CS6115: Structure versus Hardness in Cryptography Sep 16, 2021

Lecture 13 : Deniable Encryption from iO

Lecturer: *Shweta Agrawal* Scribe: *Ratnakar M*

1 Introduction

In the last lecture, we saw the Deniable Encryption scheme presented in [**SW13**] and the idea behind the proof of its security. In today's lecture, we will finish the proof of IND-CPA Security of the scheme and see the high level overview of the proof strategy for the indistinguishability of explanation.

1.1 Review of the Deniable Encryption scheme

The Encrypt program takes as input, a message bit m and randomness $u = (u[1], u[2])$ where
$|u[1]| = l_1 = 5\lambda + 2l_c + l_e$ and $|u[2]| = l_2 = 5\lambda + l_c + 1$.
The Explain program takes as input, a message bit m, ciphertext c of length l_c, and randomness $r \in \{0,1\}^\lambda$ where λ is the security parameter.
The scheme makes use of the following ingredients:

- A public key encryption scheme $Encrypt_{PKE}(PK, \cdot)$ that accepts a message bit m and randomness of length l_e and outputs a ciphertext of length l_c
- A PRG that maps $\{0,1\}^\lambda$ to $\{0,1\}^{2\lambda}$
- A puncturable extracting PRF $F_1(K_1, \cdot)$ 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 puncturable PRF $F_3(K_3, \cdot)$ that accepts an input of length l_1 and outputs strings of length l_2

The public key builds the indistinguishability obfuscation of the Encrypt and Explain programs below:

```
                              Encrypt

Hardwired: Public Key $PK$, PRF Keys $K_1, K_2$, and $K_3$
Input: Message $m$, randomness $u = (u[1], u[2])$
1. If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
2. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$
```

Explain

Hardwired: PRF Keys K_2, K_3
Input: Message m, ciphertext c, randomness $r \in \{0,1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$

2 IND-CPA Security

Proof Idea: To prove IND-CPA security, we start with the original IND-CPA experiment and proceed through a sequence of hybrid experiments, each of which is indistinguishable from the previous one. If we can argue that the output of the ”Explain” program is indistinguishable from a random string, the game will reduce to the IND-CPA security game of $Encrypt_{PKE}$ and we are done. However we can't directly argue that since the obfuscation of Explain which has the PRF keys hardwired into it is public. We are working with indistinguishability obfuscation and not a virtual black-box obfuscation, and therefore it is possible that the hardwired keys are revealed. We get around this by using the technique of punctured programs wherein we show, using the properties of punctured PRFs, that the obfuscation of the program punctured at key points is indistinguishable from the obfuscation of the original program.

IND-CPA Security Game: The adversary must be able to guess the bit g with probability $p = (\frac{1}{2} + \epsilon)$ for some non-negligible ϵ

1. Sample keys $K_1, K_2, andK_3$ at random.
2. Let $(PK, SK) \leftarrow Setup_{PKE}(1^\lambda)$.
3. Let P_{enc} = $iO(Encrypt)$. Let $P_{explain}$ = $iO(Explain)$.
4. Sample $g \in \{0,1\}$ at random .
5. Sample $u \in \{0,1\}^{l_1+l_2}$ at random.
6. Set $c = P_{enc}(g; u)$.
7. Output (u, c)

Hybrid 1:
1. Sample keys K_1, K_2, K_3 at random.
2. Let $(PK, SK) \leftarrow Setup_{PKE}(1^\lambda)$.
3. Let P_{enc} = $iO(Encrypt)$. Let $P_{explain}$ = $iO(Explain)$.
4. Sample at random $g \in \{0,1\}$.
5. Sample at random $u \in \{0,1\}^{l_1+l_2}$ at random.
6. Set $c = Encrypt_{PKE}(PK, g; x)$ where $x = F_1(K_1, (g, u))$.
7. Output (u, c)

We have replaced Step 6 - ”Set $c = P_{enc}(g; u)$.” with ”Set $c = Encrypt_{PKE}(PK, g; x)$ where $x = F_1(K_1, (g, u))$”

Effectively, we have removed the check for the hidden sparse trigger in the original Encrypt program. From the Lemma we proved in the last lecture [12], we know that with overwhelming probability, the Step 1 check will not be satisfied when we use true randomness u and therefore, with overwhelming probability Hybrid 1 and the original game are indistinguishable.

Hybrid 2:
1. Sample keys $K_1, K_2, andK_3$ at random.

2. Let $(PK, SK) \leftarrow Setup_{PKE}(1^\lambda)$.
3. Let P_{enc} = $iO(Encrypt)$. Let $P_{explain}$ = $iO(Explain)$.
4. Sample $g \in \{0, 1\}$ at random.
5. Set $c = Encrypt_{PKE}(PK, g; x)$ where x is sampled at random from $\{0, 1\}^{l_e}$.
6. Output (u, c)

We have removed step 5 and modified Step 6 by replacing it with "Set $c = Encrypt_{PKE}(PK, g; x)$ where x is sampled randomly from $\{0, 1\}^{l_e}$."
F_1 is an extracting PPRF, therefore $F_1(K_1, (g, u))$ cannot be distinguished from random and by extension, Hybrids 1 and 2 are indistinguishable

Now, observe that Hybrid 2 is the same as the indistinguishability game for $Encrypt_{PKE}$ which we know is secure. Therefore, our original deniable encryption scheme is IND-CPA secure.

3 Indistinguishability of Explainability

Like the IND-CPA security proof, this proof also follows a sequence of hybrid experiments. For the rest of the lecture, we will see the sequence of hybrid experiments used to argue indistinguishability of explainability of the scheme. We will argue the correctness more formally in the next lecture.

Explainability Game: As in the IND-CPA game, the adversary wins if (s)he can guess g with non-negligible advantage over $\frac{1}{2}$.
1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample u^*, r^* at random.
4. Set $x^* = F_1(K_1, (m^*, u^*))$, $c^* = Encrypt(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, PRG(r^*)))$, $\beta^* = F_3(K_3, \alpha^*) \oplus (m^*, c^*, PRG(r^*))$, and $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0, 1\}$ at random .
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else output $(P_{enc}, P_{explain}, e^*, c^*)$.

Encrypt

Hardwired: Public Key PK, PRF Keys K_1, K_2, and K_3
Input: Message m, randomness $u = (u[1], u[2])$
1. If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
2. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$

Explain

```
Hardwired: PRF Keys $K_2, K_3$
Input: Message $m$, ciphertext $c$, and randomness $r \in \{0,1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$
```

Hybrid 0:
1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample u^*, r^* at random.
4. Set $x^* = F_1(K_1, (m^*, u^*))$, $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, PRG(r^*)))$, $\beta^* = F_3(K_3, \alpha^*) \oplus (m^*, c^*, PRG(r^*))$, and $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0, 1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else, output $(P_{enc}, P_{explain}, e^*, c^*)$.

Encrypt

Hardwired: Public Key PK, PRF Keys K_1, K_2, and K_3
Input: Message m, randomness $u = (u[1], u[2])$
1. If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
2. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$

Explain

Hardwired: PRF Keys K_2, K_3
Input: Message m, ciphertext c, and randomness $r \in \{0, 1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$

The only change from the original experiment here is the replacement of $Encrypt$ with $Encrypt_{PKE}$. Using the same argument we employed in the IND-CPA security proof, we can conclude that with overwhelming property, Step 1 check fails and therefore, $Encrypt_{PKE}$ and $Encrypt$ are indistinguishable.

Hybrid 1

1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample u^*, r^* at random.
4. Set $x^* = F_1(K_1, (m^*, u^*))$, $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, PRG(r^*)))$, $\beta^* = F_3(K_3, \alpha^*) \oplus (m^*, c^*, PRG(r^*))$, and $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0, 1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else, output $(P_{enc}, P_{explain}, e^*, c^*)$.

Encrypt

```
Hardwired: $m^*, u^*, e^*, c^*$, Public Key $PK$, PRF Keys $K_1(\{(m^*, u^*), (m^*, e^*)\})$, $K_2$, and $K_3$
Input: Message $m$, randomness $u = (u[1], u[2])$
1. If $(m, u) = (m^*, e^*)$ or $(m, u) = (m^*, u^*)$, output $c^*$.
2. If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
3. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$
```

Explain

```
Hardwired: PRF Keys $K_2, K_3$
Input: Message $m$, ciphertext $c$, and randomness $r \in \{0, 1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$
```

In this hybrid, we hardwire m^*, u^*, e^*, c^* in addition to the public and the PRF keys. We add another check to the encrypt program where we output c^* when $(m, u) = (m^*, e^*)$ or (m^*, u^*). We can now safely puncture the PRF key K_1 at points (m^*, e^*) and (m^*, u^*) without affecting the functionality of Encrypt.

