CS396: Security, Privacy and Society (Fall 2022)
Homework #1, October 4, 2022

Instructor: Abrar Alrumayh

Instructions

Please carefully read the following guidelines on how to complete and submit your solutions.

1. The homework is **due on Tuesday, October 25, 2022, at 11:59pm**. Late submissions are accepted subject to the policy specified in the course syllabus. Starting early always helps!
2. Solutions are accepted only via Canvas, where all relevant files should be submitted **as a single .zip archive**. This should include your typed answers **as a .pdf file** and **the source code** of any programming possibly used in your solutions.
3. Unless otherwise specified, for any assignment involving programming you may use any programming language of your choice. If asked, you should be able to explain details in your source code (e.g., related to the design of your program and its implementation).
4. You are bound by the Stevens Honor System. For Homework #1, you may work either **by yourself or in pairs**—in this case, your team **names should appear clearly** on your hand-in, and **2 same hand-ins are required**. Any collaboration beyond this is not allowed. You may use any sources related to course materials, but information from external sources must be properly cited. Your submission acknowledges that you have abided by this policy.

Problem 1: Shared or forgotten keys? (15%)

Long ago, Alice and Bob shared an n-bit secret key but now they are no longer sure they still possess the same key. To verify that the key k_a currently held by Alice is the same as the key k_b currently held by Bob, they need to communicate over an insecure channel.

(1) Which two basic security properties should be considered in the design of a secure protocol for solving the above problem and why these properties become relevant in this setting?

(2) Suppose that Alice and Bob use the following protocol to check if they store the same secret.

1. Alice generates a random n-bit value r.
2. Alice computes $x = k_a \oplus r$, 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, Alice and Bob share a secret key.

Does the above protocol satisfy the two security properties identified in question (1)?

Problem 2: Perfect or imperfect ciphers? (15%)

(1) Assume that an attacker knows that a user's password is either $p_1 =$ `abcd` or $p_2 =$ `bedg`. Say the user encrypts his password using the Vigenére cipher, and the attacker sees the resulting ciphertext c. Show how the attacker can determine the user's password, or explain why this is not possible, when the period t used by cipher is 1, 2, 3, or 4 respectively.

(2) Show that the mono-alphabetic substitution cipher is trivial to break when the attacker launches a chosen-plaintext attack. How much chosen plaintext is needed to recover the entire secret key? What is the shortest chosen single-message plaintext that you can find, which is a valid English message and would successfully recover the key? Finally, under which conditions, and why, is the mono-alphabetic substitution cipher perfectly secure (against a ciphertext-only attacker)?

Problem 3: Crypt-analyze this! (40%)

I just discovered that two of my TAs, Alice and Bob, have been secretly communicating with each other in our common group chat that we use for course matters. I often see unintelligible short texts on my screen to which I didn't pay attention, but now I suspect they plan behind my back. I am pretty sure they make use of one-time pad encryption with the following parameters: The message space consists of English messages which are 33 characters long, where only letters (of either case), spaces and possibly punctuation marks are used. Please help me break their code!

(1) Below are eleven ciphertexts (in hex format) that they exchanged just minutes ago today... (Tuesday, October 4, 2022):

000d16251c07044b36171c0307280858291403500a2003450029001e5930070e52
0d0d15713c49000a2c521d120f224f0125004d00163d100011380d0359330a0b4d
151f00221a04064b2d1c0a571a2f021d6a050c145326054505231311102a084701
0d091c71020c4308231c4f1a0f2d0a582c0003501c295624103e0008592a001001
1d480d3e050c43052d521c031b220a163e550e111d6f04001328410e112d1c4701
00000425551e0c1e2e164f150b661e0d2301085016221404003e00090a2d010001
181d063a1c051a4b0d263f5707354f082f070b15103b1a1c523f04190b211b4701
1001013f0149220930131d571d2716583e1d0802166f0104016c005a1a251b0449
19091c3310491a0e365226570a2f0b163e551d110a6f171106290f0e102b014701
030d45221d06160726521d120f2a03016a190403072a18450623413b1b360e1501
1a090d71020c430a30174f13012f011f6a02081c1f6f010c06240e0f0d64070253

Write down the 11 plaintext messages that were exchanged. You may write a program that will help you with your cryptanalysis. In designing your program, remember that most likely spaces will be among the most frequent characters in the plaintexts, and carefully observe what their role may be in the mapping from plaintexts to ciphertexts. Explain what your cryptanalysis strategy is and what algorithm your program implements.

(2) To ease key management, my TAs change their shared key only every midnight. But rather than randomly generating (and securely exchanging) a new key every midnight, the TAs created an algorithm to automatically generate the new key pseudorandomly using the current (i.e., previous day's) key as input. That is, they replace the current key k_i with the new key $k_{i+1} = \mathsf{SHA}_{256}(k_i)\|00100001$, where $\|$ denotes concatenation. What is the key the TAs will be using the day that this homework is due?

Essay 4: One more Crypto controversy... (40%)

The so-called Dual_EC_DRBG pseudorandom generator (PRG) operates in the following simplified manner in order to incrementally generate blocks of pseudorandom bits $r_1, r_2, \ldots$:

- The PRG is initiated by randomly selecting two (2-dim) points P, Q in a given elliptic curve over a given prime field size p, so that for any integer t the points P^t, Q^t are well-defined.
- Starting from an initial random seed s_0 in order to generate the k-th pseudorandom block r_k:
  - the PRG's internal secret state s_k is updated to the x-coordinate of point $P^{s_{k-1}}$; and
  - the PRG's k-th output r_k is the x-coordinate of point $Q^{s_{k-1}}$, appropriately truncated to a smaller bit-string.

Yet, if the points P, Q are known to be related in the form of $Q^e = P$, or if the output truncation rate is more than 1/2, then this PRG is known to be insecure—that is, a brute-force type of attack is likely to reveal the PRG's internal state s_k. The rest is history...

Read about the Dual_EC_DRBG design, standardization, implementation, adoption and abandonment from its Wikipedia entry and Matt Green's blog entry, and answer the following questions.

1. Describe briefly the controversy related to Dual_EC_DRBG, by identifying various main stakeholders (organizations or companies rather than individuals), their involvement in the events, and their possibly conflicted goals.
2. Describe how one or more broad ethical concerns occur in the issue at hand, by clearly articulating what these concerns may be and how they are possible impacted by different choices or related tradeoffs.
3. Describe some of the standard professional or societal codes of ethics that relate to the events, and what can the impact to our society be, when such codes are not applied.