points. Say $x_1, x_2, \ldots x_n$ are the values of the polynomial evaluated at publicly known points $\alpha_1, \alpha_2, \ldots \alpha_n$ respectively. A party P_i receives the value x_i, this constitutes the party's share.

2. **Circuit evaluation:** The circuit is evaluated gate by gate while maintaining the following variant: *Given that the input values of the gate are* (n, t) *secret shared among the parties, the corresponding output value of the gate also remains* (n, t) *secret shared among the parties.*

    - Linear gates: Addition of (n, t) Shamir-shared secrets can be locally computed by individual parties due to the linearity of Shamir Sharing. Therefore evaluation of linear gates is absolutely free, no extra rounds or interactions are required for the computation.
    - Non-linear gates: Product of shares of two secrets does not result in (n, t) Shamir sharing of the product of secrets. Therefore, non-linear gates such as multiplication require an interactive technique known as degree reduction.

3. **Output Reconstruction:** The parties exchange their shares with each other and the Shamir-sharing of the output is reconstructed.

Now let us analyze the possible ways in which a malicious party may deviate from the above protocol.

1. In the first step i.e during input sharing, a malicious party may not choose a random polynomial of degree atmost t as he is supposed to. He might deal the shares in such a way that the shares define a polynomial of degree more than t.

2. In the last step i.e during output reconstruction, the parties exchange their shares using point-to-point pairwise channels. A malicious party may send a particular share

to one party and a different share to the other. How can we make sure that a party sends the same share to all the other parties? One way to do so it to use a broadcast channel to send the share. Assuming we have a physical broadcast channel, is the problem completely solved? No; even now nothing is preventing a corrupted party from providing an incorrect value instead of his correct share, thus effectively changing the value of the reconstructed secret. The other parties will have no way of knowing that the provided value is incorrect.

3. In step 2 of the protocol i.e during gate-by-gate circuit evaluation, linear gates will not pose a problem as they do not involve interaction. However, a malicious party may be able to breach the security of the protocol in degree reduction technique. We will elaborate on this a little later in section 5.

We have discussed three orthogonal problems that can occur if the BGW protocol designed for semi-honest adversary is used against a malicious adversary. Let us elaborate on each problem now and see how it can be overcome.

Consider the first problem. During Shamir-sharing, the dealer might send inconsistent shares to the parties. In other words, the shares received by the parties may lie on a polynomial of degree more than t. This will cause problems during the reconstruction step later. Thus *what we need is a mechanism to verify that the shares received by the parties are consistent*, i.e they indeed define a polynomial of degree atmost t whose constant term is the secret input of the dealer. A simple secret sharing scheme such as Shamir's does not suffice to handle a malicious adversary, this brings us to the primitive of *Verifiable Secret Sharing*.

4 Verifiable Secret Sharing

Verifiable secret sharing(VSS) extends secret sharing to the case of malicious corruption. Though it is designed as a measure to make malicious parties fail in their attempt to breach the security of the protocol; one must not forget that the scheme should also make sure that no hindrance is faced by the honest parties. It is not always the case that the dealer is corrupt. It is important to ensure that an honest dealer is not blocked. Thus the VSS protocols should satisfy the property of *duality*- VSS involves measures to block and identify malicious parties while simultaneously ensuring that the honest parties are not unnecessarily blocked.

We have seen that during output reconstruction step in the BGW protocol, if a malicious party sends an incorrect share, it might lead to the construction of a different polynomial and consequently an incorrect output if simple secret sharing is used. Thus, we must use a method that either prevents the corrupted parties from presenting incorrect shares, or ensures that $(n - t)$ correct shares are enough to reconstruct the correct secret output s. VSS provides a solution to this problem as well. Thus, VSS gives a way to overcome both the problems that we saw in step 1 and step 3 of BGW - Firstly, VSS forces the dealer to

consistently deal the shares. Second, VSS enables the correct reconstruction of the secret even if upto t shares are incorrect. Let us look at the formal definition of VSS.

