Selected Topics in Cryptography

Solved Exam Problems

Enes Pasalic
University of Primorska

Koper, 2013

Contents

1 Preface

The following pages contain solutions to core problems from exams in Cryptography given at the Faculty of Mathematics, Natural Sciences and Information Technologies at the University of Primorska.

Enes Pasalic
enes.pasalic@upr.si

2 Exam Problems

Problem 0

You have found an old ciphertext, where you know that the plaintext discusses cryptographic methods. You suspect that a Vigenere cipher has been used and therefore look for repeated strings in the ciphertext.

You find that the string TICRMQUIRTJR occurs twice in the ciphertext. The first occurrence starts at character position 10 in the text and the second at character position 241 (we start counting from 1).

You make the inspired guess that this ciphertext sequence is the encryption of the plaintext word cryptography. If this guess is correct, what is the key ?
Hint : Analyze the possible periods.

Solution To estimate the period we use the Kasiski test. The distance between the two occurrences given is

$$241 - 10 = 231 = 3 \cdot 7 \cdot 11$$

positions.
Possible periods are thus 3, 7 and 11. If the guess is correct, we can immediately find the corresponding shifts: at position 10 the shift is

$$T - c = 19 - 2 = 17 = r$$

. Similar computations for the other positions gives the shift keys

rrectcorrect

We now see that this is not periodic with periods 3 or 11, while period 7 is possible. The keyword of length 7 starts at position 15; hence the keyword is

correct.

Problem 1

Alice wants to encrypt some sequence of independent decimal digits and send to Bob. Let E_K denote the encryption function operating on decimal digits. A sequence of decimal digits $M_1, M_2, \ldots, M_n \in \mathbb{Z}_{10}$ is encrypted to a sequence of ciphertext symbols $C_1, C_2, \ldots, C_n$, $C_i \in \mathbb{Z}_{10}$ by

$$C_i = E_K(M_i), \forall i, 1 \leq i \leq n.$$

a) Determine which of the following mappings that are possible encryption functions: $E_K(M) = M$, $E_K(M) = K$, $E_K(M) = M + K$, $E_K(M) = M \cdot K$, $E_K(M) = M^{K+1}$, if $M, K \in \mathbb{Z}_{10}$, and all operations above are performed $\pmod{10}$.

b) Determine the unicity distance if the cipher is a simple substitution cipher and

$$P(M=0)=P(M=1)=4\cdot P(M=2),$$

together with

$$P(M=2)=P(M=3)=\cdots=P(M=8)=P(M=9).$$

Hint: The entropy of language, denoted by H_L in the textbook, is here the entropy per definition of plaintext (message) digits.

Solution

a) An encryption function must be bijective (invertible) for all fixed keys K. Clearly, $E_K(M)=M$ and $E_K(M)=M+K$ are bijective. The others are not bijective for all K.

b) Unicity distance (pg. 63) is defined as $n_0=\frac{\log_2|\mathcal{K}|}{R_L\log_2|\mathcal{P}|}=\frac{\log_2|\mathcal{K}|}{\log_2|\mathcal{P}|-H_L}$ using definition (pg. 61) $R_L=1-\frac{H_l}{\log_2|\mathcal{P}|}$. From the conditions we also get:

$$P(M=0)=P(M=1)=4\cdot P(M=i), i\in[2,9] \quad\Rightarrow\quad P(M=0)=\frac{1}{4};\;\; P(M=2)=\frac{1}{16}.$$

$$\begin{aligned}\log_2|\mathcal{K}| &= H(K)=\log_2(10!)\\ \log_2|\mathcal{P}| &= H(P)=\log_2(10)\\ H_L &= H(M)=-2\frac{1}{4}\log_2\frac{1}{4}-8\frac{1}{16}\log_2\frac{1}{16}=3 \;\;\text{(indicated by hint)}\end{aligned}$$

So $n_0=\frac{\log_2(10!)}{\log_2(10)-3}=67.8\approx 68$.

Problem 2

In the textbook the four basic modes of operations of block ciphers (ECB, CBC, OFB, CFB) are analyzed with respect to *error propagation in encryption.* That is, the consequences on ciphertext blocks by changing a single plaintext block are discussed.

a) For all four modes of operation analyze the effect on the decryption of remaining blocks if for the sequence of ciphertext blocks $c_1,c_2,\ldots,c_n$ some ciphertext block c_j is errorness, $1\le j<n$. That is, specify which of plaintext blocks $x_j,x_{j+1},x_{j+2},\ldots,x_n$ are recieved correctly.

b) One of the recommendations for a proper use of “One-time pad” (to ensure perfect secrecy) is to never reuse the same key for encryption of two different messages. The simplest way to implement “One-time pad” is to generate a random key sequence of the same length as message and to encrypt using,

$$C_i=M_i+K_i \pmod{26},$$

where K_i are random key characters and $\mathcal{M}=\mathcal{K}=\mathcal{C}=\mathbb{Z}_{26}$. Explain how only the knowledge of two different ciphertext sequences $C=C_1C_2\cdots C_n$ and $C'=C_1'C_2'\cdots C_n'$, obtained by applying the same secret key, can compromize the security of the system.

Solution

a) The four modes of operations are found on pages 83–85. For simplicity assume the ciphertext block c_1 (the notation in the book y_1) is incorrect.

- ECB mode: Only x_1 decrypted incorrectly.
- CBC mode: Only x_1, x_2 decrypted incorrectly.
- OFB mode: Only x_1 decrypted incorrectly.
- CFB mode: Only x_1, x_2 decrypted incorrectly.

b) Given are:

$$\begin{aligned} C_i &= M_i + K_i \pmod{26}, \\ C_i' &= M_i' + K_i \pmod{26}, \end{aligned}$$

Then (adding the two equations) $M_i + M_i' = C_i + C_i' \pmod{26}$ for $i = 1, \ldots, n$. Either of the two answers would suffice:

1. If we know C_i, C_i' using message redundancy one can find M_i, M_i' from known $M_i + M_i'$.
2. Known plaintext attack: The knowledge of M_i implies $M_i' = C_i + C_i' - M_i \pmod{26}$.

Problem 3

a) The so called S-box (Substitution box) is widely used cryptographic primitive in symmetric-key cryptosystems. In AES (Advanced Encryption Standard) the 16 S-boxes in each round are identical. All these S-boxes implement the inverse function in the Galois Field GF(2^8), which can also be seen as a mapping, $S : \{0,1\}^8 \to \{0,1\}^8$, so that

$$x \in GF(2^8) \overset{S}{\mapsto} x^{-1} \in GF(2^8),$$

that is 8 input bits are mapped to 8 output bits. What is the total number of possible mappings one can specify for function S ?
Hint: Any function $f : \mathrm{GF}(2^n) \to \mathrm{GF}(2^n)$ can be represented as a polynomial,

$$f(x) = a_0 + a_1(x) + a_2x^2 + \cdots + a_{2^n-2}x^{2^n-2} + a_{2^n-1}x^{2^n-1}, \;\; a_i \in \mathrm{GF}(2^n)$$

b) Construct the Galois field of 16 elements, GF(2^4), using a primitive polynomial $f(x) = x^4+x+1$. Compute the powers x^i, $0 \le i \le 14$ and represent these powers (multiplcative group) as polynomials of the form $a_0 + a_1x + a_2x^2 + a_3x^3$.

c) Assume we want to implement an S-box using the Galois field from b). If we would like that our S-box is bijective is it a good choice to use function $S : GF(2^4) \to GF(2^4)$ specified by,

$$x \in GF(2^4) \overset{S}{\mapsto} x^3 \in GF(2^4).$$

Motivate your answer !

Solution

a) The question is how many mappings are there over the field $GF(2^n)$. Using the hint any function $f : \mathrm{GF}(2^n) \to \mathrm{GF}(2^n)$ can be represented as a polynomial,

$$f(x) = a_0 + a_1(x) + a_2x^2 + \cdots + a_{2^n-2}x^{2^n-2} + a_{2^n-1}x^{2^n-1}, \quad a_i \in \mathrm{GF}(2^n)$$

Any a_i can be chosen in 2^n ways, the total number of mappings over $GF(2^n)$ is,

$$\overbrace{2^n2^n2^n\cdots2^n}^{2^n times} = 2^{n2^n}.$$

b) In the textbook the field of 8 elements $GF(2^3)$ is constructed. We use the primitive polynomial $x^4 + x + 1$:

x^i	$a_3a_2a_1a_0$
$\mathbf{0}$	0000
$x^0 = 1$	0001
$x^1 = x$	0010
$x^2 = x^2$	0100
$x^3 = x^3$	1000
$x^4 = x + 1 \pmod{x^4 + x + 1}$	0011
$x^5 = x \cdot x^4 = x^2 + x$	0110
$\vdots$	
$x^{14} = x^3 + 1$	1001
$x^{15} = x^4 + x = 1$	0001

c) Note that the order of any element divides the order of the group. For any polynomial above $p(x)^{15} = 1$. As our mapping is x^3 and the order of multiplicative group is 15 it means that e.g.

$$\begin{aligned} 1 \mapsto 1^3 &= 1 \\ x^5 \mapsto x^{15} &= 1 \\ x^{10} \mapsto x^{30} = (x^{15})^2 &= 1 \end{aligned}$$

Hence x^3 is not bijective, the image space contains only 6 elements.

