Preparing for Midterm #2

MATH 481/581 — Cryptography
Midterm: Friday, Mar 29, 2024

Please print your name:

Bonus challenge. Let me know about any typos you spot in the posted solutions (or lecture sketches). Any typo, that is not yet fixed by the time you send it to me, is worth a bonus point.

Problem 1.

(a) If you can only do a single modular computation, how would you check whether a huge randomly selected number N is prime or not?

(b) Which flaw of the Fermat primality test renders it unsuitable as a general primality test? How can this flaw be fixed?

(c) Despite the flaw in the previous item, in which scenario is it fine to use the Fermat primality test regardless?

(d) We want to use the Miller–Rabin primality test to decide whether $N = 377$ is prime. Each time, we randomly choose a base a (and only do a single iteration of Miller–Rabin) and compute the following:

- $a = 12$: $12^{47} \equiv 220$, $12^{94} \equiv 144$, $12^{188} \equiv 1$, $12^{376} \equiv 1 \pmod{377}$
- $a = 70$: $70^{47} \equiv 307$, $70^{94} \equiv 376$, $70^{188} \equiv 1$, $70^{376} \equiv 1 \pmod{377}$
- $a = 80$: $80^{47} \equiv 332$, $80^{94} \equiv 140$, $80^{188} \equiv 373$, $80^{376} \equiv 16 \pmod{377}$
- $a = 233$: $233^{47} \equiv 233$, $233^{94} \equiv 1$, $233^{188} \equiv 1$, $233^{376} \equiv 1 \pmod{377}$

In each case, what do we conclude? (Also point out which calculations were unnecessary.) Which of the a are strong liars? Which are Fermat liars?

(e) Repeat the previous problem for $N = 247$ and the following computations:

- $a = 12$: $12^{123} \equiv 246$, $12^{246} \equiv 1 \pmod{247}$
- $a = 17$: $17^{123} \equiv 64$, $17^{246} \equiv 144 \pmod{247}$
- $a = 27$: $27^{123} \equiv 170$, $27^{246} \equiv 1 \pmod{247}$
- $a = 68$: $68^{123} \equiv 1$, $68^{246} \equiv 1 \pmod{247}$

Solution.

(a) Compute $2^{N-1} \pmod{N}$ (using binary exponentiation). If this is $2^{N-1} \not\equiv 1 \pmod{N}$, then N is not a prime. [There's nothing special about 2, by the way.]

Otherwise, N is a prime or 2 is a Fermat liar modulo N (but the latter is exceedingly unlikely for a huge randomly selected number N; a bonus challenge from class indicates that this is almost as unlikely as randomly running into a factor of N).

(b) There exist composite numbers n such that every residue a is either a Fermat liar or $\gcd(a, n) > 1$ (in which case, a reveals a factor of n, which is as unlikely as finding a divisor of n by trial division). For these numbers (called absolute pseudoprimes) the Fermat primality test would usually suggest the wrong conclusion that the number is a prime.

The issue is fixed by the Miller–Rabin primality test, an extension of the Fermat primality test.

(c) When testing a large randomly generated number for primality. The reason is that Fermat liars are extremely rare among large numbers.

(d) Only the following computations are necessary to reach the stated conclusion:

- $a = 12$: $12^{47} \equiv 220$, $12^{94} \equiv 144$, $12^{188} \equiv 1$ $\rightsquigarrow$ 377 is not a prime
- $a = 70$: $70^{47} \equiv 307$, $70^{94} \equiv 376 \equiv -1$ $\rightsquigarrow$ 377 is likely a prime
- $a = 80$: $80^{47} \equiv 332$, $80^{94} \equiv 140$, $80^{188} \equiv 373$ $\rightsquigarrow$ 377 is not a prime

  [Note that we don't need to compute 80^{376}. Make sure you understand why!]
- $a = 233$: $233^{47} \equiv 233$, $233^{94} \equiv 1$ $\rightsquigarrow$ 377 is not a prime

The computations show that 70 is a strong liar modulo 377, and that $12, 70, 233$ are Fermat liars modulo 377.

(e) Only the following computations are necessary to reach the stated conclusion:

- $a = 12$: $12^{123} \equiv 246 \equiv -1$ $\rightsquigarrow$ 247 is likely a prime
- $a = 17$: $17^{123} \equiv 64$ $\rightsquigarrow$ 247 is not a prime
- $a = 27$: $27^{123} \equiv 170$ $\rightsquigarrow$ 247 is not a prime
- $a = 68$: $68^{123} \equiv 1$ $\rightsquigarrow$ 247 is likely a prime

The computations show that $12, 68$ are strong liars modulo 247, and that $12, 27, 68$ are Fermat liars modulo 247.

Problem 2. Bob's public RSA key is $N = 65$, $e = 5$.

(a) Encrypt the message $m = 10$ and send it to Bob.

(b) Determine Bob's secret private key d.

(c) You intercept the message $c = 2$ from Alice to Bob. Decrypt it using the secret key.

Solution.

(a) The ciphertext is $c = m^e \pmod{N}$. Here, $c \equiv 10^5 \pmod{65}$

$10^2 \equiv 35 \equiv -30$, $10^4 \equiv 30^2 \equiv 55 \pmod{65}$. Hence, $10^5 = 10^4 \cdot 10 \equiv 55 \cdot 10 \equiv 30 \pmod{65}$. Hence, $c = 30$.

(b) $N = 5 \cdot 13$, so that $\phi(N) = 4 \cdot 12 = 48$.

To find d, we compute $e^{-1} \pmod{48}$ using the extended Euclidean algorithm:

$$\begin{array}{rlrcl} & \gcd(5, 48) & \boxed{48} & = & 10 \cdot \boxed{5} - 2 \\ = & \gcd(2, 5) & \boxed{5} & = & 2 \cdot \boxed{2} + 1 \\ = & 1 & & & \end{array}$$

Backtracking through this, we find that Bézout's identity takes the form

$$1 = \boxed{5} - 2 \cdot \boxed{2} = \boxed{5} - 2 \cdot \left(10 \cdot \boxed{5} - \boxed{48}\right) = -19 \cdot \boxed{5} + 2 \cdot \boxed{48}.$$

Hence, $5^{-1}\equiv -19\equiv 29 \pmod{48}$ and, so, $d=29$.

Advanced comment. Actually, as discussed in class, $\phi(N)=(p-1)(q-1)=48$ can effectively be replaced with $\operatorname{lcm}(p-1,q-1)=12$. That is, $d=5^{-1}\equiv 5 \pmod{12}$ serves as private key as well (note that $29\equiv 5 \pmod{12}$).

(c) We need to compute $m=c^d \pmod{N}$, that is, $m=2^{29} \pmod{65}$.

$2^2=4$, $2^4=16$, $2^8\equiv 61\equiv -4$, $2^{16}\equiv 16 \pmod{65}$. Hence, $2^{29}=2^{16}\cdot 2^8\cdot 2^4\cdot 2\equiv 32 \pmod{65}$, so that $m=32$.

Comment. Let us check that the private key $d=5$ (as determined in the previous comment) can be used for decryption with the same effect. Indeed, $m=2^5=32 \pmod{65}$ as well.

Problem 3. Bob's public ElGamal key is $(p,g,h)=(61,10,21)$.

(a) Encrypt the message $m=11$ ("randomly" choose $y=17$) and send it to Bob.

(b) Break the cryptosystem and determine Bob's secret key.

(c) Use the secret key to decrypt $c=(13,7)$.

Solution. We only record the final answers. Make sure the necessary computations pose no challenge to you.

(a) The ciphertext is $c=(c_1,c_2)$ with $c_1=g^y \pmod{p}$ and $c_2=h^y m \pmod{p}$.

