****

|  |
| --- |
| **2019 Tool Pilot Assignment: Personal & Enterprise Security Tools & Practices** |

1. **Assignment Objectives:**

The main goal of this assignment is to get you acquainted with some of the techniques and tools that exemplify various security related concepts discussed in this course. Towards this, you are required to complete the tasks outlined below in the themedshort exercises. Please note that the tools that you will be using represent simplified versions of security forensics, auditing, and encryption tools that may be used in an enterprise production environment. Also, as an introductory assignment, each requirement is elaborated through a complete series of steps to guide you as much as possible.

1. **Assignment Submission Requirements:**

The **deliverable** for this assignment is:

1. a typed report (fill-in-the-blanks worksheet in this document is provided after each section) with your answers to the questions in each of the short exercises in the assignment along with the required screenshot exhibits. This document, with the completed worksheets, is to be submitted electronically via **BrightSpace**
2. Exemplar 5 involves a deliverable component that is submitted via email (because, it is an email). Instructions for this are provided on that worksheet.
3. Your solutions document should use the following naming format:
   1. ***“*FirstName-LastName-EBC6170.docx*”***
   2. (i.e., Jane Doe would submit: “Jane-Doe-EBC6170.docx”).

Your zip file must be submitted through the **BrightspaceWebsite**.**Do not submit directly to the professor**.

|  |  |
| --- | --- |
| **Personal Work Statement**  I, the undersigned:   * warrant that the work submitted herein is my own work and not the work of others * acknowledge that I have read and understood the University Regulations on Academic Misconduct * acknowledge that it is a breach of University Regulations to give or receive unauthorized and/or unacknowledged assistance on a graded piece of work * acknowledge that I am individually and solely responsible for the work submitted herein | |
| **Juveriya Zameer** | **300116986** |
|  |  |

1. **Assignment Notes:**

|  |
| --- |
| This assignment requires: i) some use of the windows command-line to enter various commands, and ii) pasting screenshots of windows in your solution document (this document).   * To get into the command prompt, you can click the start button, or type the “Windows” key and “R” concurrently to bring up the Run bar. Once in the Start menu or Run bar, type “cmd” or “command” to bring up the command prompt. It is recommended that you use the “Run as administrator” option to enter the command prompt. This option can be brought up in the context menu by right-clicking the Command Prompt menu item (as shown below).      * For the parts requiring you to take a screenshot of the window, you can capture an active window onto the clipboard by pressing ***Alt+Print Scrn***, and paste it in a word document by pressing ***Ctrl+V***. You can also make use of the Windows “**Snipping Tool**” which is usually installed by default * It is highly recommended that you run the utilities in this assignment by explicitly using Administrator privileges. You can do this by right clicking the relevant application file and selecting “Run as administrator”. * **Attention Mac Users / Non-Windows Users:**This assignment pilots tools compiled for the Windows operating system. Although variants of these tools may exist for other OS, the assignment has only been vetted against Windows. If you do not have access to a Windows machine or a Windows emulator, you can make use of any of the machines in the Campus computer labs or in the library. |

Table of Contents

[Application Exemplar 1: Basic Network Diagnostics & Forensics Tools 4](#_Toc24901253)

[Worksheet 1: Basic Network Diagnostics & Forensics Tools RESPONSES 9](#_Toc24901254)

[Application Exemplar 2: Using Hash Functions to Verify Integrity & Passwords 10](#_Toc24901255)

[Worksheet 2: Using Hash Functions to Verify File Integrity & Passwords 14](#_Toc24901256)

[Application Exemplar 3: Proactive Phishing Safeguards Training 17](#_Toc24901257)

[Worksheet 3: Anti-Phishing Phil 18](#_Toc24901258)

[Application Exemplar 4: Symmetrical Encryption + Blue Team Case 19](#_Toc24901259)

[Worksheet 4: Symmetrical Encryption + Blue Team Case 21](#_Toc24901260)

[Application Exemplar 5: Asymmetric & Session Key Encryption 23](#_Toc24901261)

[Worksheet 5: Asymmetric & Session Key Encryption 32](#_Toc24901262)

[Application Exemplar 6: Informational Components in an X.509 Digital Certificate 33](#_Toc24901263)

[Worksheet 6: Digital Certificates 44](#_Toc24901264)

[Application Exemplar 7: Other Security Enhancing & Privacy Preserving Tools 46](#_Toc24901265)

[Worksheet 7: Other Security Enhancing & Privacy Preserving Tools 47](#_Toc24901266)

* There are a total of **115 pts** in this assignment. Scoring breakdowns are given for each question in the worksheet.   
  + **Hint:**If a question is worth 3 marks, it suggests that 3 points should be provided, and so on.
* **Note**: All solutions are to be **recorded** in the **worksheet** sections of **thisdocument**.
* This is an individual assignment. If you require assistance, please **contact** the professor (**Karim**) or the TA (**Tarun**) as soon as possible to schedule an appointment. You may also direct questions via email to either of these resources.
* I recommend doing the sections **in order**, as sometimes a section will rely on a previous tool or activity.
* This assignment should take from **3-5 hours** to complete. You may find it best to break it up into multiple, shorter sessions.
* For this assignment, you will need to **install** some PC based software. You take full responsibility for the safety of these installs. Please ensure you have an **active anti-virus**scannerrunning and be sure to take advantage of any **digital fingerprints** the author may publish.
* Print submissions will not be accepted. Submit via **BrightSpace** only, please.

# Application Exemplar 1: Basic Network Diagnostics & Forensics Tools

In this exercise, you will use a series of tools that can be used for monitoring the activities and the state of your network.

In order to complete this exercise, follow the steps outlined below to answer the questions in each part, and use the ***worksheet*** provided **at the end of this Exemplar** to record your answers to the questions. NOTE: Please do not record your answers inline with the Exemplar. Instead, please scroll down to the end of the Exemplar and you will find a worksheet designed for you to edit.

**A. Your Local Host & Network Information**

**What:** Find out your own **computer’sIPaddress** and that of your network **gateway**.

**How:**

* On the windows **commandline**, use the “**ipconfig**” command.
* ***Record your answers in the worksheet space below***.

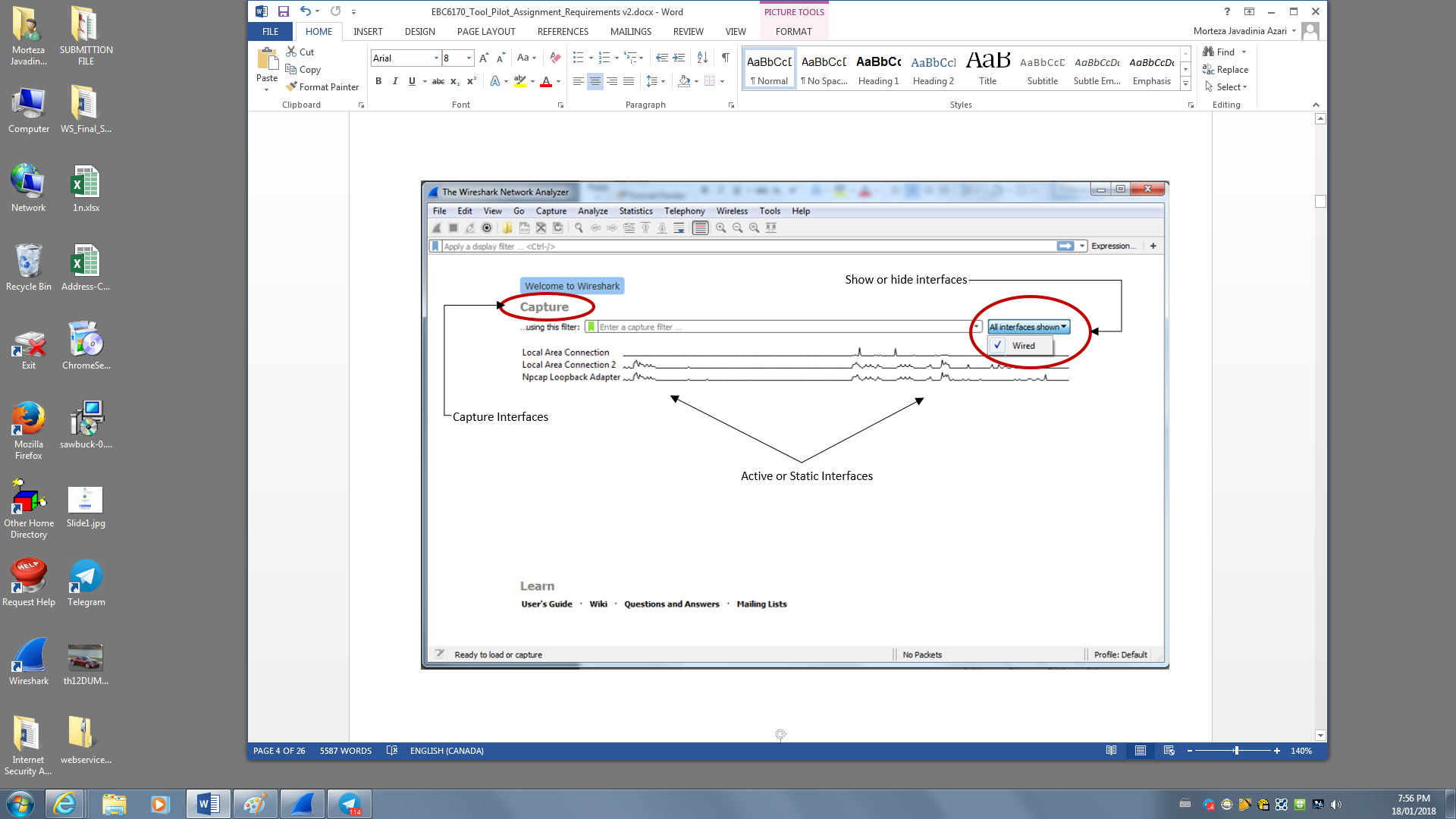
Henceforth, in this exercise, your own computer’s IP address will be referred to as *<<ip-address>>* and that of your gateway will be referred to as <<*gateway-ip-address>>*.

**B. Real-Time Network Monitoring**

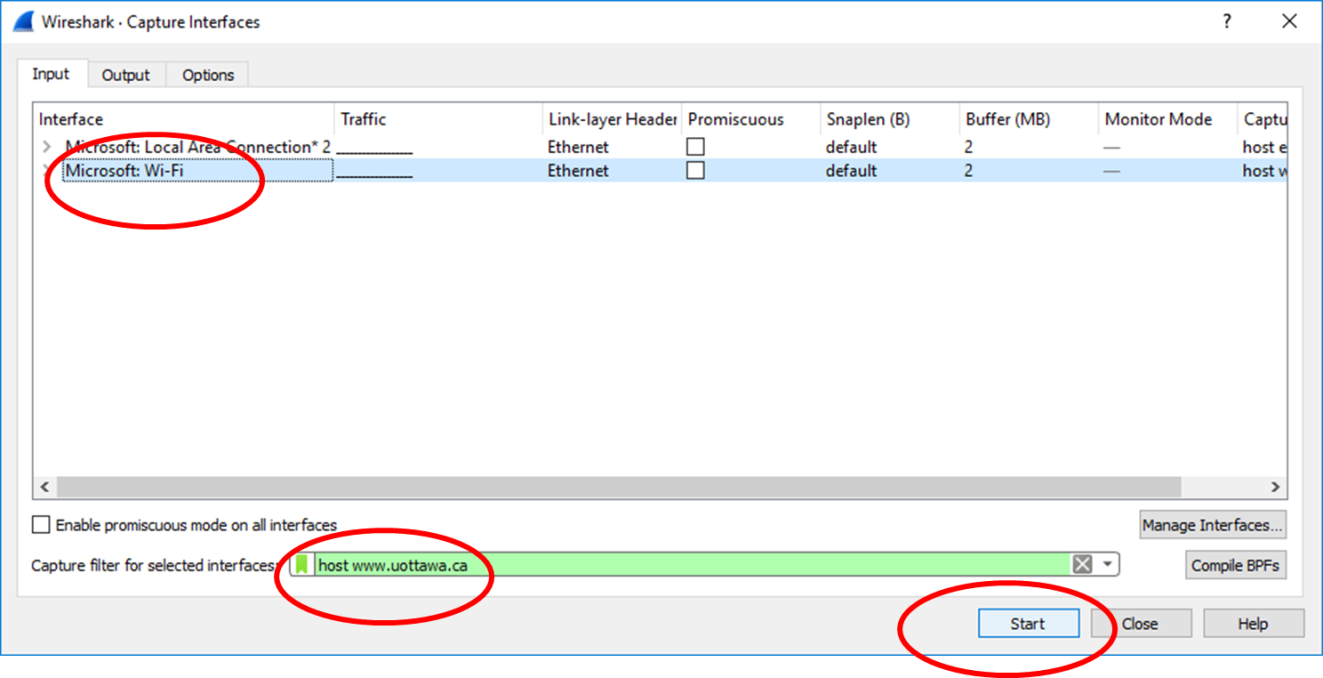
**What:** Explore the network activity and data transfers related to different types of client requests and server responses.

**How:**

* Download and install the Wireshark network protocol analyzerfrom <https://www.wireshark.org/download.html> (select between 32-bit and 64-bit versions depending on your operating system). During install, you can use the default installation options. The installation may also require you to **install** a separate package called“**WinPCap**”. Also install **USBPcap** if asked. The installation procedures will require you to confirm setup instructions in several stages.
* Once installed, run the applicationfrom your desktop, start menu or your apps drawer.***It is recommended that you run the program as an administrator***.
* In the Wireshark network analyzer utility, you can view a list of your **networkinterfaces**. Typically, your active interface shows some activity while others are static.
* You can click the**Capture**menu item (top navigation menu) and configure the network interfaces under “Options”. You do not have to use “**Promiscous**” mode for this assignment, and you can **uncheck** the checkboxes with the interface items as well as the “Enable Promiscous mode…” checkbox at the bottom.
* Within the **Capture**🡺**Options** dialog, you can also **hide** all the inactive/static interfaces by clicking the “**ManageInterfaces**” button.



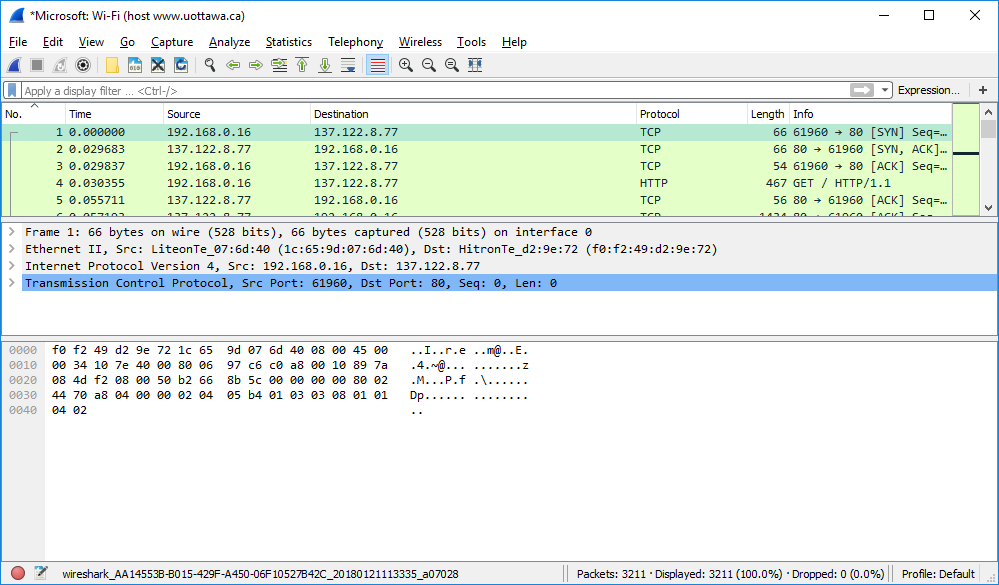
* The **Capture** options button for the interface can be used to specify the criteria that you want to use for capturing network packets. For example, in the next exercise, you will select the correct network interface, and use the following value as a **Capture Filter: “host www.postalgate.com”.** This value simple specifies that you are interested in capturing traffic between your computer and the postalgate.com website as specified by the host name.



* After specifying the capture filter as shown above, click the **Start** button to commence the network traffic analysis session.
* ***Optional:*** For the following parts of this assignment, if you are using Chrome as your browser, you can use the Incognito mode. This can help reduce the amount of network traffic through your browser.
* Launch your web browser and **browse**to the Postal Gate website using its URL:  
  “**www.postalgate.com**”.
* **Stop** the network analysis session by clicking the stop button in the Wireshark window toolbar.



* Quickly scroll through the sequence of network packets to **see the different protocols** that were used to send information to and receive information from the website.



