**TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES**

**Quezon City**

**IT 001 - Information Assurance and Security 1**

| **Member 1: Baita, Arlen Sean G.**  **Member 2: Magabilin, Gian Matthew F.**  **Member 3: Mesinas, Christian Jeremy D.**  **Member 4: Sabino, Jeronimo M.**  **Member 5: San Juan, Emmanuel Jose T.** | **Date: February 6, 2025** |
| --- | --- |
| **Assignment 1. Practice Set 1. Practice Set 1: Basic Cryptography Concept – Hash Function** | **Lecturer: Ms. Nila D. Santiago** |

**Objectives:**

1. Determine the basic concepts of cryptography.
2. Distinguish hashing as a cryptographic approach and its applications.
3. Discover and utilize online hashing tools.

**Instructions:**

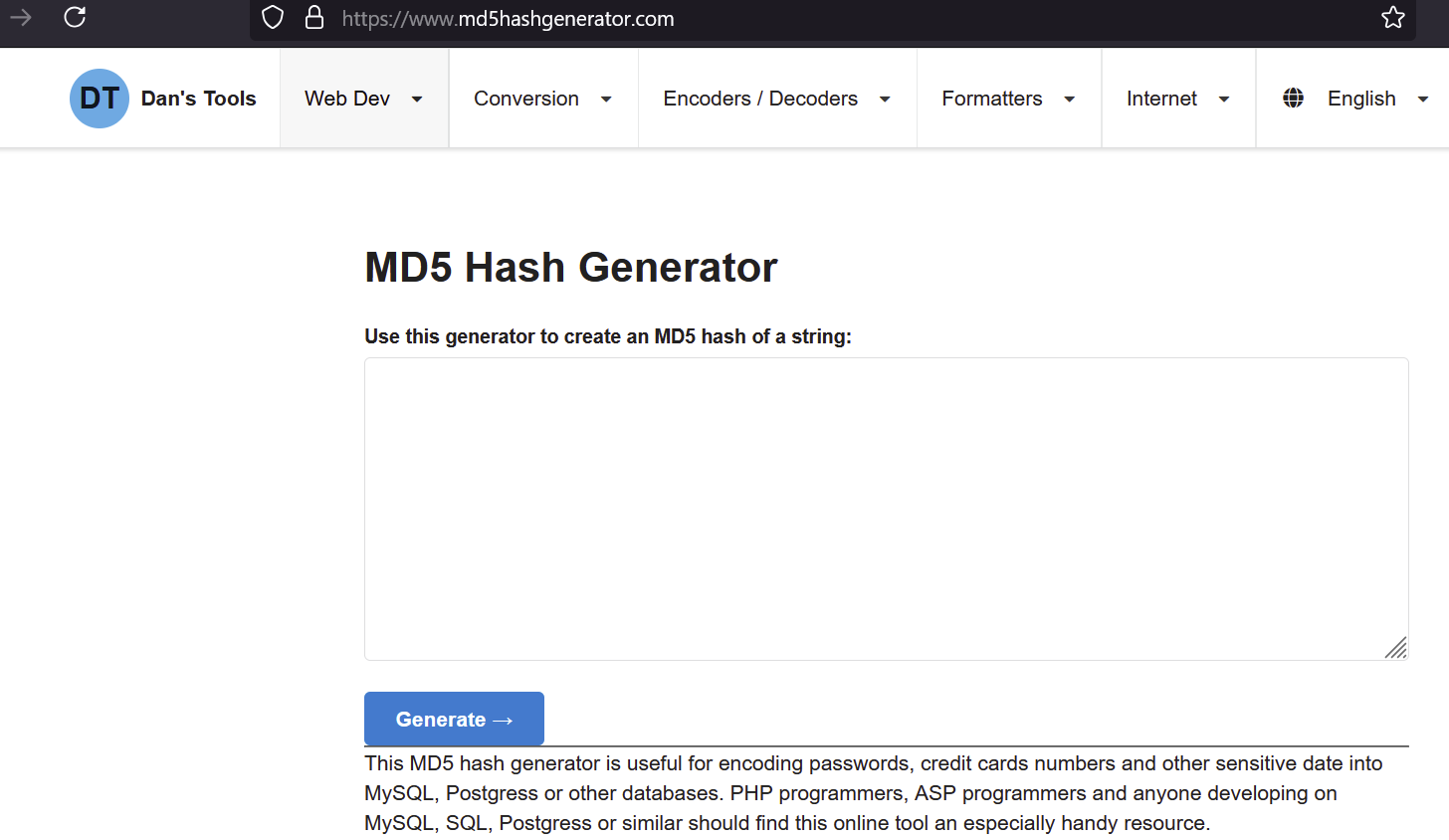
1. Form a group with a maximum of 5 members.
2. **Group Research**

* What is cryptography?
* How does hashing differ from encryption?
* Examples of cryptographic applications (e.g., passwords, digital signatures).
* Make a research about the role of hashing in cryptography. Focus on its purpose (e.g., data integrity, digital signatures) and characteristics (e.g., one-way function, unique output).

