Cryptography and Architectures for Computer Security

Exam Code: 095947 (old 090959), A.Y. 2017–2018, Semester: 2

Prof. G. Pelosi

June 21st, 2018 – Exam Session

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

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

Time: 2h:30'. Use of textbooks, notes, or Internet connected devices (including smartphones) is not allowed. The usage of simple calculators is allowed. Prior to turn in your paper, write your name on any additional sheet and sign it.

Question 1 [4 pts]
The CBC mode of operation requires an Initialization Vector, `IV`, as the initial ciphertext block to kick things off, and the `IV` must be **i)** Unique (i.e., must not be repeated for any message encrypted with a given key); and **ii)** Unpredictable (i.e., an attacker who observes any number of messages encrypted with a given key and their `IV`s should have no information to predict the next one with probability of success greater than 50% per bit – indistinguishable from random).

(a) explain why uniqueness property is necessary;

(b) explain why unpredictability property is necessary.

Solution:
(a) Uniqueness of `IV`s is necessary because if two messages employ the same `IV` value and the first CBC-encrypted blocks of the two are equal this implies that also the corresponding plaintext blocks are equal. Just like in a ECB mode of operation, it is possible to correctly guess if the first $x \geq 1$ blocks of any pair of ciphertexts contain the same data (... the plaintext indistinguishability property in a ciphertext-only scenario is not provided).
(b) Unpredictability refers to the plaintext indistinguishability guarantees of the cipher in a chosen-plaintext scenario.
Given two plaintext blocks $\overline{m}$, $\widetilde{m}$ and their CBC encrypted ciphertexts $c = \langle \texttt{IV}, c_1 \rangle$, $c' = \langle \texttt{IV}', c'_1 \rangle$, an adversary should have 50% probability to correctly guess the correspondence of c and c' to $\overline{m}$, $\widetilde{m}$.

If the sequence of the `IV`s is predictable, then the adversary can pick one of the plaintext messages (e.g., $\overline{m}$) and submit to the oracle the request to encrypt the following block: $\texttt{IV} \oplus \overline{m} \oplus \texttt{IV}_{\text{next}}$, where $\texttt{IV}_{\text{next}}$ is the value predicted to be employed by the oracle to process this new block.
The ciphertext block returned to the adversary will be $\text{Enc}_k\left(\texttt{IV}_{\text{next}} \oplus (\texttt{IV} \oplus m \oplus \texttt{IV}_{\text{next}})\right) = \text{Enc}_k\left(\texttt{IV} \oplus \overline{m}\right)$. Comparing the value of the new ciphertext block with c_1, the adversary can easily guess the correct correspondence between the originally given plaintexts and ciphertexts, without any uncertainty.

When the `IV` values are predictable, a similar approach can be employed to guess the plaintext value of any block whose encryption has been already observed.

Question 2 [5 pts]
Consider the following 4×4 bit S-Box.

in	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
out	0001	0001	0111	0111	0110	0110	0000	0000	1100	1111	1110	1001	1000	1011	1101	1010

(a) Let $\langle b_3, b_2, b_1, b_0 \rangle$ be the output bits of the S-Box above corresponding to the input $\langle a_3, a_2, a_1, a_0 \rangle$. Compute the linear bias for the expression $a_2 \oplus a_0 = b_3 \oplus b_0$.

(b) Compute the probability of the differential $(\Delta a, \Delta b) = (0001, 0011)$ and of the differential $(\Delta a, \Delta b) = (0001, 0000)$. Which one is best employed by an attacker (without considering questions on the number of involved bits)?

Consider a block cipher with a 128-bit key and a *Substitution-Permutation Network* (SPN) design. The cipher has a 128-bit wide block. The *key addition layer* is performed via bitwise `xor`. The *substitution layer* of the cipher is constituted by 32 equal S-Boxes defined as shown above, and acting on 4 bits of the state each. The *linear permutation layer* is built in such a fashion that causes a single bit change in the output of an S-Box to propagate to four distinct S-Boxes in the next round.

