# Solutions 4

CSC 152/252 – Cryptography

Please notify me of any errors you find. If you need help, ask.

**1)** *Recall that you can compute a value that is equivalent to* $x \bmod (2^a - b)$ *as* $(x \text{ div } 2^a)b + (x \bmod 2^a)$*. Use this fact to reduce* $123456789 \bmod (2^{16} - 2)$ *to a 16-bit value. If after doing this reduction once, the result is more than 16 bits, do it a second time to reduce it further.*

There are a couple of ways to do this problem.

The mathematical way goes like this. If you divide $123456789/2^{16}$ you get 1883 and remainder 52501. This means $123456789 = 1883 \cdot 2^{16} + 52501$. If you reduce all the values on the right-hand-side mod $2^{16} - 2$ you get $123456789 = 1883 \cdot 2 + 52501 = 56267$. Since this result is less than $2^{16} - 2$ it needs no further reduction.

Optimized on a computer, it goes like this. 123456789 in binary is 111010110111100110100010101. Since the bits beyond the low 16 are compatible for addition with the low 16 bits when first multiplied by 2, we get $111010110110 + 1100110100010101 = 1101101111001011$ and $1101101111001011 = 56267$.

**2)** *Recall that* $H$ *is* $\varepsilon$*-almost-universal if the probability* $h(a) = h(b)$ *is no more than* $\varepsilon$ *when* $a \neq b$ *and* $h \in H$ *is chosen randomly. The following* $H$ *is a family of functions all with domain* $\mathbb{Z}_6$ *and co-domain* $\mathbb{Z}_4$*. For what value of* $\varepsilon$ *is* $H$ $\varepsilon$*-almost-universal? Show your work.* $H$ *is defined as follows:*

```
     h1   h2   h3   h4   h5
---+------------------------
 0 |  2    3    0    1    3
 1 |  3    2    1    0    0
 2 |  0    1    3    2    1
 3 |  0    0    2    2    3
 4 |  2    1    1    3    2
 5 |  0    3    3    2    0
```

If the adversary chooses 2 and 5, then when $h$ is chosen randomly, there is a 3/5 chance that $h(2) = h(5)$ because $h_1, h_3, h_4$ all cause a collision. No other pair of inputs has a higher probability of collision when $h$ is chosen randomly, so the collection of functions is $\varepsilon$-almost-universal for $\varepsilon = 3/5$. On an exam with a small domain you should list all possible pairs of domain elements, give each probability, and identify the maximum.

**3)** *Write one of the following reductions: from PreimageFinder($H, y$) to 2ndPreimageFinder($H, x$), or from 2ndPreimageFinder($H, x$) to PreimageFinder($H, y$). What security implication does your reduction establish?*

Showing FIND2NDPREIMAGE($H, x$) reduces to FINDPREIMAGE($y$) is straightforward.

```
Find2ndPreimage(H,x)
    do
        x' = FindPreimage(H,H(x))
    while (x' == x)
    return x'
```

I used a loop in case FINDPREIMAGE($H, H(x)$) returns $x$. This reduction will fail if FINDPREIMAGE($H, H(x)$) always returns $x$ which could happen if FINDPREIMAGE($H, H(x)$) is deterministic and finds $x$ as its first answer, or if $x$ is the only preimage of $y$ (unlikely).

This establishes that the existence of an efficient FINDPREIMAGE implies the existence of an efficient FIND2NDPREIMAGE. It's contrapositive tells us if there is no efficient FIND2NDPREIMAGE then there is no efficient FINDPREIMAGE, or in other words second-preimage resistance implies preimage resistance.

The other direction would look like this.

```
FindPreimage(H,y)
    ...
    x' = Find2ndPreimage(x)
    ...
```

The problem is that FINDPREIMAGE is given an element of the range and FIND2NDPREIMAGE requires an element of the domain, so we'd have to find an $x$ where $H(x) = y$ in order to ask FIND2NDPREIMAGE to do its work.

**Programming assignments)** Programming assignments will be discussed in class (or office hours if you'd like).