Exam – Cryptology 1 – 01410

15.05.2023

With solutions

Instructions and advice

- For all computations in Part III: explain how you have computed your results. Unexplained results will receive few or no points. If you use a computer (or similar), then you need to be able to explain how the computer arrived at the answer.
- The problems have been created such that it is possible to solve them without the help of a computer or similar.
- Read all the questions first, and begin to work on the ones you find easy.

Part I – Select all that apply (18 points)

You get 2 points for every correct selection and -2 points for every incorrect selection, but never less then 0 points for a question. As for any multiple choice quiz, here only the selection counts. There is no need to describe your reasoning.

1. Which of the following equations involving modular arithmetic hold? As in the lecture, "=" is used for modular arithmetic, where "≡" is used in the book. Select all that apply.

   A. $7^2 = 7 \bmod 14$ true

   B. $5^{-1} = 8 \bmod 43$ false, 5*8=40 mod 43

   C. $5^{-1} = 9 \bmod 11$ true

   D. $5 \cdot 5 = -5 \bmod 29$ false, 5*5=-4 mod 29

   E. $14^{77} = 1 \bmod 64$ false, $8^1$7 is even mod 14

2. Which of the following statements about block ciphers are true? Select all that apply.

   A. A block cipher is an algorithm with exactly one input and one output. false, the block cipher encryption algorithm has two inputs, a key and a plaintext

   B. Block ciphers are only used for encrypting data at rest (Disk encryption etc.). For encrypted communication, public-key encryption is used. false, block ciphers are used virtually everywhere, example: TLS

   C. CBC mode is more secure than ECB mode. true

   D. The block length is always smaller than the key length. false, see e.g. AES192...

   E. ECB mode is more secure than CTR mode. false, ECB mode is insecure for almost any purpos and CTR is not

   F. Block ciphers are not, in general, safe for use as an encryption scheme. Only when combined with a mode of operation they can be used safely for encryption. true

G. AES is an iterated block cipher true

3. Which of the following statements about Diffie-Hellmann key exchange are true? Select all that apply.

   A. If Diffie-Hellman mod p is secure, the discrete logarithm problem mod p is hard. true

   B. Diffie-Hellmann uses arithmetic modulo a composite number. false, the modulus in DH is prime

   C. The public parameter g that is exponentiated by Alice's and Bob's secret exponents can be any number in $\mathbb{Z}_p^*$ false, it should be primitive

   D. Bob needs to wait until he has received Alice's protocol message before he sends his false, the two protocol messages can be sent simultaneously or asynchronously

   E. In the Diffie-Hellman protocol, Alice and Bob need to perform modular exponentiation. true

   F. Diffie-Hellman can be converted into a public-key encryption scheme by directly using the exchanged key in a one-time-pad-like encryption scheme. true

   G. The Diffie-Hellmann protocol consists of a key generation, an encryption and a decryption algorithm. false, DH is a key exchange protocol and not an encryption scheme.

4. Which statements about the Miller-Rabin test are true? Select all that apply.

   A. If the Miller-Rabin prime number generation algorithm finds during execution that n is not prime, n is not prime. true

   B. If the Miller-Rabin prime number generation algorithm finds during execution that n is prime, n is prime. false, there are false positives

   C. The Miller-Rabin algorithm is not randomized. false, Miller-Rabin is indeed randomized.

Part II – Select the right answer (8 points)

You get 4 points if (only) the right answer is selected. As for any multiple choice quiz, here only the selection counts. There is no need to describe your reasoning.

5. How many elements does $\mathbb{Z}_{77}^*$ have? Select the right answer.

   A. 66 B. 60 true, $77 = 7 \cdot 11$, so $\phi(77) = 6 \cdot 10 = 60$. C. 18 D. 70 E. 76 F. 15

6. Assume you want to brute-force a collision of the SHA2 variant SHA-512/224 with outputs of length 224 bits. How many evaluations of the function on random inputs do you require approximately? Select the most appropriate answer.

   A. 112 B. $\frac{2^{224}}{2}$ C. 2^{224} D. 224 E. 224^2 F. 2^{112} true, by the "birthday attack"

Part III (44 points)

7. (8 points) Recall the CTR mode of operation, where a block cipher e is used to encrypt the ith n-bit block m_i of a message $m = (m_1, m_2, ...)$ as

$$c_i = m_i \oplus \hat{k}_i^{\mathrm{CTR}}, \qquad (1)$$

where $\hat{k}_i^{\text{CTR}} = e_k(IV + i - 1 \bmod 2^n)$, $c_0 = IV$ is a uniformly random n-bit string, and in the expression $IV + i - 1 \bmod 2^n$, the string IV is interpreted as an integer between 0 and $2^n - 1$, and the result of the computation is regarded as a string again.

