OUTLINE

1 Basis of authentication:

- what you know, what you possess, what you are.

2 Password-related techniques

3 Attacks on passwords and defense mechanisms

4 Authentication tokens and biometrics

AUTHENTICATION PROTOCOL

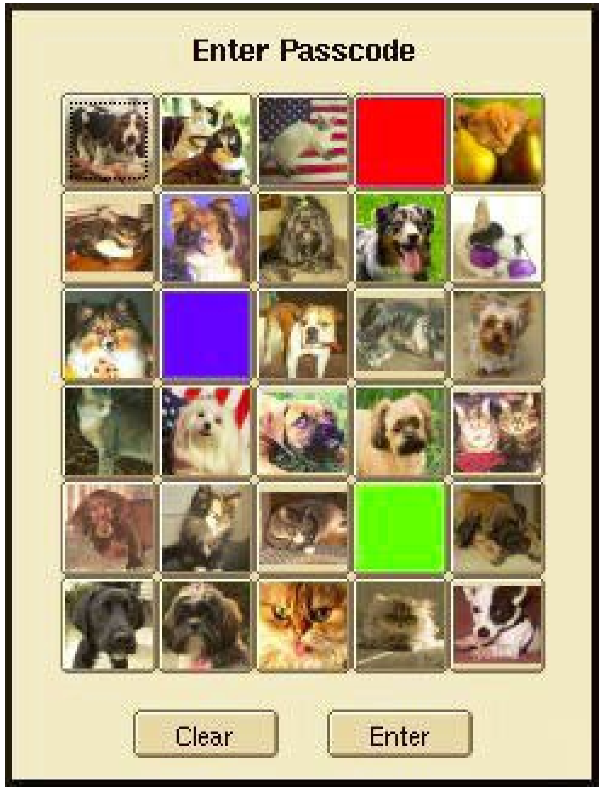
Weak/Simple Authentication:

- Password-based.
- Unilateral: one entity (claimant) proves its identity to the verifier.
- Prove knowledge of secret by giving up the secret

Strong Authentication:

- Involves mutual authentication; both parties take both the roles of claimant and verifier:
- Challenge-response protocols: sequence of steps to prove knowledge of shared secrets.
- Prove knowledge of secret WITHOUT giving up the secret (zero knowledge proofs)

