COMP 332, Homework 10: *Cryptography, Security, and Privacy*
Due by 11:59pm on December 5, 2018

---

1. Written problems (20 points)

Problem 1. *Consider RSA with $p = 5$ and $q = 11$.*

a: *What are n and z?*

b: *Let e be 3. Why is this an acceptable choice for e?*

c: *Find d such that $de \equiv 1 (mod\ z)$, that is $de \mod z = 1$ and $d < 160$.*

d: *Encrypt the message $m = 8$ using the key (n, e). Let c denote the corresponding ciphertext. Show all work. Hint: to simplify calculations, use the fact: $[(a \mod n) \cdot (b \mod n)] \mod n = (a \cdot b) \mod n$.*

Solution:

a: $n = pq = 55$, $z = (p-1)(q-1) = 40$

b: Using $e = 3$ is an acceptable choice as it is less than n and it has no common factors with z.

c: $d = 27$

d: Let c be the ciphertext. Then $c = m^e \mod n$. Substitution gives $c = 8^3 \mod 55 = 17$.

Problem 2. *In this problem, we explore the Diffie-Hellman (DH) public-key encryption algorithm, which allows two entities to agree on a shared key. The DH algorithm makes use of a large prime number p and another large number g less than p. Both p and g are made public (so that an attacker would know them). In DH, Alice and Bob each independently choose secret keys S_A and S_B respectively. Alice then computes her public key T_A by raising g to S_A and then taking mod p. Bob similarly computes his own public key T_B by raising g to S_B and then taking mod p. Alice then calculates the shared secret key S by raising T_B to S_A and then taking mod p. Similarly Bob calculates the shared key S' by raising T_A to S_B and then taking mod p.*

a: *Prove that, in general, Alice and Bob obtain the same symmetric key, that is, prove $S = S'$.*

b: *With $p = 11$ and $g = 2$, suppose Alice and Bob choose private keys $S_A = 5$ and $S_B = 12$ respectively. Calculate Alice's and Bob's public keys, T_A and T_B. Show all work.*

c: *Following up on part (b) now calculate S as the shared symmetric key. Show all work.*

d: *Provide a timing diagram that shows how Diffie-Hellman can be attacked by a man-in-the-middle. The timing diagram should have three vertical lines, one for Alice, one for Bob,*

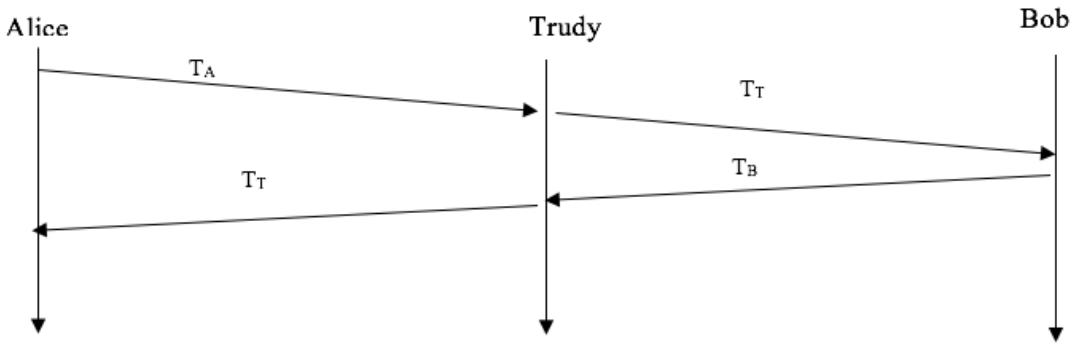

FIGURE 1. Question 2(d): Man-in-the-middle attack on Diffie-Hellman.

and one for the attacker Trudy.

