Homework 8

Erik Brakke

November 12, 2015

Collaborators:

Answer 1

(a) First, let's assume we have an experiment where D sends two messages m_0, m_1 and gets back an encryption of one of these messages where b is chosen at random. b is the message that is encrypted.
We know that Pr[D outputs b$|exp-b] = 1/2 + neg$
That is, D does not have any advantage by seeing (y, d) in choosing b
Let's assume this is not the case
Given a y_b, d_b from m_b where m_b is one of m_0, m_1 chosen by D, Pr[D outputs b$|exp-b] = 1/2 + \epsilon$ Where ϵ is some non-negligible probability
This means that D is able to distinguish (y_0, d_0) from (y_1, d_1) with some non-negligible advantage ϵ
Let the event E be the event that D queries H on $r = f^{-1}(y)$ If $\bar{E}$, then that means D never asked for $H(r)$.
Because d_0 is just $p \oplus m$, and p is $H(r)$, D needs to know $H(r)$ to have any kind of advantage on distinguishing
Thus, if $\bar{E}$, then D does not have any advantage on distinguishing regardless of b
Because he has no advantage, then Pr[Guessing correctly$|\bar{E}] = 1/2$ If we let S be the event that D guesses b correctly, we have $\Pr[S] = \Pr[S|E]\Pr[E] + \Pr[S|\bar{E}]\Pr[\bar{E}]$
We know that $Pr[S] = 1/2 + \epsilon$
$1/2 + \epsilon = \Pr[S|E]\Pr[E] + 1/2(1 - \Pr[E])$
$1/2 + \epsilon = \Pr[E] + 1/2 - \Pr[E]/2$ (Because if E, then D will distinguish)
$2 * \epsilon \leq \Pr[E]$

(b) Let's use D to build a function R that can reverse the OWF f
We will give R the inputs PK, y and will expect that R can use D to output $f^{-1}(y)$ with non-negligible probability
F will then ask q_{hash} queries to H and q_{enc} queries to $Enc(m)$ and distinguish between them
We can assume that F will query H for $H(a_j)$ before asking for $Enc(m_j)$ because without the hash information, he cannot tell anything about the response he will get
We can also assume that before generating m_0, m_1 F will ask H for the hash of two number $r_0, r_1 \in D_i$ because F will need this info to distinguish the output of Enc
R must create $H(a_j)$ as such:
If a_j has been queried before, then output the same value s_j. If not then output a random $s \in D_i$ and store the value

R must create $Enc(m_j)$ in the following way:
Find a_j, s_j and output $(f_i(a_j), s_j \oplus m_j)$ where f is the OWF
When D is ready to guess, he will send $Enc(m_0)$ and $Enc(m_1)$
R will return $(y, H(r_0) \oplus m_0)$
If D returns 0, then R can output r_0 as the inverse of y, else abort
This gives R a probability of ϵ chance of inverting y
This is because we know that there is a non-negligible chance that D has to ask $H(r)$ where $r = f^{-1}(y)$
Therefore, if we choose m_0 to be the message we encrypt and send y to D along with $H(r)\oplus m_0$, and F says that this message is m_0, then that means that D was able to distinguish this because r was chosen correctly.
We know that D has a probability of ϵ of asking for r by the previous part
And now we have a 1/2 chance of choosing the correct m to encrypt. Therefore R has an $\epsilon/2$ chance of reverting f, which is non-negligible
This is a contradiction, because we assumed f was a OWF, therefore this encryption scheme is polynomially secure

□

Answer 2

Given a random oracle H we can construct another random oracle H' to hash to arbitrary lengths by doing the following:
$h_1 = H(n||s)$
$h_2 = H(n||h_1)$
...
$h_i = H(n||h_{i-1})$
Then you have $h = h_1||h_2||...||h_i$ and output n bits $h_1...h_n$
In this case $i = \lceil n/l \rceil$
The outputs will always be independent of eachother because either s will be different or n will be different between two pains (n, s)
Because we take $H(n||s)$ first, we are guarenteed that this will be unique (i.e. no collisions with greater than negligible probability)
This means that $h_2...h_i$ will also be unique, so concatonating them together will give us indepent outputs

---

References

None