
2. Hash Functions - Today, Monday

Hash Function

1) Variable length input
2) Fixed bit output.
3) It's possible for 2 different inputs $x, y$ such that $H(x) = H(y)$. But this shouldn't occur easily.
4) Should be fast to compute.

Uses

1) Message Authentication
   - Proof message hasn't been changed since it left the sender

2) Digital Signature
   - Proof of who sent the message.
4. Methods to use a hash function for message authentication

(a) \( E(K, [M \| H(m)]) \)

\( \| \) → concatenate
\( E(K, \text{something}) \) : encrypt something using the symmetric key \( K \).

I receive \( E(K, [M \| H(m)]) \), I decrypt, get \( M \| H(m) \) → run through hash function to double check that this matches.

(b) \( M \| E(K, H(m)) \)

I receive \( M \| E(K, H(m)) \). Decrypt to get \( H(m) \). Then calculate \( H(m) \) and see if they match.

(c) \( M \| H(m) \| S \), \( S = \text{Secret Value} \).

I receive this, I know \( S \). Calculate \( M \| H(m) \| S \).
Then I calculate \( H(m) \| S \) and see if it matches.

(d) \( E(K, [M \| E(\text{PRa}, H(m)]) \)

Both digital sign AND msg auth AND confidential.

I receive + decrypt = \( M \| E(\text{PRa}, H(m)) \).
I calculate \( H(m) \).
Use \( \text{PRa} \) to decrypt this part to reveal \( H(m) \), see if they match.
Storing Passwords (hashing is used)

Stealing a file with plaintext passwords is very bad!

Instead this better:

**Name** \( H(\text{pswd}) \)

Bob 110100... Just because I know you.

Alice 010111... it's hard to calc \( x \) s.t. \( H(x) = y \).

**Rainbow Table**

Generate few million "common" passwords. For each, calculate \( H(x) \). Then look for matches in this...

If ANY of the 1000 users has abc123, we'll figure who it is...

**Improvement = SALT**

\[
\begin{array}{ccc}
\text{Name} & \text{Salt} & H(\text{pswd} || \text{Salt}) \\
Bob & 110101... (rand bit string) & 01.... \\
Alice & 011011... & Multiples work by # of users. \\
\end{array}
\]

Now if I add SALT to Bob's SALT, I can only break Bob's if he had one I guessed.
Requirements for a good hash function
1. Variable input size
2. Fixed output size
3. Fast to compute

4. Given output value $y$, it should be computationally infeasible to find any $x$ such that $H(x) = y$. [PRE-IMAGE RESISTANCE]

5. Given an input $x$, it's computationally infeasible to find $y$, $y \neq x$, such that $H(x) = H(y)$. [SECOND IMAGE RESISTANCE]

6. Hard to find ANY $x, y$ such that $H(x) = H(y)$ [COLLISION RESISTANCE]

\[ \text{Birthday Paradox!} \]

\[ \#y \leq 3 \quad \text{prob} = \frac{1}{2^{30\text{ bits}}} \]

30 room. What's prob all birthdays are different?

\[ 1 \times \frac{364}{365} \times \frac{363}{365} \times \frac{362}{365} \times \cdots \frac{336}{365} \]

\[ \hat{p} \text{ (some 2 people)} = 1 - \frac{336}{365} \]