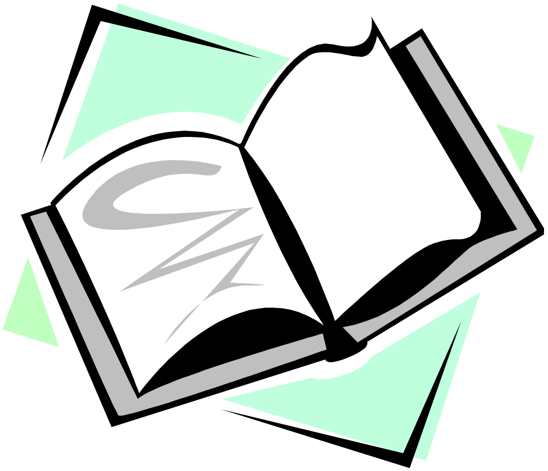
Cryptography
CS 555

Topic 13: Message Authentication Code

Outline and Readings

- Outline
  - Review of HW1
  - Message authentication code and its security definition
  - Construction of MAC using PRF



- Readings:
  - Katz and Lindell: : 4.1-4.4

HW1: Problem 2: Breaking enhancement of Vigenere

- Let k_i denote the Vigenere key stream
- Let m_i denote the message stream
- Let z_i denote the ciphertext stream
  - $z_1=x_1+k_1; \ldots; z_{13}=x_{13}+k_{13}; z_{14}= m_1+x_1+k_{14}$
- We have
  - $z_{14}-z_1=m_1+k_{14}-k_1$ and more generally

    $$z_{j+13} - z_j = m_j + k_{j+13} - k_j$$
- Under known message attack, one could easily decrypt another ciphertext (of same or less length)
- Under ciphertext-only attack against the sequence $z_{j+13}-z_j$ this is similar to Vigenere with 13 times original key length

Problem 5 & 6

- For arbitrary symmetric cipher
  - It must be that $|M| \le |C|$, and it is possible that $|M| < |C|$.
  - Each is possible: $|M| > |K|$, $|M| = |K|$, and $|M| < |K|$.
  - Each is possible: $|C| > |K|$, $|C| = |K|$, and $|C| < |K|$.

- For symmetric cipher that gives perfect secrecy
  - It must be that $|M| \le |C|$, and it is possible that $|M| < |C|$.
  - It must be that $|M| \le |K|$, and it is possible that $|M| < |K|$.
  - Each is possible: $|C| > |K|$, $|C| = |K|$, and $|C| < |K|$.
    - Different keys can have same effect
    - Encryption can be randomized

Problem 7 & 8

- Problem 7. Exercise 2.5. Consider the "encryption" scheme $Enc_k(m)=m$
- Problem 8. Exercise 2.7 Prove that Pr [**M=**m | C=c] = Pr [**M** = m] implies Pr[$\textbf{PrivK}^{\textbf{eav}}_{\textbf{A},\Pi}$=1] =½ prob
  - For any pair (m0,m1) chosen by A, A's behavior can be defined by giving a prob p_c for each c, which is the prob that A outputs 0 when seeing c; then the prob of A winning is
    $\sum_c \Pr[c](\Pr[A(c)=0]\Pr[m_0|c] + \Pr[A(c)=0]\Pr[m_1|c])$
    $= \sum_c \Pr[c]\left(p_c \frac{1}{2} + (1-p_c)\frac{1}{2}\right) = \frac{1}{2}$

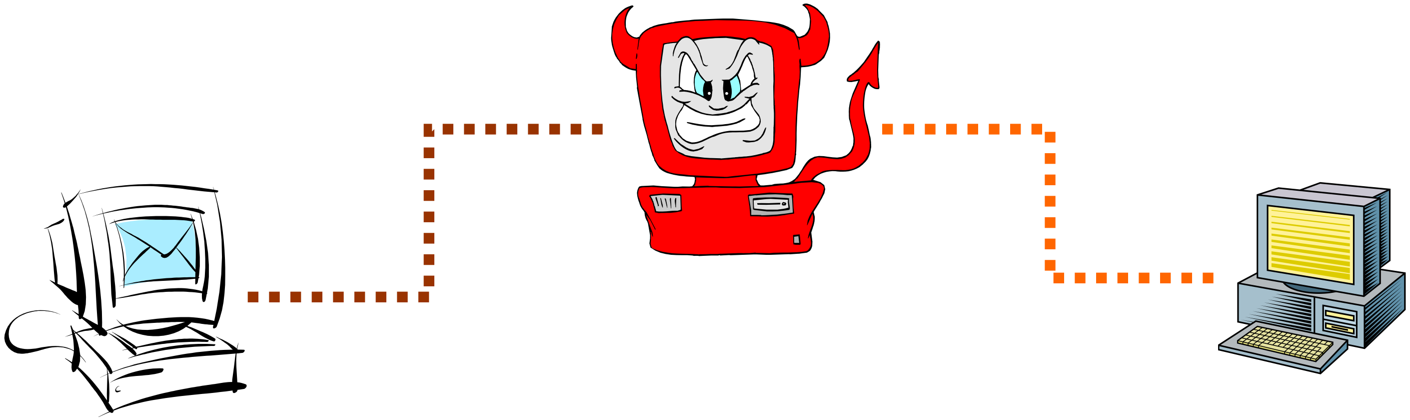
Problem 9

- Exercise 2.7 Prove that $\Pr[\mathbf{PrivK}^{eav}_{\mathbf{A},\Pi}=1] = ½$ implies that $\Pr[\mathbf{M}=m \mid C=c] = \Pr[\mathbf{M}=m]$
- Proof idea. If $\Pr[\mathbf{M}=m \mid C=c] = \Pr[\mathbf{M}=m]$ does not hold, then $\exists c_0, m_0, m_1$ s.t.

