Cryptography and Network Security: Symmetric Cryptography Exercises

March 14, 2013

If x and y are strings, then $x\|y$ is their concatenation and $x \oplus y$ is the bitwise XOR of the strings.

1 Warmup

Problem 1.1 (Vaudenay)**.** The *One-Time Pad* (also known as the *Vernam Cipher*) is defined as follows. A plaintext is considered as a random variable $X \in \{0,1\}^n$, where n is some positive integer. It is encrypted with a uniformly distributed random key $K \in \{0,1\}^n$, independent of X, using a bitwise XOR operation. The ciphertext is thus $Y = X \oplus K$. Why is the One-Time Pad insecure if the key is used more than once?

Answer. Using the One-Time Pad more than once with the same key allows one to compute the XOR of two messages:

$$Y_1 \oplus Y_2 = (X_1 \oplus K) \oplus (X_2 \oplus K) = X_1 \oplus X_2.$$

This could lead one to determine information about the messages X_1 and X_2. □

Problem 1.2 (Boneh)**.** Data compression is often used in data storage or transmission. Suppose you want to use data compression in conjunction with encryption. Does it make more sense to

1. compress the data and then encrypt the result, or
2. encrypt the data and then compress the result.

Answer. Assuming you have a good encryption scheme, encrypting the data ensures that the result looks random to any efficient adversary. Compression algorithms can be considered a type of efficient adversary, hence any compression algorithm should not be able to find patterns in the encrypted data. As a result, compressing encypted data allows for little to no compression. Compressing first and then encrypting is more efficient. □

2 DES

Problem 2.1 (Boneh)**.** Before DESX was invented, the researchers at RSA Labs came up with DESV and DESW, defined by

$$DESV_{kk_1}(M) = DES_k(M) \oplus k_1 \text{ and}$$
$$DESW_{kk_1}(M) = DES_k(M \oplus k_1).$$

As with DESX, $|k| = 56$ and $|k_1| = 64$. Show that both these proposals do not increase the work needed to break the cryptosystem using brute-force key search. That is, show how to break these schemes using on the order of 2^{56} DES encryptions/decryptions. You may assume that you have a moderate number of plaintext-ciphertext pairs, $C_i = DES\{V/W\}_{kk_1}(M_i)$.

Answer. In both cases you can remove the dependency on k_1. For DESV we have

$$C_1 \oplus C_2 = DES_k(M_1) \oplus DES_k(M_2)$$

and for DESW we have

$$M_1 \oplus M_2 = DES_k^{-1}(C_1) \oplus DES_k^{-1}(C_2).$$

Once you determine k from these relations, it is easy to determine k_1. □

Problem 2.2. One variant of triple encryption with DES using two rather than three 56 bit keys is defined in the following way:

$$C = \mathbf{Enc}(K_2, \mathbf{Enc}(K_1, \mathbf{Enc}(K_1, P)))$$

Show that this cipher succumbs to a meet-in-the-middle-attack (write down the sets/lists you compute and between which you look for an overlap). How much time/space does the attack need?

Answer. Let $\hat{\mathbf{Enc}}(K_1, P) = \mathbf{Enc}(K_1, \mathbf{Enc}(K_1, P))$ be a DES encryption with 32 rounds instead of 16, since the initial permutation cancels out, where the round keys are equal in round i and $i + 16$. This operation needs roughly double times compared to an ordinary DES operation. Then we come to a a very similar situation as for the 2DES meet-in-the-middle-attack.

$$\mathbf{Enc}(K_2, \hat{\mathbf{Enc}}(K_1, P)) = \mathbf{Enc}(K_2, \mathbf{Enc}(K_1, \mathbf{Enc}(K_1, P))).$$

We can assume that we have one plaintext/ciphertext pair (P, C). Then by brute-force we search the whole keyspace for DES. We use every $K_i \in \{0,1\}^{56}$, $i \in \{1, \ldots, 2^{56}\}$ to create table with rows $(K_i, \mathbf{Dec}(K_i, C))$, where $\mathbf{Dec}(K_i, C) = C_i$. We sort the table with respect to the C_i entry (time $2^{56} \cdot \log 2^{56}$). Then we start encrypting P with each key $K_j \in \{0,1\}^{56}$, $j \in \{1, \ldots, 2^{56}\}$: $\hat{\mathbf{Enc}}(K_j, P) = C'_j$. For each encryption C'_j, we test if $C'_j = C_i$ for some i. If we find these we find the right keys with high probability. The meet-in-the-middle-attack needs roughly $3 \cdot 2^{56}$ DES operations, as building the table costs roughly 2^{56} operations, and encrypting P takes roughly $2 \cdot 2^{56}$ operations. Additionally a table of size $(56 + 64) \cdot 2^{56}$ bits is needed. □

3 Modes of Operation

Problem 3.1 (Boneh)**.** Let m be a message consisting of l AES blocks. Alice encrypts m using CBC mode and transmits the resulting ciphertext c to Bob. Due to a hardware error, ciphertext block number $l/2$ is corrupted during transmission. All other ciphertext blocks are transmitted and received correctly.

1. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted?
2. Answer the same question for randomized counter mode.

Answer.

1. When decrypting, only message blocks $l/2$ and $l/2+1$ depend on ciphertext block $l/2$, hence two plaintext blocks will be corrupted.

2. Counter mode generates a keystream which is XORed with the plaintext, hence the error will remain limited to plaintext $l/2$.

□

Problem 3.2 (Vaudenay)**.** Assume that someone sends encrypted messages using DES in the OFB mode of operation with a secret, but fixed, IV.

1. Show how to perform a known plaintext attack in order to decrypt transmitted messages.

2. Is it better with the CFB mode?

3. What about the CBC mode?

Answer.

1. In a known plaintext attack we are given a plaintext P and its corresponding ciphertext C. From these we can compute $X = P \oplus C$. Then as long as the key and IV remain fixed, we can compute the plaintext of any ciphertext C' via $P' = C' \oplus X$.

2. In CFB mode you will only have an issue with the first ciphertext block, i.e. you can determine the first plaintext block of any given ciphertext if the IV is fixed. The next values in the sequence depend on the plaintext. Similarly, if two plaintexts are equal on their first n blocks, then knowledge of one of the plaintexts allows one to recover the $(n+1)$the block of the other plaintext.

3. This type of attack does not work with CBC mode.

□

Problem 3.3 (Vaudenay)**.** Consider the encryption of an n-block message $x_1 \| \dots \| x_n$ with the block cipher E in CBC mode. Let $y_1 \| \dots \| y_n$ be the resulting encryption of the message in CBC mode.

1. Show that if there is a collision, i.e. $y_i = y_j$ for $i \neq j$, then one can extract information about the plaintext.

2. What is the probability of getting a collision when the block size is 64 bits?

3. For which n does the attack become useful?

Answer.

