Cryptography (SS2006) Prof. Michael Backes

Exercise Sheet 9

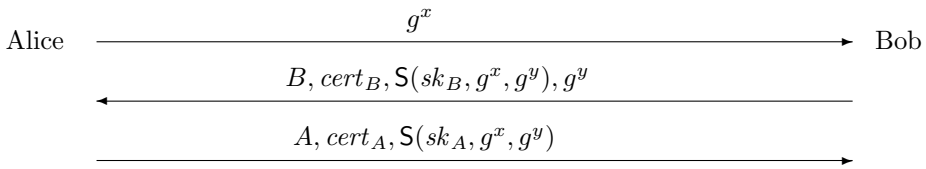
Out: June 27, 2006 Saarland University

Problem 1: Certificate Revocation Trees

Consider certificate revocation trees as introduced in the lecture. Prove that a verification authority (VA) cannot trick a user to belief that a certificate is still valid although it was revoked, provided that the hash function used in construct the tree is collision-resistant. Describe exactly which information needs to be included in the proof that the VA sends to the user, i.e., the exact message the VA has to send to the user in general. Note that in the lecture notes, we only gave the construction for a specific example, and for the sake of readability we did not outspecify it. For example, one had to equip each intermediate value with a flag telling if the value was input to the left or the right argument of the hash function.

Problem 2: Authenticated Diffie-Hellman

We consider a natural protocol for authenticated Diffie-Hellman key exchange. The goal is to provide mutual authentication with key exchange. We assume that each party has a private signing key for some signature scheme and a certificate on the corresponding public key. The protocol proceeds as follows:



Finally, both Alice and Bob can compute the shared secret $K = g^{xy}$ from which the parties derive a session-key to encrypt and MAC all traffic between A and B.

(a) Briefly explain the purpose of the signatures in the protocol above. What standard attack do they defend against?

(b) Show that an active attacker, Eve, can interfere with the protocol by a man-in-the middle attack so that at the end we have the following:

- A thinks that she is communicating securely with B (as required), but
- B thinks he is communicating securely with Eve.

In other words, B is fooled into thinking that the encrypted messages he is receiving (from A) are coming from Eve. Note that Eve cannot eavesdrop on the resulting encrypted channel. Briefly explain why your man-in-the-middle attack results in the confusion described above.

(c) Briefly describe a hypothetical example of how Eve can use this attack to steal money from A. For example, suppose A makes money by giving expert advice in a private chat room run by B.

Problem 3: Offline Signatures

One approach to speeding up signature generation is to perform the bulk of the work offline, before the message to sign is known. Then, once the message m is given, generating the signature on m should be very fast. Our goal is to design a signature system with this property.

(a) Does the RSA Full-Domain-Hash (FDH) signature system enable this form of offline signatures? In other words, can we substantially speed-up RSA signature generation if we are allowed to perform offline computation before the message m is given?

(b) Our goal is to show that any signature system can be converted into a signature where the bulk of the signing work can be done offline. Let $(\mathsf{Gen}, \mathsf{S}, \mathsf{V})$ be a digital signature scheme (such as RSA FDH) and let G be a group of order q where discrete logarithm is hard. Consider the following modified signature system $(\mathsf{Gen}', \mathsf{S}', \mathsf{V}')$:

- The algorithm Gen' runs algorithm Gen to obtain keys pk, sk. It also picks a random group element $g \in G$ and a random $x \leftarrow_{\mathcal{R}} \{1, \ldots, q\}$ and sets $h = g^x$. Let $pk' := (pk, g, h)$ and $sk' := (sk, g, h, x)$ be the public and the secret key, respectively.
- The algorithm $\mathsf{S}'((sk, g, h, x), m')$ picks a random $r \in \{1, \ldots, q\}$, computes $m := g^{m'} h^r \in G$, and then runs $sig \leftarrow \mathsf{S}(sk, m)$. It outputs the signature $sig' := (sig, r)$.
- The algorithm $\mathsf{V}'(pk, m', (sig, r))$ computes $m := g^{m'} h^r \in G$ and outputs the result of $\mathsf{V}(pk, m, sig)$.

Show that the bulk of the work in the S' algorithm can be done before the message is given.

Hint: First, observe that since x is part of the secret key sk', the signer can compute $m = g^{m'} h^r$ as $m = g^{m'+xr}$. Now, offline, try running $\mathsf{S}(sk, m)$ on a message $m = g^s$ for a randomly chosen $s \leftarrow_{\mathcal{R}} \{1, \ldots, q\}$. Let sig be the resulting signature. Then, once the message m' is given, show that the signer can easily convert sig into a valid signature sig' for m' using only one addition and one multiplication modulo q.

(c) Prove that the modified signature scheme is secure. In other words, show that an existential forger under a chosen-message attack on the modified scheme gives an existential forger on the underlying scheme. You may use the fact that $\mathsf{H}(m, r) = g^m h^r$ is a collision resistant hash function and hence the adversary cannot find collisions for it.