

# CSE 508 spring 2012: Network Security- Lecture Notes-1

## Network Security Goals:

Network security goal is to provide Confidentiality, Access Control, Availability and Data Integrity. In order to achieve this goal any security mechanism should prevent an attacker from doing the following operations on the data traffic.

Preventing Operation	To provide
Read	Confidentiality
Inject	Access Control
Block	Availability
Modify	Integrity

*Confidentiality:* Keep the contents of data secret.

*Access Control:* Determine whether users are authorized to conduct different actions.

*Availability:* Data should be available to the valid users. (Ex: DOS attacks are to reduce availability).

*Integrity:* Correctness and Completeness of Data i.e. No corruption of data.

## WEP (Wired Equivalent Privacy protocol):

WEP protocol is described in the IEEE 802.11 standard. It is a security algorithm for 802.11 wireless networks. WEP uses the stream cipher RC4 for confidentiality and the CRC-32 checksum for integrity. Every data frame send by a station in a WEP protected network is encrypted. When a station sends a packet, the following steps are executed. (Fig 1)

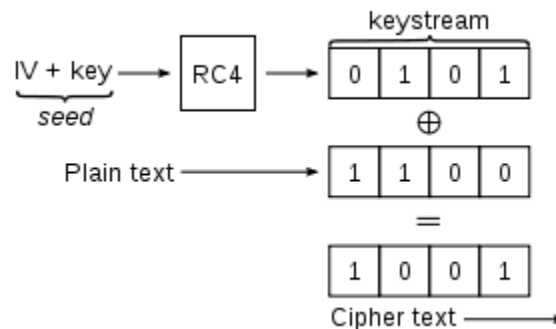


Figure1- WEP process

1. The station picks a 24 bit value called initialization vector **IV**.
2. **IV** is prepended to the **key** (every node has this key 104 bits) and form the per packet key **K**.
3. A CRC32 checksum of the payload is produced and appended to the payload. This checksum is called Integrity Check value (**ICV**).
4. The per packet key **K** is feed into the RC4 stream cipher to produce a key stream **X** of the length of the payload with checksum.
5. The plaintext with the checksum is XORed with the key stream and forms the cipher text of the packet.

6. The cipher text, the IV and some additional header fields are used to build a packet, which is now send to the receiver.

### **Initialization vector (IV):**

An initialization vector (IV) is a fixed-size input to a cryptographic primitive that is typically required to be random or pseudorandom. IV is used because we don't want to reuse the same key stream. WEP uses a short, 24-bit IV. This leads to reuse IVs with the same key and hence makes the key stream vulnerable. The following two methods can be used to implement IV

1. The IV is chosen by a pseudo random number generator (**PRNG**) independently.
2. The node always remembers the last IV used and when a new IV needs to be chosen, add 1 to the last IV and use it as a new IV. When the highest possible number is reached, the node starts again with 0. On startup the IV counter either takes a fixed value or a random number is assigned to it.

### **WEP Weakness:**

#### ***IV is too small:***

WEP's IV size of 24 bits provides only 16 million different key streams for a given WEP key. If IV is reused, the key stream for a given IV is found and hence an attacker can decrypt subsequent packets that were encrypted with the same IV even without knowing the WEP key.

Since there are only 16 million IV values, how the IV is chosen makes a big difference in the attacks based on IV. As explained above, some implementations choose IVs randomly and some assigned sequentially. With a randomly chosen IV, there is a 50% chance of reuse after less than 5000 packets and if assigned sequentially, collisions are inevitable if the values are re-initialized.

#### ***Integrity Check Vector (ICV) is linear:***

WEP's integrity check field is implemented as a CRC-32 checksum, which is part of the encrypted payload of the packet. However, CRC-32 is linear, which means that it is possible to compute the bit difference of two CRCs based on the bit difference of the messages over which they are taken. In other words, flipping bit 'n' in the message results in a deterministic set of bits in the CRC that must be flipped to produce a correct checksum on the modified message. An attacker can therefore change a bit in an encrypted message and know which bit of the encrypted ICV will change as a result.

### **Possible Threat models:**

#### ***1. Chosen Plaintext attack (CPA):***

A chosen-plaintext attack is an attack model for cryptanalysis which presumes that the attacker has the capability to choose arbitrary plaintexts to be encrypted and obtain the corresponding cipher texts. The goal is to gain some further information which reduces the security of the encryption scheme.

#### ***Attack:***

1. Send the known plaintext P1 to AP (access point).
2. AP will encrypt the packet and retransmit the cipher text CT1 and IV.
3. Sniff for the cipher text CT2 (original packet P2) which uses the same IV.
4. XOR both the packets. Since they are using same IV (which means same key stream), the result of the XOR will be P1 XOR P2.
5. Now, XOR P1 with the above result and obtain the text P2.

$E(P1) = P1 \text{ XOR } KS$  (KS-Key stream, E-Encryption)

$E(P2) = P2 \text{ XOR } KS$  (same IV => same KS)

$E(P1) \text{ XOR } E(P2) = P1 \text{ XOR } KS \text{ XOR } P2 \text{ XOR } KS$

$= P1 \text{ XOR } P2 \text{ XOR } KS \text{ XOR } KS$  (since XOR is commutative)

$= P1 \text{ XOR } P2$  (since  $KS \text{ XOR } KS$  cancels out)

$(P1 \text{ XOR } P2) \text{ XOR } P1 = P2$

Thus, with one known plain text, an attacker can decrypt packets with same IV (which was used in the encryption of known plain text). Therefore, with  $2^{24}$  known plaintexts, an attacker can decrypt every packet.