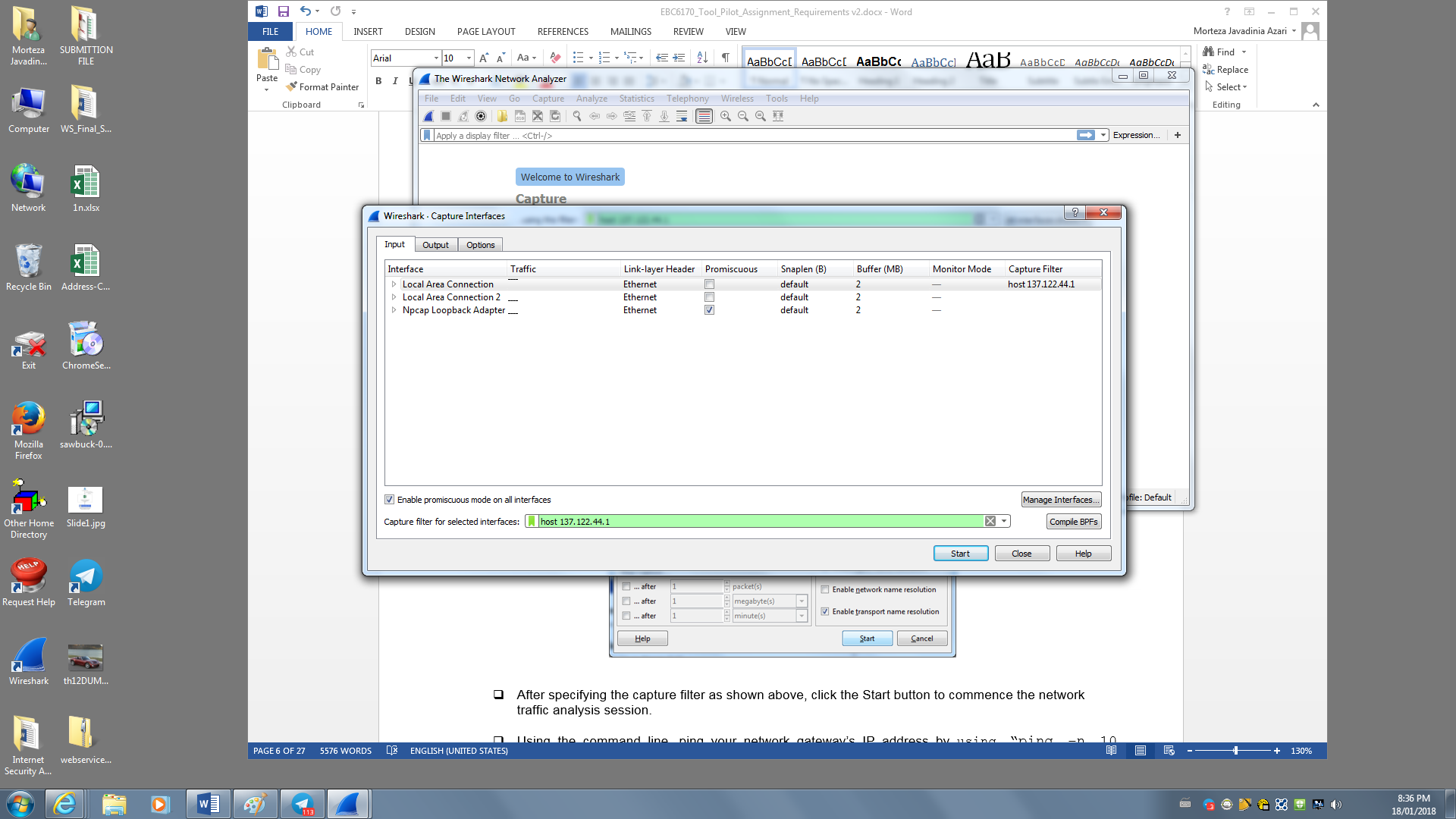
* To **view** the contents of packets, you can **expand** the trees in the middle pane by clicking the plus sign.
* **Q:** What**high-level protocols** were used to communicate with the website? Which of these protocols contains the actual web page content being transferred to your web browser?***Record your answers in the responses section below***.
* **Q:**There are some **image** files (PNG, JPG etc.) that were exchanged during this network session. Identify packets with these files. Highlight a packet containing such an image file in Wireshark, and provide a screenshot of the window. The **Source**, **Destination**, **Protocol** and **Info** columns should be visible in your screenshot.***Paste your screenshot in the responses section below***.
* You can **close** the logs window (File 🡺 Close) once you’re done answering the questions above. There’s no need to save the log to a file.

**C. Real-Time Network Monitoring**

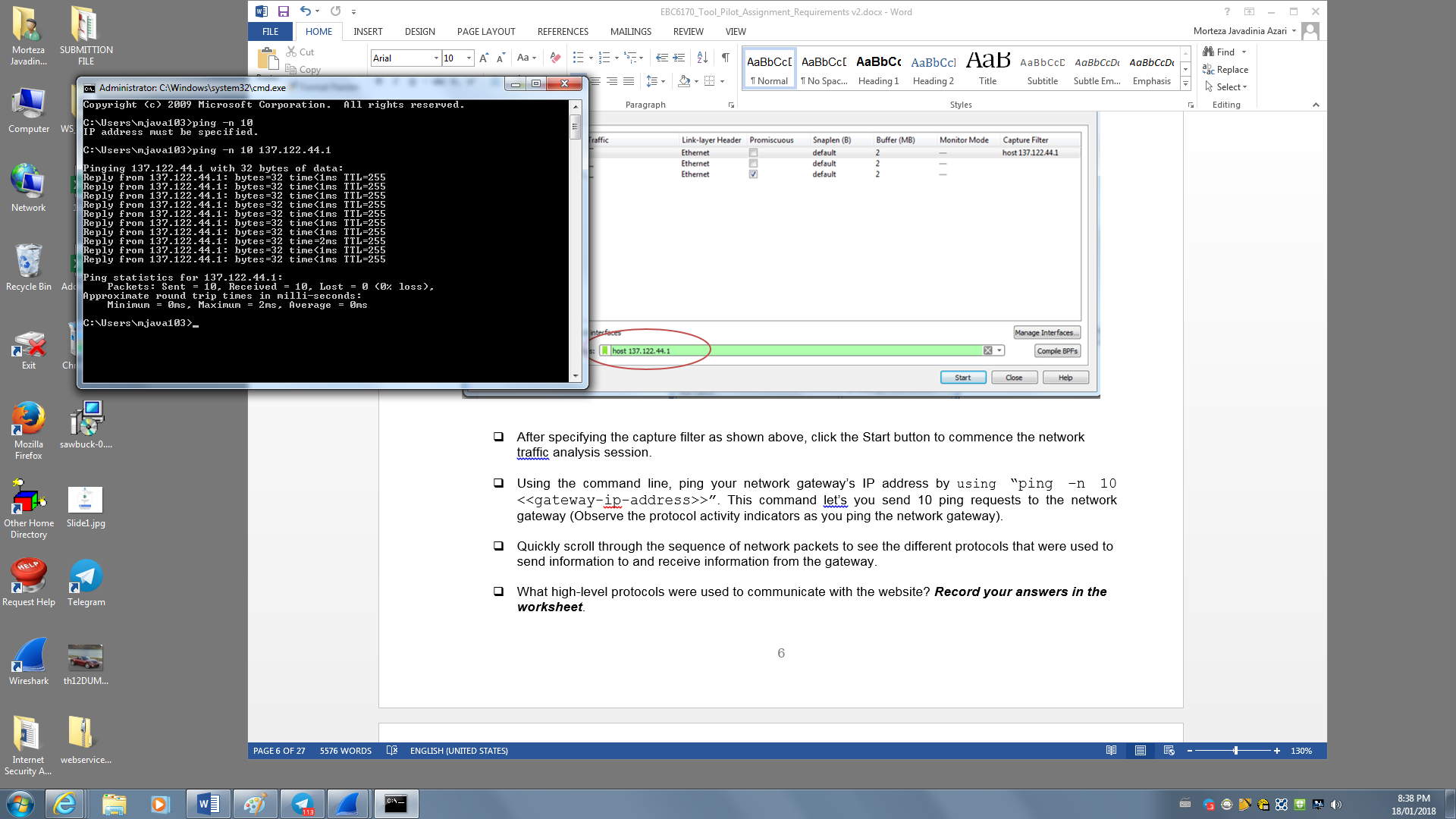
**What:** Explore the network activity and data transfers related to different types of client requests and server responses.

**How:**

* In the next exercise, you will continue to use the wireshark utility. Use the following value as a **CaptureFilter**:“**host << *gateway-ip-address*>>**” (replace with your **gatewayaddress** from part A). This value specifies that you are interested in capturing traffic between your computer and your network gateway as specified by the IP-address. You simply need to enter the correct value in the text field corresponding to the Capture Filter as shown below:



* After specifying the capture filter as shown above, click the **Start** button to commence the network traffic analysis session.
* Using the **commandline**, **ping** your network gateway’s IP address by using “**ping –n 10 <<gateway-ip-address>>”.** This command lets you send 10 ping requests to the network gateway (Observe the protocol activity indicators as you ping the network gateway).



* Quickly scroll through the sequence of network packets to **see the different protocols** that were used to send information to and receive information from the gateway.
* What high-level protocols were used to communicate with the website?***Record your answersthe responses section below***.
* You can **close** the logs window once you’re done answering the questions above. There’s no need to save the log to a file.

**D. Network Packet Sniffing over HTTP**

**What:** Explore network data transfers and HTTP packet content.

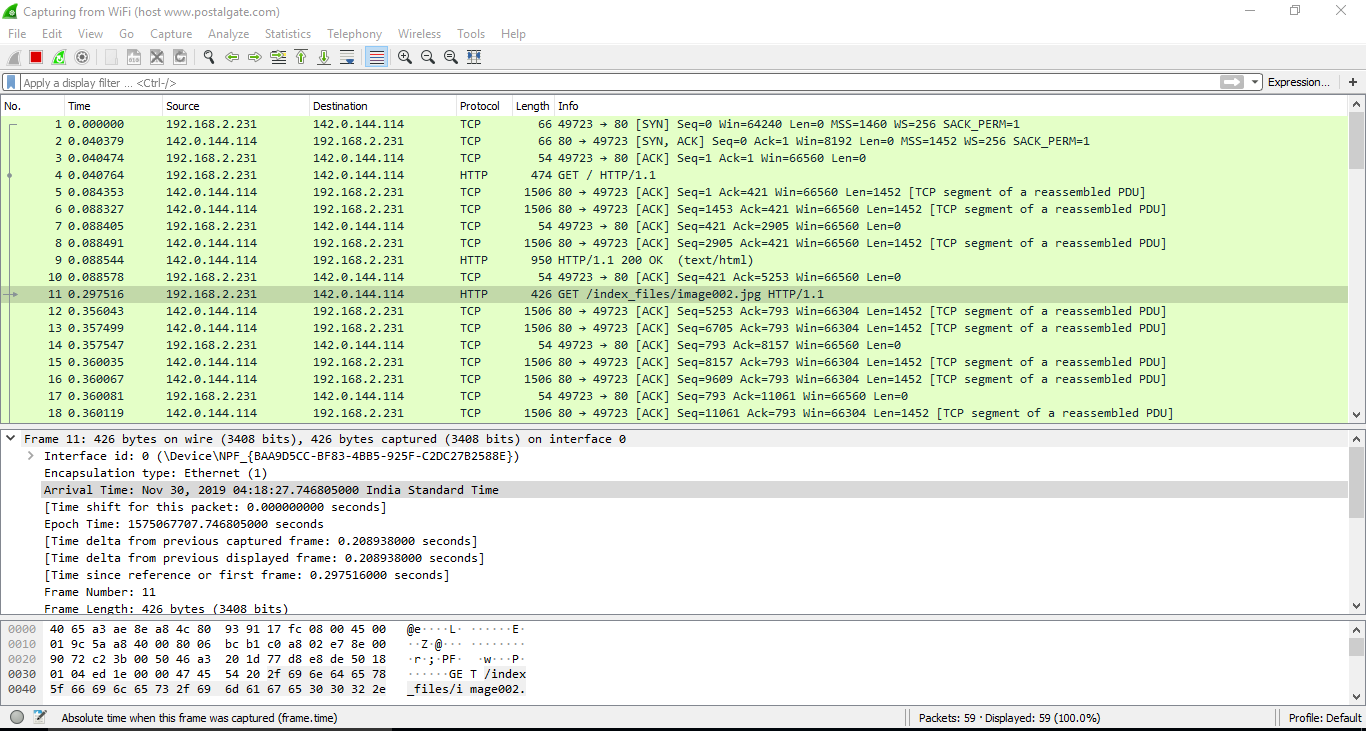
**How:**

* In the next exercise, you will continue to use the Wireshark utility. Use the following value as a Capture Filter:“**host www.postalgate.com**”. This value specifies that you are interested in capturing traffic between your computer and the **WWW.POSTALGATE.COM** website as specified by the host name.After specifying the capture filter, click the Start button to commence the network traffic analysis session.
* Navigate to <http://www.postalgate.com/login/>
* On the website, in the login form, enter a bogus (fake) userid and password. Upon clicking the login button, you should be directed to a response page stating that the authentication credentials are **invalid**.
* **NOTE**: Some browsers, such as FireFox, will attempt to enforce **securityconstraints** and will stop you from being able to submit the form. If this happens, select “**Advanced**” and **accept** the Security Exception to continue.
* Stop the network analysis session by clicking the **stopbutton** in the Wireshark window toolbar.
* Quickly scroll through the sequence of network packets **to find the HTTP packets** that were used for receiving authentication prompts and sending authentication credentials.
* Were the credentials sent in plaintext or encrypted format? ***On the worksheet, record your answer and provide a screenshot showing the credentials embedded within the packet****.*
* You can **close** the logs window once you’re done answering the questions above. There’s no need to save the log to a file.

## Worksheet 1: Basic Network Diagnostics & Forensics Tools RESPONSES

(**15pts**) Complete the following worksheet after completing the tasks outlined in exercises 1A, 1B, 1C and 1D.

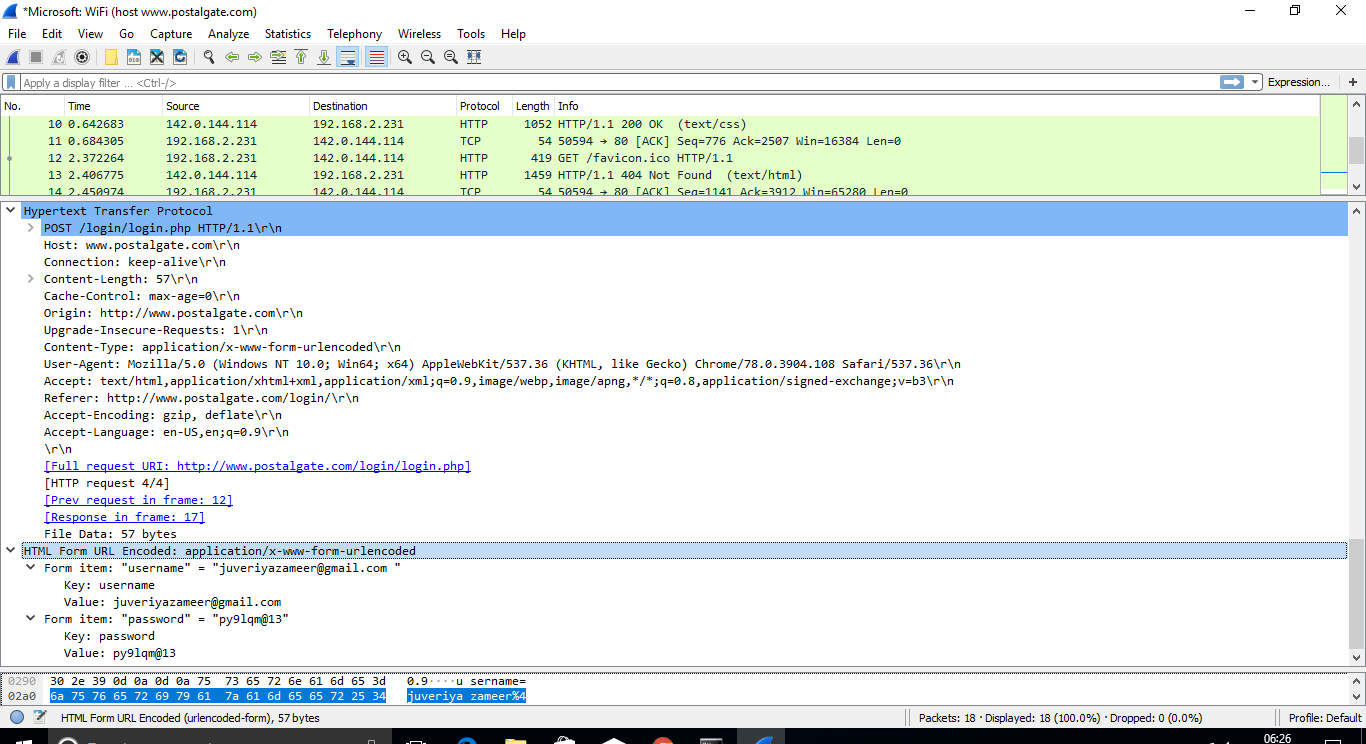
* (Q1**2pts**) Your Computer’s IP Address – is it an IPv4 or an IPv6 address?
* 192.168.2.231 - IPv4
* (Q2**2pts**) Your Network Gateway’s IP Address– is it an IPv4 or an IPv6 address?
* 192.168.2.1 – IPv4
* (Q3**2pts**) Main Protocols used to communicate with [www.postalgate.com](http://www.postalgate.com):
* High level protocols used to communicate with the web browser is https, TCP.
* The different protocols that were used to send and received information from gateway protocols are , SSDP, IGMPv2, ARP, ICMP
* (Q4**3pts**) Provide Screenshot of Wireshark HTTP packet with image file content data (i.e., not the actual image, but the image’s binary data as might be sent in a packet):



* (Q5**2pts**) Main Protocols used to ping your network gateway:

 The different protocols that were used to send and received information from gateway protocols are , SSDP, IGMPv2, ARP, ICMP

* (Q6**1pt**) Were the credentials sent to ww.postalgate.com in plaintext or encrypted format?:
* Plain text
* (Q7**3pts**) Provide Screenshot of the Wireshark window showing the ebscohost credentials embedded within the packet:



# Application Exemplar 2: Using Hash Functions to Verify Integrity& Passwords

A hash function is a method of converting a message or data into a numeric value called a *hash value, checksum* or *digest* (it’s a form of *digital fingerprint*) To verify data integrity after transmission, the hash value on the sender’s end can be compared with the hash value on the recipient’s end. A match between the hash values indicates that data hasn’t been tampered with during transmission.

In this exercise, you will use the HashPrint v1.2 utility which implements many popular hash algorithms, namely, the SHA-1 and SHA-2 families (Secure Hash Algorithm), and the MD5 MDA (Message Digest Version 5 Algorithm).

Note: Both the MD5 and SHA1 algorithms have been found to have security vulnerabilities, and while still widely in use they are often supported by newer algorithms such as the SHA2 set (SHA256, SHA512).

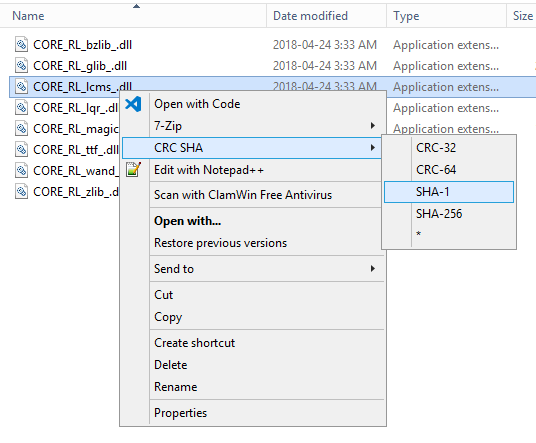
**A. Preparation**

* Download and install the HashPrint tool from:<http://www.postalgate.com/downloads/hashprint.zip>
* Unzip the package into a temporary directory and run setup.exe.
* The file is safe; you may receive warnings, but you can safely continue to install HashPrint

**A. Calculate the SHA and MDA Checksum for the car-image.bmp file:**

**What:**Download a file from the Internet and determine if it was tampered with by comparing SHA256 Hashes. Since both the common algorithms MD5 and SHA-1 are now considered deprecated, we will use only the more secure hash algorithms from the SHA-2 family.