Definition 1 In a VSS protocol there is a distinguished party known as dealer $D \in P$; that holds an input $s \in F$[1], referred to as the secret. The protocol consists of two phases, a *sharing phase* and a *reconstruction phase*. We call an n party protocol with adversary A an (n, t) *VSS protocol* if it satisfies the following conditions for a dealer D holding secret s:

1. **Secrecy:** If D is honest then A's view during the sharing phase reveals no information on s. Formally, A's view is identically distributed for all different values of s.

2. **Correctness:** If D is honest then the honest parties output s at the end of the reconstruction phase. Moreover, this is true for any choice of the random inputs of the uncorrupted parties and A's randomness.

3. **Strong Commitment:** If D is corrupted, then at the end of the sharing phase there is a unique value $s' \in F$, such that at the end of reconstruction phase all honest parties output s', irrespective of the behavior of the corrupted parties.

$\diamondsuit$

VSS proceeds in two phases - Sharing and Reconstruction. For an honest dealer, the VSS ensures that the secret s remains private till the end of sharing phase. If the dealer is corrupted, the VSS ensures that either the dealer is forced to commit some (n, t) secret s which is uniquely reconstructed in the reconstruction phase, or he is disqualified. In other words, commitment of a secret in sharing phase ensures that the t corrupted parties cannot change the secret in the reconstruction phase. VSS guarantees secrecy to honest parties and forces the dealer(who may be corrupt) to commit to a valid (n, t) shared secret in the sharing phase.

In the earlier lectures, we have studied simple secret-sharing *(SS)* such as Shamir's which assumed that the dealer is honest and the parties semi-honest. There is a variant of VSS known as *Honest Dealer VSS* or *SS with cheaters* in which the dealer is honest but the other parties may be malicious. (n, t) VSS is secure against t corrupted parties which may include the dealer as well.

5 Secure Evaluation of Multiplication Gate against malicious adversary

In the BGW protocol, non-linear gates such as multiplication involve an interactive technique known as *degree-reduction*. Suppose the n parties have their respective shares of a and b (inputs of the multiplication gate) with respect to t degree polynomial $f_a(x)$(used for Shamir-sharing of a) and $f_b(x)$(used for Shamir-sharing of b). By multiplying his shares $f_a(\alpha_i)$ and $f_b(\alpha_i)$, a party P_i can obtain the value of the product polynomial $g(x) = f_a(x)f_b(x)$ evaluated at a point α_i. Let z_i denote this product for party P_i i.e

[1] F is a field such that $|F| \geq n$

$z_i = f_a(\alpha_i) f_b(\alpha_i) = g(\alpha_i)$. The constant term of this polynomial $g(x)$ is $g(0) = f_a(0) f_b(0) = ab$. We know that any point on a polynomial of degree $2t$ can be expressed as a linear combination of $(2t+1)$ points on the polynomial using Legrange's interpolation. Suppose each party P_i shares their share z_i using (n,t) Shamir-sharing. Now, the parties can perform linear combination on these $n = 2t+1$ points of the polynomial to generate an (n,t) Shamir-sharing of $g(0) = ab$ as follows.

$$\begin{aligned} g(x) &= \sum_{i=1}^{n} z_i.\delta_i(x), \text{where } \delta_i(x) = \Pi_{j \in \{1,2 \ldots n\} j \neq i} \frac{x-j}{i-j} \\ ab = g(0) &= \sum_{i=1}^{n} z_i.\delta_i(0) = \sum_{i=1}^{n} z_i.r_i \end{aligned} \tag{1}$$

