One-time passwords

Madison Garofolo

Problems with passwords

- If a password is leaked, security is gone
- If you always use the same password to log in, an adversary only needs to eavesdrop on a single interaction to get access to your account

The solution: One time passwords

- Goal: ID protocol secure against eavesdropping
- Define new attack game for measuring security

Secure identification: Eavesdropping Attack

- Identification protocol: I = (G, V, P)
- Adversary: A, given *vk*
- P must prove to V their identity

Key generation:

$$(vk, sk) \xleftarrow{\$} G()$$

Eavesdropping:

A requests Q interactions between P and V
Run Q times:
$\quad T_i \leftarrow (P_{sk}, V_{vk}(P_{sk}))$
Give $T_1, T_2, ..., T_Q$ to A

Impersonation:

A communicates with V, and wins if V accepts

ID2adv[A,I] = probability that A wins

Verification key (vk)

- Regular password protocol requires vk to be kept secret
- In this security definition, vk is given to A from the beginning

Eavesdropping Attack (weak version)

- Identification protocol: I = (G, V, P)
- Adversary: A, NOT given *vk*
- P must prove to V their identity

Key generation:

$$(vk, sk) \xleftarrow{R} G()$$

Eavesdropping:

A requests Q interactions between P and V
Run Q times:
  $T_i \leftarrow (P_{sk}, V_{vk}(P_{sk}))$
Give $T_1, T_2, \ldots, T_Q$ to A

Impersonation:

A communicates with V, and wins if V accepts. A gets unlimited attempts

wID2adv[A,I] = probability that A wins

Stateful vs Stateless

- Old password protocols were stateless: (vk, sk) never changes
- Stateful protocol: (vk, sk) changes after each successful interaction
- For stateful protocols we modify the game so that:
  - Adversary has unlimited verification attempts (unless vk is public)
  - Adversary can make verification attempts *between* receiving transcripts
    - Each round, adversary either eavesdrops or impersonates

HOTP

- HOTP: Hash-based one time password (weakly secure)
- Let F be a PRF defined over (K, Z_N, Y) (for a large integer N)
  - Usually HMAC-SHA is used

G:
$k \xleftarrow{\$} K$
output $sk := (k, 0), vk := (k, 0)$

P with $sk = (k, i)$:
send $r := F(k, i)$ to V
set $sk \leftarrow (k, i+1)$

V with $vk = (k, i)$ on input r:
if $r = F(k, i)$ accept; set $vk \leftarrow (k, i+1)$
Otherwise, reject

Security proof relies on the fact that F is a secure PRF

HOTP Problems

- Must maintain shared state between V and P
  - Can use time as implicit counter
- Infrequent validation
  - Current counter value is valid for a long time

TOTP

- TOTP: Time-based one time password (weakly secure)
- Counter incremented every 30 seconds
- Security proof relies on the fact that F is a secure PRF

S/key

- In TOTP, if *vk* is leaked then security is completely lost
- With S/key this is not the case
  - but you can only use it so many times before you need to regenerate (*sk, vk*)
- Uses hash chain: $H^{(j)}(x)$; meaning x is hashed j times

G:
$k \xleftarrow{\$} X$
return $sk := (k, n), vk := H^{(n+1)}(k)$

P with $sk = (k, i)$:
send $t := H^{(i)}(k)$ to V
set $sk \leftarrow (k, i-1)$

V with vk on input t:
if $vk = H(t)$ accept; set $vk \leftarrow t$
Otherwise, reject

Fully secure under the strong definition! Even if *vk* is public only P can successfully prove their identity

Security of S/key

- Security relies on H being a one way function
    - Specifically, given $H^{(j)}(x)$ it is hard to find x (or any element) that maps to $H^{(j)}(x)$
    - For any $j = 1, 2, \ldots, n$ for some n.

Drawbacks of S/key

- In order for H to be one way, X (key space) must be very large
    - In practice, at least 128 bits (at least 22 characters)
- Very inconvenient when someone has to manually type the one time password

Sources

Boneh, D., Shoup, V. (2023). A Graduate Course in Applied Cryptography.