Therefore, it can be assumed that a single bit change in the plaintext will involve one S-Box in the first round, 4 in the second, 16 in the third, and 32 from the fourth round onwards.

(c) Compute, employing the aforementioned assumptions and the linear bias obtained in (a), the minimum number of rounds for the cipher to be immune to linear cryptanalysis.

(d) Compute, employing the aforementioned assumptions and the highest probability obtained in (b), the minimum number of rounds for the cipher to be immune to differential cryptanalysis.

Solution:

(a) Computing how many times the aforementioned expression holds true on all the possible S-Box input/output pairs yields that it is true 6 times out of 16, in turn giving a bias equal to $\frac{6}{16} - \frac{1}{2} = -\frac{1}{8}$.

(b) Computing how many times flipping the a_0 bit of the S-Box input causes a bit flip in only the b_1, b_0 bit results in 4 times out of 16. Computing how many times flipping the a_0 bit in the S-Box input generates no changes in the result yields 8 times out of 16. The second differential is thus preferable to the first.

(c) Considering the aforementioned bias of $\epsilon = \pm\frac{1}{8}$, the attacker is able to obtain a linear relation with bias
$\epsilon_{\texttt{1st-round}} = \pm\frac{1}{8}$ after the 1st round,
$\epsilon_{\texttt{2nd-round}} = \pm 2^{5-1} \cdot (\frac{1}{8})^4 \cdot \epsilon_{\texttt{1st-round}} = \pm 2^{-11}$ after the 2nd round (...pile-up of 5 relations),
$\epsilon_{\texttt{3rd-round}} = \pm 2^{17-1} \cdot (\frac{1}{8})^{16} \cdot \epsilon_{\texttt{2nd-round}} = \pm 2^{-43}$ after the 3rd round (...pile-up of 17 relations),
$\epsilon_{\texttt{4th-round}} = \pm 2^{33-1} \cdot (\frac{1}{8})^{32} \cdot \epsilon_{\texttt{3rd-round}} = \pm 2^{-107}$ after the 4th round (...pile-up of 33 relations),
From the 5th round onwards, the bias will be decreased by a further $2^{33-1} \cdot (\frac{1}{8})^{32} = 2^{-64}$ per round (... the number of active S-boxes is always 32).
$\epsilon_{\texttt{5th-round}} = \pm 2^{171}$. It is thus sufficient to have a **six rounds** cipher to provide immunity from linear cryptanalysis, which targets the last-but-one subkey of the cipher as the computational effort needed is higher than a bruteforce over the keyspace.

(d) Considering the aforementioned probability of $\frac{1}{2}$, we have that an attacker employing differential cryptanalysis will have a differential taking place with a probability of $\frac{1}{2} \cdot (\frac{1}{2})^4 \cdot (\frac{1}{2})^{16} \cdot (\frac{1}{2})^{32} = 2^{-53}$ after the 4th round. Each additional round lowers the probability of the differential by $(\frac{1}{2})^{32}$. We thus need an **eight round cipher**, so that the attacker, willing to employ an seven-round differential to recover the last-but-one subkey, will be faced with a probability of the differential holding equal to 2^{-149}.

Question 3 [4 pts]
Consider the SSH protocol.

(a) It is often advised to avoid password-based authentication whenever possible, replacing it by per-user public key authentication. Such an advice comes from an advantage in case of a disclosure of the private key of the SSH server. Justify whether or not this is a correct advice.

(b) Consider a user, willing to generate a large number X of RSA user-authentication keypairs for SSH. In order to save time, the user generates only $\sqrt{X}$ distinct prime numbers and obtains the public moduli of the keypairs multiplying pairs of the said primes. Argue in favor or against the said procedure, and describe if the administrators of two distinct SSH servers where the user is authenticating with two separate keypairs may abuse of the said procedure.

