Cryptography Part V: Secret Sharing

ECE 4156/6156 Hardware-Oriented Security and Trust

Spring 2024

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

Reading

- Handbook of Applied Cryptography, Chapter 12.7, pp. 524-528
- Introduction to Modern Cryptography, Chapter 13.3, pp. 501-507

Background

- Consider a situation where you want to require t out of N users to make a request or else the request is not granted
  - For example, consider a safe vault with secret documents and N executive officers
    - E.g., $N = 5$
  - The policy may be to require t of the N officers to open the safe vault
    - E.g., $t = 3$
- Other similar situations may exist with missile codes, encryption keys (e.g., in a secure boot process), passwords distributed geographically among several servers, and other financial/bank account scenarios

Some Initial Cases

- Case #1: $t = N$
  - Suppose we use an ℓ-bit number where $N \ll 2^\ell$
  - Choose $t = N$ ℓ-bit numbers s_i uniformly at random – note that each is called a "share"
    - $s_1, \ldots, s_i, \ldots, s_N$ – note that each s_i is called a "share" of the secret
    - Define the secret s to be $s = s_1 \oplus \ldots \oplus s_i \oplus \ldots \oplus s_N$
  - Clearly, all $t = N$ users' secrets are needed to recover s
  - Also, any set of N-1 users' secrets reveals nothing about s
    - This "reveals nothing" claim can be statistically proven
    - You can also see this by trying to devise an attack
- Naive approach
  - For example, consider a 128-bit key divided into eight locations on a chip for secure boot
  - You might say let's divide this into 16-bit numbers, i.e., $|s_i|$=16 for each "share"
    - But now suppose that the adversary finds seven of the locations
    - With brute force effort, the 128-bit key can be guessed in 2^{16}=65,536 steps which can be < 1 second

Cases Where $t < N$

- Case #2: $t < N$
  - There are two subcases
    - Exactly t users' shares are needed to open the safe (more generally, obtain the secret)
    - t or more than t users' shares can be quickly combined to open the safe
  - Can we use the XOR based approach (see previous page)?
    - Consider $N = 6$
      - There are 15 combinations of two people A and B: A&B, A&C, A&D, A&E, A&F, B&C, B&D, B&E, B&F, C&D, C&E, C&F, D&E, D&F, E&F
      - There are $\binom{6}{3} = \frac{6!}{3!(6-3)!} = \frac{6*5*4*3*2*1}{3*2*1(3*2*1)} = 20$ combinations of three people
      - Et cetera
    - Conclusion: for allowing any three out of six executives to open a safe, each executive would have to be issued 20 keys with the safe performing 20 comparisons in the worst case each time the safe is opened
    - In Computer Science, the XOR based approach is not considered "efficient"

Efficient Secret Sharing for $t < N$

- Example case for $t = 2$

Efficient Secret Sharing

- Mathematics for efficient secret sharing was simultaneously and independently developed by Adi Shamir (of RSA fame) and George Blakley in 1979
  - Blakley, G.R. (1979). "Safeguarding Cryptographic Keys" (PDF). *Managing Requirements Knowledge, International Workshop on (AFIPS)*. **48**: 313–317. doi:10.1109/AFIPS.1979.98
  - Shamir, Adi (1 November 1979). "How to share a secret". *Communications of the ACM*. **22** (11): 612–613. doi:10.1145/359168.359176
  - https://en.wikipedia.org/wiki/Secret_sharing