CMSC 858K — Introduction to Secure Computation September 13, 2013

Lecture 4

Lecturer: Jonathan Katz *Scribe(s): Aishwarya Thiruvengadam*

1 Summary

In this lecture, we present two variants of the Even-Goldreich-Lempel Oblivious Transfer (OT) protocol we saw in the previous lecture. We show how to do domain extension for OT (i.e.) go from ℓ-bit string OT to n-bit string OT. We also show how to construct 1-out-of-N OT from 1-out-of-2 OT.

2 Variants of Even-Goldreich-Lempel OT protocol

We present the following variant of the OT protocol seen in the last lecture. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a CPA secure encryption scheme that has an algorithm Samp that can sample valid looking ciphertexts.

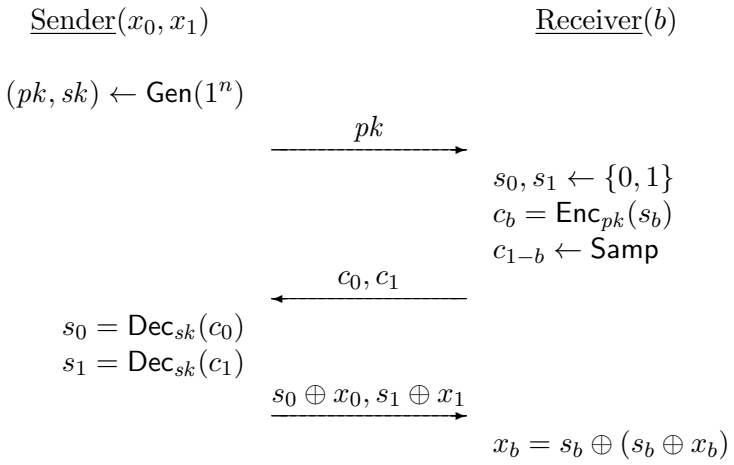

Figure 1: Variant of Even-Goldreich-Lempel OT

The protocol in Figure 1 guarantees information theoretic security against the receiver and computational security against the sender. This protocol can be instantiated with the El Gamal encryption scheme.

The next variant is based on the Decisional Diffie-Hellman (DDH) assumption. Consider the group G of order q with generators g_0 and g_1. The DDH assumption states that, for $r, r' \in \mathbb{Z}_q$, the tuples $(g_0, g_1, {g_0}^r, {g_1}^r)$ and $(g_0, g_1, {g_0}^r, {g_1}^{r'})$ are computationally indistinguishable.

Consider the following protocol with group G of order q and generators g_0 and g_1.

