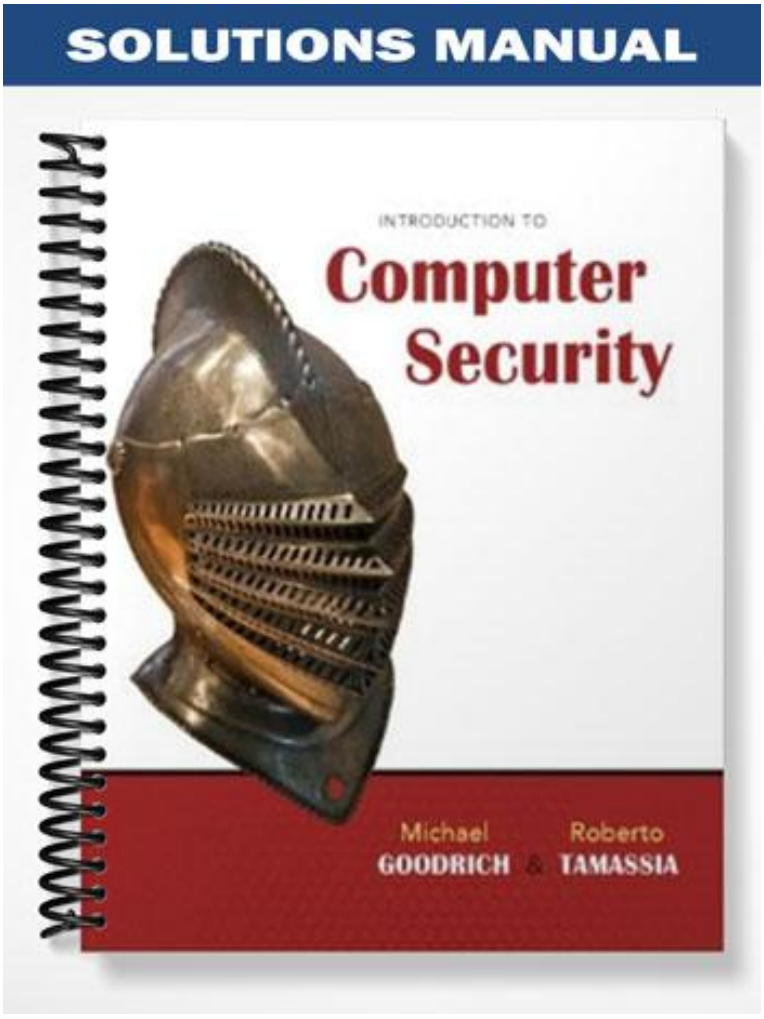
SOLUTIONS MANUAL



Solutions Manual
Introduction to Computer Security
Version 1.1

M. T. Goodrich and R. Tamassia

December 20, 2010

Terms of Use

This manual contains solutions for selected exercises in the book Introduction to Computer Security, by Michael T. Goodrich and Roberto Tamassia, published by Addison Wesley. It is intended for use by instructors adopting the book in a course. Please contact the authors if you find errors in the solutions.

You can make available to your students the solutions for the exercises assigned in your course on hardcopy handouts or web pages that are password-protected or accessible only from your institution's domain.

You are not allowed to make the solutions available on a publicly accessible Web site

Chapter 1

Reinforcement

Problem R-1.12

Compare and contrast symmetric encryption with public-key encryption, including the strengths and weaknesses of each.

Solution Scalability: with public-key encryption, multiple users can send encrypted messages to Alice using her public key and these messages can be decrypted only by Alice; thus, a linear number of public-private key pairs need to be established, distributed and protected to allow pairwise confidential communication between any two users; instead, symmetric encryption requires a quadratic number of secret keys. Efficiency: existing symmetric encryption methods are much faster and use much shorter keys than existing public-key encryption methods. Usability: symmetric-key encryption is easier to understand by an non-expert than public-key encryption.

Problem R-1.14

Suppose the author of an online banking software system has programmed in a secret feature so that program emails him the account information for any account whose balance has just gone over $10,000. What kind of attack is this and what are some of its risks?

Solution This is a Trojan horse, since it has a hidden malicious action that goes with a useful service.

Problem R-1.16

Give an example of the false sense of security that can come from using the "security by obscurity" approach.

Solution There are many examples. One possibility would be to use a weak encryption algorithm, like the Caesar cipher and try to keep secret the type of algorithm that you are using, in addition to keeping the key secret. The problem with this approach is that if someone guesses you are using such an algorithm or is able to reverse engineering your software, then they will discover your algorithm. From there it is a simple matter to break your weak encryption scheme.