Problem 4

a) Factor the RSA number $n = 3844384501$ using the knowledge that

$$31177611852^2 \equiv 1 \pmod{3844384501}.$$

b) Prove that the number 31803221 is not a prime number using the hint

$$2^{31803212} \equiv 27696377 \pmod{31803221}.$$

Motivate your answer.

Solution

a) We want to factor the RSA number $n = 3844384501$ using the knowledge that

$$3117761185^2 \equiv 1 \pmod{3844384501}.$$

Note that: $(3117761185 - 1) \cdot (3117761185 + 1) \equiv 0 \pmod{n}$, then:

$$\begin{aligned} p &= \gcd(3117761184, 3844384501) = 67801 \\ q = p/n &= 56701. \end{aligned}$$

b) We want to prove that the number $n = 31803221$ is not a prime number using the hint $2^{n-9} \equiv 27696377 \pmod{31803221}$. By the little Fermat's theorem for any prime number p and $a \in \mathbb{Z}_p$ we have $a^{p-1} \equiv 1 \pmod{p}$, **remark** $a^{\mathbf{p-1}}$ not a^p.

By testing: $2^{n-9} \cdot 2^8 \equiv 27696377 \cdot 256 \equiv 29957450 \neq 1 \pmod{31803221}$. Hence, n is not a prime number!

Problem 5

a) Given are two protocols in which the sender's party performs the following operation:

Protocol A:

$$y = e_{k_1}(x||H(k_2||x)),$$

where x is the message, H is a hash function such as SHA-1, e is a symmetric-key encryption algorithm, "||" denotes simple concatenation, and k_1, k_2 are secret keys which are only known to the sender and the receiver.

Protocol B:

$$y = e_k(x||sig_{k_{pr}}(H(x))),$$

where k is a shared secret key, and k_{pr} is a private key of the sender (not shared with the receiver).

Provide a step-by-step description (e.g., with an itemized list) of what the receiver does upon reception of y.

b) State whether the following security services:

- confidentiality
- integrity

- non-repudiation (preventing an entity from denying previous commitments or actions)

is given for each of the two protocols given in the previous problem.

Solution

a) Protocol A performs the following:

1. Decryption of y using symmetric key k_1

$$d_{k_1}(y) = x||H(k_2||x).$$

2. Concatenate k_2 and x, where k_2 is 2nd secret key (shared).
3. Compute hash of $k_2||x$, that is $H(k_2||x)$.
4. Compare computed hash value with the one obtained in 1.

Protocol B performs the following:

1. Decrypt as in 1A, $d_k(y) = x||sig_{k_pr}(H(x))$ using shared symmetric key k.
2. Compute $H(x)$
3. Feed $H(x)$ and $sig_{k_pr}(H(x))$ into verification algorithm, check if signature on $H(x)$ is valid. Verification algorithm needs public key of the sender.

b) For protocol A we have:

- confidentiality, YES through encryption
- integrity, YES through hashing; changing y lead to invalid pair x' and $H(k_2||x')$.
- non-repudiation, NO, both Alice (sender) and Bob (receiver) can generate valid message:

$$y = e_{k_1}(x||H(k_2||x)).$$

For protocol B we have:

- confidentiality, YES through encryption
- integrity, YES through signing; changing y lead to invalid pair x' and $sig_{k_{pr}}(H(x'))$.
- non-repudiation, YES, only sender can send a message with valid signature.

Problem 6

We wish to encrypt a memoryless source with alphabet $\mathcal{M} = \{0,1,2\}$ and $P(M = 0) = 1/2, P(M = 1) = p, P(M = 2) = 1/2 - p$, $0 \le p \le 1/2$. Let the key $K = (K_0, K_1, K_2)$ be chosen uniformly from the set of binary 3-tuples. A sequence of messages $M_1, M_2, \ldots, M_n$ is encrypted to a sequence of ciphertexts $C_1, C_2, \ldots, C_n$ by,

$$C_i = M_i + K_{i \bmod 3} \pmod 3, \quad \forall i, 1 \leq i \leq n.$$

a) Find all values of p that give a unicity distance larger than 20.

b) Let $p = 0$. Propose a new cipher for this source that has an infinity unicity distance.

Solution

a) Unicity distance is defined as $n_0 = \frac{\log_2 |\mathcal{K}|}{R_L \log_2 |\mathcal{P}|} = \frac{\log_2 |\mathcal{K}|}{\log_2 |\mathcal{P}| - H_L}$ using definition $R_L = 1 - \frac{H_l}{\log_2 |\mathcal{P}|}$.

$$\begin{aligned}
\log_2 |\mathcal{K}| &= H(K) = \log_2 8 = 3 \\
\log_2 |\mathcal{P}| &= H(P) = \log_2 3 \\
H_L &= H(M) = -\frac{1}{2}\log_2 \frac{1}{2} - p \log_2 p - (\frac{1}{2} - p)\log_2(\frac{1}{2} - p) = \\
&= \ldots = 1 + \frac{1}{2}(-2p \log_2 2p - (1-2p)\log_2(1-2p) = 1 + \frac{1}{2}h(2p).
\end{aligned}$$

So

$$n_0 = \frac{3}{\log_2 3 - 1 - \frac{1}{2}h(2p)} = \frac{3}{\log_2 \frac{3}{2} - \frac{1}{2}h(2p)} > 20.$$

This gives $h(2p) > 0.87$. By trial method $h(2p) = 0.87$ has two solutions $2p = 0.291$ and $2p = 0.709$ (symmetric around 1/2). Thus $0.291 \leq 2p \leq 0.709$ so that $0.15 \leq p \leq 0.35$.

b) When $p = 0$ there are only two plaintexts $M = 0$ and $M = 2$. Define,

$$\theta(M) = \begin{cases} 0, & \text{if } M = 0 \\ 1, & \text{if } M = 2 \end{cases}$$

Then $C_i = \theta(M) + K_{i \bmod 3} \pmod 2$ has infinite unicity distance.

Problem 7

Differential cryptanalysis is based on the so-called characteristics, that are essentially differences in plaintext pairs that have a high probability of causing certain differences in ciphertext pairs.

a) Explain why the input differences to the first round of DES are chosen in specific form so that (L, R) and (L^*, R^*) differ only in few positions. In the second round characteristic R_1' is always chosen to be 00000000_{16}. Why ? Careful motivation is needed.

b) DESX was proposed by R. Rivest to protect DES against exhaustive key search. DESX uses one 64-bit secret key W to perform pre- and postwhitening of data and a 56-bit DES key K, and operates as follows,

$$C = W \oplus E_K(P \oplus W).$$

Show that a similar construction,

$$C = W \oplus E_K(P)$$

without prewhitening is insecure and can be broken using an attack of complexity 2^{56}.

Solution

The input pairs are chosen so that their difference is of low weight in order to keep the number of active S-boxes as low as possible. The idea is to have 0 as the input difference to seven S-boxes, while the input to the remaining S-box is nonzero, chosen to maximize the probability the input (L'_0, R'_0) may cause in the output (distribution table).
Choosing R'_1 to be 00000000_{16} the 1-round characteristic has the maximum probability $p = 1$, see the textbook.

b) Assume we have a small number of plaintext/ciphertext pairs (P_i, C_i). Then for all 2^{56} possible values of K we can compute $E_K(P_0)$ and $E_K(P_1)$. For a correct guess we must have:

$$C_0 \oplus C_1 = W \oplus E_K(P_0) \oplus W \oplus E_K(P_1) = E_K(P_0) \oplus E_K(P_1).$$

However, if key is not correct then the probability that $C_0 \oplus C_1 = E_K(P_0) \oplus E_K(P_1)$. is negligible (one may test further with more (P_i, C_i) pairs).
Finally, the key W is computed as $W = C_i \oplus E_K(P_1)$. The complexity is approximately 2^{56} operations.

Problem 8

a) For a fast encryption in RSA it is popular to use $e = 3$. Although RSA is considered to be a secure public-key cryptosystem, the implementations of RSA can made encryption completely insecure.

Assume that $M \in \mathbb{Z}_{2^{64}}$ is a 64 bit plaintext that is encrypted using a 512 bit RSA modulus n and encryption exponent $e = 3$. Explain why this is completely insecure.

Demonstrate this by finding the pliantext corresponding to the ciphertext $C = 33076161$ when $n = 100082119$.

b) Suppose an active adversary wishes to decrypt a particular message $c = m^e \bmod n$ intended for A. Assume also that A will decrypt arbitrary ciphertext for adversary other than c itself. Describe how the adversary can make A to reveal the plaintext message m corresponding to c.

Solution

a) Using 64 bit messages together with $e = 3$ there does not occur modular reduction for n 512 bit modulus. For given C we have $P = C^{1/3} = 321$.

b) The adversary gives $\overline{c} = cx^e$ to A for decryption, $x \neq 1$ is random element from $\mathbb{Z}_n^*$. A computes $\overline{m} = \overline{c}^d \pmod{n}$ which equals to

$$\overline{m} \equiv \overline{c}^d \equiv c^d (x^e)^d \equiv mx \pmod{n}.$$

From this the adversary computes $m = \overline{m}x^{-1} \pmod{n}$.