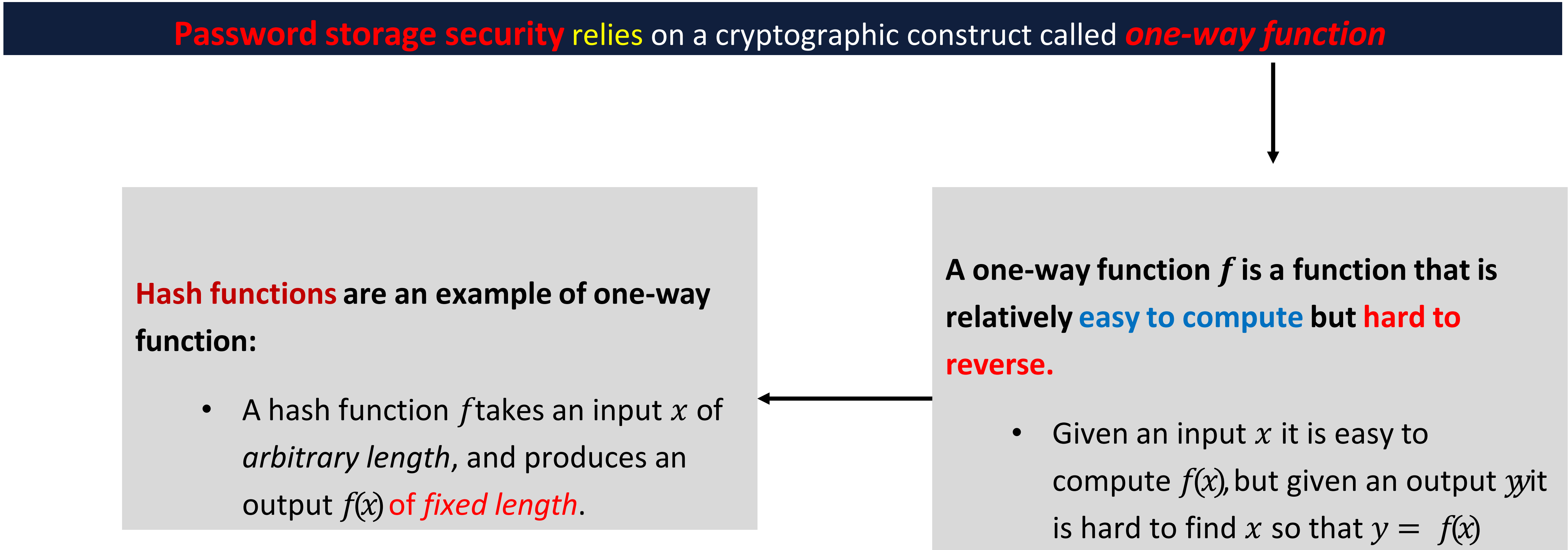
PASSWORD-RELATED TECHNIQUES



- **Password storage:**
  - **Plaintext (BAD)** or **“encrypted” (fair)** or **“hashed” (good).**
- **Password policies:**
  - What rules need to be imposed on the selection of passwords by users, number of failed attempts, etc.
- **“Salting” of passwords.**
- **Alternative forms of passwords**
  - Passphrases, one-time passwords, visual passwords.

Salt is random data that is used as an additional input to a one-way function that "hashes" a password. Salts are used to safeguard passwords in storage. The primary function of salts is to defend against dictionary attacks.

ONE WAY FUNCTIONS



Password storage security relies on a cryptographic construct called *one-way function*

A one-way function f is a function that is relatively easy to compute but hard to reverse.

- Given an input x it is easy to compute $f(x)$, but given an output y it is hard to find x so that $y = f(x)$

Hash functions are an example of one-way function:

- A hash function f takes an input x of *arbitrary length*, and produces an output $f(x)$ of *fixed length*.

PROPERTIES OF HASH FUNCTIONS

Suppose *H* is a hash function. We say *H* satisfies:

- *Pre-image resistant* if given a hash value y, it is computationally infeasible to find x such that $H(x) = y$.
- *Collision resistant* if it is computationally infeasible to find a pair (x, y) such that $x \neq y$ and $H(x) = H(y)$.

Recap: A one-way function f is a function that is very easy to compute but hard to reverse. Hash function is an example of one-way function. Impt Hash Functions: : **SHA256,512,KECCAK (crypto)** ,ARGON2,bcrypt (for password hashing)

PASSWORD STORAGE

Plaintext

- Passwords stored in plaintext.
- Claimant’s password is checked against the database of passwords.
- No protection against insider (system admin) or an attacker who gains access to the system. Hence dispute is possible!

Hashed/ encrypted passwords

- Passwords are encrypted, or hashed, and only the encrypted/hashed passwords are stored.
- Claimant’s password is hashed/encrypted, and checked against the database of hashed/encrypted password.
- Some degree of protection against insider/attacker.

PASSWORD STORAGE

In operating systems, password hashes are stored in a password file.

