*Exam of 26th January 2018, a.y. 2017-18. Tim e: 2 hours*

**Assertion**

False ECB is insecure for encrypting one single block of plaintext

True ECB is parallelizable

False CBC-encryption is parallelizable

True CBC-decryption is parallelizable

False In CBC decryption: a bit flip in the ciphertext corrupts only the current block

False Ciphertext stealing is a technique for reducing the size of the ciphertext by a constant factor

True CFB makes a block cipher into a self-synchronizing stream cipher

True OFB: knowledge of the.initialization vector is not sufficient for breaking its security -

True OFB: can preprocessing speed-up the encryption/decryption process?

False CTR: reusing the initialization vector does not introduce a vulnerability

**02: Odd/even game**

Alice and Bob want to play the odd/even game b y exchanging messages on the net. In the classic

odd/even game the players choose two non-negative integers ZA and ZB, after having betted on the

parity (even or odd) of Z = Z A + Z B; at time of betting the players have not yet chosen their

numbers. The players play in the net by the following protocol. In what follows h(.) is a   
cryptographic hashing function, and II denotes concatenation.

A -> B: (p, h(ZA ll nA)) [Alice chooses parity pϵ{even,odd}, Z A , nonce nA, and sends info to Bob]

B -> A: ZB [Bob chooses ZB and sends it to Alice; now Alice can compute ZA + ZB

A -> B: (ZA , nA) [Alice reveals her data, then Bob can check hash and compute ZA+ZB too]

02.1 [4/30] Show that Alice can cheat so that she can manage to win all the games.

02.2 [4/30] Show how to fix the protocol (by adding/ changing messages) so that it is made

more secure wrt possible Alicemisbehaviors. (Do not introduce 3rd parties)

**Q3: Authentication**

Q3. 1 [3/30] Describe a scheme of authentication based on Needham-Schroeder that makes use of a trusted third party. Discuss the type of authentication (one/two way) and its robustness against replay attacks.

Q3.2 [4/30] Inspired to the scheme above, design a scheme of mutual authentication between three parties, that makes use of a trusted fourth party.

Risposta

Q3.1

It is the basis for the Kerberos protocol and aims to establish a session key between two parties on a network, typically to protect further communication.

1. A chooses N (nonce) and sends to C: A, B, N

2. C chooses K and sends to A: KAC(N, K, B, KBC(K, A))

3. A decodes, checks N and B, and sends to B: KBC(K, A)

4. B decodes, chooses nonce N’ and sends to A: K(this is B, N’)

5. A sends to B K(this is A, N’ -1) now B has checked A

Per garantire l’integrità e non far avvenire attacchi di replay:

How to guarantee data integrity

• timestamps (a message is valid only in a small time window)

• sequence number (A and B remember sequence number of exchanged messages to avoid replay attacks in which the attacker sends old messages)

• nonce should be used carefully

Copiare lo schema Expanded Needham-Schroeder è la versione con i ticket (timestamps).



Q3.2

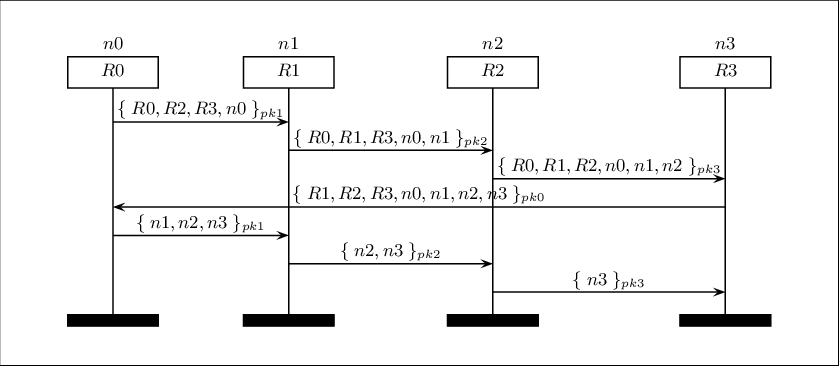
1. A chooses N (nonce) and sends to C: A, B, D, N

2. C chooses K and sends to A: KAC(N, K, B, D, KBC(K, A), KBD(K, A))

3. A decodes, checks N ,B and D, and sends to B: KBC(K, A) and sends to D: KDC(K, A)

4. B decodes, chooses nonce N’ and sends to A: K(this is B, N’) D decodes, chooses nonce N’ and sends to A: K(this is D, N’)

5. A sends to B K(this is A, N’ -1) now B has checked A . A sends to D K(this is A, N’ -1) now D has checked A



Q4: **lptables**

Describe at your best each the following iptables commands, *clarifying whether they are meant to*

*protect* a *network or* a *single host* (explain why). In what follows eth0 is a network interface

exposed to the extern, eth1 is a network interface to the LAN.

