CAB340: Cryptography

Stream ciphers, Block ciphers, Modes of Operation, Applications

Callum McNeilage – n10482652

1 LFSRs found in real systems

a)

$$2^{16} = 65536$$
$$2^{24} = 16777216$$

b)

$$h(t) = \frac{pq}{gcd(p,q)}$$
$$h(t) = \frac{17 \cdot 25}{1}$$
$$h(t) = 425$$
$$h(t) = \frac{25 \cdot 25}{25}$$
$$h(t) = 25$$

For period to be as long as possible, you want to have both p and q to be different lengths, preferably with one or both being prime in order to maximise the period of the function.

c)

$$\frac{2^{40}}{2} = 549755813888\ of\ possible\ keys$$

d)

CSS was an 'epic failure' because at the time CSS was introduced, the US forbade manufacturers to export cryptographic systems with keys exceeding 40 bits which had been shown to be wholly inadequate against brute-force attack.

e)

One of the 6 designs of Kerckhoff's Cryptographic principle is that "It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will. This principle is broken by DRM, which in relation to Music and Video media, obscures all content from a user who is not using the signed application, even if they have paid for it themselves. This was originally designed to prevent attackers from copying and redistributing copyright material yet is inherently flawed from a cryptographic point of view because it requires that nothing be public knowledge unless accessed through the application that signed the content.

2 Do common block modes protect against cipher linearity?

a)

$$c_1 = \binom{N}{t} \times K \oplus p_1$$

$$c_2 = \binom{N}{t+1} \times K \oplus p_2$$

$$c_3 = \binom{N}{t+1} \times K \oplus p_3$$

Where $K = d \times d$ inverted matrix

b)

Assuming you already have an algorithm that can derive the key for the problem in ECB mode and that the counter is incrementing by 1 for each block in CTR, you can derive the nonce of the CTR by feeding the counter until the counter rolls back to 0, hence, you can find all values used in the encryption - breaking the encryption.

c)

$$c_1 = (c_0 \oplus p_1) \times K$$

$$c_2 = (c_1 \oplus p_2) \times K$$

$$c_3 = (c_2 \oplus p_3) \times K$$

Where $c_0 = IV$ and $K = d \times d$ inverted matrix

d)

Since a block cipher in CBC mode uses the previous ciphertext block as the initialisation vector in the following block's encryption/decryption algorithm, if the attacker already has an algorithm that can derive the key of a block cipher in ECB mode they can then run the CBC encrypted ciphertext through the same algorithm doing a final XOR at the end of each block using the previous block's ciphertext. The attacker can then derive the key for the CBC algorithm as they have knowledge of the plaintext, ciphertext and initialisation vector for each ciphertext block after the first.

3 Using Block Ciphers in Hash Functions

a)

$$W_0 = K$$

$$W_1 = AES(K, m_1)$$

$$\vdots$$

$$W_n = AES(K, m_n)$$

b)

Since for any given run of a plaintext p_1, c_1 will be the same, the function is not one-way

c)

A modern Hash Function requires a minimum 2^{256} attempts to find a collision, whereas AES-128 only requires 2^{128} attempts. Therefore, modern Hash Functions are more secure than AES-128.

4 Message Authentication Codes

a)

i)

$$d = E(m \oplus IV, K)$$

$$d' = E(m' \oplus IV', K)$$

Let d be any given 1-block hash output

$$E(m \oplus IV, K) = E(m' \oplus IV', K)$$

$$m \oplus IV = m' \oplus IV'$$

XOR IV'

$$IV' \oplus m \oplus IV = IV' \oplus m' \oplus IV'$$

$$IV' \oplus m \oplus IV = m'$$

ii)

$$m = m_1 \,||m_2\,|| \,\ldots||m_n$$

$$d \;=\; CBC - MAC(m, K)$$

$$m' = m_1'||m_2'||\ldots||m_n'$$

$$d' \;=\; CBC - MAC(m', K)$$

$$w' = E(m_1' \oplus IV, K)$$

$$w'' = E(m_1^* \oplus d, K)$$

w' = w''

$$E(m_1' \oplus IV, K) = E(m_1^* \oplus d, K)$$

$$m_1' \oplus IV = m_1^* \oplus d$$

$$m_1^* = m_1' \oplus IV \oplus d = d'$$

new message: $m^* = m_1\big||m_2\big|| \ldots \big||m_n\big||m_1^*||m_2^*|| \ldots ||m_n^*$

b)

i)

Since the padding function is $MAC_k(m) = MAC_k(m||0)$, if an attacker can use an oracle to find the tag of any message, the tag will authenticate the same message with 0s appended to the end of the message as there is not a unique unpadded message for any given padded message.

ii)

Assuming the message that the attacker wants to decrypt is the same length as the message the attacker is given, an attacker can find the message tag of the message they are given and then use the message tags of both the given message and the intercepted message in order to derive the intercepted message such that:

$$CBC - MAC(a) = x$$

$$CBC - MAC(b) = y$$

$$CBC - MAC(x \oplus y \oplus a) = b$$

iii)

A padding method that would be more secure against similar attacks would be if the message ends in 0, pad a leading 1 followed by 0s to fill up the block. Additionally, if the message ends in 1, pad a leading 0 followed by 1s.

c)

i)

If the attacker has two messages m and m' then the attack works by slicing the messages together. However, since the value of m' changes based on whether padding has occurred, the attacker does not know the value of m' and so cannot perform the attack.

ii)

Since $k_2 = k_3$, m' is modified in the same way and so attacker knows m' and can perform the attack.

5 Proofs of Work

a)

$$min\ell = \delta + 1$$

$$max\ell = \delta + n$$

Where n = no. characters in message

Because ℓ is length of output and output is δ leading 0s + message length, the minimum possible length of ℓ is $\delta + 1$ and maximum length of ℓ will be equal to δ + the length of the message.

b)

It may still work as a hash function as it is not easy to find second preimage or invert hash, however, it will not work for the Proof of Work application as difficulty is lowered due to the flaw making it easier to find collisions.

c)

It will not work as making it easy to construct collisions will make it easy to derive the cryptographic function due to linearity.