**How:**

* Download the file<http://www.postalgate.com/downloads/moby.zip>
* How can we tell if this is the original file, and that it has not been tampered with (for example, via man-in-the-middle attack)? We use the following process:  
  + Download the file but do not use it yet;
  + Copy the file author’s posted digital fingerprint (in this case, SHA256) from the download location;
  + Use a tool (HashPrint) to compute the SHA256 hash for the downloaded file;
  + Compare the computed value to the posted value. If they match, we can assume the file has not been tampered with
  + You will be required to use two approaches, 1) via the HashPrint tool and 2) via the command line
* Approach 1: HashPrint  
  + Launch HashPrint
  + Drag and drop your downloaded file “moby.zip” onto the tool window
  + HashPrint will automatically calculate 6 different common hash algorithm digests
  + Now, visit the page <http://www.postalgate.com/downloads/fingerprints.html>
  + In the provided table, locate the moby.zip filename and copy the SHA256 fingerprint
  + Paste the value into the “Test a Hash” field in the tool window
  + Is the hash valid or invalid?**Paste a screenshot of the tool window (showing match/no match status) into the worksheet below and answer the questions on the worksheet.**
* Approach 2: Windows 10 Command Line  
  + As computing file hashes is a common operation, most major OS will have a built-in facility to do so. In Windows 10, you can use the built-in shell command, “certutil”:
  + certutil –hashfile moby.zipsha256
  + Compare the generated hash to the listed hash at <http://www.postalgate.com/downloads/fingerprints>
  + Do the hashes match? **Paste a screenshot of the command window (showing the generated hash) into the worksheet below and answer the questions on the worksheet.**
* Approach 2: Alternatives for other OS
* For **Linux**, use the built-in shell command:  
  + wget http://www.postalgate.com/downloads/moby.zip
  + sha256sum < moby.zip
* For Mac, from a command line use the built-in tool shasum (-a specifies algorithm):   
  + shasum -a 256 moby.zip
* For **Windows Server 2012+**, you can right click on the file and select CRC-SHA -> SHA1 to generate the hash quickly.  
    
  
* As you can see, generating hashes is easily done in all major OS and should be a regular security control.

**B. Password Hashes and SALTing:**

* **Part 1: Hashing Passwords**
* For this workshop, you will have an opportunity to hash, crack and salt passwords. **You will be required to record some screenshots and answer some questions in the worksheet below.**
* Open the hashing tool “**HashPrint**”. Make sure all input fields are clear.
* An April 2019 study done by CNN found the top 3 most common passwords to be:  
  + qwerty
  + password
  + 111111 (six 1s)
* Using HashPrint, hash “qwerty” (without the quotes). Cut and paste the (weak) MD5 hash into <https://www.crackstation.net> andclick “**Crack Hashes**”.**HINT:** In HashPrint, click on the label “MD5” to copy the hash into the clipboard, without the “:” separator characters.**Record your answer to the following questions in the worksheet below:**  
  + Crackstation.net uses a dictionary attack where all possibilities are pre-computed by hashing all possible values, and then providing a look-up table that maps the hash value back to the original password. What resource does a dictionary attack require to have a better chance of success (Computing Time, Memory, Storage Space, or Bandwidth)
  + ***COMPUTING TIME***
  + Was the MD5 hash for “qwerty” in the Crack Station lookup table?
  + D8578EDF8458CE06FBC5BB76A58C5CA4
  + You may think that your answer for the previous question is due to MD5 being a weak and deprecated algorithm. As computing power increases, older algorithms are expired. But it may also be because “qwerty” is a common password, thus more likely to be hashed for a dictionary attack. Most dictionary attacks start with a list of common passwords as their initial data. Repeatedly hash “qwerty” and see if it can be cracked on Crackstation.net, for SHA1, SHA256, SHA384, and SHA512. Do not use RIPE as Crack Station does not provide support for RIPEMD160.
  + Repeat the above steps with the next 2 passwords and record your results. Are there any hash algorithms which are not vulnerable to a dictionary attack for these passwords?
  + **Record your results into the worksheet below.**
* **Part 2: SALTing Passwords**
* SALTing passwords is the act of concatenating a unique string to a password prior to hash generation, to make dictionary attacks less effective.  
  + In HashPrint, once again enter the password “qwerty” to be hashed.
  + In the SALT field, add a SALT value of “123”.
  + Again, copy and paste the value (click the MD5 label) into Crack Station and apply the dictionary attack.
  + **Paste a screenshot of the results**. Did SALTing the password make it immune to the dictionary attack? Why or why not?
  + Let’s try a stronger SALT with higher entropy: SALT the password with “**A1$3!sq%”.** What was the result this time, and why?**Paste a screenshot of the results.**

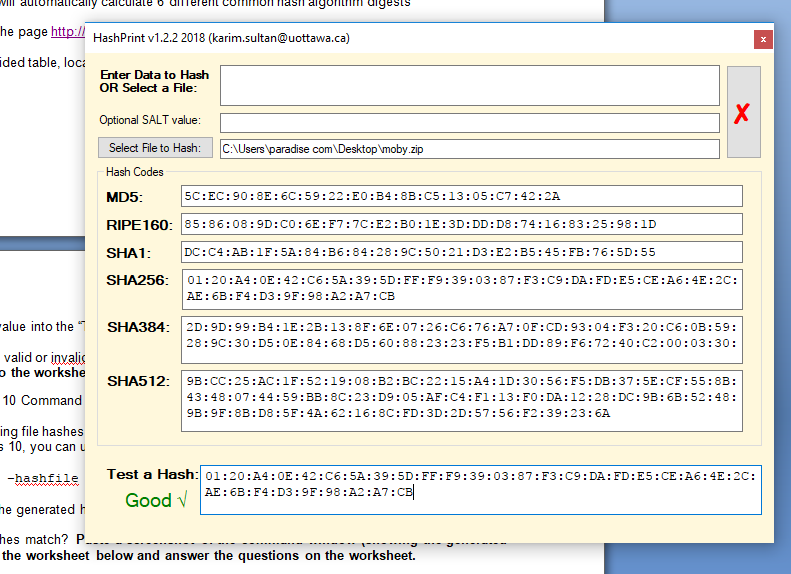
**C. Password Cracking**

* A website has leaked information about user accounts and passwords via an SQL injection attack. The result of the SQL Injection attack can be found at <http://www.postalgate.com/admin/users.html>  
  + For each entry in the table, you need to:  
    1. Identify the hash algorithm being used (hint: count bytes in fingerprint; there are 8 bits per byte). RIPE is not being used. Example:  
       1. “80:65:C9:46:72:60:69:2F:7F:B1:6C:08:97:DE:03:12:5D:4E:0F:14” has 20 bytes in it (each hexadecimal number represents one byte). Therefore, this is a 160 (8 bits \* 20 bytes) bit hash. Only RIPE and SHA1 have 160 bit hashes, and RIPE is not being used so this must be a SHA1 hash. Note that SHA-1 is considered to be deprecated as well as MD5 and is not recommended.
    2. Run the dictionary attack against the password hash. NOTE: CrackStation can not process hashes with “:” characters in them.
    3. **Record your responses in the worksheet below.**
  + How many of the discovered hashed passwords were susceptible to a dictionary attack?
  + Five of the hashed passwords are vulnerable to dictionary attack only id 105 remain to be immune from the attack
  + Suggest 3 security controls that would improve the security of the passwords on this site.
* **D. Observations**
  + What is the difference between a password hash, and a file hash? Should file hashes be salted?
  + In HashPrint, enter “UOttawa” as your data. Observe the SHA512 output field and add “.” to the data 3 times. Each time a period character is added to the data, note how the SHA512 hash varies drastically from the previous one. This is known as the “Avalanche Effect” (often called the “Cascade” property) and it is highly desired in any cryptographically secure hash algorithm. Why is the Avalanche Effect so important?
  + It provides non-correlation between input and output, which is significant for a good hash function.Also if there will be a small change small change in the input it will be reflected in the output as well, which made it statistically indistinguishable from random. A hash function that does not do this is considered to have poor randomization, which would be easy to break by hackers.
  + What does it mean when one says, “The file hash guarantees Integrity?” Define Integrity in this context.
* **Additional References**
  + For a good discussion on hashing security and salting, please see:<https://crackstation.net/hashing-security.htm>
  + Curious as to what dictionaries are used to pre-compute hashes? You can download them from: <https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm>
  + Need to calculate more Hashes in more algorithms than HashPrint supports? Try this online tool: <https://defuse.ca/checksums.htm>

## Worksheet 2: Using Hash Functions to Verify File Integrity& Passwords

(**30pts**) Follow the instructions in the Exemplar to answer these questions.

(Part A1**3pts**) Paste screenshot of the HashPrint tool window, with the result of verifying the hash for moby.zip:



(Part A2**2pts**) Paste a screenshot of the OS command window (showing the generated file hash for moby.zip):



(Part B1**15pts**) Password Hashes + Dictionary Attack Cracking

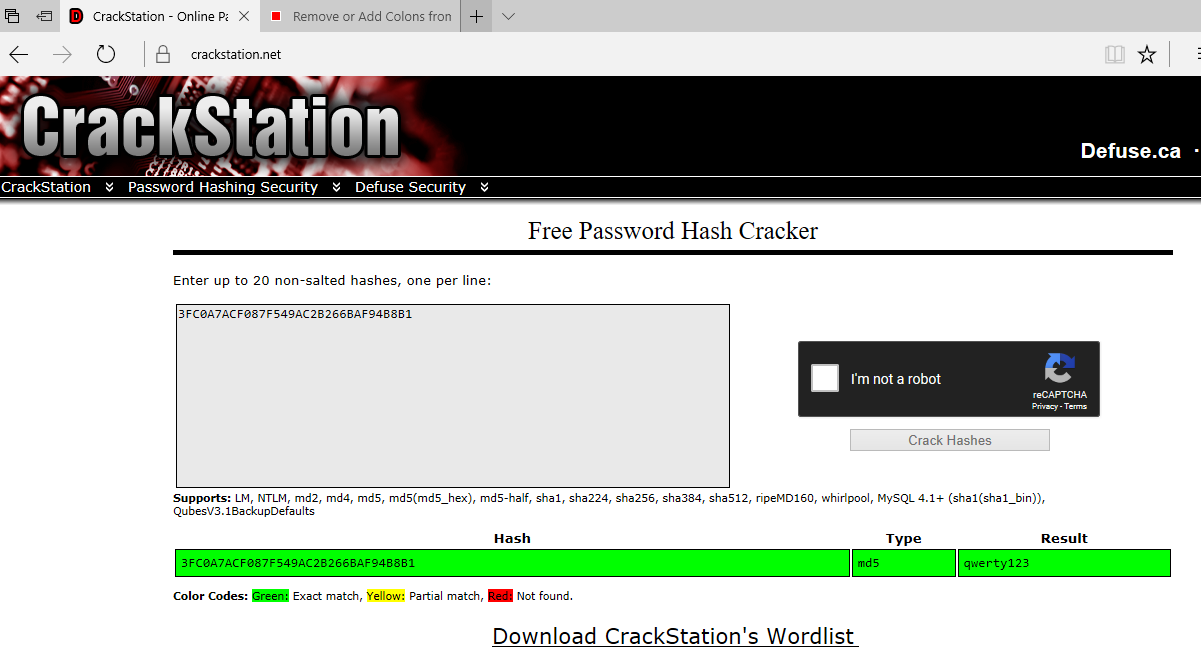
|  |  |  |  |
| --- | --- | --- | --- |
| Common Password | Algorithm | Hash | Crackable? |
| qwerty | MD5 | D8578EDF8458CE06FBC5BB76A58C5CA4 | Yes (**Exact match**) |
| qwerty | SHA1 | B1B3773A05C0ED0176787A4F1574FF0075F7521E | yes |
| qwerty | SHA256 | 65E84BE33532FB784C48129675F9EFF3A682B27168C0EA744B2CF58EE02337C5 | yes |
| qwerty | SHA384 | 1AB60E110D41A9AAC5E30D086C490819BFE3461B38C76B9602FE9686AA0AA3D2  8C63C96A1019E3788C40A14F4292E50F | yes |
| qwerty | SHA512 | 0DD3E512642C97CA3F747F9A76E374FBDA73F9292823C0313BE9D78ADD7CDD8F  72235AF0C553DD26797E78E1854EDEE0AE002F8ABA074B066DFCE1AF114E32F8 | yes |
| password | MD5 | 5F4DCC3B5AA765D61D8327DEB882CF99 | yes |
| password | SHA1 | 5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8 | yes |
| password | SHA256 | 5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8 | yes |
| password | SHA384 | A8B64BABD0ACA91A59BDBB7761B421D4F2BB38280D3A75BA0F21F2BEBC45583D446C598660C94CE680C47D19C30783A7 | yes |
| password | SHA512 | B109F3BBBC244EB82441917ED06D618B9008DD09B3BEFD1B5E07394C706A8BB980B1D7785E5976EC049B46DF5F1326AF5A2EA6D103FD07C95385FFAB0CACBC86 | yes |
| 111111 | MD5 | 96E79218965EB72C92A549DD5A330112 | yes |
| 111111 | SHA1 | 3D4F2BF07DC1BE38B20CD6E46949A1071F9D0E3D | yes |
| 111111 | SHA256 | BCB15F821479B4D5772BD0CA866C00AD5F926E3580720659CC80D39C9D09802A | yes |
| 111111 | SHA384 | 1B0268A40AE44C012946C974D60BF5291E7BB7C63CDB72A904D9283E3DC0A34DE9AFEBE4035665768AAA503A4E7A30C3 | yes |
| 111111 | SHA512 | B0412597DCEA813655574DC54A5B74967CF85317F0332A2591BE7953A016F8DE56200EB37D5BA593B1E4AA27CEA5CA27100F94DCCD5B04BAE5CADD4454DBA67D | yes |

(Question B1 **2pts**): Are there any hash algorithms which were not vulnerable to the dictionary attack / look-up table crack for the above 3 common passwords?

No, all of the hash algorithms are vulnerable to the dictionary attack for the above mentioned passwords. The look up table for them showed in green color which means that the password matches exactly with the hashes and vice versa.

(Part B2**2pts**) Weak SALT-> Paste a screenshot of the results from Crack Station:

Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list and 123 which is a part of a common password like 123456 whose hash values have been indexed in the database.



(Question B2**2pts**): Did SALTing the password make it immune to the dictionary attack? Why or why not?

No! because Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list and 123 which is a part of a common password like 123456 whose hash values have been indexed in the database, which could be vulnerable to dictionary attack.

(Part B2**2pts**) Strong SALT ->Paste a screenshot of the results from Crack Station:



(Question B2**2pts**): Explain the above results – If different, why?

The crack station could not interpret the hash value this time .there could be some good reason for it

1. Unique value and characters which is difficult to be the part of list of common passwords and databases whose hash values are stored in the rainbow table “A hash table is a large list of pre-computed hashes for commonly used passwords.”

2. It has added long salt value to the password. If a salt is too short, it will be easy for an attacker to create a rainbow table consisting of every possible salt appended to every likely password. Using a long salt ensures that a rainbow table for a database would be prohibitively large

<https://en.wikipedia.org/wiki/Salt_(cryptography)>

(Part C) Password Cracking -> Complete the table:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Name | Hash Algorithm | Cracked Password |
| 101 | **admin** | sha512 | Secure! |
| 102 | **super** | sha384 | MamaMia |
| 103 | **guest** | md5 | cybersecurity |
| 104 | **bob\_slob** | Sha256 | longpassword |
| 105 | **pradeep\_ramani** | **Sha512** | Unknown |
| 106 | **kelly\_green** | **Sha256** | Vegan |

(Question C1) How many of the discovered hashed passwords were susceptible to a dictionary attack?

Five of the hashed passwords are vulnerable to dictionary attack only id 105 remain to be immune from the attack

The block size of sha512 hashed password is 64 byte which is in case of pradeep\_ramani, hence the hash algorithm becomes known.

(Question C2) Suggest 3 security controls that would improve the security of the passwords on this site.

1. Should use at least 2 or more special characters with passwords
2. The length of salts used with password should not be too short
3. Passwords should not be chosen from common databases or common list of passwords.

(Question D1) Observations ->What is the difference between a password hash, and a file hash? Should file hashes be salted?

There is no difference between password hash and a file hash. Yes, they should be salted too as strong salting makes them secure

(Question D2) Observations ->Why is the Avalanche Effect so important?

It provides non-correlation between input and output, which is significant for a good hash function. Also, if there will be a small change small change in the input it will be reflected in the output as well, which made it statistically indistinguishable from random. A hash function that does not do this is considered to have poor randomization, which would be easy to break by hackers.

(Question D3) Observations ->Suggest 3 security controls that would improve the security of the passwords on this site:

ALREADY ANSWERED ABOVE

What does it mean when one says, “The file hash guarantees Integrity?” Define Integrity in this context.

1. Integrity can be defined as that the data must not be changed in transit.
2. Proper steps must be taken to ensure that data cannot be altered.

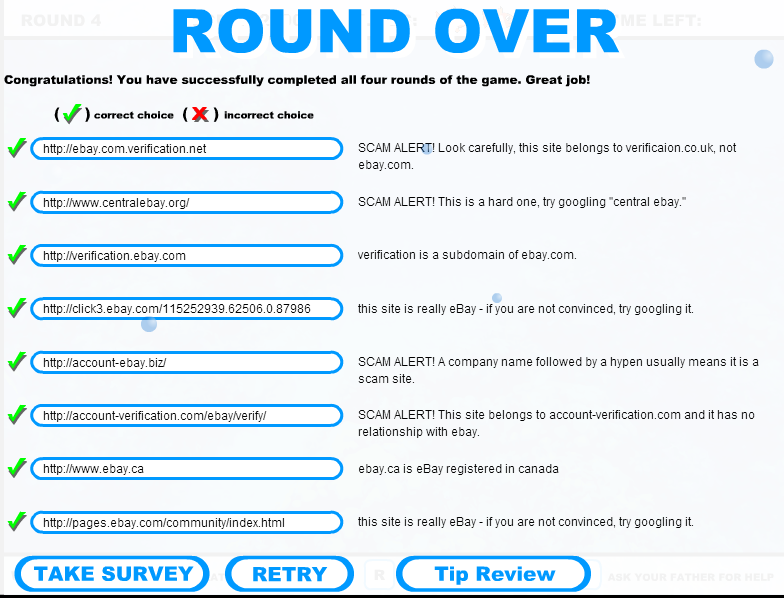
# Application Exemplar 3: Proactive Phishing Safeguards Training

Phishing refers to fraudulent communications designed to deceive consumers into divulging personal, financial, or account information. Websites such as eBay, PayPal and some financial institutions have been common phishing targets in the past. First and foremost, you should get into the habit of checking URLs manually. While this sounds rather simple, there are various techniques used by phishing perpetrators into tricking end-users. So training yourself to recognize such phishing attempts is a necessary first step to safeguard your information. Furthermore, you can use automated tools that work within the realm of your web browser such as security toolbars that can make your life a little easier.

