Introduction to Cryptography
Solution to Maman 16

Yehuda Lindell

January 6, 2008

Solution 1: Each party generates an asymmetric encryption key-pair, signs on the public encryption key and sends it to the other party. Upon receiving a signed public-key, each party verifies the signature and if it is valid, accepts the key as belonging to the other party. Note that an active adversary cannot make one of the parties accept an encryption key for which it knows the associated decryption key because this would involve forging a signature.

Another option is to run something called "authenticated Diffie-Hellman key exchange". In this protocol, the parties run the basic Diffie-Hellman key exchange protocol, but also sign on the messages they send. This protocols form the basis for the key-exchange method used in IPsec (a popular standard for securing communication between servers).

Solution 2: *solution omitted.*

Solution 3:

1. In the attack on RSA where the adversary first chooses the signature and then computes the message by "encrypting" with RSA, the adversary cannot make the message be reasonable because it essentially has no control over it; the message is derived via the RSA encryption function which is very unlikely to yield something that "makes sense". Similarly, a multiplicative attack can only work if the attacker has two signed messages with the property that their product is a message that makes sense. This is unlikely...

2. Recall that the proof of security for the Rabin encryption scheme shows that if you can decrypt Rabin (or equivalent if you can find square roots) then you can factor the modulus n into its prime factors. Thus, if Rabin signatures are used and an attacker can ask for signatures as it pleases, this means that it can ask for square roots. As in the proof of security, this can be used to factor n into its prime factors. Note that once this is achieved, the attacker can sign on any message (because the secret key is the prime factors and these are now known to the attacker).

Solution 4:

1. This scheme *is* secure assuming that RSA encryption is secure. However, the security of Shamir's secret sharing scheme is *unconditional*, meaning that we have a mathematical proof that it cannot be broken (somewhat like perfect secrecy). Thus it does *not* have the same level of security has Shamir's scheme.

2. Yes, this scheme is secure enough to be used.

3. There is an important advantage here being that it is not possible to lie about the value of a share. Specifically, after decrypting it is possible for all to re-encrypt and check that the decryption value given was valid. This is in contrast to Shamir's scheme where nothing stops a malicious party from lying about its share.

4. This scheme is highly inefficient. For example, if w is large (say $w = 100$) and t is not small (say $t = 50$), then the number of ciphertexts that need to be distributed is $\binom{100}{50} > \frac{2^{100}}{100}$. This is clearly infeasible. Note, that even for smaller numbers, it becomes very large. For example, take $w = 30$ and $t = 15$. Then the number of ciphertexts is greater than 35 million.

Solution 5: Recall that a monotone circuit is one that has only OR and AND gates; there aren't any NOT gates. Now, let $x = x_1 \dots x_n$ and $y = y_1 \dots y_n$ such that for every i for which $x_i = 1$ it holds that $y_i = 1$. We prove that if $C(x) = 1$ then it must hold that $C(y) = 1$. We do this by induction on the gates $g_1, \dots, g_m$ where each g_i is either AND or OR, and if $i < j$ then g_i can be evaluation before g_j (when evaluating in the natural way from the input to the output). Specifically, we show that if the output of any gate is 1 when the circuit is evaluated on input x then its output must also be one when the circuit is evaluated on input y. The output gate is g_m. Thus, if we show the above (for any gate) it also holds for the output gate, as required.

For the base case, note that g_1 is evaluated directly on input bits. Now if $g_1 = AND$ then all of its input bits must be 1 in x. By the assumption on y, this means that all of g_1's input bits from y are also 1 and so the output of g_1 on y is 1. Likewise, if $g_1 = OR$ then at least one of the input bits equals 1 in x and thus also in y. Assume that this property holds for $g_1, \dots, g_i$; we show that it holds for g_{i+1}. Now, the inputs of g_{i+1} can be from outputs of $g_1, \dots, g_i$ and bits from the input. For the inputs, we have that whatever is 1 with input x is also 1 with input y. By the inductive assumption, this also holds for the outputs of $g_1, \dots, g_i$. Given this, the fact that g_{i+1} has output 1 when the input is y if it had output 1 when the input was x follows from the exact reasoning as the base case. This concludes the proof.

This property is important for access structures because each bit represents a party which is either in the set or not in the set (i.e., $x_i = 1$ if and only if the ith party is in the set). Now, any superset of an authorized set must also be authorized (because the subset can already find the secret so the superset definitely can). In terms of inputs and circuits, a superset of $x = x_1 \dots x_n$ is represented by a vector y for which when any $x_i = 1$ it also holds that $y_i = 1$.

Solution 6: *Technical calculation and thus omitted.*