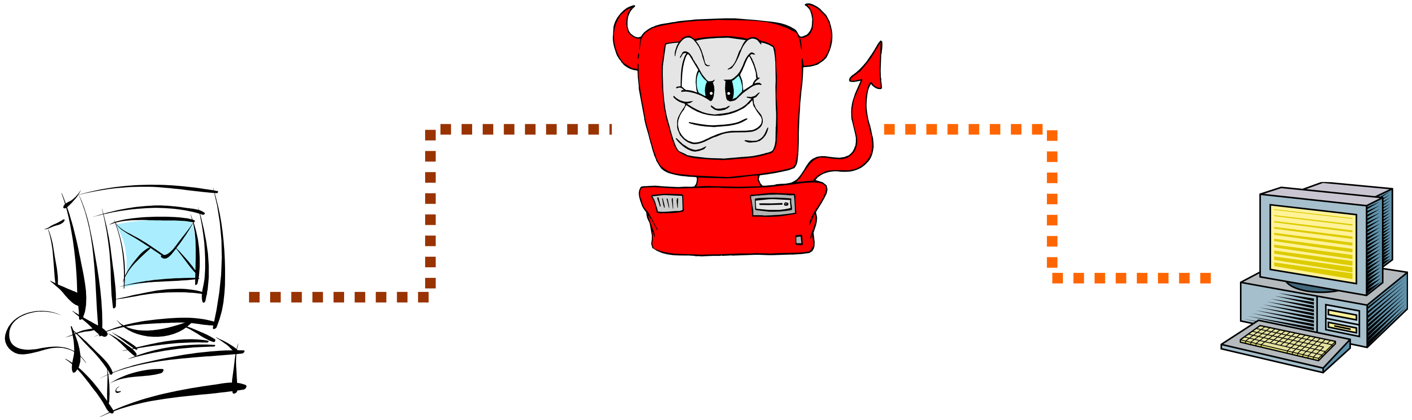
ENEE 459-C
Computer Security

Message authentication

UNIVERSITY OF MARYLAND

Data Integrity and Source Authentication



- Encryption does not protect data from modification by another party.
  - Why?
- 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.

Hash Functions

- A hash function maps a message of an arbitrary length to a m-bit output
  - output known as the fingerprint or the message digest

- What is an example of hash functions?
  - Given a hash function that maps Strings to integers in $[0,2^{32}-1]$
  - $F(x) = A x + b \bmod q$, where $x = 0,1,\ldots,T$ where $T \gg q$

Using Hash Functions for Message Integrity

- Method 1: Uses a Hash Function h, assuming an authentic (adversary cannot modify) channel for short messages
  - Transmit a message M over the normal (insecure) channel
  - Transmit the message digest h(M) over the secure channel
  - When receiver receives both M' and h, how does the receiver check to make sure the message has not been modified?

- This is insecure. How to attack it?
- A hash function is a many-to-one function, so collisions can happen.

Cryptographic Hash Functions

Given a function $h:X \rightarrow Y$, then we say that h is:

- preimage resistant (one-way):

  if given $y \in Y$ it is computationally infeasible to find a value $x \in X$ s.t. $h(x) = y$

- 2-nd preimage resistant (weak collision resistant):

  if given $x \in X$ it is computationally infeasible to find a value $x' \in X$, s.t. $x' \neq x$ and $h(x') = h(x)$

- collision resistant (strong collision resistant):

  if it is computationally infeasible to find two distinct values $x', x \in X$, s.t. $h(x') = h(x)$

Relations between properties

- collision resistance $\Rightarrow$ 2nd preimage resistance
- 2nd preimage resistance $\Rightarrow$ preimage resistance

Non-crypto Hash (1)

- Data $X = (X_0, X_1, X_2, \ldots, X_{n-1})$, each X_i is a bit
- **hash**$(X) = X_0 + X_1 + X_2 + \ldots + X_{n-1}$
- What is the compression of this hash?
- Show it does not satisfy preimage resistance
- Show it does not satisfy collision resistance

Non-crypto Hash (2)

- Data $X = (X_0,X_1,X_2,\ldots,X_{n-1})$
- Suppose hash is
  - $h(X) = nX_0+(n-1)X_1+(n-2)X_2+\ldots+1\cdot X_{n-1}$
- What is the compression of this hash?
- Show it does not satisfy preimage resistance
- Show it does not satisfy collision resistance

Non-crypto Hash (3)

- Cyclic Redundancy Check (CRC)
- Essentially, CRC is the remainder in a long division calculation
- Find a collision (modulo x^8+1)
- Good for detecting burst **errors**
- Easy to construct collisions
- CRC sometimes mistakenly used in crypto applications (WEP)

Find collisions for crypto-hashes?

- The brute-force birthday attack aims at finding a collision for a cryptographic function h
  - Randomly generate a sequence of plaintexts $X_1, X_2, X_3, \ldots$
  - For each X_i compute $y_i = h(X_i)$ and test whether $y_i = y_j$ for some $j < i$
  - Stop as soon as a collision has been found
- If there are m possible hash values, the probability that the i-th plaintext does not collide with any of the previous $i - 1$ plaintexts is $1 - (i - 1)/m$
- The probability F_k that the attack fails (no collisions) after k plaintexts is

$$F_k = (1 - 1/m)(1 - 2/m)(1 - 3/m) \ldots (1 - (k - 1)/m)$$

- Using the standard approximation $1 - x \approx e^{-x}$

$$F_k \approx e^{-(1/m + 2/m + 3/m + \ldots + (k-1)/m)} = e^{-k(k-1)/2m}$$

- The attack succeeds with probability p when $F_k = 1 - p$, that is,

$$e^{-k(k-1)/2m} = 1 - p$$

- For p=1/2

$$k \approx 1.17\, m^{1/2}$$

- For m = 365, p=1/2, k is around 24