TTM4135 Worksheet 4: Hashing, MACs and Number theory

Tjerand Silde and Emil August Hovd Olaisen
{tjerand.silde, emil.august.olaisen}@ntnu.no

Spring 2026

1. Review the definitions of the following concepts:
   (a) collision resistance, second preimage resistance and one-wayness;
   (b) birthday paradox;
   (c) HMAC;
   (d) GCM mode;
   (e) big O notation;
   (f) Fermat test;
   (g) Miller–Rabin test.
   (h) factorization and discrete logarithm problems.

2. (a) Suppose that 10 items are chosen randomly (with replacement) from a set of 30 items. Find the probability that there is no collision (or, in other words, all the items are different). (Hint: consider the experiment one item at a time and multiply the probability that there is no collision each time.)
   (b) Explain why SHA-256 can be said to match the security of AES with 128-bit keys.

Solution:

(a) The probability of no collision is the product of the probabilities of no collision each time a new element is collected. For 10 elements this is

$$\frac{30}{30} \cdot \frac{29}{30} \cdot \frac{28}{30} \cdot \ldots \cdot \frac{21}{30} = 0.19.$$

Therefore the probability of at least one collision is $1 - 0.19 = 0.81$.

(b) Due to the birthday paradox around 2^{128} computations are required to obtain a collision in SHA-256. This is roughly the same amount of computation required for a brute-force attack on 128-bit key AES.

3. A rough, but simple, estimate for the probability $p(n)$ of obtaining one or more collisions when choosing n items (with replacement) from a set of H items is:

$$p(n) \approx \frac{n^2}{2H}.$$

Use this formula to estimate the probability of finding a collision in SHA-256 after 2^{128} trials, after 2^{100} trials, and after 2^{80} trials.

Solution: Since SHA-256 has a 256-bit output we have $H = 2^{256}$.

- When $n = 2^{128}$ we get $p(n) = \frac{2^{256}}{2^{257}} = 0.5$.
- When $n = 2^{100}$ we get $p(n) = \frac{2^{200}}{2^{257}} = 2^{-57}$.
- When $n = 2^{80}$ we get $p(n) = \frac{2^{160}}{2^{257}} = 2^{-97}$.

We see that even after 2^{100} trials, which is way beyond normal attackers, the success probability is tiny.

4. When does the addition of the padding and length field in the SHA-2 family of hash functions result in an extra block to be processed?

   Consider, for example, a SHA-2 variant with 1024-bit blocks (such as SHA-512) and a message m of between $(l-1) \times 1024$ bits and $l \times 1024$ bits. The message, after adding padding and the length field, is either l blocks or $l+1$ blocks. Exactly how long can the message be before $l+1$ blocks will be used?

Solution:

Suppose m has length $(l-1) \times 1024 + r$ bits. At least 1 bit of padding is added and 128 bits are needed for the length field. So if $r \leq 1024 - 129 = 895$ then no block is added to the pre-processed data.

5. Suppose that HMAC is implemented using a hash function H, where H is an iterated hash function with compression function h.
   (a) How many additional applications of the compression function h are required to compute the MAC of a message m, in comparison with computing only $H(m)$?
   (b) If a MAC tag is to be computed for many different messages but the same key, how can pre-computation be used to reduce this overhead?

Solution:

(a) The computation of $I = H\left((K \oplus \text{ipad}) \parallel m\right)$ requires one extra application of h to compute, compared with computing simply $H\left(m\right)$. The computation of $H((K \oplus opad) \parallel I)$ requires only two or three applications of h, depending on whether the padding adds one extra block. For all the SHA variants the output size of I is much smaller than the block size, so in the standardized versions there are only two extra applications of h. So there are in total 3 extra applications of h to compute $\text{HMAC}(m)$ compared to computing $H(m)$. For large messages this is a very small overhead.

(b) We can compute $h(K \oplus opad)$ and $h(K \oplus ipad)$ in advance since they are independent of the message. This saves at two of the three overhead computations of h, and can be useful if HMAC is applied to many short messages.

6. Consider the following simplification of HMAC, defined from any Merkle-Damgård hash function H:

$$\text{HMAC}'(m, K) = H(K \parallel m).$$

Show that this variant allows an attacker to forge a new valid MAC tag given any valid message/tag pair (m, T) by extending m to a new message m' and finding a valid tag T' for m'.

Solution:

Assume $\text{HMAC}'(m, K) = T$ is known by the attacker where m has n-bit blocks $m_1 \| m_2 \| \ldots \| m_\ell$. If we write $h_i = h(m_i, h_{i-1})$ as the intermediate outputs of h, the final value of the tag is $\text{HMAC}'(m, K) = h(\text{pad}, h_\ell)$. The attacker can, for example, choose an extra block m_{l+1}, compute the new correct padding block pad', and then compute the new correct MAC tag

$$T' = h(\text{pad}', h(m_{l+1}, T))$$

which is a valid tag for the message:

$$m' = m \parallel \text{pad} \parallel m_{l+1}.$$

7. If possible, solve for the following equations for x using the Chinese Remainder Theorem (CRT). If they do not have a solution, then explain why.
   (a) $x \equiv 5 \pmod 7$ and $x \equiv 7 \pmod{10}$.
   (b) $x \equiv 3 \pmod 7$ and $x \equiv 7 \pmod{14}$.
   (c) $x \equiv 2 \pmod 6$ and $x \equiv 3 \pmod{11}$.
   (d) $x \equiv 8 \pmod 9$ and $x \equiv 11 \pmod{15}$.