1. If $y_i = y_j$ for $i \neq j$, then
$$y_{i-1} \oplus x_i = y_{j-1} \oplus x_j$$
and since y_{i-1} and y_{j-1} are known one can deduce partial information about the plaintext from $x_i \oplus x_j = y_{i-1} \oplus y_{j-1}$.

2. From the birthday paradox we know that the probability of getting a collision when we have $n = \theta\sqrt{2^{64}}$ blocks available is approximately equal to $1 - e^{\frac{\theta^2}{2}}$.

3.

n	Size of data	Success probability
2^{17}	1 MB	$4.66 \cdot 10^{-10}$
2^{21}	16 MB	$1.19 \cdot 10^{-7}$
2^{24}	128 MB	$7.63 \cdot 10^{-6}$
2^{26}	512 MB	$1.22 \cdot 10^{-4}$
2^{27}	1 GB	$4.88 \cdot 10^{-4}$
2^{32}	32 GB	0.393
2^{33}	64 GB	0.865
2^{34}	128 GB	0.9997

□

Problem 3.4. A big drawback of the CBC mode is that it is highly sequential. One could try to circumvent this problem by using a modified CBC, called CBC*. CBC* uses an initial value $Y_0 = IV$, which is randomly chosen for every new encryption and sent together with the ciphertext. For $i = 1$ to L (where L is the number of message blocks) the encryption operation is defined as follows:

$$Y_i = M_i \oplus Y_{i-1}$$
$$C_i = E_K(Y_i).$$

Notice that the mode uses a deterministic encryption algorithm.

1. Show how the decryption operation for CBC* mode works.
2. Analyze the properties of the CBC* mode (efficiency, error propagation, synchronization, parallelizability, etc.).
3. Does the mode admit a chosen-plaintext attack that allows for distinguishing the encryption of two distinct messages?

Answer.

1. The decryption operation is defined as follows:

$$Y_i = E_K^{-1}(C_i)$$
$$M_i = Y_i \oplus Y_{i-1}.$$

2. Efficiency: same as CBC; Error propagation: an error in the ciphertext (e.g. flipped bit) introduces an error in 2 blocks: the corresponding decrypted and next blocks. The error does not leak the position of the introduced error in the ciphertext, because both decryptions are affected by the decryption operation; Synchronization: no self-synchronization; Parallelizablity: not in encryption, but yes in decryption.
3. A chosen-plaintext attack is possible with the use of only two message pairs. Let $M_1 = 0^n \| 1^n$ and $M_2 = 1^n \| 0^n$, where n is the size of the block cipher. The encryption of M_1 results in two ciphertext blocks which are different. The encryption of M_2 results in

$$C_1 = E_K(IV \oplus 1^n) \quad \text{and} \quad C_2 = E_K(IV \oplus 1^n \oplus 0^n),$$

hence $C_1 = C_2$.

□

Problem 3.5 (Boneh)**.** Let us see why in CBC mode an unpredictable IV is necessary for CPA security. Suppose a defective implementation of CBC encrypts a sequence of packets by always using the last ciphertext block of packet number i as the IV for packet number $i+1$ (up until a few years ago all web browsers implemented CBC this way). Construct an efficient adversary that wins the chosen plaintext attack (CPA) game against this implementation with advantage close to 1. In the CPA game the attacker submits packets (a.k.a. messages) to the challenger one by one and receives the encryption of those packets. The attacker then submits two messages m_0 and m_1 of equal length, which the challenger treats as the next pack in the packet stream. The challenger picks one of m_0 and m_1 at random and returns its ciphertext. If the attacker can tell whether m_0 or m_1 was encrypted, then the attacker wins the game.

Answer. The attacker first queries $m = 0^n \| 0^n$, after which it receives

$$c_1 = E_k(IV \oplus 0^n)$$
$$c_2 = E_k(c_1 \oplus 0^n).$$

Then the attacker submits $m_0 = c_1 \oplus c_2$ and $m_1 \neq m_0$ to the challenger, in which case it receives

$$c = \begin{cases} E_k(c_1) = c_2 & \text{in response to } m_0 \\ E_k(c_2 \oplus m_1) \neq c_2 & \text{in response to } m_1. \end{cases}$$

□

Cryptography and Network Security: Exercises

Alfredo Rial
Alfredo.Rial@esat.kuleuven.be

Public Key Cryptography

I General Exercises

1. Compute $10^{11} \bmod 21 \equiv ?$
2. Compute efficiently $\varphi(36)$
3. $7^{54} \bmod 26 \equiv ?$

   (Do not use repeated squaring)
4. $\gcd(1326, 7272) = ?$
5. Compute $97^{-1} \bmod 205$
6. Compute $3^{-1} \bmod 5$
   (Use extension of Fermat's Theorem)
7. Solve $122x \equiv 1 \bmod 343$
8. Find the smallest non-negative solution for the following system of congruences:

$$x = 8 \bmod 23$$
$$x = 3 \bmod 7$$
$$x = 3 \bmod 5$$

II RSA encryption/decryption

1. **a)** Consider an RSA encryption-system with modulus $n := 589 = 31 \cdot 19$. Choose the smallest possible public exponent. What is the corresponding secret exponent?
   b) Calculate the ciphertext for the message '55'.
   c) Decrypt your result using the Chinese remainder theorem; verify whether you retrieve the plaintext.

d) Is it a problem to have a common modulus for RSA? This means, Suppose a message m is encrypted twice with the public keys (n, e) and (n, f) with $gcd(e, f) = 1$, then can you break RSA given these two ciphertexts $c_e = m^e \bmod n$ and $c_f = m^f \bmod n$. If so, how? (Hint: the Euclidean algorithm allows to find integers x, y s.t. $gcd(a, b) = ax + by$)

e) Is a small public exponent e a security problem? Why? (Hint: Suppose an entity wishes to send the same message m to three entities whose public moduli are n_1, n_2, n_3, and whose encryption exponents are $e = 3$.)

2. Prove that if $n = pq$ is a product of two primes, then determining $\varphi(n)$ is equivalent to factoring n

III ElGamal

1. Given $p = 97$, $g = 5$, $y = 64$. Verify for a message $m = 81$ whether the signature $(r, s) = (13, 93)$ is a valid signature.

2. Show that a different random number k must be selected for each message signed; otherwise the private key x can be determined with high probability.

Cryptography and Network Security: Solutions

Alfredo Rial
Alfredo.Rial@esat.kuleuven.be

Public Key Cryptography

I General Exercises

1. Compute $10^{11} \bmod 21 \equiv ?$

   (a) Write 11 in binary representation: $11 = 1011$.

   (b) Compute the needed powers of 10 ($10^{2^i} \bmod 21$)

i	2^i	$10^{2^i} \bmod 21$	
0	1	10	∘
1	2	16	∘
2	4	4	
3	8	16	∘

   Multiply the needed terms from the table for $10^{11} \bmod 21$ to get:

$$10^{11} \equiv 10 \cdot 16 \cdot 16 \equiv 19 \bmod 21$$

2. Compute efficiently $\varphi(36)$

   $36 = 2^2 3^2$. Then $\varphi(36) = 36(1 - \frac{1}{2})(1 - \frac{1}{3}) = 12$.

