Q1. Maintaining the integrity of sensitive files in my organization using WinMD5, a hashing tool. This project requires that I install the WinMd5 software and using it to verify the integrity of a document I created, confidential_document.txt. With the help of my chrome browser, I installed WinMD5 using from https://www.winmd5.com/, and then I ran the installation file and launched the software on my windows pc. Then I uploaded the confidential_document.txt to generate a hash function as shown in the Figure 1.1. The hash function is also called the checksum value.

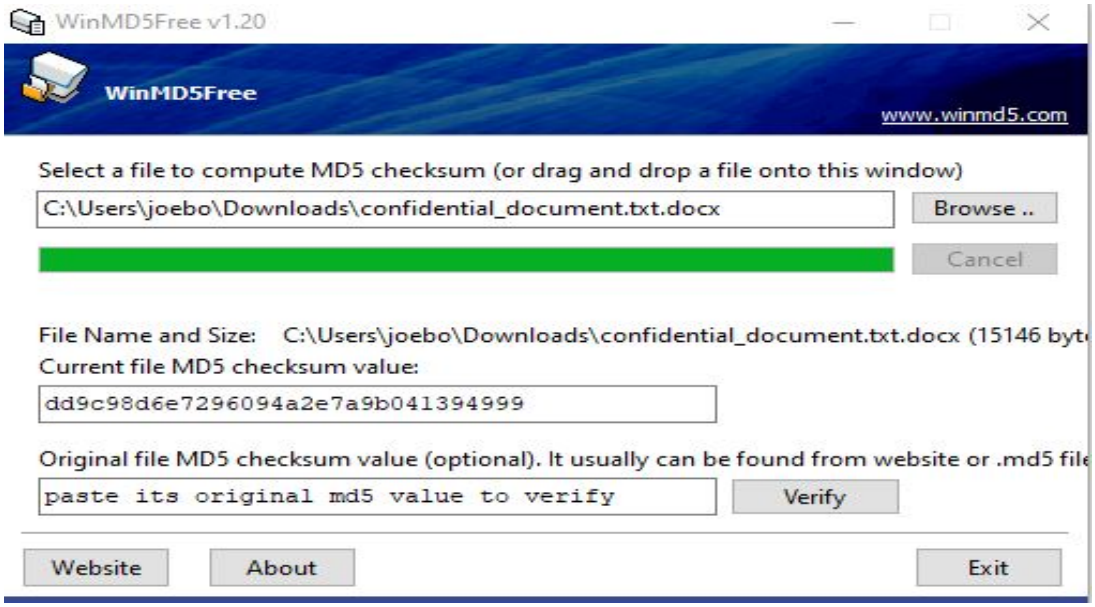

Figure 1.1: Hash function generated for confidential_document.txt

I altered the content of the confidential_document.txt and then, I uploaded it onto MD5 and generated a checksum value for it. This is shown below in Figure 1.2

