Applied Cryptography
CPEG 472/672
Lecture 4A

Instructor: Nektarios Tsoutsos

Cryptographic Security

- We want to quantify security
  - Not necessarily binary (secure, insecure)
  - What is the effort to break a cipher?
- Two types of security
  - Informational Security
    - E.g. One Time Pad (impossible to break)
    - Theoretical focus
  - Computational Security
    - E.g. AES (can’t break in reasonable time)
    - Practical focus

Computational Security

- Expressed using
  - t: number of operations
  - e: probability of attack success
- We say (t, e)-secure cipher
  - Attacker that performs at most t operations has success probability at most e
- Ideal block cipher with n-bit key:
  - Here, the best attack is brute force
  - (t, t/2^n)-secure cipher

Measure security level in bits

- Assume successful attack (e almost 1)
  - We can express security as the number of operations t required for the attack
  - t-bits of security == 2^t steps required
- Security level may differ from key size
  - E.g., RSA with 2048-bit secret key offers ~100 bits of security

Attack parameters

- Parallelism?
  - Sequentially dependent operations
  - Independent operations
- Memory cost?
  - How much memory is consumed?
  - How many memory lookups are needed?
- Precomputation (offline stage)
  - Time/memory tradeoff
- Number of targets

Achieving security

- How to ensure ciphers can't be broken?
- Provable security
  - Mathematical proofs that algorithms can't be broken
  - Security reductions: Any method to break the cipher also yields a method to solve a known hard problem
- Heuristic security
  - Evidence that highly skilled people tried and failed

Provable Security

- Proofs relative to a hard math problem
  - E.g., factorization of large integers
  - Breaking the cipher is as hard as solving a mathematical problem
- Proofs relative to another crypto problem
  - We can only break the given cipher if we can first break another crypto primitive
  - E.g., A Feistel network is secure as along as the underlying function F is a secure PRP

Heuristic Security

- Provable security does not apply to all cryptographic algorithms
  - Many symmetric ciphers don’t have a proof
- Heuristic Security
  - AES can’t be reduced to another problem
  - AES itself is the hard problem!
  - Experts try to break reduced versions of algorithms (e.g., cipher with fewer rounds)
  - Establish a security margin

Key generation

- Random keys (e.g., via PRNG)
  - Symmetric (e.g., AES)
  - Asymmetric (primes for RSA)
    - Requires a key generation algorithm
    - openssl genrsa 4096
- Using a password
  - Requires a key derivation function
- Using a key agreement protocol
  - Two or more parties establish a shared a key

Protect generated keys

- Wrap keys using AES encryption
  - openssl genrsa -aes128 4096
  - User provides a password to unlock
- Generate keys from password on the fly
  - Vulnerable if password is weak
  - Dictionary attacks
- Store key using secure hardware
  - USB dongle
  - User enters a password to unlock

Reading for next lecture

- Aumasson: Chapter 5