3. $7^{54} \bmod 26 \equiv ?$

   (Do not use repeated squaring)

   $\varphi(26) = 12$. Then $7^{54} \bmod 26 \equiv 7^{54 \bmod 12} \bmod 26 \equiv 7^6 \bmod 26 = 25$.

4. $\gcd(1326, 7272) = ?$

   Use the Euclidean Algorithm on input $a = 7272$ and $b = 1326$.

$$\begin{aligned} 7272 &= 5 \cdot 1326 + 642 \\ 1326 &= 2 \cdot 642 + 42 \\ 642 &= 15 \cdot 42 + 12 \\ 42 &= 3 \cdot 12 + 6 \\ 12 &= 2 \cdot 6 + 0 \end{aligned}$$

   Return $d = 6$.

5. Compute $97^{-1} \bmod 205$

   Use the Extended Euclidean Algorithm on input $a = 205$ and $b = 97$. Set $x_2 \leftarrow 1$, $x_1 \leftarrow 0$, $y_2 \leftarrow 0$ and $y_1 \leftarrow 1$.

   $q \leftarrow 2 \quad r \leftarrow 11 \quad x \leftarrow 1 \quad y \leftarrow -2$
   $a \leftarrow 97 \quad b \leftarrow 11 \quad x_2 \leftarrow 0 \quad x_1 \leftarrow 1 \quad y_2 \leftarrow 1 \quad y_1 \leftarrow -2$

   $q \leftarrow 8 \quad r \leftarrow 9 \quad x \leftarrow -8 \quad y \leftarrow 17$
   $a \leftarrow 11 \quad b \leftarrow 9 \quad x_2 \leftarrow 1 \quad x_1 \leftarrow -8 \quad y_2 \leftarrow -2 \quad y_1 \leftarrow 17$

   $q \leftarrow 1 \quad r \leftarrow 2 \quad x \leftarrow 9 \quad y \leftarrow -19$
   $a \leftarrow 9 \quad b \leftarrow 2 \quad x_2 \leftarrow -8 \quad x_1 \leftarrow 9 \quad y_2 \leftarrow 17 \quad y_1 \leftarrow -19$

   $q \leftarrow 4 \quad r \leftarrow 1 \quad x \leftarrow -44 \quad y \leftarrow 93$
   $a \leftarrow 2 \quad b \leftarrow 1 \quad x_2 \leftarrow 9 \quad x_1 \leftarrow -44 \quad y_2 \leftarrow -19 \quad y_1 \leftarrow 93$

   $q \leftarrow 2 \quad r \leftarrow 0 \quad x \leftarrow 97 \quad y \leftarrow 205$
   $a \leftarrow 1 \quad b \leftarrow 0 \quad x_2 \leftarrow -44 \quad x_1 \leftarrow 97 \quad y_2 \leftarrow 93 \quad y_1 \leftarrow 205$

   Therefore, $1 = 205 \cdot (-44) + 97 \cdot 93$ and thus $97^{-1} = 93 \bmod 205$.

6. Compute $3^{-1} \bmod 5$
   (Use extension of Fermat's Theorem)

   $\varphi(5) = 4$. Therefore $3^{-1} \bmod 5 = 3^{4-1} \bmod 5 = 3^{4-1} \bmod 5 = 2$.

7. Solve $122x \equiv 1 \bmod 343$

   $\varphi(343) = 343 \cdot (1 - \frac{1}{7}) = 294$. Then $x = 122^{294-1} \bmod 343 = 194$.

8. Find the smallest non-negative solution for the following system of congruences:

$$x = 8 \bmod 23$$
$$x = 3 \bmod 7$$
$$x = 3 \bmod 5$$

   Use the Chinese Remainder Theorem.

   $x_1 = 8 \quad M_1 = 35 \quad N_1 = 35^{-1} \bmod 23 = 2$
   $x_1 = 3 \quad M_2 = 115 \quad N_2 = 115^{-1} \bmod 7 = 5$
   $x_1 = 3 \quad M_3 = 161 \quad N_3 = 161^{-1} \bmod 5 = 1$

   $M = 805$. Then $x = \sum_i x_i M_i N_i \mod M = 2768 \bmod 805 = 353$.

II RSA encryption/decryption

1. **a)** Consider an RSA encryption-system with modulus $n := 589 = 31 \cdot 19$. Choose the smallest possible public exponent. What is the corresponding secret exponent?

$\lambda(589) = 180$. The smallest possible public exponent is $e = 7$. Use the Extended Euclidean Algorithm to compute $d = e^{-1} \bmod 180 = 7^{-1} \bmod 180 = 103$.

b) Calculate the ciphertext for the message '55'.

$c = m^e \bmod n = 55^7 \bmod 589 = 499$.

c) Decrypt your result using the Chinese remainder theorem; verify whether you retrieve the plaintext.

$v_1 = 499^{103} \bmod 31 = 24$ $v_2 = 499^{103} \bmod 19 = 17$

Apply the Chinese Remainder Theorem to the following equations:

$$x = 24 \bmod 31$$
$$x = 17 \bmod 19$$

Then $x = \sum_i x_i M_i N_i \mod M = 12424 = 55 \bmod 589$.

d) Is it a problem to have a common modulus for RSA? This means, Suppose a message m is encrypted twice with the public keys (n, e) and (n, f) with $gcd(e, f) = 1$, then can you break RSA given these two ciphertexts $c_e = m^e \bmod n$ and $c_f = m^f \bmod n$. If so, how? (Hint: the Euclidean algorithm allows to find integers x, y s.t. $gcd(a, b) = ax + by$)

Run the Extended Euclidean Algorithm to obtain (x, y) such that $ex + fy = 1$. Compute: $c_e^x \cdot c_f^y = (m^e)^x \cdot (m^f)^y = m^{ex+fy} = m \bmod n$.

e) Is a small public exponent e a security problem? Why? (Hint: Suppose an entity wishes to send the same message m to three entities whose public moduli are n_1, n_2, n_3, and whose encryption exponents are $e = 3$.)

$$c_1 = m^3 \bmod n_1$$
$$c_2 = m^3 \bmod n_2$$
$$c_3 = m^3 \bmod n_3$$

$$m^3 = c_1 \bmod n_1$$
$$m^3 = c_2 \bmod n_2$$
$$m^3 = c_3 \bmod n_3$$

Use the Chinese Remainder Theorem to obtain $m^3 \bmod n_1 n_2 n_3$. Since $m^3 < n_1 n_2 n_3$, $m = \sqrt[3]{m^3}$.

2. Prove that if $n = pq$ is a product of two primes, then determining $\varphi(n)$ is equivalent to factoring n

First, we show that, if we can factor n, then we can compute $\varphi(n)$. On input n, factor n to obtain p and q and output $\varphi(n) = (p-1)(q-1)$.