In this exercise, you will complete an ***online training session*** by Carnegie Mellon University and Wombat Securities aimed at creating awareness about phishing among end-users. Delivered in the format of a simple interactive game, you will complete Basic, Intermediate, and Advanced levels of training.

Browse to <http://www.ucl.ac.uk/cert/antiphishing/>to access the Interactive game called Anti-Phishing Phil:

* There are four levels in the game, and you can try the game as many times as you like in order to master the various tips and tricks that allow you to score more (multiple trials of the game are not penalized). You can also freely utilize the help features in the game to improve your scores.
* **NOTE:** This game requires the use of Flash, which most browsers will block. You will need to enable flash for this site and permit the flash app to run on this page in order to participate. Yes, I am aware of the irony of an anti-phishing site requiring you to use Flash. However, I can assure you that the app is safe.
* The requirement for the assignment is to complete the four levels in the game (hopefully, you will have gone through some level multiple times).
* The first part of the ***deliverable*** for this exercise is to list ***5 practices that you can include in your daily browsing routine to help safeguard against phishing attacks***. You will need to do this in the space provided in the solutions worksheet.
* In addition to the above, ***please also include a screenshot of your results page from the last level (level 4)*** which looks something like this:



## Worksheet 3: Anti-Phishing Phil

(**10pts**) Play the Anti-Phishing Game (who says assignments can’t be fun) and answer the questions below.

(Part A1**5pts**)List 5 practices that you can include in your daily browsing routine to help safeguard against phishing attacks:

• Always keep an eye on the address bar to check for spammers like, after // in URL if too many numbers are present then it is like to be a trick.

• Keyword such as verify update delete in the domain space indicates a fraud site.

• Not mandatory but, try to use website that uses https instead of http, here HTTPS referred as “secured”

• Many a times, the phishing e-mails are quite obvious as they have typos capitalize words and special characters.

• Always be suspicious when someone promises you something for nothing. To get ensured about the site authenticity try to search the domain name on Google which shows right after the left forward slash in ,https:// sitename.com/

(Part A2**5pts –** you will lose 1 point for each wrong answer on the final round, and will score 0 if you don’t complete round 4.) Paste a screenshot from your results page of the last round (Level 4) of Anti-Phishing Phil game:

# *C:\Users\paradise com\Desktop\cyberSecurit\assignment3.PNG*Application Exemplar 4: Symmetrical Encryption + Blue Team Case

The Advanced Encryption Standard, or AES, is a symmetrical encryption algorithm: only one private key is used, which must be shared between all parties participating in the message. AES is specified in two variants, AES128 and AES256, with the number representing the size in bits of the key space (i.e., the number of possible keys).

In this exemplar, you will play the role of a white-hat hacker and you will use several online tools to intercept an encrypted message, crack the password, and decrypt the message.

**A. Interception**

* The corporation Postal Gate has hired you to do some white-hack hacking of their system to find potential security problems.One morning in the middle of routine pentesting, the CEO Skypes you. She looks concerned.  
    
  “I have a special task for you today,” she starts. “As you know, we were gearing up to ship our new flagship product, the Widget. But just prior to shipping, 40 crates of our key component, our custom 6502 CPU, went missing from the warehouse. As a result, we had to postpone order fulfillment. This resulted in us losing our largest, oldest and most valued account with F-3 Aerotech. I suspect this may be an inside job, but I have no proof. Security cameras were mysteriously turned off in the warehouse on October 25.I need you to investigate this for me. I suggest you start by accessing <http://www.postalgate.com/warehouse/forum.html> “
* Visit the warehouse forum web page and find the encrypted message.To prove you have found it, you will need to calculate the SHA256 hash of the data and**record it in the worksheet below.**  
  + ***Hint:*** When calculating the SHA256 hash, only use the actual data of the message and do not include any headers / footers / separators.

**B. Decryption**

* Although there are many AES encryption tools available, I suggest the site:  
  + <https://aesencryption.net/>
* Recall that encrypted data can contain binary data (unprintable characters). By convention, Base64 encoding is used -> this converts binary characters into a set of 64 printable ASCII characters, allowing the encrypted binary data to be displayed on screen. Of course, one must remember that the data is actually “Base64 Encoded AES Encryption output”.
* To **decrypt**, one would have to convert the **Base64** ASCII representation back to its original **binary** data form first. But if this was done, then the data would no longer be postable into the above web-form. So to allow this process to happen, the above webform accepts (and outputs) Base64 encoded data. Other tools may require you to convert the data first, hence the suggestion to use AESEncryption.net’s site. Note that the site is using the popular **OpenSSL** library to handle AES128.

* AES typically **requires**: the **data**, the **password**, the**initializationvector**, and the **mode**. You will need to supply the data from your intercept above. AES128 uses a 16 byte password; any password smaller than 128 bits is **padded** to 128bits by the algorithm and any password over 128 bits is **truncated** to 128.  
  + Hence the 16 byte password “SixteenBytesHere” and the 17 byte password “SixteenBytesHere1” will both produce **identicaloutput**. Long passwords are of no advantage here.
  + By default, you can assume **CBCmode** and the **defaultIV**. You will not need to provide either.
* **Testing**: Test the web form by typing the following brief test message:  
  + “Courage and clarity, in thought and statement, intent and action.”
* Enter a password (use your **studentID** number) and click **Encrypt**. Base64 encoded output will appear beneath the form. Paste that output directly into the data fieldand reenter the password if necessary. Click **Decrypt**.  
  + You should obtain your original message. **Take a screenshot and paste it into the worksheet below. The screenshot must show the Base64 encrypted data, the password, and the decrypted output.**
* **Investigation:** In order to decrypt the intercepted message, you will also need the password. AES128 can not be hacked. You will need to determine the password for the data through your investigation.  
  + ***Hint:*** *The CEO reminds you that often, people get lazy, and reuse passwords.*
* **Deliverable:Copy the contents of the decrypted message in the worksheet.**

**C. Action**

* Good job. You have played your role well.This was a “blue team” (defense) operation. Now that you have decrypted the message, you will need to answer a few questions in your report. **Answer the questions in the worksheet below.**

## Worksheet 4: Symmetrical Encryption + Blue Team Case

(**25pts**) Undertake the investigation using the instructions and clues above and respond below;

(Part A1**3pts**)Message intercept🡪 Paste the SHA256 hash for the intercepted message below:

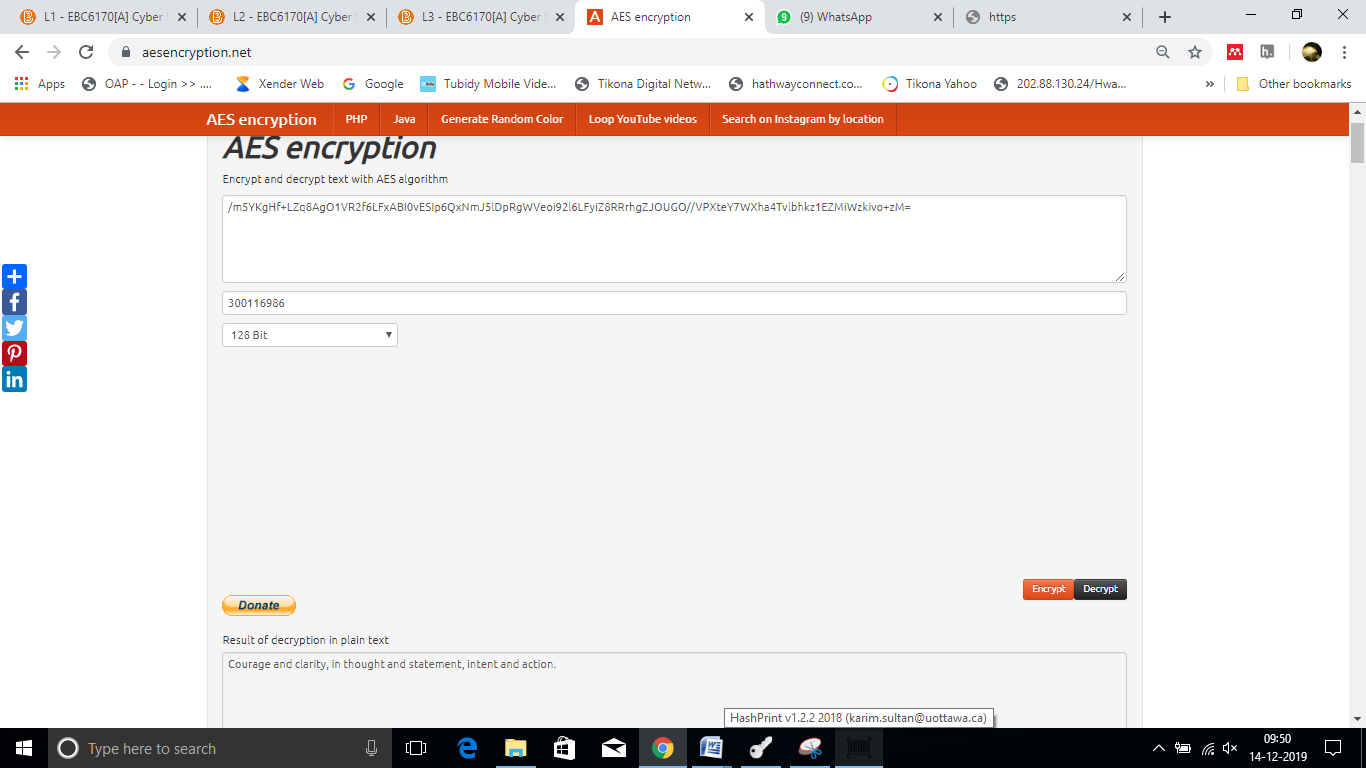
HASH ON PLAINTEXT:

3901ED966886803FCA55D703E8B1EA0D606C231A9A130F195E4CB8F97DDC8BAE

HASH ON ENCRYPTED TEXT:

1BEA9DAA05DCB135D9D70C4806AF63DAEF3A367C7B3F2C6DEE8EF8378BBDE25B

(Part B1**2pts**) AES128 Encryption Test🡪Perform a test Encrypt -> Decrypt process as described. Paste a screenshot showing the Base64 encrypted data, the password, and the decrypted output below:



(Part B2**10pts**) Paste the decrypted message below (paste the text, not a screenshot):

============================================

To: Mario and Luigi

--------------------------------------------

From: Bowser

--------------------------------------------

Subject: Rapid Star 6502 Heist

--------------------------------------------

Yo Mario Bros:

Princess Peach is in the warehouse, with 39

of her friends.

Operation is a go for the 25th.

Security Cameras will be disabled.

All shift crew will be in a special meeting

that Frank has called,to discuss his missing

Porsche.

I can taste the Gizmo success already.

Just be sure to transfer me my price in

shares before the press release.

Bowser Green

============================================

(Part C) Action. Please answer the following questions to the best of your ability:

(C1**2pts**) Culprit🡪 Who does it appear to be is the disgruntled employee working as the inside-person in this case?

Kelly Green appeared to be the disgruntled employee working as the inside person in this case.

(C2**2pts**) Motive 🡪 From the information you have collected, can you infer what the accused’s motive was?

His motive is to earn money in terms of shares.

(C3**2pts**) Impact 🡪 Was there any impact to Postal Gate during this incident? If no, why not? If yes, what?

Yes! There is an evident impact of the incident on the Postal Gate, as the F3- order has cancelled from Postal Gate and they decided to use GIZmos from rival company. Also, Postal Gate has been supply partner for almost 2 decades for F3 Aerotech and they lose their good client due to the theft incident.

There is a possibility that F3 may continue the business relationship with Rapid star which is a loss for Postal Gate.

(C4**2pts**) Controls 🡪 Can you recommend any formal controls that Postal Gate should implement as a result of this incident?

1)Employees should be properly trained and assessed.

2) Emails should be properly tracked and employees should not access the other sites by putting restriction to the other websites, they should have secure business email like outlook

3)Regular meetings should be taken in case of suspicious activity and should not be ignored as in this case Bob replied “what are you trying to pull” instead of taking a necessary action.

(C5**2pts**)Response 🡪**Pick one** of the following responses, and justify why it is the **best option**:

* + 1. The accused should have a stern talking-to and a permanent mark in their employee record should be noted. Postal Gate should write a letter to Rapid Star requesting an apology. Frank should open his eyes and take his Porsche back.
    2. The accused should be suspended immediately, to take time to consider their actions. Pradeep should be offered a role in the software development team. Postal Gate should immediately buy shares in Rapid Star.
    3. The accused should be terminated with cause immediately. Rapid Star should be reported to law enforcement agencies and to government regulators, including the stock exchange on which they are listed. Postal Gate should make a press release disclosure of the breach.
    4. The accused should be terminated, criminally charged, and imprisoned pending trial. Postal Gate should open a civil tort suit against Rapid Star for compensatoryand punitive damages.The CBC should be engaged to film a documentary detailing the breach and highlighting you and your blue team’s success.

The accused ought to be terminated as the company has lost a good client also, the reputation of the company put on stake due to the incident. In addition, proper legal actions should be taken against the rapid star.

# Application Exemplar 5: Asymmetric & Session Key Encryption

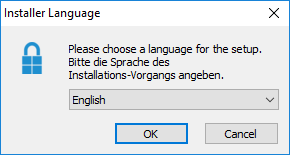
Part of this exercise requires a practice exercise involving two end-users (i.e. you will have to work with another colleague). Each user will use his/her own workstation which will have a local installation of the GPG4WIN application. Gpg4win is an installer package for Windows for eMail and file encryption using the core component GnuPG for Windows. Both relevant cryptography standards are supported, OpenPGP and S/MIME.

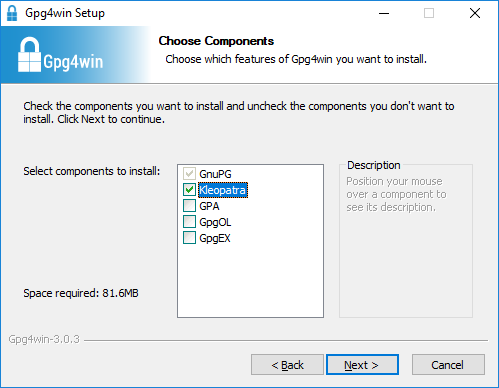
For purposes of highlighting the various steps, we will refer to the two users as User1 and User2.

***A. Installation of GNUPG:***

**GnuPG** is a complete and free implementation of the **OpenPGP** standard as defined by the IETF (RFC4880).User1 and User2 will need to install GNUPG from the **gpg4win-3.1.10.exe** file from <https://www.gpg4win.org/>. Once downloaded, but prior to installation, first check the integrity of the file (SHA1 hashes are at <https://www.gpg4win.org/package-integrity.html>) by repeating the process from Exemplar 2 above.

During the installation, you will have the opportunity to select the components that you want to install. For this assignment,***You only need to install the Kleopartra***component. Kleopatrais a Certificate Manager tool for managing”[X.509](http://en.wikipedia.org/wiki/X.509)” and “[OpenPGP](http://en.wikipedia.org/wiki/Pretty_Good_Privacy#OpenPGP)” certificates.The other components such as GpgOL and GpgEX are useful, but will not be used in the assignment.





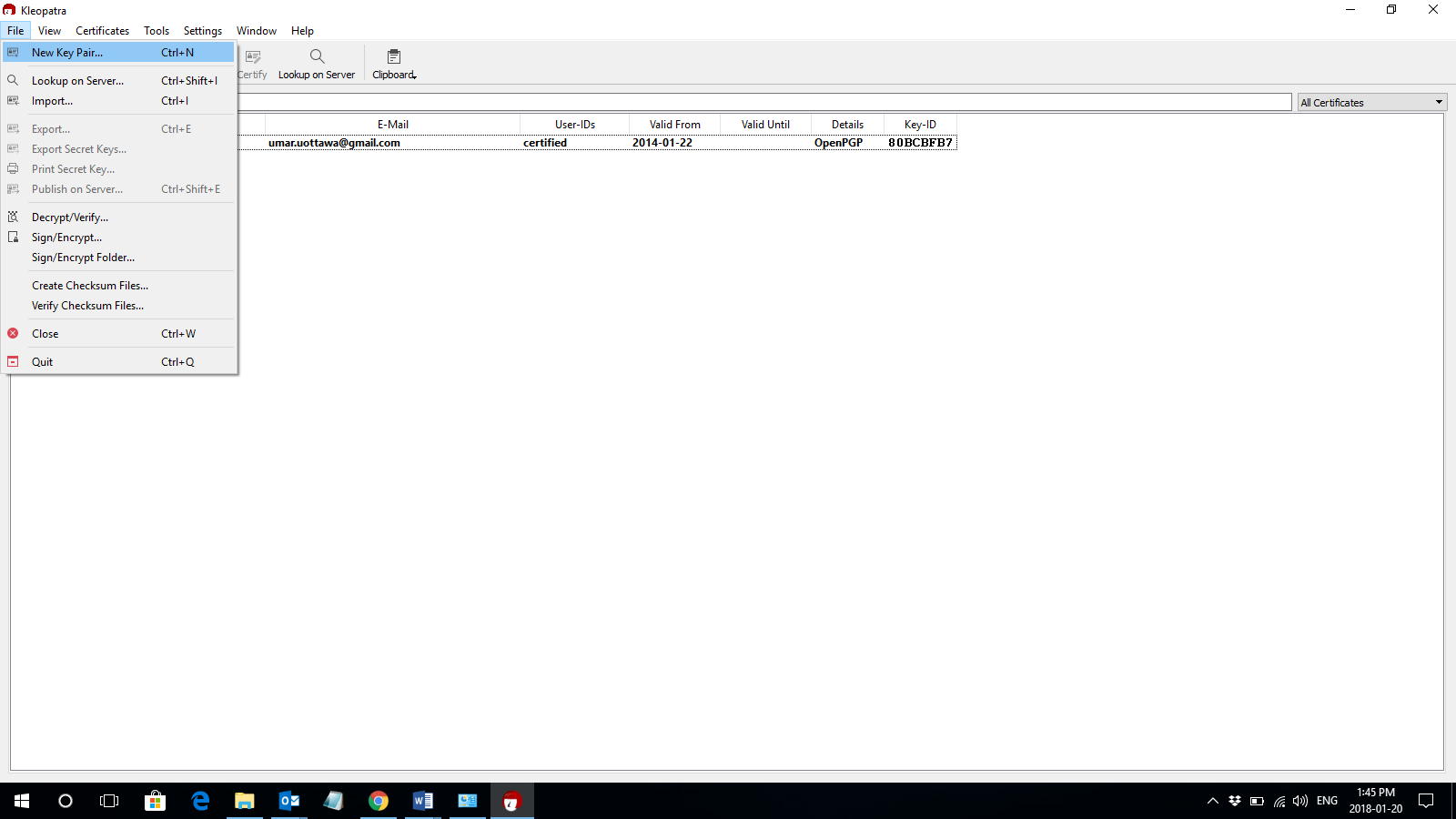
After a successful installation, you should see the Kleopatra icon in your system tray in the windows toolbar. If you don’t see this, you might need to reboot your system and/or manually start it by going to:

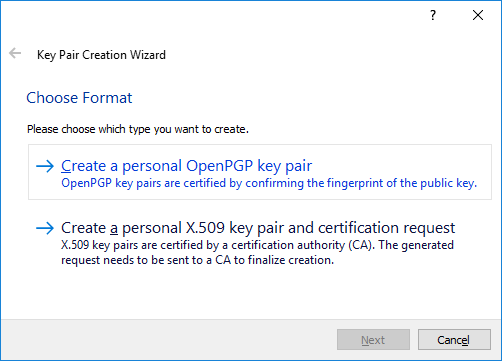
Start 🡺Kleopatra.

