**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **17/6/2020** | **Name:** | **Chesmi B R** |
| **Course:** | **Cyber Security** | **USN:** | **4AL16EC100** |
| **Topic:** | **Governnace and Risk**  **Introduction to cryptography** | **Semester & Section:** | **8th sem A sec** |
| **Github Repository:** | **Chesmibr** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **REPORT:**  **Cryptography** is the study and practice of techniques for secure communication in the presence of third parties called adversaries. It deals with developing and analyzing protocols which prevents malicious third parties from retrieving information being shared between two entities thereby following the various aspects of information security.  Secure Communication refers to the scenario where the message or data shared between two parties can’t be accessed by an adversary. In Cryptography, an Adversary is a malicious entity, which aims to retrieve precious information or data thereby undermining the principles of information security.  Data Confidentiality, Data Integrity, Authentication and Non-repudiation are core principles of modern-day cryptography.   1. **Confidentiality** refers to certain rules and guidelines usually executed under confidentiality agreements which ensure that the information is restricted to certain people or places. 2. **Data integrity** refers to maintaining and making sure that the data stays accurate and consistent over its entire life cycle. 3. **Authentication** is the process of making sure that the piece of data being claimed by the user belongs to it. 4. **Non-repudiation** refers to ability to make sure that a person or a party associated with a contract or a communication cannot deny the authenticity of their signature over their document or the sending of a message.  A Brief Word on Polymorphism Before we continue, I want to touch on a more advanced topic known as [polymorphism](https://www.wilderssecurity.com/threads/polymorphic-cipher.321583/).  While the intricacies of this topic stretch far beyond the realm of this guide, its increasing prevalence mandates that I include a brief explanation.  Polymorphism is basically a cipher that changes itself with each use. Meaning that each time it is used, it produces a different set of results. So, if you encrypted the exact same set of data twice, each new encryption would be different from the previous one.  Let’s go back to our original example with the plaintext “Hello.” While the first encryption would result in “Khoor”, with the application of a polymorphic cipher, the second encryption could result in something like “Gdkkn” (where each letter is shifted down a rung of the alphabet)  Polymorphism is most commonly used in cipher algorithms to encrypt computers, software, and cloud-based information.  **Types of Cryptography** There are four primary types of cryptography in use today, each with its own unique advantages and disadvantages.  They are called hashing, symmetric cryptography, asymmetric cryptography, and key exchange algorithms. 1. Hashing Hashing is a type of cryptography that changes a message into an unreadable string of text for the purpose of verifying the message’s contents, not hiding the message itself.  This type of cryptography is most commonly used to protect the transmission of software and large files where the publisher of the files or software offers them for download. The reason for this is that, while it is easy to calculate the hash, it is extremely difficult to find an initial input that will provide an exact match for the desired value.  For example, when you download Windows 10, you download the software which then runs the downloaded file through the same hashing algorithm. It then compares the resulting hash with the one provided by the publisher. If they both match, then the download is completed.  However, if there is even the slightest variation in the downloaded file (either through the corruption of the file or intentional intervention from a third party) it will drastically change the resulting hash, potentially nullifying the download.  Currently, the most common hashing algorithms are [MD5 and SHA-1](https://crypto.stackexchange.com/questions/18612/how-is-sha1-different-from-md5), however due to these algorithm’s multiple weaknesses, most new applications are transitioning to the [SHA-256](http://www.xorbin.com/tools/sha256-hash-calculator) algorithm instead of its weaker predecessors. 2. Symmetric Cryptography Symmetric Cryptography, likely the most traditional form of cryptography, is also the system with which you are probably most familiar.  This type of cryptography uses a single key to encrypt a message and then decrypt that message upon delivery.  Since symmetric cryptography requires that you have a secure channel for delivering the crypto key to the recipient, this type of cryptography is all but useless for transmitting data (after all, if you have a secure way to deliver the key, why not deliver the message in the same manner?).  As such, its primary application is the protection of resting data (e.g. Hard Drives and data bases)  In the Revolutionary War example that I mentioned earlier, Washington’s method for transmitting information between his officers would have relied on a symmetric cryptography system. He and all of his officers would have had to meet in a secure location, share the agreed upon key, and then encrypt and decrypt correspondence using that same key.  Most modern symmetric cryptography relies on a system known as AES or [Advanced Encryption Standards](https://thebestvpn.com/advanced-encryption-standard-aes/).  While the traditional DES models were the industry norm for many years, [DES was publicly attacked](https://www.sans.org/reading-room/whitepapers/vpns/day-des-died-722) and broken in 1999 causing the National Institute of Standards and Technology to host a selection process for a stronger and more updated model.  After an arduous 5-year competition between 15 different ciphers, including MARS from IBM, RC6 from RSA Security, Serpent, Twofish, and Rijndael, the NIST selected [Rijndael as the winning cipher](http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf" \t "_blank).  It was then standardized across the country, earning the name AES or Advanced Encryption Standards. This cipher is still widely used today and is even implemented by the NSA for the purposes of guarding top secret information. 