Solutions to Daily Due Oct 2 (AI generated)

Math 4440/5440, Coding and Cryptography, Fall 2026

Generated by Claude for Katherine E. Stange
(all computations computer-verified)

Conventions as in lecture: the extended Euclidean algorithm is written as rows $a = qb + r$, with a "recipe card" (x, y) over each number d meaning $ax + by = d$ (for the two original inputs a, b). The first two cards are $(1, 0)$ over a and $(0, 1)$ over b, and each remainder's card is (left card) $-q\cdot$ (middle card).

Problem 1: the inverse of 35 modulo 61

Run the Euclidean algorithm on 61 and 35, keeping recipe cards for $61x + 35y = d$:

row			cards: (x, y) over 61, 35, remainder
61	$=$	$1 \cdot 35 + 26$	$(1, 0),\ (0, 1),\ (1, -1)$
35	$=$	$1 \cdot 26 + 9$	$(0, 1),\ (1, -1),\ (-1, 2)$
26	$=$	$2 \cdot 9 + 8$	$(1, -1),\ (-1, 2),\ (3, -5)$
9	$=$	$1 \cdot 8 + 1$	$(-1, 2),\ (3, -5),\ (-4, 7)$
8	$=$	$8 \cdot 1 + 0$	

The last nonzero remainder is 1, so $\gcd(61, 35) = 1$, and its card says

$$61 \cdot (-4) + 35 \cdot 7 = 1.$$

Reducing modulo 61: $35 \cdot 7 \equiv 1 \pmod{61}$, so

$$35^{-1} \equiv 7 \pmod{61}.$$

Check: $35 \cdot 7 = 245 = 4 \cdot 61 + 1$. ✓

(The same equation reduced modulo 35 says $61 \cdot (-4) \equiv 1$, i.e. $61^{-1} \equiv -4 \equiv 31 \pmod{35}$: one equation, two inverses, as on the daily due Sep 28.)

Problem 2: $x \equiv 3 \pmod{244}$ and $x \equiv 17 \pmod{495}$

Step 1: extended Euclid on 495 and 244.

row			cards over 495, 244, remainder
495	$=$	$2 \cdot 244 + 7$	$(1, 0),\ (0, 1),\ (1, -2)$
244	$=$	$34 \cdot 7 + 6$	$(0, 1),\ (1, -2),\ (-34, 69)$
7	$=$	$1 \cdot 6 + 1$	$(1, -2),\ (-34, 69),\ (35, -71)$
6	$=$	$6 \cdot 1 + 0$	

So $\gcd(495, 244) = 1$ (the moduli are coprime, so CRT applies) and

$$495 \cdot 35 + 244 \cdot (-71) = 1.$$

Hence $495^{-1} \equiv 35 \pmod{244}$ and $244^{-1} \equiv -71 \equiv 424 \pmod{495}$.

Step 2: build the solution. The CRT formula is

$$x \equiv 3 \cdot \big(495 \cdot 495^{-1} \bmod 244\big) + 17 \cdot \big(244 \cdot 244^{-1} \bmod 495\big) \pmod{244 \cdot 495}.$$

The first bracket is $495 \cdot 35 = 17325$, which is $\equiv 1 \pmod{244}$ and $\equiv 0 \pmod{495}$; the second is $244 \cdot 424 = 103456$, which is $\equiv 0 \pmod{244}$ and $\equiv 1 \pmod{495}$. So

$$x \equiv 3 \cdot 17325 + 17 \cdot 103456 = 51975 + 1758752 = 1810727 \equiv 119807 \pmod{120780},$$

since $244 \cdot 495 = 120780$ and $1810727 - 14 \cdot 120780 = 119807$.

Alternatively (the substitution method): write $x = 3 + 244k$. Then $3 + 244k \equiv 17 \pmod{495}$, i.e. $244k \equiv 14 \pmod{495}$, so $k \equiv 14 \cdot 244^{-1} \equiv 14 \cdot 424 = 5936 \equiv 491 \pmod{495}$. Then $x = 3 + 244 \cdot 491 = 119807$.

Check: $119807 = 491 \cdot 244 + 3$ and $119807 = 242 \cdot 495 + 17$. ✓

Problem 3: one more CRT example

Any coprime moduli work; here is $x \equiv 2 \pmod 7$, $x \equiv 5 \pmod{12}$. Extended Euclid on 12 and 7:

$$12 = 1 \cdot 7 + 5, \quad 7 = 1 \cdot 5 + 2, \quad 5 = 2 \cdot 2 + 1, \quad \text{cards } (1, -1),\ (-1, 2),\ (3, -5),$$

so $12 \cdot 3 + 7 \cdot (-5) = 1$: $12^{-1} \equiv 3 \pmod 7$ and $7^{-1} \equiv -5 \equiv 7 \pmod{12}$. Then

$$x \equiv 2 \cdot (12 \cdot 3) + 5 \cdot (7 \cdot 7) = 72 + 245 = 317 \equiv 65 \pmod{84}.$$

Check: $65 = 9 \cdot 7 + 2$ and $65 = 5 \cdot 12 + 5$. ✓ (The CRT Demo page, with moduli 7 and 12, lands on 65 too.)