Problem 9

An RSA cryptosystem has open parameters n, e and trapdoor parameters $d, p, q, \phi(n)$, where p, q are primes and $ed \equiv 1 \pmod{\phi(n)}$.

a) Determine how many numbers in $\{0, 1, \ldots, \phi(n)\}$ that are possible values for e if $p = 2p_1 + 1$ and $q = 2q_1 + 1$ where p_1 and q_1 are primes.

b) The prime number theorem states that the number of primes not exceeding N is approximately $N/\ln N$. Thus the number of primes is relatively dense compared to nonprimes.
Therefore for generation of p and q we may adopt the following strategy: test the prime p with some primality test and then choose q close to p with same primality tests. This ensures that p and q are about the same size. Can you use this method to generate primes for RSA crypto system ? Motivate your answer.

c) Prove that $D(E(M)) = M$ for the case $\gcd(M, n) = 1$.

Solution

a) $\phi(\phi(n)) = \phi((p-1)(q-1)) = \phi(4p_1q_1) = 2(p_1 - 1)(q_1 - 1)$.

b) This is not a good method because if p and q are close to each other then factoring N reduces to computing $\sqrt{N}$ and finding close prime integers.

c) See the lecture notes.

Problem 10

The following authenticated key agreement protocol is given:

$$\begin{aligned} 1: A \to B &: \quad g^x \bmod p \\ 2: B \to A &: \quad g^y \bmod p, E_k(S_B(g^y \bmod p,\ g^x \mathrm{mod} p)) \\ 3: A \to B &: \quad E_k(S_A(g^x \bmod p,\ g^y \bmod p)) \end{aligned}$$

We assume that the parties have agreed on a (g, p) pair for Diffie-Hellman key exchange, that each user has RSA keys for digital signatures and that they have agreed on a block cipher E for use in subsequent encryption. Furthermore, k is the agreed secret key and S_A and S_B denotes A:s and B:s signature operations, respectively. Describe in details (as a list) A:s and B:s actions at receipt of messages 2 and 3 and what beliefs they have at that stage. Are A and B successfully authenticated to each other after protocol run ?

Solution

We describe the actions and knowledge of the parties after all three messages. (a) After receiving X as message 1, B can choose a y and compute $k = X^y \bmod p$ as the session key. He then computes $Y = g^y \bmod p$, signs $(Y; X)$ and encrypts it using key k. At this stage, B has no reason to believe that the received message was actually from A. (b) After receiving $(Y; c)$ as message 2, A can compute $k = Y^x \bmod p$. Then k is the agreed common key, so she can use this to decrypt c, getting s. Finally, she verifies that s is B:s signature on $(Y; X)$. A can now conclude that the sender of message 2 knows:

- k, since he could encrypt using it.
- B:s signing key, since could produce the signature s.
- X and y, the discrete log of Y (since A successfully decrypted c using $k = Y^x$, but anybody else could only have computed k as X^y).
- $(Y; X)$, since he signed it; this knowledge must be recent, since it includes X, which A herself chose just before sending message 1.

From this evidence, A believes that the sender of message 2 is B and that therefore A and B share k. (c) After receiving c' as message 3, B decrypts it and verifies that the plaintext is A:s signature on $(X; Y)$. From similar reasoning as above, B concludes that the sender of message 3 is A and that A and B share k.

Problem 11

We wish to encrypt a memoryless source with alphabet $\mathcal{M} = \{0, 1, 2\}$ and $P(M = 0) = 1/2, P(M = 1) = p, P(M = 2) = 1/2 - p$, $0 \leq p \leq 1/2$. Let the key $K = (K_0, K_1, K_2)$ be chosen uniformly from the set of binary 3-tuples. A sequence of messages $M_1, M_2, \ldots, M_n$ is encrypted to a sequence of ciphertexts $C_1, C_2, \ldots, C_n$ by,

$$C_i = M_i + K_{i \bmod 3} \pmod 3, \quad \forall i, 1 \leq i \leq n.$$

a) Find all values of p that give a unicity distance larger than 20.

b) Let $p = 0$. Propose a new cipher for this source that has an infinity unicity distance.

Solution

a) Unicity distance is defined as $n_0 = \frac{\log_2 |\mathcal{K}|}{R_L \log_2 |\mathcal{P}|} = \frac{\log_2 |\mathcal{K}|}{\log_2 |\mathcal{P}| - H_L}$ using definition $R_L = 1 - \frac{H_l}{\log_2 |\mathcal{P}|}$.

$$\begin{aligned}
\log_2 |\mathcal{K}| &= H(K) = \log_2 8 = 3 \\
\log_2 |\mathcal{P}| &= H(P) = \log_2 3 \\
H_L &= H(M) = -\frac{1}{2}\log_2 \frac{1}{2} - p \log_2 p - (\frac{1}{2} - p) \log_2(\frac{1}{2} - p) = \\
&= \ldots = 1 + \frac{1}{2}(-2p \log_2 2p - (1 - 2p) \log_2(1 - 2p) = 1 + \frac{1}{2} h(2p).
\end{aligned}$$

So

$$n_0 = \frac{3}{\log_2 3 - 1 - \frac{1}{2} h(2p)} = \frac{3}{\log_2 \frac{3}{2} - \frac{1}{2} h(2p)} > 20.$$

This gives $h(2p) > 0.87$. By trial method $h(2p) = 0.87$ has two solutions $2p = 0.291$ and $2p = 0.709$ (symmetric around 1/2). Thus $0.291 \leq 2p \leq 0.709$ so that $0.15 \leq p \leq 0.35$.

b) When $p = 0$ there are only two plaintexts $M = 0$ and $M = 2$. Define,

$$\theta(M) = \begin{cases} 0, & \text{if } M = 0 \\ 1, & \text{if } M = 2 \end{cases}$$

Then $C_i = \theta(M) + K_{i \bmod 3} \pmod 2$ has infinite unicity distance.

Problem 12

Differential cryptanalysis is based on the so-called characteristics, that are essentially differences in plaintext pairs that have a high probability of causing certain differences in ciphertext pairs.

a) Explain why the input differences to the first round of DES are chosen in specific form so that (L, R) and (L^*, R^*) differ only in few positions. In the second round characteristic R_1' is always chosen to be 00000000_{16}. Why ? Careful motivation is needed.

b) DESX was proposed by R. Rivest to protect DES against exhaustive key search. DESX uses one 64-bit secret key W to perform pre- and postwhitening of data and a 56-bit DES key K, and operates as follows,

$$C = W \oplus E_K(P \oplus W).$$

Show that a similar construction,

$$C = W \oplus E_K(P)$$

without prewhitening is insecure and can be broken using an attack of complexity 2^{56}.

Solution

The input pairs are chosen so that their difference is of low weight in order to keep the number of active S-boxes as low as possible. The idea is to have 0 as the input difference to seven S-boxes, while the input to the remaining S-box is nonzero, chosen to maximize the probability the input (L_0', R_0') may cause in the output (distribution table).
Choosing R_1' to be 00000000_{16} the 1-round characteristic has the maximum probability $p = 1$, see the textbook.

b) Assume we have a small number of plaintext/ciphertext pairs (P_i, C_i). Then for all 2^{56} possible values of K we can compute $E_K(P_0)$ and $E_K(P_1)$. For a correct guess we must have:

$$C_0 \oplus C_1 = W \oplus E_K(P_0) \oplus W \oplus E_K(P_1) = E_K(P_0) \oplus E_K(P_1).$$

However, if key is not correct then the probability that $C_0 \oplus C_1 = E_K(P_0) \oplus E_K(P_1)$. is negligible (one may test further with more (P_i, C_i) pairs).
Finally, the key W is computed as $W = C_i \oplus E_K(P_1)$. The complexity is approximately 2^{56} operations.

Problem 13

a) For a fast encryption in RSA it is popular to use $e = 3$. Although RSA is considered to be a secure public-key cryptosystem, the implementations of RSA can made encryption completely insecure.

Assume that $M \in \mathbb{Z}_{2^{64}}$ is a 64 bit plaintext that is encrypted using a 512 bit RSA modulus n and encryption exponent $e = 3$. Explain why this is completely insecure.

Demonstrate this by finding the plaintext corresponding to the ciphertext $C = 33076161$ when $n = 100082119$.

b) Suppose an active adversary wishes to decrypt a particular message $c = m^e \bmod n$ intended for A. Assume also that A will decrypt arbitrary ciphertext for adversary other than c itself. Describe how the adversary can make A to reveal the plaintext message m corresponding to c.

Solution

a) Using 64 bit messages together with $e = 3$ there does not occur modular reduction for n 512 bit modulus. For given C we have $P = C^{1/3} = 321$.

b) The adversary gives $\overline{c} = cx^e$ to A for decryption, $x \neq 1$ is random element from $\mathbb{Z}_n^*$. A computes $\overline{m} = \overline{c}^d \pmod{n}$ which equals to

$$\overline{m} \equiv \overline{c}^d \equiv c^d (x^e)^d \equiv mx \pmod{n}.$$

From this the adversary computes $m = \overline{m}x^{-1} \pmod{n}$.