Output Feedback (OFB) mode is defined as follows. Define $\hat{k}_i^{\text{OFB}} = e_k(\hat{k}_{i-1}^{\text{OFB}})$ and $\hat{k}_0 = c_0 = IV$, a uniformly random string. Then for $i \geq 1$,

$$c_i = m_i \oplus \hat{k}_i \tag{2}$$

For CTR and OFB mode, the initial value $c_0 = IV$ is prepended to the ciphertext, i.e., it is considered to be the 0-th ciphertext block. For the sake of this problem, we call $\hat{k}_i^{\text{MOD}}$ the i-th one-time key of the mode MOD $\in \{\text{CTR}, \text{OFB}\}$.

(a) How is decryption done in OFB mode?

On input a ciphertext $c = (c_0, c_1, ...)$, first the key stream is recomputed via $\hat{k}_i^{\text{OFB}} = e_k(\hat{k}_{i-1}^{\text{OFB}})$, setting $\hat{k}_0 = c_0$. Then the ciphertext blocks are decrypted via $m_i = c_i \oplus \hat{k}_i$ and $m = (m_1, m_2...)$ is output.

(b) For some modes of operation, the evaluation of e_k can be parallelized across different blocks. Discuss how CTR and OFB modes differ in that respect.

For CTR, any number of $\hat{k}_i$ can be computed by generating the inputs $IV + i - 1 \bmod 2^n$ and performing the evaluations of e_k on these inputs in parallel. For OFB, the input for computing $\hat{k}_i$ is $\hat{k}_{i-1}$ which means the $\hat{k}_i$ *have* to be computed in sequence.

(c) If $\hat{k}_i^{\text{CTR}} = \hat{k}_j^{\text{CTR}}$ for $i \neq j$ and $i, j \leq \ell$, an adversary can learn something about a plaintext m with ℓ blocks by analyzing a ciphertext resulting from encrypting m with CTR mode. What is the minimal ℓ where this problem arises? What exactly can the adversary learn about m_i and m_j in that case?

If $i \neq j \bmod 2^n$ then $\hat{k}_i \neq \hat{k}_j$, so the problem arises as soon as $\ell > 2^n$. In case $\hat{k}_i = \hat{k}_j =: s$ we have $c_i \oplus c_j = m_i \oplus s \oplus m_j \oplus s = m_i \oplus m_j$, so the adversary learns the XOR of the two message blocks.

(d) For MOD $\in \{\text{CTR}, \text{OFB}\}$, let $\ell_{\max,\text{MOD}}$ be the maximal number of blocks such that it is still guaranteed that $\hat{k}_i^{\text{MOD}} \neq \hat{k}_j^{\text{MOD}}$ for all $i \neq j$ and $i, j \leq \ell$. In other words, $\ell_{\max,\text{MOD}}$ is the maximum number of blocks such that all one-time keys of the mode MOD with IV IV are distinct. Which of the following inequalities are true?

$$\ell_{\max,\text{CTR}} \leq \ell_{\max,\text{OFB}} \tag{3}$$

$$\ell_{\max,\text{OFB}} \leq \ell_{\max,\text{CTR}} \tag{4}$$

Explain your answer!

It holds that $\ell_{\max,\text{OFB}} \leq \ell_{\max,\text{CTR}}$, but in general the reverse inequality does not hold. We have already determined $\ell_{\max,\text{CTR}} = 2^n$, and this is actually the maximum possible value as there are only 2^n possible values for $\hat{k}_i$. For OFB, $ell_{\max,\text{OFB}}$ might be smaller, depending on the permutation e_k ($ell_{\max,\text{OFB}}$ is equal to the length of the cycle of e_k containing IV). Example: if there exists $x \neq IV$ such that $e_k(x) = x$, then $\ell_{\max,\text{OFB}} \leq 2^n - 1$.

8. (8 points) Let $p = 17, q = 13$ be two prime numbers and let $N = p \cdot q$.

(a) Consider all numbers from the set $\{0, \ldots, N-1\}$. Which of these numbers are not valid plaintexts for RSA as described in the lectures, i.e. are not from $\mathbb{Z}_N^*$?

The elements of $\mathbb{Z}^*$ are the numbers $m \in \{0, \ldots, N-1\}$ such that m is invertible mod N. This excludes $m = 0$, as well as any other m such that $\gcd(m, N) > 1$, i.e. $m = a \cdot q$ for $a = 1, 2, ..., p-1$ and $m = b \cdot p$ for $b = 1, 2, ..., q-1$.

