- Salt can be stored in plain text
- Hash should be final (not reversible)
- Passwords should not be able to find plaintext
- Attacker steals password file
- Save password in hash or salted hash
62g = t g password = 2.47
52 + 10 = 62

Algorithm: 
PassWord = Reverse of Alphabet

L > lower case = 2, 160
160 bit hash

Password = reversed hash.
redacted attack set -> hashes

{ general hashes

build dict of candidate pw

D'chenary aitches
Salted password

Find password

Start password recovery (whenever)

\textit{Password lost (case)}

Password is not saved
Salted password

- for each password
  - generate hash of plain
  - look for match
Exercise of shares + witness

Share

Increase

Benefit from not being too expensive

Receivers are happy
as table

STOR

pm

hash

hash reversed function

Random table
# of Causal Pol = 1,000,000

Loops steps

Lesson Chain

Use

Lot

$12k$
$\text{Shared Secret} \leftarrow f(s', s)$

Verifier

Challenge Response

Host

Authentication
Pre-v key match (H(r))

1. Pop
2. Push key (no cleanup cost)
fixed in timestamp

1995

then in a reply address based

1978

and key version 1978

key

1977

show key

Nechemia & Selveder

KA4 - 1977
2. Aus Anlass des systems

Wunsch Domains

4 in 1980 (MIT)

Symmetrie

Kernbrenser (basis an Needham scholars)
Known to Bob 1 Tags $\rightarrow$ K-B-Tgs

Server (Bob) has a secret key

K-B has a secret key known to everyone as a user id or password

Alice has a user id or password