ETH Zurich, Department of Computer Science
SS 2018

Prof. Ueli Maurer
Dr. Martin Hirt
Chen-Da Liu Zhang

Cryptographic Protocols
Notes 9

Scribe: Sandro Coretti (modified by Chen-Da Liu Zhang)

> *About the notes:* These notes serve as written reference for the topics not covered by the papers that are handed out during the lecture. The material contained therein is thus a *strict* subset of what is relevant for the final exam.

This week, the notes treat an information-theoretically secure (distributed) commitment protocol that can be used to instantiate last week's generic actively secure MPC protocol. The notes also treat an impossibility proof for actively secure MPC with $t \geq n/2$.

9.1 Information-Theoretically Secure Commitment Protocol

In this section we provide a homomorphic, information-theoretically secure commitment scheme (cf. Section 8.2) along with commitment multiplication (CMP) and transfer (CTP) protocols. The way around the fact that there can be no commitment scheme that is both information-theoretically binding and hiding is to construct a *distributed* scheme that is based on Shamir's sharing scheme. Instantiating our generic protocol with this commitment scheme results in (a variant of) the protocol by [BGW88].

In the following we assume that the number of corrupted parties is $t < n/3$. We start by constructing protocols Commit and Open, which allow some dealer D to (w.r.t. all players) commit to a value and to open it, respectively. Recall that an execution of them can be rejected by the players. We require that they satisfy the following properties:

- Consistency: If Commit or Open is rejected by some honest player, all honest players do so. If D is honest, no honest player rejects.
- Uniqueness: If Commit is successful, D is committed to some value, i.e., there exists only one value that is accepted in Open.
- Privacy: If D is honest an commits to some value, then, in Commit, the adversary obtains no information about this value.

It is easily seen that these properties imply that the scheme is perfectly binding and hiding.

Commit. Recall that the players want to evaluate an arithmetic circuit over a finite field $\mathbb{F}$ of size $q > n$ (cf. Section 7.2). The main idea is that the dealer D, to commit to some value s, chooses a random polynomial $g(x)$ of degree at most t with $g(0) = s$ and sends the *commit share* $s_i := g(\alpha_i)$ to P_i for each i. Furthermore, D somehow "proves" that the commit shares indeed lie on a polynomial of degree at most t. The protocol is given in Figure 1.

Distributed Commit Protocol COMMIT

1. Distribution: To share secret s, dealer D chooses a random bivariate polynomial of (single) degree at most t
$$f(x,y) = \sum_{i,j=0}^{t} f_{ij}\, x^i y^j, \quad \text{where } f_{00} = s \text{ and } f_{ij} \in_R \mathbb{F} \text{ for } i,j \neq 0$$
and sends the polynomials $h_i(x) := f(x, \alpha_i)$ and $k_i(y) := f(\alpha_i, y)$ to P_i for $i = 1, \ldots, n$.
2. Consistency checks: For $1 \leq i,j \leq n$: Players (P_i, P_j) verify that $h_i(\alpha_j) = k_j(\alpha_i)$. To that end, P_i sends the value $h_i(\alpha_j)$ to P_j, who compares the received value to $k_j(\alpha_i)$. Each player P_i broadcasts a complaint for all coordinates (i,j) where the values do not match. Dealer D must answer such complaints by broadcasting $f(\alpha_i, \alpha_j)$.
3. Accusations: If, in step 2, dealer D broadcasts values that are not consistent with polynomials $h_i(x)$ and $k_i(y)$ of some player P_i, this player accueses the dealer via broadcast. The dealer must answer such an accusation by broadcasting both $h_i(x)$ and $k_i(y)$. If the broadcasted polynomials are inconsistent with the polynomials of some other player P_j (i.e., $h_i(\alpha_j) \neq k_j(\alpha_i)$ or $k_i(\alpha_j) \neq h_j(\alpha_i)$), this player also accuses the dealer. This continues until no new players accuse D.
4. Determine commit share: If, in step 3, dealer D was accused by more than t players, refused to answer any of the accusations, or the answers contradicted each other, he is disqualified.
Otherwise, each player computes his commit share $s_i = k_i(0)$ as the output of the protocol, where, if P_i accused D, he uses the polynomial broadcast by D. The dealer's output is the polynomial $g(x) = f(x, 0)$.

Figure 1: *Distributed commit protocol secure against up to $t < n/3$ corrupted players.*

Open. Assume the dealer D is committed to some value s by a polynomial $g(x)$ and each player holds the share $s_i = g(\alpha_i)$. To open the commitment, D first broadcasts $g(x)$ (by broadcasting the coefficients). Each player P_i then checks if his share lies on the broadcast polynomial and accuses the dealer via broadcast if this is not the case. If at most t players accuse D, the opening is accepted. The protocol is depicted in Figure 2.

