601.445/601.645: Practical Cryptographic Systems February 17, 2020

Weekly Homework 3

Instructor: Matthew Green *Due: 11:59pm, February 24*

Name: ______________________________

The assignment should be completed individually. You are permitted to use the Internet and any printed references. You may find Katz-Lindell §11.5 (§10.4 in the first edition) or Handbook of Applied Cryptography §8.2 (in the online public edition) helpful.

Please submit the completed assignment via Blackboard.

Problem 1: Do the following problems. Show your work.

1. Apply the extended Euclidean algorithm to primes 59 and 17 to find x and y such that $59x + 17y = 1$.
2. What is the inverse of 59 (*i.e.,* 8) modulo 17 and what is the inverse of 17 modulo 59?
3. Prove that 2 is a generator of $\mathbb{Z}_{59}^*$, while 4 is not a generator of $\mathbb{Z}_{59}^*$. (HINT: Recall, g is a generator of $\mathbb{Z}_{59}^*$, where p is prime, if and only if $g^a \neq 1 \; mod \; p$ for every non-trivial divisor $1 < a < p-1$ of $(p-1)$.
4. List all of the subgroups of $\mathbb{Z}_{23}^*$ and provide one generator for each subgroup.[1] See HAC §2.5 for definitions if this is helpful.

Problem 2: Suppose that $n = pq$, where p and q are distinct odd primes and $ab \equiv 1 \; mod \; \phi(n)$.[2] The RSA encryption operation is $e(x) = x^b \; mod \; n$ and the decryption operation is $d(y) = y^a \; mod \; n$. Answer the following questions:

1. Explain why $d(e(x)) = x$ if $x \in \mathbb{Z}_n^*$. Hint: observe that $ab = t\phi(n) + 1$ for some integer $t \geq 1$. Now note that if $x \in \mathbb{Z}_n^*$ then $(x^b)^a \equiv x^{t\phi(n)+1} \; mod \; n$.
2. Explain why $d(e(x)) = x$ if $x \in \mathbb{Z}_n \setminus \mathbb{Z}_n^*$. Hint: see Katz and Lindell or HAC.
3. You are given an RSA pubic key $n = 52,810,853, e = 5$ and a ciphertext $c = 23,273,341$. Find the corresponding plaintext.
4. Imagine that two different RSA keypairs share the same modulus n. This means that there are different public exponents b_1, b_2. We will further specify that $gcd(b_1, b_2) = 1$. Now a sender encrypts the same message x with each exponent, producing $c_1 \equiv x^{b_1} \; mod \; n$, $c_2 \equiv x^{b_2} \; mod \; n$. Imagine an eavesdropper intercepts both ciphertexts. Show how this can be used to attack the cryptosystem.

[1] Note: a subgroup is a cyclic group that is contained within the larger group. You can find subgroups by doing what we did in class: picking an element of the group and seeing whether it generates the whole group or just a subset.

[2] Remember that $\phi(n) = (p-1)(q-1)$.

Protocol Needham-Schroeder shared-key protocol

SUMMARY: A interacts with trusted server T and party B.
RESULT: entity authentication (A with B); key establishment with key confirmation.

1. *Notation*. E is a symmetric encryption algorithm (see Remark 12.19).
   N_A and N_B are nonces chosen by A and B, respectively.
   k is a session key chosen by the trusted server T for A and B to share.
2. *One-time setup*. A and T share a symmetric key K_{AT}; B and T share K_{BT}.
3. *Protocol messages*.

$$\begin{array}{llr} A \to T: & A, B, N_A & (1) \\ A \leftarrow T: & E_{K_{AT}}(N_A, B, k, E_{K_{BT}}(k, A)) & (2) \\ A \to B: & E_{K_{BT}}(k, A) & (3) \\ A \leftarrow B: & E_k(N_B) & (4) \\ A \to B: & E_k(N_B - 1) & (5) \end{array}$$

4. *Protocol actions*. Aside from verification of nonces, actions are essentially analogous to those in Kerberos (Protocol 12.24), and are not detailed here.

Figure 1: Needham-Schroeder protocol, excerpted from §12.26 of the Handbook of Applied Cryptography. The terms "A" and "B" refer to the *identities* of the parties. The *nonces* N_A, N_B (literally, "number used once") are fresh and hopefully random strings generated by A and B respectively.

5. A trusted central party wants to make and distribute many RSA keypairs that all share a single public modulus $n = pq$. For each of m users in the system she generates a different public/secret exponent pair $(e_1, d_1), (e_2, d_2), \ldots, (e_m, d_m)$ that work with m, and sends each exponent pair to a party so that the i^{th} party's public key is (n, e_i). What is the risk of this system?

Problem 3: **Protocols** Review the Needham-Schroeder shared-key key distribution protocol shown in Figure 1.[3] This protocol allows two parties (A and B) to agree on a shared key, using a trusted party T (with whom each party already shares a key) as an introduction point. You should assume that A, B and T are all trustworthy, and that the encryption scheme provides strong confidentiality. You can also assume that the attacker controls the network and is allowed to eavesdrop, block or *modify* any messages sent between the parties.

Answer the following questions:

1. Assuming no fancy attacks on the encryption scheme, what prevents an evil user (C) from impersonating A to B?
2. What are the nonces N_A, N_B needed for?
3. Why does the final message encrypt $E_k(N_B - 1)$ rather than $E_k(N_B)$? What attack

[3] This diagram is excerpted from §12.26 of the HAC. It should not be confused for the Needham-Schroder *public key* protocol, which is a different protocol entirely!

might be possible if the subtraction was removed, and the final message simply contained $E_k(N_B)$?

4. In the second (and third) message, why does the ciphertext $E_{K_{BT}}(k, A)$ encrypt the identity A?
5. Imagine that the encryption scheme E is implemented using AES-CTR with no MAC. What attacks could you come up with against this protocol?
6. What happens to this protocol if A is ever compromised and all of its data (keys etc.) are stolen?