where $(r_1, r_2 \ldots r_n)$ is the public recombination vector. In the BGW protocol party P_i shared z_i using Shamir-sharing. To ensure that indeed (n,t) sharing is done properly, the party can use VSS instead. Suppose each party P_i used VSS to share respective z_i. The protocol still remains insecure against malicious party. This is because a malicious party P_i can use VSS to share an incorrect value z_i. This will subsequently lead to reconstruction of incorrect secret value when computed using linear combination as shown in the above equation 1. A party needs to prove that he is indeed sharing the correct z_i. This can be done using tools such as zero knowledge with $n = 2t+1$ parties. However, we need a mechanism that can reconstruct the correct secret s even though $n-t$ shares are corrupted. Suppose we view the Shamir-sharing as the Reed-Solomon code of the polynomial. According to coding theory, with a polynomial of degree t, Reed-Solomon code can correct upto $\frac{n-t-1}{2}$ errors. Suppose we have $n \geq 3t+1$ parties, $\frac{n-t-1}{2} \geq t$ errors can be corrected. In other words, even if t malicious parties send incorrect values, the honest parties can use error correction and recover if $n \geq 3t+1$.

From the above argument, it is clear that for perfect VSS , we need $n \geq 3t+1$. In the next section, we will look at an instance of a perfectly secure VSS protocol with $n = 3t+1$.

6 Perfect VSS with $n \geq 3t+1$

Let us recall how Shamir-secret sharing was done. The dealer chooses a random univariate polynomial $f(x)$ of degree atmost t whose constant term was the secret s. The polynomial is evaluated at n publicly known points say $\alpha_1, \alpha_2, \ldots \alpha_n$. Each party receives his own share i.e P_i receives $f(\alpha_i)$. Privacy is intact since atmost t among the n points may be leaked to the adversary, it will leak no information about the t degree polynomial. Correctness is ensured as $(t+1)$ correct shares suffice to reconstruct the secret using Legrange's Interpolation. Let us draw an analogous comparison of how VSS can be done. The dealer chooses a bivariate polynomial $F(x,y)$ of degree atmost t in x and y. Similar to simple secret sharing, the secret is the constant term of the polynomial i.e $F(0,0)$. Now the question is what constitutes the share of the parties? The dealer sends to party P_i two univariate polynomials: $f_i(x) = F(x,i)$ and $g_i(y) = F(i,y)$. We make the following two claims.

Claim 1 *The knowledge of t $F(x,i)$ and t $F(i,y)$'s leak no information about the secret $F(0,0)$.*

Claim 2 *$(t+1)$ $F(x,i)$ and t $F(i,y)$'s will suffice to uniquely determine $F(x,y)$ and thus the secret $F(0,0)$.*

$F(x,y)$	$F(x,1)$	$F(x,2)$	$F(x,j)$	$F(x,n)$
$F(1,y)$	$F(1,1)$	$F(1,2)$	$F(1,j)$	$F(1,n)$
$F(2,y)$	$F(2,1)$	$F(2,2)$	$F(2,j)$	$F(2,n)$
$F(i,y)$	$F(i,1)$	$F(i,2)$	$F(i,j)$	$F(i,n)$
$F(n,y)$	$F(n,1)$	$F(n,2)$	$F(n,j)$	$F(n,n)$

