Course Howto

Lecture Videos & Lecture Notes You can find the lecture videos and lecture notes on Zulip: `https://crypto22.zulip.aalto.fi/join/5uu25r7rbfzrkhhxtugn57hr/`.

Lecture Discussion You can ask questions on the lecture to Chris/Russell each Monday from 12:15 - 14:00 during the exercise session (in person). These slots are reserved for *technical discussions*. We do not answer questions pertaining to passing criteria in these sessions.

Exercise Sessions We have exercise sessions on Monday 12:15 - 14:00, Tuesday 14:15-16:00, and Thursday 14:15-16:00. The goal of the exercise sessions is to get started working on the exercises together with others, to discuss your understanding with others and with the TAs and generally, to be part of our crypto course community and have fun.

Companion We refer to the crypto companion for definitions and clarifications you might need: `https://github.com/cryptocompanion/cryptocompanion`.

Zulip You can ask questions on the lectures and exercises on Zulip anytime. Chris/Russell will answer questions on Mondays. During the rest of the week, other course participants or teaching assistants might also help on Zulip (no promises). Teachers are not available during the week-end. Join Zulip here:
`https://crypto22.zulip.aalto.fi/join/5uu25r7rbfzrkhhxtugn57hr/`

Submission Submit your exercise solutions to MyCourses before Monday, September 12, 11:30. Your teaching assistant will carefully study your ideas and provide helpful suggestions. We provide a nice LaTeX template in the Materials section in MyCourses:
`https://mycourses.aalto.fi/course/view.php?id=33603&section=2`
which you can use for your exercise solutions, but you can also write on paper and scan the result or take a well-lit, high-quality picture. Please return your solutions as a single pdf.

Mistakes We encourage you to choose the option of choosing many exercises and be open to the possibility of making mistakes—studies on learning[1] tend to indicate that we learn when we make mistakes and get feedback to correct and/or refine our thinking. This is a central part of learning, so we encourage you to be open to the possibility of pushing the boundaries of your understanding in a safe space which supports your learning, which is appreciative of your effort to learn and acknowledges that learning means to experiment with thinking.

Passing the course Creating a safe space for experimenting with thinking and at the same time defining "course passing criteria" is somewhat in a tension with one another. Ideally, there would be no course passing criteria, but this is not possible, so we try to make course passing criteria such that they encourage and support engaging genuinely with the material. In particular, our point system is designed to allow to obtain full points also on an exercise sheet where none of the provided answers was correct—because point-giving should encourage learning and not get in the way of it by forcing everyone to only hand-in perfectly correct exercises from the start.

There are 10 exercise sheets, each worth at most 4 points (40 points total). A mostly correct solution to one exercise yields 2 points and a good attempt yields 1 point. You can attempt as many exercises per exercise sheet as you want, but the maximum amount of points is 4 per sheet (e.g. even if you submit more than 2 correct solutions, you can only get 4 points). No points are given for late submissions, but feedback is always given.

Watching the lecture and participating in all embedded quizzes of the lecture before 11:30 on Mondays yields 1 bonus point (up to 12 bonus points for the entire course). This is another way of obtaining quick feedback on your understanding.

[1] Unsuccessful Retrieval Attempts Enhance Subsequent Learning, Nate Kornell, Matthew Jensen Hays, and Robert A. Bjork, Journal of Experimental Psychology: Learning, Memory, and Cognition, 2009, Vol. 35, No. 4, 989-998, `https://sites.williams.edu/nk2/files/2011/08/Kornell.Hays_.Bjork_.2009.pdf`