Problem 4: Sunzi's problem

We want $x \equiv 2 \pmod 3$, $x \equiv 3 \pmod 5$, $x \equiv 2 \pmod 7$. The moduli are pairwise coprime, so there is a unique answer modulo $3 \cdot 5 \cdot 7 = 105$.

Do it two congruences at a time. First, $x \equiv 2 \pmod 3$ and $x \equiv 3 \pmod 5$: the numbers $\equiv 3 \pmod 5$ are $3, 8, 13, \ldots$, and $8 \equiv 2 \pmod 3$, so $x \equiv 8 \pmod{15}$. Next, $x \equiv 8 \pmod{15}$ and $x \equiv 2 \pmod 7$: the numbers $\equiv 8 \pmod{15}$ are $8, 23, 38, \ldots$, and $23 \equiv 2 \pmod 7$. So

$$x \equiv 23 \pmod{105}.$$

(Or with the formula: $x \equiv 2 \cdot 35 \cdot 35^{-1}_{(3)} + 3 \cdot 21 \cdot 21^{-1}_{(5)} + 2 \cdot 15 \cdot 15^{-1}_{(7)} = 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 140 + 63 + 30 = 233 \equiv 23 \pmod{105}$.) The smallest answer is 23 things: $23 = 7 \cdot 3 + 2 = 4 \cdot 5 + 3 = 3 \cdot 7 + 2$. ✓

Problem 5: non-coprime moduli

$x \equiv 3 \pmod 6$, $x \equiv 5 \pmod 8$. Here $\gcd(6, 8) = 2$. Both conditions force x to be odd ($3 \equiv 5 \equiv 1 \pmod 2$), so there is no immediate contradiction. Listing $x \equiv 5 \pmod 8$: $5, 13, 21, \ldots$; and $21 \equiv 3 \pmod 6$. So $x = 21$ works, and the solutions are $x \equiv 21 \pmod{24}$ (check: $21 = 3 \cdot 6 + 3 = 2 \cdot 8 + 5$).

$x \equiv 1 \pmod 6$, $x \equiv 2 \pmod 8$. The first says x is odd, the second says x is even. **No solution.**

In general. For $x \equiv a \pmod m$, $x \equiv b \pmod n$ with $g = \gcd(m, n)$: any solution satisfies $x \equiv a \pmod g$ and $x \equiv b \pmod g$, so we need $a \equiv b \pmod g$. Conversely, if $a \equiv b \pmod g$, write $x = a + mk$; then $mk \equiv b - a \pmod n$, and since $g \mid (b - a)$ this reduces to $(m/g)k \equiv (b - a)/g \pmod{n/g}$, which has a unique solution for k modulo n/g because $\gcd(m/g, n/g) = 1$. So:

- a solution exists if and only if $a \equiv b \pmod{\gcd(m, n)}$;
- when it exists, it is unique modulo $\operatorname{lcm}(m, n)$.

For $m = 6$, $n = 8$: gcd = 2, lcm = 24. The first system has exactly **one** solution modulo 24 (namely 21); the second has **none**. (On the CRT Demo page, the greyed-out teeth on the mod-48 dial are exactly the residues the gears can never reach.)

Problem 6: inverting by Fermat

Since 61 is prime and $61 \nmid 35$, Fermat says $35^{60} \equiv 1 \pmod{61}$, so $35 \cdot 35^{59} \equiv 1$: the inverse is $35^{59} \bmod 61$. By successive squaring,

$$35^1 \equiv 35, \quad 35^2 \equiv 5, \quad 35^4 \equiv 25, \quad 35^8 \equiv 15, \quad 35^{16} \equiv 42, \quad 35^{32} \equiv 56 \pmod{61},$$

and $59 = 32 + 16 + 8 + 2 + 1$, so

$$35^{59} \equiv 56 \cdot 42 \cdot 15 \cdot 5 \cdot 35 \equiv 7 \pmod{61},$$

matching Problem 1. ✓

Which to prefer? For a 300-digit prime, Fermat needs about $\log_2 p \approx 1000$ squarings and up to 1000 multiplications of 300-digit numbers; the extended Euclidean algorithm needs at most about $2 \log_2 p \approx 2000$ division steps, but on numbers that *shrink* as it goes, and it is a few times faster in practice. Both are polynomial time; most software uses extended Euclid (or a variant). Also, Fermat silently gives a wrong answer if p is not actually prime, whereas Euclid also tells you the gcd.

Modulo a composite n: the extended Euclidean algorithm still works, whenever $\gcd(a, n) = 1$. The Fermat trick does not: a^{n-2} is not the inverse in general (one would need Euler's theorem, $a^{\varphi(n)-1}$, and computing $\varphi(n)$ requires factoring n).

Problem 7: CRT speeds up RSA

Alice's key: $n = 3233 = 61 \cdot 53$, $e = 17$, $d = 2753$, ciphertext $c = 2790$.