Here, $c_1=10^{17}\equiv 59 \pmod{61}$ and $c_2=21^{17}\cdot 11\equiv 29\cdot 11\equiv 14 \pmod{61}$. Hence, the ciphertext is $c=(59,14)$.

(b) We need to solve $10^x\equiv 21 \pmod{61}$. This yields $x=5$.

(Since we haven't learned a better method (no "good" method is known!), you can just try $x=1,2,3,\ldots$ until you find the right one.)

(c) We decrypt $m=c_2c_1^{-x} \pmod{p}$.

Here, $m=7\cdot 13^{-5}\equiv 30 \pmod{61}$.

Problem 4.

(a) For his public RSA key, Bob has selected $N=91$. What is the smallest choice for e with $e\geqslant 2$?

(b) How many primitive roots are there modulo 13? Determine all of them.

(c) Find x such that $9\equiv 7^x \pmod{13}$.

(d) For his public ElGamal key, Bob has selected $p=61$. How many possible choices does he have for g?

(e) Alice and Bob select $p=61$ and $g=55$ for a Diffie–Hellman key exchange. Alice sends 32 to Bob, and Bob sends 54 to Alice. What is their shared secret?

(f) Determine the multiplicative orders of 2, 4, 8, 16 modulo 61. Are any of these primitive roots? How many primitive roots are there in total?

(g) Spell out the computational Diffie–Hellman problem as well as the decisional Diffie–Hellman problem. Which of these is more difficult?

(h) For his public RSA key, Bob needs to select p,q and e. Which of these must be chosen randomly?

(i) For his public ElGamal key, Bob needs to select p,g and x. Which of these must be chosen randomly?

(j) When using vanilla RSA, why must we never directly encrypt messages that can be predicted (like "yes", "no", "maybe"; or a social security number)?

Solution.

(a) Recall that e must be invertible modulo $\phi(N) = 6 \cdot 12$. Hence, $e = 2, 3, 4$ are not allowed.

Therefore, the smallest possible choice for e is $e = 5$.

(b) Recall that the number of primitive roots modulo a prime p is $\phi(\phi(p)) = \phi(p - 1)$.

Here, there are $\phi(12) = 4$ primitive roots modulo 13.

To find a first primitive root, we try $g = 2$ (if that doesn't work, we move on to $g = 3$, $g = 4$, $\ldots$). $g = 2$ is a primitive root if its order is 12. Since the order must divide 12, it is enough to check that $2^4 \not\equiv 1 \pmod{13}$ and $2^6 \not\equiv 1 \pmod{13}$ [because then automatically $2^2 \not\equiv 1 \pmod{13}$ and $2^3 \not\equiv 1 \pmod{13}$]. Indeed $2^4 \equiv 3 \pmod{13}$ and $2^6 \not\equiv -1 \pmod{13}$, so that $g = 2$ is a primitive root.

Now it is easy to list all 4 primitive roots: $2^1, 2^5, 2^7, 2^{11} \pmod{13}$ (the exponents are the invertible residues modulo 12). Explicitly computing these powers, the primitive roots are $2, 6, 7, 11 \pmod{13}$. [$2^5 \equiv 6$, $2^7 \equiv 11$, $2^{11} \equiv 7$]

(c) Since we haven't learned a better method, we just try $x = 1, 2, 3, \ldots$ until we find the right one. We find $x = 4$.

Comment. Since 7 is a primitive root modulo 13, we know that the most general solution is $x \equiv 4 \pmod{12}$.

(d) Since g must be a primitive root modulo p, Bob has $\phi(\phi(p)) = \phi(p - 1)$ many choices for g.

Here, Bob has $\phi(60) = 16$ choices.

(e) If Alice's secret is y and Bob's secret is x, then $55^y \equiv 32$ and $55^x \equiv 54 \pmod{61}$.

We compute $55^2, 55^3, \ldots$ until we find either 32 or 54:

$55^2 \equiv 36$, $55^3 \equiv 28$, $55^4 \equiv 15$, $55^5 \equiv 32 \pmod{61}$.

