H2.

This is a first year class in security. If you are not familiar with some of the mathematical notions of notation presented here, please do not hesitate to ask the course staff. Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.

Please type your answers and submit them as a PDF. To submit, commit a file named `hw2.pdf` to your gitlab repository.

PROBLEM 1 *GCD*

EXTENDED-EUCLID(a, b) where $a \geq b$

```
1  if a mod b = 0 return (0, 1)
2  (x, y) ← EXTENDED-EUCLID(b, a mod b)
3  return (y, x − y(⌊a/b⌋))
```

Use the extended Euclid algorithm presented in class to compute the multiplicative inverse of 71 in the group $\mathbb{Z}_{101}^*$ (i.e., find $y \in \mathbb{Z}_{101}^*$ such that $71 \cdot y = 1 \bmod 101$).

PROBLEM 2 *Compute* $11^{12^{400}} \bmod 67$ *using Euler's theorem (show your work).*

PROBLEM 3 *Textbook RSA is insecure*

Consider the textbook RSA signature scheme with the public key and private keys $pk = (n, e)$, $sk = (n, d)$, and $d \cdot e = 1 \bmod \phi(n)$ for an RSA modulus $n = pq$ and:

1. $\text{Sign}_{sk}(m)$: Compute $\sigma \leftarrow m^d \bmod n$. Output σ
2. $\text{Verify}_{pk}(m, \sigma)$: Verify that $m = \sigma^e \bmod n$.

If an adversary knows the signature σ_1 of message m_1 and the signature σ_2 of message m_2 under the same key, show how the adversary can compute the signature of $m_1 \cdot m_2$. Does this scheme satisfy the notion of unforgeability given in class?

PROBLEM 4 *MD5 Collisions*

As we discussed in class, MD5 was once considered a secure collision resistant hash function. However, in 2004, a group of researchers, Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu, discovered a catastrophic flaw in the construction and published several collisions. One pair they published was:

```
d131dd02c5e6eec4693d9a0698aff95c 2fcab58712467eab4004583eb8fb7f89
55ad340609f4b30283e488832571415a 085125e8f7cdc99fd91dbdf280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e2b487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080a80d1e c69821bcb6a8839396f9652b6ff72a70

d131dd02c5e6eec4693d9a0698aff95c 2fcab50712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325f1415a 085125e8f7cdc99fd91dbd7280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e23487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080280d1e c69821bcb6a8839396f965ab6ff72a70
```

1. Using the `xxd` program and `openssl dgst -md5`, what was the collision value?
2. Suppose one uses PKCS1.5 signature padding, as discussed in class, with the MD5 hash. Explain how an attacker can break the system.
3. *Extra credit:* Using the fastcoll program from , find a new pair of collisions for MD5. Name the two files `c1.bin` and `c2.bin` and submit them.