Network Security

Professor Gaurav S. Kasbekar
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Week - 06
Lecture - 36

Securing Wireless LANs : Part 2

Hello, in the previous lecture, we discussed the authentication and encryption protocols that are used in WEP, or Wired Equivalent Privacy. We will now discuss the security flaws that have been discovered in WEP. So, several weaknesses have been discovered in WEP, and we will discuss some of them in detail, while for the others, we'll provide references. One weakness is the following. In wireless networks, mutual authentication is usually required.

That is, the mobile device should be able to check that the AP is legitimate, apart from the AP checking that the mobile device is legitimate. So, this is because it's easy for an intruder to set up a decoy access point. So, a decoy access point is a fake access point that is set up by the intruder to capture packets. So, this decoy access point is meant for users to believe that it is a genuine access point. And if some users connect to this decoy access point, then the information that they send is captured by the intruder.

So, this decoy access point is set up specifically to capture useful information from legitimate users. So, it's easy for an intruder to set up a decoy access point. For this reason, the mobile device should be able to check that the access point is legitimate. Now, under the WEP authentication scheme that we discussed, can the mobile device check whether the AP is legitimate? So, recall that the protocol ap4.0 was used for WEP authentication.

And that's a one-way authentication protocol. So, the AP can authenticate the client, but the client cannot authenticate the AP. So, the mobile device cannot check whether the AP is legitimate because a decoy access point can just send some value as the nonce R and then falsely claim that it was able to verify the encrypted nonce without even knowing the key. So, authentication is only one-way in WEP. The mobile device is not able to check whether the AP is legitimate, although the AP can check whether the mobile device is legitimate.

So, this is one weakness in WEP: authentication is only one-way. Another weakness is the following: recall that it's good security practice to use different keys for authentication and for encryption. So, we discussed that there is a long-term key used for authentication, and for every session, we set up separate session keys for encryption and message integrity. So, we use a long-term key for authentication and session keys for encryption as well as message integrity. And we discussed different techniques for generating session keys from the long-term authentication key.

However, a weakness of WEP is that, under WEP, the key used for authentication, that is, the long-term key, is also used for encryption of data packets in each session. So, this is a weakness of WEP. So, recall that after authentication, the two parties should agree upon session keys, and we discussed various procedures for agreeing upon the session keys. For example, if the protocol ap4.0 is used, then we mentioned that $(K_{AB}+1)(R)$ can be used as a session key securely. So, we discussed several procedures for agreeing upon session keys.

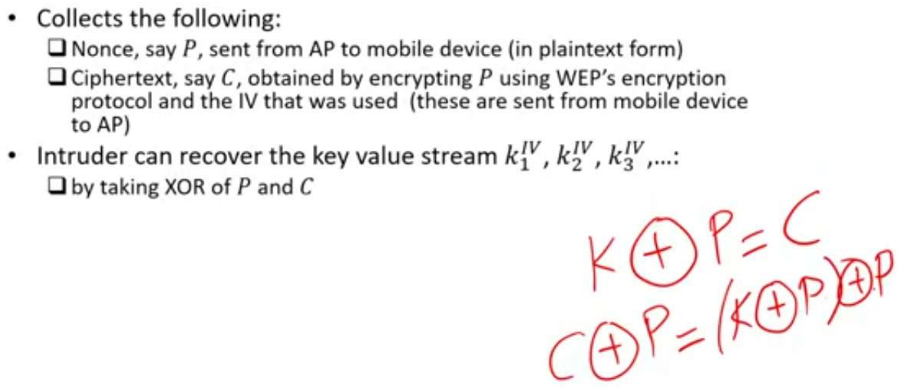
But after WEP authentication, no session keys are agreed upon. Instead, after WEP authentication, the AP and the mobile device just start exchanging data encrypted using the long-term secret shared symmetric key. That is, the key used for authentication. So, for this reason, the authentication process is futile. The authentication process provides no advantage over a scenario where the access point and mobile device directly start exchanging data encrypted using their long-term secret shared symmetric key.

So, the main purpose of authentication is to set up session keys, which can be securely used for encryption and message integrity within the session. But in WEP, authentication is performed, and then the long-term key itself is used for encryption. Hence, there was no point in performing the authentication process. So, if the mobile device and access point had omitted the authentication process and directly exchanged encrypted packets, that would have been equivalent from a security perspective. So, in fact, it's not even equivalent, but we'll see that the authentication process helps an intruder guess the encryption key.

So, the authentication process, in fact, has a negative impact on security. It can help in leaking the encryption key. Now, we show how an intruder can authenticate itself to an access point without even knowing the key. So, the procedure is as follows. Suppose the intruder sniffs the channel while a legitimate mobile device is authenticating itself.

