Lecture 4.1: Hash Functions: Introduction

CS 436/636/736
Spring 2012

Nitesh Saxena
Course Administration

- HW1 being graded
- Solution was emailed
- Expect HW2 by coming Monday
  - Covers mostly Public Key Cryptography
  - Will be due in 12-14 days
Outline of Today’s lecture

- Hash Functions
  - Properties
  - Birthday Paradox
  - Generic Design
Cryptographic Hash Functions

- Requirements of cryptographic hash functions:
  - Can be applied to data of any length.
  - Output is fixed length, usually very short
  - Relatively easy to compute $h(x)$, given $x$
  - Function is deterministic
  - Infeasible to get $x$, given $h(x)$. **One-wayness property**
  - Given $x$, infeasible to find $y$ such that $h(x) = h(y)$. **Weak-collision resistance property**
  - Infeasible to find any pair $x$ and $y$ ($x \neq y$) such that $h(x) = h(y)$. **Strong-collision resistance property or just collision resistance**
Some Applications of Hash Functions

- In general, can be used as a checksum for large amounts of data
- Password hashing
- Digital signatures
- Message authentication codes (will study later)
- Used also in RSA-OAEP, and many other cryptographic constructions
Hash Output Length

- How long should be the output (n bits) of a cryptographic hash function?
- To find collision - randomly select messages and check if hash matches any that we know.
- Throwing k balls in N = 2^n bins. How large should k be, before probability of landing two balls in the same becomes greater than ½?
- **Birthday paradox** - a collision can be found in roughly
  \[ \sqrt{N} = 2^{(n/2)} \] trials for an n bit hash
  - In a group of 23 (~ \( \sqrt{365} \)) people, at least two of them will have the same birthday (with a probability > ½)
- Hence n should be at least 160
Birthday Paradox

- Probability that hash values of $k$ random messages are distinct is (that is, no collisions) is:

$$= \left(1 - \frac{1}{N}\right) \left(1 - \frac{2}{N}\right) \ldots \left(1 - \frac{k-1}{N}\right) = \prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right)$$

$$\approx \prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right) (\text{as for small } x, \ e^{-x} \approx 1 - x, \text{as } e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} \ldots)$$

$$= e^{-k(k-1)/2N}$$

So for at least one collision we have probability of

$$\left(1 - e^{-k(k-1)/2N}\right)$$

whose value is above 0.5 when $k = 1.17\sqrt{N}$
Generic Hash Function – Merkle-Damgard Construction

- This design for $H()$ is collision-resistant given that $h()$ is collision resistant
- Intuitively, this is because there is a avalanche effect – even if the inputs differ in just 1 bit, the outputs will be completely different
- IV is a known public constant
An Illustrative Example from Wikipedia

<table>
<thead>
<tr>
<th>Fox</th>
<th>cryptographic hash function</th>
<th>DFCD 3454 BEEA 788A 751A 696C 24D9 7009 CA99 2D17</th>
</tr>
</thead>
<tbody>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>8FDB 7558 7851 4F32 D1C6 76B1 78a9 0DA4 AEEF 4819</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>FCD3 7FDR 5AF2 C6FF 915F D401 CA99 7D9A 46AF FB45</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>8ACA D682 D688 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C</td>
</tr>
</tbody>
</table>
Further Reading

- Stallings Chapter 11
- HAC Chapter 9