(b) For the given modulus N, compute the ciphertext c for the encryption key $e = 5$ and message $m = 12$. Describe all steps of the encryption operation.

We need to compute $m^5 \bmod N$. We use square-and-multiply. We compute

$$m^2 = 144 \bmod 221$$

square once more,

$$m^4 = 144^2 = 20736 = 183 \bmod 221,$$

and now multiply

$$m^5 = m^4 \cdot m = 183 \cdot 12 = 2196 = 207 \bmod 221.$$

(c) For the encryption key $e = 5$ and modulus $N = 13 \cdot 17$ compute the secret RSA decryption key using the extended GCD algorithm. Describe all involved steps.

We use the extended GCD algorithm with inputs $\phi(N) = 12 * 16 = 192$ and $e = 5$.

i	q	r	s	s
-1		192	1	0
0		5	0	1
1	38	2	1	-38
2	2	1	-2	77

We read of the multiplicative inverse $5^{-1} = 77 \bmod 192$.

(d) Using the computed RSA decryption key d, decrypt the ciphertext $c = 11$.

We compute $11^{77} \bmod N$ using square-and-multiply. $77 = 1 + 2 * 2 * (1 + 2 * (1 + 2^3))$, so we compute

$$\begin{aligned}
11^2 &= 121 \bmod 221 \\
11^4 &= 121^2 = (-100)^2 = 10000 \bmod 221 = 55 \bmod 221 \\
11^8 &= 55^2 = 3025 = 152 \bmod 221 \\
11^9 &= 152 * 11 = 1672 = 125 \bmod 221 \\
11^{18} &= 125^2 = 15625 = 155 \bmod 221 \\
11^{19} &= 155 * 11 = 1705 = 158 \bmod 221 \\
11^{38} &= 155^2 = 24025 = 212 \bmod 221 \\
11^{76} &= 212^2 = (-9)^2 = 81 \bmod 221 \\
11^{77} &= 81 * 11 = 891 = 7 \bmod 221.
\end{aligned}$$

9. (6 points) Let $p = 23, q = 11$, $N = p \cdot q$ and $d = 17$ be a signing key for RSA signatures.

(a) Assume that we perform RSA signing without computing the hash function first (so the message is already in $\mathbb{Z}_N^*$.) For the message $m = 12$ and the given modulus and signing key, compute the RSA signature.

The signature is computed as $m^d \bmod N = 12^17 \bmod 11 \cdot 23 = 78$.

(b) Compute the public verification key and verify your computed signature from (a) as valid.

The public verification key is $e = d^{-1} \bmod \phi(N) = 17^{-1} \bmod 220 = 13$.

(c) Consider the following compression function H, which maps strings of length 512 to integers modulo N:

1. On input a bit string $\mathbf{x} \in \{0,1\}^{512}$ set the last bit of $\mathbf{x}$ to 0.
2. Compute $\mathbf{y} :=$ SHA3-256($\mathbf{x}$), i.e. apply the 256-bit output version of the SHA-3 family of hash functions to $\mathbf{x}$.

3. Interpret the binary string $\mathbf{y}$ as bits $b_0, b_1, \ldots, b_{255}$. Then let $z = \sum_{i=0}^{255} b_i 2^i \bmod N$ be the output of the compression function H.

Describe an attack that allows you to forge signatures for RSA-FDH which uses the compression function H, independent of the size of N. To show that your attack works, construct an explicit attacker against the definition of EUF-CMA security of the resulting signature scheme.

For any string $x \in \{0,1\}^*$, clearly $H(x\|0) = H(x\|1)$. We construct the following EUF-CMA attacker: The attacker queries the signing oracle on $m_0 = x\|0$ for some string x to obtain signature σ, and outputs $(x\|1, \sigma)$.

10. (6 points) Let $p = 47$ and $g = 11$.

(a) (2 points) Show that g has order 46 modulo 47, and explain why your answer is correct.

$46 = 2 * 23$, so to see that g has order 46, we check $11^2 = 121 = 27 \neq 1 \bmod 47$ and $11^23 = -1 \neq 1 \bmod 47$.

(b) (2 points) Assume Alice chooses a secret $a = 4$ and Bob chooses a secret $b = 7$ for the Diffie Hellman Key Exchange. Compute the messages that Alice and Bob exchange in the Diffie Hellman Key Exchange protocol, as well as the key that they both obtain in the end.