***B. Public/Private Key Pair Generation:***

* Once installed, User1 and User2 will need to generate their public/private key pair.
* The public key is shared with colleagues while the private key must remain secret to you only. Anything encrypted with the private key can be decrypted with the public key, and vice versa. To share your public key, an X.509 Digital Certificate is used which binds your public key to your ID. Usually a Trusted Third Party (TTP) vouches for you and signs your certificate, for a fee. In this workshop you will be making “self-signed” certificate instead, and will use an OpenPGP key pair (No CA certification).
* Open the Kleopatra Certificate Managerprogram by clicking the tray icon OR going to:   
  + **Start 🡺 Kleopatra**

* You will be prompted to **generate a new Key Pair**. You can also start the process manually by using the “New Key Pair” option in the File menu of the program as shown below.  
  + For the certificate format, select the ***OpenPGP key pair***. You will be prompted to enter your personal information. Enter your ***Name*** and ***Email*** address. There is no need to look into ***Advanced Settings***, the default options should work fine.

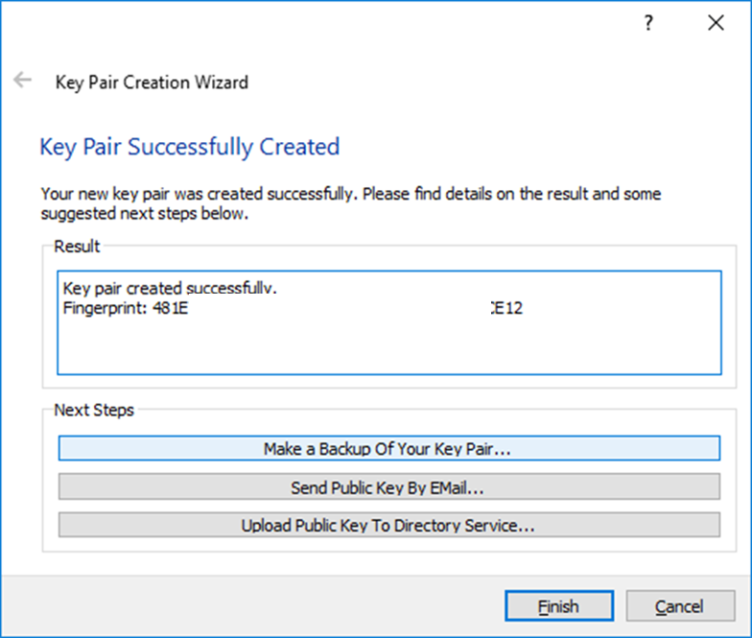




* Make sure you choose a memorable “**pass phrase**” when prompted. You will need this later!

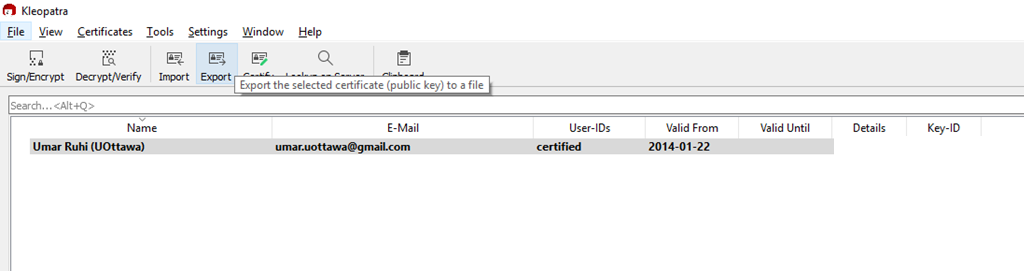
Note: Sometimes the pass phrase entry dialog box gets hidden behind other program windows.   
If you think that the program is stuck at the certificate generation phase, try minimizing your  
program windows.

* When prompted to backup your certificate files, make sure you do so (a **gpg** file will be generated). However, there is **no need to send the certificate by email or to upload the certificate to a directory service**.

******

***C. Export and Exchange your Public Keys:***

* Both User1 and User2 will now exchange their public keys.
* Start the KleopatraCertificate Manager from the desktop tray or Start 🡺 Kleopatra
* Highlight your certificate in the list(there should be only one entry in the list right now), and from the menu bar, select “**Export**”. You will be prompted to save your public key to a file. You will **save the file as an ASC file**.

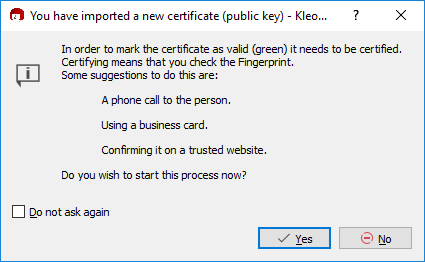


* Send your public key file to the other user via email by attaching the saved ASC file from the previous step.

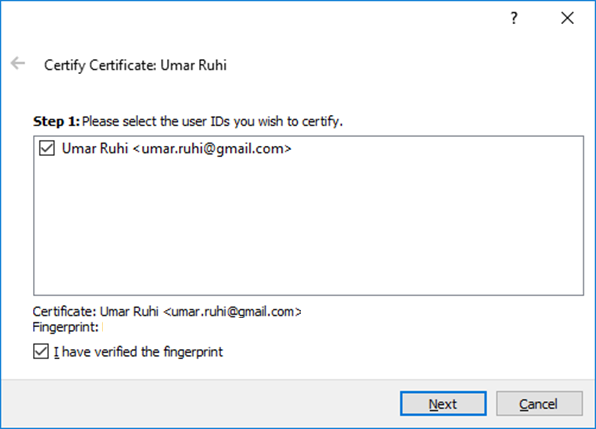
***D. Import Public Keys:***

Once User1 and User2 have each other’s public keys, they can now import it into their own “keyring” by following these steps:

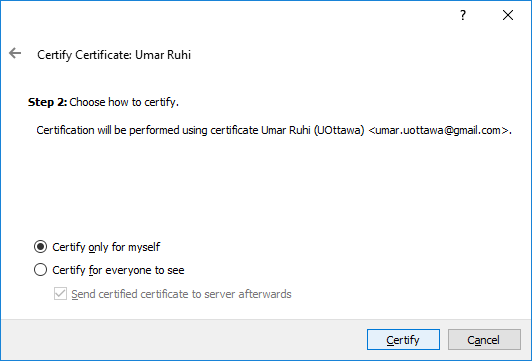
* Save the public key file that you received via email.
* Start the Kleopatra Certificate Manager from the desktop tray or Start 🡺 Kleopatra
* Select“Import” from the menu bar. Select the ASC public key file that was received from the other user. You will be asked to verify the public key through an out-of-band mechanism. For purposes of this assignment, we will assume that you have already done this, and will proceed with the verification step. When prompted, select “Yes” to start the verification process.



* When prompted, make sure to check both the key identifier and the “I have verified the fingerprint” checkboxes.



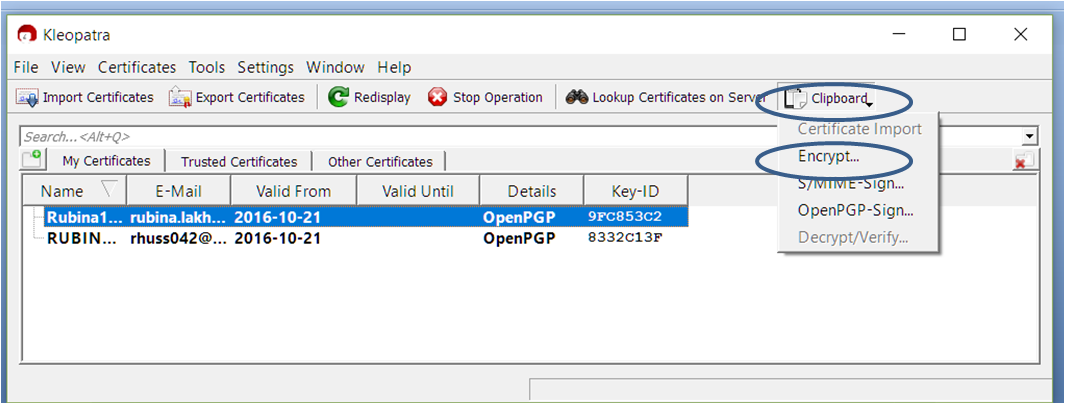
* On the last dialog, selecting “Certify only for myself” is sufficient for this assignment.



* You will be prompted for the passphrase associated with your own private key (generated in the previous step). Once the process is complete, a “Certification Successful” message will be displayed.

***E. Encrypt a Message & Email it to the other User:***

* Create a simple text message (using Notepad or MS Word) and copy it into your Windows clipboard (by using CTRL+C OR by highlighting the text message 🡺 Right Clicking 🡺 Selecting Copy ).
* Right click the Kleopatra icon in the system tray, Select "Clipboard", then "Encrypt":

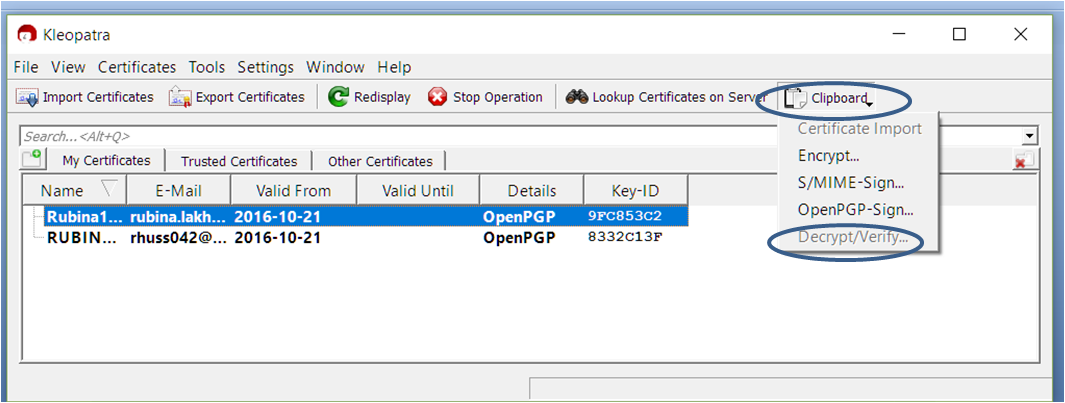


* a popup window will appear asking you to select the recipient of the message.
  + Click on the “**Add Recipient**” button, and in the certificate selection dialog, select “**All Certificates**” from the dropdown menu on the top right side.
  + Once you see the recipient’s certificate (public key) listed, select it and click OK.
  + You will be presented with a warning that **once encrypted, the content of the message may not be decrypted on your end**. This is **fine** since we want the recipient to be able to decrypt the message using his/her private key. Click **Next** to proceed.
  + After a few seconds, you will see an “**All Operations Completed / Encryption Succeeded**" message. Click OK.
* Your **clipboard** now holds the **encrypteddata**, and you can **pasteit into your email message** (by using CTRL+V OR by right clicking the email message text space 🡺 Selecting Paste).
* **Send this encrypted email message to the other user**.

***F. Decrypt a Message:***

Once User1 and User2 receive encrypted messages from each other, they can **decrypt** it by following these steps:

* Select the encrypted text in the received email and copy it into your Windows clipboard (by using CTRL+C OR by highlighting the text message 🡺 Right Clicking 🡺 Selecting Copy ). Make sure you copy the entire block of encrypted message starting at the “-----BEGIN PGP MESSAGE-----“ header and ending with the “-----END PGP MESSAGE-----” footer.
* From the Kleopatra tool, select "**Clipboard**", then "**Decrypt/Verify**".



* You will be prompted for your **passphrase** (since you’re using your private key to decrypt the message).
* You will see a "**Results: All Operations Completed**" message, also notifying you that no signatures were found in the message. This is fine since the sender only encrypted the message and he/she did not sign it. This will be explained further in the next part of the exercise. Click “Finish” to continue.
* Your clipboard now holds the decrypted data, and you can paste it into notepad or a word document (by using CTRL+V OR by right clicking 🡺 Selecting Paste).

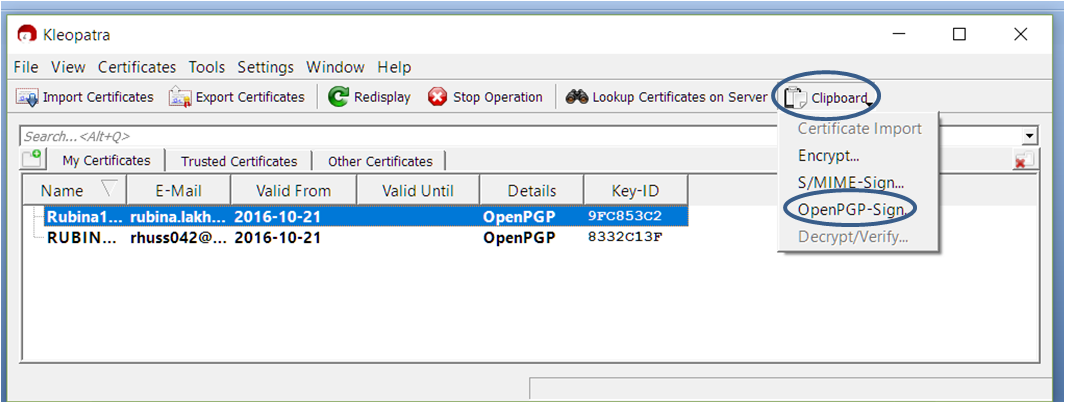
Hopefully, if you succeeded in the above mentioned tasks, you should now have a **good idea of using public keys and private keys for encryption and decryption respectively**, i.e. User1 used User2’s public key to encrypt… and then User2 used his/her own private key to decrypt the data. This procedure can be used to ensure **confidentiality** of data during transmission.

***G. Encrypt and Sign a Message 🡺 Verify and Decrypt a Message***

**Digitally signing** a document uses your private key to compute and encrypt a hash. Only your public key can decrypt and verify the hash. Since only you have access to the private key, this means the document could only have originated from you. This is called **non-repudiation** and is used to show proof of ownership and proof of transaction. A digitally signed transaction cannot be later denied.

To use session key encryption (which combines **confidentiality** with **authentication**):

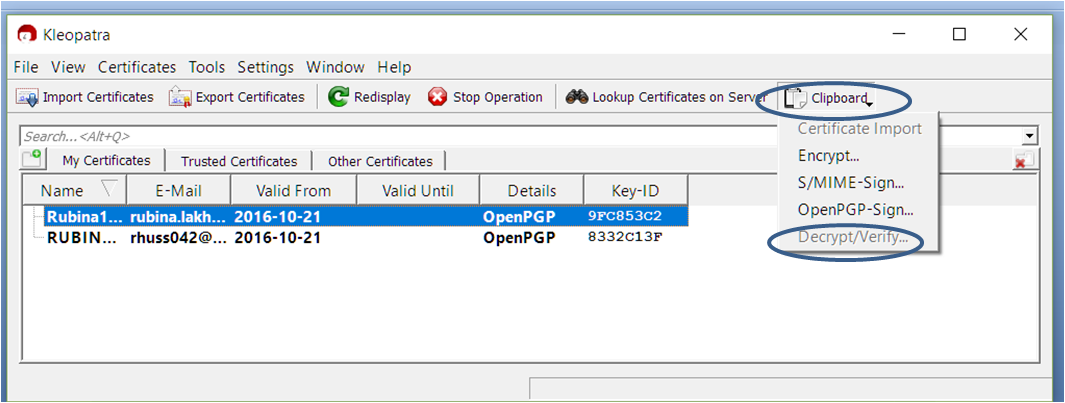
* ***Repeat step F***to encrypt a message and paste the encrypted message into notepad or a word document. Make sure you have the entire block of encrypted message starting at the “-----BEGIN PGP MESSAGE-----“ header and ending with the “-----END PGP MESSAGE-----” footer.
* Now select the entire encrypted message block from the previous step and copy it into your Windows clipboard (by using CTRL+C OR by highlighting the text message 🡺 Right Clicking 🡺 Selecting Copy ).
* From the Kleopatra tool, select"**Clipboard**", then "**OpenPGP-Sign**":



* + a popup window will appear asking you to select the signing certificate for the message.
    - If your own certificate is not currently selected as the default, you will have to specifically select it by clicking the “**Change Signing Certificates**” button. This will open a small dialog box with your own certificate listed. Select **yourcertificate** for signing and click “OK” to close the dialog box.
    - Click “**Next**” to proceed with the signing process.
  + An “All Operations Completed / Signing Successful” message will appear. Click “OK”.
* Your **clipboard now holds the encrypted data enveloped in your signature**, and you can paste it into your email message (by using CTRL+V OR by right clicking the email message text space 🡺 Selecting Paste). Notice that the header of the block now reads: “-----BEGIN PGP SIGNED MESSAGE-----“ and the footer reads: “-----END PGP SIGNATURE-----”. Within the signed block, you can see a separate block with the encrypted message content.
* **Send this encrypted and signed email message to the other user**.

Once User1 and User2 receive messages from each other, they can verify and decrypt it by following these steps:

* Select the encrypted and signed text in the received email and copy it into your Windows clipboard (by using CTRL+C OR by highlighting the text message 🡺 Right Clicking 🡺 Selecting Copy ). Make sure you copy the entire block of encrypted message starting at the “-----BEGIN PGP SIGNED MESSAGE-----“header and ending with the “-----END PGP SIGNATURE-----” footer.
* From the Kleopatra tool, select"**Clipboard**", then "**Decrypt/Verify**".



