Cryptography and Architectures for Computer Security

Exam Code: 095947 (old 090959), A.Y. 2014–2015, Semester: 2

Prof. G. Pelosi

July 22nd, 2015 – 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 [3 pts]
Consider a Vigenère Cipher. Each (lower-case) letter of the English alphabet is put into one-to-one correspondence with the integer denoting its position, i.e.: a=0, b=1, ..., z=25. The key $k=(k_1, k_2, \ldots, k_m)$, $0 \leq k_i \leq 25$ is composed by $m \geq 2$ letters and is employed to perform a block encryption of the plaintext message $x=(x_1, x_2, \ldots, x_m, x_{m+1}, \ldots)$, with $0 \leq x_i \leq 25$, as follows:

$$c=(x_1+k_1 \bmod 26, \ldots, x_m+k_m \bmod 26, x_{m+1}+k_1 \bmod 26, \ldots)$$

- Show how to apply a *Known Plaintext Attack*
- Describe how to execute a *Ciphertext-Only Attack*
- How is it possible to design a *Perfectly Secure Cipher* employing a Vigenère Cipher?

Question 2 [3 pts]
Let p be a large prime and g be a generator of $(\mathbb{Z}_p^*, \cdot)$. Suppose we are considering the function $h : \mathbb{Z} \mapsto \mathbb{Z}_p^*$ for use as a hash function, where $h(x) = g^x \bmod p$, assuming x to be an arbitrary binary string interpreted as a positive integer number, i.e.: $x \in \mathbb{Z}$, $x \geq 0$.
Note that the *compression property* of typical hash functions is satisfied by the above definition as messages x of arbitrary bit-length are hashed into a fixed size digest.

Assuming the computational difficulty of the discrete logarithm problem in $(\mathbb{Z}_p^*, \cdot)$, which properties of cryptographic hash functions does h satisfy? Discuss if h is a sound cryptographic hash function or not.

Question 3 [6 pts]
Threefish is a block cipher with a 256-bit (32 bytes) sized block and allows the use of three different key lengths: 256, 512 and 1024 bits. Consider the three following password hashing strategies relying on Threefish:

(a) Split the password p in 9-byte wide blocks $p_0 \dots p_n$, and generate $n{+}1$ 23-byte unpredictable random salt blocks s_i, $0{\leq}i{\leq}n$. Store on the disk a sequence of blocks $h_i{=}\textsc{Threefish}_{p_i||s_i}(p_i||s_i)$ followed by the concatenation of all the s_i, i.e.: $h_0||h_1||\dots||h_n||s_0||s_1||\dots||s_n$.

(b) Split the password p in 9 bytes wide blocks $p_0 \dots p_n$, and generate $n{+}1$ 23-byte salt blocks s_i obtained as the concatenation of a 1-byte random value for each s_i. Store on the disk a sequence of blocks $h_i{=}\textsc{Threefish}_{p_i||s_i}(p_i||s_i)$ followed by the concatenation of all the s_i.

(c) Split the password p in 32-byte wide blocks $p_0 \dots p_n$, obtain a 1024-bit unpredictable random salt s and store on the disk a sequence of blocks h_i, with $h_0{=}s$ and $\forall i > 0, h_i{=}\textsc{Threefish}_s(p_{i-1} \oplus h_{i-1})$

Assume you have 16 TiB of disk space available (average access time 1ms), 32 GiB of RAM (average access time 0.1μs) and you are able to compute 2^{32} Threefish encryptions or decryptions per second on a good GPU.

State how much computational effort (i.e., how much time) is required to break the proposed password hashing schemes with the best possible technique, and whether this is practically feasible considering only fully lowercase passwords.

Question 4 [6 pts]
Show if the irreducible polynomial $m(x){=}x^8{+}x^4{+}x^3{+}x{+}1{\in}\mathbb{F}_2[x]$ specified in the Advanced Encryption Standard (AES) to represent the elements of the finite field $\mathbb{F}_{2^8}$ and employed in the S-box construction, is also a primitive polynomial.

Question 5 [7 pts]
Consider the cyclic group $(\mathbb{Z}^*_{53}, \cdot)$ with one of its generators being $g{=}2$.

(a) Show the order of the group and exhibit at least one generator for each of its subgroups.

(b) Consider the following two discrete logarithms, x_0, x_1, and state if each of them exists, motivating your answer:

$$x_0 = \log_g^{\mathtt{D}}(101^{101})$$
$$x_1 = \log_{g^{20}}^{\mathtt{D}}(2)$$

(c) Compute the logarithms which you stated to be existing.

(d) Consider a generic cyclic group of the form $\mathrm{G}{=}(\mathbb{Z}^*_{p^k}, \cdot)$, with $p{\geq}2$ a prime integer, and $k{\geq}1$. What is an efficient and secure choice for p and k to properly set up the public parameters of a Diffie-Hellman protocol?

Question 6 [3 pts]
Consider a textbook RSA scheme, with $n{=}pq$ being a product of two different primes and the relation between the public exponent $e{\in}\mathbb{Z}^*_{\varphi(n)}$ and the private exponent $d{\in}\mathbb{Z}^*_{\varphi(n)}$ such that $ed \bmod \varphi(n){=}1$.

(a) Show the correctness of the identity

$$m^{ed} \equiv_n m \qquad \text{with} \quad m \in \mathbb{Z}_n$$

which states that we obtain the same message after encryption and decryption (or viceversa)

(b) Consider an RSA public modulus computed as the square of a prime number, i.e.: $p{=}q$ and $n{=}p^2$. Show a simple example for the fact that, in this case, the condition $ed \bmod \varphi(n){=}1$ does not imply $m^{ed} \equiv_n m,\ \forall\ m{\in}\mathbb{Z}_n$.

Question 7 [6 pts]
Many cryptosystems such as RSA and Diffie–Hellman key exchange are based on arithmetic operations modulo a large integer number.

(a) Explain why the Montgomery multiplication method is the preferred choice for implementing the aforementioned cryptographic schemes.

(b) Describe the Montgomery multiplication technique, specifying the notions of Montgomery transformation, Montgomery reduction and the basic idea for applying it to multi-precision integer representations.

(c) Assume to work into the Montgomery domain: $(\mathbb{Z}_N, +, \times)$, $N = 21$.
Compute the Montgomery multiplication $C{=}A{\times}B \bmod N$, where $A{=}16_{\text{decimal}}$ and $B{=}18_{\text{decimal}}$ are binary encoded values in the Montgomery domain. Show every step of the procedure.