Problem R-1.17

The English language has an information content of about 1.25 bits per character. Thus, when using the standard 8-bit ASCII encoding, about 6.75 bits per character are redundant. Compute the probability that a random array of t bytes corresponds to English text.

Solution Since each byte has 8 bits, the total number of t-byte arrays is $T = \left(2^8\right)^t = 2^{8t}$. Given that the information content of English text is 1.25 bits per character, the number of t-byte arrays corresponding to English text is $E = \left(2^{1.25}\right)^t = 2^{1.25t}$. Thus, the probability that a random array of t bytes corresponds to English text is given by $E/T = 2^{-6.75t}$.

Problem R-1.18

Suppose that a symmetric cryptosystem with 32-bit key length is used to encrypt messages written in English and encoded in ASCII. Given that keys are short, an attacker is using a brute-force exhaustive search method to decrypt a ciphertext of t bytes. Estimate the probability of uniquely recovering the plaintext corresponding to the ciphertext for the following values of t: 8, 64, and 512.

Solution Brute-force decryption generates 2^{32} candidate plaintexts, one for each possible key value. Each plaintext has probability $2^{-6.75t}$ of being English text. Thus, the attack is expected to produce $2^{32-6.75t}$ candidate English plaintexts. Since this number is less than one for the given values of t, the attack is expected to always recover the plaintext.

Problem R-1.19

Suppose you could use all 128 characters in the ASCII character set in a password. What is the number of 8-character passwords that could be constructed from such a character set? How long, on average, would it take an attacker to guess such a password if he could test a password every nanosecond?

Solution There are 128^8 possible passwords with 8 ASCII characters. Guessing a password will take on average $\frac{1}{2}128^8 10^{-9}$ seconds. This is $9,223,372,037$ seconds or about 417 days.

Creativity

Problem C-1.2

Describe an instance of a file that contains evidence of its own integrity and authenticity.

Solution Take a file and concatenate a digital signature on that file from the owner of that file or from another trusted authority. Don't forget to include the digital certificate of the signer.

Problem C-1.3

Suppose an Internet service provider (ISP) has a voice over IP (VoIP) telephone system that it manages and sells. Suppose further that this ISP is deliberately dropping 25% of the packets used in its competitors VoIP system when those packets are going through this ISP's routers. Describe how a user could discover that his ISP is doing this.

Solution Suppose the user bought both VoIP solutions. He could then do a set of simple end-to-end performance tests to see if one had degraded throughput with respect to the other in terms of packet delivery. If, say, in 10 tests, one is 25% worse than the other, then it is highly likely that this is due to a deliberate packet dropping strategy on the part of the ISP.

Problem C-1.5

Suppose that you are a computer virus writer; hence, you know that you need to store a copy of the code for your virus inside the virus itself. Moreover, suppose you know that a security administrator is also aware of this fact and will be using it to detect the presence of your virus in operating systems files, as described in the previous problem. Explain how you can hide the embedded copy of your virus so that it is difficult for the security administrator to find it.

Solution If the embedded virus code is stored in encrypted form and only decrypted just before it is replicated in another operating system file, then it would be difficult to see the repeated pattern when the virus is at rest inside the infected file.

Problem C-1.9

Benny is a thief who tried to break into an Automated Teller Machine (ATM) using a screwdriver, but was only able to break five different keys on the numeric keypad and jam the card reader, at which point he heard Alice coming, so he hid. Alice walked up, put in her ATM card, successfully entered her 4-digit PIN, and took some cash. But she was not able to get her card back, so she drove off to find help. Benny then went back to the ATM, and started entering numbers to try to discover Alice's PIN and steal money from her account. What is the worst-case number of PINs that Benny has to enter before correctly discovering Alice's PIN?

Solution Since Benny broke 5 different keys and Alice was still able to enter her PIN, it must only use the 5 remaining keys. So the total number of possible keys is now $5^4 = 625$. In the worst case, Benny will have to enter 625 before he enters the correct one.