So, this is not difficult to do. So, the intruder just has to capture packets until some legitimate mobile device wants to authenticate, and then the intruder sniffs the entire exchange. So, the intruder collects the following. The nonce sent from the AP to the mobile device is in plaintext form. Let’s call that nonce P, and then, the mobile device encrypts that nonce and generates a ciphertext.

Let’s call that ciphertext C. So, C is the ciphertext obtained by encrypting the nonce P using WEP’s encryption protocol and the IV that was used. So, the IV and the ciphertext C are sent from the mobile device to the access point. The intruder sniffs the channel and collects all this information: the nonce P, as well as the ciphertext C and the IV, which is included in plaintext form in the packet. Now, the intruder can recover the key value stream, that is, $k_1^{IV}, k_2^{IV}, k_3^{IV}$, and so on.



- Collects the following:
  - ❑ Nonce, say P, sent from AP to mobile device (in plaintext form)
  - ❑ Ciphertext, say C, obtained by encrypting P using WEP’s encryption protocol and the IV that was used (these are sent from mobile device to AP)
- Intruder can recover the key value stream $k_1^{IV}, k_2^{IV}, k_3^{IV}$,…:
  - ❑ by taking XOR of P and C

$$K \oplus P = C$$
$$C \oplus P = (K \oplus P) \oplus P$$

How can the intruder recover the key value stream? Simply, by taking the XOR of P and C. So, if we denote the key value stream by K, then recall that in the case of WEP, we have that this is true. $K \oplus P = C$. Hence, we get that $C \oplus P$ equals, if we substitute for C from here, we get that this is equal to $(K \oplus P) \oplus P$, but $P \oplus P = 0$, so hence this is the same as K, the key value stream. So, the intruder can record the key value stream just by taking the XOR of P and C.

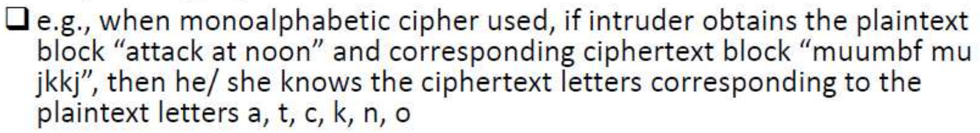
So, the intruder has sniffed the values of P and C, so by taking the XOR, the intruder can get the key value stream. Now, the intruder has obtained enough information to later authenticate itself. The intruder now knows the key value stream, that is K, corresponding to the above IV value. So, now the intruder requests authentication. In this case, the access point sends a nonce P’, which is different from the original nonce P. The intruder then takes this nonce P’ and XORs it with the recovered key value stream, that is K, and sends it along with the same IV value as before.

So, since the IV is the same as before, the key value stream is also the same as before. And then the intruder XORs that key value stream with the nonce P’ sent by the access point.

Hence, the intruder gets the correct encrypted value of the nonce, and the authentication is successful. So, in this way, the intruder can easily break the authentication process of WEP. The only thing that the intruder needs to do is that when some legitimate mobile device is authenticating, the intruder has to sniff the information that is exchanged on the channel, that is this information, P, C, and IV.

So, using this information, the intruder can later authenticate itself to the access point without knowing the key. In summary, the intruder can authenticate itself without knowing the secret key, that is K_S. Now, consider an intruder who is trying to break WEP encryption. So, while attacking an encryption algorithm, as we have seen before, it is often useful for an intruder to obtain some known plaintext blocks and their corresponding ciphertext blocks. So, as a simple example, consider the monoalphabetic cipher which we discussed earlier.

Suppose the monoalphabetic cipher is used, and the intruder obtains a plaintext block 'attack at noon' and the corresponding ciphertext block that is this. So, this was the ciphertext block generated using the monoalphabetic cipher in an example that we discussed earlier. So, if the intruder obtains this plaintext block and the corresponding ciphertext block, then the intruder knows the ciphertext letters corresponding to the letters that appear in the plaintext, that is, a, t, c, k, n, and o. So, the intruder has broken part of the cipher. The intruder knows the ciphertext letters corresponding to all these plaintext letters.



❑ e.g., when monoalphabetic cipher used, if intruder obtains the plaintext block "attack at noon" and corresponding ciphertext block "muumbf mu jkkj", then he/ she knows the ciphertext letters corresponding to the plaintext letters a, t, c, k, n, o

So, whenever these ciphertext letters appear in a future ciphertext, the intruder can know the corresponding plaintext letters. So, in summary, if an intruder is able to get some plaintext blocks and their corresponding ciphertext blocks, then it becomes easy to break the cipher. Now, usually it's hard for an intruder to obtain known plaintext blocks and their corresponding ciphertext blocks because over the communication channel only the ciphertext blocks are sent, and even if the intruder sniffs the channel, they are not able to obtain the plaintext blocks, since these plaintext blocks are never sent without encryption. But when WEP authentication is used, the intruder can easily obtain a known plaintext block and its corresponding ciphertext block. So, how can the intruder obtain these?