Solution:

(a) The advice is correct. In case a disclosure of the SSH server private key is made, it is possible for an eavesdropper, which obtain the transcript of login sessions on the said SSH server, to successfully decrypt the *user authentication* password, as he possesses the SSH server's private key (... see in the last pages of this document a longer answer with a summary of the SSH protocol).

(b) The key generation procedure is insecure. In particular, the administrators of two servers where the user is logging into will know a public key each. If the moduli n', n'' of such public keys $\langle n', e' \rangle$, $\langle n'', e'' \rangle$ happen to share a prime factor (i.e., $n' = pq, n'' = pr$ ... this is verifiable in polynomial time by computing a gcd) the administrators are able to derive both user's private keys, and thus impersonate him wherever he is using the same keypairs.
Clearly, the described key generation procedures can be effectively employed to generate only $\frac{X}{2} - \sqrt{X}$ pairs of distinct public moduli.

Question 4 [9 pts]

(a) Consider the cyclic group $(\mathbb{Z}^*_{625}, \cdot)$. Compute the cardinality of the group, the number of its generators and the number of subgroups.

(b) Consider the finite field $\mathbb{F}_{5^4}$. Compute the cardinality of the multiplicative group. Find the number of irreducible and primitive polynomials. Determine if $f(x) = x^5 + 4x - 1 \in \mathbb{F}_5[x]$ is reducible or not.

(c) State the conditions to properly configure the public parameters of a Diffie-Hellman protocol justifying your answer. Which cyclic group between $(\mathbb{Z}^*_{625}, \cdot)$ and $(\mathbb{F}^*_{5^4}, \cdot)$ would be preferable?

Solution:

(a) $|(\mathbb{Z}^*_{625}, \cdot)| = \varphi(625) = 5^4 - 5^3 = 500$;
Num. of generators: $\varphi(\varphi(625)) = (2^2 - 2) \cdot (5^3 - 5^2) = 2 \cdot 100 = 200$.
Num. of proper divisors of $\varphi(625) = 500$ is equal to 10 (i.e., 2, 4, 5, 10, 20, 25, 50, 100, 125, 250); this value coincides with the num. of proper subgroups of $(\mathbb{Z}^*_{625}, \cdot)$

(b) $|\mathbb{F}_{5^4}| = 5^4 - 1 = 624$.
$4 \cdot N_4(5) + 2 \cdot N_2(5) + 1 \cdot N_2(5) = 5^4$; $N_1(5) = 5$; $N_2(5) = \frac{5^2-5}{2} = 10$;
$\Rightarrow$ Number of irreducible polynomials: $N_4(5) = 150$.
Number of primitive polynomials: $M_4(5) = \frac{\varphi(624)}{4} = \frac{2^4 \cdot 3 \cdot 13}{4} = \frac{8 \cdot 2 \cdot 12}{4} = 48$.
Assuming $f(x) = x^5 + 4x - 1 \in \mathbb{F}_5[x]$ as irreducible, it should be possible to build a representation of any element of the field $\mathbb{F}_{5^4}$ as $\theta_4\alpha^4 + \theta_3\alpha^3 + \theta_2\alpha^2 + \theta_1\alpha + \theta_0$, where $\theta_i \in \mathbb{F}_5$, $\alpha \in \mathbb{F}_{5^4} \setminus \mathbb{F}_5$, and $f(\alpha) = 0$.
Being α a root of $f(x)$ it follows that $\alpha^5 \equiv \alpha + 1$ and if $f(x)$ is irreducible then $\alpha^{5^4} \equiv \alpha$.
$\alpha^{5^4} \equiv (((\alpha^5)^5)^5)^5 \equiv (((\alpha+1)^5)^5)^5 \equiv ((\alpha+2)^5)^5 \equiv (\alpha^5+2)^5 \equiv (\alpha+3)^5 \equiv \alpha^5+1 \equiv \alpha + 2 \not\equiv \alpha \Rightarrow f(x)$ cannot be used to build the representation of the elements of $\mathbb{F}_{5^4} \Rightarrow f(x)$ is reducible.

