

Handout 3: Trapdoor Permutation implies Encryption

Let's do cleanly 2 things: Show that TDP's imply encryption on one-bit messages, and then show that one-bit encryptions can be extended to encryption schemes which encrypt messages of any length. The second argument is a so-called "hybrid" argument which is a "universal tool" that can be applied to many crypto arguments.

1 Trapdoor Permutations imply Encryptions for one-bit long messages

Assume $f_i: \{0, 1\}^k \rightarrow \{0, 1\}^k$ is a family of trapdoor permutations (for each k , we have index set I_k from which we can pick trapdoor permutations f_i of security parameter k). Let B_i be a hard-core bit of f_i , i.e. one cannot predict the value of $B_i(x)$, given $y = f_i(x)$ with probability non-negligibly better than a half, for $x \leftarrow \{0, 1\}^k$.

Now we can build an encryption procedure which encrypts single bits as follows: Key Generation, on security parameter k , picks $i \in I_k$ (and hence the function f_i) as the public key, and the trapdoor t_i as the secret key (and hence the function inverse f_i^{-1}). To encrypt $m \in \{0, 1\}$, pick $x \leftarrow \{0, 1\}^k$ and outputs $(c_1, c_2) = (f_i(x), B_i(x) \oplus m)$. To decrypt, invert $x = f_i^{-1}(c_1)$ and output $m = c_2 \oplus B(x)$.

Let's show that this is an encryption secure in the sense of indistinguishability:

Theorem 1 *If $\{f_i\}$ is a family of trapdoor permutations, then the above encryption scheme is secure in the sense of indistinguishability.*

Proof: Assume a probabilistic polynomial-time adversary A who has non-negligible advantage in distinguishing encryptions of $m_0 = 0$ from encryptions of $m_1 = 1$. If A breaks the indistinguishability test for encryption, then without loss of generality, we can assume that

$$\Pr[A(Enc(0)) = 1] = p$$

and

$$\Pr[A(Enc(1)) = 1] = p - \epsilon(k)$$

for some p , where $\epsilon(k)$ is not a negligible function of k , where the probability ranges over the coins of Enc and A . (We actually don't need that $p > 1/2$: It can be any value in $[0, 1]$.)

By plugging in our encryption procedure, it follows that

$$\Pr[A(f(x), B(x)) = 1] = p$$

and

$$\Pr[A(f(x), \neg B(x)) = 1] = p - \epsilon(k)$$

Let's construct A' which given $y = f(x)$ predicts the bit $B(x)$ with probability $1/2 + \epsilon(k)$ as follows: On input y , A' picks bit $b \leftarrow \{0, 1\}$, runs $A(y, b)$, and outputs b if A says 1, and $\neg b$ otherwise. In other words, we take the output 1 of A as a vote that b was a correctly guessed hardcore bit, and 0 as a vote that it was incorrect (and hence that $\neg b$ is the correct bit).

What's the probability that A' is correct? Assume the worst case that $Pr[B(x) = 1] = 1/2$, for random x . Then we have:

- If A' guesses b correctly (which happens with probability $1/2$), then A' outputs the correct value for $B(x)$ if it outputs b , which happens if $A(f(x), b)$ outputs 1. But given that $b = B(x)$, running A on $(f(x), b)$ means running A on $(f(x), B(x))$. Therefore, given that $b = B(x)$, the probability that A' outputs b is the probability $Pr[A(f(x), B(x)) = 1] = p$.
- If A' guesses b incorrectly (which happens with probability $1/2$), then A' outputs the correct value for $B(x)$ if it outputs $\neg b$, which happens if $A(f(x), b)$ outputs 0. But given that $b = \neg B(x)$, running A on $(f(x), b)$ means running A on $(f(x), \neg B(x))$. Therefore, given that $b = \neg B(x)$, the probability that A' outputs $\neg b$ is the probability $Pr[A(f(x), \neg B(x)) = 0] = 1 - Pr[A(f(x), \neg B(x)) = 1] = 1 - (p - \epsilon(k))$.

Therefore, our total probability of correctness is

$$Pr[A'(f(x)) = B(x)] = 1/2 * p + 1/2 * (1 - (p - \epsilon(k))) = 1/2 + \epsilon(k)/2$$

□

2 Encryptions of one-bit messages imply encryptions of messages of any length: A “Hybrid” Argument

OK, so encrypting one-bit long messages doesn't seem so impressive. But we can extend such encryptions to handle messages of any length (but polynomial in the security parameter) as follows:

Let $(KGen, Enc, Dec)$ be a one-bit encryption scheme secure in the sense of indistinguishability, for example the above construction. Now, $(KGen, Enc', Dec')$ is also secure in the sense of indistinguishability, and it works for messages of arbitrary length as follows:

$Enc'_f(m)$, where $|m| = n = l(k)$, s.t. k is the security parameter and $l(k)$ is some polynomial, encrypts every bit of $m = [m_1 \dots m_n]$ separately as $c_i = Enc_f(m_i)$, and outputs (c_1, \dots, c_n) as the total ciphertext. Decryption $Dec'_{f^{-1}}$ works in the obvious way, by decrypting each c_i separately, and recombining the resulting m_i 's into a bit-string $m = [m_1 \dots m_n]$.