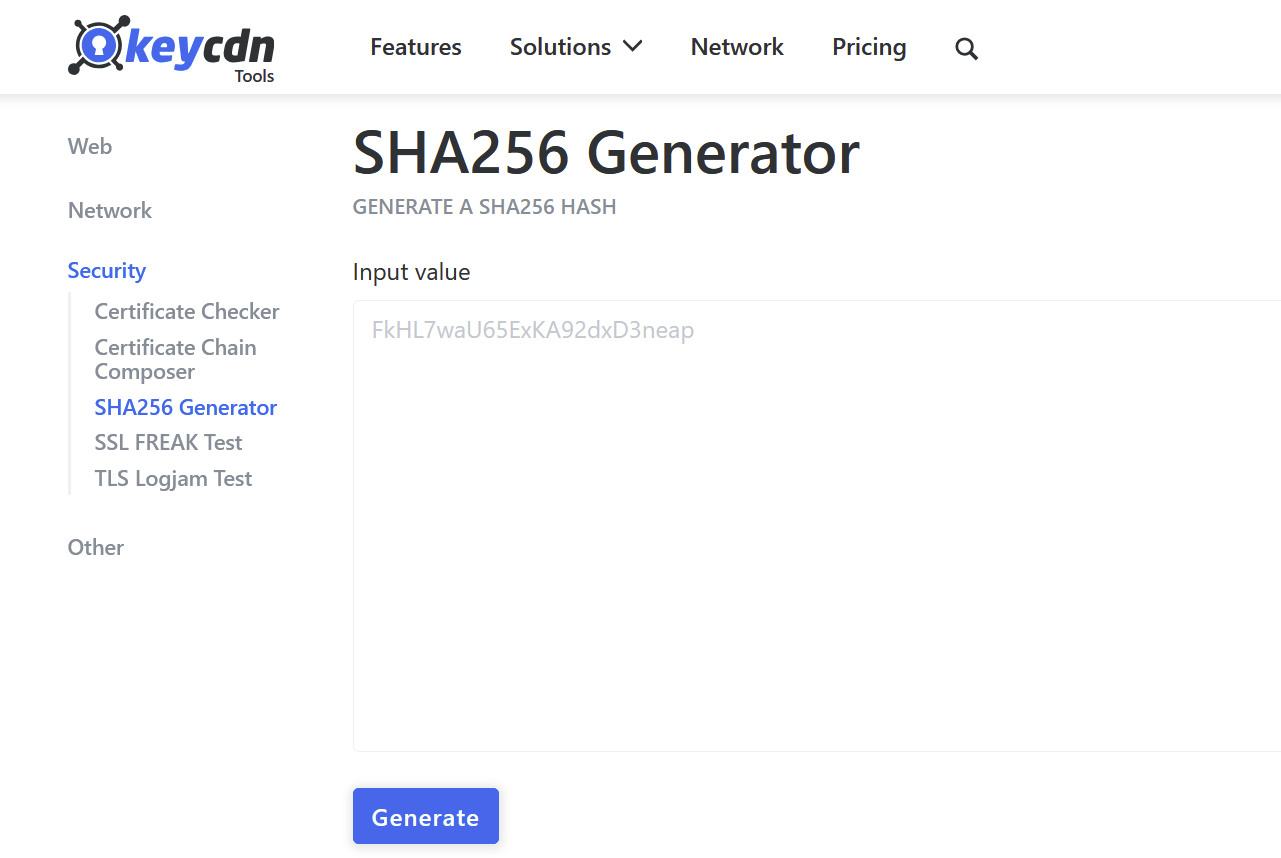
1. **Individual Hands-on Activity**

**Generate Hashes Using Online Tools**:

* Visit the following online tools:
  + [MD5 Hash Generator](https://www.md5hashgenerator.com/)

****

* + [SHA256 Online Tool](https://tools.keycdn.com/sha256-online-generator)



* Generate hashes for the following inputs:
  + Your full name.
  + A 10-character random password.
  + The phrase: *Information Assurance and Security*.
* Compare the outputs and observe the differences when altering input data.
* What happens when even one character of the input changes?
* What are the advantages and limitations of these hashing algorithms?
* Search for and identify at least two other online tools that support hashing (e.g., SHA1, SHA512).
* Test these tools using the same inputs as above.
* Document the features of each tool (e.g., speed, user interface, supported algorithms).
* Write a brief paragraph comparing the tools used.
* Which tool do you find the most user-friendly and why?

1. **Group Synthesis**

* Considering that hashing is one-way and the hash is never reversed, what makes hashing a useful security technique?
* Can you describe some real-world situations where you used basic security techniques such as authentication, access control, and encryption?

### **Submission Requirements**

1. Group Research
2. Individual Output (Hands-On Activity and Reflection)
3. Group Synthesis

**Answer:**

**PART 1: Group Research**

What is cryptography?

According to IBM (n.d.), cryptography is the science of implementing computer algorithms to conceal information as a ciphertext to limit its access only to authorized subjects. Therefore, intercepted information becomes worthless for unauthorized parties. Cryptography is traced back to ancient times when Julius Caesar developed the Caesar Cipher to protect the messages he sent. At present, cryptography is governed by technological agencies such as the National Institute of Standards and Technology (NIST) to ensure data security. Cryptography is also associated with advanced mathematics in the development of its algorithms.

How does hashing differ from encryption?

According to GeeksforGeeks (2024), hashing is the practice of subjecting information to an irreversible hash function for database retrieval purposes. Unlike encryption, hashing is an irreversible process. The hash key is only used to compare whether the original information matches with input data, including passwords. Its sole purpose is to index and retrieve database items. Examples include Bcrypt and SHA256.

Meanwhile, encryption is the conversion of a plaintext to an obscure ciphertext through a standard algorithm. However, unlike hashing, encrypted ciphertext can be converted back to plaintext using an encryption key. Examples of algorithms include Blowfish and RSA.

Examples of cryptographic applications (e.g., passwords, digital signatures).

Cryptography is a fundamental tool for securing digital information, ensuring data integrity, authenticity, and confidentiality. It plays a crucial role in modern cybersecurity by protecting sensitive data through encryption, hashing, and digital signatures. The applications of cryptography are widespread, ranging from password protection to secure communication protocols.

* **Passwords**

One of the most common applications of cryptography is password security. Instead of storing actual passwords, websites store cryptographic hashes of passwords. A hash function converts a password into a fixed-length, unique string of characters. When a user logs in, their entered password is hashed and compared to the stored hash. If they match, access is granted.

Importance: Hashing ensures that even if an attacker gains access to a website’s database, they cannot retrieve the actual password. Strong cryptographic hashing algorithms, such as bcrypt, PBKDF2, or Argon2, add an additional layer of security.

* **Digital Signatures**

Digital signatures use asymmetric cryptography to verify the authenticity and integrity of digital documents. This involves two keys: a private key (kept secret) and a public key (shared openly). The sender signs a document using their private key, creating a unique digital signature. The recipient verifies this signature using the sender’s public key, ensuring that the document has not been altered and is genuinely from the sender.

Importance: Digital signatures provide non-repudiation, meaning the sender cannot deny having signed the document. They also guarantee that the document has not been tampered with during transmission.

* **Secure Communication (SSL/TLS)**

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols use cryptography to encrypt communication between users and websites. When a website uses "https," it means SSL/TLS is securing the data exchange between the browser and the server. This encryption prevents eavesdroppers from intercepting sensitive information like login credentials and financial details.

Importance: SSL/TLS protects sensitive online transactions, ensuring data confidentiality, integrity, and authenticity. It prevents man-in-the-middle attacks and safeguards personal data from unauthorized access.

* **Data Encryption**

Data encryption transforms readable data (plaintext) into an unreadable format (ciphertext) using encryption algorithms. Only someone with the correct decryption key can convert the ciphertext back into plaintext. Encryption is used for securing stored data on devices, cloud storage, and transmitted data over networks.

Importance: Encryption safeguards sensitive data from unauthorized access, even if a device is lost or stolen. Common encryption algorithms include AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman).

* **Cryptocurrency**

Cryptocurrencies, such as Bitcoin, rely on cryptography for security, authenticity verification, and maintaining a decentralized transaction ledger known as the blockchain. Public-key cryptography ensures that transactions are legitimate while hashing algorithms secure the blockchain by preventing unauthorized modifications.

Importance: Cryptography enables secure, tamper-proof digital transactions, making cryptocurrencies resistant to fraud and counterfeiting. It also allows for decentralized financial systems without the need for intermediaries.

