COS 597C: Recent Developments in Program Obfuscation — Princeton University
Lecture 7 (10/06/16)
Lecturer: Mark Zhandry — Scribe: Jordan Tran

Notes for Lecture 7

1 Introduction

In this lecture, we show how to build fully homomorphic encryption (FHE) from indistinguishability obfuscation (iO). In previous applications, we use iO in conjunction with one-way functions (OWF). However, it seems that OWF alone is not sufficient in this case. We know that FHE implies collision resistance hash function (CRHF) [IKO'05]. Therefore, if we can obtain FHE from iO and OWF, we will also get CRHF. However, there is an argument (not a concrete proof) that iO and OWF alone is unlikely to imply CRHF [AS'15]. As such, we believe that in order to build FHE from iO, we are going to need more than just OWF.

2 Construction of FHE from iO

2.1 First Attempt

Suppose we have a public key encryption (PKE) scheme consists of three algorithms: $\mathsf{Gen}_{\mathsf{PKE}}$, $\mathsf{Enc}_{\mathsf{PKE}}$, and $\mathsf{Dec}_{\mathsf{PKE}}$. We are going to start of with a FHE scheme that looks very similar to what we have seen with other applications of iO:

- $\mathsf{Gen}(\lambda)$: $(\mathsf{dk}, \mathsf{ek}) \leftarrow \mathsf{Gen}_{\mathsf{PKE}}(\lambda)$, $\mathcal{K} \leftarrow \mathsf{Gen}_{\mathsf{PRF}}(\lambda)$. Sets ek as the encryption key and $(\mathsf{dk}, \mathcal{K})$ as the decryption key. Outputs $(\mathsf{ek}, (\mathsf{dk}, \mathcal{K}))$.
- $\mathsf{Enc}(\mathsf{ek}, m)$: $\mathsf{Enc}_{\mathsf{PKE}}(\mathsf{ek}, m)$
- $\mathsf{Dec}(\mathsf{dk}, ct)$: $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{dk}, ct)$
- $\mathsf{Eval}(\mathsf{op}, ct_0, ct_1)$: Outputs $\mathsf{Obf}(P_{\mathsf{dk},\mathcal{K},\mathsf{ek}})(\mathsf{op}, ct_0, ct_1)$ where op is an operation $(+, \times)$ and $\mathsf{Obf}(P_{\mathsf{dk},\mathcal{K},\mathsf{ek}})$ is an obfuscation of the following program:

```
$P_{\mathsf{dk},\mathcal{K},\mathsf{ek}}(\mathsf{op}, ct_0, ct_1)$ :

    $m_b = \mathsf{Dec}(\mathsf{dk}, ct_b)$        $b \in \{0, 1\}$
    $m = m_0 \ \mathsf{op} \ m_1$
    $r = \mathsf{PRF}(\mathcal{K}, (\mathsf{op}, ct_0, ct_1))$
    Outputs $\mathsf{Enc}(\mathsf{ek}, m; r)$
```

Note that obfuscation only works with deterministic programs. Since the encryption algorithm is necessarily randomized, we need to explicitly calculate the random coin r and give it to Enc in $P_{\mathsf{dk},\mathcal{K},\mathsf{ek}}$. However, we do not know how to prove security with the above construction. Therefore, we will have to relax the definition of FHE.

2.2 Leveled FHE

In a regular FHE scheme, the homomorphic operation should produce a ciphertext under the same encryption key as the two input ciphertexts. However, in a leveled FHE scheme, we will allow the homomorphic operation to produce a ciphertext under a different encryption key.

So instead of just one pair of $(\mathsf{ek}, \mathsf{dk})$, we now have $(\mathsf{ek}_1, \mathsf{dk}_1), (\mathsf{ek}_2, \mathsf{dk}_2), ..., (\mathsf{ek}_n, \mathsf{dk}_n)$. If we add or multiply two ciphertexts under ek_1, we will get a ciphertext under the next level ek_2. As long as the users have all the decryption keys, they would be be able to decrypt any messages. If we want to perform operation on two ciphertexts at different levels, we can simply add an encryption of 0 (under the appropriate ek) to the ciphertext of lower level to "push" it up the levels.

Our homomorphic evaluation now consists of the obfuscations of multiple programs (one for each level of encryption).

```
$P^{i}_{\mathsf{dk}_i,\mathcal{K}_i,\mathsf{ek}_{i+1}}(\mathsf{op}, ct_0, ct_1)$ :

        $m_b = \mathsf{Dec}(\mathsf{dk}_i, ct_b)$                $b \in \{0, 1\}$
        $m = m_0 \ \mathsf{op} \ m_1$
        $r = \mathsf{PRF}(\mathcal{K}_i, (\mathsf{op}, ct_0, ct_1))$
        Outputs $\mathsf{Enc}(\mathsf{ek}_{i+1}, m; r)$
```

However, this leveled FHE scheme still implies CRHF, and therefore is affected by the implausibility result which suggests that we cannot build it from just iO and OWF [AS'15]. As such, we will need to modify this program.