Problem 14

An RSA cryptosystem has open parameters n, e and trapdoor parameters $d, p, q, \phi(n)$, where p, q are primes and $ed \equiv 1 \pmod{\phi(n)}$.

a) Determine how many numbers in $\{0, 1, \ldots, \phi(n)\}$ that are possible values for e if $p = 2p_1 + 1$ and $q = 2q_1 + 1$ where p_1 and q_1 are primes.

b) The prime number theorem states that the number of primes not exceeding N is approximately $N/\ln N$. Thus the number of primes is relatively dense compared to nonprimes.
Therefore for generation of p and q we may adopt the following strategy: test the prime p with some primality test and then choose q close to p with same primality tests. This ensures that p and q are about the same size. Can you use this method to generate primes for RSA crypto system ? Motivate your answer.

c) Prove that $D(E(M)) = M$ for the case $\gcd(M, n) = 1$.

Solution

a) $\phi(\phi(n)) = \phi((p-1)(q-1)) = \phi(4p_1q_1) = 2(p_1-1)(q_1-1)$.

b) This is not a good method because if p and q are close to each other then factoring N reduces to computing $\sqrt{N}$ and finding close prime integers.

c) See the textbook, page 124 (actually this problem is missprinted, the idea was to prove the case $\gcd(M, n) \neq 1$, thus you get 7 points just to find the proof in the book).

Problem 15

The following authenticated key agreement protocol is given:

$$\begin{aligned} 1: A \to B &\quad:\quad g^x \bmod p \\ 2: B \to A &\quad:\quad g^y \bmod p, E_k(S_B(g^y \bmod p,\ g^x \text{mod} p)) \\ 3: A \to B &\quad:\quad E_k(S_A(g^x \bmod p,\ g^y \bmod p)) \end{aligned}$$

We assume that the parties have agreed on a (g, p) pair for Diffie-Hellman key exchange, that each user has RSA keys for digital signatures and that they have agreed on a block cipher E for use in subsequent encryption. Furthermore, k is the agreed secret key and S_A and S_B denotes A:s and B:s signature operations, respectively. Describe in details (as a list) A:s and B:s actions at receipt of messages 2 and 3 and what beliefs they have at that stage. Are A and B successfully authenticated to each other after protocol run ? We describe the actions and knowledge of the parties after all three messages.

Solution

(a) After receiving $X = g^x$ as message 1, B can choose a y and compute $k = X^y \bmod p$ as the session key. He then computes $Y = g^y \bmod p$, signs $(Y; X)$ and encrypts it using key k. At this stage, B has no reason to believe that the received message was actually from A.

(b) After receiving $(Y; c)$ as message 2, A can compute $k = Y^x \bmod p$. Then k is the agreed common key, so she can use this to decrypt c, getting s. Finally, she verifies that s is B:s signature on $(Y; X)$. A can now conclude that the sender of message 2 knows:

- k, since he could encrypt using it.
- B:s signing key, since he could produce the signature s.
- X and y, the discrete log of Y (since A successfully decrypted c using $k = Y^x$, but anybody else could only have computed k as X^y).
- $(Y; X)$, since he signed it; this knowledge must be recent, since it includes X, which A herself chose just before sending message 1.

From this evidence, A believes that the sender of message 2 is B and that therefore A and B share k.

(c) After receiving c' as message 3, B decrypts it and verifies that the plaintext is A:s signature on $(X;Y)$. From similar reasoning as above, B concludes that the sender of message 3 is A and that A and B share k.

Problem 16

Let $E_k(m), D_k(c)$ be a block cipher. Fischer Spiffy Mixer (FSM) mode encrypts a sequence of message blocks $m_1, m_2, \ldots,$ by the sequence of ciphertext blocks $c_1, c_2, \ldots$ using the following method:

$$c_i = m_{i-1} \oplus E_k(m_i \oplus c_{i-1}), \quad i \geq 1$$

m_0 and c_0 are fixed (public) initialization vectors.

(a) Describe how decryption is performed.

(b) Suppose ciphertext block c_i is damaged in transit. Which plaintext blocks become undecipherable as a result? Explain.

Solution

(a) XORing m_{i-1} to both sides of the encryption equation gives

$$c_i \oplus m_{i-1} = E_k(m_i \oplus c_{i-1}).$$

Applying the decryption function on both sides gives

$$D_k(c_i \oplus m_{i-1}) = m_i \oplus c_{i-1},$$

so $m_i = c_{i-1} \oplus D_k(c_i \oplus m_{i-1})$.

b) If c_i was damaged then m_i is damaged. If m_i is damaged then m_{i+1} is damaged. From then on all messages are damaged.

Problem 17

In the RSA cryptosystem encryption is performed using $C \equiv M^e \pmod{N}$, where $N = pq$ for suitably chosen large primes p, q, and $\gcd(e, \phi(N)) = 1$. In a chaining attack on RSA, given a ciphertext $C \equiv M^e \pmod{N}$ the atacker computes,

$$C^e \pmod{N}, C^{e^2} \pmod{N}, \ldots, C^{e^k} \pmod{N},$$

unless $C \equiv C^{e^k} \pmod{N}$ is obtained. That is, k is the least positive integer that specifies the cycle.

(a) Explain why the attacker can always find $k \in [1, N-1]$ so that $C \equiv C^{e^k} \pmod{N}$.
Hint: Recall that RSA is an encryption algorithm and therefore bijective, i.e. $M_1 \neq M_2$ cannot be mapped to the same ciphertext.

(b) Can attacker recover the message M from the observed sequence above in case $C \equiv C^{e^k}$ $\pmod{N}$ is valid ?

(c) Explain how the attacker can factor N by finding integer u such that $\gcd(C^{e^u}, N) > 1$.
Hint: Analyze different cases w.r.t. $\pmod p$ and $\pmod q$ congruences.

Solution

(a) Since encryption is a permutation on the message space $\{0, 1, \ldots, N-1\}$ we have $C_1^e \not\equiv C_2^e \pmod N$ for $C_1 \neq C_2$. Thus there must exist a positive integer k such that $C^{e^k} \equiv C \pmod N$. Otherwise, assume that there is no k satisfying this for $k \in \{1, \ldots, N-1\}$. Then it must be the case that $C^{e^r} = C^{e^s} \equiv C^* \pmod N$ for some $1 \leq r \neq s \leq N-1$. This means that the ciphertext C^* is an encryption of two messages $M^{e^{s-1}}$ and $M^{e^{r-1}}$. Note that,

$$C^* = (M^{e^{r-1}})^e = (M^{e^{s-1}})^e \pmod N.$$

(b) The attacker reveals the plaintext M as,

$$C^{e^k} = (C^{e^{k-1}})^e = M^e \equiv C \pmod N,$$

therefore $C^{e^{k-1}} \equiv M \pmod N$.

(c) Assuming $\gcd(C^{e^u}, N) = f > 1$ for some $u > 0$ we have the following situations:
If,

$$C^{e^u} \equiv C \pmod p \text{ and } C^{e^u} \not\equiv C \pmod q$$

then $f = p$.
If,

$$C^{e^u} \not\equiv C \pmod p \text{ and } C^{e^u} \equiv C \pmod q$$

then $f = q$. But if both,

$$C^{e^u} \equiv C \pmod p \text{ and } C^{e^u} \equiv C \pmod q$$

then $f = N$. There is no factorization but $C^{e^u} \equiv C \pmod N$ in this case, thus $C^{e^{u-1}} \equiv M \pmod N$.

Problem 18

Let $h : \{0,1\}^* \to \{0,1\}^n$ be a hash function that is second-preimage and collision resistant. Let $h' : \{0,1\}^* \to \{0,1\}^{n+1}$ be the hash function given by the rule

$$h'(x) = \begin{cases} 0||x & x \in \{0,1\}^n, \\ 1||h(x) & \text{otherwise.} \end{cases}$$

Prove that h' is not preimage resistant, but still second-preimage and collision resistant.

Solution

The modified hash function h' is not preimage resistant, since for any hash value y of the form $0||x$, a preimage is x.

Therefore, we can find a preimage for at least one half of all possible hash values.

Next we prove that h' inherits second-preimage and collision resistance from h. We show that if we can find a collision or a second preimage for h', then we can easily do so for h. Suppose

$$\exists x_0 \neq x_1 : h'(x_0) = h'(x_1).$$

Two cases:

1. First bit of $h'(x_0)$ is 0. Impossible as implies $x_0 = x_1$.
2. First bit of $h'(x_0)$ is 1. Then $h(x_0) = h(x_1)$ a contradiction, as h is collision resistant.

Problem 19

The S/Key protocol is a variation of a well-known Lamport's one-time password protocol. The idea is that a user U can efficiently derive a new password for each session, based on a master password P_U. The n one-time passwords are derived recursively by applying for instance hash function f and the sequence of password is given as,

$$f^n(P_U) \overset{def.}{=} \underbrace{f(\cdots(f}_{n}(P_U))\cdots)$$

S/KEY PROTOCOL:

PREMISE : User U and Host H have setup U's initial password entry
$(ID_U, f^n(P_U), n)$; f is a hash function, and U must memorize P_U.
The current password entry of U in H is $(ID_U, f^c(P_U), c)$,
for $1 \leq c \leq n$.

GOAL: U authenticates to H without transmitting P_U in cleartext.