Second, we show that, if we can compute $\varphi(n)$, then we can factor n. On input n, we can solve the following system of equations:

$$\begin{aligned} p \cdot q &= n \\ p + q &= n + 1 - \varphi(n) \end{aligned}$$

III ElGamal

1. Given $p = 97$, $g = 5$, $y = 64$. Verify for a message $m = 81$ whether the signature $(r, s) = (13, 93)$ is a valid signature.

   $g^m = 5^{81} \bmod 97 = 19$. $y^r \cdot r^s = 64^{13} \cdot 13^{93} \bmod 97 = 19$. Therefore, it is valid.

2. Show that a different random number k must be selected for each message signed; otherwise the private key x can be determined with high probability.

   Let a signer sign two messages m_1 and m_2 with the same random value k.

   $$\begin{aligned} (r_1, s_1) &= (g^k \bmod p, m_1 - xr_1)k^{-1} \bmod p - 1) \\ (r_2, s_2) &= (g^k \bmod p, m_2 - xr_2)k^{-1} \bmod p - 1) \end{aligned}$$

   Therefore $r_1 = r_2 = r$ and we have the following:

   $$\begin{aligned} (r, s_1) &= (g^k \bmod p, m_1 - xr)k^{-1} \bmod p - 1) \\ (r, s_2) &= (g^k \bmod p, m_2 - xr)k^{-1} \bmod p - 1) \end{aligned}$$

   We can relate the two equations by solving them for $-xr$.

   $$(s_1 - s_2)k = m_1 - m_2 \bmod p - 1 \tag{1}$$

   If $(s_1 - s_2)$ was invertible modulo $p - 1$ (i.e., $\gcd(s_1 - s_2, p - 1) = 1$), k has a unique solution. If $\gcd(s_1 - s_2, p - 1) = d$, then there can be d solutions, and for each of them we can choose the correct one by checking whether g^k equals r. Once k is found, we need to solve the following equation:

   $$xr = (m_1 - ks_1) \bmod p - 1 \tag{2}$$

   Again, if r is invertible modulo $p - 1$, there is a unique solution for x. Otherwise if $\gcd(r, p-1) = d$, there can be d solutions, and we choose the correct solution for x by checking whether g^x equals y.

Exercises and solutions public-key cryptography

This document contains some exercises and their solutions related to the lecture(s) on public-key cryptography. You are advised to try to solve the exercises yourself, before you read the answers.

Exercise 1: RSA

1. Public operation
   Given: $n = 11413$, $e = 33$
   Question: encrypt the message $m = 81$

2. Secret operation
   Given: $p = 101$, $q = 113$
   Question: $\lambda(n)$, d; decrypt the ciphertext using the Chinese Remainder Theorem

Exercise 2: Digital signature based on the discrete logarithm

1. ElGamal
   Given: $p = 97$, $g = 5$, $y = 64$
   Question: Verify for the message $m = 81$ the signature $r = 13$, $s = 93$

2. Alternative mechanism
   Let g denote a generator modulo the prime number p
   $y = g^x \bmod p$ and $r = g^k \bmod p$
   Compute $s = (m - rk)x^{-1} \bmod p - 1$
   The signature for m is the tuple (r, s)

   Given: $p = 97$, $g = 5$, $x = 13$
   Questions:

   - Generate the signature for $m = 81$
   - Give the equation that is to be used to verify the signature. Show that for this verification, only public values are needed. Do the verification.

Exercise 3 (past exam)
There exist several variants of the ElGamal scheme for digital signatures. Some variants allow to recover the message from the signature like with RSA (the message should contain some redundancy). It's also possible to work with an RSA modulus $n = p \cdot q$ instead of a prime number p (alas this results in a few extra security concerns).

We consider here the following variant: the signature for a message m consists of two numbers (r, s).

r is determined from $r \equiv m \cdot \alpha^{-k} \bmod n$

s is determined from $\alpha^k \equiv \alpha^{r^{-1} \cdot x} \cdot \alpha^{-s \cdot r^{-1}} \bmod n$ (1)

Here x denotes the secret key, α an element with 'large' order en k the random number used for the computation of the signature.

Use the following numerical values:

$\alpha = 5$, $n = 77$, $m = 21$, $e =$ your exam number

$k =$ the largest prime number smaller than 70

$x = (e \bmod 4) + 21$

The redundancy lies in the fact that m is always a multiple of 7.

1. Compute the signature (r, s).
2. Compute the pubklic key y.
3. Show that m can be recovered from (1) and that this allows to verify the signature using only public parameters (hint: multiply both sides of the equation by r).
4. Verify the correctness of your numerical result for the signature.
5. Compute y efficiently using the Chinese Remainder Theorem.
6. Compute efficiently the order of α mod n.

Give clearly *all* intermediate results in the computations. First explain the general formula and then fill out the numerical values. For the choice of computing algorithms, pretend that the numbers are "large" numbers.

Exercise 4 (past exam)

1. Given that $x \equiv 8 \bmod 23$, $\equiv 3 \bmod 7$ and $\equiv 3 \bmod 5$. Further, $0 \leq x < 805$.
   Find$x$.
2. What is the order of 4 mod 1001?
3. What values can the order of a number modulo 1001 take?

Exercise 5: Primality tests

1. Determine all bases a for which 15 is pseudo-prime and all bases for which 15 is a strong pseudo-prime.
2. Check whether 6601 is prime using Fermat's test and using Rabin's test.

Exercise 6 (past exam)

Consider an RSA variant where the modulus n consists of 3 prime numbers: $n = p \cdot q \cdot r$ where $p = 5$, $q = 7$ and $r = 11$. The encryption is defined by $c = m^e \bmod n$, the decryption by $m = c^d \bmod n$.

1. Choose for the encryption exponent e the smallest possible value. Determine the corresponding value for d.

2. Determine the ciphertext c for the message $m = 12$.

3. Decrypt c using the Chinese Remainder Theorem.

4. Determine 4 fixed points for this value of n and e. Fixed points are messages m for which $c = m$. How can you determine more fixed points? (Hint: *Don't* look for them exhaustively; use the Chinese Remainder Theorem.)
   What is the implication on the security of RSA and on the security of this RSA variant?

Verify all your computations. Give clearly *all* intermediate results in the computations. First explain the general formula and then fill out the numerical values. For the choice of computing algorithms, pretend that the numbers are "large" numbers. Work as if $2^{26} = 67\,108\,864$ is the largest number that your pocket calculator can handle.

Exercise 7
A Pentium Pro 200 MHz computer can multiply two 32-bit numbers in 1 clock cycle. Consider an RSA modulus n, counting k bits, and the numbers a and b, both also of length k. Choose as examples $k = 512$ and $k = 1024$.

1. How long does it take to compute $a \cdot b$?

2. How long does it take to compute $a \cdot b \bmod n$? (hint: there exists an algorithm that makes modular multiplication about two times slower than an ordinary multiplication.)

3. How long does it take to compute $a^2 \bmod n$?

4. How long does is take to compute $a^e \bmod n$, with $e = 3$, $e = 65537$, $e =$ a random 40-bit number, $e =$ a k-bit number?