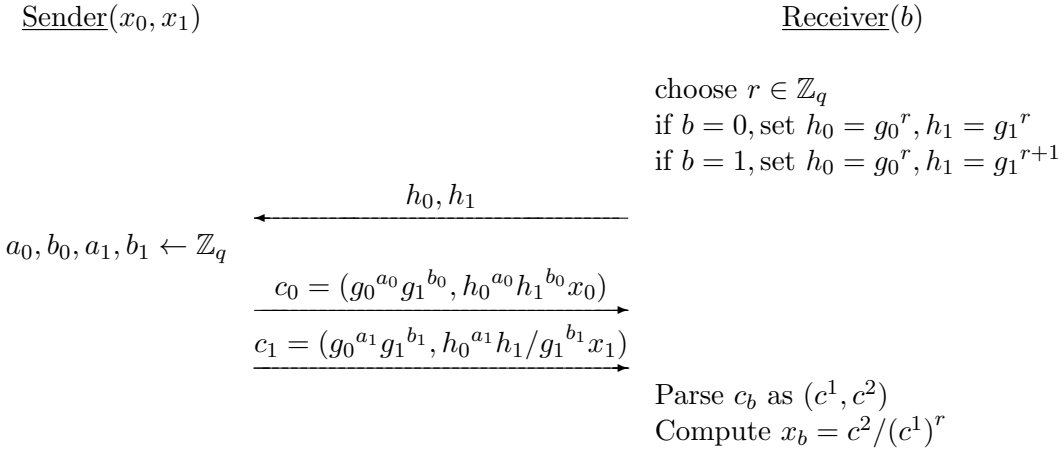

Figure 2: OT protocol based on DDH

The protocol in Figure 2 is information theoretically secure against the receiver and computationally secure against the sender. Intuitively, for the sender, the security can be reduced to the DDH assumption. If $b = 0$, then he sees exactly a DDH tuple. If $b = 1$, the two exponents are dependent but this can still be reduced to the DDH assumption. For the receiver, it is possible to recover message x_b while he receives no information on x_{1-b}. Note that $k = {g_0}^a {g_1}^b$ imposes a single linear constraint $\log_{g_0} k = x + y \log_{g_0} g_1$ on x and y. And for x_{1-b}, the exponents r and r' do not match in the ciphertexts. And, $k' = {h_0}^a {h'_1}^b$ implying $\log_{g_0} k' = a \log_{g_0} h_0 + b \log_{g_0} h'_1$. Given that $r \neq r'$ for x_{1-b}, the two equations are not multiples of each other and hence, the receiver cannot recover x_{1-b}.

3 More constructions using OT hybrid

3.1 Domain Extension

We show a construction of OT for ℓ-bit strings from OT for n-bit strings.

The sender and receiver use the OT protocol for n-bits to choose key k_b corresponding to the receiver's choice bit from keys k_0, k_1 held by the sender. The sender sends both ℓ-bit strings m_0, m_1 encrypted using the keys k_0, k_1 respectively. The receiver then recovers the correct message using his key k_b.

The protocol is as described in Figure 3.

Note that n is the security parameter. This protocol cannot be used for small values of n. For example, when $n = 1$, the key would be of length 1 and this protocol is no longer secure.

3.2 Constructing 1-out-of-N OT from 1-out-of-2 OT

Let us show how to construct 1-out-of-N OT (i.e.) the receiver choosing to receive 1 string among N held by the sender from a 1-out-of-2 OT construction. To illustrate the general idea, we show a construction of 1-out-of-4 OT from 1-out-of-2 OT.

$\underline{\text{Sender}}(m_0, m_1 \in \{0,1\}^\ell)$		$\underline{\text{Receiver}}(b)$
$k_0, k_1 \leftarrow \{0,1\}^n$		
Send k_0, k_1 to OT		Send b to OT
		Receive k_b from OT
	$c_0 = \mathsf{Enc}_{k_0}(m_0), c_1 = \mathsf{Enc}_{k_1}(m_1)$ $\longrightarrow$	
		$m_b = \mathsf{Dec}_{k_b}(c_b)$

Figure 3: OT protocol for ℓ-bit strings

The sender holds two pairs of keys and the receiver chooses one key from each pair corresponding to the message he wants to receive. Note that the receiver has to hold two bits here. Let F be a pseudorandom function such that $F_k(.) : \{0,1\}^n \to \{0,1\}^n$.

$\underline{\text{Sender}}(m_{00}, m_{01}, m_{10}, m_{11})$		$\underline{\text{Receiver}}(b_1 b_2)$
$k_0, k_1, k'_0, k'_1 \leftarrow \{0,1\}^n$		
Send k_0, k_1 to OT		Send b_1 to OT
		Receive k_{b_1} from OT
Send k'_0, k'_1 to OT		Send b_2 to OT
		Receive k'_{b_2} from OT
	$c_{00} = m_{00} \oplus F_{k_0}(00) \oplus F_{k'_0}(00)$ $\longrightarrow$	
	$c_{01} = m_{01} \oplus F_{k_0}(01) \oplus F_{k'_1}(01)$ $\longrightarrow$	
	$c_{10} = m_{10} \oplus F_{k_1}(10) \oplus F_{k'_0}(10)$ $\longrightarrow$	
	$c_{11} = m_{11} \oplus F_{k_1}(11) \oplus F_{k'_1}(11)$ $\longrightarrow$	
		$m = c_{b_1 b_2} \oplus F_{k_{b_1}}(b_1 b_2) \oplus F_{k'_{b_2}}(b_1 b_2)$

Figure 4: 1-out-of-4 OT protocol

Note that using the corresponding keys as pads for each message (i.e.) setting $c_{ii'} = m_{ii'} \oplus k_i \oplus k'_{i'}$ reveals more than just the message chosen by the receiver. This is because the receiver can xor all the ciphertexts to learn the xor of all the messages.

Another secure construction would be to consider the keys as encryption keys and encrypt the messages as $c_{ii'} = \mathsf{Enc}_{k_i}(\mathsf{Enc}_{k'_{i'}}(m_{ii'}))$ and let the receiver obtain the correct message by decrypting the corresponding ciphertext with the keys he received from the OT.