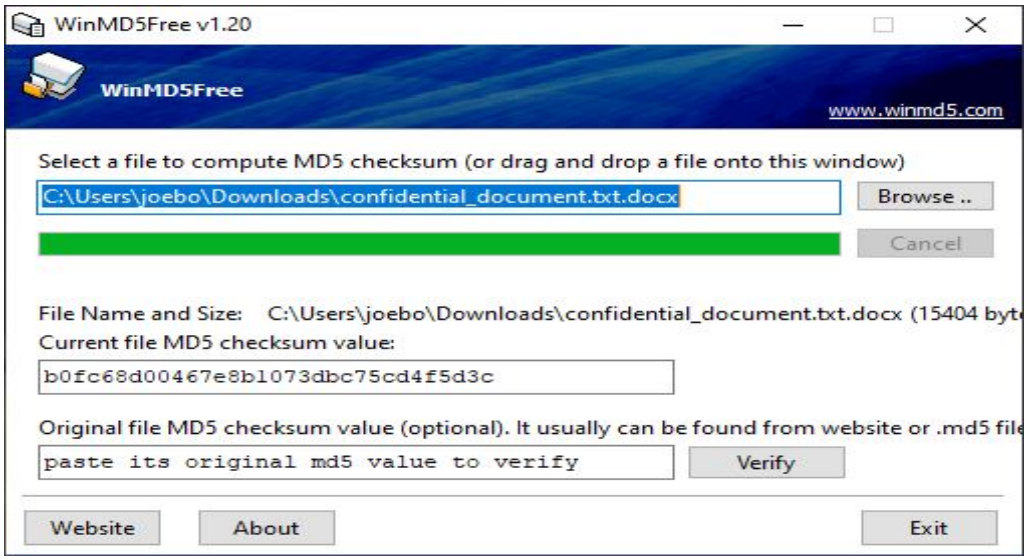

Figure 2.1: Hash function generated for the altered confidential_document.txt

I compared the hash function of the confidential_document.txt before and after it was altered. I observed as shown below in Figure 1.3, that MD5 show not matched and this means that, the confidential_document.txt is not the same file as after is was altered.

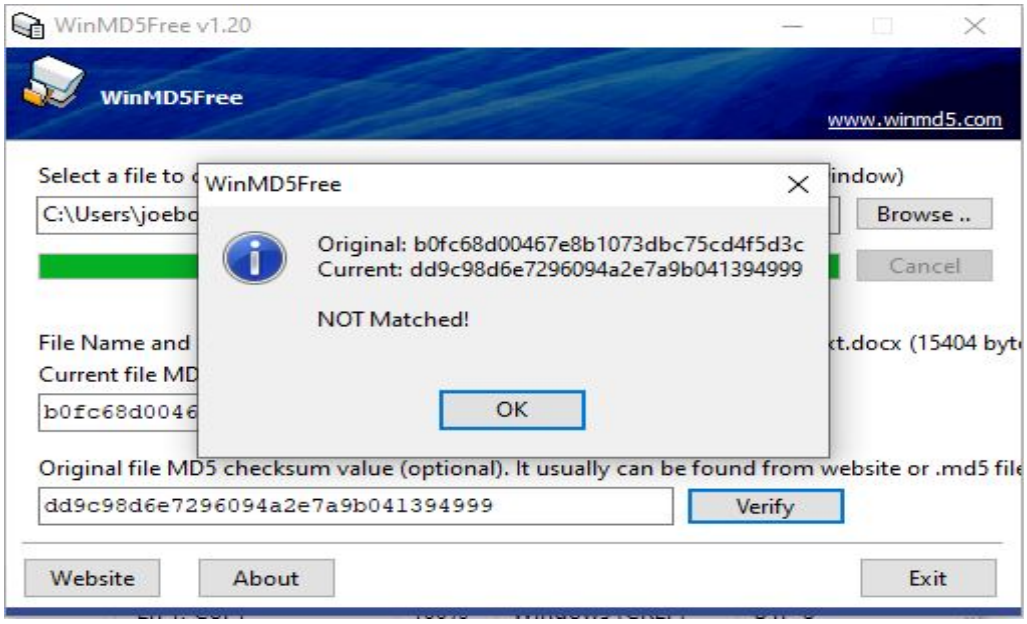

Figure 1.3: MD5 showing a mismatch between confidential_document.txt before and after it was altered.

Next, I created another document, and I named it Top Cyber Stories. Then I uploaded it onto MD5 and generated a hash function for it. This is shown in Figure 1.4 below.

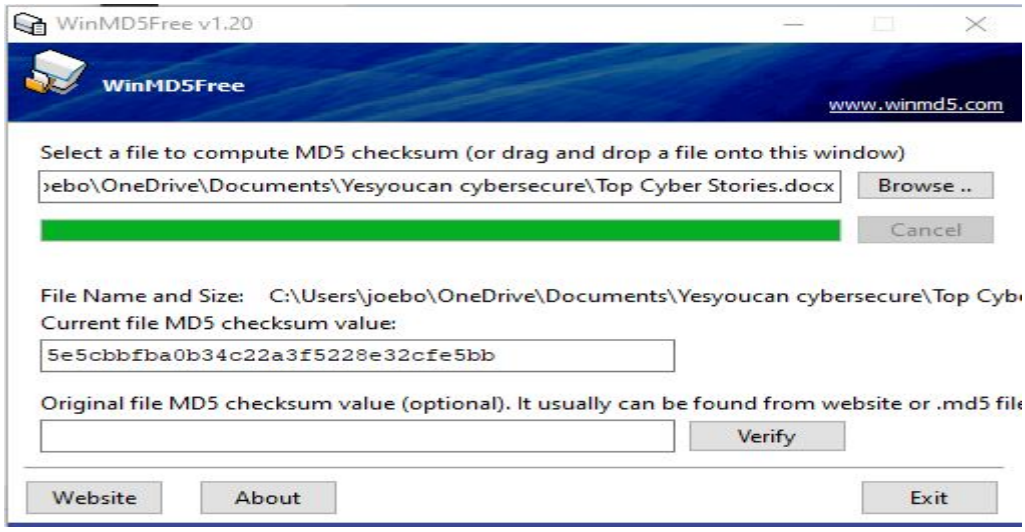

Figure 1.4: Hash function generated for the Top Cyber Stories document.

Then I compared the same hash functions and as shown in Figure 1.5 below, Matched. This means that nothing has been altered in the Top Cyber Stories file.

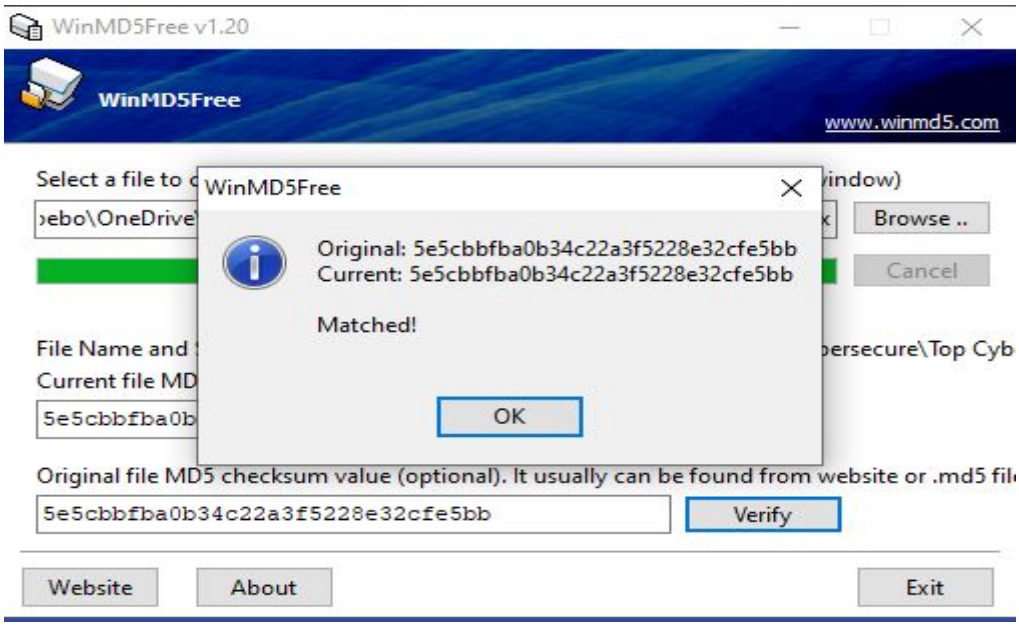

Figure 1.5: MD5 showing a match when Top Cyber Stores file was not altered and compared.

Q2. Symmetric encryption refers to using the same security key to encrypt and decrypt data. This key also know as the private key. Encryption is the process of converting readable data (plaintext) into an unreadable form (ciphertext) so that only someone with the right key can turn it back into its original form. There are many software that be used to perform encryption but this project will be limited to the “Hat.sh” Figure 2.1 shows the interface of the “Hat.sh” website.

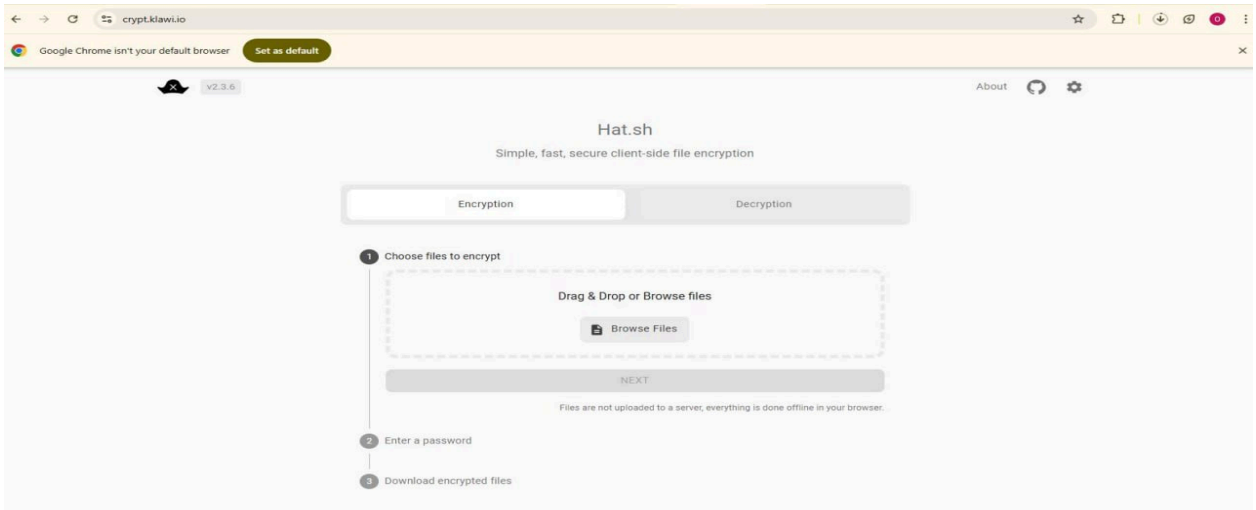

Figure 2.1: An interface of Hat.sh website.

After opening the website, you will upload your file and select a secure password(encryption key) to perform the symmetric encryption. You will get a prompt showing that the encryption was successful. This is shown in Figure 2.2 and Figure 2.3 below.

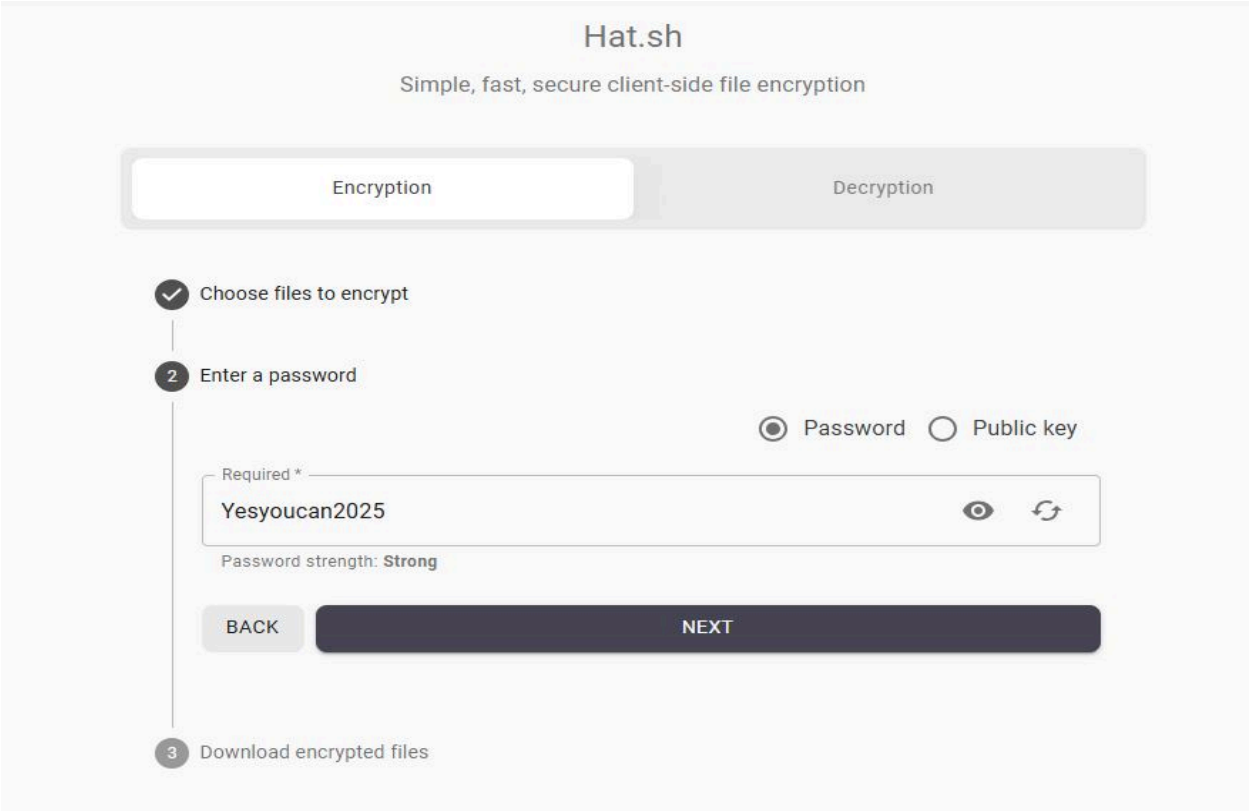

Figure 2.2: Document upload and password selected for encryption.

