COMP7370 Advanced Computer and Network Security

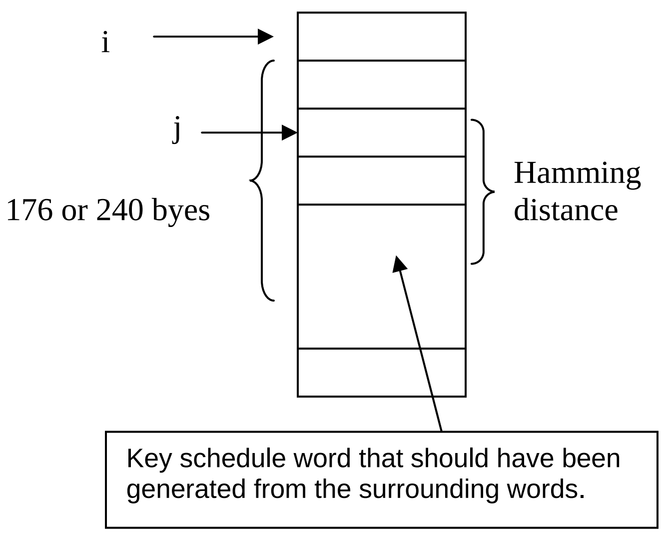
Cold Boot Attacks on Encryption Keys (3)

Topics:

1. Identifying keys in memory
2. Attacking encrypted disks
3. Countermeasures

Topic 1: Identifying keys in memory

- **Question:** How will you identify keys in RAM?
    - Statistical tests
    - Locate program data structures
- Identify AES keys (see slide)
    - Input: a memory image
    - Output: a list of likely keys
    - Basic idea: (1) key schedules rather than original keys
      (2) Recover keys from their key schedules



1. Iterate through each byte of memory. Treat the following block of 176 or 240 bytes of memory as an AES key schedule.
2. For each word in the potential key schedule, calculate the Hamming distance from that word to the key schedule word that should have been generated from the surrounding words.
3. If the total number of bits violating the constraints on a correct AES key schedule is sufficiently small, output the key.

Topic 2: Attacking disks (encrypted)

- Conditions:
    - Laptops are stolen (**why** we have this condition for memory attacking threats? – physical access to DRAM)
    - Powered on
    - Suspended. (**why?** To attack keys on DRAM)
    - **Discussions on an exception**:
        - How to extract keys from DRAM even if computers are powered off for a long time?
        - When the machine boots, the keys will be loaded into RAM automatically

- BitLocker
    - Windows Vista, 7, server 2008 (Enterprise and Ultimate editions)
    - full disk encryption
    - AES, 128-bit keys

AES in CBC mode(Elephant(Encrypt(Sector pad key, byte offset of section) XOR (sector plaintext), CBC key)
Step 4 Step3; Step 1 Step 2

Note: elephant is a diffuser function developed by Microsoft. The purpose of these un-keyed functions is solely to increase the probability that modifications to any bits of the ciphertext will cause unpredictable modifications to the entire plaintext sector.

- Attack BitLocker
    - cuts the power (Windows)
    - connect the USB disk, and then reboots
    - dump the memory image to the external disk
    - run *keyfind* on the image -> candidate keys (**What types of keys?** sector pad key and the CBC encryption key)
    - if keys are found, mounts the encrypted volume in Linux.

- FileVault
    - Apple, reverse-engineered (see [44])
    - Mac OS X10.4
    - 128-bit AES in CBC mode
    - I: block with logical index
    - IV = encryp(I, AES key, k2) = HMAC-SHA1_k2(I)
    - Second key k2 and AES key are protected in a header
        - Header = (AES key, k2)
        - Encrypt(user pwd, header)

- Attack FileVault
    - A Mac System with a FileVault volume mounted
    - Extract a memory image
    - run *keyfind* on the image -> AES keys
    - AES key can decrypt 4080 bytes for each 4096-byte block
    - Attack IV key in DRAM:
        - Test 160-bit substrings of DRAM
        - Substrings XOR decryption of the first part of the disk block -> plausible plaintext?
        - **Why?** First part of disk block(plaintext) XOR IV key = cipher i.e., 160-bit substrings
    - Use AES key and IV key to decrypt

Topic 3: Countermeasures

- Scrubbing memory: (**Discussions**)
    - Do not save keys in memory

- Do not page out key to disks
- Clear memory at boot time
- **Attacker** moves DRAM to another PC

- Limiting booting from network or removable media
  - Boot from primary disk. Is it safe? No
  - **Attacker** swaps out this disk

- Suspending a system
  - Lock screen. Is it safe? No
  - Sleeping and hibernating modes. Safe? No if pwd is in RAM
  - Safe way: Key in DRAM = encrypt(External pwd)

- No precomputation
  - Precomputation speed cryptographic oprations
  - Keys are vulnerable (attack subkeys = redundant key information)

- Key expansion
  - Make it more difficult to reconstruct keys
  - How? More key transforms

- Physical defense
  - Physically protect memory (lock)

- Encryption in disk controller
  - No software, no key in DRAM
  - Key register in disk controller
  - Safe? When OS is booted, key must be erased in disk controller