By leveraging cryptographic techniques, digital systems ensure data confidentiality, integrity, and authenticity, making cybersecurity a cornerstone of modern digital infrastructure.

Make a research about the role of hashing in cryptography. Focus on its purpose (e.g., data integrity, digital signatures) and characteristics (e.g., one-way function, unique output).

Hashing is considered a cornerstone of modern cryptographic systems since it has a significant impact on cryptography by ensuring that every data will maintain its integrity, authenticity, as well as security. Hashing works by converting input data into unique and fixed-length hash values making it ideal for different applications such as password storage, digital signatures, and verification of data integrity. The role of hash in cryptography is essential since the slightest change within the input process could provide a different hash. Further on, hashing is widely used in tasks like file identification, data indexing, and blockchain for proof-of-work systems, which secures and validates every transaction performed.

Cryptographic hash functions have several key characteristics that ensure security and reliability:

* **Collision Resistance** – No two different inputs should produce the same hash output, preventing data tampering. If you have two pieces of text, "hello" and "hi," they should produce different hash values (e.g., SHA-256("hello") ≠ SHA-256("hi")). If two inputs produce the same hash, it's a collision, which would undermine data integrity.
* **Preimage Resistance** – Given a hash output, it should be impractical to determine the original input, keeping information secure. Given a hash value like 5d41402abc4b2a76b9719d911017c592, it is nearly impossible to figure out that the original input was "hello." Hashing keeps sensitive data protected since you can't reverse-engineer the original input.
* **Second Preimage Resistance** – It should be challenging to find another input that produces the same hash as a given input, preventing forgery. If a file with hash abc123 is already created, it's difficult to find another file that produces the same hash (another file that also results in abc123). Therefore, hashing helps prevent data forgery, ensuring that the original file hasn't been tampered with.
* **One-Way Function** – Hashing is irreversible, meaning the original data cannot be reconstructed from the hash. If you hash "password123" using SHA-256, you get a fixed hash output. But you can't reverse the hash back into "password123" from the hash value. Hashing is designed to be irreversible, protecting your data from being exposed.
* **Deterministic** – The same input always produces the same hash output, ensuring consistency. No matter how many times you hash the word "apple," you will always get the same hash value (e.g., SHA-256("apple") = 1f3870be274f6c49b3e31a0c6728957f). This consistency is essential for verifying data, as the same input will always produce the same result.
* **Fast and Efficient** – Hash functions generate fixed-length outputs quickly, making them practical for real-time applications. If you hash a small text like "hello world" or a large document, both will be processed quickly to generate a fixed-length hash. Hashing functions allow real-time verification without noticeable delay.
* **Puzzle-Friendliness** – Finding an input that produces a specific hash should be computationally tricky, making brute-force attacks impractical. If you are trying to find an input that produces a specific hash like d2d2d2d2..., it's tough to guess the correct input because there are so many possibilities. For instance, it could take millions of guesses before the right one is found, which makes attacks like brute force infeasible.

These characteristics make cryptographic hashing essential for password storage, digital signatures, and data integrity verification in cybersecurity.

**PART 2: Individual Hands-on Activity**

**BAITA**

Generate hashes for the following inputs:

Your full name.

| MD5:    SHA256: |
| --- |

A 10-character random password.

| MD5:    SHA256: |
| --- |

The phrase: Information Assurance and Security.

| MD5:    SHA256: |
| --- |

Compare the outputs and observe the differences when altering input data.

When you alter the input data, the output hash changes drastically even if just one character is modified. For example, MD5 produces a 32-character hash, while SHA256 generates a 64-character hash. Despite the difference in length, both algorithms will produce entirely different hash values with any small change in the input.

What happens when even one character of the input changes?

When even one character of the input changes, the result will change completely, which shows the sensitivity of hash functions. This is true for both MD5 and SHA256, where a small alteration leads to a very different output hash.

What are the advantages and limitations of these hashing algorithms?

MD5 is fast and widely used, but it is vulnerable to attacks, making it less secure for sensitive applications. SHA256 is more secure and resistant to attacks, making it ideal for cryptographic use, but it is slower than MD5, especially with larger data.

Search for and identify at least two other online tools that support hashing (e.g., SHA1, SHA512).

browserling and Tools4noobs are two online tools that support hashing algorithms like SHA1 and SHA512. browserling offers a simple, ad-free interface for generating multiple hashes by pasting text and clicking a button. Tools4noobs supports many algorithms, including MD5, SHA, and RIPEMD, providing flexibility for different hashing needs.

Test these tools using the same inputs as above.

| Browserling:        Browserling displays all the results of different cryptographic hashes in one click. It also has a button to copy all the generated hashes to the clipboard if you want to. The tool also has no ads which is just clean. |
| --- |

| Tools4noobs:        Tools for noobs has a text box to insert the text you want to hash with a dropdown box that contains all the supported hashing algorithms. After choosing an algorithm, clicking the hash this button generates the hashing result. |
| --- |

Write a brief paragraph comparing the tools used.

Tools4noobs and Browserling are both user-friendly hashing tools. Tools4noobs features a text box for input, a dropdown menu to select hashing algorithms, and a button to generate results. It supports a wide range of algorithms. Browserling, on the other hand, generates multiple cryptographic hashes in one click and allows users to copy all results at once. Both tools are clean and ad-free, offering efficient and straightforward hashing capabilities.

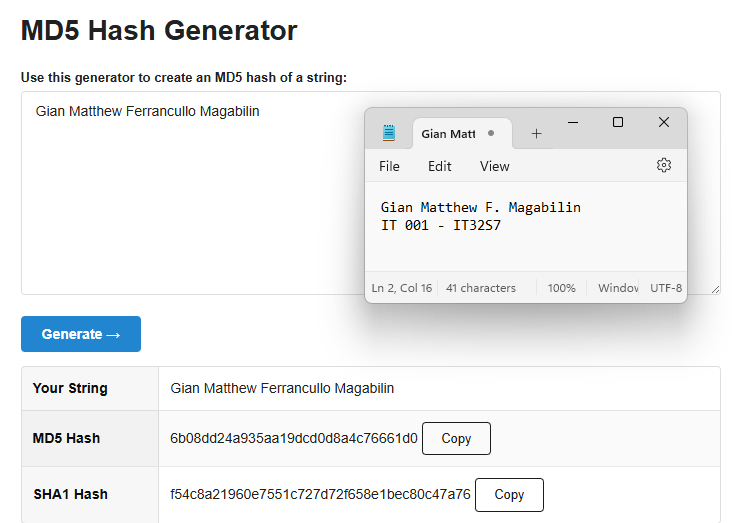
Which tool do you find the most user-friendly and why?

