Selected Topics in IT-Security

Frederik Armknecht
Christian Müller
Jochen Schäfer

Exercise Sheet 1

University of Mannheim
Practical Computer Science IV
FSS 2023

Release Date:	Friday, 2023-02-23, 1345h	Flags:	8
Tips & Tricks:	Friday, 2023-02-23, 1345h	XP:	22
Flag-Deadline:	—, —, —		
Solution Discussion:	Friday, 2023-03-02, 1345h		

Hints for Exercise Sheet 1

If you are experiencing problems reading an exercise on this exercise sheet, recall the block ciphers you were introduced to during the lecture. Some (okay, well, honestly, only one) exercises are encrypted with AES-256-CTR and the output is Base64-encoded. The used IV is `C0FFEEC0DE1337`.

Remember to remove any line breaks if you copy text from the PDF. Now recall Kerckhoffs' Principle: "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge." Luckily, you do not only know everything about the system, but are also given the following leaked set of $4 \times 4 = 16$ keys — including the correct one:

```
Donald Trump is not my President     Orange rocks
Star Wars Episode IV--A New Hope     It's a Trap!
Something, Something, Something,     Dark Side...
Not the key you are looking for!     MoveAlong...

The secure ASCII AES Key             IT-Security 2022
Life,... uh, finds a way             This is so wrong
Do you want to get flags             I am Pickle Rick
I had something for this             This could be it
```

To narrow down your search, recall the requirements for AES regarding the key's length. You can use any programming language/library/service/application of your choice to decrypt the exercises. However, you may need to "convert" the key to a hexadecimal representation for which you should use a standard ASCII table.

Exercise 1.1: Get your very first flag! — 1 Flag, 1 XP

If you have not joined our ILIAS course yet, please do so now, e.g., by following this short-link:

http://ilias.itsec.uni.ma

Your flag is the MD5-hash of a very well hidden "secret" in the ILIAS course's Glossary. It gives you 1 XP. During the first exercise session, you will learn how to compute hash values and how to submit them as a flag. Until then, feel free to gather information about MD5 and to experiment with it!

Exercise 1.2: Implemented Cryptography — 2 Flags, 5 XP

a)
```
Z6ToZz06OYT3NBxZ4fhJRelDB+Wq22EhwT0Lj2CSKvEVQRBUXRRkcFezq1BSTOYR
4aFm3ID4DKTczuVSjqeVy5u0ljA5sWN27E7NbTqWUOZEPfUnhrd7mlOHImnhcXTv
cPpE8JDOngfTAXOteJM+LRJ6YAJ/CLZhksJQ59dr1PSgaas2LxFZN5CsugI8cD53
KLPMZL/usplL4kBUBWTRn6rl6MA=
```

b) Suppose Alice and Bob share a collision resistant hash function H. Now they define a new hash function F as $F(x) = H(H(x_L)\|H(x_R))$. That is, the input x is first split into two halves, x_L and x_R, and then apply H as shown. The symbol "‖" denotes the concatenation of strings. Is the new function collision resistant?

c) Now assume that Alice and Bob have two functions H and F of which only one is collision resistant but they do not remember which.

i) Is the function $G(x) = H(F(x))$ collision-resistant?

ii) How about $G(x) = H(x)||F(x)$?

d) Which hash algorithm (that should not be used for cryptographic purposes anymore) is used for submitting flags in the IT-Sec Flag System? What is another hash algorithm that produces more output bits, but is also considered broken for cryptographic purposes? Combine the names and digest sizes (number of output bits) in the following format to get your flag: `HASH+nnn_HASH+nnn`

Example: GOST, 256 bits digest size; Whirlpool, 512 bits digest size ⇒ `GOST+256_Whirlpool+512`

Exercise 1.3: Brute-Force Attacking AES — 1 Flag, 2 XP

Suppose that by using low-cost specialty hardware it is possible to build a machine for $10000 (e.g. COPACOBANA, cf. https://en.wikipedia.org/wiki/COPACOBANA) which performs about 65.28 billion AES decryptions per second (or in other words, it can test as many key candidates per second). Suppose an organization wants to run an exhaustive key search for a single 128-bit AES key and is willing to spend 6.8 trillion dollars to buy these machines (which is approximately the US federal budget for 2021). How many years (with 365 days per year) would it take the organization to brute-force this single 128-bit AES key with these machines in the worst case? Ignore additional costs such as power, storage, and maintenance. Your flag is the (integer) number of years.

Exercise 1.4: Encrypting Grades Using RSA — 2 Flags, 6 XP

The following students participated in an exam last year:

Name	Student #	Name	Student #
Alice	207748	Judy	910433
Bob	645169	Mallory	402151
Craig	646212	Olivia	104367
David	502676	Peggy	983877
Eve	253231	Sybil	154354
Faythe	358425	Trudy	551345
Grace	575680	Victor	807272
Heidi	634292	Wendy	426431

Their grades have been encrypted using RSA and sent to the office for student affairs. From the university web page we know that the public key of the office is $\text{pk} = (N = 191325677, e = 5)$.