Problem C-1.10

As soon as Barack took office, he decided to embrace modern technology by communicating with cabinet members over the Internet using a device that supports cryptographic protocols. In a first attempt, Barack exchanges with Tim brief text messages, encrypted with public-key cryptography, to decide the exact amounts of bailout money to give to the largest 10 banks in the country. Let p_B and p_T be the public keys of Barack and Tim, respectively. A message m sent by Barack to Tim is transmitted as $E_{p_T}(m)$ and the reply r from Tim to Barack is transmitted as $E_{p_B}(r)$. The attacker can eavesdrop the communication and knows the following information:

- Public keys p_B and p_T and the encryption algorithm, such that there is exactly one ciphertext for each plaintext.
- The total amount of bailout money authorized by congress is \$900B
- The names of the largest 10 banks
- The amount each bank will get is a multiple of \$1B
- Messages and replies are terse exchanges of the following form:

  Barack: How much to Citibank?
  Tim: \$144B.

Barack: How much to Bank of America?
Tim: $201B.
...

Describe how the attacker can learn the bailout amount for each bank even if he cannot derive the private keys.

Solution The attacker performs a dictionary attack. Since the message format is fixed and there are 10 possible banks and 900 possible bailout amounts, the attacker encrypts the 10 candidate messages from Barack (one for each bank) using public key p_B, and the 900 candidate responses from Tim (one for each bailout amount), using public key p_T. The attacker then matches the ciphertexts exchanged by Barack and Tim with the precomputed ones and determines the corresponding plaintexts. Note that the attacker does not need access to the private keys used by Barack and Tim.

Problem C-1.11

As a result of the above attack, Barack decides to modify the protocol of Exercise C-1.10 for exchanging messages. Describe two simple modifications of the protocol that are not subject to the above attack. The first one should use random numbers and the second one should use symmetric encryption.

Solution In the first case, Barack can add a random value with b bits to his message, which increases the number of possible messages by a factor of 2^b. In the second case, Barack can first encrypt a (random) key K for a symmetric encryption scheme, and then send the encrypted version of K along with an encryption of his actual message using key K and the symmetric cryptosystem .

Problem C-1.12

Barack often sends funny jokes to Hillary. He does not care about confidentiality of these messages but wants to get credit for the jokes and prevent Bill from claiming authorship of or modifying them. How can this be achieved using public-key cryptography?

Solution Barack digitally signs his jokes and sends each joke together with its signature.

Problem C-1.13

As public-key cryptography is computationally intensive and drains the battery of Barack's device, he comes up with an alternative approach. First, he shares a secret key k with Hillary but not with Bill. Next, together with a joke x, he sends over the value $d = h(k||x)$, where h is a cryptographic hash function. Does value d provide assurance to Hillary that Barack is the author of x and that x was not modified by Bill? Justify your answer.

Solution Value d is a message authentication code (MAC), which gives Hillary assurance of the authorship and integrity of Barack's jokes. The reason is that a cryptographic hash function is one-way, Bill cannot recover the key k from value d, Thus, Hillary knows that only Barack could have computed value d from joke x. Also, if Bill replaces joke x with a joke of his, x', it would infeasible for Bill to compute the MAC value corresponding to x'.

Problem C-1.14

Barack periodically comes up with brilliant ideas to stop the financial crisis, provide health care to every citizen, and save the polar bears. He wants to share these ideas with all the cabinet members but also get credit for the ideas. Extending the above approach, he shares a secret key k with all the cabinet members. Next, he broadcasts each idea z followed by value $h(k||z)$. Does this approach work or can Tim claim that he came up with the ideas instead of Barack? Justify your answer.

Solution Barack is using a message authentication code (MAC). However, since all cabinet members know the secret key, k, they infer that the idea z came from either Barack or one of them. However, it is impossible to determine exactly the individual in this group who had idea z. Indeed, any of the people who know key k can compute the MAC value of an idea and claim ownership of the idea. Thus, Tim can claim he is the one who came up with the ideas.

Problem C-1.15

Describe a method that allows a client to authenticate multiple times to a server with the following requirements:

1. The client and server use constant space for authentication.
2. Every time the client authenticates to the server, a different random value for authentication is used (for example, if you have n authentication rounds, the client and the server have to use n different random values—this means that sharing a key initially and using it for every round of authentication is not a valid solution).

Can you find any vulnerabilities for this protocol?

Solution One solution is to have the client and server use public-key cryptography for authentication and then send the random messages as a way of issuing challenge-responses, say, by having the client send the server a digitally-signed copy of the random string. The problem with this approach is that it is subject to replay attacks. Someone listening in to this communication could replay it to the server and then be authenticated as the original client.

Problem C-1.16

Consider the following method that establishes a secret session key k for use by Alice and Bob. Alice and Bob already share a secret key K_{AB} for a symmetric cryptosystem.