Personally, I prefer Browserling because it generates all the hash results quickly with just one click and supports a wide range of algorithms. The ability to copy all the generated hashes at once adds to its convenience, making it very user-friendly for tasks that require multiple hash types.

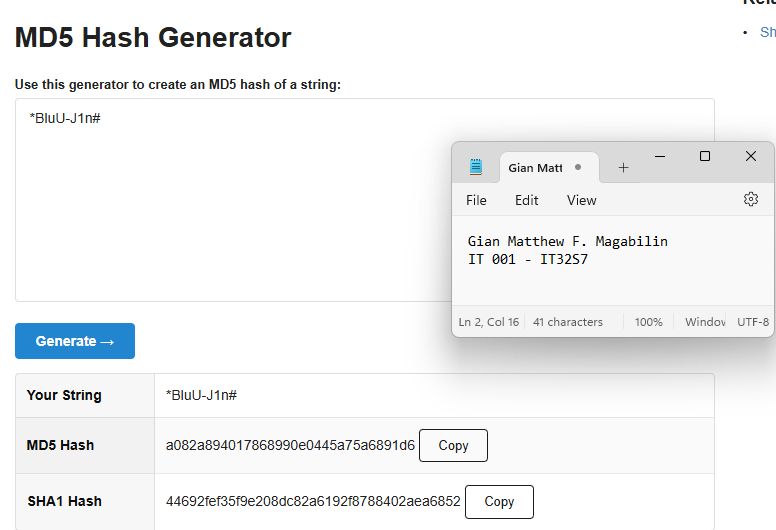
**MAGABILIN**

MD5 Hash Generator

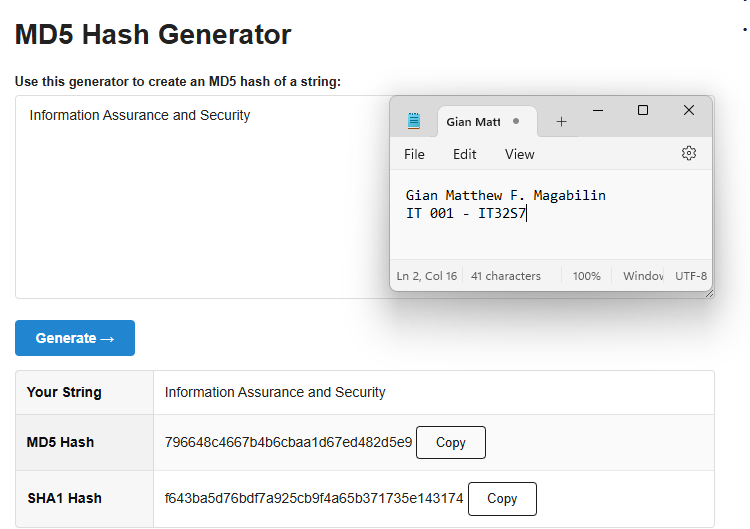
Full Name (Gian Matthew Ferrancullo Magabilin)



10 Character Random Password (\*BluU-J1n#)

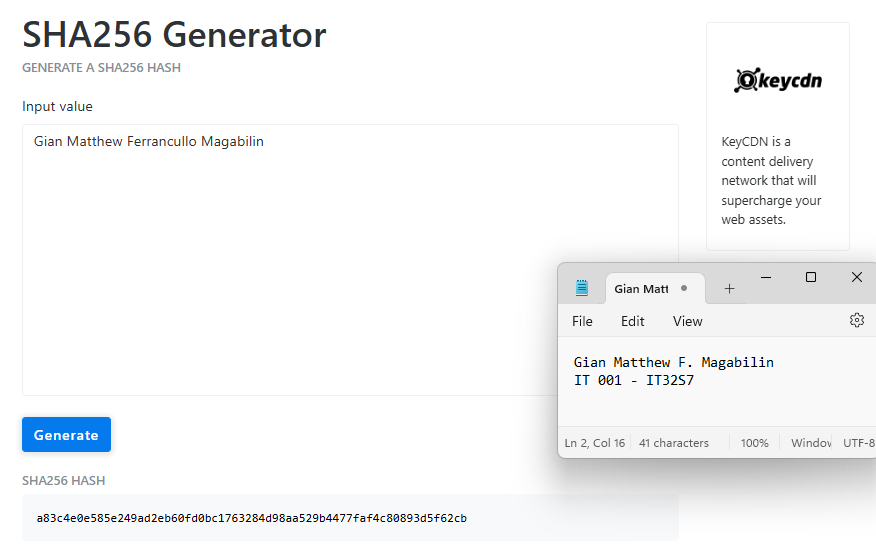


The phrase: *Information Assurance and Security*

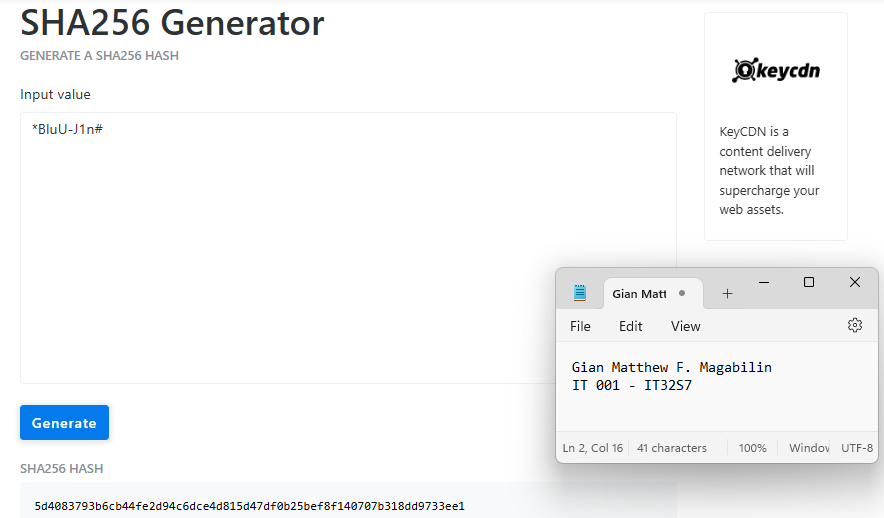


SHA256 Online Tool

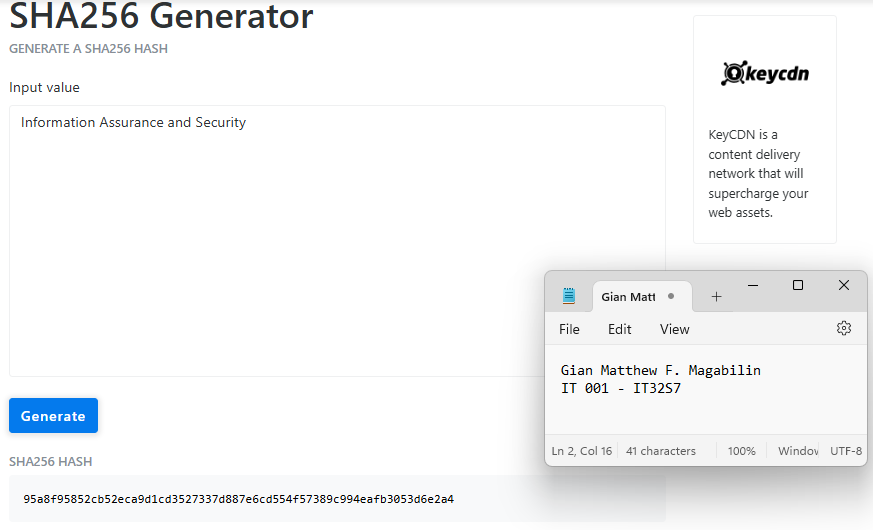
Full Name (Gian Matthew Ferrancullo Magabilin)