Hence, Alice's secret is $y = 5$. The shared secret is $(55^x)^y \equiv 54^5 \equiv 29 \pmod{61}$.

(f) The total number of primitive roots is $\phi(\phi(61)) = \phi(60) = \phi(4)\phi(3)\phi(5) = 2 \cdot 2 \cdot 4 = 16$.

We note that the multiplicative order of elements modulo 61 must divide $\phi(61) = 60 = 2^2 \cdot 3 \cdot 5$. If the order divides 60 and is not equal to 60, then it must divide either 30, 20 or 12 (make sure you see that!). Using some binary exponentiation, we find $2^{30} \equiv -1$, $2^{20} \equiv 47$, $2^{12} \equiv 9$ modulo 61. Hence, 2 must have order 60. We conclude that 2 is a primitive root.

Recall that if x has order n, then x^r has order $n / \gcd(n, r)$. Therefore, $4 = 2^2$ has order $60 / \gcd(60, 2) = 30$, $8 = 2^3$ has order $60 / \gcd(60, 3) = 20$, and $16 = 2^4$ has order $60 / \gcd(60, 4) = 15$. None of these are primitive roots.

Comment. The 16 primitive roots are $2, 2^7, 2^{11}, 2^{13}, \ldots$ where the exponents are coprime to 60.

(g) The CDH problem is the following: given $g, g^x, g^y \pmod{p}$, find $g^{xy} \pmod{p}$.

The DDH problem is the following: given $g, g^x, g^y, r \pmod{p}$, decide whether $r \equiv g^{xy} \pmod{p}$.

Obviously, DDH is simpler than the CDH problem.

(h) p and q must be chosen randomly.

(i) x must be chosen randomly.

(j) Because an attacker can make a list of likely messages (for instance, a list of all possible social security numbers) and encrypt all of them using the public key. As soon as one of these matches the ciphertext, the attacker has broken the message.

Problem 5. Consider the finite field $\mathrm{GF}(2^4)$ constructed using x^4+x+1.

(a) Add and multiply x^2+1 and x^2+x+1 in $\mathrm{GF}(2^4)$.

(b) What is the inverse of x^2+x+1 in $\mathrm{GF}(2^4)$?

(c) What is the inverse of x^3+x in $\mathrm{GF}(2^4)$?

Solution.

(a) $(x^2+1)+(x^2+x+1)=x$ in $\mathrm{GF}(2^4)$.

$(x^2+1)\cdot(x^2+x+1)=x^3$ in $\mathrm{GF}(2^4)$. This is because $(x^2+1)\cdot(x^2+x+1)=x^4+x^3+2x^2+x+1$, which reduces to x^4+x^3+x+1 modulo 2. Further, reducing modulo x^4+x+1, we are left with x^3. (Here, we can just subtract x^4+x+1. In general, we would do polynomial division by x^4+x+1 and take the remainder.)

(b) In general, we use the extended Euclidean algorithm and reduce modulo 2 at each step. Here, we are lucky and are actually done after a single polynomial division:

$$\boxed{x^4+x+1} \equiv (x^2+x)\cdot\boxed{x^2+x+1}+1$$

Hence, $(x^2+x+1)^{-1}=x^2+x$ in $\mathrm{GF}(2^4)$.

(c) We use the extended Euclidean algorithm, and always reduce modulo 2:

$$\begin{aligned}
\boxed{x^4+x+1} &\equiv x\cdot\boxed{x^3+x}+(x^2+x+1)\\
\boxed{x^3+x} &\equiv (x+1)\cdot\boxed{x^2+x+1}+(x+1)\\
\boxed{x^2+x+1} &\equiv x\cdot\boxed{x+1}+1
\end{aligned}$$