* In the first step, the signature of the sender will be verified, and the results will show the identity of the sender. Click on the “**Finish**” button to continue.
* **The clipboard now contains only the encrypted portion** of the message (i.e. the **signature portion has been stripped off**). To **verify** this, paste the contents of the clipboard into notepad or a word document (by using CTRL+V OR by right clicking 🡺 Selecting Paste). Notice the header: “-----BEGIN PGP MESSAGE-----“ and footer: “-----END PGP MESSAGE-----”. This is only the encrypted message portion.
* Select the encrypted text from the previous step and copy it into your Windows clipboard (by using CTRL+C OR by highlighting the text message 🡺 Right Clicking 🡺 Selecting Copy ). Make sure you copy the entire block of encrypted message starting at the “-----BEGIN PGP MESSAGE-----“ header and ending with the “-----END PGP MESSAGE-----” footer.
* From the Kleopatra tool, select"**Clipboard**", then "**Decrypt/Verify**".
  + This time the software will simply try to decrypt the data.
  + You will be prompted for your **passphrase** (since you’re using your private key to decrypt the message).
  + You will see a "**Results: All Operations Completed**" message, also notifying you that no signatures were found in the message.
* Your clipboard now holds the **decrypted** data, and you can **paste** it into notepad or a word document (by using CTRL+V OR by right clicking 🡺 Selecting Paste).

The steps outlined above (A to H) should help in getting you familiarized with the **Kleopatracertificatemanager**tool. Now you’re ready to work through the following tasks (to be performed individually by both User1 and User2):

***H.Import, Sign & Trust the Instructor’s Public Key:***

* The goal of **Steps H – J** is to **exchangepublickeys** with the instructor, and to **send** the instructor an **encryptedmessage**.
* You can download the instructor’s public key (a test key for purposes of this assignment) from the following location:
  + <http://www.postalgate.com/downloads/ksultan_public_key.zip>
  + NOTE: You should verify the integrity. A SHA256 hash for the zip file can be found at:  
    <http://www.postalgate.com/downloads/fingerprints.html>
  + You will need to **unzip** the file to get the **asc** file).
  + Save the**asc** file on your computer, and then follow **Step D** from the exercises above to **import** this certificate into Kleopatra.
  + Import and verify the public key for [ksultan2@uottawa.ca](mailto:ksultan2@uottawa.ca)following the same procedure as outlined in **Step D** above.

***I.Share your Public Key***

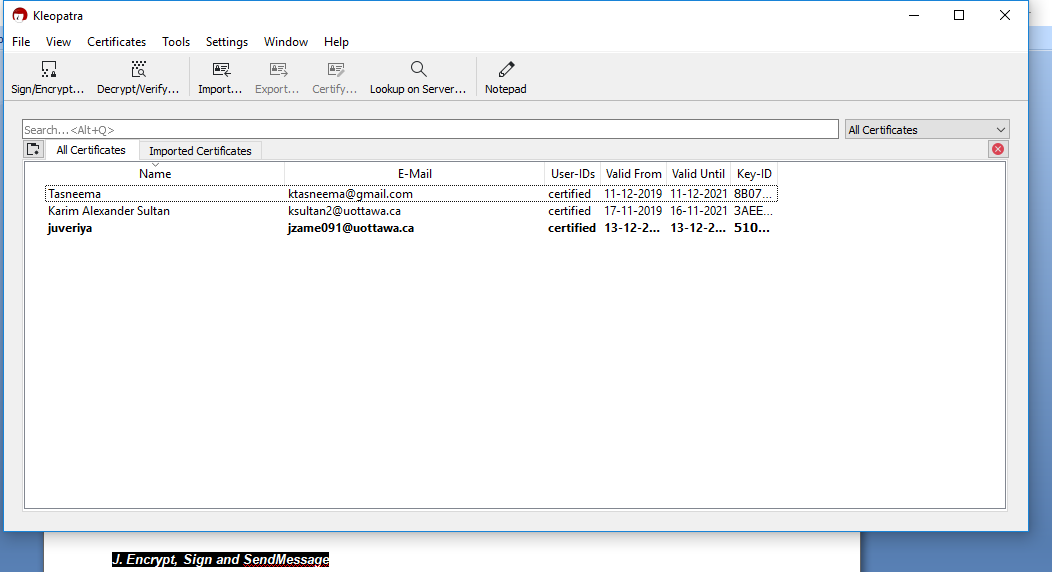
* Send **your own public key** to the instructor’s[ksultan2@uottawa.ca](mailto:ksultan2@uottawa.ca)email address (you would have **exported** your public key in **step C** above).
* **Please make sure that your email subject is as follows: “[EBC6170] Public Key(FirstName LastName) (e.g. “[EBC6170] Public Key (John Doe)”)**.
* **NOTE: Please be very careful not to send me your private key. Should you send me your private key, this task will be considered unsuccessful and no marks will be awarded. Open the ASC file in Notepad and make sure it reads“BEGIN PGP PUBLIC KEY BLOCK”.**

***J. Encrypt, Sign and SendMessage***

* Compose a plain text message with the following information:
  + Date of Transmission of Message
  + Your Full Name and Student ID
  + A Brief Message for your Instructor, including:
    - A random headline from the day’s news
    - Any other message content you’d like to include
* **Encrypt** and **Sign**your message. Make sure you use the Public Key associated with the[ksultan2@uottawa.ca](mailto:ksultan2@uottawa.ca)email address to encrypt the message and your own Private Key to sign the message.
* **Please make sure that your email subject is as follows: “[EBC6170]Encrypted (FirstName LastName)”**
  + **e.g. “[EBC6170]Encrypted (John Doe)”**

## Worksheet 5: Asymmetric & Session Key Encryption

(**10pts**) Checklist of tasks:

1. **(4pts) Downloaded**instructor’s public key, **verified**file integrity, and **imported** instructor’s public key into Kleopatra?**Paste a screenshot of the imported public key in Kleopatra:  
     
   **

1. **(2pts – 0pts if you send Private Key)Sent**your **PublicKey** as specified to [ksultan2@uottawa.ca](mailto:ksultan2@uottawa.ca) ?  
     
     
     
     
   Record Date Completed: \_\_\_\_\_\_\_\_\_12/12/2019\_\_\_\_\_\_\_\_\_\_\_\_\_
2. **(4 pts)**Sent a **Signed**&**EncryptedMessage**using Karim’s public key, as specified, to [ksultan2@uottawa.ca](mailto:ksultan2@uottawa.ca) ?  
     
     
     
   Record Date Completed: \_\_\_\_\_\_\_\_12/12/2019\_\_\_\_\_\_\_\_\_\_\_\_\_\_

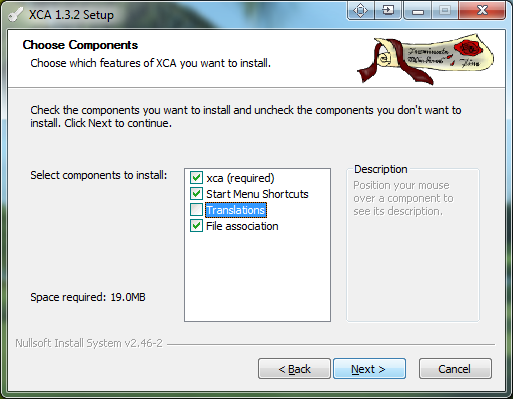
# Application Exemplar 6: Informational Components in an X.509 Digital Certificate

A **certificate** is a **digitallysignedstatement** from one entity (person, company, etc.), **confirming** that **thepublickey** (and some other information) **of** some other **entity** has a particular value. When data is digitally signed, **thesignature can be verified to check its *integrity* and *authenticity***. **Integrity** signifies that the data has not been modified or tampered with, and **authenticity** indicates that the data indeed comes from whomever claims to have created and signed it (recall, **non-repudiation** means that person cannot later deny this action). The purpose of a digital certificate is to bind a public key to an identify.

The objective of this exercise is to familiarize you with the various informational components that make up the industry standard ***X.509 Digital Certificate***. For this exercise you will be generating your own Digital Certificate using the XCA application. Simply follow the instructions provided below.

**A. Install the XCA application from the setup\_xca-1.3.2.exefile:**

* XCA is a tool for managing and creating X.509 certificates. The website for XCA is at:  
  + [https://hohnstaedt.de/xca](https://hohnstaedt.de/xca/)
  + Download XCA v2.1.2 from: <https://hohnstaedt.de/xca/index.php/download> (Windows users should choose the “.exe” version instead of the “portable” version).
  + Please remember to verify the download’s SHA256 hash (it is posted beside the download link)
* During the installation, you can skip the Translations component.

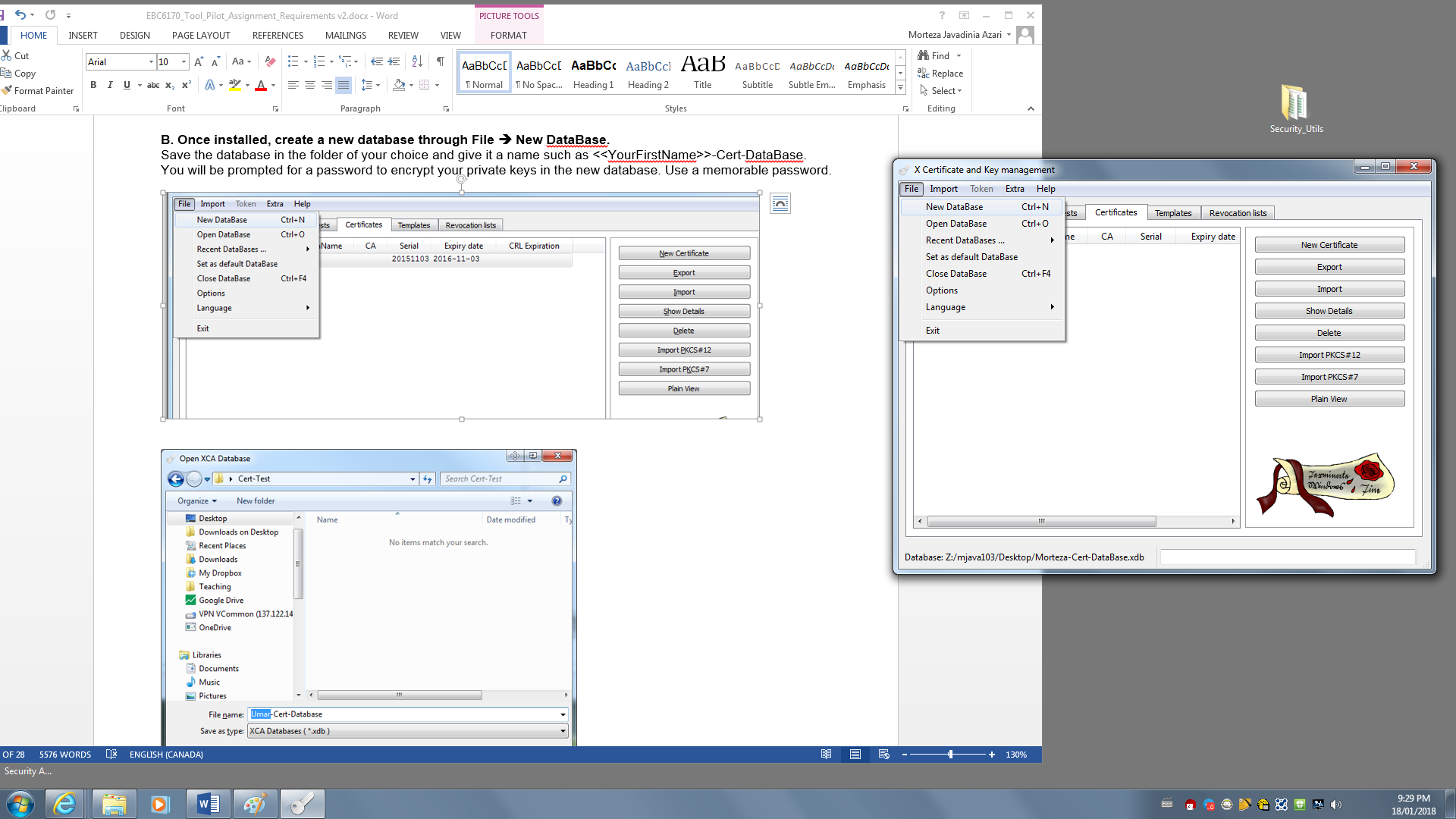


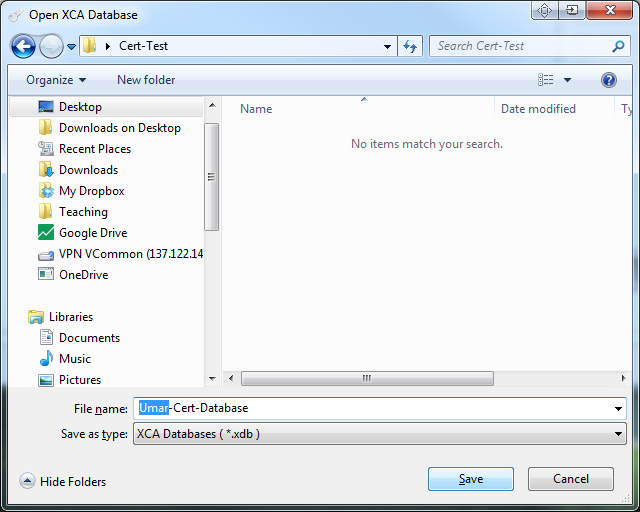
* **NOTE:** As the version numbers have updated, some of these screenshots may be slightly different.

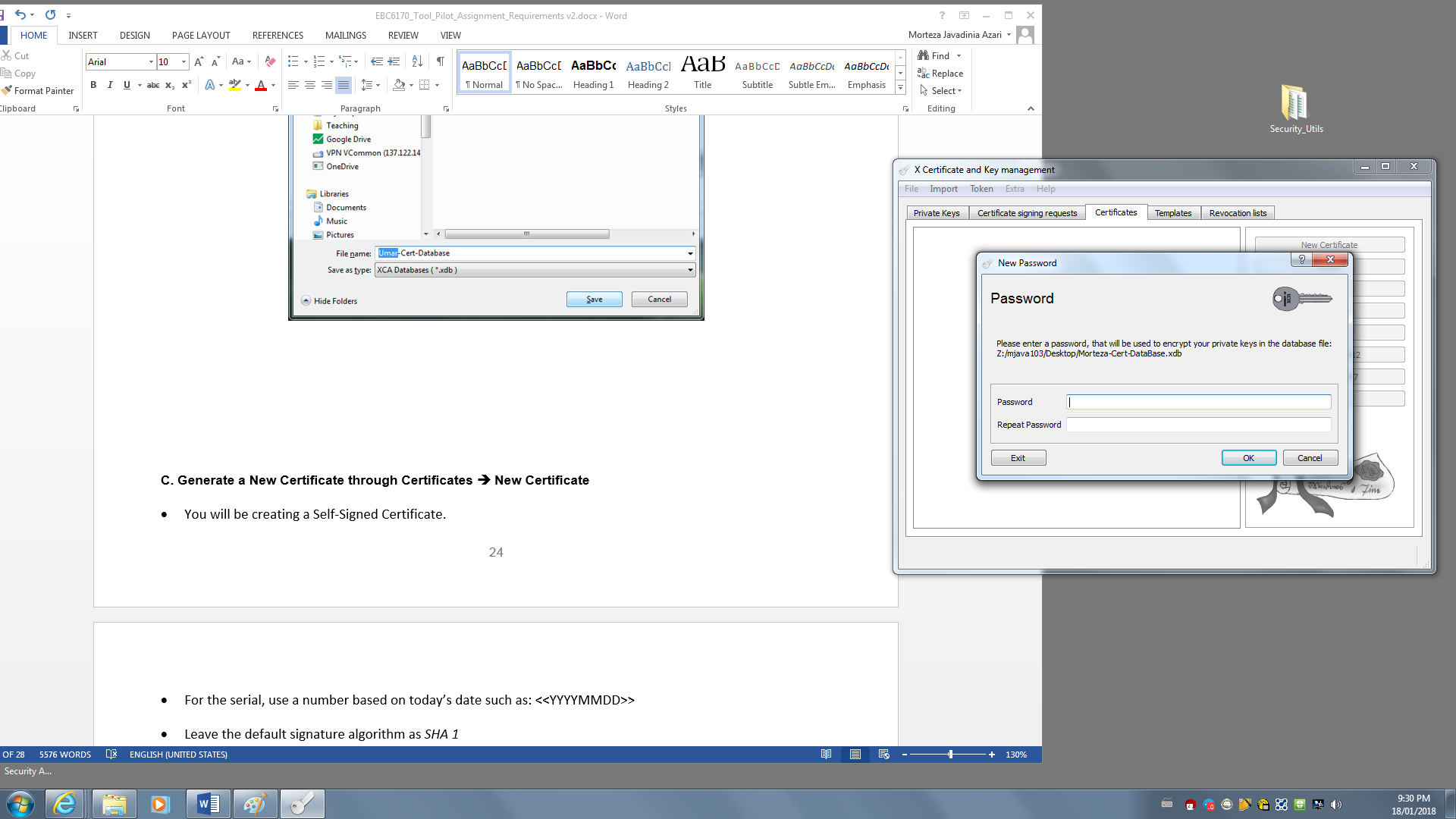
**B. Once installed, create a new database through File 🡺 New DataBase.**

Save the database in the folder of your choice and give it a name such as <<YourFirstName>>-Cert-DataBase.

You will be prompted for a password to encrypt your private keys in the new database. Use a memorable password; do not lose this.