5. Let n be an RSA modulus, e the public exponent, and d the secret exponent. How long does a decryption take, with and without the Chinese Remainder Theorem?

6. How long does the decryption take (with and without the Chinese Remainder Theorem) on a 5 MHz 8-bit processor, where a multiplication takes 5 cycles?

Exercise 8
Consider the same Pentium Pro as in the previous question. For the generation of a 1024-bit RSA moduluas we use a sieve that eliminates the prime numbers 2, 3, 5 and 7 (this takes 100 microseconds per number). Subsequently we use the Rabin-Miller test, which eliminates almost all composite numbers after 1 step. If we found a candidate prime number, we do 7 additional tests. How long does it take on average to generate an RSA modulus?

Exercise 9
Three users have RSA moduli $n_1 = 87$, $n_2 = 115$ and $n_3 = 187$ respectively. They use exponent $e = 3$. We see the following ciphertexts on the communication channel: $c_1 = 5$, $c_2 = 20$ and $c_3 = 181$. We suspect that these ciphertexts correspond to a unique message. Find this message.

Answers to Exercise 1 en 2

These solutions are merely a summary. At the exam, you have to give also *all intermediate results* of a calculation like e.g. $81^{33} \bmod 11413 = 4195$. Only then we can distinguish numerical mistakes from others.

1. RSA
   (a) Public operation
   Answer: $c = 81^{33} \bmod 11413 = 4195$
   (b) Secret operation
   answer:
   $\lambda(n) = (100 \cdot 112)/4 = 2800$
   $13 \cdot 2800 - 1103 \cdot 33 = 1 \Rightarrow d = -1103 \bmod 2800 = 1697$
   $c_p = 54$, $d_p = 97$, $c_p^{d_p} \bmod p = 81$
   $c_q = 14$, $d_q = 17$, $c_q^{d_q} \bmod q = 81$
   $p^{-1} \bmod q = 47$, $q^{-1} \bmod p = 59$
   $m = 59 \cdot 113 \cdot 81 + 47 \cdot 101 \cdot 81 \bmod 11413 = 81$
2. Digital signature based on discrete logarithm:
   (a) ElGamal
   Answer:
   $5^{81} = 64^{13} \cdot 13^{93} \bmod 97$?
   $19 = 33 \cdot 77 \bmod 97$!
   (b) Alternative mechanism
   Answer:
   - $y = 29$, choose $k = 10$, then $r = 53$
     $s = (81 - 10 \cdot 53) \cdot 13^{-1} \bmod 96 = 31 \cdot 37 \bmod 96 = 91$
     signature $(r, s) = (53, 91)$
   - $sx + rk = m \bmod p - 1 \Leftrightarrow g^{sx} g^{rk} = g^m \bmod p \Leftrightarrow y^s r^r = g^m \bmod p$
     $29^{91} \cdot 53^{53} = 5^{81} \bmod p$?
     $7 \cdot 72 = 19 \bmod p$!

Answer to Exercise 3

Since $a^{\lambda(n)} = 1 \bmod n$ for $\gcd(a, n) = 1$, we compute in the exponent modulo $\lambda(n)$ (this holds for all operations in the exponent: multiplication, multiplicative inverse, additive inverse).

1. $k = 67$ (sieve of Eratosthenes)
   $\lambda(n) = \operatorname{lcm}(6, 10) = 30$
   $\alpha^{-k} = 5^{-67} \bmod 77 = 1^3 \cdot 5^{-67} \bmod 77 = (5^{30})^3 \cdot 5^{-67} \bmod 77 = 5^{23} \bmod 77 = 59$
   or: $\alpha^{-k} = 5^{-67} \bmod 77 = 5^{-67 \bmod 30} \bmod 77 = 5^{23} \bmod 77 = 59$.

$r = 21 \cdot 59 \bmod 77 = 7$
Take $x = 21$. It follows that $s = x - kr \bmod \lambda(n) = 21 - 67 \cdot 7 \bmod 30 = 2$

2. $y = 5^{21} \bmod 77 = 27$
3. $m = ry^{r^{-1}}\alpha^{-sr^{-1}} \bmod n$
4. $r^{-1} \bmod 30 = 13$, $y^{r^{-1}} \bmod n = 27^{13} \bmod 77 = 48$
   $\alpha^{r^{-1}} \bmod n = 5^{13} \bmod 77 = 26$
   $(\alpha^{r^{-1}})^{-s} \bmod n = 26^{-2} \bmod 77 = 26^{28} \bmod 77 = 9$
   $m = 7 \cdot 48 \cdot 9 \bmod 77 = 21$
   m is a multiple of 7 and hence a valid message

Answer to Exercise 4

1. We choose an x of the following form:

$$x = (\alpha x_1 + \beta x_2 + \gamma x_3) \bmod n$$

with $x_1 = 3$, $x_2 = 3$, $x_3 = 8$. Define $m_1 = 5$, $m_2 = 7$, $m_3 = 23$.

α satisfies the following equations:

$$\begin{aligned} \alpha &= 1 \bmod m_1 \\ \alpha &= 0 \bmod m_2 \\ \alpha &= 0 \bmod m_3 \,. \end{aligned}$$

We find $\alpha = m_2 \cdot m_3 \cdot \alpha'$ with $\alpha' \cdot m_2 \cdot m_3 = 1 \bmod m_1$ or

$$\alpha' = (m_2 \cdot m_3)^{-1} \bmod m_1 \,.$$

Numerically: $\alpha' = (7 \cdot 23)^{-1} \bmod 5 = (161)^{-1} \bmod 5 = 1^{-1} \bmod 5 = 1$
(in general we find the inverse using Euclid's algorithm.)
Hence $\alpha = m_2 \cdot m_3 \cdot \alpha' = 1 \cdot 23 \cdot 5 = 161$.

Analogously we find $\beta = 575$ en $\gamma = 70$.

Hence

$$x = (161 \cdot 3 + 575 \cdot 3 + 70 \cdot 8) \bmod 805 = 353 \,.$$

Two remarks:

- The correctness of the solution can easily be verified. Hence, do this in order to eliminate numerical mistakes!
- Since $x_1 = x_2$ it follows immediately that $x \equiv 3 \bmod 35$, which can be used to simplify the computations significantly

2. Step 1: use trial division be small prime numbers to find the prime factors of 1001. The first prime factor equals 7, which leaves only 143 to factorize. For this, it suffices to try out all prime numbers smaller than $\sqrt{143}$. 11 is a factor, and hence we obtain $1001 = 7 \cdot 11 \cdot 13$.

   The order of a number modulo n is always a divisor of $\lambda(n)$. In this case $\lambda(n) = \text{lcm}(6, 10, 12) = 60$.

   The divisors of $60 = 2^2 \cdot 3 \cdot 5$ zijn: $\{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$.

3. ord (4) mod1001 $= \min_d(d \mid 60 \text{ and } 4^d \equiv 1 \bmod 1001)$.