Backtracking through this, we find that Bézout's identity takes the form

$$\begin{aligned}
1 &\equiv \boxed{x^2+x+1}+x\cdot\boxed{x+1}\\
&\equiv \boxed{x^2+x+1}+x\cdot\left(\boxed{x^3+x}+(x+1)\cdot\boxed{x^2+x+1}\right)\equiv x\cdot\boxed{x^3+x}+(x^2+x+1)\cdot\boxed{x^2+x+1}\\
&\equiv x\cdot\boxed{x^3+x}+(x^2+x+1)\cdot\left(\boxed{x^4+x+1}+x\cdot\boxed{x^3+x}\right)\equiv(x^2+x+1)\cdot\boxed{x^4+x+1}+(x^3+x^2)\cdot\boxed{x^3+x}
\end{aligned}$$

Hence, $(x^3+x)^{-1}=x^3+x^2$ in $\mathrm{GF}(2^4)$.

Problem 6.

(a) The design of a block cipher is almost an art, but there are two guiding principles due to Claude Shannon, the father of information theory.

- What are these two principles? Briefly explain what they refer to.
- Which of these are the classical ciphers lacking?

(b) In a Feistel cipher, how does the encryption in one round look like?

Can any function be used in this construction?

How does decryption work?

Solution.

(a) The two principles are confusion and diffusion.

Confusion refers to making the relationship between the ciphertext and the key as complex and involved as possible (for instance, changing one bit of the key should change the ciphertext completely).

Diffusion refers to dissipating the statistical structure of the plaintext over the bulk of the ciphertext (for instance, changing one bit of the plaintext should change the ciphertext completely; likewise, changing one bit of the ciphertext should change the plaintext completely).

Diffusion is completely missing in the classical ciphers we discussed. Changing bits of the plaintext only changes corresponding parts of the ciphertext. That's why frequency analysis can break these ciphers so easily.

(b) Let us describe one round of a Feistel cipher which takes m and produces $R_k(m)$. Here, k is the round key.

- Split the plaintext m into two halves (L_0, R_0).
- Set $L_1 = R_0$ and $R_1 = L_0 \oplus f_k(R_0)$.
- Then, $R_k(m)$ is (L_1, R_1).

The function $f_k(x)$ is referred to as the round function. It can be any function (taking the appropriate amount of input bits, and producing the same number of output bits).

To obtain $m = (L_0, R_0)$ from $R_k(m) = (L_1, R_1)$, we set $R_0 = L_1$ and then compute $L_0 = R_1 \oplus f_k(R_0)$.

Problem 7.

(a) What is the block size of DES? What is the key size? How many rounds?

(b) What does each S-box do?

To store an S-box in DES as a lookup table, how many bytes are needed?

(c) How many bits are the round keys? How are they obtained?

(d) How does 3DES encryption work? What is the key?

What is the effective key size and why is it different?

(e) Why is there no 2DES?

(f) To (naively) brute-force DES, how much data must we encrypt?

Solution.

(a) The block size of DES is 64 bits. Its key size is 56 bits. It consists of 16 rounds.

(b) The S-boxes (there is eight different ones) are lookup tables. For each 6 bit input (meaning there is a total of 2^6 possible inputs), they specify 4 bits of output.

To store one S-box, we therefore need to list $2^6 \cdot 4 = 256$ bits, or 32 bytes.

(c) Each round key is 48 bits. Each of these 48 bits is taken (in a prescribed manner) from one of the 56 bits of the DES key.

(d) 3DES consists of three applications of DES

$$c = E_{k_3}(D_{k_2}(E_{k_1}(m)))$$

The 3DES standard allows three keying options for the key $k = (k_1, k_2, k_3)$:

- k_1, k_2, k_3 independent keys: $3 \times 56 = 168$ key size, but effective key size is 112 bit

- $k_1 = k_3$: $2 \times 56 = 112$ bit key size, effective key size is stated as 80 bit by NIST
- $k_1 = k_2 = k_3$: this is just the usual DES, and provides backwards compatibility (which is a major reason for making the middle step a decryption instead of another encryption).

The reason for the reduced effective key sizes is the meet-in-the-middle attack.