Hybrid 2:
1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample u^*, r^* at random.
4. Sample $x^* \in \{0,1\}^{l_e}$. Set $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, PRG(r^*)))$, $\beta^* = F_3(K_3, \alpha^*) \oplus (m^*, c^*, PRG(r^*))$, and $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0,1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else, output $(P_{enc}, P_{explain}, e^*, c^*)$.

Encrypt

```
Hardwired: $m^*, u^*, e^*, c^*$ , Public Key $PK$, PRF Keys $K_1(\{(m^*, u^*), (m^*, e^*)\}), K_2$, and $K_3$
Input: Message $m$, randomness $u = (u[1], u[2])$
1. If $(m, u) = (m^*, e^*)$ or $(m, u) = (m^*, u^*)$, output $c^*$.
2. If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
3. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$
```

Explain

```
Hardwired: PRF Keys $K_2, K_3$
Input: Message $m$, ciphertext $c$, and randomness $r \in \{0,1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$
```

The only change in this experiment is that x is sampled at random from $\{0,1\}^{l_e}$ rather than the output of $F_1(K_1, (m^*, u^*))$.

Hybrid 3:
1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample $u^*, \tilde{r}$ at random.
4. Sample $x^* \in \{0,1\}^{l_e}$. Set $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, \tilde{r}))$, $\beta^* = F_3(K_3, \alpha^*) \oplus (m^*, c^*, \tilde{r})$, and $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0,1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else output $(P_{enc}, P_{explain}, e^*, c^*)$.

```
Encrypt

Hardwired: $m^*, u^*, e^*, c^*$ , Public Key $PK$, PRF Keys $K_1(\{(m^*, u^*), (m^*, e^*)\}), K_2$, and $K_3$
Input: Message $m$, randomness $u = (u[1], u[2])$
1. If $(m, u) = (m^*, e^*)$ or $(m, u) = (m^*, u^*)$, output $c^*$.
2. If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$ and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
3. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$
```

```
Explain

Hardwired: PRF Keys $K_2, K_3$
Input: Message $m$, ciphertext $c$, and randomness $r \in \{0,1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$.
Output $e = (\alpha, \beta)$
```

In this hybrid, we replace the output of the PRG with a random value $\tilde{r} \in \{0,1\}^{2\lambda}$.

Hybrid 4:
1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample $u^*, \tilde{r}$ at random.
4. Sample $x^* \in \{0,1\}^{l_e}$. Set $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, \tilde{r}))$, $\beta^* = F_3(K_3, \alpha^*) \oplus (m^*, c^*, \tilde{r})$, and $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0,1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else, output $(P_{enc}, P_{explain}, e^*, c^*)$.

Encrypt

Hardwired: m^*, u^*, e^*, c^* , Public Key PK, PRF Keys $K_1(\{(m^*, u^*), (m^*, e^*)\}), K_2$, and $K_3(\{u^*[1], e^*[1]\})$
Input: Message m, randomness $u = (u[1], u[2])$
1. If $(m, u) = (m^*, e^*)$ or $(m, u) = (m^*, u^*)$, output c^*.
2. If $u[1] = e^*[1]$ or $u[1] = u^*[1]$, go to Step 3.
If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
3. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$

Explain

Hardwired: PRF Keys $K_2, K_3(\{u^*[1], e^*[1]\})$
Input: Message m, ciphertext c, and randomness $r \in \{0,1\}^{\lambda}$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$

We add a check in Step 2 of the Encrypt program to skip the step "if $u[1] = e^*[1]$ or $u[1] = u^*[1]$". We then puncture K_3 on those points.

Hybrid 5:

1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample $u^*, \tilde{r}$ at random.
4. Sample $x^* \in \{0,1\}^{l_e}$. Set $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, \tilde{r}))$, $\beta^* =$ random, $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0,1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else, output $(P_{enc}, P_{explain}, e^*, c^*)$.

Encrypt

```
Hardwired: $m^*, u^*, e^*, c^*$ , Public Key $PK$, PRF Keys $K_1(\{(m^*, u^*), (m^*, e^*)\}), K_2$, and $K_3(\{u^*[1], e^*[1]\})$
Input: Message $m$, randomness $u = (u[1], u[2])$
1. If $(m, u) = (m^*, e^*)$ or $(m, u) = (m^*, u^*)$, output $c^*$.
2. If $u[1] = e^*[1]$ or $u[1] = u^*[1]$, go to Step 3.
If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
3. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$
```

Explain

```
Hardwired: PRF Keys $K_2, K_3(\{u^*[1], e^*[1]\})$
Input: Message $m$, ciphertext $c$, and randomness $r \in \{0,1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$
```

In this hybrid, we set β^* to be random instead of $F_3(K_3, \alpha^*) \oplus (m^*, c^*, \tilde{r})$.

