Cybersecurity: Secret Sharing, Key Escrow

Ozalp Babaoglu

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

Problem with private keys

- In many situations, a secret (key, password) is known to only a single individual
  - The key in symmetric cryptography used to encrypt a file or the the entire hard drive (e.g., *FileVault* in MacOSX, *BitLocker* in Windows)
  - Password, PIN for a system or service
- What happens if you *lose* or *forget* the secret or if you *die* (and your boss needs to access the files you have encrypted)?
- Also a legal and political issue: a government would like to be able to eavesdrop on encrypted communications if there is a court order or there is danger of national security; FBI wants to unlock a phone belonging to a dead terrorist

Escrow

- ***Escrow*** (Dictionary definition): a contractual arrangement in which a third party receives and disburses money or documents for the primary transacting parties, with the disbursement dependent on conditions agreed to by the transacting parties
- Similar to the role that a third party like *Paypal* plays in an on-line transaction (e.g., *eBay* auction) between two untrusting parties (a *seller* and a *buyer*)
- ***Key Escrow***: an arrangement in which private keys are "held in escrow" such that under certain circumstances, an authorized third party may access or reveal them

Key Escrow: Naïve solutions

- "Give" a copy of your private key to a trusted figure known as the company's *Chief Security Officer* (CSO)
- Instead of "giving" the private key, it can be saved it on a storage device (e.g., hard disk, USB stick), after being encrypted with the public key of the CSO
- Or it can be kept on a smart card whose use is audited
- What if you do not trust the CSO?
- Or worse, disaster strikes!

Key Escrow in Practice

- September 11, 2001
  - "Not long after the planes struck the twin towers, killing 658 of his co-workers and friends, including his brother, one of the first things on Lutnick's (CEO of *Cantor Fitzgerald*) mind was passwords. No one knew the passwords for hundreds of accounts and files that were needed to get back online in time for the reopening of the bond markets. Cantor Fitzgerald did have extensive contingency plans in place, including a requirement that all employees tell their work passwords to *four nearby* colleagues. But now a large majority of the firm's 960 New York employees were dead"

Secret sharing

- Split the secret into n pieces, and send each to a different security officer (or encrypt the i^{th} piece with the public key of security officer i and keep them all on your disk)
- No proper subset of security officers should be able to reconstruct the secret (any easier than brute-force attempt)
- All n security officers collaborating should be able to do so
- How to split the secret into n pieces?

Secret sharing: naïve solution

- Simply divide the secret S that is k bits long into n segments of $\lceil k/n \rceil$ bits each (except for the last one)

$S = S_1 \,||\, S_2 \,||\, \ldots \,||\, S_n$

- Example:

S_1 S_2 S_3

$S = 111010110001\,010001010101\,01001110000$

$k = 35$

$n = 3$

$\lceil k/n \rceil = 12$

Problems with naïve solution

- Does not satisfy the requirement "No proper subset of security officers are able to reconstruct the secret any easier than brute-force attempt"
- Each security officer *knows exactly* $\lceil k/n \rceil$ bits of the secret
- Any subset of $m<n$ security officers can carry out a brute-force attack over the remaining unknown bits
- Example:
  - initial brute-force search space: 2^{35}
  - with just one security officer: reduced to 2^{23}
  - with two security officers: further reduced to 2^{11}

2-way secret sharing

- Split secret S among two security officers such that
  - ***Confidentiality***: Neither security officer alone has any information regarding S
  - ***Availability***: The two of them together can reconstruct S

2-way secret sharing

- Let S be a secret k bits long and let $n = 2$
- Generate a random string R of length k
- Give to the first security officer the string $S_1 = R$
- Give to second security officer the string $S_2 = R \oplus S$
- Reconstruct secret as $S_1 \oplus S_2$

2-way secret sharing: example

Secret:	1	1	0	1	0	0	0	1	
Random string R:	1	0	0	0	1	1	1	0	S_1
$\oplus$	0	1	0	1	1	1	1	1	S_2
$\oplus$	1	1	0	1	0	0	0	1	Secret