For encryption, the grades have been encoded as follows: A student is identified by their respective student number $x = x_1x_2x_3x_4x_5x_6$ and graded with the mark $y_1.y_2$. The resulting value for this student's encoding has the form

$$x_1x_2x_3x_4x_5x_6y_1y_2,$$

always consisting of eight digits, which is done for each student individually. Then, each encoded value is encrypted under the public key pk of the office for student affairs, and finally, all encoded and encrypted values are sent to the office. The possible grades are $\{1.0, 1.3, 1.7, 2.0, 2.3, 2.7, 3.0, 3.3, 3.7, 4.0, 5.0\}$.

You have intercepted the following message with ciphertexts on their way from the chair's office to the office of student affairs:

$$\{49808316,\ 128708215,\ 84193207,\ 189082259,\ 139617208,\ 3282703,\ 32071966,\ 37390843,$$
$$169592696,\ 155042072,\ 128581744,\ 123061210,\ 53728808,\ 163521634,\ 61016482,\ 5246259\}$$

We know that each number represents the encrypted result of one of the students. However, the order is not necessarily identical to the table above. In general, the RSA encryption is difficult to invert without knowledge of the secret key d. Nonetheless, here, it is quite feasible to determine the students' grades.

a) What are the grades of Alice, Bob, Eve, Peggy, and Victor? Concatenate them to get your flag. *Example:* If the grades are as follows: Alice: 1.0, Bob: 2.0, Eve: 3.0, Peggy: 4.0, Victor: 5.0
Then, your flag would be the MD5-hash of: `1.02.03.04.05.0`

b) Craig, Judy, Mallory, and Wendy want to have the best possible grade. Also, another student actually failed but does not want to take the exam again, so a 4.0 would suffice. Can you "cheat" for them before the message arrives at the office of student affairs? How would the "new" message containing the ciphertexts look like, if you replace the grades of these five students but keep the rest the same? This ciphertext (in the same form as above, including { and } but without any whitespace characters or line breaks) is your flag.

Exercise 1.5: RSA-Encrypted Money Transfer — 2 Flags, 8 XP

A company is hosting a password cracking event, where the winner will be given a fair amount of money m. m is not known by the contestants, however, they know the prize money will be handed out on a USB thumb drive. The winner then has to bring this drive to the bank in order to get the money. Since only employees can participate in this event, the company knows which bank each employee is using and holds for each bank a device which creates one-time-passwords synchronized with the bank in some secure way. This one-time-password changes each time by an unknown rule and pseudorandomly produces a password p. The company wants to make sure that the data on the USB thumb drive is encrypted in a secure manner. Their security expert chose the asymmetric scheme RSA to perform this task and exchanged according data with the banks. Furthermore, he 'blinds' the amount of money m with the password p by multiplying them, i.e. $b = m \cdot p$. Then, nobody can see the actual amount of money which is stored on the USB drive, since the public key $pk = (N, e)$ of each bank is known and the encrypted value

$$c = (m \cdot p)^e \bmod N \tag{1}$$

cannot be tested against possible values for the amount of money. For example, if the winner thinks he gets $m = 50$ money, he could normally simply test if $c = 50^e \bmod N$. But, due to the one-time-password p, this is not possible and the employee cannot determine the amount before handing the USB drive over to the bank.

John is clever and was the first to find the secret password. He was awarded with the USB drive formed as a trophy. John's bank has the following public key: $pk = (N = 64741063, e = 5)$.

a) John, however, did not only know about cracking passwords, he also knew he wanted to have more money than the company put on the USB drive as a reward. When he examined the USB drive, it contained the ciphertext c, that is

$$c = 32813107 \equiv (m \cdot p)^5 \bmod 64741063, \tag{2}$$

which is the RSA encryption of $m \cdot p$ under the public key (N, e) of his bank. After some minutes of typing, he changed the ciphertext on the USB drive to a different value. When he brought the USB drive to his bank, he claimed his reward and was very satisfied: he got twice the amount of money the company wanted to originally give out as a reward. What was the ciphertext value he put on the USB drive?

b) The company heard about this weakness and realized it was their own fault. They immediately got rid of the synchronized one-time-passwords and switched to a Message Authentication Code (MAC) instead, which not only made them independent of the OTP service of the bank but also yields integrity of the data. "John's trick won't work anymore – if someone tries to change the amount of money like before, it will be detected.", the company's security expert stated. The MAC in use is defined as

$$\mathrm{MAC}(c) = o(\mathrm{SHA512}(c)) \bmod 100.000, \tag{3}$$

where $o(x)$ removes all non-digit (i.e. non-decimals) characters of x, then, this result is reduced modulo 100.000. This MAC is then applied to an RSA ciphertext c as follows:

$$\hat{c} := \mathrm{MAC}(c) \parallel c \tag{4}$$

That is, the digits of both MAC and ciphertext are simply concatenated and form the new ciphertext $\hat{c}$. John heard about this, went to his bank the same evening, and had withdrawn 50.000 money which he got 'spontaneously' from the company on a USB drive. What ciphertext did John put on this USB drive in order to subvert the company's new security mechanism?