  $$\Pr[\mathbf{M}=m_0 \mid C=c_0] > \Pr[\mathbf{M}=m_1 \mid C=c]$$

- Construct A as follows:
  - A outputs m_0, m_1
  - If A receives c_0, output 0. Otherwise, A outputs 0 with prob ½ .
  - $\Pr[A(c_0)=0]\Pr[m_0|c_0] + \Pr[A(c)=0]\Pr[m_1|c_0] = \Pr[m_0|c_0] > \frac{1}{2}$

Data Integrity and Source Authentication



- Encryption does not protect data from modification by another party.
- Need a way to ensure that data arrives at destination in its original form as sent by the sender and it is coming from an authenticated source.

Security Objectives/Properties (C, I, A)

- Confidentiality (secrecy, privacy)
  - only those who are authorized to know can know

- Integrity (also authenticity in communication)
  - Only modified by authorized parties and in permitted ways
  - Any unauthorized modification can be detected
  - Do things that are expected

- Availability
  - those authorized to access can get access

Encryption vs. Message Authentication

- Encryption using stream ciphers
  - Flipping any bit in ciphertext results in corresponding bit flipped after deryption
- Encryption using block ciphers
  - OFB & CTR the same as above
  - What about the ECB mode?
  - What about the CBC mode?
- An observation
  - Encryption schemes so far have the property that every string of certain length are valid ciphertexts
  - To provide message authentication, must make valid ciphertext “sparse” among all string

Message Authentication Code

- Assume that sender and receiver share a secret key, which can be used for authentication.
- A message authentication code (or MAC) consists of the following three PPT algorithms
  - $k \leftarrow \mathbf{Gen}(1^n)$ key generation
  - $t \leftarrow \mathbf{Mac}_k(m)$ tag-generation
  - $b := \mathbf{Vrfy}_k(m,t)$ verification algorithm

    b=1 meaning valid, b=0 meaning invalid

  Must satisfy $\forall k\ \forall m\ \mathbf{Vrfy}_k(m, \mathbf{Mac}_k(m)) = 1$

  When m must be from $\{0,1\}^{\ell(n)}$, this is a **fixed-length** MAC.

Security of MAC

- The experiment **Mac-forge**$_{\mathbf{A},\Pi}$
  - $k \leftarrow$ **Gen**(1^n)
  - Adversary A is given oracle access to **MAC**$_k(\cdot)$
  - Adversary outputs (m, t). Let Q denote the set of all queries that A asked to the oracle.
  - Adversary wins if **Vrfy**$_k(m, t) = 1$ and $m \notin Q$

- A MAC Π **is existential unforgeable under an adaptive chosen-message attack** (or just **secure**) if for all PPT A, there exists a negligible function negl such that

  Pr[**Mac-forge**$_{\mathbf{A},\Pi}$=1] $\leq$ negl(n)

Types of Forgery Attacks

- **Existential forgery**: adversary chooses the message to forge after querying the MAC oracle

- **Selective forgery**: adversary chooses one message before carrying out the attack, and then cannot query the message

- **Universal forgery**: adversary can create MAC for any message after querying the MAC oracle

Replay Attacks

- A secure MAC ensures that adversary cannot generate new messages that can be authenticated
- It does not prevent replaying of an old message

- Standard ways to defend against replay attacks include
  - Using sequence numbers for messages
  - Using timestamp for messages
  - Using random nonce
    - $A \rightarrow B$: n where n is a freshly chosen random number, aka, a nonce
    - $B \rightarrow A$: (m, n, MAC_k(m,n))

Fixed-length MAC using PRF: Construction 4.3

- Let F be a PRF. Define a fixed-length MAC as follows:
  - **Gen**(1^n) outputs $k \leftarrow \{0,1\}^n$ uniformly at random
  - **Mac**$_k(m)$ outputs $t := F_k(m)$
  - **Vrfy**$_k(m,t) = 1$ iff $t = F_k(m)$

  This is fixed-length because m can be chosen only from the input domain of F.

- Theorem 4.4. If F is a PRF, then this construction is a secure fixed-length MAC.
- Proof idea. Obviously secure if F is a random function. How to construct the distinguisher given A that breaks the MAC?

Extensions to Variable-Length Messages

- Methods that do not work. First divide message into blocks then,
  - XOR all blocks together, and then compute tag on the result.
  - Authentication each block separately.
  - Authentication each block with a sequence number.

Construction 4.5

It seems best to skip this construction.

- Basic idea: a msg is divided into blocks, the last block is padded with 0's; then compute the tag for each block separately; when computing the tag for the *i*'th msg block, include the following information
  - A newly generated random identifier r
    - This ensures that the tag for one msg cannot be used for another msg
  - The length of the message ℓ
  - The index of the block i
  - The i'th block of the msg m_i

  Need to ensure that each of the four fields is at most n/4 bits long.

  What if ℓ is not included?

Proof Idea.

- Creating an existential forgery (m,t) implies one of the following event must occur
  - **Repeat**: the same identifier r is used in two msgs
    - How a forgery can occur?
    - Prob of this occurring is negligible
  - **Forge**: The adversary's msg and tag (m,t) includes one block that does not appear before (in answers to oracle queries)
    - When neither **Repeat** not **Forge** occurs:
      - t must be from one of previous msgs.
      - m must be of the same length as the previous msg
      - Every single block must be the same

Coming Attractions …

- CBC-MAC; Collision-resistant hash functions

- Reading: Katz & Lindell: 4.5,4.6