Proof. Consider the matrix view of the figure above which shows the distribution of shares among the parties. P_i receives $f_i(x)$ and $g_i(y)$ as his share which is represented by the ith column $F(x,i)$ and the ith row $F(i,y)$ of the matrix. The proof of the first claim can be done by the following counting argument- Atmost t parties are corrupted by the adversary, which implies that t rows and t columns are leaked to the adversary. Without loss of generality, let us consider that the first t rows and first t columns have been leaked to the adversary. $F(x,y)$ is a bivariate polynomial of degree atmost t in x and y. Hence the polynomial is uniquely defined by $(t+1)^2$ points[2]. A row i defines the polynomial $g_i(y) = F(i,y)$ of degree t. This means a row i contains $(t+1)$ independent points that define the polynomial $g_i(y)$. So, the adversary to whom t rows have been leaked gets access to $t(t+1)$ independent points. Now let us consider the column 1 that has been leaked to the adversary. This column defines the t degree polynomial $f_1(x) = F(x,1)$ and has $(t+1)$ independent points. Among these, we have already counted $F(1,1), F(2,1) \ldots F(t,1)$ while counting the points on the first t rows. We get a single additional independent point i.e $F(t+1,1)$. For each of the t columns, we get one additional point not counted before. Thus, we get t points in this manner. Totally, we have obtained $t(t+1)+t$ points on the polynomial. However $(t+1)^2$ independent points are needed to define $F(x,y)$; the adversary falls short of one point. Thus the distributed shares leak no useful information about the secret $F(0,0)$. Consider the second claim. $(t+1)$ parties have knowledge of $(t+1)$ rows and $(t+1)$ columns. These include $(t+1)^2$ points on the bivariate polynomial $F(x,y)$ that suffice to define the polynomial uniquely. The secret $F(0,0)$ can be correctly reconstructed by any $(t+1)$ honest parties.

The intuition for the design of VSS as above is as follows: The parties need a way to verify that the dealer has consistently distributed the shares. The use of bivariate polynomial enables pairwise checking among the parties. Consider two parties P_i and P_j. P_i receives

[2]The polynomial $F(x,y)$ contains $(t+1)(t+1)$ terms or coefficents. To define the polynomial uniquely, we consider coefficients as variables, this means we need $(t+1)^2$ independent equations obtained from the value of the polynomial evaluated at $(t+1)^2$ points.

$F(x,i)$ and $F(i,y)$ as his share. P_j receives $F(x,j)$ and $F(j,y)$ as his share. The parties have common shares $F(i,j)$ and $F(j,i)$ and check among themselves whether the values match. For an honest dealer, this check will surely pass. However a corrupted dealer who has not distributed shares consistently can be detected by a pair of honest parties who check in this manner. This was not feasible in single sharing where each party was given only one share which cannot be revealed. The use of bivariate polynomials is to verify the pairwise consistency of the distributed shares. The conflicts among the parties are resolved in the sharing phase of VSS. At the end of this phase, there exists a unique bivariate polynomial defined by the shares (consistent pairwise among the parties) held by the honest parties. Now we have two t degree polynomials $F(x,0)$ and $F(0,y)$ with constant term $F(0,0)$ as the secret. Each party P_i has its respective share $F(i,0)$ and $F(0,i)$ corresponding to these two polynomials. Any of these two polynomials can be interpreted as Shamir-Sharing of the secret $F(0,0)$ and the reconstruction can be done in the usual manner. This is a high-level overview of the way VSS works, below we define the exact steps of the protocol.

Perfect VSS with $n = 3t + 1$

Sharing Phase

1. The dealer chooses a random bivariate polynomial $F(x,y)$ of degree t in x and y and with $F(0,0) = s$. The dealer sends to P_i the polynomials $f_i(x) = F(x,i)$ and $g_i(y) = F(i,y)$.
2. Party P_i sends $f_i(j)$ and $g_i(j)$ to P_j.
3. Let $f'_{j,i}$ $g'_{j,i}$ denote the values P_i received from P_j. If $f'_{j,i} \neq g_i(j)$ or $g'_{j,i} \neq f_i(j)$, then P_i sends *complain*(i,j) to the dealer.
4. If the dealer receives *complain*(i,j) from P_i in the last round, then the dealer sends *complain*(i,j) to P_j.
5. For every ordered pair (i,j), the parties P_i, P_j and the dealer do the following
   - If P_i had sent *complain*(i,j) to the dealer in round 3, then P_i broadcasts the conflicting share i.e $g_i(j)$ and $f_i(j)$ to all parties.
   - If P_j received *complain*(i,j) from the dealer in the previous round, then P_j broadcasts the conflicting share i.e $g_j(i)$ and $f_j(i)$ to all parties.
   - If the dealer received *complain*(i,j) from P_i in round 3, the dealer broadcasts $F(i,j)$ and $F(j,i)$ to all parties.