Solution:

a:

Alice's secret key: S_A
Bob's secret key: S_B
Alice's public key: $T_A = g^{S_A} \mod p$
Bob's public key: $T_B = g^{S_B} \mod p$
Alice's shared key: $S = T_B^{S_A} \mod p$
Bob's shared key: $S' = T_A^{S_B} \mod p$

$$\begin{aligned} S &= T_B^{S_A} \mod p \\ &= (g^{S_B} \mod p)^{S_A} \mod p \\ &= (g^{S_B S_A}) \mod p \\ &= (g^{S_A} \mod p)^{S_B} \mod p \\ &= (T_A^{S_B}) \mod p \\ &= S' \end{aligned}$$

b,c: $p = 11$, $g = 2$

Alice's secret key: $S_A = 5$
Alice's public key: $T_A = g^{S_A} \mod p = 10$
Alice's shared key: $S = T_B^{S_A} \mod p = 1$
Bob's secret key: $S_B = 12$
Bob's public key: $T_B = g^{S_B} \mod p = 4$
Bob's shared key: $S' = T_A^{S_B} \mod p = 1$

d: See Figure 1. The Diffie-Hellman public key encryption algorithm can be attacked by man-in-the-middle. In this attack, Trudy first receives Alice's public value (T_A) and sends

her own public value (T_T) to Bob. When Bob transmits his public value (T_B) in response, Trudy sends her public key to Alice (T_T). Trudy and Alice thus agree on one shared key (S_{AT}) and Trudy and Bob agree on another shared key (S_{BT}). After this exchange, Trudy simply decrypts any messages sent out by Alice or Bob by the public keys S_{AT} and S_{BT}.

PROBLEM 3. *In this last assignment I want to give you a chance to explore a problem that is of interest to you. A list of potential problem options to choose from is below:* **choose one these options or propose your own**.

Option 1: *Update the chat app that you implemented in homework 5 so that your communication protocol provides confidentiality and message integrity to its users. You may wish to make certain simplifying assumptions, such as users having pre-shared symmetric keys. You may import whatever python modules you would like, but you must still run over TCP rather than TLS: i.e., you cannot just use the security properties of TLS. Submit your *.py files along with a 1-page write-up that describes your protocol and discusses how your protocol could be modified to additionally provide endpoint authentication as well as not require the use of pre-shared keys.*

Option 2: *Get familiar with Tor. Good references to read are the following.*

```
https://www.torproject.org/about/overview
https://en.wikipedia.org/wiki/Tor_(anonymity_network)
https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf
```

Then download and test out the Tor browser, including looking at the traffic generated in Wireshark.

```
https://www.torproject.org/projects/torbrowser.html.en
```

Submit a 1-page document describing tor at a (very) high level. Include a screenshot of tor traffic that you see generated in Wireshark when you run the Tor browser along with a brief description of what you have observed.

Option 3: *Figure out how to implement the TCP handshake in scapy: this should be a very small program.*

```
https://scapy.net/
https://scapy.readthedocs.io/en/latest/
```

Submit a 1-page document describing scapy at a (very) high level along with what should be a small python script. Include a screenshot of scapy traffic that you generated in Wireshark along with a brief description of what you have observed.

Option 4: *Read up about the Internet of Things (IoT) and about the Shodan search engine that searches IoT devices.*

```
https://en.wikipedia.org/wiki/Internet_of_things
https://www.shodan.io/
https://en.wikipedia.org/wiki/Shodan_(website)
https://www.wired.com/2016/12/botnet-broke-internet-isnt-going-away/
```

Submit a 1-page document describing how the IoT operates and how Shodan searches the IoT. Describe strengths as well as weaknesses (i.e., vulnerabilities) in the IoT.

Option 5: *Update the router code from homework 9 so that it can handle routers that leave and arrive (rather than just arrive) and changing link costs. Submit your *.py files along with a 1-page write-up that describes your overall code design and how you have updated your code to handle these changes.*

Option 6: *Read up about how to automate web-browsing using Selenium*

```
https://www.seleniumhq.org/
```

Submit a 1-page document describing selenium at a (very) high level, along with a small program. Include a screenshot of scapy traffic that you generated in Wireshark along with a brief description of what you have observed.

Option 7: *Another problem of your choosing. This should involve either some programming or writing up of an approximately 1-page document that could potentially additionally describe interesting features of traffic that you have observed using Wireshark.*

2. Submission

Upload your written work as `hw10.pdf` and any *.py files or other programming scripts to the WesFiles directory I have created for you at the following URL. All files should include your name!

```
https://wesfiles.wesleyan.edu/home/vumanfredi/web/comp332-f18/submissions/hw10/USERNAME
```

You should replace `USERNAME` with your Wesleyan username. You will be asked to enter your Wesleyan username and password to access the page. Once the page opens, you should click on the "Open Web View" link that shows up on the page, and that should take you to a page that gives you options to upload files.