1. $U \to H : \mathrm{ID}_U$
2. $H \to U : c$, ”Input Password:” ;
3. $U \to H : Q = f^{c-1}(P_U)$;
4. H finds entry $(ID_U, f^c(P_U), c)$ in its database;
   Access is granted if
   $$f(Q) = f^c(P_U),$$
   and then U's password entry is updated to $(ID_U, Q, c-1)$

Analyze the security of the protocol with an active man-in-the-middle adversary. The adversary is capable of intercepting the messages and sending fraudulent messages to U and H. Can adversary fool the protocol and gain the knowledge of the next session key ?

Solution

Malice (adversary) simply supplies the user with incorrect session value c.

U		Malice		H
	$\xrightarrow{\mathrm{ID}_U}$		$\xrightarrow{\mathrm{ID}_U}$	
	$\xleftarrow{\mathbf{c-1}, \text{”Password”}}$		$\xleftarrow{\mathbf{c}, \text{”Password”}}$	
	$\xrightarrow{f^{c-2}(P_U)}$		$\xrightarrow{f^{c-1}(P_U)}$	

Malice is in possession of $f^{c-2}(P_U)$ which he can use for logging-in in the name of U in the next session.

Problem 20

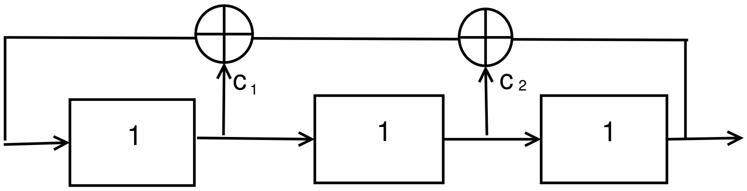
This problem concerns generation of periodic sequences.

1. We consider the possibility of obtaining the periodic sequence $\{100011\}^{\infty}$, that is

   $$101001011001\cdots$$

   using the LFSR of minimum length. That is, start with $L = 2$ and increase the length by one in case the sequence cannot be generated by this length. Specify the connection polynomial of the shortest LFSR found in this way. Motivate your answer and ensure yourself that you really get the desired sequence.

   Solution The maximum period of LFSR of length 2 is 3 and therefore we may check the length 3. The general appearance of such an LFSR is,

   We have the following equations (using $c_3 = 1$),

   $$0 \cdot c_1 + 0 \cdot c_2 + 1 \;=\; 0$$

   Thus, if $c_3 = 1$ we get a contradiction immediately. Thus we need to check $L = 4$ (using $c_4 = 1$). Then,

   $$\begin{aligned} 0 \cdot c_1 + 0 \cdot c_2 + 0 \cdot c_3 + 1 &= 1 \\ c_1 &= 1 \\ c_1 + c_2 &= 1 \\ c_1 + c_2 + c_3 &= 0 \end{aligned}$$

   The solution is $c_1 = 1, c_2 = 0, c_3 = 1$ and the recurrence can be written as,

   $$s_{t+4} = s_{t+3} + s_{t+1} + s_t, \;\; t \geq 1.$$

   Using $(s_1, s_2, s_3, s_4) = (1, 0, 0, 0)$ we get $s = 100011|100011\cdots$.

2. Assume that we have two maximum length sequences s_1 and s_2 generated by LFSRs of respective length L_1 and L_2, where L_1 and L_2 are relatively prime. What is the period of the sequence $s(t) = s_1(t) + s_2(t)$?

   Solution The period is $(2^{L_1} - 1)(2^{L_2 - 1})$. See the lecture notes regarding the linear complexity of sequences. A valid answer could also have been deduced through an example.

Problem 21

Alice and Bob use a block cipher for encryption and need to choose a mode of operation. Recall the following two modes:

- CBC mode: Here an n block plaintext $M_1 M_2 \ldots M_n$ is encrypted to an n block ciphertext $C_1 C_2 \ldots C_n$ using,
$$C_i = E_K(M_i \oplus C_{i-1}), \;\; i \geq 1,$$
where $C_0 = IV$.
- Counter mode. Here an n block plaintext $M_1 M_2 \ldots M_n$ is encrypted to an n block ciphertext $C_1 C_2 \ldots C_n$, where
$$\begin{aligned} K_i &= E_K(IV||i) \\ C_i &= M_i \oplus K_i, \;\; i \geq 1. \end{aligned}$$

An adversary is able to intercept and changes messages sent between Alice and Bob. Now consider the following scenarios.

1. In some messages sent by Bob, it is the case that the last block is a randomly generated secret key. Decide for the two modes whether the adversary can corrupt messages sent, so that Alice receives a message that looks good after decryption, but contains the wrong key.

   Solution For both modes it is the case that the adversary can replace the last ciphertext block with any other block. When Alice decrypts the message all previous blocks will be unchanged and the message looks good; the last block will be corrupt, but since it is random, there is no way for Alice to discover this. See the lecture notes and the "lion cage example".

2. In some messages sent by Bob, the adversary may know the first block M_1 and want to replace it by another block A_1 of his choice, leaving the rest of the message unchanged. Show that the adversary can achieve this if Counter mode is used. Do you think he can do it with CBC mode (assume changing $C_0 = IV$ is allowed)?

   Solution The adversary can achieve this if the encryption is in Counter mode. The encryption of the first block is $C_1 = M_1 \oplus E_K(IV||1)$, from which he can compute $E_K(IV||1) = M_1 \oplus C_1$. He wants to replace C_1 by
$$C_1' = A_1 \oplus E_K(IV||1)$$

and can easily compute $C_1' = A_1 \oplus M_1 \oplus C_1$. The other blocks are not affected by this.

For CBC mode, we have $M_1 = D_K(C_1) \oplus C_0$. The adversary cannot change C_1, since that would affect Alices decryption of C_2. Instead, he must try to find C_0' such that $A_1 = D_K(C_1) \oplus C_0$. Solving for C_0 we get

$$C_0' = A_1 \oplus D_K(C_1) = A_1 + M_1 + C_0.$$

Thus, this is also possible in the CBC mode.

Problem 21

This questions only require YES or NO answer without any motivation.

1. (T/F)In theory, if the key is truly random, never reused, and kept secret DES and AES are both provably secure against known plaintext attacks.

   Solution FALSE, not provably secure but rather computationally secure.

2. (T/F) A Feistel cipher structure lets you use the same hardware or software for decryption as for encryption.

   Solution TRUE, just reverse the order of the subkeys.

3. (T/F) All block ciphers use S-boxes and permutation P-boxes.

   Solution FALSE, IDEA does not have S and P boxes, though achieving confusion and diffusion by other means.

4. (T/F) DiffieHellman key exchange is an asymmetric scheme that can be used for encryption and signatures, but is not as efficient as RSA.

   Solution FALSE, DH is only used for key exchange, see also the textbook.

5. (T/F) A hash function given by

   $$h(m_1, m_2) = m_1^e m_2^e \pmod{pq},$$

   (where p, q are RSA primes and e has the inverse $\pmod{\phi(pq)}$) is a collision resistant hash functions, that is it is computationally hard to find (m_1', m_2') such that $h(m_1, m_2) = h(m_1', m_2')$.

   Solution FALSE, e.g. one can take $(m_1', m_2') = (m_2, m_1)$ and get a collision.

Problem 22

This problem treats the RSA public key cryptosystem.

1. Can a user of RSA choose the encryption exponent e to be even, e.g. $e = 4$.

   Solution No, since then $\gcd(e, \phi(N)) \neq 1$ as $2 | \phi(N) = (p-1)(q-1)$.

2. Let e and e' be two different public keys such that e' is derived from e by flipping one zero to one ($0 \to 1$) in the binary representation of e. Show that $\gcd(e, e') = 1$. Hint: Compare the divisors of e and e'.

   Solution We know that $e' = e + 2^i$ for some i. Any non-trivial divisor of e must be odd, hence not a divisor of 2^i. Therefore it cannot divide e' and thus $\gcd(e, e') = 1$.

Problem 23

Alice wants to send an encrypted message to Bob using RSA, but doesn't know his public key. So, she sends Bob an email asking for the key. Bob replies with his RSA public key (e, N). However, the active adversary intercepts the message and changes one bit in e from 0 to 1, so Alice receives an email claiming that Bobs public key is (e', N), where e' differs from e in one bit. Alice encrypts m with this key and sends it to Bob. Of course, Bob cannot decrypt, since the message was encrypted with the wrong key. So he resends his key and asks Alice to send the encrypted message again, which she does. The adversary eavesdrops to the whole communication without interfering further. Describe how he can now recover m.

Solution The adversary has eavesdropped and thus knows $c = m^e$ and $c' = m^{e'}$. He also knows that e and e' and furthermore, $\gcd(e, e') = 1$. So the adversary can find integers x and y such that

$$ex + e'y = 1.$$

Hence,

$$c^x \cdot c'^y = m^{ex+e'y} = m.$$

Problem 24

A DH-based key exchange protocol for wireless mobile networks was proposed by Park: The system has a common prime modulus p and a generator g. Each party i has a long-term private key $x_i \in \mathbb{Z}_{p-1}$ and a public key $X_i = g^{x_i} \pmod{p}$. To establish a session key between a mobile subscriber M and a base station B, the following protocol is executed (with all arithmetic in $\mathbb{Z}_p$):

$$\begin{aligned} &1.\ B \to M &:& \quad g^{x_B+N_B} \\ &2.\ M \to B &:& \quad N_M + x_M \end{aligned}$$

where N_B and N_M are one-time random nonces (once used random numbers). B calculates the session key as

$$K_{MB} = (g^{x_M+N_M} X_M^{-1})^{N_B}$$

and M calculates it as

$$K_{MB} = (g^{x_B+N_B} X_B^{-1})^{N_M}$$

Then they complete the authentication with a challenge-response using this K_{MB}.

