CSEP 590A: Practical Aspects of Modern Cryptography Winter 2011

Assignment #4

Due: February 3, 2011

1) Assume that two companies A and B have deployed Kerberos-based systems to secure internal communications inside their respective boundaries (e.g. each company maintains its own independent Kerberos realm). Let KDC_A and KDC_B be the central Key Distribution Centers for A and B, respectively.

   Now assume that A and B decide to collaborate on a new product and want to link their two Kerberos networks together so that a Client C_A in company A can get Kerberos tickets to a Server S_B in company B, and similarly a client C_B in company B can get Kerberos tickets to talk to a Server S_A in company A. (You may assume that A and B have linked their computer networks so that every client in either A or B has access to the KDC, TGS and Servers of both companies.) Note that only KDC_A (respectively KDC_B) can authenticate company A (respectively, B) Clients.

   Assume that the two companies have exchanged a shared secret key K_{AB} out-of-band and that this key is known only to each company's KDC. Describe how you would enhance or modify the Kerberos scheme as presented in class in order to support cross-realm trust.

2) In this problem we're going to compare the relative cost of moving to a larger key length in AES vs. RSA. For the purposes of this problem you may assume the following simplifying facts:
   - Your implementation of AES-128 is able to encrypt or decrypt a single block of data (16 bytes) in time t, and your implementation of AES-256 is able to encrypt or decrypt a single block of data (16 bytes) in time $1.4t$. (Why 1.4x? Because AES-128 requires 10 rounds/block and AES-256 requires 14 rounds/block.)
   - Your implementation of RSA-n has the following performance characteristics:
     - A single RSA encryption takes time an^2.
     - A single RSA decryption takes time bn^3.
     - A single RSA key generation step takes time cn^4.

   You have just taken over responsibility for maintaining a system that uses AES-128 and RSA-1024 to secure communication links.

   a) How many AES-128 encryption operations can you perform in the time it takes to do a single RSA-1024 encryption?

b) How many AES-128 decryption operations can you perform in the time it takes to do a single RSA-1024 decryption?

c) Your boss asks you what the performance impact would be if you move from AES-128 to AES-256. Assuming you run the system with AES-256 and RSA-1024, how many AES-256 encryptions can you perform in the time it take to do an RSA-1024 encryption?

d) Now your boss asks you what the performance impact would be to you move to both AES-256 and RSA-2048. How many AES-256 decryption operations can your system perform in the time it takes to do one RSA-2048 decryption?

e) Establishing a single secure communications link in your system requires one RSA key generation, two RSA encryptions and two RSA decryptions. Assume that using RSA-1024 and AES-128, your system can compute 2^{20} AES-128 encryption operations in the time it takes to perform a single RSA-1024 decryption. If you send 16MB of data using AES-128 and RSA-1024, what fraction of the overall time is spent in the RSA operations? What fraction of the overall time would be spent in the RSA operations if you used AES-256 and RSA-2048?

3) Assume that you maintain a computer network where the “root of trust” is a certificate signed with RSA-1024 and SHA-1. You’ve just read a newspaper article reporting on some new cryptography results, specifically that it is now possible to (a) factor a 768-bit RSA modulus, and (b) find an MD5 collision by brute-force, in “reasonable” time. (It doesn’t matter exactly how long “reasonable” is, just a short enough time with that neither RSA-768 nor MD5 are considered secure.)

Assume Moore’s Law continues to hold and the amount of computing power you can access in “reasonable” time doubles every 18 months.

a) If one can find a hash collision in MD5 via a birthday attack today, when would you predict it would be similarly possible to find a hash collision in SHA-1?

b) The following equation approximates the time it takes the best general-purpose factoring algorithm currently known to factor an n-bit composite integer:

$$e^{2(n^{1/3})(\log_2 n)^{2/3}}$$

Which should you do first: migrate to SHA-2, or move from RSA-1024 to RSA-2048?

4) Alice and Bob are cryptographers who live in different countries. Suppose that they meet at a conference and exchange a randomly-generated secret key K face-to-face (an “out-of-band” exchange), which they plan to use to continue communicating securely once they return home. Specifically, each day either Alice or Bob will send a message to the other, and they want to encrypt it with a key only they know. Let $m_1, m_2, m_3, \ldots$ be the sequence of messages Alice and Bob exchange.

Normally, we would expect Alice and Bob to be able to use key K directly to encrypt the $m_i's$, but there’s a problem. Alice lives in a country where her government asserts the right to eavesdrop on all internet traffic and seize her computer at any point in time, which means that on any day all of the information on her computer (including any and all secret keys) could be seized, revealing all the messages (past and present) encrypted with whatever keys are seized.

Design a protocol that Alice and Bob can use to protect the $m_i's$ such that if Alice's computer is seized none of the messages sent prior to seizure are at risk.

5) The SSL and TLS protocols are "session-based" – they enable the establishment of a secure connection between a client and server that can continue as long as the two desire. Establishing a secure connection is far more expensive than using it, so it would be nice if a client that established a secure connection one day could somehow remind a server of the details another day so that they could continue the old session rather than establishing a new one.

   Consider the following modifications to the SSL/TLS protocols. Whenever a session is established, the pre-master secret is used to derive a session identifier that can be retained by the client and server. Both parties can "remember" the pair consisting of the identifier and the original pre-master secret as long as they wish. A client that wishes to continue an old session contacts the server with the identifier and reminds the server of the session's details – including the previously agreed upon cipher suite. If the server has retained this identifier and the associated pre-master secret, the server re-derives the session key and resumes the session using the cipher suite given by the client. Describe how an attacker can exploit this protocol modification to compromise the supposedly secure session between the client and server.