(c) (see lectures)...the cyclic group must have a prime cardinality (properly sized) to prevent the application of the Poligh-Hellman attack...this has the additional advantage that each element of the group will also be a generator (with the exception of the neutral element)... the factorization of the order of $(\mathbb{Z}^*_{625}, \cdot)$ is 6-smooth, while the factorization of the order of $(\mathbb{F}^*_{5^4}, \cdot)$ is 13-smooth. Thus, keeping into account the computational complexity of the Poligh-Hellman attack, the latter group is preferable w.r.t. former.

Question 5 [14 pts]

(a) Apply the Pollard's ρ method to factorize the RSA modulus $n = p \cdot q = 667_{\text{dec}}$.
Assume $f(x) = x^2 + 1 \bmod n$ as the "random-walking" function. Show every step of the computation. (As a backup alternative, apply a "trivial division" strategy).

(b) Describe the Miller-Rabin and the Fermat primality tests, pointing out the reasons to prefer the former.

(c) Choose an admissible private exponent d between the values $d{=}31_{\texttt{dec}}$ and $d{=}77_{\texttt{dec}}$ and show the value of the corresponding RSA public and private keys $k_{\text{pub}}{=}(e, n)$, $k_{\text{priv}}{=}(p, q, \varphi(n), d)$.

(d) Sign the message $m{=}500_{\texttt{dec}} \in \mathbb{Z}_n$ (without employing any padding scheme) through applying the CRT. Describe each step of the procedure.

(e) Consider a schoolbook RSA cryptosystem employing the Montgomery multiplication as building block. Write down the pseudo code of both the encryption and the CRT-based decryption transformations, pointing out their asymptotic computational complexities.

(f) Consider an RSA modulus $N = 3_{\text{dec}} \cdot 7_{\text{dec}} = 21_{\text{dec}}$. Compute the Montgomery multiplication between two operands $A = 18_{\text{dec}}$, $B = 17_{\text{dec}}$ that are already in the Montgomery domain $(\widetilde{\mathbb{Z}}_N, +, \times)$. Show every step of the computation, assuming a binary encoding of the operands and the smallest possible value for the Montgomery radix R.

Solution:

(a) $p = 23$, $q = 29, n = p \cdot q = 667$.

(b) see lectures ...

(c) $\varphi(n) = 616 = 2^3 \cdot 7 \cdot 11$, $d = 31_{\text{dec}} \in \mathbb{Z}^*_{\varphi(n)}$, $e \equiv_{\varphi(n)} d^{-1} = 159$, ...;

(d) $m_p \equiv_{23} 500^{31 \bmod 22} \equiv_{23} 17^9 \equiv_{23} 7$,
$m_q \equiv_{29} 500^{31 \bmod 28} \equiv_{29} 7^3 \equiv_{29} 24$, ...,
$s = 53$

(e) see lectures ...

(f) $N = 21$, $R = 32$, $RR' - NN' = 1$, $R' = R^{-1} \bmod N = 2$, $N' = N^{-1} \bmod R = 3$, ..., $C = \texttt{MMul}(A, B) = A \cdot B \cdot R^{-1} \bmod N = 18 \cdot 17 \cdot 2 \bmod 21 = 3$, ... see lectures for the execution trace of the Montgomery multiplication with binary encoded operands.

Long answer to question 3(a), with a summary of the SSH protocol.

The SSH protocol employs a client-server model to authenticate two parties and encrypt the data between them.

The server component listens on a designated port for connections. It is responsible for

(s1) negotiating the secure connection,

(s2) authenticating the connecting party,

(s3) and spawning the correct environment if the credentials are accepted.