10 Character Random Password (\*BluU-J1n#)



The phrase: *Information Assurance and Security*



Compare the outputs and observe the difference when altering input data.

Based on the hash generators MD5 and SHA256 generated, there are several dissimilarities between the characteristics of the output when altering the input data. In this scenario, I have generated a random 10-character random password (\*BluU-J1n#) using MD5 and SHA256 to observe the variation of the outputs. MD5 provides a 32-character hexadecimal string where on the other hand, the SHA256 makes a 64-character hexadecimal string. This supports the idea that the SHA256 is more secure than MD5 since it produces longer hash which makes it stronger than what the MD5 produces. However, the downside of this is that SHA256 is slower than MD5 due to its longer hash. Lastly, the input is highly sensitive and could produce completely different outputs in order for it to ensure that every hash that will be generated is unpredictable.

What happens when even one character of the input changes?

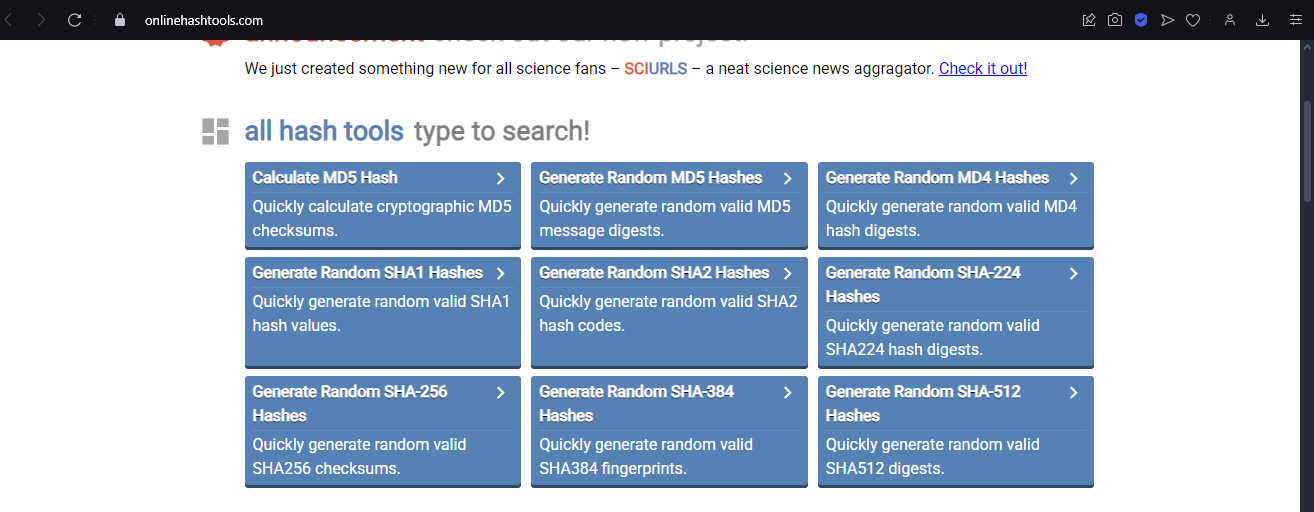
Based on the two hash generators, even a single character change in the input results in an entirely unique output. This ensures the unpredictability of the hash that was generated to maintain the quality of its security and reliability. In addition, this regulates the idea of having similarities and patterns.

What are the advantages and limitations of these hashing algorithms?

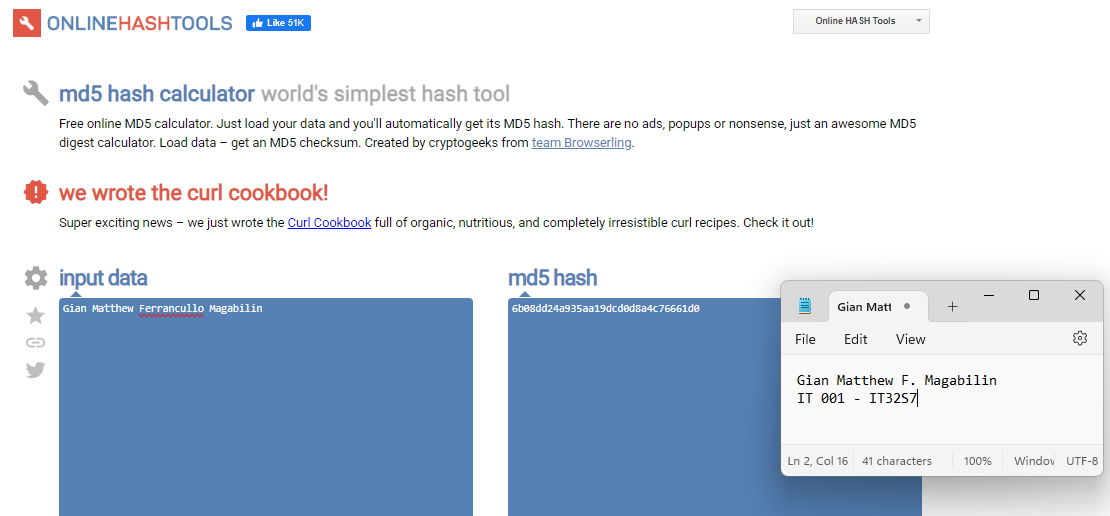
| **Hash Algorithm** | **Advantages** | **Limitations** |
| --- | --- | --- |
| **MD5** | * Excels in Speed and Efficiency. * Has a Fixed-Length Output due to its 128-bit hash value/32-character hexadecimal number. | * Vulnerable to attacks whenever two inputs provide the same hash output. * Does not do well in handling security for tasks like digital signatures. |
| **SHA256** | * High level security * Ensures data does not change during transmission. | * Slower performance. * There is a chance that it could produce the same hash values due to collisions. |