That is because the access point sends a 128-byte nonce in plaintext form, and the mobile device responds with the encrypted nonce. Hence, by sniffing the channel, the intruder can obtain one plaintext, that is, the nonce, and the corresponding ciphertext, that is, the encrypted nonce. Now, we mentioned earlier that the same key is used for encryption and authentication. Hence, this plaintext-ciphertext pair can be used by an intruder who is trying to break WEP encryption. So, these plaintext and the corresponding ciphertext, which are captured by the intruder by sniffing on the authentication exchange, can be used to break WEP encryption.

So, in summary, not only does the WEP authentication process fail to authenticate, but it can also assist an intruder in attacking the encryption keys. So, we have discussed that even without knowing the secret key K_S, an intruder can authenticate itself. Hence, the authentication process is not secure. But the authentication process is also harmful because, during the authentication process, the nonce and encrypted nonce are exchanged. And these serve as plaintext and corresponding ciphertext to an intruder who sniffs the channels.

So, that can help the intruder in attacking the encryption keys. Next, we discuss whether replay prevention is provided in WEP. Suppose a legitimate mobile device exchanged some data packets with the access point, which were recorded by an intruder. Then, later on, the intruder can replay one or more of those packets. Recall that there is a CRC in every packet, which can detect modification of packets.

But if an intruder takes an old packet and replays it again, in that case, can the receiver detect that? So, we need a mechanism to prevent such replay attacks. So, recall that we discussed that, in general, sequence numbers can be used for preventing replay, but then the message authentication code must be calculated over some fields, including the sequence numbers, to detect any modification in the sequence numbers. So, assuming that a MAC is computed over the sequence numbers, they can defend against replay attacks. So, we saw that in the case of SSL as well as in the case of IPsec.

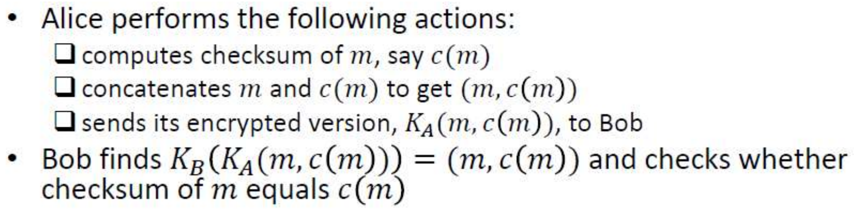
So, sequence numbers were used in those scenarios to prevent replay attacks. But WEP provides no protection against replay. There is indeed a sequence number in the 802.11 medium access control (MAC) header, which increases monotonically. So, the sequence number increases for every successive packet that is sent. But the sequence number is not encrypted, and no MAC is computed over the sequence number.

So, for this reason, an intruder can just capture an old packet, modify the sequence number in it, and send it again. So, only the sequence number is different in the modified packet.

So, the intruder can modify the sequence number, and in that case, the receiver will not be able to detect that this is a replayed packet. The receiver will think that this is a new packet sent by the transmitter. Hence, WEP does not defend against replay attacks.

Next, we discuss whether WEP can provide message integrity. So, what does message integrity mean? It should not be possible for an intruder to sniff a packet from the sender, modify it, and send the modified version to the receiver. So, the receiver should be able to check that the packet indeed came from the legitimate sender and was not modified during transit. We earlier studied the following general method for achieving message integrity when Alice sends a message to Bob.

So, in our lecture on message integrity, we discussed the following protocol. Alice performs the following actions. She computes the checksum of m. Let's call it c(m), and concatenates m and c(m) to get (m, c(m)) and encrypts the result to get a K_A(m, c(m)), where K_A is Alice's secret key and Alice sends K_A(m, c(m)) to Bob. So, Bob then decrypts the value obtained from Alice, that is, Bob finds K_B(K_A(m, c(m))), where K_B is Bob's decryption key. So, the result is (m, c(m)), and then Bob checks whether c(m) is the checksum of m. If yes, then the message integrity check is successful.

