Hash Functions, MACs, Digital Signatures

Dawn Song

dawnsong@cs.berkeley.edu

Review

- **Modes of Operations for Block Ciphers**
  - **How to encrypt long messages**
- **Public-key encryption**
  - **RSA**
  - **Why textbook RSA is not secure?**

How to Fix?

- **Padding:**
  - **Pad short plaintext to block size**
  - **Add randomness**
- **Can't just do random padding**
  - **E.g., given data D, pad message m to be m= 00 | 02 | r | 00 | D, where r is a random number of appropriate length**
  - **Bleichenbacher found an attack (1998)**
- **Standard: OAEP (Optimal Asymmetric Encryption Padding)**
  - **With a formal proof of security**

Sample Applications

- **Integrity check for storage**
- **Commitment**

Hash Function Properties

- **Hash function: a function h with properties**
  - **Compression: h maps an input x of arbitrary length to an output h(x) of a fixed length**
  - **Ease of computation: given h and x, it's easy to compute h(x)**
- **Additional important properties**
  - **Preimage resistance**
  - **2^{nd}-preimage resistance**
  - **Collision resistance**

Three Properties

- **Preimage resistance**
  - **For any y (in the range of h) for which a corresponding input is not known, it is computationally infeasible to find any input x such that h(x) = y.**
- **2^{nd}-preimage resistance**
  - **It is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x, to find $x' \neq x$ s.t. h(x) = h(x')**
- **Collision resistance**
  - **It is computationally infeasible to find any two distinct inputs x and x' which has to the same output, i.e., h(x) = h(x')**

Examples

- **RSA-based one-way function**
  - $f(x) = x^e \bmod N$, where factorization of N is unknown
  - Under RSA assumption, f(x) is preimage resistant
  - What about 2^{nd}-preimage resistance?
- **DES-based one-way fucntion**
  - $f(x) = E(k, x) \oplus x$, for any fixed known key k.
  - Under the assumption that E is a random permutation, f(x) is preimage resistant

Relationships btw Properties (I)

- **Does collision resistance imply 2^{nd}-preimage resistance?**
  - yes
- **Does preimage resistance imply 2^{nd}-preimage resistance?**
  - No
- **Does 2^{nd}-preimage resistance imply preimage resistance?**
  - No

Relationships btw Properties (II)

- **Does collision-resistance imply preimage resistance?**
  - E.g., let g be a hash function which is collision resistant and maps arbitrary-length inputs to n-bit outputs. Consider function h:

    h(x) = 1 || x, if x has bitlength n
    0 || g(x), o.w.
  - Is h collision resistant?
  - Is h preimage resistant?
- **Different applications need different properties**

Cryptographic Hash Functions

- **MD5**
  - **Output 128-bit**
  - **Designed by Ron Rivest, 1991**
  - **Xiaoyun Wang et. al. found collision in one hour using IBM p690 cluster, 2004**
  - **Klima find collision with one minute on a notebook computer, using tunneling, 2006**
- **SHA-1**
  - **Output 160-bit**
  - **Designed by NSA, adopted by NIST, 1993**
  - **Xiaoyun Wang et. al. found attack on SHA-1, 2005**
    - **Requiring fewer than 2^{69} operations to find a collision, whereas brute force would require 2^{80} operations**
  - **More improvements on attacks**
- **NIST is looking for new hash functions**
  - **Similar competition as in AES**
  - **Submissions due Oct 31, 2008**

Administravia

- **Waitlist**

Message Authentication Code (MAC)

- **Encryption: secrecy/confidentiality**
- **What if Mallory tries to change the message?**
- **Can encryption alone help?**
- **What about adding a checksum?**
- **Message authentication code (MAC)**
  - **Provides assurance of source & integrity of msg (data origin authentication)**
  - **$f(k, M) = f_k(M)$, k is secret key**
  - **Unforgeability:**

    For any fixed value of k unknown to adversary, given a set of values $(x_i, f_k(x_i))$, it is computationally infeasible to compute $f_k(x)$ for any new input x.
- **Sample construction: HMAC**
  - **$HMAC(x)= h((k \oplus r)||h((k \oplus s)||x))$, r and s are random numbers**

Secure Two-party Communication

- **Confidentiality**
- **Integrity**
- **For a message m, send Enc(k1, m), MAC(k2, Enc(k1, m))**
  - **Alice and Bob share k1 and k2**
- **Is the problem solved?**

Replay attacks

- **Cryptosystems are vulnerable to replay attacks**
- **Record message; playback later identically**
  - **"Yes"/"No"**
- **Solution: use nonces (random bits; timestamp) etc.**
  - **Freshness property**
- **Message is <text, timestamp>**

Digital Signatures

- **MACs**
  - **Only parties who have the shared key can verify data integrity & origin**
  - **Symmetric-key model**
- **Digital signatures**
  - **Asymmetric-key model**
  - **Sender has public/private key**
  - **Anybody with public key can verify data integrity & origin---non-repudiation**
  - **Applications**
    - **Broadcast setting**
    - **Proof of endorsement**
      - **Comparison with physical signatures**

RSA Signature

- **Idea:**
  - **Let p, q be large secret primes, N = pq**
  - **Given e, find d, such that ed ≡ 1 mod φ(N), where φ(N)=(p-1)(q-1)**
  - **public key: e, N**
  - **private key: d, p, q**
  - **Signature: $s = h(m)^d \bmod N$**
  - **Verification: s^e ?= h(m) mod N**
- **What if h is not collision-resistant?**
- **In practice, RSA-PKCS (public-key cryptography standards)**