(e) The meet-in-the-middle attack is also the reason why 2DES does not provide significantly increased security over DES.

(f) DES uses 56 bit keys and has a 64 bit block size.

Hence, given m and c, to make a list of all possible $E_k(m)$ (to check for which k we have $E_k(m) = c$), we need to encrypt 2^{56} times 64 bits.

This is $2^{56} \cdot 8 = 2^{59}$ byte, or 512 pebibyte (binary analog of petabyte) or 576 petabyte (since $2^{59} \approx 5.76 \cdot 10^{17}$).

Problem 8.

(a) What is the block size of AES? What is the key size? How many rounds?

(b) How is it possible that AES uses fewer rounds than DES?

(c) What are the four layers that each round consists of?

(d) Which layer makes AES highly nonlinear? Describe the crucial mathematical operation involved in this layer.

(e) To store the ByteSub layer of AES as a lookup table, how many bytes are needed?

Solution.

(a) The block size of AES is 128 bits. Its key size is 128/192/256 bits. It consists of 10/12/14 rounds.

(b) Unlike DES, AES is not a Feistel network. While for a Feistel network, each round only encrypts half of the bits, all bits are being encrypted during each round of AES. That's one indication why AES requires fewer rounds than DES.

(c) The 4 layers are:

- ByteSub (each byte gets substituted with another byte (like a single S-box in DES); provides confusion)
- ShiftRow (the 16 bytes are permuted (like a P-box in DES but on bytes, not bits); provides diffusion)
- MixCol (each column in the 4x4 matrix is linearly transformed; provides diffusion)
- AddRoundKey (the state is xored with a 128 bit round key)

(d) The ByteSub layer is highly nonlinear (while all other layers are linear; assuming we adjust the key schedule accordingly).

For ByteSub an input byte y is interpreted as an element of the finite field $\mathrm{GF}(2^8)$. Then y^{-1} is computed in $\mathrm{GF}(2^8)$. This is the crucial and highly nonlinear operation. (The final output of ByteSub is another linear transformation of these 8 bits.)

(e) As the name indicates, ByteSub takes a byte and substitutes it with another byte. Since we have $2^8 = 256$ inputs, with 1 byte of output each, the corresponding lookup table is 256 bytes large.

Problem 9. Consider a block cipher with 5 bit block size and 5 bit key size such that

$$E_k(b_1b_2b_3b_4b_5) = (b_2b_5b_4b_3b_1) \oplus k.$$

(a) Encrypt $m = (010101010101010\ \ldots)_2$ using $k = (10001)_2$ and ECB mode.

(b) Encrypt $m = (010101010101010\ \ldots)_2$ using $k = (10001)_2$ and CBC mode ($\mathrm{IV} = (10011)_2$).

Solution. $m = m_1m_2m_3\ldots$ with $m_1 = 01010$, $m_2 = 10101$ and $m_3 = 01010$.

(a) $c_1 = E_k(m_1) = 10100 \oplus 10001 = 00101$

$c_2 = E_k(m_2) = 01011 \oplus 10001 = 11010$

Since $m_3 = m_1$, we have $c_3 = c_1$. Hence, the ciphertext is $c = c_1c_2c_3\ldots = (00101\ 11010\ 00101\ \ldots)$.

(b) $c_0 = 10011$

$c_1 = E_k(m_1 \oplus c_0) = E_k(01010 \oplus 10011) = E_k(11001) = 11001 \oplus 10001 = 01000$

$c_2 = E_k(m_2 \oplus c_1) = E_k(10101 \oplus 01000) = E_k(11101) = 11011 \oplus 10001 = 01010$

$c_3 = E_k(m_3 \oplus c_2) = E_k(01010 \oplus 01010) = E_k(00000) = 00000 \oplus 10001 = 10001$

Hence, the ciphertext is $c = c_0c_1c_2c_3\ldots = (10011\ 01000\ 01010\ 10001\ \ldots)$.