Figure 2.3: Top Cyber Stories document has been successfully encrypted and ready for download.

After encrypting the document, it has to be downloaded using the tab that says “encrypted files” as shown in Figure 2.3. Symmetric encryption is mostly used on data at rest, that is data stored on the hard drive.

To prevent bad actors from accessing both an encrypted file and its decryption key when sending information by email, the most effective method is to separate the delivery of the file and the key. The encrypted file can be sent as an email attachment, while the encryption key should be transmitted through a different channel such as SMS, a secure messaging app like Signal or WhatsApp, or even in person. This ensures that intercepting one channel does not compromise both pieces of information.

An even stronger approach is to use asymmetric encryption. In this method, the file is first encrypted with symmetric encryption for speed, and then the symmetric key is encrypted using the recipient’s public key. Only the recipient’s private key that is kept a secret can decrypt it. This layered method ensures that even if the encrypted file and encrypted key are intercepted together, they remain unusable without the private key, greatly enhancing security.

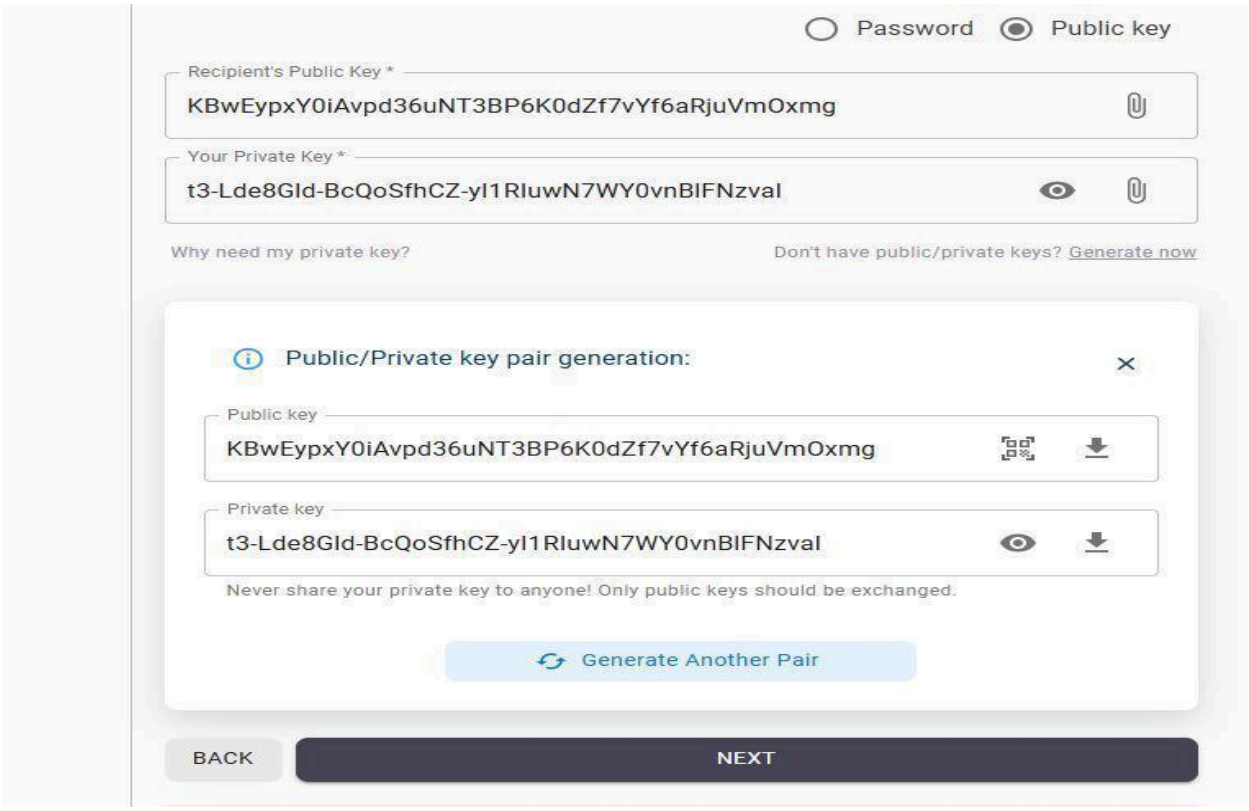

Figure 2.4: Asymmetric encryption

In Figure 2.4, for asymmetric encryption to work, parties involved in the exchange of data must generate two pairs of keys; a public key and private key. The sender encrypts the data with the recipient's public key and the recipient uses his or her private key to decrypt it.