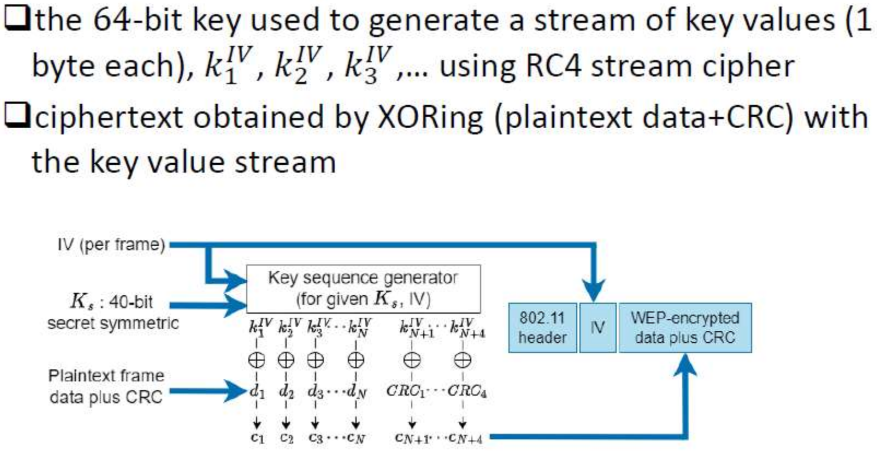


- Alice performs the following actions:
  - ❑ computes checksum of m, say $c(m)$
  - ❑ concatenates m and $c(m)$ to get $(m, c(m))$
  - ❑ sends its encrypted version, $K_A(m, c(m))$, to Bob
- Bob finds $K_B(K_A(m, c(m))) = (m, c(m))$ and checks whether checksum of m equals $c(m)$

We discussed this protocol for message integrity and concluded that this method correctly achieves message integrity. So, WEP uses the same method for achieving message integrity. As the checksum, WEP uses the cyclic redundancy check. But this method is applied in a way that makes it insecure. So, we'll discuss what it is in the protocol used by WEP that makes the message integrity insecure.

So, the mechanism in WEP to provide message integrity is the following. Suppose the plaintext data is N bytes in length, so this is the plaintext data consisting of plaintext bytes d_1 to d_N. Then, a 4-byte cyclic redundancy check is computed for it, that is this CRC$_1$ to CRC$_4$. And then the 64-bit key, which is a combination of the secret symmetry key K_S and the IV, which is of 24 bits. So, this 64-bit key is used to generate a stream of key values, one byte each, which are denoted by $k_1^{IV}, k_2^{IV}, k_3^{IV}$, and so on, up to k_{N+1}^{IV}.

This generation is done using the RC4 stream cipher. And then the ciphertext is obtained by XORing the (plaintext data + the CRC) with the key value stream. So, we see that this mechanism used to provide message integrity is a special case of the general method that we discussed on the previous slide. So, in particular, for a plaintext message that is d_1 to d_N, first the checksum is computed, CRC_1 to CRC_4, and then it is encrypted by XORing it with the key value stream, k_1^{IV} to k_{N+4}^{IV}. So, this is a special case of the method that we discussed on the previous slide.

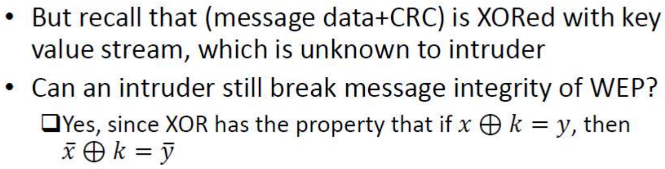


❑the 64-bit key used to generate a stream of key values (1 byte each), k_1^{IV}, k_2^{IV}, k_3^{IV},… using RC4 stream cipher

❑ciphertext obtained by XORing (plaintext data+CRC) with the key value stream

IV (per frame)
K_s : 40-bit secret symmetric
Plaintext frame data plus CRC
Key sequence generator (for given K_s, IV)
k_1^{IV} k_2^{IV} $k_3^{IV}\cdots k_N^{IV}$
$k_{N+1}^{IV}\cdots k_{N+4}^{IV}$
d_1 d_2 $d_3\cdots d_N$
$CRC_1\cdots CRC_4$
c_1 c_2 $c_3\cdots c_N$
$c_{N+1}\cdots c_{N+4}$
802.11 header
IV
WEP-encrypted data plus CRC

But this mechanism for message integrity is not secure for the following reasons. The method used to compute the checksum, that is the 4-byte CRC, is such that it is possible to predict which bits in the CRC will change if we change a single bit in the data message. So, first assume that there was encryption used to encrypt the plaintext and the CRC. So, in this case, to modify the message, an intruder could just change some of the bits in the plaintext and then predict which bits in the CRC should change and then change those corresponding bits in the CRC. So, that way the intruder could modify the message.