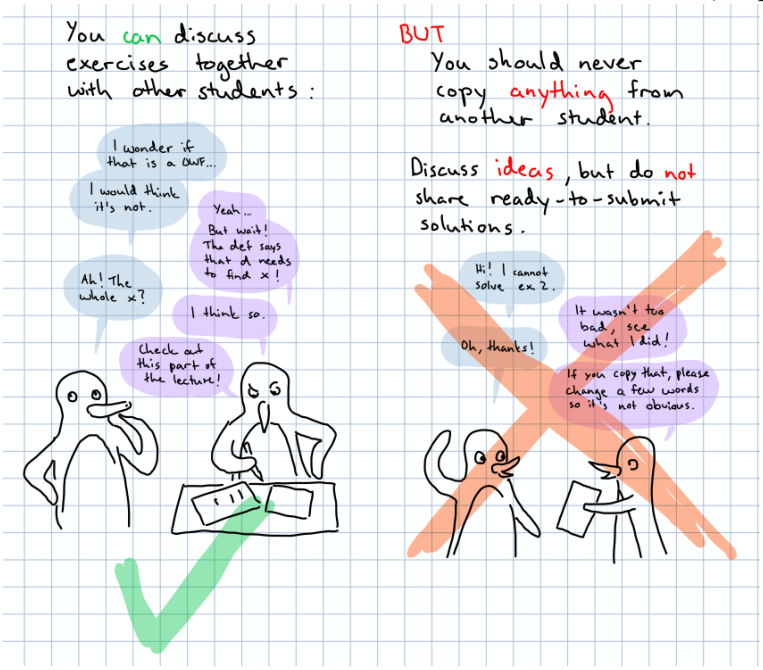
In-person sessions are reserved for discussions on cryptography. Question on system things such as passing criteria will only be answered on Zulip, not in in-person session (This preserves the informal character of in-person sessions and, additionally, is more fair, because everyone will be able to read the answers on Zulip, while in in-person sessions, only some people are present.).

Passing criteria. There are no grades, just pass or fail. Passing criteria are

- at least 30 points throughout the entire course, and
- at least 10 points in the first half of the course, and
- at least 10 points in the second half of the course

Thereby, you can choose and emphasize topics you enjoy and skip some exercises which you enjoy less. Also, if you are unable to submit one exercise sheet on time due to personal reasons or illness, it should not be too difficult to pass nonetheless. Hence, we do **not** extend the deadline even if you are unable to submit one exercise sheet for a good reason. Please take this into account when planning your studying.

Code of conduct. Remember to follow Aalto code-of-conduct[2], in particular:



[2] See in particular 'unattributed borrowing' here: `https://into.aalto.fi/display/ensaannot/Aalto+University+Code+of+Academic+Integrity+and+Handling+Violations+Thereof`

CS-E4340 Cryptography: Exercise Sheet 1

—One-Way Functions, Algorithms & Probabilities—

Submission deadline: September 12, 2022, 11:30, via MyCourses

Each exercise can give up to two participation points, 2 for a mostly correct solution and 1 point for a good attempt. Overall, the exercise sheet gives at most 4 participation points. We encourage to **choose** exercises which seem **interesting** and/or adequately challenging to you.

Exercise Sheet 1 is intended to help...

(a) ...understand the *definition* of One-Way Functions (OWFs) and gain *intuition* for OWFs.
(b) ...familiarize yourself with the idea of generic counterexamples.
(c) ...familiarize yourself with security experiments.
(d) ...practice thinking about probabilities.
(e) ...practice the use of pseudocode.

Exercise 1 aims to help understand probabilities.
Exercise 2 helps understand the notion of one-wayness and is the **most important** exercise on this sheet. We warmly encourage it.
Exercise 3 gives an opportunity to practice writing *inverters*[1] for some easy-to-invert functions, i.e. bad one-way function candidates.
Ex. 4 & Ex. 5 are advanced exercises, where you are asked to provide an attack on a generic one-way function counterexample and analyze the probability of the attack.

Exercise 1 (Probability and Pseudocode)**.** *2 points*

(a) You roll three (six-sided) dice D_1, D_2 and D_3. There are $6 \cdot 6 \cdot 6 = 216$ possible combinations of the results (D_1, D_2, D_3). For how many of the results is it true that $D_1 + D_2 + D_3 = 17$? Divide this number by 216 and determine: What is the probability that the sum $D_1 + D_2 + D_3$ is equal to 17?
(b) Define the function f and attacker $\mathcal{A}$ as

$$\begin{array}{l|l} f(x) & \mathcal{A}(y, 1^{|x|}) \\ \hline y \leftarrow x \oplus 1^{|x|} & z \leftarrow y \oplus 1^{|x|} \\ \textbf{return } y & \textbf{return } z \end{array}$$

and show that it holds that $\Pr\left[\mathsf{Exp}^{\mathsf{OW}}_{f,\mathcal{A}}(1^\lambda) = 1\right] = 1$. The above $\oplus$ means bitwise XOR operation. For more notation, we refer to the crypto companion `https://github.com/cryptocompanion/cryptocompanion`. Recall that the experiment $\mathsf{Exp}^{\mathsf{OW}}_{f,\mathcal{A}}(1^\lambda)$ is defined as:

$$\begin{array}{l} \mathsf{Exp}^{\mathsf{OW}}_{f,\mathcal{A}}(1^\lambda) \\ \hline x \leftarrow\!\!\$\ \{0,1\}^\lambda \\ y \leftarrow f(x) \\ x' \leftarrow\!\!\$\ \mathcal{A}(y, 1^\lambda) \\ \textbf{if } \left|x'\right| \neq \lambda \textbf{ then} \\ \quad \textbf{return } 0 \\ \textbf{if } f(x') = y \textbf{ then} \\ \quad \textbf{return } 1 \\ \textbf{return } 0 \end{array}$$

[1] For OWFs, the term *adversary* and *inverter* are synonymous, because an adversary against a OWF tries to invert.

Solution 1. (a) One die can take integer values between 1 and 6. Hence, the sum of three dice is at most $18 = 6 + 6 + 6$. The only way to get $D_1 + D_2 + D_3 = 17$ is when two dice show 6 and the third die shows 5. Thus, there are 3 such possibilities for (D_1, D_2, D_3) so that $D_1 + D_2 + D_3 = 4$, namely $(6, 6, 5)$, $(6, 5, 6)$ and $(5, 6, 6)$. Hence, the probability that $D_1 + D_2 + D_3 = 17$ is

$$\frac{\#\text{ of combinations which yield 17}}{\#\text{ of overall possible combinations}} = \frac{2}{216} = \frac{1}{72} \approx 0.0139.$$

(b) For any bit a, $a \oplus a = 0$ (since $0 \oplus 0 = 0$ and $1 \oplus 1 = 0$). In addition, $a \oplus 0 = a$, that is, XORing with zero does not change the bit. Now

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\mathsf{OW}}_{f,\mathcal{A}}(1^\lambda) = 1\right] &= \Pr\left[\mathcal{A}(f(x), 1^{|x|}) \in f^{-1}(f(x'))\right] && \text{| definition of } \mathsf{Exp}^{\mathsf{OW}}_{f,\mathcal{A}} \\
&\geq \Pr\left[\mathcal{A}(f(x), 1^{|x|}) = x\right] && \text{| true for any } \mathcal{A} \text{ and } f \\
&= \Pr\left[f(x) \oplus 1^{|x|} = x\right] && \text{| definition of } \mathcal{A} \\
&= \Pr\left[(x \oplus 1^{|x|}) \oplus 1^{|x|} = x\right] && \text{| definition of } f \\
&= \Pr\left[x \oplus 1^{|x|} \oplus 1^{|x|} = x\right] && \text{| XOR is commutative} \\
&= \Pr\left[x \oplus 0^{|x|} = x\right] && \text{| XORing } 1^{|x|} \text{ with itself} \\
&= \Pr[x = x] = 1 && \text{| XORing with zero}
\end{aligned}$$

(Above the probabilities are over the choice of x.)

Exercise 2 (One-Way Functions)**.** *2 points* Assume the existence of a length-preserving one-way function[2]. Say for each of the following statements whether you believe they are true or false and provide your intuition. You are not expected to *know* the answer to these questions, i.e., reasoning suffices (for 2 points) even if not all answers are correct. $||$ denotes concatenation of strings.