   We say a party P_i is *unhappy* if the value P_i broadcast does not match with the value broadcast by the dealer.
6. For each unhappy party P_j, the dealer broadcasts the polynomials $f_j(x)$ and $g_j(y)$. Each party P_i who is not unhappy, broadcast their pairwise common shares with the parties in unhappy set i.e $b'_{ij} = f_i(j)$ and $c'_{ij} = g_i(j)$.

In this round, the values broadcast by the party P_i who is not unhappy should be consistent with the polynomials broadcast by the dealer for the unhappy parties. The party P_i complains if for some unhappy party P_j, the dealer broadcasts $f_j(x)$ and $g_j(y)$ but $b'_{ij} \neq g_j(i)$ and $c'_{ij} \neq f_j(i)$. If all the values that P_i broadcast are consistent with the polynomials corresponding to every unhappy party, then P_i is said to be happy.

Output Determination. The dealer is disqualified if the number of happy parties is less than $(n-t)$. If a dealer is not disqualified, happy party P_i keeps the polynomials $f_i(x)$ and $g_i(y)$ that it received from the dealer in the first round. An unhappy party P_i takes the polynomials broadcast by the dealer in the last round as $f_i(x)$ and $g_i(y)$. Now, either of the two t degree polynomials $f_i(x) = F(x,0)$ or $g_i(y) = F(0,y)$ can be considered as (n,t) Shamir-sharing of secret $F(0,0)$ with party P_i containing the share $F(i,0)$ and $F(0,i)$ respectively. A t degree polynomial $h(x)$ can be reconstructed using Reed-Solomon error correction on the $(3t+1)$ shares of the parties. The secret reconstructed is $h(0)$.

Analysis of Protocol. Let us check if this protocol satisfies all the properties of VSS mentioned in its definition in section 4.

Consider an honest dealer. In round 3, for any pair of honest parties (P_i, P_j), there will be no conflicts among them since the dealer has distributed shares correctly. Since there will be no complaint and the dealer is honest, their common shares will not be broadcast in round 5. Secrecy of the input of the dealer is preserved as no extra points on the polynomial $F(x,y)$ unknown to the adversary have been leaked. Another important observation one can make is that an honest party P_i can never be unhappy since the values he broadcast will always match with the value broadcast by the honest dealer. This means there are atleast $(n-t)$ happy parties in this case which guarantees that an *honest dealer can never be disqualified.* In round 6, an honest dealer broadcasts polynomials of only unhappy parties; this does not leak any extra information since none of the honest parties can be unhappy in case of an honest dealer. In the reconstruction step, the correct secret will be reconstructed as the shares of the honest parties are consistent with the original polynomial chosen. Therefore, correctness and secrecy of input holds for honest dealer.

Consider a malicious dealer. Suppose two honest parties (P_i, P_j) hold inconsistent shares. The inconsistency is made known to all in round 5. The share of atmost one party may match with the dealer. Suppose none of the shares match with the one broadcast by dealer in round 5, then both these honest parties will be unhappy. They will consider the polynomials sent by the dealer in round 6; here the dealer is forced to commit since he has to broadcast the polynomials corresponding to the unhappy parties. One can observe that the property of commitment is satisfied here. A similar argument holds in case the pair of honest parties has exactly one unhappy party. The honest party not in the unhappy set is consistent with all the other non-unhappy parties(otherwise there would have been a complaint in round 2). This honest party will complain if its share is not consistent with

the polynomial broadcast by the dealer corresponding to every unhappy party in round 6. The honest party will be happy only if its shares match with the polynomials broadcast corresponding to all the unhappy parties. The dealer is not disqualified only if the number of happy parties is atleast $(n-t) > 2t+1$ which will include atleast $(t+1)$ honest parties. Therefore at the end of sharing phase, all the honest parties have pairwise consistent shares. In other words, there exist a unique bivariate polynomial corresponding to the shares of the honest parties and committing to some secret s'.