1. Show that the Park's protocol is correct in the sense that B and M calculate the same K_{MB} value.

   Solution We compute,

$$(g^{x_M+N_M} X_M^{-1})^{N_B} = (X_M g^{N_M} X_M^{-1})^{N_B} = g^{N_M N_B} = K_{MB},$$

and

$$(g^{x_B+N_B} X_B^{-1})^{N_M} = (X_B g^{N_B} X_B^{-1})^{N_M} = g^{N_M N_B} = K_{MB}.$$

2. Show that an attacker who has compromised a session key from a previous run, for which (s)he has recorded the messages, can impersonate B. (Hint: Let the attacker replay B's message from the previous session.)

   Solution If the attacker knows $g^{N_M N_B} = K_{MB}$ used in the previous session (s)he can send,

$$\begin{array}{lcl} 1.\, C(B) \to M & : & g^{x_B+N_B} = X_B g^{N_B} \\ 2.\, M \to C(B) & : & N'_M + x_M \end{array}$$

   The attacker knows $N_M + x_m$ and $N'_M + x_M$ so he can compute $\delta = N'_M - N_M$ by subtracting two values. Also the knowledge of $X_B g^{N_B}$ allows him to compute g^{N_B} as X_B is public (use EEA to find the inverse of X_B). Now the two equations,

$$\begin{array}{lcl} K_{MB} & = & (g^{x_M+N_M} X_M^{-1})^{N_B} \\ K'_{MB} & = & (g^{x_M+N'_M} X_M^{-1})^{N_B} \end{array}$$

   gives

$$K'_{MB} = K_{MB}(g^{\delta})^{N_B} = K_{MB}(g^{N_B})^{\delta}.$$

   All the values are known so the attacker can efficiently compute K'_{MB}. Now whatever is the value of N'_M both parties will compute,

$$K'_{MB} = g^{N'_M N_B}.$$

3. In fact this protocol can be broken without having any previous session keys compromised: Show how the attacker can impersonate B by just knowing his public key.

   Solution If the attacker only knows X_B and X_M the protocol is run as,

$$\begin{array}{lcl} 1.\, C(B) \to M & : & g^{x_B+N_C} = X_B g^{N_C} \\ 2.\, M \to C(B) & : & N'_M + x_M \end{array}$$

   Now M computes,

$$K_{MC} = g^{N'_M N_C},$$

   but also the fake server (base station) can compute,

$$(g^{x_M+N'_M} X_M^{-1})^{N_C}$$

   where $x_M + N'_M$; X_M and N_C is known to the attacker. Obviously the same key is computed.

Problem 25

Consider the following cryptosystem:

$$\begin{array}{lll} \mathcal{K}=\{A,B\} & \Pr(A)=2/3 & \Pr(B)=1/3 \\ \mathcal{P}=\{0,1\} & \Pr(0)=3/5 & \Pr(1)=2/5 \\ \mathcal{C}=\{a,b\} & E_A(0)=a & E_A(1)=b \\ E_B(0)=b & E_B(1)=a & \end{array}$$

a) Compute $\Pr(a)$ and $\Pr(0|a)$.

Solution (a) $\Pr(a)=\Pr(0)\cdot\Pr(A)+\Pr(1)\cdot\Pr(B)=\frac{8}{15}$. Use the Bayes' theorem to compute,

$$\Pr(0|a)=\frac{\Pr(0)\cdot\Pr(a|0)}{\Pr(a)}=\frac{\frac{3}{5}\cdot\frac{2}{3}}{\frac{8}{15}}=\frac{3}{4}.$$

b) Is this system a perfect cryptosystem ? If not, what probabilities you would change to make it perfect ?

Solution This is not a cryptosystem with perfect secrecy. We need to change the key distribution. Due to Shannon for a cryptosystem with $|K|=|C|=|P|$ we must have that $Pr(A)=Pr(B)=1/2$. In this case

$$Pr(a)=1/2(Pr(0)+Pr(1))=1/2=Pr(b).$$

Also,

$$\Pr(0|a)=\frac{\Pr(0)\cdot\Pr(a|0)}{\Pr(a)}=\frac{\frac{3}{5}\cdot\frac{1}{2}}{\frac{1}{2}}=\frac{3}{5}=Pr(0).$$

c) We wish to encrypt a memoryless source with alphabet $\mathbb{Z}_3=\{0,1,2\}$ and

$$P(M=0)=1/3;\ \ P(M=1)=1/3;\ \ P(M=2)=1/3;$$

Let the key $K=(K_0,K_1,\dots,K_{l-1})$ be chosen uniformly from the set of ternary l tuples ($K_i\in\mathbb{Z}_3$). A sequence of message symbols

$$\mathbf{M}=(M_0,M_1,\dots,M_{n-1}),$$

is encrypted to a sequence of cyphertext symbols

$$\mathbf{C}=(C_0,C_1,\dots,C_{n-1}),$$

using,

$$C_i=M_i+K_{i \bmod l} \pmod 3,\ \ \forall i, 0\le i\le n-1.$$

Prove or disprove the following statement (by computing the unicity distance).
For $l=64$ the unicity distance n_0 defined as,

$$n_0=\frac{\log_2|\mathcal{K}|}{R_L\log_2|\mathcal{P}|}=\frac{\log_2|\mathcal{K}|}{\log_2|\mathcal{P}|-H_L}.$$

lies in the interval $700 < n_0 < 800$.

Solution Note that $H(\mathcal{P}) = \log_2 3$ and for uniformly distributed messages we have,

$$H_L = H(\mathcal{M}) = \frac{1}{3}\log_2 3 + \frac{1}{3}\log_2 3 + \frac{1}{3}\log_2 3 = \log_2 3$$

and therefore $n_0 = \infty$. Means that we encrypt a random language and no extra information is available to the attacker.

d) For what choices of l and n this cryptosystem has a perfect secrecy ?

Solution For any $n \leq l$. This also comes from the fact that $H(K) \geq H(M)$ for a cryptosystem with perfect secrecy. Note that in general

$$H(K) = \log_2 3^l$$

and

$$H(M) = \log_2 3^n.$$

Another, less formal way is to say that for a perfect secrecy we need the key length at least as the message length (no repetition of key bits) and therefore $n \leq l$.

Problem 26

a) We consider the possibility of obtaining the periodic sequence $\{111000\}^\infty$, that is

$$111000111000\cdots$$

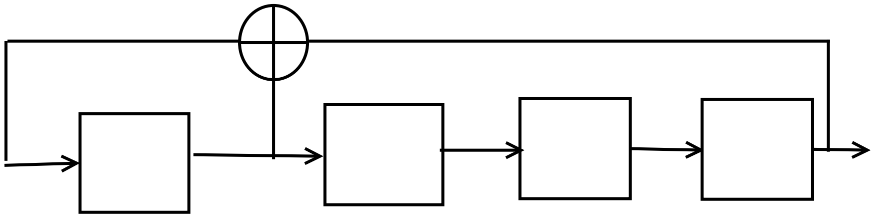
using the LFSR of length 3. Specify the connection polynomial of LFSR of length 3 in case it is possible to generate such a sequence with this length. Motivate your answer.

Solution We cannot get such a sequence since when the state of LFSR is 000 which is a part of the sequence it will output only zeros.

b) Alice wants to encrypt a string of bits. She decides to encrypt using an LFSR as a generator in a stream cipher. However, she knows that just using an LFSR is a bad choice, so she makes a modification. She only uses every second bit of the LFSR sequence. The encryption process would then be as follows. A sequence of bits $m = m_1, m_2, \ldots, m_n$ is encrypted to a sequence of ciphertext symbols $c = c_1, c_2, \ldots, c_n$ by

$$c_i = m_i \oplus s'_i, \ \ \forall i, 1 \leq i \leq n$$

where $s'_i = s'_1, s'_2, \ldots$ is obtained from the binary LFSR sequence $s = s_1, s_2, \ldots$ using $s'_i = s_{2i}$, $i = 1, 2, \ldots$. Finally, s is generated by a length 4 LFSR with connection polynomial $C(x) = 1+x+x^4$, with initial (secret) state (s_1, s_2, s_3, s_4) (the LFSR outputs first s_1 then s_2 etc.)



Eve observed the ciphertext $\mathbf{c} = 0,1,1,1,1,1,0$. Also, she knows that the plaintext starts as $1,1,1,1,\ldots$, i.e.,

$$\mathbf{m} = 1,1,1,1,m_5,m_6,m_7.$$

Find the remaining plaintext bits.