   After the previous this is easy.
   We see immediately that the order of 4 needs to be larger than 4, since $4^4 = 256 < 1001$.
   We know that $4^{60} \equiv 1 \bmod 1001$.
   Compute $4^{30} \bmod 1001 = 1$.
   We know then that the order equals 30 or is a divisor of $30 = 2 \cdot 3 \cdot 5$. Compute $4^{30/2} \bmod 1001 = 155$, $4^{30/3} \bmod 1001 = 529$ and $4^{30/5} \bmod 1001 = 92$. Since these 3 numbers are different from 1, the order of 4 mod 1001 equals 30.

Answer to Exercise 5

This exercise requires quite some computational effort. Its purpose is to give some extra examples of primality tests rather than to be an example of an exam question.

1.i) A number s is a *basis a pseudo-prime* if $(1 < a < s)$: $a^{s-1} \equiv 1 \bmod s$. For $s = 15$ we obtain:

$a = 2$: $2^2 \bmod 15 = 4$, $2^4 \bmod 15 = 1$ en $2^8 \bmod 15 = 1$.
Hence $2^{14} \bmod 15 = 2^2 \cdot 2^4 \cdot 2^8 \bmod 15 = 4 \cdot 1 \cdot 1 \bmod 15 = 4$.
$a = 3$: $3^2 \bmod 15 = 9$, $3^4 \bmod 15 = 6$ en $3^8 \bmod 15 = 6$.
Hence $3^{14} \bmod 15 = 9 \cdot 6 \cdot 6 \bmod 15 = 9$.
$a = 4$: $4^2 \bmod 15 = 4^4 \bmod 15 = 14^8 \bmod 15 = 1$.
Hence $4^{14} \bmod 15 = 1$.

Analogously for $a = 5, 6, ..., 14$ resp. $a^{14} \bmod 15 = 10, 6, 4, 4, 6, 10, 1, 9, 4, 1$. 15 is hence a basis a pseudo-prime for $a \in \{4, 11, 14\}$.

2.ii) A number s is a *strong basis a pseudo-prime* $(1 < a < s)$ if the following conditions are met. Write $s - 1 = 2^\nu \cdot s'$: either $a^{s'} \equiv 1 \bmod s$ orl $a^{2^k \cdot s'} \equiv -1 \bmod s$ for a k with $0 \leq k < \nu$. Now $s - 1 = 15 - 1 = 2^1 \cdot 7$ or $s' = 7$ and $\nu = 1$. A number can be a strong basis a pseudo-prime only if it is a basis a pseudo-prime. Hence we need to examine only 3 elements: 4, 11 and 14.

$a = 4$: $4^2 \bmod 15 = 4^4 \bmod 15 = 1$.
Hence $4^7 \bmod 15 = 4 \cdot 1 \bmod 15 = 4$. Since this is different from 1, we have that 15 is not a strong basis 4 pseudo-prime. (Note that it follows that 4 is a square root of 1 mod 15 different from ± 1.)

$a = 11$: $11^2 \bmod 15 = 1$, $11^4 \bmod 15 = 1$.
Hence $11^7 \bmod 15 = 11 \cdot 1 \bmod 15 = 11$. Same conclusion as with 4.
$a = 14$: $14^2 \bmod 15 = 1$, $14^4 \bmod 15 = 1$.
Hence $14^7 \bmod 15 = 14 \cdot 1 \bmod 15 = 14 = -1$. Hence the condition is met for $k = 0$.

Remark: it follows that there are 4 square roots of 1 modulo 15: 1, 4, 11 en 14 (-1). For $a = 2, 3, 4, 5, 6, ..., 14$ we have that $a^7 \bmod 15 = 8, 12, 4, 5, 6, 13, 2, 9, 10, 11, 3, 7, 14$.

1.i) Fermat's little theorem tells us that an integer n is composite if $a^{n-1} \not\equiv 1 \bmod n$ for an a with $\gcd(a, n) = 1$ (notice that if we find an a with $1 < a < n$ and $\gcd(a, n) > 1$, then n can't be composite).

Example: $2^{6600} \bmod 6601 = 1$.

For each a between 1 and 6601 where $\gcd(a, n) = 1$ we find (using a computer program) that $a^{6600} \bmod 6601 = 1$, while we know from the Rabin-Miller primality test that 6601 is a composite (see 2.ii).

We determine the factorisation of $6601 = 7 \cdot 23 \cdot 41$ (this can be done by looking for small prime factors) and observe that $7-1|6601-1$, $23-1|6601-1$ and $41-1|6601-1$.

Numbers for which $p_i - 1|n - 1$, with $n = p_1 \cdots p_r$ with the factorisation p_i of n, are called Carmichael numbers. All Carmichael numbers satisfy for each a between 1 and n with $\gcd(a, n) = 1$ Fermat's little theorem

Observe that for example $91^{6600} \bmod 6601 = 3773$, but also $\gcd(6601, 91) = 7$.

2.ii) The Rabin-Miller primality test accepts a number n as a possibly prime number if for a random a between 1 and n either $a^{s'} \equiv 1 \bmod s$ or $a^{2^k \cdot s'} \equiv -1 \bmod s$ for a k with $0 \leq k < \nu$ (for the definition of s' and ν, see 1.ii). For $n = 6601$ we obtain $\nu = 3$ and $s' = 825$. Take for example $a = 117$, then $n = 6601$ is not a basis a strong pseudo-prime: firstly, we compute $117^r \bmod 6601$, for $r = 6, 9, 18, 72, 144, 576$ we obtain respectively $3606, 3548, 197, 1513, 5223, 5881$ (we used here a variant of the square and multiply method; one can also use other steps to reduce the computational effort). We derive that $117^{825} \bmod 6601 = 1749 \neq 1$. We find then $1749^2 \bmod 6601 = 2738$, $2738^2 \bmod 6601 = 4509$, en $4509^2 = 1 \bmod 6601$. This basis proves hence that 6601 is not a prime number priemgetal: we found a square root of 1, different from ± 1.

On the other hand, taking $a = 141$, we obtain $141^{825} \bmod 6601 = 1$, from which we could conclude that 6601 is a prime number. For $a = 195$ we obtain $195^{825} \bmod 6601 = 6600 = -1$, which also makes the primality test succeed.

Note: there are no a for $i = 1, 2$ where $a^{2^i \cdot 825} \bmod 6601 = 6600$ (this follows from a test with a computer program).