Q4.1 [2/30] iptables -A FORWARD -p udp -i eth0 -o eth1 --dport 53 --sport

1024:65535 -j ACCEPT

Q4.2 [2/30] iptables -A INPUT -p tcp -i eth1 --dport 22 --sport 1024:65535 -m state

--state NEW -j ACCEPT

Q4.3 [2/30] iptables -A OUTPUT -p tcp -i eth0 --dport 22 --sport 1024:65535 -m

state --state NEW-j ACCEPT

**Q4.1**

Allow DNS access from/to firewall

**-iptables -A CATENA ...**- Aggiunge una regola alla fine della catena indicata

-ORIGIN OF CONNECTION/PACK. = FORWARD (F)

-PROTOCOL (PROT) =udp

-NETWORK INTERFACE (IFACE) = eth0 | eth1

-dport 53 Destination port 53. (destination port or range of TCP ports)

-Sport 1024:65535 The port or range of source TCP ports (for a range of 1024 to 65535 1024:65535)

-j, --jump Specifies the target to handle the matched package: it can be a custom chain or an existing target (drop, accept etc ..) in this case Accept.

-Target: accept = let the packet through

-j, --jump Specifies the target to handle the matched package: it can be a custom chain or an existing target (drop, accept etc ..) in this case Accept.

**Q.4.2**

Allow www & ssh access to firewall

**- iptables -A CATENA ...**- Aggiunge una regola alla fine della catena indicata

- ORIGIN OF CONNECTION/PACK. = INPUT (I)

- PROTOCOL (PROT) =tcp

- NETWORK INTERFACE (IFACE) = eth1

- dport 22 Destination port 22. (destination port or range of TCP ports)

- Sport 1024:65535 The port or range of source TCP ports (for a range of 1024 to 65535 1024:65535)

- STATE OF THE CONNECTION (STATE) = NEW

-j, --jump Specifies the target to handle the matched package: it can be a custom chain or an existing target (drop, accept etc ..) in this case Accept.

-Target: accept = let the packet through

**Q.4.3**

Allow www & ssh access to firewall

**- iptables -A CATENA ...**- Aggiunge una regola alla fine della catena indicata

- ORIGIN OF CONNECTION/PACK. = OUTPUT (O)

- PROTOCOL (PROT) =tcp

- NETWORK INTERFACE (IFACE) = eth0

- dport 22 Destination port 22. (destination port or range of TCP ports)

- Sport 1024:65535 The port or range of source TCP ports (for a range of 1024 to 65535 1024:65535)

- STATE OF THE CONNECTION (STATE) = NEW

-j, --jump Specifies the target to handle the matched package: it can be a custom chain or an existing target (drop, accept etc ..) in this case Accept.

-Target: accept = let the packet through

Q5: **Short questions on BLP (You have to show your ability to be ability to be concise)**

Provide short answers to the following questions.

(Answers must be short! Using many lines reduces the quality of the answers)

Q5.1 [2/30] Does BLP protect against Trojan horses? (Few lines)

Q5.2 [2/30] Does BLP protect against covert channels? (Few lines)

Q5.3 [2/30] Does BLP help in preserving the data integrity? (Few lines)

**Q.5.1**

BLP has three security properties: (The Simple Security Property, The \* (star) Property, The Discretionary Security Property)

1. The Simple Security Property states that a subject at a given security level may not read an object at a higher security level.

2. The \* (star) Property states that a subject at a given security level may not write to any object at a lower security level.

3. The Discretionary Security Property states that use of an access matrix to specify the discretionary access control.

These rules together serve to prevent information from being propagated without control by the owner, and are the means to prevent the system from being taken over by a trojan horse.

**Q.5.2**  
Covert channels are mentioned but are not addressed comprehensively.

 A covert channel is one that is undetected by analysis of the security model. The fact that metadata (data about data) is leaked from one security level to another is not prevented by the Bell-LaPadula model.

**Q.5.3**

The Bell–LaPadula model only addresses confidentiality, control of writing (one form of integrity), \*-property and discretionary access control, in contrast to the [Biba Integrity Model](https://en.wikipedia.org/wiki/Biba_model) which describes rules for the protection of [data integrity](https://en.wikipedia.org/wiki/Data_integrity).