Hint. recall from the lecture that if f is one-way function, then $h(x_l||x_r) := f(x_l)||x_r$ and $h'(x) := f(x)||0^\lambda$ are also one-way functions, where $|x_l| = |x_r|$. You can use the examples from the lecture and Section 4 of the crypto companion without justifying them.

(a) For all length-preserving one-way functions f and g, the following function h is a one-way function: $h(x) := f(x)||g(x)$.

(b) For all one-way functions f and all polynomially computable functions b with one bit output, the following function h is a one-way function: $h(x) := f(x)||b(x)$.

(c) For all length-preserving one-way functions f and g, the following function h is a one-way function: $h(x) := g(f(x))$.

(d) For all length-preserving one-way functions f, the following function h is a one-way function: $h(x) := f(x)_{1\ldots\lceil|x|/2\rceil}$. I.e., h returns all bits that f returns, except for half of the bits (rounded up).

(e) (*Advanced*) For all length-preserving one-way functions g, the following function h is a one-way function: $h(x) := g(x)_{1\ldots|x|-1}$. I.e., h returns all bits that g returns, except for the last bit.

(f) For all length-preserving one-way functions f, g, the following function h is a one-way function: $h(x) := f(x) \oplus g(x)$. I.e., h is the bitwise XOR of two OWFs.

[2] A function $f : \{0,1\}^* \to \{0,1\}^*$ is *length-preserving* if for all $x \in \{0,1\}^*$, it holds that $|f(x)| = |x|$.

(g) There exists a one-way function h with 1 bit output, i.e., for all $x \in \{0,1\}^*$, $|h(x)| = 1$.
(h) For all length-preserving one-way functions f, g, the following function h is a one-way function: $h(x) := f(x)||(x \oplus g(f(x)))$. I.e., h first applies f to x and then g to x, then xors the result with x, this is the second half of the function h, and the first half of the function h is just $f(x)$.
(i) For all length-preserving one-way functions f, the following function h is a one-way function: $h(x) := f(1^{|x|})$
(j) For all length-preserving one-way functions f, the following function h is a one-way function:

$$h(b||x) := \begin{cases} 0||f(x) & \text{if } b = 0 \\ 1||x & \text{if } b = 1. \end{cases}$$

where b is of length 1 (first bit of the input).

Solution 2. Some of the model solutions below are very detailed. For Exercise 2, this level of detail is not always necessary, but an intuitive argument is enough. However, as Exercises 3-5 require more details, we provide those already here to avoid repeating solutions.

Statement (a) The answer is no. We construct counterexample functions f and g which are one-way functions, but h is easily inverted.

In the lecture, we saw that if owf is a length-preserving one way function, then the following function f is a one-way function as well.

$$\begin{aligned} f : \{0,1\}^{**} &\to \{0,1\}^{**} \\ x &\mapsto \mathsf{owf}(x_l)||x_r \end{aligned}$$

where $x = x_l||x_r$ and $|x_l| = \frac{|x|}{2}$. By symmetry,

$$\begin{aligned} g : \{0,1\}^{**} &\to \{0,1\}^{**} \\ x &\mapsto x_l||\mathsf{owf}(x_r), \end{aligned}$$

where the left side of the input is leaked (instead of the right side), is also a one-way function.
Now, we have two one-way-functions ready, f and g. Let's combine them and see, why the formed function h is no longer one-way. Denote by $\{0,1\}^{****}$ the set of all inputs with length divisible by 4.

$$\begin{aligned} h : \{0,1\}^{**} &\to \{0,1\}^{****} \\ x &\mapsto f(x)||g(x) = \mathsf{owf}(x_l)||x_r||x_l||\mathsf{owf}(x_r) \end{aligned}$$

It is very easy to describe algorithm $\mathcal{A}$ that inverts h in polynomial time with non-negligible probability. This algorithm, given $h(x)$, reads the middle part of the string, $x_r||x_l$, and arranges it correctly in order to output $x_l||x_r = x$. This works in time $O(n)$ and gives a correct input with probability 1. Therefore h is **not** a one-way function, even if it is constructed from two one-way functions. We have found a counterexample, and the statement is false.
Statement (b) Yes. Intuitive proof: suppose for contradiction, that h is not OWF. Now there is an efficient inverter for h for some function b. Let's call the inverter $\mathcal{A}_h$. Now we can build an inverter $\mathcal{A}_f$ for f as follows:

$\mathcal{A}_f(y, 1^{|x|})$

$z \leftarrow \mathcal{A}_h(y||1)$

if $f(z) = y$

  return z

$z \leftarrow \mathcal{A}_h(y||0)$

return z

This inverter is efficient, since $\mathcal{A}_h$ is efficient. The inverter for f just tries both possible values for $b(x)$ and asks $\mathcal{A}_h$ to invert with both of the guesses. One of the guesses is correct for sure, since b can only have outputs 0 and 1. Hence, when the guess is correct, $\mathcal{A}_h$ inverts the function with non-negligible probability, and hence $\mathcal{A}_f$ is able to invert f with non-negligible probability, which is a contradiction, since f is OWF. So h has to be OWF too.

Take home: revealing just one bit of information of the input is not enough to invert a OWF, since one bit can just be guessed.

Statement (c)

The statement is false. We show this by constructing two one-way functions in such a way, that their composition results in a trivial function.

Let f be a length-preserving one-way function. Define

$$g_1(x) = f(x_{[1..k]})||0^{|x|-k}$$
$$g_2(x) = 0^k||f(x_{k+1..|x|})$$

where $k = \left\lfloor \frac{|x|}{2} \right\rfloor$. One can prove via reduction that both g_1 and g_2 are one-way functions: inverting them would require inverting f with a half-sized input.

The composition becomes now

$$h(x) = g_1(g_2(x)) = g_1(0^k||f(x_{[k+1..|x|]})) = f(0^k)||0^{|x|-k}$$

which is a constant function. All the strings map to the same output, so inverting h is trivial: consider $\mathcal{A}$ that returns a constant string. This is in a preimage of $h(x)$ since it is a constant and therefore $\mathcal{A}$ succeeds with probability 1.

Statement (d) The statement is false. This can be proven similarly to (e) or more easily as follows.

Suppose g is a length-preserving one-way function. Now define $f(x_l||x_r) = x_l||g(x_r)$ where $|x_l| = |x_r|$. Recall from the lecture that such f is a one-way function.

Now $h(x_l||x_r) = f(x_l||x_r)_{1...\lceil |x_l||x_r|/2 \rceil} = f(x_l||x_r)_{1...|x_l|} = x_l$.

Now h can easily be inverted, since the adversary can simply append any string to the output of h, as long as the appended string is of the same length with the output. For example, if the output of h is y, the inverter can simply return $y||1^{|y|}$. Such inverter inverts h with probability 1. See probability analysis below.

Define

$\mathcal{A}(y, 1^{|x|})$

return $y||1^{|y|}$

This adversary is clearly linear time and hence efficient.

Now

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\mathsf{OW}}_{h,\mathcal{A}}(1^\lambda)=1\right] &= \Pr\left[\mathcal{A}(h(x),1^{|x|}) \in h^{-1}(h(x))\right] && \text{| definition of } \mathsf{Exp}^{\mathsf{OW}}_{h,\mathcal{A}} \\
&= \Pr\left[\mathcal{A}(x_l,1^{|x|}) \in h^{-1}(x_l)\right] && \text{| definition of } h \\
&= \Pr\left[x_l||1^{|x_l|} \in h^{-1}(x_l)\right] && \text{| definition of } \mathcal{A} \\
&= 1 && \text{| since } h(x_l||1^{|x_l|}) = x_l
\end{aligned}$$

Since h can be efficiently inverted with probability 1, h is not OWF.

Statement (e)

The statement is false. We construct a counterexample, such that g is a length-preserving one-way function but $h(x) := g(x)_{1\ldots|x|-1}$ is easily inverted.

The high-level idea is to construct g in such a way that sometimes it leaks the input and sometimes doesn't. The last bit of the output of g tells whether the input has been leaked. By removing the last bit from the output thus allows the adversary to assume every output came from leaked inputs, hence making inverting h trivial.

Let $f : \{0,1\}^* \to \{0,1\}^*$ be a length-preserving one-way function. Define g as

$$\begin{aligned}
g :&\{0,1\}^* \to \{0,1\}^* \\
g(x) &= \begin{cases} x_r||0^{|x_l|}, & \text{if } x_l = 0..0 \\ f(x_r)||0^{x_l-1}||1, & \text{otherwise.} \end{cases}
\end{aligned}$$

where $x = x_l||x_r$ and $|x_l| = \left\lfloor \frac{|x|}{2} \right\rfloor$ as earlier. Indeed, g is length-preserving by construction.

Since $f(x)$ is computable in polynomial time, also $g(x)$ is. Inverting g is just as hard as inverting f, **except** in the case when $x_l = 0..0$. This, however, only happens with probability $2^{-\frac{n}{2}}$, which makes the probability of this case occuring negligible. This can be proven rigorously via a reduction argument, which we omit here. Now take

$$\begin{aligned}
h &: \{0,1\}^* \to \{0,1\}^* \\
h(x) = g(x)_{[1\ldots|x|-1]} &= \begin{cases} x_r||0^{|x_l|-1}, & \text{if } x_l = 0..0 \\ f(x_r)||0^{|x_l|-1}, & \text{otherwise.} \end{cases}
\end{aligned}$$

Consider algorithm $\mathcal{A}$, that takes input $(y, 1^{|x|})$ and outputs $0^{|x_l|}||y_{1\ldots|x_r|}$. Then $\mathcal{A}$ inverts h successfully with probability 1, since $h(0^{|x_l|}||y_{|x_r|}) = y_{|x_r|}||0^{|x_l|-1} = y$, where the last equality holds as the last $|x_l| - 1$ bits of any output of h are zero. Inverting h is easy, and therefore it is not a one-way function.

Statement (f) The statement is false. Consider the case $g = f$. Now h is a constant 0 function, which is not OWF. Consider for example the following inverter for h:

$$\begin{array}{l}
\mathcal{A}(y, 1^{|x|}) \\
\hline
x' \leftarrow\!\$\ \{0,1\}^{|x|} \\
\textbf{return } x'
\end{array}$$

Since h is constant function, every input x gives the same output. Hence, to invert h, one can return any x of the correct length and that is a correct preimage with probability 1.

Probability analysis:

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\mathsf{OW}}_{h,\mathcal{A}}(1^\lambda)=1\right] &= \Pr\left[\mathcal{A}(h(x),1^{|x|}) \in h^{-1}(h(x))\right] && \text{| definition of } \mathsf{Exp}^{\mathsf{OW}}_{h,\mathcal{A}} \\
&= \Pr\left[\mathcal{A}(0^{|x|},1^{|x|}) \in h^{-1}(0^{|x|})\right] && \text{| definition of } h \\
&= \Pr\left[x' \in h^{-1}(0^{|x|})\right] && \text{| where } x' \text{ is a random string} \\
&= 1 && \text{| since } h(x')=0^{|x|} \text{ for all } x'
\end{aligned}$$

(Above x' is of the same length as x.)
Statement (g) The statement is false. The adversary can simply guess the input (i.e. sample a uniformly random x), and with probability at least $1/2$, the input yields the correct output bit. The success probability of such adversary is exactly $1/2$ if h is balanced, i.e. if h outputs 1 half of the time and 0 half of the time. If h is not balanced, the success probability of the adversary is even higher (consider for example the extreme case where h is the constant 1 function, in that case the adversary succeeds always.)
Statement (h) $h(x) = f(x)||(x \oplus g(f(x)))$: not OWF. Consider the following adversary against h:

$\mathcal{A}(y, 1^{|x|})$
$l \leftarrow y_{1,\ldots,|x|}$
$r \leftarrow y_{|x|+1,\ldots,|y|}$
$z \leftarrow g(l)$
$x' \leftarrow z \oplus r$
return x'

The adversary is efficient, since it only uses linear-time operations (such as XOR and reading half of the input) and a single call to g, which is polynomial time, since g is a OWF. Hence, the adversary is polynomial time in $|x|$ overall.
This adversary inverts h with probability 1, see probability analysis below.