## **2. Chosen Cipher text attack (CCA):**

A chosen-ciphertext attack (CCA) is an attack model in which an adversary has a chance to enter one or more known cipher texts into the system and obtain the resulting plaintexts.

**Attack:** (assume attacker knows the destination ip address DS)

1. Sniff an encrypted packet (IV+CT1) from the network.
2. XOR the Destination address with intended destination address (attackers own address 'ADS') and let it be A.B.C.0
3. Create a new packet by modifying the destination address field. This can be done by XORing the packet with A.B.C.0 with same IV and modified CRC. Since CRC is linear this modification is possible.
4. Now send the modified cipher text (with attacker's destination address) to the AP.
5. AP will decrypt the packet and retransmit the plaintext. Since Ip checksum is also modified, the packet is valid and hence AP will decrypt the packet.
6. Attacker knows the original message.

**Linear property:**  $CRC(P1 \text{ XOR } P2) = CRC(P1) \text{ XOR } CRC(P2)$

**Modifying Destination address:**

$DS \text{ XOR } ADS = NDS$

Packet XOR NDS = New packet with destination address as 'ADS' (since XOR will cancel out DS from the packet).

**Modifying CRC Checksum:**

Let  $DS_H$  and  $DS_L$  denote high and low 16-bit words of the original destination IP address (DS) and we wish to change it to  $ADS_H$  and  $ADS_L$ . Let X is the old checksum.

New checksum  $X_{new} = X + ADS_H + ADS_L - DS_H - DS_L$

Modify the packet by XORing in  $(X \text{ XOR } X_{new})$  which will change the checksum to the correct value  $X_{new}$ .

### 3. Reaction Attack:

Assume an attacker can guess some of the bits in a message. With this information he can determine the value of the bits he does not know.

Example:

Given the highly predictable nature of certain fields of TCP/IP packets, an attacker usually knows some bits in the message. The attacker then flips certain bits in the message, rebroadcasts it, and views whether the packet had a valid TCP checksum by looking for a TCP ACK packet. (NOTE: although ACK packets are encrypted it can be recognized by its length.) By flipping selected bits, the attacker can deduce whether other bits were 0 or 1 by the presence or absence of an ACK response.

### 4. Inductive Attack:

Assume that an attack knows some bytes (say first n bytes) of key stream K for a given IV and a given WEP key.

Attack:

1. Create a packet (like ping which demands a reply) of length n+1 including checksum.
2. Guess the n+1 byte of the key stream (totally 256 possibilities).
3. Send the packet to AP by encrypting with guessed key stream.
4. AP discards the incorrect ones. For correct one (proper encrypted checksum), AP accepts it and responds.
5. If AP responds, remember the n+1 bytes of key stream and repeat the process to find n+2<sup>nd</sup> byte of the key stream.
6. If AP discards (can be determined by the timeout like mechanism), try different possibility for the n+1<sup>st</sup> byte and repeat the steps.

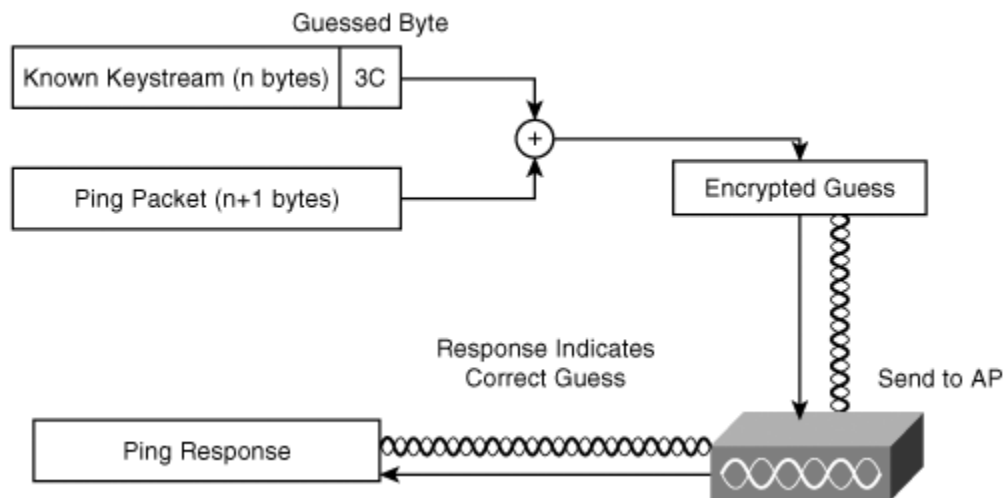


Figure2- Inductive attack

**NOTE:**

If you find any mistakes in the notes, please feel free to correct it.

**Some Useful Resources:**

1. <http://www.opus1.com/www/whitepapers/whatswrongwithwep.pdf>
2. <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>
3. <http://eprint.iacr.org/2007/471.pdf>
4. <http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf>