Cryptography and Architectures for Computer Security

Exam Code: 095947 (old 090959), A.Y. 2015–2016, Semester: 2

Prof. G. Pelosi

July 4th, 2016 – Exam Session

Name: ..... Surname: .....

Student ID: ..... Signature: .....

Time: 2h:30'. Use of textbooks, notes, phones or Internet connected devices is not allowed. Prior to turn in your paper, write your name on any additional sheet and sign it.

Question 1 [4 pts]
A5/1 is a stream cipher with a 64-bit key employed to secure GSM communications, which is still in use whenever the link between the mobile phone and the cell is downgraded to GSM. Performing an exhaustive key search to find a key is thus within practical feasibility, although taking quite some computation. To this end, Time-to-Memory Tradeoffs (TMTO) are an appealing approach, and have effectively been used to crack A5/1 encrypted messages in practice.

(a) Assuming to employ a nVidia Geforce GTX 1070, able to compute around 29×2^{30} A5/1 keystreams per second, each one 64-bit long, calculate how much memory is required to break A5/1 in less than 0.1s in the following two cases:

i) GDDR5 DRAM, access time 0.5μs,

ii) SATA-SSD 10μs.

Assume, for the sake of simplicity, that the data structure in memory allows you to check for the presence of a given value with a single access, and that you are given an A5/1 plaintext-ciphertext pair, where both are 64-bit long to the end of finding the key.

Argue on which one of the two strategies may be practically viable.

(b) Which one(s) of the following compensation for the weakness of A5/1 would render unfeasible a TMTO based strategy? Does it prevent an exhaustive search for a single key too?

1. Employ two different instances of A5/1, one running on the actual key, and one on a 32-bit random salt (supposed to be sent at the beginning of the communication in clear), padded with zeros up to 64-bit. Encrypt the plaintext with the first engine, and then with the second one.
2. Add a 192b random *salt* (supposed to be sent at the beginning of the communication in clear) to the key, splitting it into three 64-bit words and combining them via `xor` with the key, employing the result as the A5/1 key.

Discuss the resistance against a bruteforce attack of each of these strategies.

Question 2 [4 pts]
While evaluating block cipher designs, you encounter an SPN one, where you are free to tweak one out of the three following features: the S-Box, the linear diffusion layer or the number of rounds. The basic version of the block cipher employs 16 8×8 bit identical S-boxes for the substitution layer, and a single linear diffusion layer per round. The entire cipher is r round long, so to provide a 2^{128} linear and differential cryptanalysis immunity, as the block size of the cipher is 128b.

(a) Compute r so that the desired immunity is guaranteed, assuming that the best differential probability for the S-box is $\frac{1}{8}$, the best linear bias is $\frac{1}{4}$ and the diffusion layer guarantees that a single bit in input to it will influence at least 4 different S-boxes in the next round. Perform the computations providing a reasonable, but conservative approximation.

(b) Assume to be measuring the latency of the cipher in round-sized units, and its area if implemented in hardware, following the same criterion. As a consequence, the cipher you just analyzed has a latency of r_t, and an area of r_a. Do the following tweaks turns out to be more efficient than the original cipher, i.e., have a lower area-time product?

- Strengthen the S-box: $\frac{1}{32}$ maximum differential probability, $\frac{1}{16}$ maximum linear bias, +50% on the area of a single round, with no latency penalty.
- Perfect diffusion layer: a single bit change in the diffusion layer will change the input of all the following S-Boxes. +25% on the area of a single round, +5% on the latency.

Question 3 [8 pts]
Consider the cyclic group $G=(\mathbb{Z}_{41}^*, \cdot)$.

(a) List the subgroups of G (both proper and trivial) showing for each of them the order and one generator.

(b) Given the following discrete logarithms, x_0, x_1, state if each one of them exists, motivating your answer and compute them.

$$x_0 \equiv \log_2^{\mathtt{D}}((-1)^{40^3-1})$$
$$x_1 \equiv \log_4^{\mathtt{D}}(2)$$

(d) Describe how to properly choose the parameters of the (Ephemeral) Diffie-Hellman protocol and point out the advantages of using it.

Question 4 [4 pts]
Consider the finite field $\mathbb{F}_{3^4}$

(a) Write the number of irreducible and primitive polynomials

(b) Verify that $f_1(x)=x^4+x-1 \in \mathbb{F}_3[x]$ is primitive and show each of its roots in $\mathbb{F}_{3^4}$.

Question 5 [12 pts]

(a) Apply the Pollard's ρ method to factorize the RSA modulus $n = p \cdot q = 1537$.
(As a backup alternative, apply a "trivial division" strategy).

(b) Choose an admissible public exponent e between the values $e{=}9_{\texttt{dec}}$ and $e{=}13_{\texttt{dec}}$ and compute the value of the corresponding RSA private key $k_{priv}{=}(p, q, \varphi(n), d)$. Show every step of the computation.

(c) Sign the message $m{=}333_{\texttt{dec}}{\in}\mathbb{Z}_n$ (without employing any padding scheme) through applying the CRT. Describe each step of the procedure.

(d) A system administrator wants to set up the RSA cryptosystem in a network including U users, and for doing so he decides to generate a pool of P prime numbers each 2048 bits long and keeping $1{<}P{<}\frac{U^2}{2}$. Subsequently, he draws pairs of numbers from such a pool to set up the public moduli. The public exponent is assumed to be the same for every user and equal to $e{=}2^{16}+1$, while the secret parameters are computed knowing the factorization of each modulus.

Discuss whether the aforementioned procedure is secure or not, justifying your answer.

(e) Assume to work into the Montgomery domain: $(\widetilde{\mathbb{Z}}_N, +, \times)$, $N{=}22$

- Show the definition of the Montgomery Multiplication and explain the advantages of applying it for implementing the RSA cryptosystem.
- Compute the Montgomery multiplication $C{=}A{\times}B \bmod N$, where $A{=}19_{\text{dec}}$ and $B{=}16_{\text{dec}}$ are values in the Montgomery domain, assuming a binary encoding of the operands