Q3.When visiting Amazon.com or any other shopping site, employees should first confirm that the website address begins with https:// and displays a padlock icon in the browser's address bar. The "https" indicates that the connection is encrypted, protecting sensitive details like passwords and payment information from interception. This simple visual check ensures that the data sent between your browser and the website is secure.

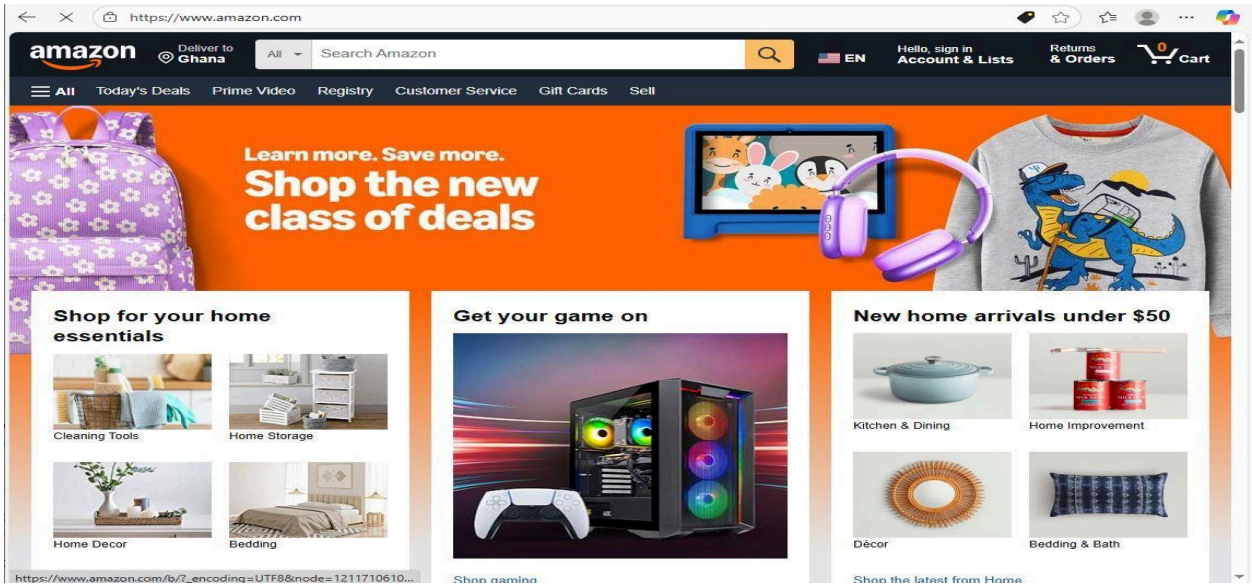

Figure 2.5: Https and padlock displayed in the address bar showing website is secure

Next, click on the padlock icon to view the site's

SSL/TLS certificate and confirm that it is valid, has not expired, and is issued to Amazon.com, Inc. by a trusted Certificate Authority. A valid certificate helps verify that the website is genuine and not a fake designed to steal information. By checking both the secure connection and the certificate validity before entering financial details, employees can greatly reduce the risk of online fraud. Figure 2.6 below shows that amazon.com has a valid certificate.