2-way secret sharing

- The algorithm is correct:
- ***Confidentiality***: Neither security officer alone has any information regarding S
  - one has a random string
  - the other has S encrypted with a one-time pad
- ***Availability***: The two of them together can reconstruct S as $S_1 \oplus S_2 = R \oplus (R \oplus S) = (R \oplus R) \oplus S = 0 \oplus S = S$

n-way secret sharing

- Generalize to n
- Split secret S among n security officers such that
  - ***Confidentiality***: No subset of m security officers with $m<n$ has any information regarding S
  - ***Availability***: All n of them together can reconstruct S

n-way secret sharing

- Let S be a secret k bits long
- Generate $n-1$ random strings $R_1 R_2 ... R_{n-1}$ each k bits
- Give to the first $n-1$ security officers the strings

$$R_1 R_2 ... R_{n-1}$$

- Give to the last (n^{th}) security officer the string

$$R_1 \oplus R_2 \oplus ... \oplus R_{n-1} \oplus S$$

- Reconstruct the secret

$$S_1 \oplus S_2 \oplus ... \oplus S_{n-1} \oplus S_n$$

n-way secret sharing: example (n=4)

Secret:	1	1	0	1	0	0	0	1	
Random string R_1:	1	0	0	0	1	1	1	0	S_1
Random string R_2:	0	1	0	1	0	1	1	1	S_2
Random string R_3:	1	0	1	0	1	0	1	1	S_3
$\oplus$	1	0	1	0	0	0	1	1	S_4
$\oplus$	1	1	0	1	0	0	0	1	Secret

Threshold schemes

- What if one of the security officers dies? The remaining pieces are useless and the secret cannot be recovered
- Need to relax the requirement that all n pieces be present to reconstruct the secret
- (t, n)-Threshold scheme (where $t \le n$):
  - Secret split into n "shares"
  - t (or more) shares sufficient to recover secret
  - Less than t shares unable to recover secret

Secret sharing: (t, n)-Threshold Shamir's method over a finite field

- Generate a random polynomial of degree $t-1$

$$g(x) = (a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \ldots + a_1x + S) \textbf{ mod } p$$

where the coefficients $a_1 \ldots a_{t-1}$ are selected randomly, S is the secret and p is a random prime (made public) larger than any of the coefficients

- Generate the n shares as

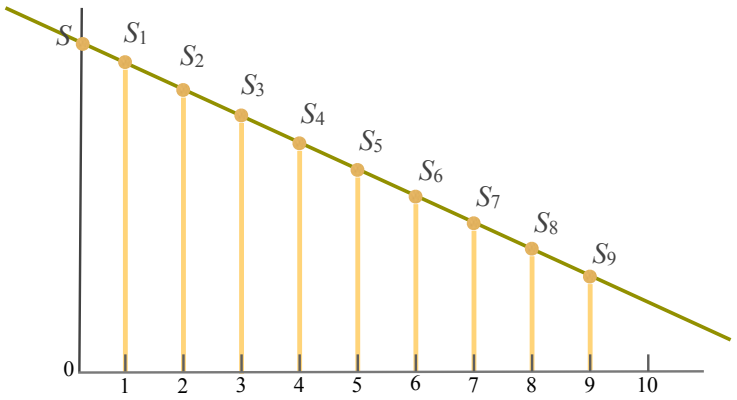
$$S_1 = g(1), S_2 = g(2), \ldots, S_n = g(n)$$

- Discard the polynomial (coefficients and S)
- Any t shares sufficient to solve for the t unknowns to obtain the polynomial (and the secret S)

Shamir's method over a finite field Example

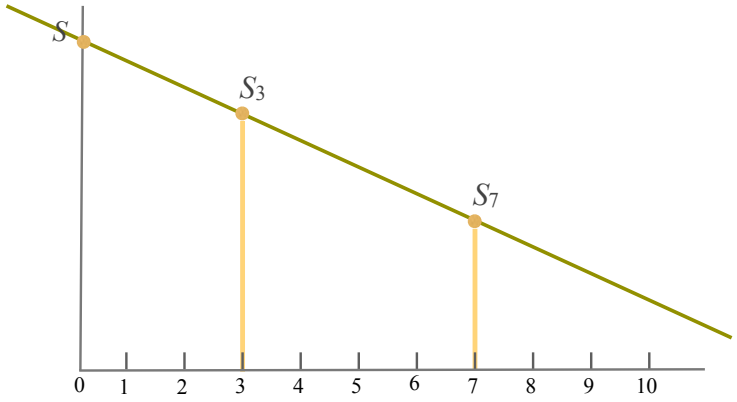
- Consider an example of $(2,n)$-Threshold scheme

