CS 120/CSCI E-177: Introduction to Cryptography

Problem Set 8

Assigned: Dec. 7, 2006 Due: FRI Dec. 15, 2006 (1:10 PM)

Justify all of your answers. See the syllabus for collaboration and lateness policies. You can submit by email to `ciocan@eecs` (please include source files) or by hardcopy to Carol Harlow in MD 343.

Problem 1. (Authentication + Encryption) Let (G_E, E, D) be a private-key encryption scheme secure against chosen plaintext attack and let (G_M, M, V) be a secure message authentication code (i.e. one that is existentially unforgeable under chosen message attack). Alice wants to send Bob a message m in a private and authenticated way. They share a secret key $k = (k_1, k_2)$, where $k_1 \overset{\mathrm{R}}{\leftarrow} G_E$ and $k_2 \overset{\mathrm{R}}{\leftarrow} G_M$.

Consider her sending each of the following as means to this end. For each, say whether you think it is secure or not, and briefly justify your answer. In analyzing these, you should assume that the primitives have the properties guaranteed by their definitions, but no more. For an option to be good it must work for *every* choice of secure encryption scheme and secure authentication scheme. Thus, to show that one is not good, you should find a secure encryption scheme and authentication scheme which make the result insecure. For the cases which are good, you don't need to give a formal proof, just intuition. Out of all the ones you deemed secure, which would you prefer in terms of performance?

Note that this question is considered in Katz–Lindell §4.7, except that they use security against *chosen ciphertext attack* as their notion of security for encryption (both for the initial encryption scheme (G_E, E, D) and the combined scheme), which can result in different answers for identical constructions.

1. $(m, M_{k_2}(E_{k_1}(m)))$
2. $(E_{k_1}(m), E_{k_1}(M_{k_2}(m)))$
3. $E_{k_1}(m, a)$, where a denotes a unique identifier for Alice known to Bob (e.g. her Harvard ID number).
4. $(E_{k_1}(m), M_{k_2}(m, E_{k_1}(m)))$, where both occurrences of $E_{k_1}(m_1)$ refer to the same ciphertext (i.e. use the same randomness and state).
5. $(E_{k_1}(m), M_{k_2}(E_{k_1}(m)))$, where again both occurrences of $E_{k_1}(m)$ refer to the same ciphertext.

Extra credit: Suppose the adversary Eve could determine whether messages corrupted or injected by her are accepted or rejected by Bob. Which of the methods you deemed secure above are still secure?

Problem 2. (Off-line/On-line Signatures) Public-key signatures are quite expensive. The idea of designing off-line/on-line signatures is to split the signing process into two components. The off-line component will prepare some information σ_1 before the message to be signed is known. This component could be a little slow since it is done off-line. The on-line component is performed after the message m arrives. It uses σ_1 together with m and the signing key to produce the "final" signature σ. The on-line signature component should be "fast".

Let (G, S, V) be a standard secure signature scheme, and let (G', S', V') be a secure one-time signature scheme. The signing and verification keys of the off-line/on-line signature scheme will just be a pair of keys $(sk, pk) \overset{\mathrm{R}}{\leftarrow} G(1^n)$ for the standard signature scheme. In the off-line phase, we pick the random one-time keys $(sk', pk') \overset{\mathrm{R}}{\leftarrow} G(1^n;)$ and sign pk' using the standard signing key: $\sigma_1 = S_{sk}(pk')$. In the on-line phase, we use the one-time signature to produce $\sigma_2 \overset{\mathrm{R}}{\leftarrow} S'_{sk'}(m)$. The overall signature is $\sigma = (\sigma_1, \sigma_2, pk')$.

- What is the verification algorithm for the off-line/on-line scheme?
- Prove that the off-line/on-line scheme is secure (i.e. is existentially unforgeable under chosen message attack).

Problem 3. (Merkle Trees)

1. Suppose we have a family of collision-resistant hash functions $\mathcal{H} = \{h_k : \{0,1\}^{2n} \to \{0,1\}^n\}_{k \in \{0,1\}^n}$. One way to build collision-resistant hash functions for longer inputs is to use the Merkle–Damgård construction given in class. Another way is to use a *Merkle tree*, defined as follows. Define $H_k^i : \{0,1\}^{2^i \cdot n} \to \{0,1\}^n$ by setting $H_k^0(x) = x$ and and $H_k^{i+1}(x \circ y) = h_k(H_k^i(x) \circ H_k^i(y))$, where $\|x\| = \|y\|$ and $\circ$ denotes concatenation. Show that for $i = O(\log n)$, the family $\mathcal{H}^i = \{H_k^i : \{0,1\}^{2^i \cdot n} \to \{0,1\}^n\}$ is collision-resistant.

2. Suppose you want to be able to detect that no one has tampered with your hard disk D when you are on vacation. One way to accomplish this is to apply a collision-resistant hash function H_k to your entire hard disk D, take both the key k and the hash value $z = H_k(D)$ with you (e.g. as a printout or on a USB disk), and check that the hash value is consistent upon return. Then, even if an adversary learns k and z, it will be infeasible for the adversary to modify your disk into D' such that $H_k(D') = z$ (by collision resistance). Suppose however, you do not want to spend the time to verify the entire disk upon return, but just a small file (say n bits long). Describe how the verification process can be made more efficient by using H_k constructed from a Merkle tree. (You still only need to bring k and $H_k(D)$ with you on vacation, but you can store more information insecurely at your home computer.)