

HW #4: Hash Functions, Message Authentication Code and Key Distribution

CS 392/6813: Computer Security
Fall 2007

[100pts] DUE 10/08/2006 (midnight)

Problem 1 [30pts]

Assume that the US social security number (SSN) assignment is performed using the function which on input of a resident r , outputs its SSN as a 9-digit long number uniformly distributed at random. How many US residents would we have to collect such that at least two have of them will have the same SSN, with a probability greater than 0.8? Show all steps involved. Does this represent a good way for SSN assignment? Why or why not?

Problem 2 [5+5+5+5pts]

- (1) Schnorr signature scheme is described “some questions” portions of the slides. Alice needs to sign (using Schnorr signature) two messages m_1, m_2 . She chooses a random number k_1 in Z_q , computes $r_1 = g^{k_1} \bmod p$, $c_1 = H(m_1, r_1)$, and $s_1 = k_1 + c_1x \pmod{q}$, and outputs (r_1, s_1) . Alice becomes lazy and uses the same random number k_1 (as random number generation is costly) while signing the message m_2 . She computes $c_2 = H(m_2, r_1)$, and $s_2 = k_1 + c_2x \pmod{q}$, and outputs (r_2, s_2) . Is this secure? Explain why or why not. Note that (p, q, g) are discrete-log parameters and are publicly known. You don't need to worry about what $H()$ function is, just assume that it is a random function.
- (2) Bob downloaded a 30GB tar.gz file from Alice's server today. Bob needs to know if he downloaded the correct file and that there were no errors in the transmission. Unfortunately, Alice and Bob do not share any keys nor do they have a common CA. How can Bob ensure the correctness of the file? Alice and Bob know each other personally, and they are scheduled to meet in a couple of days.
- (3) Is $H(m_1 \text{ xor } m_2) = H(m_1) \text{ xor } H(m_2)$? Why or why not? Assume $H()$ is MD-5.
- (4) Does HMAC exhibit a complementation property, i.e., does $h = \text{HMAC}(K, m) \rightarrow h^c = \text{HMAC}(K^c, m^c)$? Why or why not? Assume $H()$ is MD-5.

Problem 3 [30pts]

In this exercise, you will get some hands on experience with hashing, signing/verification using RSA and hashing.

- Use the same code that you used in the last exercise: downloadable from: <http://www.funet.fi/pub/crypt/cryptography/asymmetric/rsa/rsaref2.tar.gz>. Get familiarized with MD5 hashing function, signing, and verification functions and the sample code.
 1. execute the key generation function to generate the public key (e,n) and private key d
 2. choose any large message M and execute the signing function to sign M using (e,n), and obtain the signature S
 - compute the execution time for MD5 hashing
 - compute the execution time for signing (this includes hashing)
 3. execute the verification function to verify (M, S) using (e,n).
 - compute the execution time (this includes hashing too)

Repeat (2) and (3) an appropriate number of times and give the average execution time for each cases above. To get the timing for MD5, you need to iterate a large number of times to be able to record something. List the type and speed of the processor, and the memory (RAM) of the machine you execute the code on. Please turn in your modified source code with a small “readme”.

Problem 4 [4+4+4+4+4=20pts]

We discussed public key cryptography, and “certification by a trusted authority” as a means of key distribution in public key cryptography. There is also an emerging paradigm called identity-based cryptography, which works as follows. Each user sets its public key as its “identity” ID (such as an email address); obtains a signature of a trusted authority on its public key and set this signature as its private key. The users can use the public key and private key pairs for secure communication with each other (everyone needs to know the public key of the trusted authority to do so, as in the public-key cryptography).

Compare the identity-based cryptography with the public-key cryptography (say which one is better and explain why), in terms of the following:

1. Security of the “joining” process (which is the process of obtaining a certificate in public key crypto, and a signature in identity-based crypto)
2. Cost of key generation on the user
3. Level of trust on the trusted authority
4. Usability of secure communication (both for confidentiality and authentication)
5. Revocation