Exercises for the course
Cryptography and Network Security

[Part I]

Stefaan Seys

October 2002

1. $14^3 \bmod 17 \equiv (-3)^3 \bmod 17 \equiv 7$
2. $5^{101\times 103} \bmod 97 \equiv ?$

$$\begin{aligned} 5^{101\times 103} \bmod 97 &\equiv (5^{101})^{103} \bmod 97 \\ &\equiv (5^5)^7 \bmod 97 \equiv 10 \end{aligned}$$

3. $7^{54} \bmod 26 \equiv ?$

   $26 = 13 \times 2 \Longrightarrow \phi(26) = (13-1)(2-1) = 12$

$$\begin{aligned} 7^{54} \bmod 26 &\equiv 7^{54 \bmod 12} \bmod 26 \\ &\equiv 7^6 \bmod 26 \\ &\equiv (343 \bmod 26)(343 \bmod 26) \\ &\equiv 5.5 \equiv 25 \end{aligned}$$

   Note that here again it is possible to write $7^6 = 49^3 \equiv (-3)^3 \bmod 26$

4. $13^{97} \bmod 103 \equiv ?$

   Write 97 in binary form: $97 = \texttt{1100001}$. Now use the following table to efficiently compute the terms 13^{2^i}.

2^i	$13^{2^i} \bmod 103$
1	13
2	66
4	30 ($\equiv 66^2 \bmod 103$)
8	76
16	8
32	64
64	79

Use the values in the table to compute the final result: $13^{97} \bmod 103 \equiv 13^{64}13^{32}13^{1} \equiv 13.64.79 \equiv 14$.

5. $\gcd(1326, 7272) = ?$

   Use Euclides to solve this problem:

$$\begin{aligned} 7272 &= 5.\mathbf{1326} + \mathit{642} \\ \mathbf{1326} &= 2.\mathit{642} + 42 \\ 642 &= 15.42 + 12 \\ 42 &= 3.12 + 6 \\ 12 &= 2.6 + 0 \end{aligned}$$

   Proof that this works: suppose that $x = ax + b$, then

$$\begin{aligned} \gcd(x, y) &= \gcd(ay + b, y) \\ &= \gcd(b, y) \end{aligned}$$

Extra exercises on cryptography (2001)

Stefaan Seys

October 2001

General exercises

1. Show that $3 \mid n^3 - n$ for all n.

   Answer:
   We are asked to show that $n^3 - n \equiv 0 \pmod 3$ for all n; that is, $n^3 \equiv n \pmod 3$. Now, any number n is congruent to 0, 1 or 2 modulo 3; hence we only need to verify the statement for the three number 0, 1 and 2.

2. Proof that if $\gcd(m, n) = 1$, then $a \equiv b \pmod m$ and $a \equiv b \pmod n$ if and only if $a \equiv b \pmod{mn}$.

   Proof:
   If $m \mid a - b$, then $a - b = km$ for some integer k. Now $n \mid a - b$, but $\gcd(n, m) = 1$; hence $n \mid k$; therefore, $nm \mid a - b$ or $a \equiv b \pmod{mn}$.

   Conversely, it is clear that if $a \equiv b \pmod{mn}$, then $a \equiv b \pmod d$ for any divisor d of mn, in particular, for m and n.

3. Prove that if $a \equiv b \pmod m$, then $\gcd(a, m) = \gcd(b, m)$.

   Proof:
   If $a \equiv b \pmod m$, then $a - b = km$ or $a = km + b$ for some integer k.
   So $\gcd(a, m) = \gcd(km + b, m) = \gcd(b, m)$.

4. Rabin's squaring function is defined as follows: $\text{RABIN}(x, n = pq) \equiv x^2 \pmod n$, with p and q distinct odd primes. Prove that, unlike RSA, this function has *no* unique defined inverse, but is a 1 to 4 function.
   Hint: use the Chinese Remainder Theorem.

   Proof:
   If $a \equiv x^2 \pmod p$ then x and $-x$ are the distinct square roots of a modulo p, and if $a \equiv y^2 \pmod q$ then y and $-y$ are the distinct square roots of a modulo q. Then, there are four solutions to $a \equiv z^2 \pmod n$, constructed as follows. We are looking for a value z that satisfies the following two equations:

   $$\begin{cases} z \equiv \{x, -x\} \pmod p \\ z \equiv \{y, -y\} \pmod q \end{cases}$$

   Let α, β be the CRT coefficients, then the value z can be calculated using the CRT resulting in four distinct solutions for z:

   $$z \equiv \alpha\{x, -x\} + \beta\{y, -y\} \pmod{n = pq}.$$

Block ciphers

1. The Merkle-Hellman attack on triple DES begins by assuming a value of $A = 0$ (see figure 1). Then, for each of the 2^{56} possible value of K_1, the plaintext P that produces $A = 0$ is determined. Describe the rest of the algorithm.

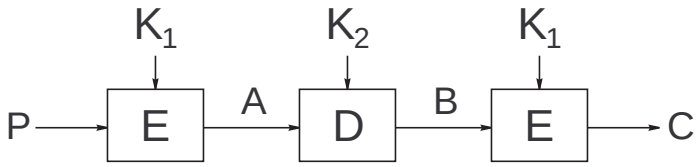

Figure 1: Triple encryption

Answer:

a) Create table 1 ([$K_1 = i$ | P_i]) so that for every $K_1 = i$ you have $E_{K_i}(P_i) = 0$.
b) Obtain all the ciphertexts C_i for the P_i's in table 1.
c) Decrypt all the C_i's with key $K_i = i$ and create table 2 ([C_i | B_i]).
d) For one of the plaintexts P_i, the encryption will look like this (see figure 2).

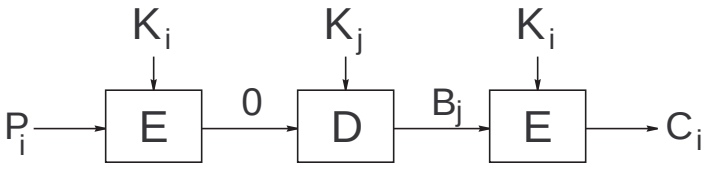

Figure 2: Encryption for one of the plaintexts P_i

Now we have to find out for which P_i this is true.

From figure 2 we can see that the B_j we obtain, is the decryption with key K_j of 0 or $B_j = D_{K_j}(0)$. If we invert this, we obtain $0 = E_{K_j}(B_j)$.

In step a) we created table 1 that contains all plaintexts that encrypt to 0, so B_j has to be one of these plaintexts.

So now we look for a match between the P_i entries in table 1 and the B_j entries in table 2. Once we find a match ($P_i = B_j$), key K_1 is very likely to be i and key K_2 is very likely to be j.

We can check this using a couple of other plaintext/ciphertext pairs.

RSA

1. Prove that if $n = pq$ is a product of two distinct primes. Then determining $\phi(n)$ is equivalent to factoring n ($\phi(n)$ is Euler's Phi function).

   Proof:
   If we know the factorization of n, then $\phi(n) = (p-1)(q-1)$.
   On the other hand, if n and $\phi(n)$ are known, then it is easy to compute the factors p and q. We write $q = n/p$ and substitute this in the formula for $\phi(n)$.

   $$\begin{aligned} \phi(n) &= (p-1)(q-1) \\ &= (p-1)(n/p-1). \end{aligned}$$

   Rewriting shows that p is one root of the quadratic equation

   $$p^2 - (n+1-\phi(n))p + n = 0.$$

   It is easy to see that q is also a root of this quadratic equation. Hence we have shown that determining $\phi(n)$ is equivalent with factoring n.