Alice sends $g^a = 11^4 = 24 \bmod 47$ and Bob sends $g^b = 11^7 = 31 \bmod 47$. The shared key is $k = g^{a \cdot b} = 31^4 = 18 \bmod 47$.

(c) (2 points) Alice and Bob repeatedly communicate with each other, and use Diffie Hellman Key Exchange to derive a new session key every time. But instead of always choosing new secrets a, b when running the key exchange, they always double their old values. So when Alice used the secret $a_1 = 4$ to compute the first key, she uses $a_2 = 8$ to derive the second key and $a_3 = 16$ to derive the third key etc. Likewise, Bob uses $b_1 = 7$ as his contribution to compute the first key, $b_2 = 14$ for the second key etc. Assume an attacker at some point learns the secret key k_i which they agreed on in round i. Can the attacker derive the key that Alice and Bob will use in round $i+1$, $i+2$ etc? Describe in detail how an attack would work, or why no attack is possible.

It is indeed possible to compute the keys k_j from key k_i, for $j > i$. Observe that $k_j = g^{a_j b_j} = g^{2^{j-i} a_i 2^{j-i} b_i} = k_i^{2^{2(j-i)}} \bmod p$, so an attacker can compute k_j from k_i using square-and-multiply.

11. (6 points) Let q be a prime such that $p = 2q + 1$ also is a prime. Moreover, let $g \in \mathbb{Z}_p$ be a number of order q modulo p. We consider the following encryption scheme:

Key Generation: Choose a random number $s \in \mathbb{Z}_q$. Define the public key $pk := g^s \bmod p$ and the secret key $sk := s$.

Encryption: To encrypt a bit $m \in \{0,1\}$, choose a random number $r \in \mathbb{Z}_q$. Then compute $c_0 := g^m \cdot pk^r \bmod p$ and $c_1 := g^r \bmod p$. Output the ciphertext $c = (c_0, c_1)$.

Decryption: For a ciphertext $c = (c_0, c_1)$ compute $h := c_0 \cdot c_1^{-s} \bmod p$. Output 0 if $h = 1$ and 1 otherwise.

(a) Show that the encryption scheme is correct, namely that after encrypting a bit m for a public key pk, the decrypted value is again m.

Encrypting a message m and compting the auxiliary quantity h in the decryption algorithm yields

$$h = c_0 \cdot c_1^{-s} = g^m \cdot pk^r \cdot (g^r)^{-s} = g^m \cdot g^{rs} \cdot g^{-rs} = g^m.$$

as g is of order q, we have $g^0 = 1$ and $g^1 = g \neq 1$, ensuring correctness.

(b) Describe the IND-CPA security game, adapted to the described encryption scheme. For this, write down explicitly which group elements the challenger generates.

- The challenger generates a key pair by chosing $sk = s \in \mathbb{Z}_q$ and computing the public key $pk = g^s \bmod p$, a group element.
- The challenger sends pk to the adversary
- The adversary submits messages m_0 and m_1. (This step can also be omitted for this particular encryption scheme as there are only two distinct messages, 0 and 1, so the challenger can also just set $m_i = i$ for $i = 0, 1$.)
- The challenger samples $b \leftarrow \{0, 1\}$ and encrypts m_b. To do so, the Challenger samples $r \leftarrow \mathbb{Z}_q$ and generates the group elements $c_0 = g^{m_b} \cdot pk^r \bmod p$ and $c_1 = g^r \bmod p$.
- The challenger sends $c = (c_0, c_1)$ to the adversary.
- The adversary outputs $b' \in \{0, 1\}$.

The adversary wins if $b = b'$.

(c) Assume that an attacker has access to an algorithm A which efficiently solves the Computational Diffie-Hellman problem for inputs from $\mathbb{Z}_p^*$. Show that an attacker using A can break the IND-CPA security of the encryption scheme that is described above.

Let $c = (c_0, c_1)$ be the challenge ciphertext given to the adversary. Given $c_1 = g^r \bmod p$ and $pk = g^s \bmod p$, A can be used to compute $t = g^{rs} \bmod p$. Now it is easy to decrypt the challenge ciphertext: If $c_0 \cdot t^-1 \bmod p = 1$ then c is a ciphertext for message 0, otherwise for message 1.