Solution This is a routine exercise, given the LFSR and its connection polynomial. From the assumption we know that the state bits of LFSR are

$$s_{2i} = c_i \oplus m_i, \quad for \;\; 1 \le i \le 4.$$

That is,

$$s_2 = 1, \quad s_4 = 0, \quad s_6 = 0, \quad s_8 = 0.$$

We note that the connection polynomial corresponds to the recursion,

$$s_i = s_{i-1} + s_{i-4} \quad i \ge 5$$

Hence, we have

$$\begin{aligned} s_5 &= s_4 + s_1 \\ s_6 &= s_5 + s_2 \\ s_7 &= s_6 + s_3 \\ s_8 &= s_7 + s_4 \end{aligned}$$

This gives that $s_5 = s_2 + s_6 = 1$ and therefore $s_1 = s_4 + s_5 = 1$. Also, $s_7 = s_4 + s_8 = 0$ which gives $s_3 = s_6 + s_7 = 0$. Thus the secret state is $(s_1, s_2, s_3, s_4) = (1,1,0,0)$ and we can find all the plaintext bits easily. To find m_5, m_6, m_7 we need to find s_{10}, s_{12}, s_{14}. Using recursion we compute,

$$\begin{aligned} s_9 &= s_8 + s_5 = 1 \\ s_{10} &= s_9 + s_6 = 1 \\ s_{11} &= s_{10} + s_7 = 1 \\ s_{12} &= s_{11} + s_8 = 1 \\ s_{13} &= s_{12} + s_9 = 0 \\ s_{14} &= s_{13} + s_{10} = 1 \end{aligned}$$

Thus, $m_5 = c_5 + s_{10} = 1 + 1 = 0$, $m_6 = c_6 + s_{12} = 1 + 1 = 0$, $m_7 = c_7 + s_{14} = 0 + 1 = 1$.

Problem 28

In class we discussed the method of index calculus for solving Discrete Log Problem. The idea was to compute discrete logs (where basis corresponds to the group generator g) of some small prime base B. Thus to solve $X = g^x$ for a given X we would compute $\log_g b$ for all $b \in B$.

a) Given a prime number 83 check efficiently whether 2 is a generator of $\mathbb{Z}_{83}^* = \{1, 2, \ldots, 82\}$. Use the fact that $2^8 \equiv 7 \pmod{83}$. Apply square and multiply method and show your computations !

Solution Here we use the fact that the order of an element divides the order of the group (Lagrange, Fermat ...). If 2 is a generator we must have

$$2^{82} \equiv 1 \pmod{83}$$

and

$$2^a \not\equiv 1 \pmod{83}$$

for all $a : a|(p-1)$. But $p - 1 = 2 \times 41$ and 41 is prime so we only need to test that

$$2^{41} \not\equiv 1 \pmod{83}.$$

Using the fact $2^8 \equiv 7 \pmod{83}$ we need to compute

$$2 \cdot 2^8 \cdot 2^{32} = 2 \times 7 \times 7^4 \pmod{83}.$$

Easiest to check that $83 \not| (2 \times 7 \times 7^4) - 1$. Thus 2 is a generator.

b) Let $B = \{2, 3, 5, 7\}$ (or you may choose other more suitable basis) and compute discrete logs of the basis.

Solution Since one can freely choose the basis we may try $B = \{2, 7\}$ and we can directly compute $\log_2 2 = 1$ and from the above fact $\log_2 7 = 8$.

c) Let again $B = \{2, 3, 5, 7\}$ (or other basis) and we need to compute $29 = 2^x \pmod{83}$. Compute $x = \log_2 29$ using the result in b).

Solution Using the lecture notes (with similar problem) the solution is found quickly. Just a few attempts of computing $29 \cdot 2^k$ gives for $k = 4$

$$29 \cdot 2^4 \equiv 49 = 7^2 \pmod{83}$$

so we may write (applying $\log_2$ to the equation)

$$\log_2 29 + 4 \log_2 2 = 2 \log_2 7 \pmod{82}$$

That is,

$$\log_2 29 = 2 \log_2 7 - 4 \log_2 2 = 2 \cdot 8 - 4 = 12.$$

Problem 28

We first recall ElGamal encryption. The setting is Z_p for a large prime p where $p-1$ has a prime divisor q. Further, g is a generator for a subgroup of order q of Z_p^*.

A community of users share parameters p, q and g. Typically, p is a 1024 bit number, while q has only 160 bits. Each user has a private key $x < q$ and a public key $X = g^x \pmod{p}$. To encrypt a message m for this user, the sender chooses a random number $y < q$ and encrypts the message as

$$(c_1, c_2) = (g^y, m \cdot X^y) \pmod{p}.$$

Because of the random choice of y for each message, different encryptions of the same message will be different. However, there is another quantity involving only m and q that can be computed from the ciphertext. This gives the basis for attacks on this textbook version of ElGamal.

a) Show how to compute m^q given the encryption of m.

Solution Since the second part involves m we need to do something with $c_2 = m \cdot X^y$. If we raise c_2 to the power of q we have that

$$c_2^q = (m \cdot (g^x)^y)^q = m^q \cdot (g^q)^{xy} = m^q$$

since $g^q = 1$ (g is a generator of a group of order q).

b) Given two messages m_1 and m_2 in Z_p^* with $m_1^q = m_2^q$, can one conclude that $m_1 = m_2$ (motivate your answer)? Hint: The probability that a random element in Z_p^* has order q is very small.

No. Given that $m_1^q = m_2^q$ we have

$$(m_1 m_2^{-1})^q \equiv 1 \pmod{p}$$

that is $(m_1 m_2^{-1})$ has order q. This does not mean that m_1 and m_2 are equal. However, for random messages (using the hint), the probability of getting an element of order q in this way is quite small. Therefore with high probability $m_1 = m_2$!

Problem 29

The following protocol is used to authenticate the user A to server B. But not vice versa, i.e. we assume that the identity of the server is checked by some means (e.g. certificate).

$$\begin{aligned} 1.\ A \rightarrow B &: \quad A \\ 2.\ B \rightarrow A &: \quad N_B \\ 3.\ A \rightarrow B &: \quad E_{K_{AB}}(N_B) \end{aligned}$$

a) The assumption is that the user and server shares the same encryption (symmetric) key K_{AB}. Explain why it is important that N_B is not repeated. What kind of attack is applicable when the same N_B is reused ?

Solution: The numbers N_B are called nonce (**n**umbers used **once**). If the attacker observes the usage of the same N_B he simply resends previously observed $E_{K_{AB}}(N_B)$ to the server and identifies himself as A. This is known as replay attack.

b) If N_B is 64-bit long and chosen randomly after how many such numbers you expect the repetition of N_B with high probability ? Propose a simple measure to avoid the reusage of N_B. Is it practical ? Recall that B is a server and it might share many keys with many users.

Solution: Due to the birthday paradox after some 2^{32} values the probability that the nonce N_B has already been used is 50%. The server can simply store all used nonces in a list and check whether a new randomly generated nonce has already been used. This is not practical as saving such data for many users requires a huge memory storage.

c) The protocol can be extended to mutually identify A and B. It works as,

$$\begin{aligned} 1.\ A \to B &\ :\ A, N_A \\ 2.\ B \to A &\ :\ E_{K_{AB}}(A, N_A, N_B) \\ 3.\ A \to B &\ :\ E_{K_{AB}}(N_B, N_A) \end{aligned}$$

The attacker knows all the details of the protocol implementation and is capable of observing transmitted data. Assume now that encryption is performed using ECB mode of encryption, that is each message block A, N_A, N_B is encrypted separately. Does this mode of encryption provides a secure identification scheme. Motivate your answer.

Solution: Using the textbook the ECB mode implies a separate encryption of the message blocks. For instance the message 2 is encrypted as three blocks $E_{K_{AB}}(A)$, $E_{K_{AB}}(N_A)$, $E_{K_{AB}}(N_B)$. The attacker sends the last two blocks of message 2 to the server and identifies himself as A.

d) To prevent from this attack instead of the ECB mode another mode is proposed,

$$C_i = E_K(M_i) + C_{i-1} \quad i = 1, \ldots, N,$$

where $C_0 = IV$. This means that for instance in the second step of the protocol the encryption of the message blocks A, N_A, N_B is done as,

$$\begin{aligned} C_1 &= E_K(A) + IV \\ C_2 &= E_K(N_A) + C_1 \\ C_3 &= E_K(N_B) + C_2 \end{aligned}$$

The same approach (mode usage) is applied in the third step of the protocol as well. Does this modification give a secure protocol. Motivate your answer.

Solution: This gives no more security than the ECB mode. Note that in the 2nd step B sends to A

$$C_1 = E_K(A) + IV, C_2 = C_1 + E_K(N_A), C_3 = C_2 + E_K(N_B).$$

The attacker can then easily compute

$$E_K(N_A) = C_1 + C_2$$

and

$$E_K(N_B) = C_2 + C_3$$

and use in the 3rd step to construct

$$C_1 = IV + E_K(N_B)$$

and

$$C_2 = C_1 + E_K(N_A).$$

Problem 30

We consider an LFSR of length n bits.

1. Explain why the generated key sequence cannot have a period longer than $2^n - 1$ bits.

   Solution An n bit LFSR has 2^n possible states. Thus the period is at most 2^n. However, the all zero state cannot appear in a maximal period sequence, since it would generate an all zero output.

2. Explain why an LFSR that generates maximal period sequences must have an even number of ones in its tap sequence (connection polynomial). (**5 points**) Hint: Consider the state with all ones.

   Solution If an LFSR has an odd number of taps and enters the state with all ones, then the bit to be shifted in is the xor of an odd nummber of ones and hence one. So, the device is stuck in this state.

3. The output sequence of an LFSR starts with 100000001. What is the minimal size of the LFSR? Your answer should exhibit an LFSR of this size that does produce the given sequence and give a motivation why no shorter LFSR will do.

