CS6115: Structure versus Hardness in Cryptography Sep 09, 2021

Lecture 12 : Deniable Encryption from iO

Lecturer: *Shweta Agrawal* Scribe: *Rishi Garg*

1. Introduction

In the previous lecture, we discussed three types of pseudo-random functions (PRF) to be used for encryption schemes. These were namely "Puncturable PRF (PPRF)", "an Injective PPRF", and "an Extracting PPRF".

In this lecture, we are going to construct the Deniable Encryption Scheme that we talked about in the previous lecture and also prove the security.

2. Notations

(a) Pseudo-Random Generator (PRG): PRG is a function that takes as input a key of length n randomly generated bits and gives an output of length $2n$ pseudo-random bits.

3. Deniable Encryption

Following are the pre-requisites for Deniable Encryption:

- A regular Public Key Encryption Scheme
- Assumption that Enc_{PKE} accepts 1 bit message, a randomness of length l_e and outputs a cipher text of length l_c
- A pseudo-random generator (PRG) that maps $\{0,1\}^{\lambda}$ to $\{0,1\}^{2\lambda}$
- An extracting PPRF $F_1(K_1,\cdot)$ that accepts an input of length $l_1 + l_2 + 1$ and outputs a string of length l_e
- A puncturable, statistically injective, PRF $F_2(K_2,\cdot)$ that accepts an input of length $2\lambda + l_c + 1$ and outputs strings of length l_1.
- A PPRF $F_3(K_3,\cdot)$ that accepts an input of length l_1 and outputs strings of length l_2

Before, we construct the deniable encryption scheme, we consider that the encryption algorithm Enc_{DE} takes 1 bit message m and randomness $u = (u[1], u[2])$ where the lengths of $u[1]$ and $u[2]$ are $l_1 = 5\lambda + 2l_c + l_e$ and $l_2 = 2\lambda + l_c + 1$ respectively.

4. Construction of Deniable Encryption (DE) Scheme

The main idea is to encrypt using the Regular Public Key Encryption scheme and use obfuscation to get the deniability.

Remember from the previous lecture that the ***Explain*** *algorithm takes two inputs, a message 'm' and a cipher text 'ct' and outputs a randomness 'r'*

Now, the intuition for the construction is as follows:

- First, we want the *explain* algorithm to output, for any input message m and cipher text CT, a randomness r such that

$$Enc_{DN}(m, r) = CT$$

- For this, obfuscate the *explain* algorithm along with some extra (, secret) elements such that some pseudo-random decodable function/encryption along with these extra elements encrypts CT, m and outputs the desired randomness r.
- **We will use this r as a "Hidden Sparse Trigger"**
- Now, we want that the scheme to take m and r as the inputs and output CT.
  Note that the Public Key Encryption scheme, if encrypts m and r, is unlikely to output the desired CT.
- For this, obfuscate the Enc_{DE} program along with the (secret) key (let's say, D_K) of the above-used pseudo-random decodable encryption.
- Now, on inputs m and r to this obfuscated program, perform the following operations:
  - Using the key D_K, check whether r is an encoding for some m_0 and CT_0.
  - If r is an encoding for some m_0 and CT_0:
    * Check whether $m_0 = m$.
    * Output CT_0 if $m_0 = m$.
    * Output Enc_{PKE} otherwise.
  - If r is not an encoding for any m_0 and CT_0:
    * Output Enc_{PKE}.

4.1 Deniable Encryption (DE) Using iO

The public key builds the iO of the two programs as follows:

- **iO of the *Encrypt* program i.e., Enc_{DE}:**
  This has hardwired the public key PK (of the public key encryption scheme) and the three PRF's keys namely K_1, K_2, and K_3.
  The message m and the randomness $u = (u[1], u[2])$ are given as the input.
  Step-1: Check for the the hidden sparse trigger and if

$$F_3(K_3, u[1]) \oplus u[2] = (m', c', r') \text{ for proper lengths m}', \text{c}', \text{and r}',$$
$$m = m', \text{ and}$$
$$u[1] = F_2(K_2, (m', c', r'))$$

  then output $c = c'$ and the program ends.
  Step-2: If the above trigger test fails, then encrypt as usual. Consider $F_1(K_1, (m, u))$ to be x and output $c = Enc_{PKE}(PK, m, x)$.