Theorem 2 *If $(KGen, Enc, Dec)$ is a secure “one-bit encryption scheme” then $(KGen, Enc', Dec')$ is a secure (regular) encryption, where regular means “working on messages of any length”.*

The proof will use the so-called “hybrid” argument, which often appears in cryptographic arguments:

Proof: Assume adversary A breaks encryption $(KGen, Enc', Dec')$, i.e. there exist two $n = l(k)$ -bit long strings $m \neq m'$ s.t. A distinguishes encryptions $Enc'(m)$ from encryptions $Enc'(m')$, i.e. that

$$Pr[A(Enc'(m)) = 1] - Pr[A(Enc'(m')) = 1] > \epsilon(k)$$

for some higher-than-negligible $\epsilon(k)$ function.

The hybrid argument works as follows. Let's look at the behaviour of A on encryptions of messages $m^{(i)}$, for $0 \leq i \leq n$, where $m^{(i)}$ is composed by taking the first $n - i$ bits from m and then i last bits from m' . In this way, $m^{(0)} = m$, $m^{(n)} = m'$, and all the other $m^{(i)}$'s are hybrids between m and m' . Let

$$p_i = Pr[A(Enc'(m^{(i)})) = 1]$$

Our assumption on A can be now stated as just $p_0 - p_n > \epsilon$.

By the triangle inequality, if $p_0 - p_n > \epsilon$ then there must exist an index i , $0 < i \leq n$ s.t. $p_{i-1} - p_i > \epsilon/n$, because otherwise we would have:

$$p_0 - p_n = \sum_{i=1}^n (p_{i-1} - p_i) \leq n * (\epsilon/n) = \epsilon$$

which would contradict our assumption.

So take (any) i s.t. $p_{i-1} - p_i \geq \epsilon/n$. Note that $m^{(i-1)}$ and $m^{(i)}$ can possibly differ only on the $(n - i + 1)$ -th bit. [At least I think that this is the bit... They can for sure differ only on one bit!]. Now, they must differ on that bit because otherwise the two strings would be the same and therefore p_{i-1} and p_i would have to be the same. So assume that this bit is 0 in $m^{(i-1)}$ and 1 in $m^{(i)}$.

Now we can easily construct an adversarial algorithm A' which shows the original encryption scheme $(KGen, Enc, Dec)$ is insecure. To break the underlying encryption scheme we will construct A' , s.t.

$$Pr[A'(Enc(0)) = 1] - Pr[A'(Enc(1)) = 1] > \epsilon(k)/n$$

and therefore A' will distinguish encryptions of 0 from encryptions of 1 with non-negligible probability, because if $\epsilon = 1/poly(k)$ then ϵ/n is $1/(l(k) * poly(k)) = 1/poly'(k)$.

A' works as follows: On input c , which is either $Enc(0)$ or $Enc(1)$, A' runs the Enc encryption procedure $(n - i)$ times, to encrypt each of the first $(n - i)$ bits of $m^{(i-1)}$ (which are the same as the first $(n - i)$ bits of $m^{(i)}$), and thus he computes ciphertexts c_1, \dots, c_{n-i} . Then he uses the Enc encryption $(i - 1)$ times more, to encrypt each of the last $(i - 1)$ bits of $m^{(i-1)}$ (which are again the same as the last $(i - 1)$ bits of $m^{(i)}$), from bit position $(n - i + 2)$ to the last bit position n , and he computes this way ciphertexts c_{n-i+2}, \dots, c_n . Finally, he concatenates the results as

$$\bar{c} = (c_1, \dots, c_{n-i}, c, c_{n-i+2}, \dots, c_n)$$

and he runs A on \bar{c} . If A outputs 0, A' outputs 0 and if A outputs 1, A' outputs 1.

What's the running time of A' ? The same as the time of A plus $(n - 1) = (l(k) - 1)$ times the time it takes to run encryption procedure Enc , so overall it's polynomial in k if Enc is an efficient encryption and A is a PPT adversary.

And why is A' successful? Because if c is an encryption of 0 then \bar{c} has the same distribution as a random encryption Enc' on $m^{(i-1)}$, in which case A' outputs 1 with probability p_{i-1} , whereas if c is an encryption of 1 then \bar{c} has the same distribution as a random encryption Enc' on $m^{(i)}$, in which case A' outputs 1 with probability p_i . Therefore

$$Pr[A'(Enc(0)) = 1] - Pr[A'(Enc(1)) = 1] = p_{i-1} - p_i > \epsilon(k)/n$$

□