Reading Assignment

• Please read Chapter 18 part 14 of the course textbook by Schneier
• Also NOTE that these lecture notes contain updated information not contained in the course textbook by Schneier – you are still responsible for understanding this lecture!!!
Notation from Katz and Lindell

• \(\{X\}\) is a set of elements of type \(X\)
• \(m\) is a message in plaintext
  • \(m\) is composed of smaller blocks \(m_i\) suitable for individual encryption steps
  • \(m = \{m_i\}\)
• \(c_i\) is ciphertext corresponding to message block \(m_i\)
• \(c\) is ciphertext corresponding to message \(m\)
• \(\text{Enc}_k\) is encryption with key \(k\)
  • \(c \leftarrow \text{Enc}_k(m)\)
• \(\text{Dec}_k\) is decryption with key \(k\)
  • \(m \leftarrow \text{Dec}_k(c)\)
• \(\text{MAC}_k\) is generation of a message authentication code \(t\) with key \(k\)
  • \(t \leftarrow \text{MAC}_k(m)\) or, alternatively, \(t \leftarrow \text{MAC}_k(c)\)
• \(<a,b>\) is a concatenation of \(a\) followed by \(b\)
Message Authentication

• Recall that authentication is the act of declaring something (e.g., a person, a message, or an item such as a car) to be authentic, where an identity is said to be authentic if the claimed identity truly corresponds to the thing (person, message, car, etc.)

• A message is authenticated if the identity of the sender is authenticated and the integrity of the message is verified

• We want to prevent undetected message tampering

• We begin by assuming the existence of a procedure we call a Message Authentication Code or MAC
  • E.g., can use an appropriate one-way hash function with a key
  • Typically the message length is much larger than the MAC output
Approaches

• Two keys: \( k_E \) for encryption and \( k_M \) for message authentication

• Encrypt-and-authenticate
  
  • \( c \leftarrow Enc_{k_E}(m) \)
  
  • \( t \leftarrow Mac_{k_M}(m) \)
  
  • Transmit \(<c,t>\)

• Authenticate-then-encrypt
  
  • \( t \leftarrow Mac_{k_M}(m) \)
  
  • \( c \leftarrow Enc_{k_E}(m, t) \)
  
  • Transmit \( c \)

• Encrypt-then-authenticate
  
  • \( c \leftarrow Enc_{k_E}(m) \)
  
  • \( t \leftarrow Mac_{k_M}(c) \)
  
  • Transmit \(<c,t>\)

©Georgia Institute of Technology, 2018-2019
Encrypt-and-authenticate

- First problem: cryptanalyst can look for clues regarding $m$ using $t$
  - $t \leftarrow Mac_{k_M}(m)$
  - E.g., suppose the first bit of the tag always equals the first bit of the message

- Second problem: deterministic MAC
  - For a deterministic MAC, the tag is identical if the message is identical
  - In practice, most one-way hash functions used for MACs are deterministic
  - An eavesdropper then knows when the same message has been sent twice, and hence this approach is not secure against CPA
Authenticate-then-encrypt

• Problem: CPA
  • Consider the padding oracle attack
    • If an error in the padding is detected, a “bad padding” error is returned
  • Since it is the case that $c \leftarrow Enc_k(m,t)$, there are now two potential sources of decryption error
  • Consider the modified decryption algorithm...
Encrypt-then-authenticate

Construction 4.18

• Given \( k_E, k_M, \) MAC and \( \pi_E = (Enc, Dec) \)
  • \( t \leftarrow Mac_k(m) \)
  • E.g., suppose the first bit of the tag always equals the first bit of the message

• Define \( Enc' \) and \( Dec' \) as follows
  • \( Enc'(m) \):
    • \( c \leftarrow Enc_{k_E}(m) \)
    • \( t \leftarrow Mac_{k_M}(c) \)
    • Ciphertext output is \( <c,t> \)
  • \( Dec'(<c,t>) \):
    • First check if \( Mac_{k_M}(c) = t \)
    • If yes, output \( Dec_{k_E}(c) \)
    • If no, output that there has been an error