$g(x) = (ax + S)$ **mod** p



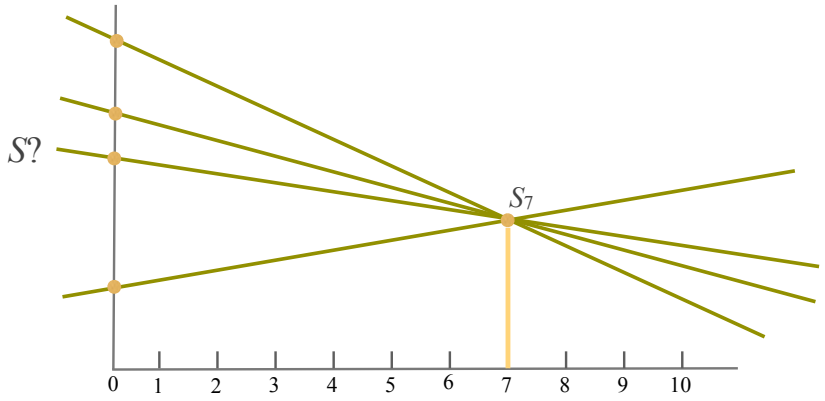
Shamir's method over a finite field Example

- Given any two (or more) shares



Shamir's method over a finite field Example

- Given only one share



Shamir's method over a finite field (2,3)-Threshold numeric example

- $n = 3,\ t = 2$
- $S = 6,\ p = 17$
- Generate a random polynomial of degree $t = 2-1 = 1$

$$g(x) = (4x + 6) \bmod 17$$

- Calculate the 3 shares to give to 3 security officers:

$$g(1) = (4 + 6) \bmod 17 = 10$$
$$g(2) = (8 + 6) \bmod 17 = 14$$
$$g(3) = (12 + 6) \bmod 17 = 1$$

Shamir's method over a finite field (2,3)-Threshold numeric example

- How can 2 security officers, say 1 and 2, reconstruct the secret?
- They know that the polynomial has the form

$$g(x) = (ax + S) \bmod 17$$

- Given their shares $g(1) = 10$ and $g(2) = 14$, they have

$$g(1) = (a + S) \bmod 17 = 10$$
$$g(2) = (2a + S) \bmod 17 = 14$$

- Solving the system of equations for the unknown S, they obtain $S = 6$ which is the initial secret

Shamir's method over a finite field (2,4)-Threshold implementation

- (2,4)-Threshold implementation using Shamir's method based on the prime $2^{128}-159$ after encrypting the secret with AES-128 using a random 128-bit key
- http://equatinelabs.com/secretshare.html

Historical note

- In 1993, the *United States National Security Agency* (NSA) developed the "Clipper Chip" with a built-in backdoor, intended to be adopted by telecommunications companies for voice transmission
- Symmetric cipher Skipjack with 80-bit key similar to DES
- Diffie-Hellman key exchange algorithm for distributing keys
- Device manufacturers required to deposit keys with the government in escrow
- The key could be given to other government agencies to decrypt telephone conversations with proper authorization
- Project abandoned in 1996

Update

- Edward Snowden revelations of 2013
- Widespread surveillance by the NSA of international partners, foreign nationals and U.S. citizens
- Apple and Google announce that they will encrypt data stored on their smartphones in a way such that they could not break the encryption even if ordered to do so with a warrant

Update

- BBC, 21 October 2015 "Apple tells US judge iPhones are "impossible" to unlock"
  - *The data encrypting services that come with the latest smartphone and computer operating systems can only be unlocked when a specific key is used, according to Dr. Joss Wright of the Oxford Internet Institute. "Apple may supply the device and system but if they don't have that key they're not able to unlock it any more than the US state department" he told the BBC*

Update

- The Washington Post, 28 March 2016 "FBI has accessed San Bernardino shooter's phone without Apple's help"
  - *The Justice Department is abandoning its bid to force Apple to help it unlock the iPhone used by one of the shooters in the San Bernardino terrorist attack because investigators have found a way in without the tech giant's assistance, prosecutors wrote in a court filing Monday*

Update

- New York Times, 13 January 2020 "Barr Asks Apple to Unlock Pensacola Killer's Phones"
  - *Attorney General William P. Barr declared on Monday that a deadly shooting last month at a naval air station in Pensacola, Fla., was an act of terrorism, and he asked Apple in an unusually high-profile request to provide access to two phones used by the gunman*