12. (10 points) The rounded version of a Gaussian random variable X with mean 0 and variance σ^2 is defined as $Y = \lceil X \rfloor$, where $\lceil \cdot \rfloor$ denotes the standard rounding operation. We denote the probability distribution of Y by $D_{\mathbf{Z}^1,\sigma}$. For the Gaussian random variable X the following tail bound holds. For any $x \geq 0$,

$$\Pr[X \geq x] \leq \frac{\sigma}{x \cdot \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}. \tag{5}$$

Recall Regev's encryption scheme (all arithmetic is modulo q except when specified otherwise):

- Secret key: $sk = \mathbf{s} \in \mathbb{Z}_q^n$
- Public key: pick $\mathbf{a}_i \in \mathbb{Z}_q^n$ independently uniformly at random and $e_i \leftarrow D_{\mathbf{Z}^1,\sigma}$, for $i = 1, ..., m$, and set $b_i = \mathbf{a}_i \cdot \mathbf{s} + e_i$ ($\mathbf{x} \cdot \mathbf{y}$ for vectors $\mathbf{x}$ and $\mathbf{y}$ denotes the inner product). The public key is $pk = (\mathbf{a}_i, b_i)_{i=1}^m$.
- Encryption Enc: To encrypt message $\mathsf{m} \in \{0, 1\}$, sample random bits $\iota_i \in \{0, 1\}$ for $i = 1, ..., m$ and set

$$\mathbf{c}_0 = \sum_{i=1}^{m} \iota_i \mathbf{a}_i, \qquad c_1 = \frac{\mathsf{m}(q-1)}{2} + \sum_{i=1}^{m} \iota_i b_i.$$

  The ciphertext is $c = (\mathbf{c}_0, c_1)$.
- Decryption Dec: To decrypt a ciphertext $c = (\mathbf{c}_0, c_1)$ using secret key $sk = \mathbf{s}$, compute $\mathsf{m}' = \left\lceil \frac{2(c_1 - \mathbf{s} \cdot \mathbf{c}_0)}{q} \right\rfloor$ (This computation is done regarding all involved numbers as reals).

(a) (6 points) Use the bound in Equation (5) to derive the bound

$$\Pr[e_1 \neq 0] \leq \frac{4\sigma}{\sqrt{2\pi}} e^{-\frac{1}{8\sigma^2}}. \tag{6}$$

You can use that e_1 is an integer, and thus

$$\Pr[e_1 \neq 0] = \Pr[e_1 \geq 1 \vee e_1 \leq -1] = \Pr[e_1 \geq 1] + \Pr[e_1 \leq -1]. \tag{7}$$

Let X be a Gaussian random variable such that $e_1 = \lceil X \rfloor$. Then $e_1 \geq 1$ exactly if $X \geq 1/2$ and $e_1 \leq -1$ exactly if $X < -1/2$. As the Gaussian dsitribution is a continuous distribution and symmetric around 0, we have $\Pr[X \geq 1/2] = \Pr[e_1 \leq -1/2] \geq \Pr[e_1 < -1/2]$. We thus get

$$\Pr[e_1 \neq 0] = \Pr[e_1 \geq 1] + \Pr[e_1 \leq -1] \leq 2\Pr[X \geq 1/2] \leq \frac{4\sigma}{\sqrt{2\pi}} e^{-\frac{1}{8\sigma^2}} \tag{8}$$

using the given inequality.

(b) (2 points) Use Equation (6) and the union bound to give an upper bound on the probability that there are at least two indices i and j such that $e_i \neq 0$ and $e_j \neq 0$.

For fixed indices i and j, e_i and e_j are independent and identlically distributed, so

$$\Pr[e_i \neq 0 \wedge e_j \neq 0] = \Pr[e_1 \neq 0]^2 \leq \frac{8\sigma^2}{\pi} e^{-\frac{1}{4\sigma^2}}. \tag{9}$$

Using a union bound over all pairs $i \neq j$ (there are $m(m-1)/2$ of them) we get

$$\Pr[e_i \neq 0 \wedge e_j \neq 0 \text{ for some pair } i,j] \leq \frac{m(m-1)}{2} \frac{8\sigma^2}{\pi} e^{-\frac{1}{4\sigma^2}}. \tag{10}$$

(c) (2 points) Assume that there is at most one index i such that $e_i \neq 0$, and let $m \geq n$. Describe how to recover the private key from the public key, and discuss the runtime of the attack.

As there is at most 1 error that is not zero, we check all possibilities: either there is no error, or there is one i such that $e_1 \neq 0$. These are $m+1$ possibilities. For each possibility we can try to compute the secret key using Gaussian elimination, and we can check our result by encrypting and subsequently decrypting a plaintext.

Note: You can solve any part of the above problem without solving the previous part

Hint: The union bound states that for two events E and F, the probability of both events occurring is not larger than the sum of the probabilities of the two events, i.e. $\Pr[E \vee F] \leq \Pr[E] + \Pr[F]$.