Let us turn to the analysis of the commitment scheme. It is straight-forward to verify that the presented commitment scheme satisfies the consistency property. Also, it is easily seen that the presented commitment scheme maintains privacy: At the beginning, the adversary gets up to t pairs of polynomials $(h_i(x),\, k_i(y))$, which give no information about the secret (cf. Slides 07). The reader can easily verify that, after this, only values are revealed that the adversary already knows.

To show that the uniqueness property is satisfied, suppose some dealer D is not disqualified. After step 3, the polynomials of all honest players are pairwise consistent. Consider the polynomial $f'(x,y)$ defined by the polynomials $h_i(x), k_i(y)$ of the $t+1$ honest players with the lowest indices. Moreover, consider the polynomial $h_j(x)$ of some other honest player P_j. It is consistent with the polynomials $k_i(y)$ of the $t+1$ first players. Thus $h_j(x) = f'(x, \alpha_j)$. Similarly, $k_j(y) = f'(\alpha_j, y)$. It follows that the polynomials of all honest players lie on a well-defined

Distributed Open Protocol Open

Starting point: Dealer D is committed to some value s by a polynomial $g(x)$ of degree at most t and each player P_i holds the commit share $s_i = g(\alpha_i)$.
Goal: Every player learns s.

1. Dealer D broadcasts $g(x)$.
2. Every player P_i checks that his share lies on the broadcasted polynomial, i.e., whether $s_i = g'(\alpha_i)$. If not, P_i accuses D via broadcast.
3. If more than t players accused D the protocol is rejected. Otherwise the opened value is $s = g'(0)$.

Figure 2: *Distributed open protocol secure against up to $t < n/3$ corrupted players.*

polynomial $f'(x, y)$ of degree at most t. In particular, the shares $k_i(0)$ lie on a polynomial $g(x) = f'(x, 0)$ of degree at most t and the secret is uniquely defined.

Suppose now that a corrupted dealer broadcasts a polynomial $g'(x) \neq g(x)$ in the open protocol. Since they must disagree in at least $n - t$ positions α_i, there will be at least $n - 2t \geq t + 1$ honest players accusing the dealer, who will thus be disqualified.

Commitment transfer protocol. Such a protocol is discussed in Exercise 10.1.

Commitment multiplication protocol. As a final step, we will construct a commitment multiplication protocol that allows a player committed to two values a, b to commit to their product $c = ab$ (see Figure 3). The protocol is generic, that is, it works with any commitment scheme (also cryptographic ones). It requires, however, that $t < n/3$. We defer its analysis to Exercise 10.2.

Commitment Multiplication Protocol CMP

Starting point: Dealer D is committed to two values a and b.
Goal: D is committed to the product $c = ab$ of the two values.

1. D computes $c = ab$ and commits to it.
2. D uses the CSP for both for a and b, which results in every player P_i being committed to two shares a_i and b_i.
3. D uses the CSP for c. However, he uses a polynomial of degree $2t$ that is the product of the two polynomials used to share a and b. P_i is now committed to c_i.
4. Each player P_i checks whether $c_i = a_i b_i$. If not, he opens the commitments to the three values, thus proving that D has cheated. If no player provides such a proof, the commitment to c from step 1 is accepted.

Figure 3: *Commitment multiplication protocol secure against up to $t < n/3$ corrupted players.*

9.2 Impossibility of Actively Secure MPC with $t \geq n/2$

In this section we prove that no MPC protocol can be secure against an adversary corrupting half of the players or more. Similarly to the impossibility proof for *information-theoretically* secure MPC against a *passive* attacker corrupting a majority of the players (cf. Section 7.3), we first show that there can be no protocol that securely computes the function $f(x_A, x_B) = x_A \wedge x_B$

between Alice and Bob (i.e., with $t = 1$ and $n = 2$). Then, the general case follows analogously (and is not repeated here). The attack provided below is efficient and thus precludes the existence of cryptographically secure protocols as well.

Consider a *shortest* (in terms of the number of messages sent) protocol between Alice and Bob that securely evaluates f and assume w.l.o.g. that Alice sends the last message. Modify said protocol s.t. Alice does not send that last message. Clearly, in the modified protocol, Alice still learns the output. Moreover, neither Alice's nor Bob's security can be violated by Alice not sending the last message. Thus, since the modified protocol is shorter and the original one was a shortest secure one, Bob cannot learn the output of f in the modified protocol (as, otherwise, the modified protocol would be secure). This implies that Alice, by dropping the last message in the protocol, can break the security of the original protocol, which is a contradiction.

Note that this proof can be generalized to protocols with varying number of rounds: Alice simply guesses in which of the at most polynomially many rounds she has to stop in order to break the protocol, which still gives her non-negligible success probability.

References

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10, 1988.