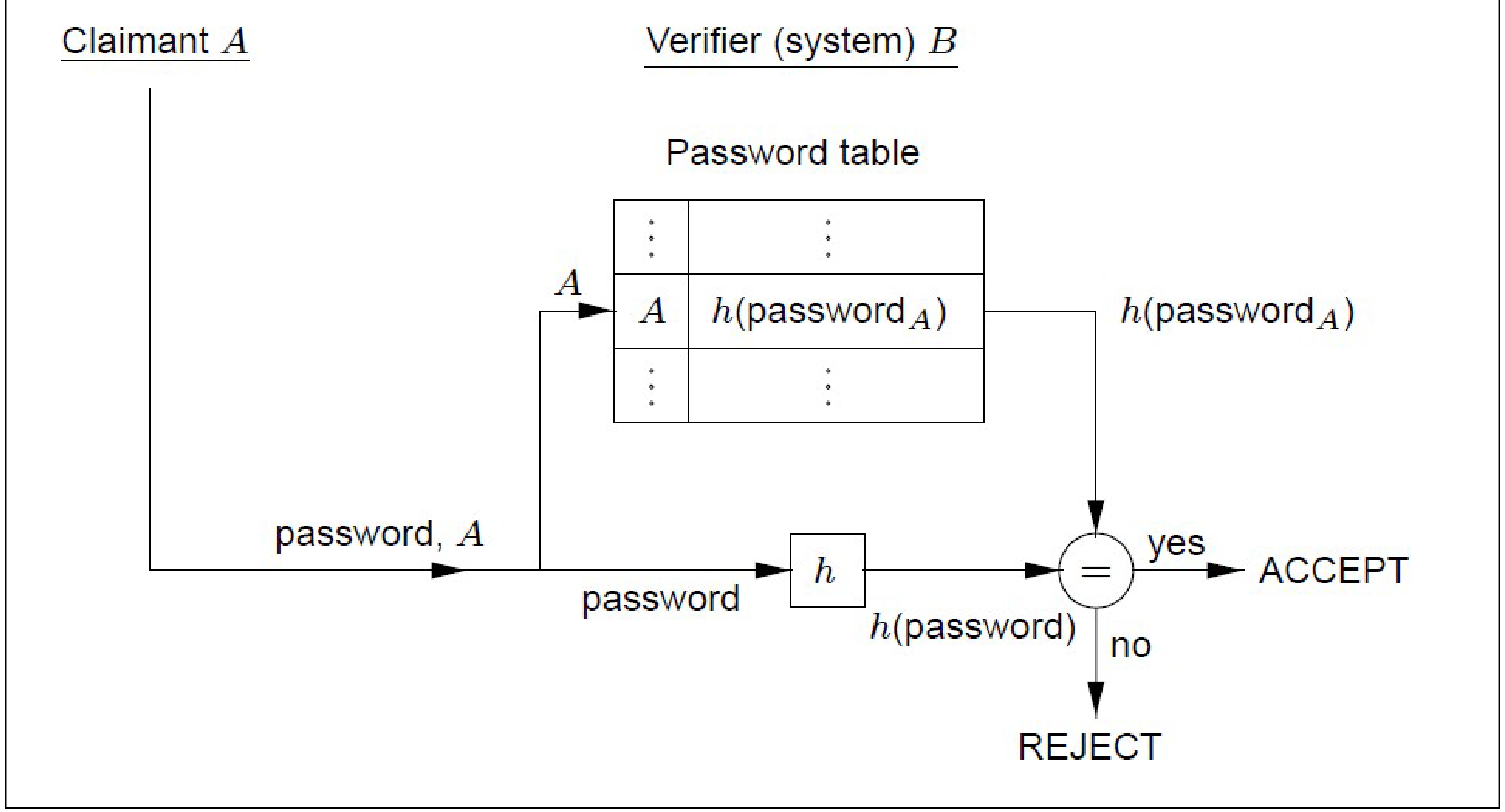
In Windows system, passwords are stored in Security Accounts Manager (SAM) file (**%windir%\system32\config\SAM**).

In Unix, this is **/etc/passwd**, but in modern Unix/Linux systems it is in the *shadow* file in /etc/shadow.

- At the application levels, passwords may be held temporarily in intermediate storage locations like buffers, caches, or a web page (don't save passwords in cache!)
- The management of these storage locations is normally beyond the control of the user; a password may be kept longer than the user has bargained for.

HASHED PASSWORD VERIFICATION

Notice that the verifier **does (should) not** store the passwords, only their hashes



Source: Menezes et al. Handbook of Applied Cryptography.