   Solution The given sequence has 7 consecutive zeros. This means that an LFSR of size seven or less would lead to an all-zero state after the first bit and then just produce zeros. It is easy to construct an LFSR of size 8 that produces the given output: we put $c_8 = 1$ and choose arbitrary values for the other taps. With initial state 10000000, this will generate the desired output.

Problem 31

This problem concerns the DES cipher and modes of usage.

1. One important property which makes DES secure is that the S-boxes are non-linear. In this problem we are going to verify this property by computing the output of S_1 for several pairs of inputs. Show that $S_1(x_1) \oplus S_1(x_2) \neq S_1(x_1 \oplus x_2)$, where $\oplus$ denotes bitwise XOR, for:

$$x_1 = 000000, \quad x_2 = 000001$$
$$x_1 = 111111, \quad x_2 = 100000$$

Solution Using the S-box table we have

$$S_1(000000) = 14 = 0111; \;\; S_1(000001) = 4 = 0010;$$

$$S_1(000000) \oplus S_1(000001) = 0101.$$

$$S_1(x_1 \oplus x_2) = S_1(000001) = 0010 \neq S_1(000000) \oplus S_1(000001) = 0101$$

Similarly for the second pair,

$$S_1(111111) = 13 = 1011; \;\; S_1(100000) = 0 = 0000;$$

$$S_1(111111) \oplus S_1(100000) = 1011.$$

$$S_1(x_1 \oplus x_2) = S_1(011111) = 0000 \neq S_1(000000) \oplus S_1(000001) = 1011.$$

2. DESX was proposed by R. Rivest to protect DES against exhaustive key search. DESX uses:

- one 64-bit secret key W to perform pre- and postwhitening of data and
- a 56-bit DES key K.

DESX operates as follows,

$$C = W \oplus E_K(P \oplus W).$$

Show how the decryption is done.

Solution

$$P = D_K(C \oplus W) \oplus W.$$

3. Show that a similar construction,

$$C = W \oplus E_K(P)$$

without prewhitening is insecure and can be broken using an attack of complexity 2^{56}.

Solution This problem was treated at the class exercises. Assumption :

- A small number of plaintext/ciphertext pairs (P_i, C_i) available.

Then for all 2^{56} possible values of K we can compute $E_K(P_0)$ and $E_K(P_1)$.

For a correct guess we must have:

$$\begin{aligned} C_0 \oplus C_1 &= W \oplus E_K(P_0) \oplus W \oplus E_K(P_1) = \\ \\ & E_K(P_0) \oplus E_K(P_1). \end{aligned}$$

However, if key is not correct then the probability that $C_0 \oplus C_1 = E_K(P_0) \oplus E_K(P_1)$ is negligible (one may test further with more (P_i, C_i) pairs). Hence, checking $C_0 \oplus C_1 = E_K(P_0) \oplus E_K(P_1)$ for all keys K we find the right key.

Finally, W is computed as $W = C_i \oplus E_K(P_i)$. The complexity is approximately 2^{56} operations for testing all keys K.

Problem 32

We consider the RSA encryption.

1. To illustrate the RSA system, we use primes $p = 23$ and $q = 17$. As public encryption key we use $e = 3$. Compute the decryption key d. Show your computations !

   Solution We have that $n = 391$ and $\Phi(N) = (p-1)(q-1) = 352$. We compute d with the extended Euclidean algorithm:

   $$\begin{array}{cccc|c} 352 & 3 & 1 & 117 & -117 \\ 3 & 1 & 0 & 3 & 1 \\ & & & & 0 \end{array}$$

   Thus $d = -117 = 235$.

2. Describe in detail how the ciphertext $C = 165$ is decrypted. You must show that you understand how the algorithm for efficient modular exponentiation works.

   Solution To decrypt 165 means to compute 165^{235} (mod 391). We use the algorithm for modular exponentiation:

i	2^i	165^{2^i}
0	1	165
1	2	165^2
…		

   This table is continued until $i = 7$; the third column is computed by repeated squaring modulo 391. Finally, one notices that 235 in binary form is 11101011, so the final result is obtained by multiplying (modulo 235) the number in the third column in rows with $i = 0, 1, 3, 5, 6, 7$.

3. Suppose Bob has an RSA Cryptosystem with a large modulus n for which the factor- ization can not be found, e.g., n is 1024 bits long and Alice sends a message to Bob by representing

each alphabetic character as an integer between 0 and 25 (i.e., $A \to 0, B \to 1, \dots Z \to 25$) and then encrypting each letter as a separate plaintext character.
Describe how Oscar can easily decrypt a message which is encrypted in this way.

Solution A message consists of, let's say, m pieces of ciphertext $y_0, y_1, \dots, y_{m-1}$. However, the plaintext space is restricted to 26 possible values and the ciphertext space too. That means we only have to test 26 possible plain-text letters for each cipher-text letter:

$$\text{test } : y_i \overset{?}{=} j^e \pmod{n}, \quad j = 0, 1, \dots, 25.$$

Problem 33

This problem discusses a common problem related to the RSA cryptosystem.

1. Alice has decided to use RSA for encryption and has generated two large primes p and q and computed $N = pq$. She has also chosen encryption key $e_A = 3$ and computed her private key d_A.

   When her friend Bob hears about this, he also wants to use RSA. Alice assists him by choosing for him $e_B = 5$ and computing d_B, using the same N. Alice gives Bob his keys (N, e_B) and d_B.

   The next day their common friend Charlie sends message m encrypted to both Alice and Bob, using their respective encryption keys. However, the adversary Deborah eavesdrops and gets hold of the two ciphertexts c_A and c_B. Deborah also notices that Alice and Bob use the same N. Show how she can recover m. You may assume that $\gcd(m, N) = 1$.

   Solution We have that $c_A = m^3 \pmod{N}$ and $c_B = m^5 \pmod{N}$. We note that $5 \times 2 - 3 \times 3 = 1$, so Deborah computes

   $$c_B^2 \cdot (c_A^{-1})^3 \pmod{N} = m.$$

   Note that c_A^{-1} can be computed using the extended Euclidean algorithm, since $\gcd(m, N) = 1$.

2. Does Deborahs attack generalize to other values of e_A and e_B than 3 and 5 ?

   Solution As long as e_A and e_B satisfy $\gcd(e_A, e_B) = 1$, we know that there exist integers x and y with $xe_A + ye_B = 1$. Thus Deborah computes $c_A^x c_B^y \pmod{N}$ to recover m. If $\gcd(e_A, e_B) > 1$ this does not work and there does not seem to be an easy attack for Deborah.

Problem 34

We consider yet another published, flawed protocol for authentication and session key agreement, the Neuman-Stubblebine protocol. It employs a trusted third party and runs as follows:

$$\begin{aligned}
1.\ A \to B &: A, N_A \\
2.\ B \to T &: B, \{A, N_A, T_B\}_{K_{BT}}, N_B \\
3.\ T \to A &: \{B, N_A, K_{AB}, T_B\}_{K_{AT}}, \{A, K_{AB}, T_B\}_{K_{BT}}, N_B \\
4.\ A \to B &: \{A, K_{AB}, T_B\}_{K_{BT}}, \{N_B\}_{K_{AB}}
\end{aligned}$$

The protocol employs both timestamps and nonces. Some remarks:

- Alice initiates the run in message 1, sending her name and a nonce N_A to Bob.
- Bob contacts the trusted third party Trent, forwarding Alices information and adding a nonce N_B of his own and a timestamp T_B. Part of the message is encrypted with the key K_{BT} shared by Bob and Trent.
- Trent generates a session key K_{AB} to be used by Alice and Bob and sends to Alice a message with two encrypted parts, one for Alice and one for Bob, and Bobs nonce in the clear. The part encrypted for Bob, $\{A, K_{AB}, T_B\}_{K_{BT}}$, is called the ticket.
- Alice checks her nonce and forwards the ticket to Bob, together with Bobs nonce encrypted with the session key. This last piece convinces Bob both that the message is fresh and that the sender is Alice.

However, the system is flawed. Assume that keys and nonces have the same sizes in bits. Show how an adversary, eavesdropping on messages 1 and 2 of the initial protocol, may intercept and himself send a valid message 4 to Bob, claiming to be Alice, and thus complete the initial protocol and communicate with Bob using encryption with a session key that Bob believes he shares with Alice.

Solution The adversary eavesdrops and hears the two first messages. He then notices he can construct a valid message 4, as follows:

He uses the encrypted part of message 2, i.e. $\{A, N_A, T_B\}_{K_{BT}}$, as the first part of his message. It has the correct structure, with A first and B:s timestamp T_B last, and in the middle the nonce N_A playing the role of K_{AB}.

To complete the message, he just needs to encrypt N_B using N_A as key (both these were sent unencrypted in messages 1 and 2). The complete protocol run is:

$$\begin{aligned}
1.\ A \to B &: A, N_A \\
2.\ B \to T &: B, \{A, N_A, T_B\}_{K_{BT}}, N_B \\
3.\ T \to A &: \ldots \\
4.\ A \to B &: \{A, N_A, T_B\}_{K_{BT}}, \{N_B\}_{N_A}
\end{aligned}$$

The third message, from T to A, plays no role in the attack.