The last program in the sequence is $P^{n-1}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}$. Our goal is to change this program so that instead of outputting the resulting ciphertext, it would always output an encryption of 0:

```
$Q^{n-1}_{\mathcal{K}_{n-1},\mathsf{ek}_n}(\mathsf{op}, ct_0, ct_1)$ :

    $r = \mathsf{PRF}(\mathcal{K}_{n-1}, (\mathsf{op}, ct_0, ct_1))$
    Outputs $\mathsf{Enc}(\mathsf{ek}_n, 0; r)$
```

Since $Q^{n-1}_{\mathcal{K}_{n-1},\mathsf{ek}_n}$ ignores the input messages completely, it does not need dk_{n-1}. If we are able to change the last level of P into Q, then we will be able to work up the chain of programs and get rid of all dk_i.

2.3 Indistinguishability of P and Q

In order to argue indistinguishability between P and Q, we are going to impose some arbitrary ordering on the input of the programs: $(\mathsf{op}, ct_0, ct_1)$. If we think of the input as a sequence of bits, then this can just be the normal ordering of the bits. We now define a new program R which will also include

```
$R^{n-1,t}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}(\mathsf{op}, ct_0, ct_1)$ :

    $r = \mathsf{PRF}(\mathcal{K}_{n-1}, (\mathsf{op}, ct_0, ct_1))$
    If $(\mathsf{op}, ct_0, ct_1) < t$
        Outputs $\mathsf{Enc}(\mathsf{ek}_n, 0; r)$
    Else
        $m_b = \mathsf{Dec}(\mathsf{dk}_{n-1}, ct_b)$
        $m = m_0 \ \mathsf{op} \ m_1$
        Outputs $\mathsf{Enc}(\mathsf{ek}_n, m; r)$
```

$R^{n-1,t}$ outputs encryption of 0 for all inputs below the threshold t, and computes the resulting ciphertext correctly for the rest of the inputs. Notice that for $t = 0$ (minimum value in the ordering), then $R^{n-1,t}$ is equivalent to P^{n-1}. If the threshold is set at MAX, where MAX larger than any possible input values, then $R^{n-1,t}$ is functionally identical to Q^{n-1}. We want to show that:

$$\mathsf{Obf}\left(R^{n-1,t}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}\right) \approx_c \mathsf{Obf}\left(R^{n-1,t+1}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}\right)$$

Essentially, we will show that we can move the threshold t up or down 1 at a time and still preserve security. This will definitely change the functionality of R. However, we are going to argue that $R^{n-1,t}$ and $R^{n-1,t+1}$ are still indistinguishable. Notice that $R^{n-1,t}$ and $R^{n-1,t+1}$ differs on only one input t. On all other inputs, the two programs are identical. We can then prove indistinguishability using a series of hybrids:

- H_0: The first hybrid corresponds to $R^{n-1,t}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}$

- H_1: Let the threshold $t = (\mathsf{op}, ct_0^*, ct_1^*)$ and $\mathcal{K}_{n-1}\{t\}$ be the punctured PRF key at t.

  $r^* \leftarrow \mathsf{PRF}(\mathcal{K}_{n-1}, (\mathsf{op}, ct_0^*, ct_1^*))$
  $m^* = \mathsf{Dec}(\mathsf{dk}_{n-1}, ct_0^*)\ \mathsf{op}\ \mathsf{Dec}(\mathsf{dk}_{n-1}, ct_1^*)$
  $ct^* \leftarrow \mathsf{Enc}(\mathsf{ek}_n, m^*; r^*)$

  We now replace $R^{n-1,t}$ with a new program $R'^{\,n-1,t}$:

  ```
  $\underline{R'^{\,n-1,t}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1}\{t\},\mathsf{ek}_n,ct^*}(\mathsf{op}, ct_0, ct_1)\ :}$

      If $(\mathsf{op}, ct_0, ct_1) = t$
          Outputs $ct^*$
      Else
          $r = \mathsf{PRF}(\mathcal{K}_{n-1}\{t\}, (\mathsf{op}, ct_0, ct_1))$
          If $(\mathsf{op}, ct_0, ct_1) < t$
              Outputs $\mathsf{Enc}(\mathsf{ek}_n, 0; r)$
          Else
              $m_b = \mathsf{Dec}(\mathsf{dk}_{n-1}, ct_b)$
              $m = m_0\ \mathsf{op}\ m_1$
              Outputs $\mathsf{Enc}(\mathsf{ek}_n, m; r)$
  ```

  We can see that $R'^{\,n-1,t}$ is functionally identical to $R^{n-1,t}$ in H_0. Therefore, by iO security, we can conclude that $H_0 \approx_c H_1$.