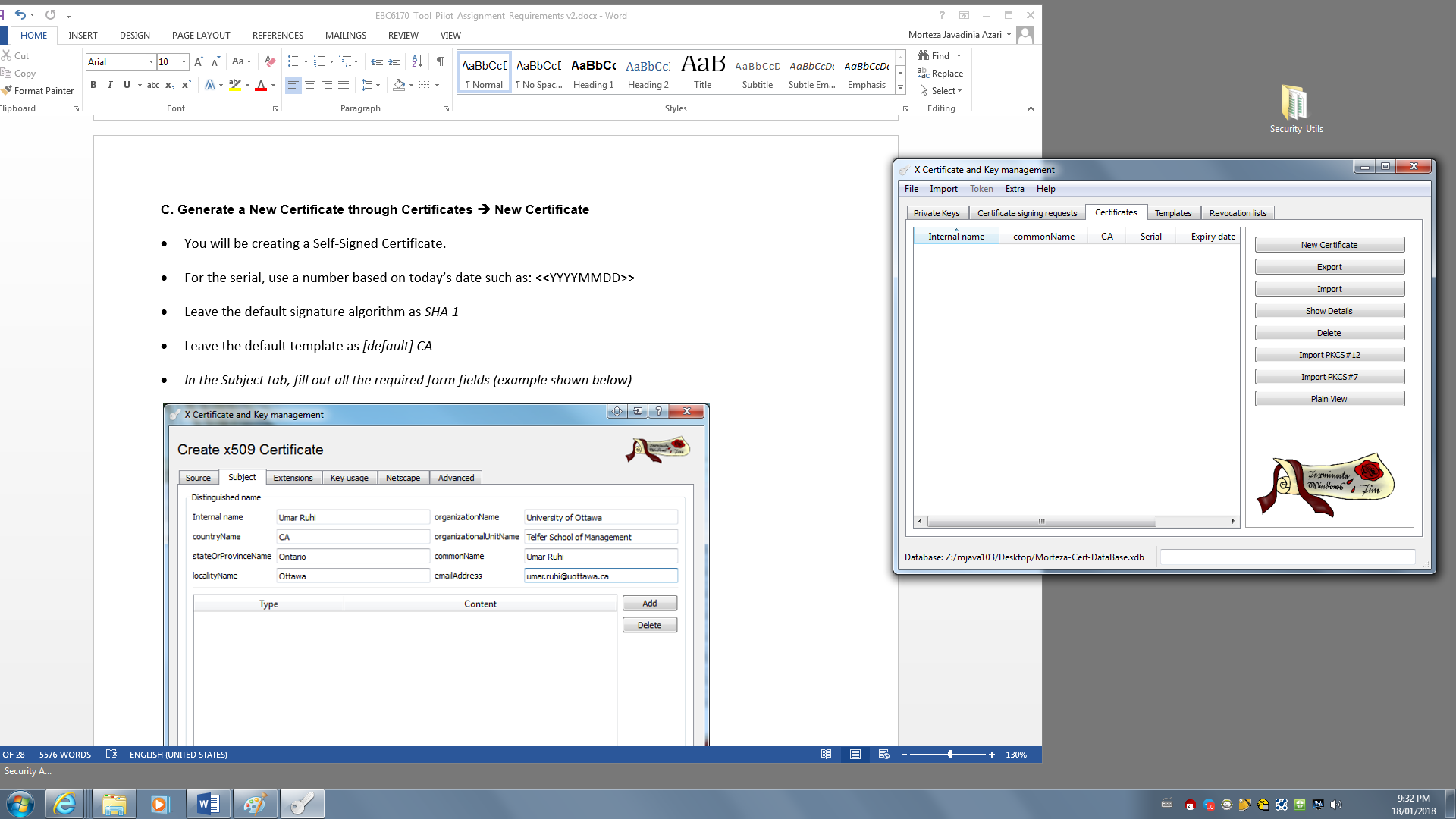






**C. Generate a New Certificate**

* Select **Certificates** 🡺 **New Certificate**



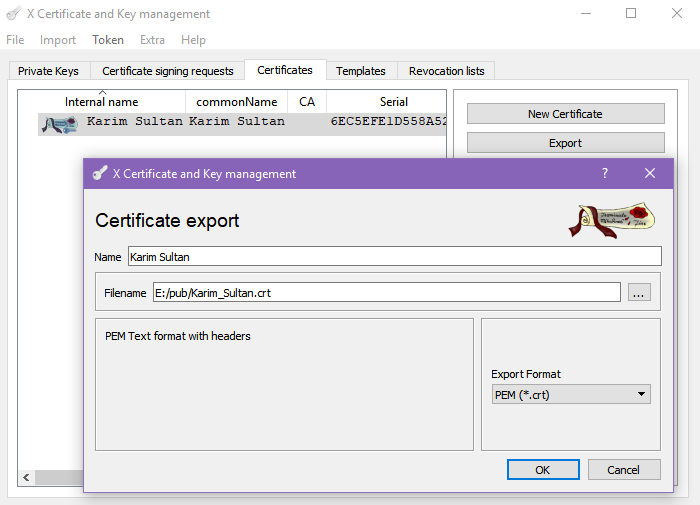
* You will be creating a **Self-Signed Certificate**.
* Leave the default **signaturealgorithm** as **SHA 256**
* Set the default template as **[default] CA**
* *In the Subject tab, fill out all the required form fields (example shown below)*

.

* Click the “Generate a New Key” button and leave the default key type and keysize as **RSA 2048**
* Note that the country name must be the **2 byte** ISO Country Code and not the full name. Despite the use of “CA” here, all other appearances of CA in the certificate mean Certificate Authority and not Canada.
* Once the key is generated, click the **OK** button (Do not select “Remember as default”).
* **RSA** is a standard algorithm used for key generation, based on finding two very large prime numbers. **EC** (**EllipticCurve**) is another common, popular algorithm which uses points on an elliptic curve. When choosing RSA, you need to provide the key size; a minimum of **2048bits** is recommended (when possible, I use **4096**). With EC, you instead need to specify the “**elliptic curve**” to use. There are many named variants that have been cryptanalyticallytested to be secure. Try selecting EC and then observe the large number of named curves (**formulae**) that can be selected from. Note: **Serious companies do NOT use curves provided by the NSA**, such as **DUAL\_EC\_DRBG** or \_**PRNG**🡪 these unfortunately contain **backdoors** and lead to a loss of integrity and trust of the NSAin security circles.

**D. Export the Certificate using Certificates 🡺 Export**

* Select the certificate generated in the previous step and click the Export button.
* Store the certificate in the same folder as the original Database from Step B.
* Feel free to name your certificate whatever you like. The generated certificate should have a **.crt** extension.
* Reviewing the various major **certificate file formats** (PEM, CRT, P7B, P12, PFX, etc…) is beyond the scope of this exemplar. However, PEM (Privacy Enhanced Mail format) is “Base64 encoded ASCII” and therefore is humanly readable. Most of the other versions are binary; some are even encrypted. A professional library like **OpenSSL** provides tools to convert between formats.

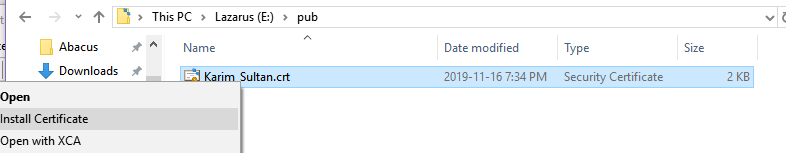


**E. Review the Certificate & Enable Trust**

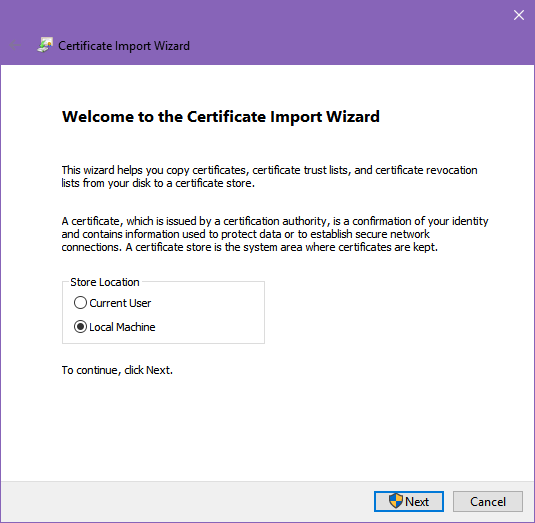
* Once the certificate is generated, you can view it by double-clicking it.
* Verify that all the information you entered during the key generation process is preserved.
* At this point, your certificate is not yet trusted by your computer, and this will be reflected in the general information.
* To have your computer trust the certificate that you just generated, you need to put it in the Trusted Root Certification Authorities store. This will allow you to use the certificate in Windows automatically.
* **Install** the certificate in your computer’s **trusted certificate store**:
  + **Rightclick** on the Certificate file you exported and select “**Install Certificate**”

**Method 1: Right Click and Install (Quick, Less control)**

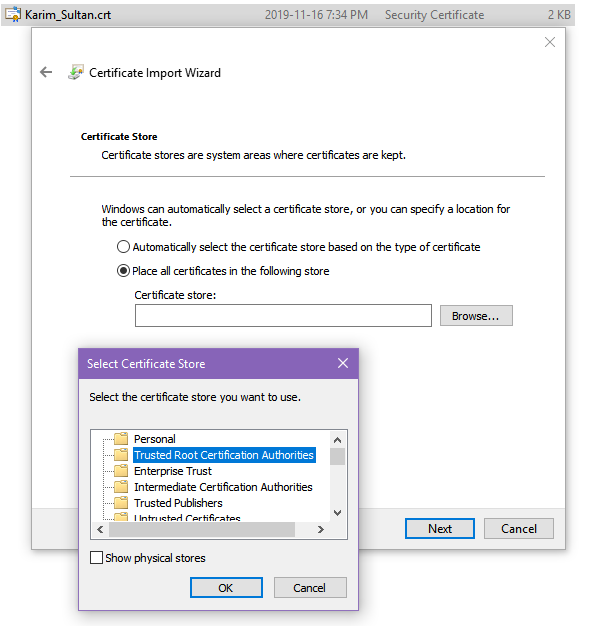
* **Rightclick** on your exported certificate and choose “**Install Certificate**”:



* Select **Local Machine** so it is available to all services:

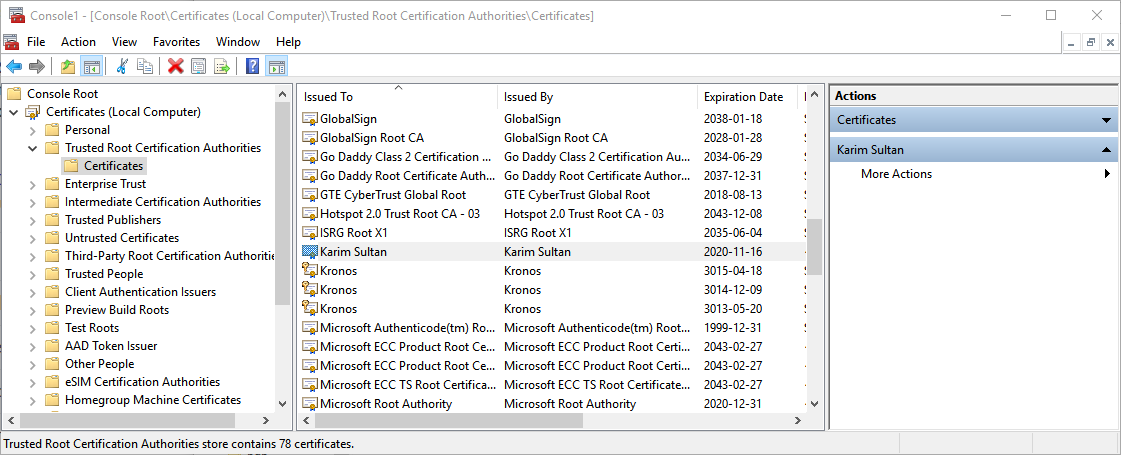


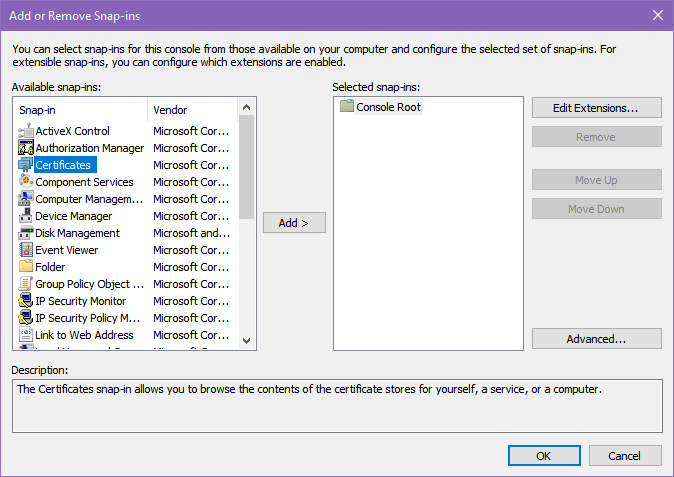
* For **destination**,select “**Place all certificates in the following store**” and **Browse** the Certificate Store to “**Trusted Root Certificate Authoritites**”:



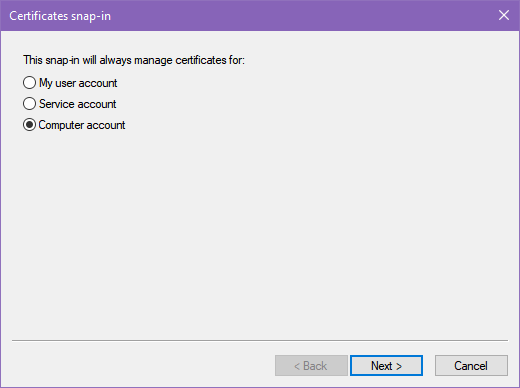
* If everything worked, you will see an **Import Successful** message.

**Viewing Certificates with the Microsoft Management Console, Certificates Snap-In**

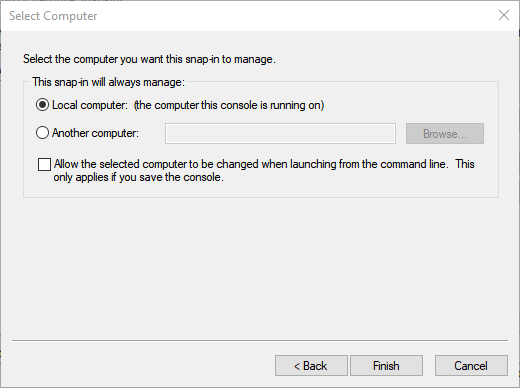
* To view your installed certificate, and all other installed certificates:
  + Use **‘mmc’:** S**tart 🡪 run 🡪 mmc**
  + From menu, choose **File | Add Snap-in…**
  + In dialog, click **Add “Certificates**”
  + Browse through **Trusted Root Certification Authorities| Certificates** until you find your certificate.
  + **Doubleclick** your certificate to view details.
  + From the list of **Trusted Root Certification Authorities**, identify **three companies** that have trusted certificates on your computer and **record them in the worksheet**. Each of these companies are declared trusted, meaning that any certificate that they sign with their key will also be trusted. Obviously, only truly trustworthy companies should have certificates stored here. If a hacker manages to convince someone to install a malicious certificate, it could cause havoc on the system, network and organization.  
      
    
* First, you must open the Microsoft Management Console (mmc):  
  + **Start Menu 🡪 RUN 🡪 mmc**
  + **Select FILE 🡪 Add Snap-In…**
  + In the dialog box, select **Certificates** and click **Add**:



* Select “**ComputerAccount**” (this will make it available to all users and services on your computer):



* Select “**LocalComputer**” if prompted and click **Finish**



* Once the console loads the snap-in, you will see a **tree view on the left pane**. Select **Trusted Root Certification Authorities** and choose “**Certificates**”.
* The**middlecontentpane** will display a **list of certificates** stored in this capacity. You should review who is a **Trusted Root CA** on your machine. Only highly **recognizable security brands** (Comodo, Verisign, Entrust, etc…), the **OS provider** (Microsoft) and your own **personal certificate** should be listed. If you see something like “Warez Malware.ru” or something completely out of place, you should Google it to ensure validity.

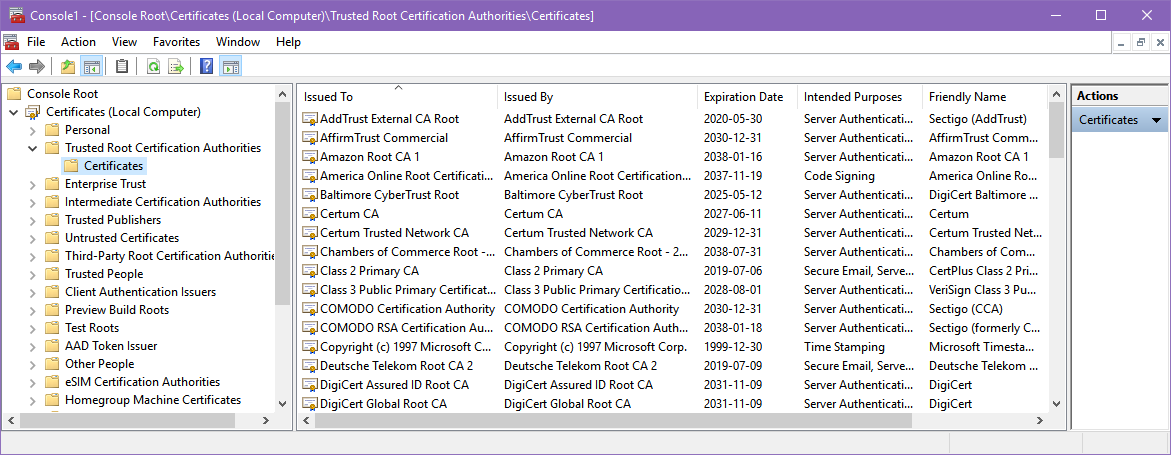


Figure 1 Example output from my personal computer - yours will vary

* **NOTE:** You can use the mmc to install certificates. Select **Action | Import…** from the menu to start a certificate import process. (This is the same effect as right-clicking on a certificate and choosing “install”).