1. Alice sends a random value N_A to Bob along with her id, A.
2. Bob sends encrypted message $E_{K_{AB}}(N_A), N_B$ to Alice, where N_B is a random value chosen by Bob.
3. Alice sends back $E_{K_{AB}}(N_B)$.
4. Bob generates session key k and sends $E_{K_{AB}}(k)$ to Alice.
5. Now Alice and Bob exchange messages encrypted with the new session key k.

Suppose that the random values and the keys have the same number of bits. Describe a possible attack for this authentication method.

Can we make the method more secure by lifting the assumption that the random values and the keys have the same number of bits? Explain.

Solution In the first three steps, the attacker, Eve, observes random values N_A and N_B and their ciphertexts, $E_{K_{AB}}(N_A)$ and $E_{K_{AB}}(N_B)$, computed by Alice and Bob. In the fourth step, Eve replaces message $E_{K_{AB}}(k)$ sent by Bob to Alice with $E_{K_{AB}}(N_A)$ (or $E_{K_{AB}}(N_B)$). Thus, Eve induces Alice to use N_A (or N_B) as the session key, which is known to Eve. When Alice sends a message to Bob using session key N_A (or N_B), Eve can decrypt it.

This attack does not work when the key has length different from the random values as Alice can check the length of the key.

Problem C-1.17

Alice and Bob shared an n-bit secret key some time ago. Now they are no longer sure they still have the same key. Thus, they use the following method to communicate with each other over an insecure channel to verify that the key K_A held by Alice is the same as the key K_B held by Bob. Their goal is to prevent an attacker from learning the secret key.

1. Alice generates a random n-bit value R.
2. Alice computes $X = K_A \oplus R$, where $\oplus$ denotes the exclusive-or boolean function, and sends X to Bob.
3. Bob computes $Y = K_B \oplus X$ and sends Y to Alice.
4. Alice compares R and Y. If $R = Y$, she concludes that $K_A = K_B$, that is, she and Bob have indeed the same secret key.

Show how an attacker eavesdropping the channel can gain possession of the shared secret key.

Solution The attacker eavesdrops X and Y. The attacker recovers key K_B by computing $X \oplus Y = X \oplus (K_B \oplus X) = (X \oplus X) \oplus K_B = K_B$.

Problem C-1.18

Many Internet browsers "lock the lock" on an encrypted web site so long as the digital certificate offered for this site matches the name for this web server. Explain how this could lead to a false sense of security in the case of a phishing attack.

Solution The name for the phishing website needs to match the digital certificate but could be different from the name of the legitimate website being spoofed. Thus, the browser can display a closed lock for a phishing website.

Problem C-1.20

Describe a good solution to the problem of having a group of students collaborate on a software construction project using the directory of one of the group members in such a way that it would be difficult for nonmembers to discover and would not require the help from

a system administrator, assuming that the only access rights the group leader can modify are those for "everyone." You may assume that access rights for directories are "read," "write," and "exec," where "read" means the files and subdirectories in that directory can be listed, "write" means members of that directory can be inserted, deleted, or renamed, and "exec" on a directory or subdirectory means the user can change his location to that directory or subdirectory so long as he specifies its exact name.

Solution Create a directory, `foo`, in the home directory, which does not have read access rights for everyone, but has exec rights for everyone. Then put a subdirectory inside `foo` with a random name. Alternatively, its name could be the output of a cryptographic hash of each day's date and a key that is shared by all the team members. So the name of the subdirectory is either random or pseudo-random and is known only to the team members, and no outsider could easily guess this name. Make the access rights for this subdirectory be read, write, and exec for everyone. Since the `foo` directory is not readable, no outsiders can list its contents and see the name of this important subdirectory. But since the team members all know its name, they can change their location to that directory and do their work.

Problem C-1.22

Suppose, in a scenario based on a true story, a network computer virus is designed so as soon as it is copied onto a computer, X, it simply copies itself to six of X's neighboring computers, each time using a random file name, so as to evade detection. The virus itself does no other harm, in that it doesn't read any other files and it doesn't delete or modify any other files either. What harm would be done by such a virus and how would it be detected?

Solution This problem is based on the true story of the Cornell graduate student, Robert Morris. His virus (which is more properly called a "worm") brought the entire Internet to its knees. The reason is that when the virus copies itself to X's neighbors, they will copy it back six times to X, which then copies 36 copies to its neighbors, and so on. Soon all of X's disk memory is full of copies of the virus and X crashes. So this is a type of denial of service attack.