The client is responsible for

(c1) beginning the initial TCP handshake with the server,

(c2) negotiating the secure connection, verifying that the server's identity matches previously recorded information,

(c3) and providing credentials to authenticate.

An SSH session is established in two separate stages.
The first is to agree upon and establish encryption to protect future communication. The second stage is to authenticate the user and discover whether access to the server should be granted.

Begin of the first stage

Client −− (c1) −− > Server.
The client sends a connection request to the server.

Client < −− (s1) −− Server.
The server responds by transmitting the protocol versions he supports and its host public key.

Client −− (c2) −− > Server.
The client confirms if he can match one of the protocols accepted by the server and in the affirmative case he checks if the server host public key is authentic. The local key storage of the client is a simple text file with one IP address-public key pair per line (`/.ssh/known_hosts`). The first time a connection to the remote host is done, the client asks the user whether the received public key is trusted to be one of the public keys of the server. In the affirmative case, the host public key of the server is added to the local file `/.ssh/known_hosts`, and considered to be authentic for all the future connections. Trust revocation is simply performed via removing the offending key from the storage file.

Client −− (c2.1) −− > Server;
Client < −− (s1.1) −− Server.
Among the information concerning the agreed protocol version, client and server agreed also on the parameters of a cyclic group $G \subseteq \mathbb{Z}_p^*$ with prime order q and generator $g \in G$, where $p>>2$ is also a prime ($q \mid p-1$, $q>>2$), and on a symmetric-key cipher (e.g., AES-128). The client starts the execution of a Diffie-Hellman protocol generating a DH private/public key pair, and sends the ephemeral DH public key to the server. The server also generates a private-public DH keypair, computes a shared secret combining the information received from the client and signs a digest obtained from the concatenation of its DH-public key, the client DH-public key, the shared value and some identification strings. The server sends back both its DH-public key and a copy of the

said signature. The client computes the shared secret, and checks if the received signature is correct employing the long term public key of the server.

After this exchange of informations both parties have a shared secret employed as an ephemeral session key of the previously agreed cipher (e.g., AES-128). The shared session key is then used to encrypt all communication that follows, with the purpose to build an encrypted tunnel that cannot be deciphered by anyone except the legitimate client and server.

End of the first stage

Begin of the second stage In this stage the user must authenticate himself to the server, while the server must decide upon the requested access. There are a few different methods that can be used for authentication, based on what the server accepts.

Client −− (c3) −− > Server. (case 1)
If the server is configured to authenticate user's connection requests with a "password-based mechanism", the client sends an authentication request transmitting its username to the server, while the server prompts back the client for the password of the account he is attempting to login with.

Even though the password will be encrypted (thanks to the DH tunnel), this method is not generally recommended due to the limitations on the complexity of the password. Automated scripts can break passwords of normal lengths very easily compared to other authentication methods. Therefore, any client can try to guess the password of another user. Even worse, if the long term private key of the server is disclosed, a man-in-the-middle during the Diffie-Hellman key agreement phase can easily masquerade as the server and can induce the client to establish a session key with him. Subsequently, even if the user's password is encrypted with the session key, its value will be disclosed to the adversary.

Client −− (c3) −− > Server. (case 2)
If the server is configured to authenticate user's connection with a per-user "public key mechanism", the client sends a message including the ID of the public-key/private-key pair it would like to authenticate with, along with a cryptographic signature of the same message. The server checks the "`./ssh/authorized_keys`" file in the account that the client is attempting to log into for the keypair ID. If a public-key with matching ID is found in the file, the server checks the validity of the signature to verify if the client actually knows the private key corresponding to the public one stored by it.

In case the server's private key is disclosed, the adversary can perform a men-in-the-middle attack during the DH key agreement phase and masquerade as the legitimate server. However, even if he knows the public key of the client, he cannot learn any secret information about it (there is no password sent in clear).

End of the second stage

Client < −− (s3) −− Server.
In case the credentials transmitted by the client are accepted, the server spawn the correct environment as per user's request.