Why we may reduce the exponent. Since $\gcd(c, 61) = 1$, Fermat gives $c^{60} \equiv 1 \pmod{61}$, so only $d \bmod 60$ matters modulo 61; likewise $c^{52} \equiv 1 \pmod{53}$, so only $d \bmod 52$ matters modulo

53. (If $61 \mid c$ then both sides are 0 and the conclusion still holds.) Here $d = 2753 = 45 \cdot 60 + 53 = 52 \cdot 52 + 49$, so

$$d \equiv 53 \pmod{60}, \qquad d \equiv 49 \pmod{52}.$$

The two small computations. $c \equiv 2790 \equiv 45 \pmod{61}$ and $c \equiv 2790 \equiv 34 \pmod{53}$, so

$$m \equiv 45^{53} \equiv 4 \pmod{61}, \qquad m \equiv 34^{49} \equiv 12 \pmod{53}.$$

Combine with CRT. Extended Euclid on 61 and 53: $61 = 1 \cdot 53 + 8$, $53 = 6 \cdot 8 + 5$, $8 = 1 \cdot 5 + 3$, $5 = 1 \cdot 3 + 2$, $3 = 1 \cdot 2 + 1$, with final card $(20, -23)$: $61 \cdot 20 - 53 \cdot 23 = 1$. So $61^{-1} \equiv 20 \pmod{53}$ and $53^{-1} \equiv -23 \equiv 38 \pmod{61}$, and

$$m \equiv 4 \cdot (53 \cdot 38) + 12 \cdot (61 \cdot 20) = 4 \cdot 2014 + 12 \cdot 1220 = 8056 + 14640 = 22696 \equiv 65 \pmod{3233}.$$

(Quicker by eye: $m \equiv 4 \pmod{61}$ means $m \in \{4, 65, 126, \ldots\}$, and $65 \equiv 12 \pmod{53}$.) **Check:** $2790^{2753} \bmod 3233 = 65$ directly. ✓ (And 65 is the message Alice signed on the daily due Sep 28.)

Why faster? Modular exponentiation costs about $\log d$ multiplications of numbers the size of the modulus, and multiplying k-digit numbers costs about k^2 (schoolbook). Working modulo p and q (half the digits each) makes each multiplication about 4 times cheaper, and the exponents are half as long, so each of the two half-size exponentiations is about 8 times cheaper than the full one: roughly a $4\times$ speedup overall, for a single cheap CRT at the end. Real RSA implementations do exactly this (the private key file even stores $d \bmod (p-1)$, $d \bmod (q-1)$ and $q^{-1} \bmod p$).

Problem 8: Håstad's broadcast attack

Moduli $n_1 = 55 = 5 \cdot 11$, $n_2 = 391 = 17 \cdot 23$, $n_3 = 1189 = 29 \cdot 41$ (pairwise coprime, as RSA moduli from different people will be), all with $e = 3$; ciphertexts $c_1 = 3$, $c_2 = 189$, $c_3 = 370$.

CRT. $N = n_1 n_2 n_3 = 25569445$. Two at a time: $X = 3 + 55k$ with $55k \equiv 186 \pmod{391}$; $55^{-1} \equiv 64 \pmod{391}$, so $k \equiv 186 \cdot 64 \equiv 174$ and $X \equiv 3 + 55 \cdot 174 = 9573 \pmod{21505}$. Then $X = 9573 + 21505j$ with $21505j \equiv 370 - 9573 \equiv 309 \pmod{1189}$; $21505^{-1} \equiv 658 \pmod{1189}$, so $j \equiv 309 \cdot 658 \equiv 3$, and

$$X \equiv 9573 + 3 \cdot 21505 = 74088 \pmod{25569445}.$$

Check: $74088 \equiv 3 \pmod{55}$, $\equiv 189 \pmod{391}$, $\equiv 370 \pmod{1189}$. ✓

Why $X = m^3$ exactly. The integer m^3 satisfies $m^3 \equiv c_i \pmod{n_i}$ for each i, so $m^3 \equiv X \pmod{N}$ by uniqueness in CRT. But $m < n_i$ for each i (a message must be smaller than the modulus), so $m^3 < n_1 n_2 n_3 = N$. Both X and m^3 lie in $[0, N)$ and are congruent modulo N, so they are *equal as integers*. Hence $m = \sqrt[3]{X}$ is an ordinary real cube root, no modular arithmetic needed:

$$m = \sqrt[3]{74088} = 42.$$

(Check: $42^3 = 74088$, and $42^3 \bmod 55 = 3$, $\bmod 391 = 189$, $\bmod 1189 = 370$.)

Moral. Sending the same message to e (or more) recipients with a small public exponent e is fatal: e ciphertexts let Eve recover m^e as an integer and take an integer e-th root. The fix is the usual one: **pad** each message with different random bits before encrypting (so the three plaintexts differ), and/or avoid tiny e (the common choice $e = 65537$ makes this attack need 65537 recipients).