- But recall that (message data+CRC) is XORed with key value stream, which is unknown to intruder
- Can an intruder still break message integrity of WEP?
  - ❑Yes, since XOR has the property that if $x \oplus k = y$, then $\bar{x} \oplus k = \bar{y}$

But the complication is that the (message data + CRC) is XORed with the key value stream, which is unknown to the intruder. So, can the intruder still modify the message data and still ensure that the CRC check is successful? So, the answer is yes. Can an intruder still break the message integrity of WEP? The answer is yes because XOR has the property that if $x \oplus k = y$, then it can be easily checked that $\bar{x} \oplus k = \bar{y}$, where $\bar{x}$ is the complement of x, and $\bar{y}$ is the complement of y. So, this property of XOR can be easily checked.

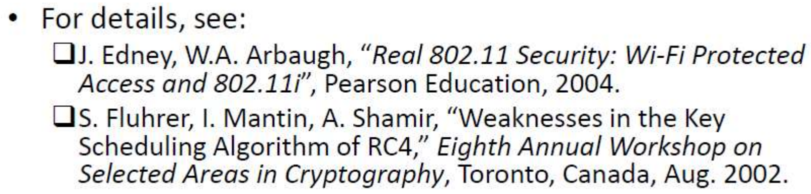
Now, assume that x is one of the plaintext bits in the original message. So, if the intruder flips the corresponding ciphertext bit, then that is equivalent to flipping the corresponding plaintext bit. So, because of this property, the attacker can change some of the bits of the ciphertext corresponding to the message data, then predict which bits of the CRC should be changed to keep the CRC valid, and then change the corresponding bits of the ciphertext. So, if the intruder wants to modify some of the bits of the plaintext, then the intruder just has to flip the corresponding ciphertext bits, then predict which bits of the CRC will change, and then flip those bits of the ciphertext corresponding to the CRC. So, using this process, the intruder can modify the message and still ensure that the ciphertext, that the CRC check will be successful.

Hence, the message integrity scheme used in WEP is not secure for this reason. So, the problem is that the intruder can easily predict which bits of the CRC should change when certain bits of the plaintext are changed. Because encryption is just an XOR operation, the intruder can exploit this property of the XOR operation to flip the desired bits. So, for this reason, message integrity is not secure. Now, another major weakness that was found in WEP was the following.

It was shown that an attacker can find the secret key used for encryption, which we denoted by K_S. An attacker can find the secret key used for encryption in a small amount of time after eavesdropping on the network. So, the attacker has to just eavesdrop on the network and collect several ciphertext frames. And after collecting certain ciphertext frames, the intruder is able to find the secret key. So, for how much time does the attacker have to eavesdrop on the network?

That depends on how much traffic is sent. So, in a busy network, a lot of packets will be sent in a short amount of time. So, depending on the amount of network traffic sent on the network, a successful key recovery may take as little as one minute. So, if it’s a very busy network, where a lot of frames are exchanged on the channel, in that case, just by collecting the ciphertext for one minute or so, the intruder can get enough information to guess the secret key, K_S. Automated tools are available on the internet that can implement this attack.

So, these tools can be used by anyone to get the secret key K_S by just sniffing ciphertext, and then the tool generates the secret key that is used. So, for details, you can see these references. This one is a book by Edney and Arbaugh. And another is this paper. So, these are the details of the procedure that can be used to sniff ciphertext and then get the key.



- For details, see:
  - ❑J. Edney, W.A. Arbaugh, "*Real 802.11 Security: Wi-Fi Protected Access and 802.11i*", Pearson Education, 2004.
  - ❑S. Fluhrer, I. Mantin, A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," *Eighth Annual Workshop on Selected Areas in Cryptography*, Toronto, Canada, Aug. 2002.

So, details are in these references. So, this was a serious weakness because the encryption of WEP itself was broken. So, anyone could just collect a lot of ciphertext, break the encryption completely, and get the key K_S. So, in that case, the security completely collapses. Because of all these weaknesses, WEP was deprecated by IEEE.

And in 2004, 802.11i, which is a more secure standard for 802.11 security, was adopted. So, we'll discuss 802.11i in the next few lectures. In summary, we discussed how WEP operates. WEP is a set of security mechanisms for Wi-Fi, which were included in the original standard introduced in the 1990s. So, we discussed the operation of WEP, how it provides authentication and encryption.

But then, we discussed that WEP has a lot of security flaws in its authentication process, message integrity, replay prevention, encryption, and so on. So, for this reason, WEP was deprecated by IEEE. In future lectures, we'll discuss 802.11i, which is a more secure standard for Wi-Fi security. Thank you.