- H_2: In this hybrid, we replace r^* with uniformly random value. We can argue that H_2 is indistinguishable from H_1 using puncturable PRF security.

- H_3: We now set $ct^* = \mathsf{Enc}(\mathsf{ek}_n, 0; r^*)$. Indistinguishability of H_3 and H_2 follows from security of the PKE scheme. r^* is not used anywhere except in the encryption of ct^*. So in H_2, we have a fresh encryption of m^*, and in H_3, we have a fresh encryption of 0. By PKE security, an adversary without the decryption key cannot tell the difference between these two ciphertexts.

- H_4: We set r^* back to be $\mathsf{PRF}(\mathcal{K}_{n-1}, (\mathsf{op}, ct_0^*, ct_1^*))$ and use punturable PRF security to argue that $H_3 \approx_c H_4$.

- H_5: We replace $R'^{\,n-1,t}$ with $R^{n-1,t+1}$. Since the two are now functionally identical (we replace the hardcoded value at t with encryption of 0 in H_3), we can conclude that $H_4 \approx_c H_5$ by iO security.

We have now shown that:

$$\mathsf{Obf}\left(R^{n-1,t}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}\right) \approx_c \mathsf{Obf}\left(R^{n-1,t+1}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}\right)$$

So if we continue the chain from $t = 0$ to $t = \mathsf{MAX}$, we will have:

$$P^{n-1} = R^{n-1,0} \approx_c R^{n-1,1} \approx_c ... \approx_c R^{n-1,\mathsf{MAX}-1} \approx_c R^{n-1,\mathsf{MAX}} = Q^{n-1}$$

So it seems that we have successfully shown that P^{n-1} is indistinguishable from Q^{n-1}. However, there is a problem in this proof. Let l be the length of the ciphertexts produced and T be the size of the set of all t values. We have $T = 2^l$, so the advantage is $2^l\mathsf{negl}(\lambda)$. However, since negl only means that the function is smaller than any inverse polynomial, $2^l\mathsf{negl}(\lambda)$ is not necessarily negligible. In addition, it is not hard to show that in any secure PKE scheme, the ciphertext size must be at least λ. Therefore, the advantage is $2^\lambda\mathsf{negl}(\lambda)$. Unfortunately, we do not know how to solve this problem. The only thing we can do is to assume that the indistinguishability between each R is very strong.

2.4 Subexponential iO

We give a new definition of iO called subexponential iO:

Definition 1 (Subexponential iO)**.** *For all functionally equivalent circuits C_0, C_1 and for all PPT adversaries $\mathcal{A}$, there exist a constant $c \in (0, 1]$ such that*

$$|Pr[\mathcal{A}(\mathsf{Obf}(C_0, \lambda)] = 1] - Pr[\mathcal{A}(\mathsf{Obf}(C_1, \lambda)] = 1]| < \frac{1}{2^{\lambda^c}}$$

This is a stronger definition of iO. We do not know that this is possible. However, it turns out that for iO (and other cryptographic primitives), the best known attacks have had this kind of probability. So it is plausible to assume that there is no kind of attack with higher success probability.

If we use subexponentially strong iO and PRF, no attacker will be able to distinguish between H_0 and H_5 with probability better than $2^l\left(\frac{5}{2^{(\lambda_n)^c}}\right)$ for level n, where λ_n is the security parameter for the level. To make the value $2^l\left(\frac{5}{2^{(\lambda_n)^c}}\right)$ small, we are going to set $\lambda_n >> l^{\frac{1}{c}} \approx (\lambda_{n-1})^{\frac{1}{c}}$ (λ_{n-1} is the security parameter of the previous level). We are raising the security parameter for the next level to be bigger than the previous level. However, this means that λ_n is exponentially big in n, which means that our obfuscated programs and ciphertexts have to also be exponentially big. We can fix this problem by introducing another primitive.

2.5 Lossy Encryption

Until now, we have not use anything more than iO and OWF. As mentioned before, FHE is unlikely to follows from just iO and OWF. Therefore, we are going to need a new primitive called lossy encryption.