Solution:

(a) Since $\gcd(7, 10) = 1$ a solution must exist. Using the CRT we have:

$$\begin{aligned} x &= (10^{-1} \bmod 7 \cdot 10 \cdot 5) + (7^{-1} \bmod 10 \cdot 7 \cdot 7) \bmod 70 \\ &= (3^{-1} \bmod 7 \cdot 50) + (7^{-1} \bmod 10 \cdot 49) \bmod 70 \\ &= (5 \cdot 50) + (3 \cdot 49) \bmod 70 \\ &= 250 + 147 \bmod 70 \\ &= 47. \end{aligned}$$

(b) Since $\gcd(7, 14) = 7$ CRT cannot be applied. In fact, $x \equiv 7 \pmod{14}$ implies (but is not equivalent to) $x \equiv 0 \not\equiv 3 \pmod 7$. Therefore no solution exists.

(c) Since $\gcd(6, 11) = 1$ a solution must exist. Using the CRT we have:

$$\begin{aligned} x &= (11^{-1} \bmod 6 \cdot 11 \cdot 2) + (6^{-1} \bmod 11 \cdot 6 \cdot 3) \bmod 66 \\ &= (5^{-1} \bmod 6 \cdot 22) + (6^{-1} \bmod 11 \cdot 18) \bmod 66 \\ &= (5 \cdot 22) + (2 \cdot 18) \bmod 66 \\ &= 110 + 36 \bmod 66 \\ &= 14. \end{aligned}$$

(d) If the answer exists, then it must exist $\bmod 45$, since it is the least common multiple of 9 and 15. We then get the following candidates:

$$\begin{aligned} x &\equiv 8 \equiv 17 \equiv 26 \equiv 35 \equiv 44 \bmod 9 \\ x &\equiv 11 \equiv 26 \equiv 41 \bmod 15. \end{aligned}$$

From these candidates we can conclude that:

$$x \equiv 26 \bmod 45.$$

8. Find $\phi(n)$ for all integers between 20 and 25 inclusive.

Solution:

1. $\phi(20) = \phi(2^2 \cdot 5) = (2^2 - 2^1) \cdot (5^1 - 5^0) = 2 \cdot 4 = 8$
2. $\phi(21) = \phi(3 \cdot 7) = (3^1 - 3^0) \cdot (7^1 - 7^0) = 2 \cdot 6 = 12$
3. $\phi(22) = \phi(2 \cdot 11) = (2^1 - 2^0) \cdot (11^1 - 11^0) = 1 \cdot 10 = 10$
4. $\phi(23) = 23^1 - 23^0 = 22$
5. $\phi(24) = \phi(2^3 \cdot 3) = (2^3 - 2^2) \cdot (3^1 - 3^0) = 4 \cdot 2 = 8$
6. $\phi(25) = \phi(5^2) = 5^2 - 5^1 = 20$

9. Find the discrete logarithm of the number 3 with regard to base 2, for the following moduli:
   (a) modulus $p = 5$;
   (b) modulus $p = 11$;
   (c) modulus $p = 29$.

Solution: We need to find the value x with $2^x = 3 \bmod p$ for the different value of p.

(a) Powers of 2 modulo 5 are 2, 4, 3, 1. So $x = 3$.

(b) Powers of 2 modulo 11 are: 2, 4, 8, 5, 10, 9, 7, 3, 6, 1. So $x = 8$.

(c) Powers of 2 modulo 29 are: 2, 4, 8, 16, 3, ... So $x = 5$.

10. Use the Fermat test to check whether the following numbers are prime or not. Run the test at most 4 times.
    (a) 979;
    (b) 983.

Solution:

(a) $2^{978} \pmod{979} = 223$. Therefore 979 cannot be prime and there is no point in looking further. The test outputs `composite`.

(b) We choose a few bases b and check whether $b^{982} \bmod 983 = 1$. For example,

$$\begin{aligned} 2^{982} \bmod 983 &= 1 \\ 3^{982} \bmod 983 &= 1 \\ 11^{982} \bmod 983 &= 1 \\ 17^{982} \bmod 983 &= 1 \end{aligned}$$

We can be (somewhat) confident that 983 is prime, and the test outputs `probable prime`. Note that the bases we chose here are not random, but in practice randomly sampled bases are usually applied.

11. (a) Show that the Carmichael number $n = 1105$ passes the Fermat test for base $a = 2$ and $a = 3$.
    (b) Now try the Miller–Rabin test for the same two bases and show that n is composite.
    (c) Hence find a square root of 1 mod n and use this to find a factor of n.

Solution:

(a) Check that $a^{1104} \bmod 1105 = 1$ for $a = 2$ and $a = 3$.

(b) $1104 = 2^4 \cdot 69$. Therefore we compute

$$\begin{aligned} 2^{69} \bmod 1105 &= 967 \\ 967^2 \bmod 1105 &= 259 \\ 259^2 \bmod 1105 &= 781 \\ 781^2 \bmod 1105 &= 1. \end{aligned}$$

Here we stop since we got to 1 and the test outputs `composite` which shows that 1105 is definitely composite.

(c) In the last equation we found that 781 is a square root of 1 modulo 1105. Therefore $\gcd(780, 1105)$ is a factor of 1105. Running the Euclidean algorithm gives:

$$\begin{aligned} 1105 &= 1 \cdot 780 + 325 \\ 780 &= 325 \cdot 2 + 130 \\ 325 &= 130 \cdot 2 + 65 \\ 130 &= 2 \cdot 65 \end{aligned}$$

So $\gcd(780, 1105) = 65$ and we find $1105 = 65 \cdot 17$.