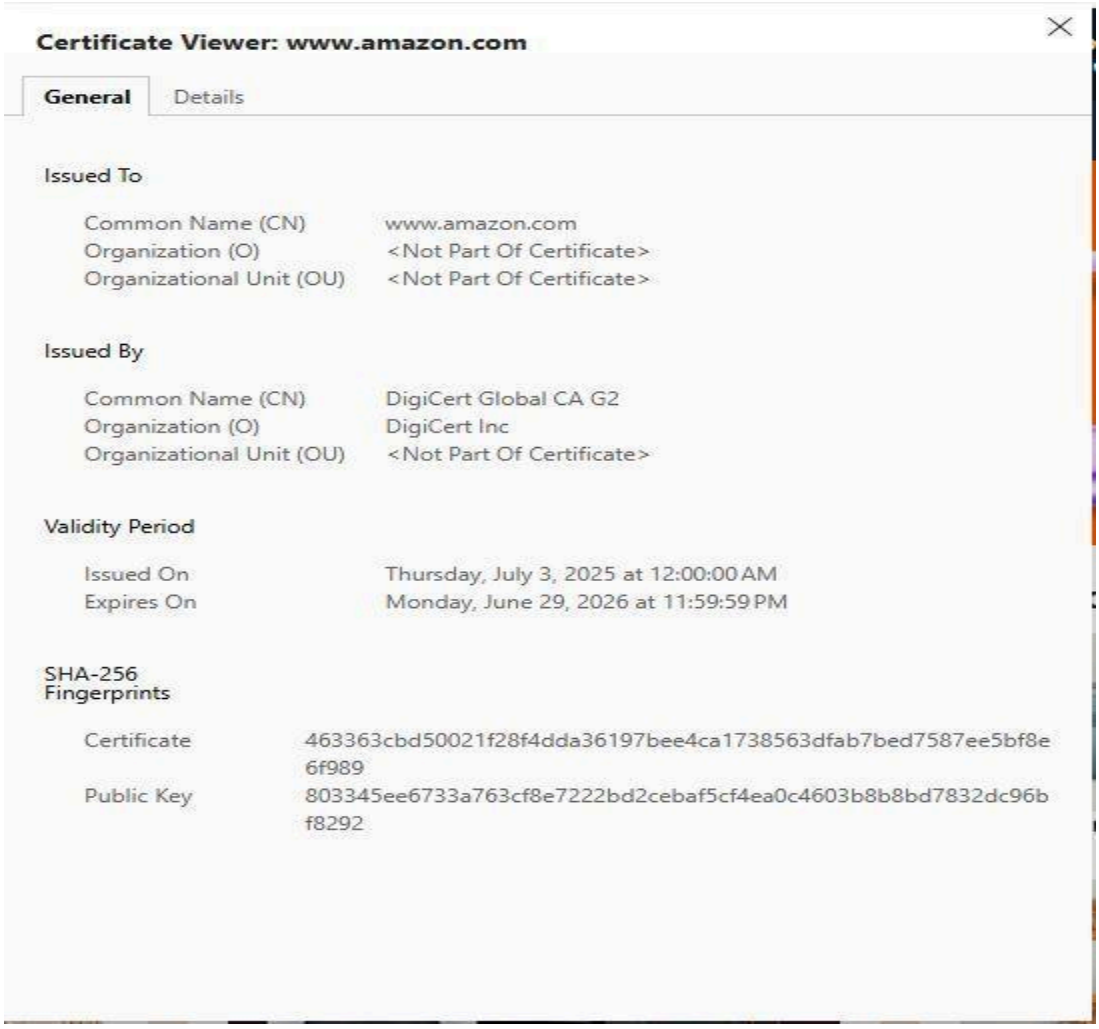

Figure 2.6: A valid SSL/TLS certificate.

How a Web Browser Interacts with a Website (Request, Retrieve, and Display Web Content)

When an employee visits https://www.amazon.com, the browser first sends a request to Amazon's servers. Amazon responds with its SSL/TLS certificate, which contains its public key and identity details. The browser verifies that the certificate is valid by checking the domain name, the issuing Certificate Authority, and the expiration date. Once verified, the browser and Amazon's server perform an SSL handshake to establish a secure connection. The browser creates a random session key, encrypts it with Amazon's public key, and sends it to the server. The server decrypts the session key using its private key, ensuring that all future communication is secure. This is shown in Figure 2.5 and Figure 2.6 above. The browser address bar showing https://www.amazon.com with the padlock icon and the Amazon's certificate details showing domain, issuer, and validity dates. Finally, Network tab showing the initial HTTPS request and response headers.

After the secure connection is established, the browser starts retrieving and displaying web content. It sends separate requests (known as “hits”) for different parts of the site such as HTML for page structure, CSS for styling, JavaScript for interactivity, and images for products. Each of these requests and responses is encrypted using the session key created during the handshake. The browser processes these files in real time and displays the fully rendered Amazon homepage for the user.