Hybrid 6:
1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample $u^*, \tilde{r}$ at random.
4. Sample $x^* \in \{0,1\}^{l_e}$. Set $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set $\alpha^* = F_2(K_2, (m^*, c^*, \tilde{r}))$, β^* = random, $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0,1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else, output $(P_{enc}, P_{explain}, e^*, c^*)$.

```
Encrypt

Hardwired: $m^*, u^*, e^*, c^*, \tilde{r}$ , Public Key $PK$, PRF Keys $K_1(\{(m^*, u^*), (m^*, e^*)\}), K_2(\{(m^*, c^*, \tilde{r})\})$, and $K_3(\{u^*[1], e^*[1]\})$
Input: Message $m$, randomness $u = (u[1], u[2])$
1. If $(m, u) = (m^*, e^*)$ or $(m, u) = (m^*, u^*)$, output $c^*$.
2. If $u[1] = e^*[1]$ or $u[1] = u^*[1]$, go to Step 3.
If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, $(m', c', r') \neq (m^*, c^*, \tilde{r})$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
3. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$
```

```
Explain

Hardwired: PRF Keys $K_2(\{(m^*, c^*, \tilde{r})\}), K_3(\{u^*[1], e^*[1]\})$
Input: Message $m$, ciphertext $c$, and randomness $r \in \{0,1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$
```

In this hybrid, we modify the Step 2 check of Encrypt to skip this step if the decrypted $(m', c', r') = (m^*, c^*, \tilde{r})$. We can then puncture the key K_2 at $(m^*, c^*, \tilde{r})$.

Hybrid 7:
1. Sample K_1, K_2, and K_3 at random.
2. Let $(PK, SK) = Setup_{PKE}$
3. Sample $u^*, \tilde{r}$ at random.
4. Sample $x^* \in \{0,1\}^{l_e}$. Set $c^* = Encrypt_{PKE}(PK, m^*; x^*)$.
5. Set α^*, β^* to be random, $e^* = (\alpha^*, \beta^*)$.
6. Let $P_{enc} = iO(Encrypt)$. Let $Pexplain = iO(Explain)$.
7. Select $g \in \{0,1\}$ at random.
8. If g = 0, output $(P_{enc}, P_{explain}, u^*, c^*)$. Else, output $(P_{enc}, P_{explain}, e^*, c^*)$.

Encrypt

```
Hardwired: $m^*, u^*, e^*, c^*, \tilde{r}$ , Public Key $PK$, PRF Keys $K_1(\{(m^*, u^*), (m^*, e^*)\}), K_2(\{(m^*, c^*, \tilde{r})\})$, and $K_3(\{u^*[1], e^*[1]\})$
Input: Message $m$, randomness $u = (u[1], u[2])$
1. If $(m, u) = (m^*, e^*)$ or $(m, u) = (m^*, u^*)$, output $c^*$.
2. If $u[1] = e^*[1]$ or $u[1] = u^*[1]$, go to Step 3.
If $F_3(K_3, u[1]) \oplus u[2] = (m', c', r')$, $m = m'$, $(m', c', r') \neq (m^*, c^*, \tilde{r})$, and $u[1] = F_2(K_2, (m', c', r'))$, then output $c = c'$
3. Else, output $c = Encrypt_{PKE}(PK, m; x)$ where $x = F_1(K_1, (m, u))$
```

Explain

```
Hardwired: PRF Keys $K_2(\{(m^*, c^*, \tilde{r})\}), K_3(\{u^*[1], e^*[1]\})$
Input: Message $m$, ciphertext $c$, and randomness $r \in \{0,1\}^\lambda$
1. Set $\alpha = F_2(K_2, (m, c, PRG(r)))$, $\beta = F_3(K_3, \alpha) \oplus (m, c, PRG(r))$
Output $e = (\alpha, \beta)$
```

In the final hybrid, we replace α^* with random. We can directly argue indistinguishability since both e^* and u^* are now random strings and therefore, their distributions are indistinguishable.

References

[1] CS6115 Lecture 12. *URL: http://www.cse.iitm.ac.in/ shwetaag/6115/Lec12.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)*