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\mathsf{OW}}_{h,\mathcal{A}}(1^\lambda)=1\right] &= \Pr\left[\mathcal{A}(h(x),1^{|x|}) \in h^{-1}(h(x))\right] \\
&= \Pr\left[\mathcal{A}(f(x)||(x \oplus g(f(x))),1^{|x|}) \in h^{-1}(h(x))\right] && \text{| definition of } h \\
&= \Pr\left[x \in h^{-1}(h(x))\right] && \text{| def. of } \mathcal{A}, \oplus\text{-rules} \\
&= 1
\end{aligned}$$

Statement (i) $h(x) = f(1^{|x|})$: not OWF.

$\mathcal{A}(y, 1^{|x|})$
$x' \leftarrow\!\!\$\ \{0,1\}^{|x|}$
return x'

Since h is constant function, every input x gives the same output. Hence, to invert h, one can return any x of the correct length and that is a correct preimage. See probability analysis for inverting a constant function in part (f).
Statement (j) The statement is false. The function can be inverted half of the time, namely, whenever the first bit of the input is 1. One half is non-negligible.
Consider the following inverter:

$\mathcal{A}(y, 1^{|x|})$

$b \leftarrow y_1$
if $b = 1$
  return y
else
  return error

This inverter inverts h with probability half. See probability analysis below.

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\mathsf{OW}}_{h,\mathcal{A}}(1^\lambda) = 1\right] &= \Pr\left[\mathcal{A}(h(b||x), 1^{|b||x|}) \in h^{-1}(h(b||x))\right] \\
&= \Pr\left[\mathcal{A}(h(b||x), 1^{|b||x|}) \in h^{-1}(h(b||x)) \,\middle|\, b = 1\right] \Pr[b = 1] \\
&\quad + \Pr\left[\mathcal{A}(h(b||x), 1^{|b||x|}) \in h^{-1}(h(b||x)) \,\middle|\, b = 0\right] \Pr[b = 0] \\
&= \Pr\left[b||x \in h^{-1}(h(b||x)) \,\middle|\, b = 1\right] \Pr[b = 1] \\
&\quad + \Pr\left[\text{error} \in h^{-1}(h(b||x)) \,\middle|\, b = 0\right] \Pr[b = 0] \\
&= 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} \\
&= \frac{1}{2}
\end{aligned}$$

Exercise 3 (Constructing Inverter)**.** *2 points* Choose *one* out of (h), (i) from Exercise 2, give an efficient inverter and argue that the inversion probability is 1. Alternatively, you can also choose *one* out of (j) or (g) and give an inverter which inverts with probability $\frac{1}{2}$.

Solution 3. See the solution for ex 2. (Ask a TA if the part you are interested in is missing too much details.)

Exercise 4 (Attack a OWF-Candidate)**.** *2 points* Choose one of the constructions h in Exercise 2 (a)-(e) that is not one-way. Argue why it is not OWF, that is, provide an inverter and argue why the inverter is efficient (intuitive argument is enough).

Solution 4. See the solution for ex 2. (Ask a TA if the part you are interested in is missing too much details.)

Exercise 5 (Analyze the Attacker)**.** *2 points* What is the inversion probability of your inverter from the previous exercise? Justify your answer. Is the probability non-negligible? **Hint:** Since we haven't discussed the definition of *non-negligible*[3] in the lecture, you can either look it up in the crypto companion, or simply argue that your inverter inverts with constant probability, e.g., $\frac{1}{10}$.

[3] A negligible function is a function tends to zero faster than any inverse polynomial as λ tends to infinity. A non-negligible function is a function which is not negligible. See Definition 2.2 in the *crypto companion* `https://github.com/cryptocompanion/cryptocompanion`.

Solution 5. See the solution for ex 2. (Ask a TA if the part you are interested in is missing too much details.)