CS 161 Fall 2006 — Computer Security Joseph/Tygar — HW 1 Solutions

Solutions to exercises:

1. **(1 pts.) Any questions?**

   [Any constructive answer will receive full credit.]

2. **(4 pts.) Getting started**

   (a) *[The sentence must be written somewhat near the name.]*

   (b) *[Your words here. This is one way to state it:]*

   Each person will hand in their own writeup. You may consult one another, on the condition that you list your collaborators on your writeup. You may not copy or look at another student's written work.

   (c) *[Registration with the grading system will be verified for credit.]*

   (d) The Cuckoo's Egg, by Cliff Stoll.

3. **(20 pts.) PGP**

   (a) —

   (b) *[Your key's fingerprint goes here.]*

   (c) —

   (d) —

   (e) —

   (f) *[You should have sent this email to your TA. You'll get full credit if your key's fingerprint matches the one given in part (b), the message was encrypted with your TA's key, and the TA's key and message were signed by your key.]*

4. **(45 pts.) Repeated DES**

   (a) If there are 2^b possible key values (assuming the key has b bits), a worst-case exhaustive search for the key of a single encryption needs to check all keys once so it takes 2^b operations. An expected-case exhaustive search will find the key after trying about half of the 2^b possibilities, so it takes about $(2^b)/2$ operations.

   The meet in the middle attack requires a DES operations for each possible key value for encryptions of the plaintext as well as decryptions of the ciphertext, so it requires $2 \cdot 2^b$ operations.

   (b) The attack would first proceed by creating the tables of E(P,K1) and D(C,K3), similar to the double-encryption case. But for the middle encryption step, we need to build the full 2^b-entry table for *each* entry in one of the tables above before comparing. So the time required is the time for double encryption plus the time for encrypting each entry in one table with each key, or $2 \cdot 2^b + (2^b) \cdot (2^b)$. Since $(2^b)^2 = 2^{2b}$, this is roughly equivalent to doubling the number of bits in the key, increasing the work exponentially.

(c) If we do n-encryption with n keys, a known-plaintext attack requires building a 2^b-entry table at each end, then building 2^b-entry tables for each entry in each of those tables, and so on until the two sides meet in the middle. We build from each side rather than from one side to the other because the number of tables grows exponentially with the distance from either end.

For even n, the work required is approximately $2 \cdot (2^b)^{n/2}$, while for odd n, the work is approximately $(2^b)^{(n-1)/2} + (2^b)^{(n+1)/2}$.

More precisely, for even n the work is

$$2 \cdot \sum_{i=1}^{n/2} 2^{b \cdot i},$$

and for odd n the work is

$$\sum_{i=1}^{(n-1)/2} 2^{b \cdot i} + \sum_{i=1}^{(n+1)/2} 2^{b \cdot i}.$$

This has about the same resistance to brute-force searching as using one key of length $b \cdot (n/2)$, rounding $(n/2)$ up for odd n, so this is a way to get more key strength for an algorithm with fixed-length keys like DES. People do actually use this technique with $n = 3$, calling it Triple-DES (or 3-DES).

5. (15 pts.) Access Control

Mandatory access control enables administrators to set rules and policies restricting who has access to data and when. Users cannot change the rules or policies once data has been created. This is appropriate for HIPAA, which is a requirement on all users. Discretionary access control would put compliance with HIPAA in the hands of each creator of a medical record rather than mandating it globally, which would not be acceptable.

6. (15 pts.) Access Control Lists and Capabilities

An ACL-based system is more efficient when you want to grant/revoke access to a resource for many users—you simply update the single ACL. An example of this is making a homework solution set available to all students at once; the change can be made in one place for all users. A capability-based system is more efficient when you want to grant/revoke access to many resources for a single user—you simply update the user's set of capabilities. An example of this is giving each employee a key to their workplace; when an employee leaves, they turn in their key and so their access is revoked without having to change all the locks.