**601.445/601.645: Practical Cryptographic Systems** October 21, 2021

# Weekly Homework 3

*Instructor: Matthew Green and Alishah Chator* *Due: 11:59pm, October 28*

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 Gradescope.**

**Problem 1**: Suppose that $n = pq$, where $p$ and $q$ are distinct odd primes and $ab \equiv 1 \ mod \ \phi(n)$.[1] 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.

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 $i^{th}$ partys public key is $(n, e_i)$. What is the risk of this system?

**Problem 2**: **Protocols** Review the Needham-Schroeder shared-key key distribution protocol shown in Figure 1.[2] 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

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

[2] 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!

**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.*

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

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.

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 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?