3. Asymmetric Cryptography Asymmetric cryptography (as the name suggests) uses two different keys for encryption and decryption, as opposed to the single key used in symmetric cryptography.  The first key is a public key used to encrypt a message, and the second is a private key which is used to decrypt them. The great part about this system is that only the private key can be used to decrypt encrypted messages sent from a public key.  While this type of cryptography is a bit more complicated, you are likely familiar with a number of its practical applications.  It is used when transmitting email files, remotely connecting to servers, and even digitally signing PDF files. Oh, and if you look in your browser and you notice a URL beginning with “https://”, that’s a prime example of asymmetric cryptography keeping your information safe. 4. Key Exchange Algorithms Although this particular type of cryptography isn’t particularly applicable for individuals outside of the cyber-security realm, I wanted to briefly mention to ensure you have a full understanding of the different cryptographic algorithms.  A key exchange algorithm, like Diffie-Hellman, is used to safely exchange encryption keys with an unknown party.  Unlike other forms of encryption, you are not sharing information during the key exchange. The end goal is to create an encryption key with another party that can later be used with the aforementioned forms of cryptography.  Here’s an example from the [Diffie-Hellman wiki](https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange) to explain exactly how this works.  Let’s say we have two people, Alice and Bob, who agree upon a random starting color. The color is public information and doesn’t need to be kept secret (but it does need to be different each time). Then Alice and Bob each selects a secret color that they do not share with anyone.  Now, Alice and Bob mix the secret color with the starting color, resulting in their new mixtures. They then publicly exchange their mixed colors. Once the exchange is made, they now add their own private color into the mixture they received from their partner, and the resulting in an identical shared mixture **The 4 Types of Cryptographic Functions** So now that you understand a little bit more about the different types of cryptography, many of you are probably wondering how it is applied in the modern world.  There are four primary ways that cryptography is implemented in information security. These four applications are called “cryptographic functions”. 1. Authentication When we use the right cryptographic system, we can establish the identity of a remote user or system quite easily. The go-to example of this is the [SSL certificate](https://www.globalsign.com/en/ssl-information-center/what-is-an-ssl-certificate/) of a web server which provides proof to the user that they are connected to the right server.  The identity in question is not the user, but rather the cryptographic key of that user. Meaning that the more secure the key, the more certain the identity of the user and vice versa.  Here’s an example.  Let’s say that I send you a message that I have encrypted with my private key and you then decrypt that message using my public key. Assuming that the keys are secure, it is safe to assume that I am the actual sender of the message in question.  If the message contains highly sensitive data, then I can ensure a heightened level of security by encrypting the message with my private key and then with your public key, meaning that you are the only person who can actually read the message and you will be certain the message came from me.  The only stipulation here is that the public keys are both associated with their users in a trusted manner, e.g. a trusted directory.  In order to address this weakness, the community created an object called a certificate which contains the issuer’s name as well as the name of the subject for whom the certificate is issued. This means that the fastest way to determine if a public key is secure is to note if the certificate issuer also has a certificate too.  An example of this type of cryptography in action is [Pretty Good Privacy,](https://en.wikipedia.org/wiki/Pretty_Good_Privacy) or PGP, a software package developed by Phil Zimmerman that provides encryption and authentication for email and file storage applications.  This software package provides users with message encryption, digital signatures, data compression, and email compatibility.  Although Zimmerman ran into some legal problems with the initial software which used an RSA for key transport, MIT PGP versions 2.6 and later are legal freeware for personal use, and Viacrypt 2.7 and later versions are legal commercial alternatives. 2. Nonrepudiation This concept is especially important for anyone using or developing financial or e-commerce applications.  One of the big problems that e-commerce pioneers faced was the pervasive nature of users who would refute transactions once they had already occurred. Cryptographic tools were created to ensure that each unique user had indeed made a transaction request that would be irrefutable at a later time.  For example, let’s say that a customer at your local bank requests a money transfer to be paid to another account. Later in the week, they claim to have never made the request and demand the full amount be refunded to their account.  However, as long as that bank has taken measures to ensure non-repudiation through cryptography, they can prove that the transaction in question was, in fact, authorized by the user. 3. Confidentiality With information leaks and a seemingly endless number of privacy scandals making the headlines, keeping your private information,, well, private is probably one of your biggest concerns. This is the exact function for which cryptographic systems were originally developed.  With the right encryption tools, users can guard sensitive company data, personal medical records, or just lock their computer with a simple password. 4. Integrity Another important use of cryptography is to ensure that data is not viewed or altered during transmission or storage.  For example, using a cryptographic system to ensure data integrity ensures that rivaling companies cannot tamper with their competitor’s internal correspondence and sensitive data.  The most common way to do accomplish data integrity through cryptography is by using cryptographic hashes to safeguard information with a secure checksum. |