- **iO of the *Explain* algorithm:**
  This has hardwired PRF's keys K_2 and K_3
  The message m, the cipher text c (i.e., the output of Enc_{DE}), and a new randomness $r \in \{0, 1\}^\lambda$ are given as the input.
  Step-1: Set $\alpha = F_2(K_2, (m, c, PRG(r)))$
  Step-2: Set $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
  Step-3: Output $e = (\alpha, \beta)$

4.2 Correctness of DE using iO

To check: If the message m' along with the output $e = (\alpha, \beta)$ of the obfuscated explain program is input to the program, the output is c.

We have

$$e = (\alpha, \beta) = (F_2(K_2, (m', c, PRG(r))), F_3(K_3, \alpha) \oplus (m', c, PRG(r)))$$

Checking the sparse trigger, with the above usual notations, we have,

$$\begin{aligned} F_3(K_3, u[1]) \oplus u[2] &= F_3(K_3, \alpha) \oplus \beta \\ &= F_3(K_3, \alpha) \oplus F_3(K_3, \alpha) \oplus (m', c, PRG(r)) \\ &= (m', c, PRG(r)) \end{aligned}$$

Since the input message was m', we now have $m' = m'$ and $u[1] = \alpha = F_2(K_2, (m', c, PRG(r)))$.
Note that all the conditions for the hidden sparse trigger are followed, therefore, we get the output to be $c' = c$ as required.
This proves the correctness of the scheme.

4.3 Proofs of security of DE

Following are the two proofs for the security of DE:

4.3.1 IND-CPA Security

This ensures that the output cipher text of this scheme is secure, in the sense that it does not reveal anything about the message bit. Intuitively, this is true because the cipher text of this scheme is using the public key encryption and the security of PKE will imply the security of this cipher text.

Lemma: For a fixed $u[1] = \alpha$, $\exists$ at most one pair (m, β) such that the 'Sparse Trigger Condition' for input m and randomness (α, β) is satisfied.
Proof: Note that the lengths of m, c, and u are 1, l_c, and 2λ respectively. Therefore, it follows that there exist at most $2^{2\lambda + l_c + 1}$ input strings for F_2 and thus, the same number of outputs by F_2.
Note that the F_2 is injective, it follows that the output of F_2 determines the input and therefore, fixes m'.
We have a fixed $u[1] = \alpha$. The "Sparse Trigger Condition" also verifies $u[2] = F_3(K_3, u[1]) \oplus (m', c', r')$ and thus, fixes $u[2] = \beta$.
Therefore, it follows that the m' and β are fixed.
To be noted that fixing m' and $u[2]$ implies that (u_1, u_2) is highly structured. This further implies that upon sampling some $u' = (u'_1, u'_2)$ at (uniformaly) random, it satisfies the above structure only with negligible probability.
Hence, fixing $u[1] = \alpha$ implies that there exist at most one pair (m, β) for which the 'Sparse Trigger Condition' is satisfied.

Lemma: There are at most $2^{2\lambda + l_c + 1}$ values of randomness u for which the 'Sparse Trigger Condition' is satisfied.

Proof: Follows from the previous lemma as the total maximum number of possibilities for $u[1]$ are $2^{2\lambda+l_c+1}$ and from the previous lemma, for every fixed $u[1]$, there exist at most one pair (m, β) implying that there exist at most $2^{2\lambda+l_c+1}$ number of $u = (u[1], u[2])$.
Hence, the maximum number of randomness for which the 'Sparse Trigger Condition' is satisfied is $2^{2\lambda+l_c+1}$.

In the below setup by the challenger, if the attacker guess the bit g with a non-negligible advantage over a random guessing, we say that the scheme is not secure.
Setup by the challenger:

1 Choose K_1, K_2, and K_3 randomly.
2 Take $(PK, SK) \leftarrow PKE.setup$
3 Write $P_{enc} = iO(Encrypt)$ and $P_{exp} = iO(Explain)$
4 Take a random $g \in \{0, 1\}$ and choose randomness $u = (u[1], u[2])$
5 Set $c = P_{enc}(g, u)$ and output c.

In the next lecture, we will use 'Hybrids' to complete this proof.

4.3.2 Explainability

This ensures that the actual randomness and the fake (pseudo) randomness are indistinguishable.
We will see this in detail in the further lectures.

References

[1] CS6115 Lecture 3. *URL: http://cse.iitm.ac.in/ shwetaag/6115/Lec3.pdf.*

[2] Amit Sahai and Brent Waters. "How to Use Indistinguishability Obfuscation: Deniable Encryption, and More". *In: ACM Symposium on Theory of Computing (STOC) (2014)*

[3] CS6115 Lecture 11. *URL: http://cse.iitm.ac.in/ shwetaag/6115/Lec11.pdf.*