CSE 590YA: Practical Aspects of Modern Cryptography
Winter 2002

Assignment #3

Due in class: January 29

1. Let Alice and Bob each have an RSA public/private key pair and assume each knows the other's public key. Alice and Bob wish to exchange encrypted data using a block cipher that requires a 128-bit secret key (e.g. AES). Design a key exchange protocol that allows Alice and Bob to establish a shared secret of 128 bits in length but prevents Eve the eavesdropper from learning the secret.

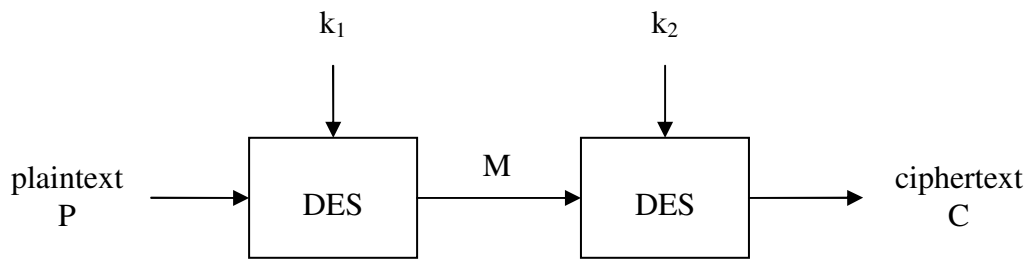
2. Modify the protocol you designed to answer Question 1 so that Alice and Bob can securely communicate with each other using a stream cipher (e.g. RC4) with a 128-bit secret key size instead of a block cipher. (Hint: First explain why Alice and Bob cannot use the same secret key for protecting both directions of communication between them.)

3. The distributed.net project (http://www.distributed.net/) is currently attempting to break via brute-force search a sample ciphertext encrypted with RC5-64, a block cipher that uses a single 64-bit secret key. Their network of loosely-coupled computers is currently able to test approximately 190 gigakeys per second (1.9 x 10^11 keys/second). Assuming that the distributed.net network operates at that constant rate, independent of key size & block cipher algorithm, how long (in years) will it take the network to completely search the keyspace (try every key) to find:

a) a 64-bit RC5 key (the current challenge)?
b) an 80-bit SKIPJACK[1] key?
c) a two-key 3DES key (112 bits of entropy)?
d) a 128-bit Rijndael/AES key?
e) a three-key 3DES key (168 bits of entropy)?
f) a 256-bit Rijndael/AES key?

4. We define the Double-DES algorithm (2DES) as the sequential application of two instances of DES using two distinct DES keys k_1 and k_2. That is, the result of applying 2DES to a plaintext block P_i is the result of calculating $DES(DES(P_i, k_1), k_2)$, where $DES(x, k)$ denotes conventional DES encryption of plaintext x with key k. The following figure shows 2DES graphically:

[1] SKIPJACK is an 80-bit symmetric cipher. It was originally developed by the NSA for use in the CLIPPER chip (the U.S. Government's proposed key escrow system).



Assume that we know two plaintext-ciphertext pairs (P_1,C_1) and (P_2,C_2) where C_i is P_i encrypted with unknown 2DES key pair (k_1, k_2). We could brute-force search for (k_1, k_2) by performing 2^{112} 2DES encryptions, trying each possible key pair sequentially, but it's possible to do much better than that. Show that for any known plaintext-ciphertext pair we can discover a candidate decryption key pair (k'_1,k'_2) in $O(2^{56})$ operations using $O(2^{56})$ space. Show further that if the candidate pair (k'_1,k'_2) also maps P_2 to C_2 then with high probability $(k'_1,k'_2) = (k_1, k_2)$.

5. In this problem we're going to explore the chance of randomly finding a hash collision for SHA1 (two elements x and x' such that SHA1(x) = SHA1(x')) among a large collection of hashes. Recall that the SHA1 hash function generates a 160-bit hash for any sized input. For a given hash value y and random x, the chance that SHA1(x) = y is thus approximately 2^{-160}.

Now, consider a series of N independent values $x_1, x_2, \ldots, x_N$ and their corresponding SHA1 hash values $y_1 = SHA1(x_1), \ldots, y_N = SHA1(x_N)$. What is the chance that $y_1, \ldots, y_N$ are all distinct (no collisions)? When we compute y_j, the chance that it collides with any of the j-1 previous hash values is $(j-1)*2^{-160}$, so the chance it does not collide is $1-((j-1)*2^{-160})$. The series

$$(1 - 0*q) * (1 - q) * (1 - 2*q) * (1 - (N-1)*q), \text{ where } q = 2^{-160}$$

may be approximated by the value $1 - N^2 * 2^{-161}$, so the chance of finding one or more collisions within the hashes of the N independent values is $N^2 * 2^{-161}$.

Compute the approximate chance of finding at least one collision in a database holding N hash values, where

a) N = 286 million (one record per person in the US, e.g. hashed Social Security Number, based on current estimated US population, slightly more than 2^{28})
b) N = 6.2 billion (approximate world population, less than 2^{33}).
c) N = 2^{43} (1000 hashes per person currently alive on Earth).