Thus, the protocol satisfies the properties which define a VSS i.e correctness, secrecy and commitment.

6.1 Reduction to 4-round VSS

The Round complexity of the VSS protocol i.e the number of rounds in the sharing phase can be reduced by some slight modifications. One can observe in the protocol, that when a party P_i detects a pairwise inconsistency in shares with another party P_j; first, the party sends a complaint in one round, and in another round the party broadcasts the value of the conflicting share i.e $F(i,j)$ and $F(j,i)$. Both these steps can be combined into a single round. Also, in round 2, each player sends his pairwise common share to every other player. Instead of this communication, a more efficient way would be if a player P_i broadcasts the value he is supposed to send to another player P_j, by padding with a random number known only to the corresponding party P_j. The exchange of these random pads, known only among a pair of parties can be included in the first round. The exact steps of the protocol are outlined below.

4-round perfect VSS with $n = 3t+1$

Sharing Phase:

1.
   - The dealer D chooses a random bivariate polynomial $F(x,y)$ of degree t in x and y and with $F(0,0) = s$. The dealer sends to P_i the polynomials $f_i(x) = F(x,i)$ and $g_i(y) = F(i,y)$.
   - Party P_i sends to every other party P_j an independent random pad r_{ij}
2. Party P_i broadcasts two values
   - $a_{ij} = f_i(j) \oplus r_{ij}$ (r_{ij} is the pad P_i sent to P_j)
   - $b_{ij} = g_i(j) \oplus r_{ji}$ (r_{ji} is the pad P_i received from P_j)
3. For each pair $a_{ij} \neq b_{ji}$, the following is done
   - P_i broadcasts $\alpha_{ij} = f_i(j)$
   - P_j broadcasts $\beta_{ji} = g_j(i)$
   - D broadcasts $\gamma_{ij} = F(j,i)$

A party P_i is *unhappy* if its broadcasted value does not match with that of the dealer. If there are more than t unhappy players, the dealer is disqualified.

4. For every unhappy player P_i, the dealer broadcasts the polynomial $f_i(x)$ and each happy party broadcasts $g_j(i)$.

5. **Local computation:** For every public polynomial $f_i(x)$, check that for atleast $(2t+1)$ parties it holds that $g_j(i) = f_i(j)$. If not, the dealer is disqualified.

Reconstruction Phase: Every happy party P_i provides his share $s_i = f_i(0)$. Every unhappy party consider $f'_i(x)$ broadcast by the dealer in round 4 and provides his share as $s_i = f'_i(0)$. $g(y)$ is the polynomial constructed by Reed-Solomon error-correction on $s_1, s_2 ... s_n$. The secret reconstructed is $g(0)$.

Summary. In this lecture we discussed what changes need to be incorporated in the BGW protocol for security against malicious adversary. Verifiable Secret Sharing (VSS), which is a very important tool that extends secret sharing to malicious setting was introduced. We saw an example of a 6-round perfect VSS protocol with $n = 3t + 1$ parties. Lastly, we saw how the round complexity of this protocol can be reduced to 4-rounds.

References

[1] Arpita Patra. *http://drona.csa.iisc.ernet.in/ arpita/SecureComputation15.html* . E0 312 - Secure Computation Course Lecture Slides

[2] Katz, Jonathan, and Chiu-Yuen Koo. Round-efficient secure computation in point-to-point networks. *Advances in Cryptology-EUROCRYPT 2007*. Springer Berlin Heidelberg, 2007. 311-328.

[3] Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T. (2001, July). The round complexity of verifiable secret sharing and secure multicast. *In Proceedings of the thirty-third annual ACM symposium on Theory of computing* (pp. 580-589). ACM 2001.