CS 427, Assignment 3

Cody Malick
malickc@oregonstate.edu

February 10, 2017

1

a

It is fairly easy to show that this function is insecure by using a chosen plaintex attack. Here is a quick example. Suppose you have some calling function $\mathcal{F}$, where H is either the function we're attacking or a function that is a secure PRP, that returns a uniformly distributed encrypted value:

$$\begin{aligned}
&\mathcal{F}():\\
&\qquad m_1 \leftarrow \{0,1\}^\lambda\\
&\qquad m_2 \leftarrow \{0,1\}^\lambda\\
&\qquad k := Keygen()\\
&\qquad //Split\ the\ resulting\ cipher\ text\ into\ two\ parts\\
&\qquad c_1, c_2 := H(k, m_1||m_2)\\
&\qquad if(\ m_2 == c_1)\{\\
&\qquad\quad return\ true\\
&\qquad \}\\
&\qquad return\ false
\end{aligned}$$

Examining this attack, we can see that $\mathcal{F}$ will return true with a probability of 1 for the function H we are attacking, and will almost never return true for a properly ecrypted value. The PRF provides no encryption for the first half of the plain text. If $H()$ were secure, we should not be able to differentiate between $H()$ and some secure PRP.

b

While this function is slightly more secure than the last, it can also be attacked with a chosen plaintext attack, but we have to be a little smarter about how we extract information. We can gain information about the plaintext by setting the first half of two messages to m_1, and the later half of either to any other string. In this case, m_2 and m_3.

2

To show that F' is a secure PRF, we have to show that the output, which in this case would be uniformly random, is indistinguishable from a secure PRF. We can show this by transforming F' to look have output like some other function, F. Starting with our base definition of F':

$$\begin{aligned}
&F'(k, r):\\
&\qquad\qquad return\ G(F(k, r))
\end{aligned}$$

We first can expand all the operations:

$$
\begin{aligned}
&F'(k,r): \\
&\qquad f \;:=\; F(k,r) \\
&\qquad g \;:=\; G(f) \\
&\qquad return\ g
\end{aligned}
$$

This does not change the calling function as we have not changed the fundamental behavior of the function. Next, we can pull F out into a calling function $\mathcal{F}$, and substitute $\mathcal{F}$ for it's resulting output, a uniformly random distribution:

$$
\begin{aligned}
&F'(k,r): \\
&\qquad f \;:=\; \mathcal{F}(k,r) \\
&\qquad g \;:=\; G(f) \\
&\qquad return\ g
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{F}(k,r): \\
&\qquad returnx \leftarrow \{0,1\}^{\lambda}
\end{aligned}
$$

Substitute the results of the new function $\mathcal{F}$:

$$
\begin{aligned}
&F'(k,r): \\
&\qquad f \;\leftarrow\; \{0,1\}^{\lambda} \\
&\qquad g \;:=\; G(f) \\
&\qquad return\ g
\end{aligned}
$$

We can do this, because we have assumed that F is a secure PRF, which outputs a random number picked from a uniformly random distribution. Next, we can perform the same proceedure with G, but we must show why it works for results of length 2λ. G is pulled out into function $\mathcal{G}$, and then is substitued:

$$
\begin{aligned}
&F'(k,r): \\
&\qquad f \;:=\; \{0,1\}^{\lambda} \\
&\qquad g \;:=\; \mathcal{G}(f) \\
&\qquad return\ g
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{G}(x): \\
&\qquad x_1 \;:=\; first\ half\ of\ x \\
&\qquad x_2 \;:=\; second\ half\ of\ x \\
&\qquad c_1 \;:=\; \mathcal{F}(x_1) \\
&\qquad c_2 \;:=\; \mathcal{F}(x_2) \\
&\qquad returnc_1||c_2 \\
&\qquad return\ g
\end{aligned}
$$

With this outline complete, we can then substitute the previous result of $\mathcal{F}$, specifically its random output:

$$
\begin{aligned}
&\mathcal{G}(x): \\
&\qquad c_1 \;\leftarrow\; \{0,1\}^{\lambda} \\
&\qquad c_2 \;\leftarrow\; \{0,1\}^{\lambda} \\
&\qquad return\ c_1||c_2
\end{aligned}
$$

This result can be rewritten as a random value of length 2λ:

$$
\begin{aligned}
&F'(k, r): \\
&\qquad f \leftarrow \{0,1\}^{\lambda} \\
&\qquad g \leftarrow \{0,1\}^{2\lambda} \\
&\qquad \textit{return } g
\end{aligned}
$$

This is the desired result, as the output is a uniformly random distribution of length 2λ. This result is indistinguishable from a secure PRF, and that completes the proof.

3

Given that F is a PRP, then it must, by definition, have an inverse, F^{-1}. The decryption algorith for this setup would then be:

$$
\begin{aligned}
&DEC(k, (x, y)): \\
&\qquad s_2 := F^{-1}(k, y) \\
&\qquad s_1 := F^{-1}(k, x) \\
&\qquad m := s_2 \oplus s_1 \\
&\qquad \textit{return } m
\end{aligned}
$$