1 Authentication Protocols (2010 p7q13)

Woo & Lam proposed a protocol that would enable a client $A$ to log on to a server $B$ using an authentication service $S$.

\begin{align*}
A & \rightarrow B \quad A \\ B & \rightarrow A \quad N_B \\ A & \rightarrow B \quad \{N_B\}_{K_{AS}} \\ B & \rightarrow S \quad \{A, \{N_B\}_{K_{AS}}\}_{K_{BS}} \\ S & \rightarrow B \quad \{N_B\}_{K_{BS}}
\end{align*}

a. Explain the protocol notation.

b. Explain why the protocol is insecure.

c. How should it be fixed?

2 WEP 802.11

While TLS is a well-regarded, if overly complex, protocol, the Wired Equivalent Protocol for wireless traffic encryption has earned the nickname “how not to design a crypto protocol.” It is regretfully implemented in millions of wireless routers despite catastrophic security flaws.

The basic setup involves a wireless client (such as a laptop) sharing a 40-bit key with the access point. When a connection is made, encryption is done using RC4-64, with the 64-bit key consisting of the 40-bit shared key and a 24-bit IV which is incremented for each packet. Within the packets, a CRC-32 checksum is appended prior to encryption. The value sent for a packet is:

$IV|E_{K_{IV}}\{M|CRC_M\}$

a. Why is a new IV used with each packet? In practice, is 24 bits enough?

b. Comment on the choice of 40-bit keys. Why might this have been chosen? Also, explain why allowing users to choose their own 40-bit hex value is dangerous.
c. Why is the CRC included? Is this appropriate? Explain specifically what attacks are possible due to the CRC, and what an appropriate fix would be.

d. The use of many closely related keys in RC4 is insecure, there are complicated but devastating algebraic attacks on RC4 given reasonable amounts of known plaintext encrypted under related keys. Explain where known plaintext is likely to come from in the WEP case.

e. At the end of the day, how critical is WEP security? Explain what attacks are possible and not possible on a laptop browsing the web through an insecure WEP channel.

3 Hash functions

The three most important properties of hash functions are:

- **One-wayness**: Given $H(x)$, it is infeasible to compute any information about $x$.
- **2nd pre-image resistance**: Given $H(x)$, it is infeasible to find any $y \neq x$ such that $H(x) = H(y)$
- **Collision resistance**: It is infeasible to find any $x, x'$ such that $H(x) = H(x')$

It is unproven whether modern high-speed hash functions achieve all of these properties.

a. Describe the relationship between the three properties. Which properties are implied by the others? Which is the hardest to achieve?

b. Describe a trivial hash function which is one-way, but not pre-image resistant.

c. Keeping the Birthday Paradox in mind, if you need to prevent an adversary who can execute your hash function $2^N$ times from finding collisions, what is the minimum length of the output of your hash-function?

d. Compare the security of a hash with $n$-bit output to a cipher with $n$-bit keys?

4 Protocol Composition

Suppose you’re designing a data transmission protocol over a faulty, insecure, and expensive channel, such as IP. You want to apply the following transforms to all data packets:

- Encryption of the data using AES-CBC to provide confidentiality and integrity
- Compression of the data using gzip to increase bandwidth efficiency
- Calculation of a BCH error-correcting code to ensure reliability of transmission

The operations `Encrypt()`, `Compress()`, and `Error-Correct()` can be applied to outgoing data packets in any of 3! = 6 orders. Is there a proper ordering for the three? Which of these orders will cause problems? Describe specifically if confidentiality, efficiency, and reliability may be broken with improper ordering.