## Worksheet 6: Digital Certificates

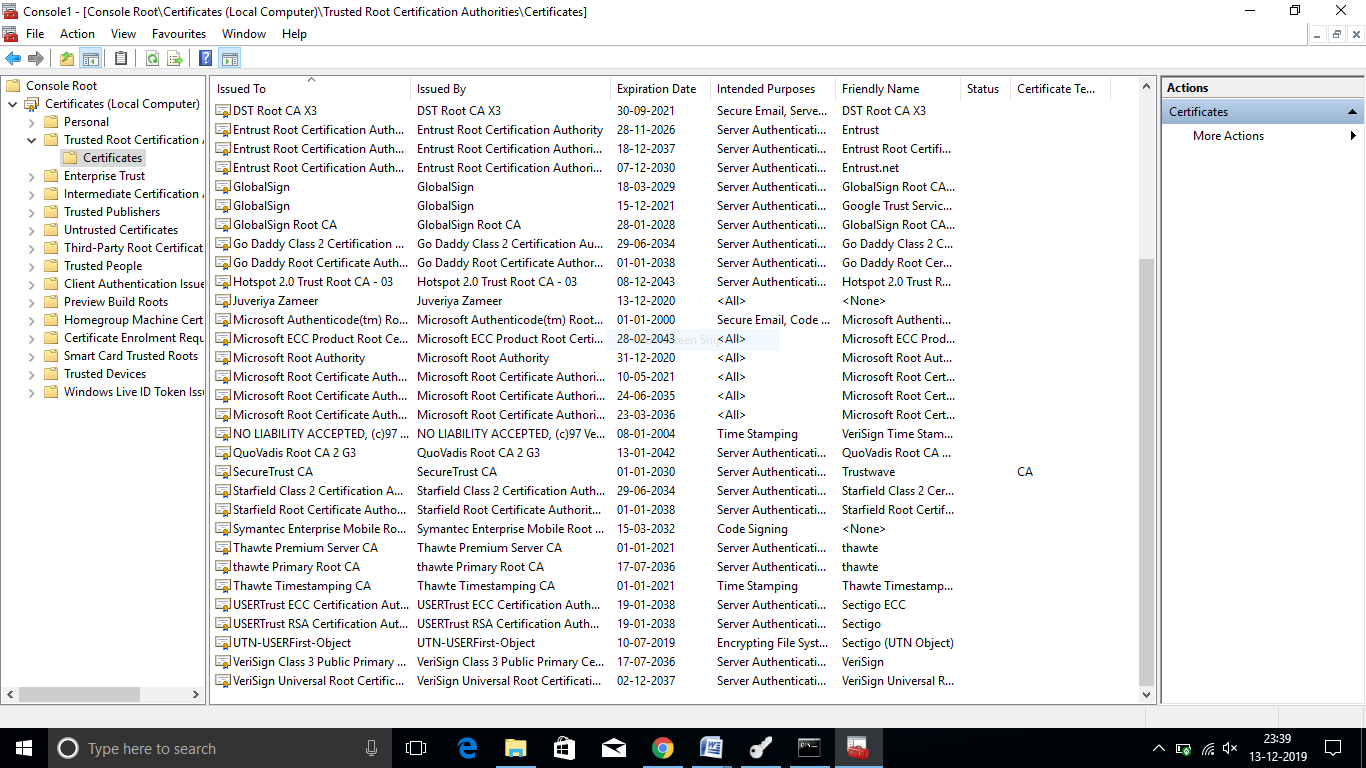
(Part A1**3pts**)**Identify three companies** that hold **trusted rootcertificate authority status** on your computer. Each deployment of Windows comes with at least 3 different corporate CA root certificates installed; but **if** for some reason **you have less than three**, just **include a screenshot** of your Trusted Root Certification Authorities certificates list instead.

1. VeriSign universal certification authority

2. Microsoft root certificate authority

3. Go Daddy root certificate authority G-2

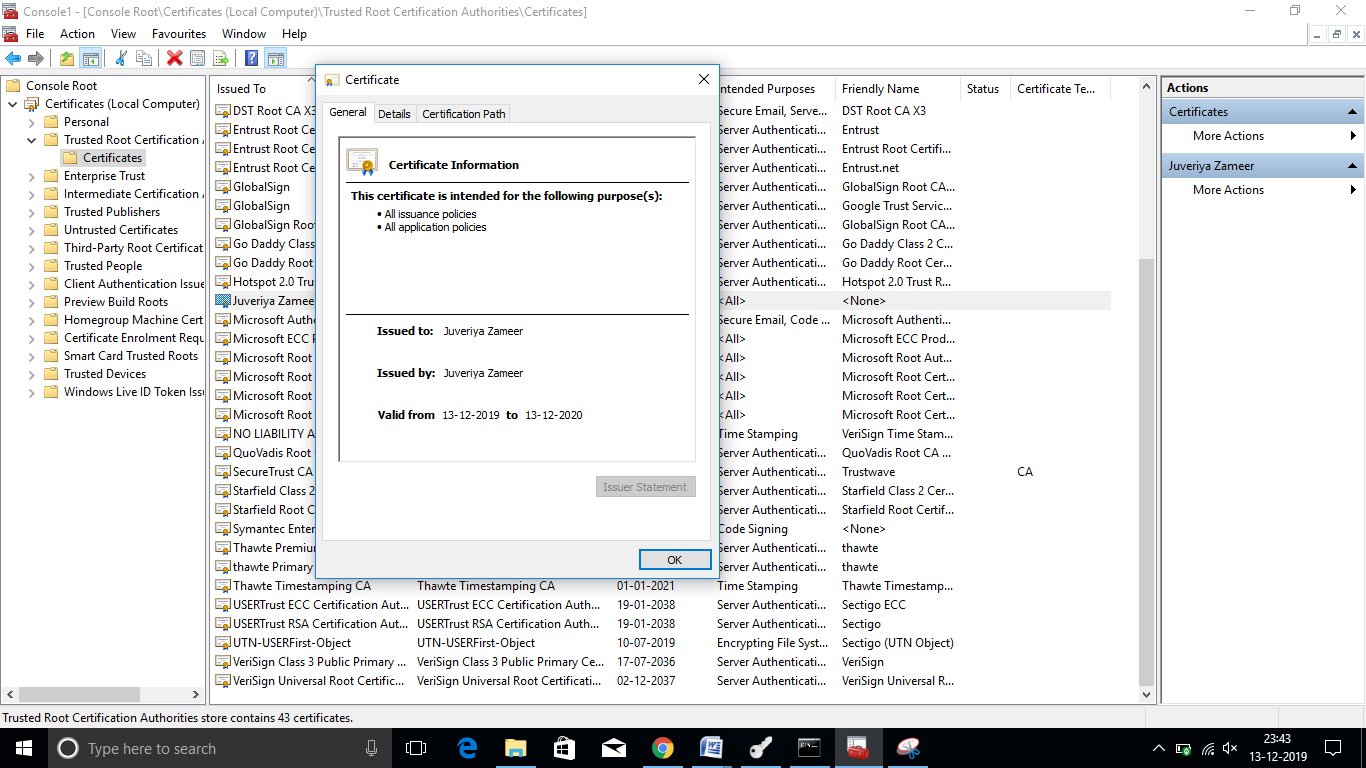
(**or screenshot**)

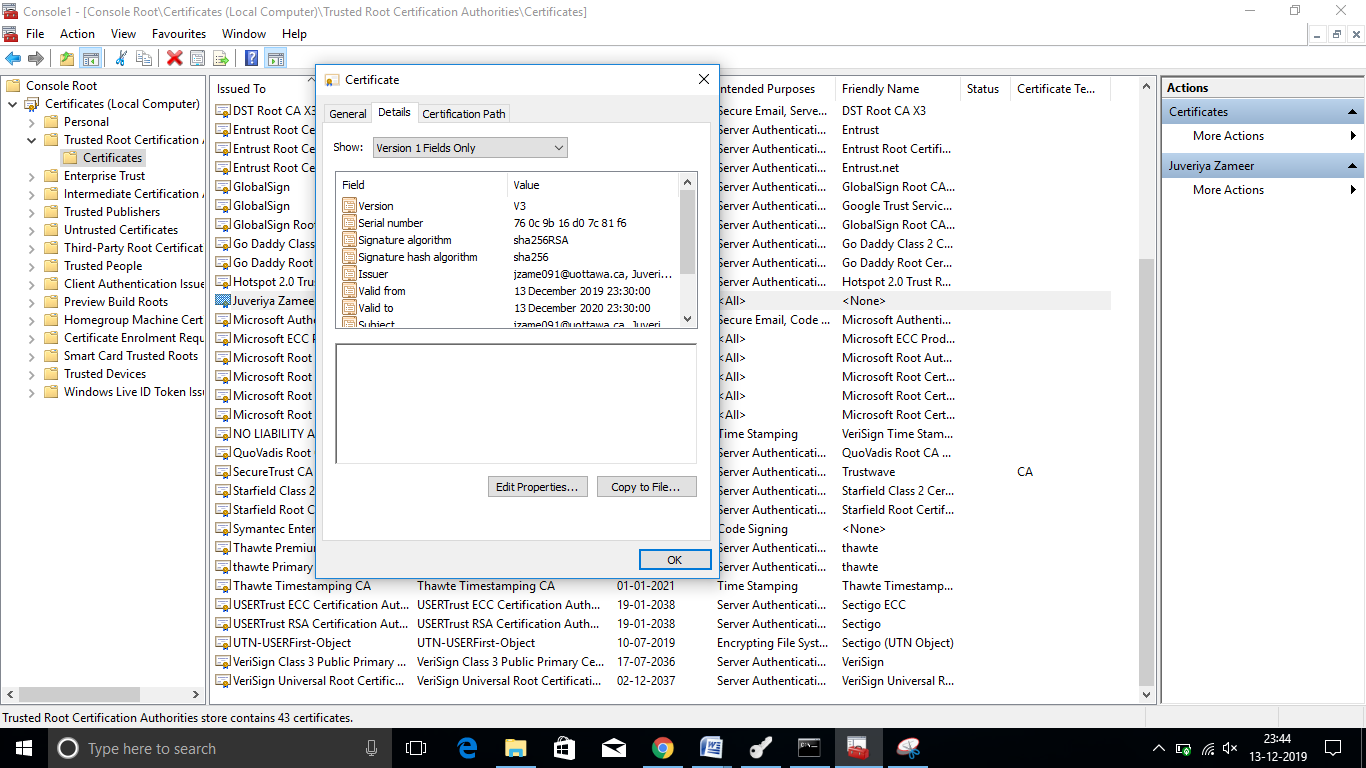


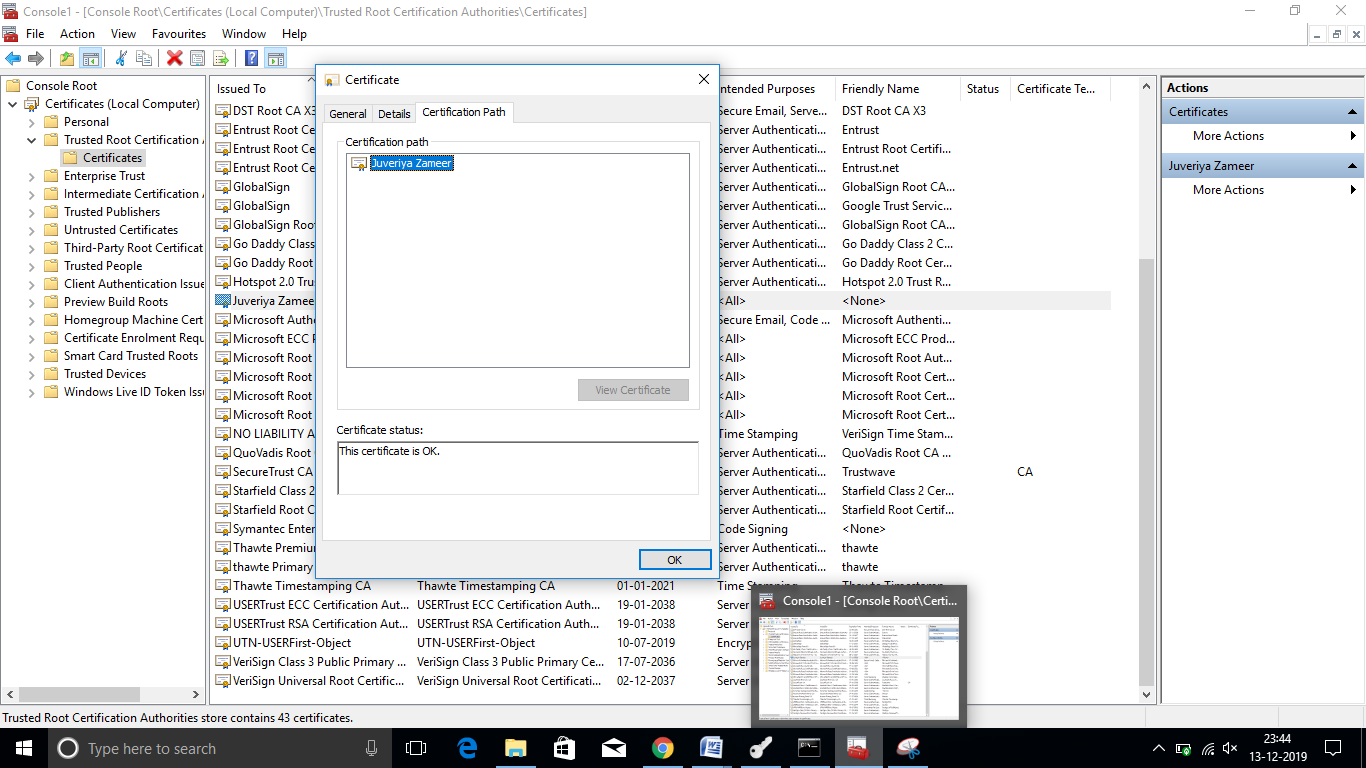
(Part A2 **1pts**) When is it ok to trust a self-signed certificate?  
Self-signed certificates can be used sometimes when the chain of trust starts and ends with me. Also, it is used for development and automated testing of services that require TLS.

(Part A3**6pts**) Certificate 🡪Double click on your exported certificate to **viewdetails** and **provide 3 screenshots** (total) of all the **three tabs** in your certificate including “**General**”, “**Details**” and “**Certification Path**”.

**NOTE**: You can also access your certificate details via the Microsoft Management Console (mmc), Certificates Snap-In as per the above instructions.







# Application Exemplar 7: Other Security Enhancing & Privacy Preserving Tools

(**15pts**) Hopefully, you have found the experience in the previous exercises helpful, and also had fun exploring a variety of end-user tools for enhancing security and privacy at a personal level.

The objective of this exercise is to get you to research other possible security or privacy tools that can be helpful for end-users and/or organizations. You are welcome to research and explore through public search engines or academic literature.

For example, you may choose to investigate Barracuda’s Intrusion Detection System Firewall (Accessibility and Availability), or Cisco’s DNS Round Robin Load Balancer (Availability) or the newest encryption tool (confidentiality, integrity and non-repudiation), etc…

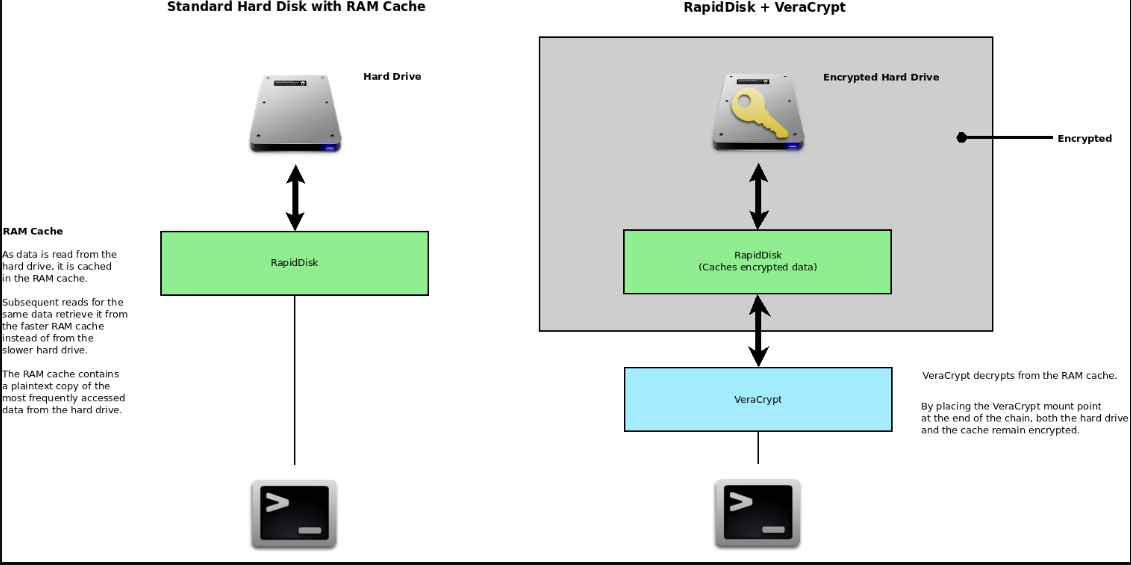
After your exploration, please provide a short summary of ONE new tool (not included in the previous exercises in this assignment or shown in class) that you have used and would recommend to others. In your summary, include:

1. (**3pts**) Name, producer, and main purpose of the tool; VeraCrypt , IDRIX (based in Paris)

Purpose: VeraCrypt is a source-available freeware utility used for on-the-fly encryption (OTFE).[5] It can create a virtual encrypted disk within a file or encrypt a partition[6] or (in Windows) the entire storage device with pre-boot authentication.

Reference <https://en.wikipedia.org/wiki/VeraCrypt>

1. (**3pts**) relevant use-case(s);



1. (**3pts**)benefits of the tool;

- Creates a virtual encrypted disk within a file and mounts it as a real disk.

- Encrypts an entire partition or storage device such as USB flash drive or hard drive.

- Encrypts a partition or drive where Windows is installed (pre-boot authentication).

- Encryption is automatic, real-time (on-the-fly) and transparent.

- Parallelization and pipelining allow data to be read and written as fast as if the drive was not encrypted.

- Encryption can be hardware-accelerated on modern processors.

- Provides plausible deniability, in case an adversary forces you to reveal the password: Hidden volume (steganography) and hidden operating system.

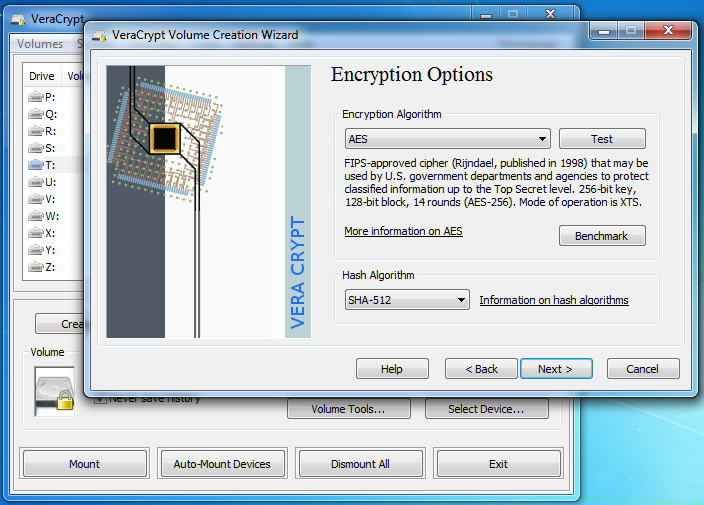
1. (**3pts**)your opinion about the tool;

I really like veracrypt because it allows us to encrypt and store our most important digital files and assets in a safe and effective way. However; veracrypt is mainly at the aesthetic level since I personally do not like the appearance of the program design is very minimalist and a bit chaotic I would like it to have a new interface more in line with all the good that it really offers so There is nothing else that upsets me.

1. (**1pts**)Include at least 1 relevant reference for further reading;

[1][1] T. Medicine, “Encrypt data using VeraCrypt,” vol. 03716461, 2017.

1. (**2pts**) Add 1 relevant screenshot of the tool in use / in demo

VeraCrypt encryption options when creating an encrypted volume

## Worksheet7: Other Security Enhancing & Privacy Preserving Tools

Tool Summary (see instructions in requirements document):