2.  Hacking and cracking (HAC p. 287).

   The RSA public key for a secret agency is

   $$n = 138148175776919890436438986961112008326178779309$$

   with public encryption exponent

   $$e = 257.$$

   Using your computer hacking skills, you broke into their mainframe and retrieved an encrypted version of their private decryption exponent d

   $$E_k(d) = \begin{bmatrix} 0855 & 0600 & 1177 & 1309 & 0640 & 0727 \\ 1653 & 0773 & 1652 & 1985 & 0922 & 0877 \\ 1404 & 0770 & 1820 & 2246 & 0700 & 0736 \\ 0924 & 0590 & 1014 & 1052 & 0764 & 0820 \end{bmatrix}.$$

   And you also discovered that the last two digits of the private decryption exponent d are 01.

   You know (because this is public information) that the company uses a very basic algorithm to encrypt their private key d.
   The algorithm $E_k(d)$ works as follows:

   a) Zeros are prepended to d so that the number of digits of d is a multiple of 8. So now we have a number $00\ldots0d$ of length $k \cdot 8$.
   b) The digits of d are combined into groups of 2 and these groups are put into vectors of length 4. All these vectors are combined in a matrix D of dimension $4 \times k$. The first digits are placed in the top left corner of the matrix.

c) The matrix D is left-multiplied by a 4×4 matrix M and the result is the ciphertext C ($C = E_k(d) = MD$). This matrix M is filled with 16 numbers between 0 and 10 (M can be seen as the secret key k of the encryption system).

Because you don't want to try all possibilities for the matrix M, you plan your next move: you dress up as a maintenance guy/girl and you go to the agency's building to secretly place a hidden surveillance camera and transmitter pointed at the CEO's keyboard. Back in your discretely parked grocery van you patiently wait for the CEO to type in the agency's password. After some time, you are able to collect the following information about the secret key M.

$$M = \begin{bmatrix} 4 & 8 & 4 & 3 \\ 3 & 10 & 5 & 10 \\ 9 & 10 & ? & ? \\ 0 & 8 & ? & ? \end{bmatrix}$$

Determine the prime factors of n.

Solution:
The attack on the encryption scheme of the secret encryption exponent d is trivial. You can for example use Matlab to construct all possible matrices M (11^4 possibilities), invert them if possible and decrypt the last part of d (decryption = left-multiplication with M^{-1}). Then you check it with the known part of d ($d = \ldots 01$). This leaves you with five possibilities for M. Each of these five possibilities for M gives you another value for d.

You can easely check the correctness of d by verifying that $x^{ed} \pmod{n} \equiv x$ for some values of x.

For the rest of the solution we assume that you have found the correct value of d that satisfies the equation $ed \equiv 1 \pmod{\phi(n)}$.

The algorithm that follows is probabilistic, in the sense that it depends on random inputs, and is not guaranteed to succeed every time. If the procedure is applied once, then the probability of success is at least $1/2$. Therefore, if the procedure is repeated k times, then the probability of k failures is at most $(1/2)^k$. In practice, this means that we are guaranteed of finding a factor in a few trials.

The algorithm is based on the fact that the equation $x^2 \equiv 1 \pmod{n}$ has four solutions when n is a product of two distinct primes. Any solution α to $x^2 \equiv 1 \pmod{n}$ is such that $n \mid (\alpha-1)(\alpha+1)$. Since $\alpha \neq \pm 1 \pmod{n}$, we can compute a factor of n by computing either $\gcd(\alpha-1, n)$ or $\gcd(\alpha+1, n)$.

Write $ed - 1 = 2^r s$ where s is odd. Choose a random number w such that $0 < w < n$. We compute the sequence

$$z_0 = w^s \mod n, z_1 = z_0^2 \mod n, \ldots, z_i = z_{i-1}^2 \mod n, \ldots,$$

$$z_r = z_{r-1}^2 \mod n$$

Notice that $z_i \equiv w^{2^i s} \mod n$. Since $\phi(n) \mid ed - 1 = 2^r s$, Euler's theorem implies that the last term of the sequence, z_r, is 1. If the first term of the sequence is not 1, then there must be a number $z_k \neq 1 \pmod{n}$ such that $z_k^2 \equiv 1 \pmod{n}$. Our hope is that $z_k \neq 1 \pmod{n}$. Suppose a a_k exists with these properties; then we have found a nontrivial square root of unity modulo n, and we can factor n.

Numeric values for this exercise:

$$M = \begin{bmatrix} 4 & 8 & 4 & 3 \\ 3 & 10 & 5 & 10 \\ 9 & 10 & 2 & 7 \\ 0 & 8 & 6 & 4 \end{bmatrix}$$

For this exercise we can take $w = 493$ to find the factors of n.

$$n = (38002490695713045329837) \times (36352400394773071992578 57).$$

ElGamal signature scheme

1. Show that a different random number r must be selected for each message signed; otherwise, the private key a can be determined with high probability.

   Answer:
   Suppose the same random number r is chosen to sign two different messages in the ElGamal system. Suppose $s_1 = (m_1 - ay)r^{-1} \mod (p-1)$ and $s_2 = (m_2 - ay)r^{-1} \mod (p-1)$, where $y = g^r \pmod p$. Now r can be computed solving $(s_1 - s_2)r \equiv (m_1 - m_2) \mod (p-1)$. If $(s_1 - s_2) \neq 0 \mod (p-1)$, then $r \equiv (s_1 - s_2)^{-1}(m_1 - m_2) \mod (p-1)$.

   Once r is known, the secret key a can easily be found.

   Since $s_1 \equiv r^{-1}(m_1 - ay) \mod (p-1)$ and all parameters are known to the attacker, a can be calculated as follows: $a \equiv (m_1 - rs_1)y^{-1} \mod (p-1)$.

Answer to RSA problem

Prove that $M^{k\phi(n)+1} \equiv M \mod pq$ for $M = pl$ (or $M \equiv 0 \mod p$).

Proof:

$$M^{k\phi(n)+1} \equiv M \mod pq$$
$$(pl)^{k\phi(n)+1} \equiv (pl) \mod pq$$

Since l is coprime with $n = pq$, $l^{k\phi(n)+1} \equiv l \mod pq$ and the inverse of $l \mod pq$ exists, we can divide by l at both sides of the equation.

$$p^{k\phi(n)+1} \equiv p \mod pq$$

Now we split this equation in two equations, one $\mod p$ and one $\mod q$.

1. $\mod p$.

$$p^{k\phi(n)+1} \equiv p \mod p$$

This is, of course, always true.

2. $\mod q$.

$$p^{k\phi(n)+1} \equiv p \mod q$$

Replacing $\phi(n)$ with $\phi(p)\phi(q)$, we can see that this is also true.

The Chinese Remainder Theorem implies immediately that

$$p^{k\phi(n)+1} \equiv p \mod pq$$