Homework 3: Applied Cryptography

This homework is due **November 5, 2019** at **10 p.m.**. You will have a budget of five late days (24-hour periods) over the course of the semester that you can use to turn assignments in late without penalty and without needing to ask for an extension. You may use a maximum of two late days per assignment. Once your late days are used up, extensions will only be granted in extraordinary circumstances.

We encourage you to discuss the problems and your general approach with other students in the class. However, the answers you turn in must be your own original work, and you must adhere to the Code of Academic Integrity. Solutions should be submitted electronically via Canvas with the template at the end of this document.

Concisely answer the following questions. (Limit yourself to at most 80 words per subquestion.)

1. **Public-key encryption.** The CIS 331 staff have decided to use public-key cryptography with this class to report the grades from your final exams. They instruct each student to create a 2048-bit RSA key pair and post the public key (e,N) publicly on Piazza. Assume that the CIS 331 staff correctly receives the public key for each student. They wish to encrypt each student's score to that student's key to produce a ciphertext c and they will post c to the website alongside the student's name. The test will be out of 100 points and you cannot earn fractional points.

   (a) Assume that $e = 3$ for each student and the staff encrypts each student's score s by computing $c := s^e \bmod N$. Is this scheme secure? If so, briefly argue why; if not, how much information can an attacker obtain and how much work does it require them to perform?

   (b) Assume that $e = 65537$ for each student and the staff encrypts each student's score s by computing $c := s^e \bmod N$. Is this scheme secure? If so, briefly argue why; if not, how much information can an attacker obtain and how much work does it require them to perform?

   (c) Assume that $e = 65537$ for each student and the staff encrypts each student's score s by first padding s with $\lfloor (2048 - \lg 100)/8 \rfloor$ hex bytes `FF` to obtain a new message t and then computing $c := t^e \bmod N$. Is this scheme secure? Argue why or why not.

   (d) What additional steps should the CIS 331 staff take to prevent the attacks you found?

2. **Applied cryptography.** You have been contracted to perform a security analysis of Shushmail, a new "secure" email provider.

   Each Shushmail user has an RSA public key with public exponent 15 and a unique 4096-bit modulus N. Another user may encrypt a message m to this key by choosing a random 256-bit key k, using the AES block cipher in CTR mode (a secure mode unlike ECB) to encrypt m using key $k : E_m = AES_k(m)$ and then using RSA to encrypt k to the recipient's public key $E_k = k^{15} \bmod N$. The encrypted message is then the pair (E_k, E_m).

   (a) Assume that the keys are properly generated, users have a way of looking up correct public keys for recipients, and the private RSA keys are stored securely. Describe two vulnerabilities in this protocol and give a precise description of how to fix them.

3. **Authentication protocols.** A large university has implemented a central sign-on facility where users authenticate themselves to an official site and then receive a token that confirms their identity to all other campus sites.

   (a) Assuming the protocol is competently implemented and deployed, how might deploying this service improve security on campus?

   (b) Under the same assumptions, how might it hurt security?

   Suppose the sign-on protocol proceeds as follows: When the user visits site A, which requires authentication, site A redirects the user to the central sign-on site. Following authentication, the central sign-on site redirects the user's browser back to a standardized HTTPS URL at site A with the following parameters: u, the user's username, and $\text{Sign}(u)$, a digital signature produced with the sign-on site's private key. (Assume that the corresponding public key is widely known.) The site checks that the signature is valid for u, and considers the user authorized if so.

   (c) If site A is controlled by an attacker, how can it trivially impersonate the user to other sites that trust the sign-on protocol?

   (d) Propose a simple change to the protocol that would fix the problem identified in (c).

4. **HTTPS.** A *self-signed certificate* makes the claim that a public key belongs to a particular server, without any trusted certificate authority (CA) to verify it. Browsers display a warning message when a site presents such a certificate, but users often override these warnings. Some websites use self-signed certs to avoid the trouble of obtaining a cert from a trusted CA.

   (a) Briefly explain how using HTTPS with a self-signed certificate provides protection against a passive eavesdropper.

   (b) How might a man-in-the-middle attacker compromise a site that uses a self-signed certificate, assuming that the client ignores browser certificate warnings?

(c) Some sites use HTTPS with a certificate signed by a trusted CA for their login pages, then set a session cookie and use HTTP for the other pages on the site. Briefly compare the security of this design to the use, for all pages on the site, of (i) a self-signed certificate and (ii) a certificate signed by a trusted CA.

5. **Collision-resistant hash functions.** Collision resistant hash functions ensure that it is hard for a computationally bounded adversary to find two different inputs x and y ($x \neq y$) that hash to the same value ($H(x) = H(y)$). Often times developers use collision resistant hash functions a way to hide the inputs (thinking that *all* collision resistant hash functions are indifferentiable from random oracles). However, collision resistant hash functions are not guaranteed to hide every bit of the input. You will prove that this is the case.

   (a) Assume that $H : \{0,1\}^* \to \{0,1\}^n$ is a collision resistant hash function. Use H to build another collision resistant hash function $H' : \{0,1\}^* \to \{0,1\}^{n+1}$ that **always** leaks one bit of the input in its output. Make sure to state explicitly which bit of the input is always leaked.

   (b) Prove that your construction of H' is also collision resistant. To do this, use a cryptographic reduction to show that if an attacker finds a collision in H' with non-negligible advantage, the attacker can also find a collision in H with non-negligible advantage.

   Hint: Construct an attacker A that uses the attacker for H' as a sub-routine.

   (c) Assume the adversary has advantage of $Adv_{H'}$ at finding collisions in H'. What is the advantage of the adversary at finding collisions in H as a function of $Adv_{H'}$?

   (d) (**Extra credit**) Use H to build a collision resistant hash function that always leaks one bit of the input, but that has the same range as H, namely $H' : \{0,1\}^* \to \{0,1\}^n$. Specify precisely which bit of the input is leaked, prove that H' is also collision resistant, and give the adversary's advantage of finding collisions for H as a function of $Adv_{H'}$.

Submission Template

Submit by uploading a `txt` file to Canvas. Use the template below to organize your submission.

```
# Problem 1

  1a.

  1b.

  1c.

  1d.

# Problem 2
  2a.

# Problem 3

  3a.

  3b.

  3c.

  3d.

# Problem 4

  4a.

  4b.

  4c.

# Problem 5

  5a.

  5b.

  5c.

  5d. (optional extra credit)
```