Definition 2 (Lossy Encryption)**.** *A lossy encryption scheme consists of four algorithms:*

- $\mathsf{Gen}(\lambda) \to (\mathsf{dk}, \mathsf{ek})$
- $\mathsf{Enc}(\mathsf{ek}, m) \to ct$
- $\mathsf{Dec}(\mathsf{dk}, ct) \to m$
- $\mathsf{Gen}_{\mathsf{Lossy}}(\lambda) \to \mathsf{ek}$

There is a strong requirement that the distribution of the encryption of 0 *and* 1 *under* $\mathsf{ek} \leftarrow \mathsf{Gen}_{\mathsf{Lossy}}(\lambda)$ *must be identical. For all* $\mathsf{ek} \leftarrow \mathsf{Gen}_{\mathsf{Lossy}}(\lambda)$ *and ciphertext ct:*

$$Pr[\mathsf{Enc}(\mathsf{ek}, 0) = ct] = Pr[\mathsf{Enc}(\mathsf{ek}, 1) = ct]$$

Clearly this cannot be the case for ek generated with dk from $\mathsf{Gen}(\lambda)$, since we will not be able to decrypt otherwise. Basically, $\mathsf{ek} \leftarrow \mathsf{Gen}_{\mathsf{Lossy}}$ is a lossy key since it loses all information about the message. For security we requires that:

$$\mathsf{ek} \leftarrow \mathsf{Gen}_{\mathsf{Lossy}}(\lambda) \approx_c \mathsf{ek} : (\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{Gen}(\lambda)$$

We can build lossy encryption from Decisional Diffie-Hellman (DDH), factoring, and learning with errors (LWE) assumptions (though the distribution of ciphertext for 0 and 1 under lossy key are only statistically indistinguishable, not identical, in the case of LWE).

We now go back to the proof and, before we switch from P to R, replace the encryption key with a lossy key. So $P^{n-1}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n}$ will be replaced with:

```
$P^{n-1,\mathsf{Lossy}}_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}^{\mathsf{Lossy}}_n}$ :

    $m_b = \mathsf{Dec}(\mathsf{dk}_{n-1}, ct_b)$
    $m = m_0 \ \mathsf{op} \ m_1$
    $r = \mathsf{PRF}(\mathcal{K}_{n-1}, (\mathsf{op}, ct_0, ct_1))$
    Outputs $\mathsf{Enc}(\mathsf{ek}^{\mathsf{Lossy}}_n, m; r)$
```

We can make this change because P^{n-1} does not include the corresponding decryption key for ek_n. Now we can repeat the sequence of hybrids, except that we do not need to invoke PKE security between H_2 and H_3 since the encryption key has been replaced with a lossy key, which has identical distribution for encryption of 0 and 1. We no longer invokes security of the encryption scheme for every step of the hybrid, only from P^{n-1} to $P^{n-1,\mathsf{Lossy}}$. At a high level, we have separated the dependency between the security parameter and the input size. Let λ_0 be the security parameter of the encryption scheme. We will now set security of iO and puncturable PRF to be $\lambda_1 >> l^{\frac{1}{c}} \approx (\lambda_0)^{\frac{1}{c}}$.

2.6 The Rest of the Proof

We have now proved security of P^{n-1}. From here, we can prove the security of P^{n-2}, because once we have switched to Q^{n-1}, we no longer have dk_{n-1}. We can then repeat the process to replace the program at each level with Q. Once we get to level 0, we can simply prove security using PKE security.

3 From Level FHE to Full FHE

Note that we have already known how to build leveled FHE under the LWE assumption before. However, we do not know how to turn this into the full FHE scheme we wanted. In both leveled FHE constructions from LWE and iO, the number of levels are bounded. However, it turns out that we can turn the level FHE construction from iO into a full FHE scheme with subexponential hardness assumption. Briefly, the instead obfuscate a program $\overline{P}$ which takes in a level n and outputs the obfuscated program P^n for that level.

```
$\underline{\overline{P}(n) :}$

    Outputs $\mathsf{Obf}(P^n_{\mathsf{dk}_{n-1},\mathcal{K}_{n-1},\mathsf{ek}_n})$
```

Here $\mathsf{dk}_{n-1}, \mathcal{K}_{n-1}$, and ek_n are generated by a PRF. At the end, we have a program of fix size that we can use to generate $\mathsf{Obf}(P^n)$ for a new level by feeding it n.

References

[AS'15] Asharov, G. and Segev, G. Limits on the power of indistinguishability obfuscation and functional encryption. *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 191- 209, 2015. 1, 2

[IKO'05] Ishai, Y., Kushilevitz, E., and Ostrovsky, R. Sufficient conditions for collision-resistant hashing. *Theory of Cryptography Conference*, 445- 456, 2005. 1