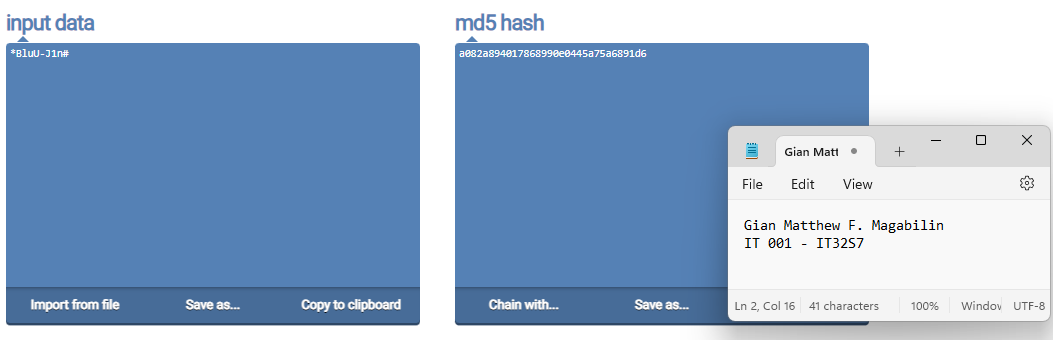
Search for and identify at least two other online tools that support hashing (e.g., SHA1, SHA512).

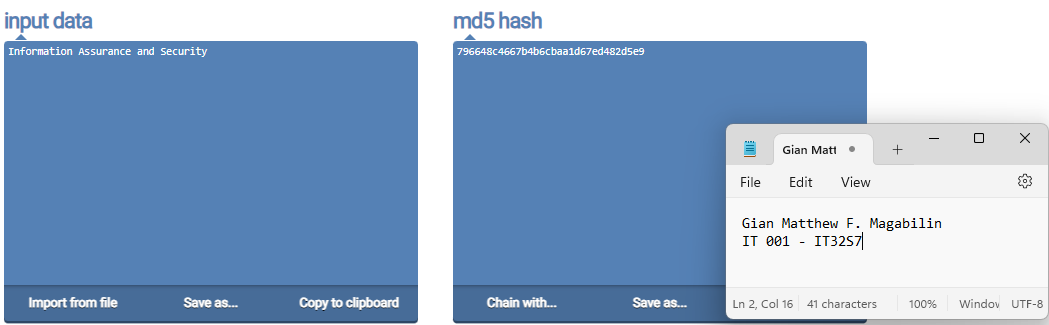
**Online Tool #1:** Online Hash Tools - is a free online tool that is useful for cryptography and hashing data. It has various hashing algorithms that are ad-free and have no popups. It enables the user to calculate their own MD5 hash as well as to generate random MD5, MD4, SHA1, SHA2, SHA-224, SHA-226, SHA-384, and SHA-512 Hashes.



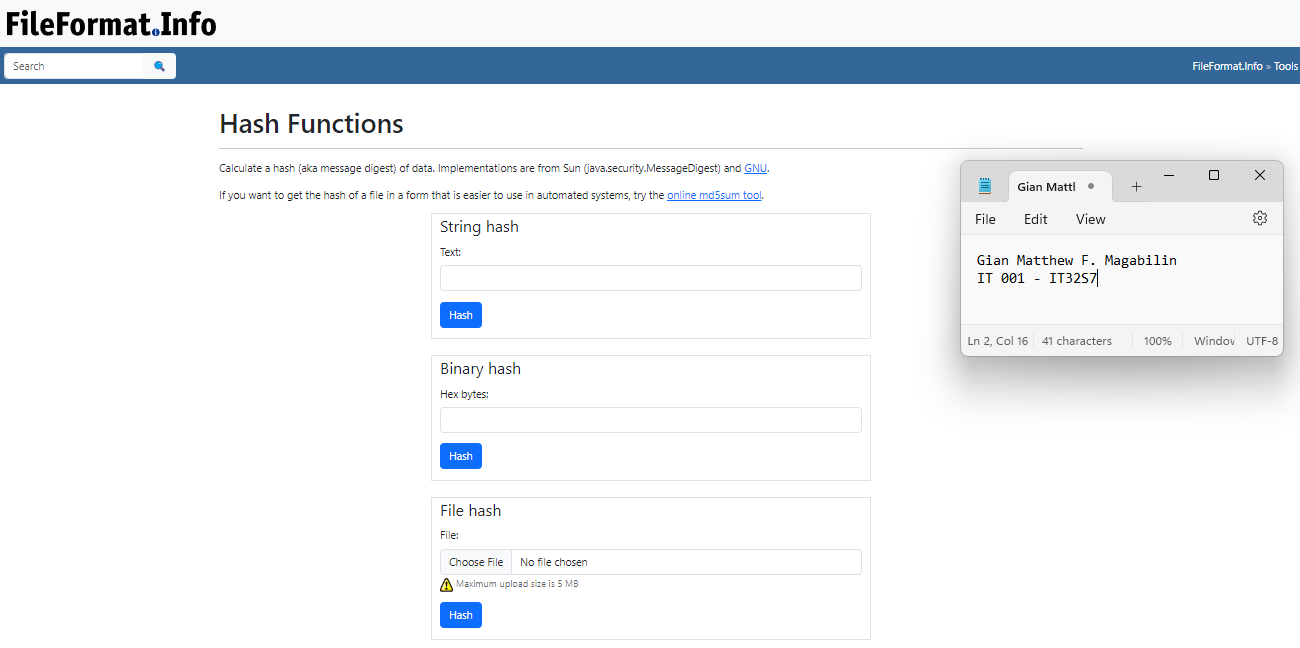
Test:



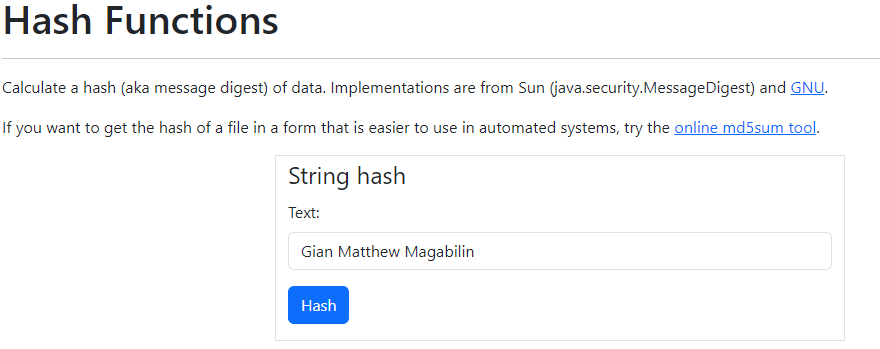


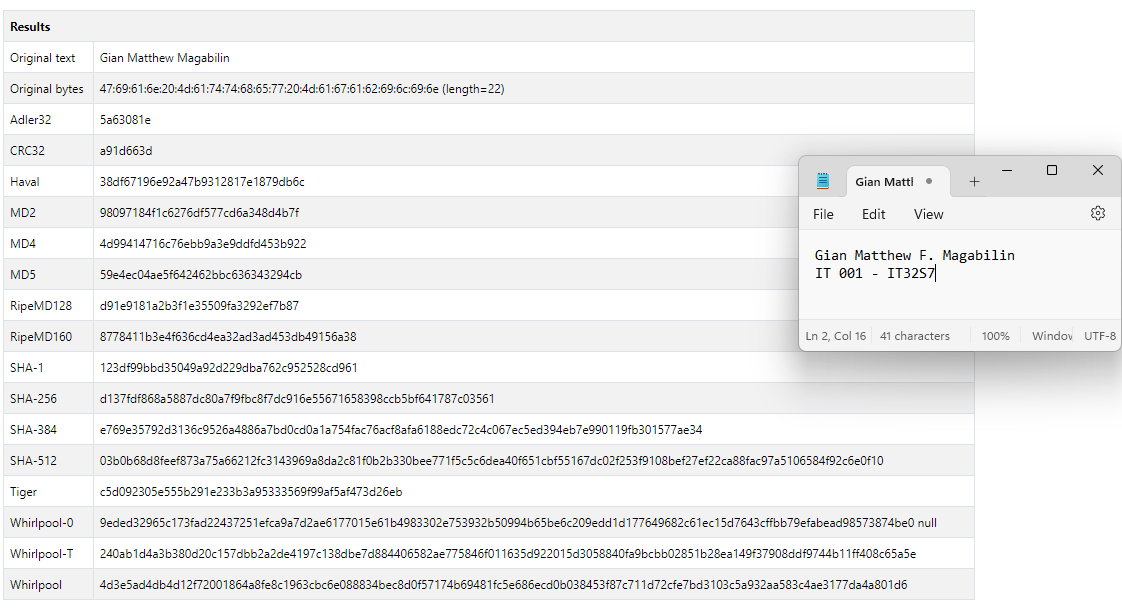


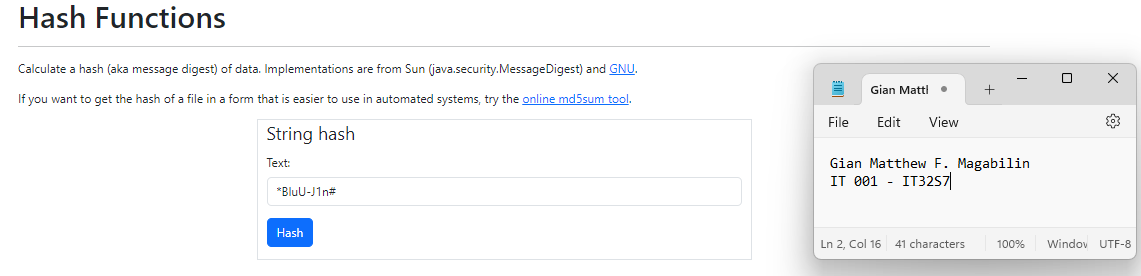
**Online Tool #2:** FileFormat.Info - is an online tool for Hash functions. It allows users to input String Hash, Binary Hash, as well as File Hash. This online tool supports various hash algorithms such as Adler32, CRC32, Haval, MD2, MD4, MD5, RipeMD160, SHA-1, SHA-256, SHA-384, SHA-512, Tiger, Whirlpool-0, Whirlpool-T, and Whirlpool.

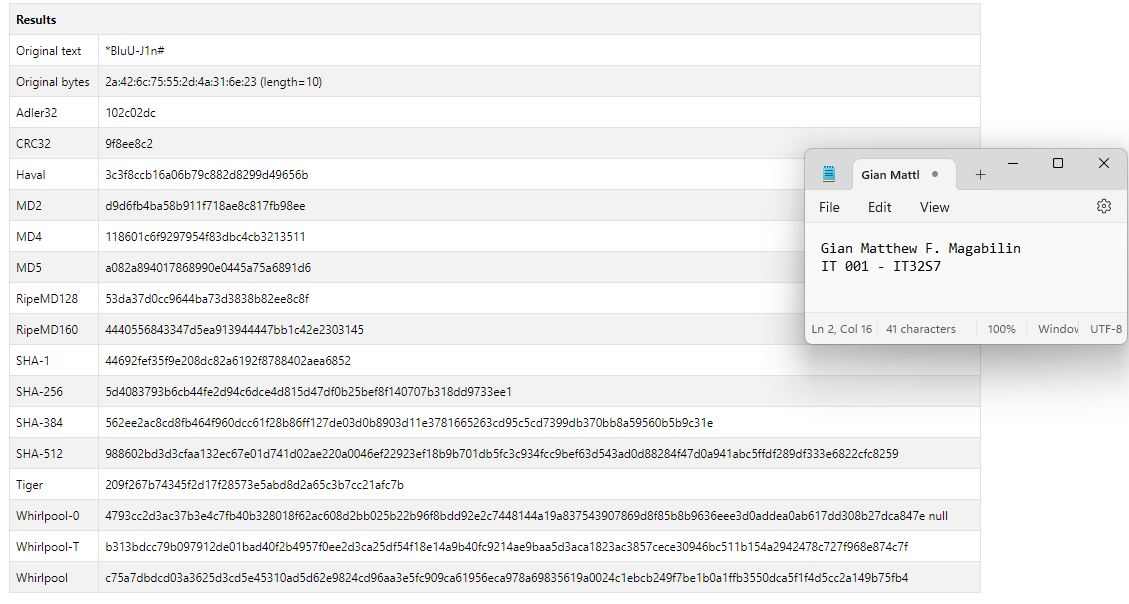


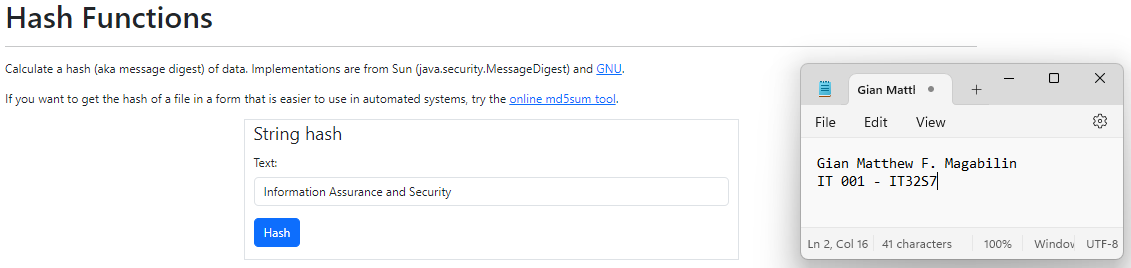
Test:

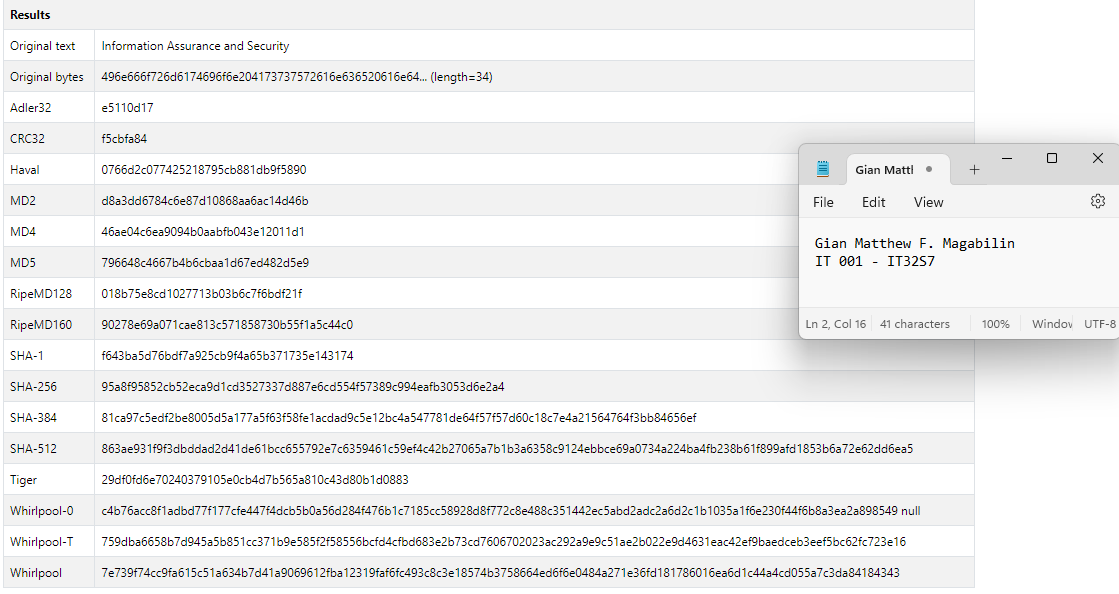












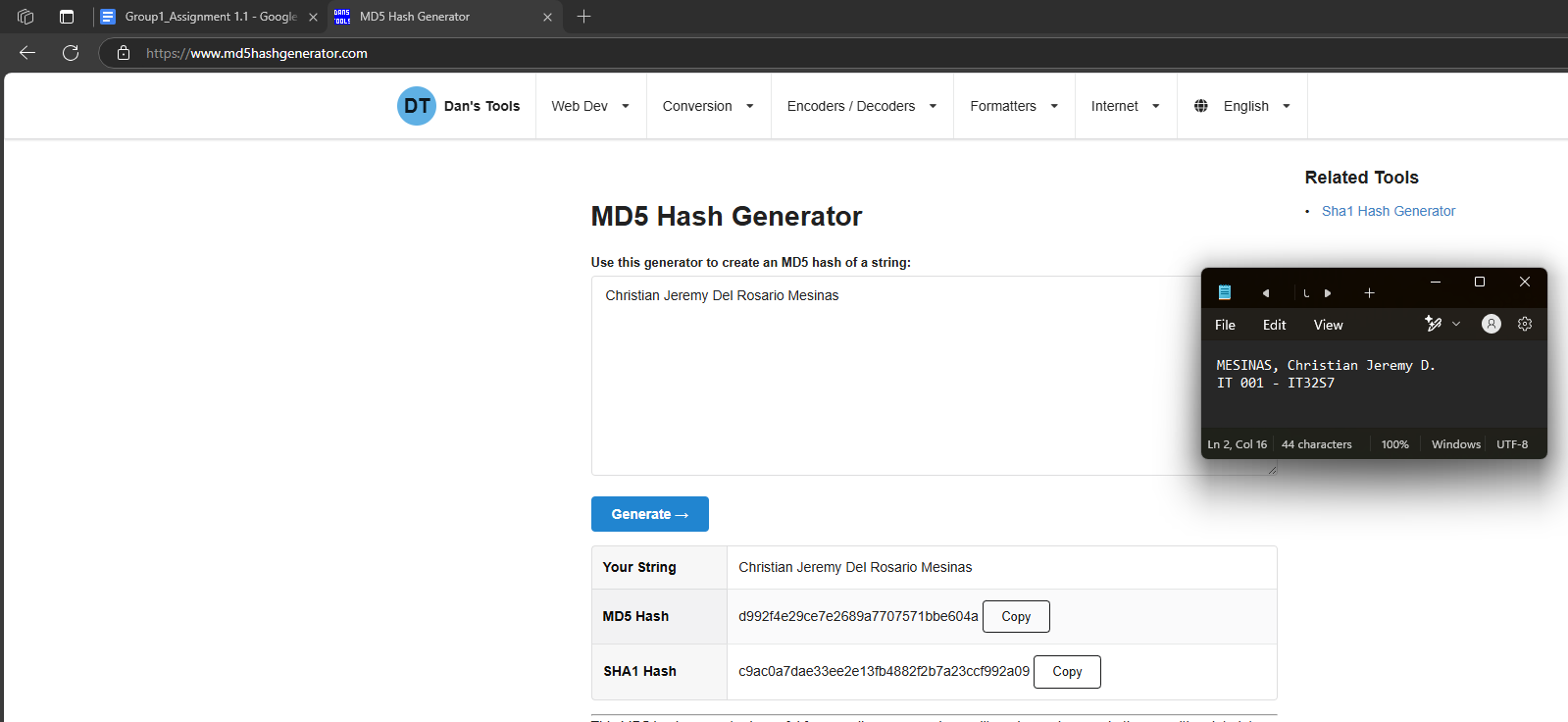
Document the features of each tool. (e.g., speed, user interface, supported algorithms)

Online Hashing Tools and FileFormat.Info, offers various features that are essential for cryptography and data integrity. Online Hash Tools is a free and ad-free platform that provides a user-friendly interface with no popups, supporting numerous algorithms like MD5, MD4, SHA1, SHA2, and others up to SHA-512. This ensures that it will generate efficient and random hashes while maintaining its quality of speed and responsiveness. In addition, another online tool that was used is FileFormat.Info. This online platform for hashing calculations is more versatile since it supports string, binary, and file hashing. This also has a lot of algorithms such as Adler32, CRC32, Haval, MD5, SHA variants, RipeMD160, Tiger, and Whirlpool. Both are efficient in their own ways, however, FileFormat.Info offers more for better cryptography needs.

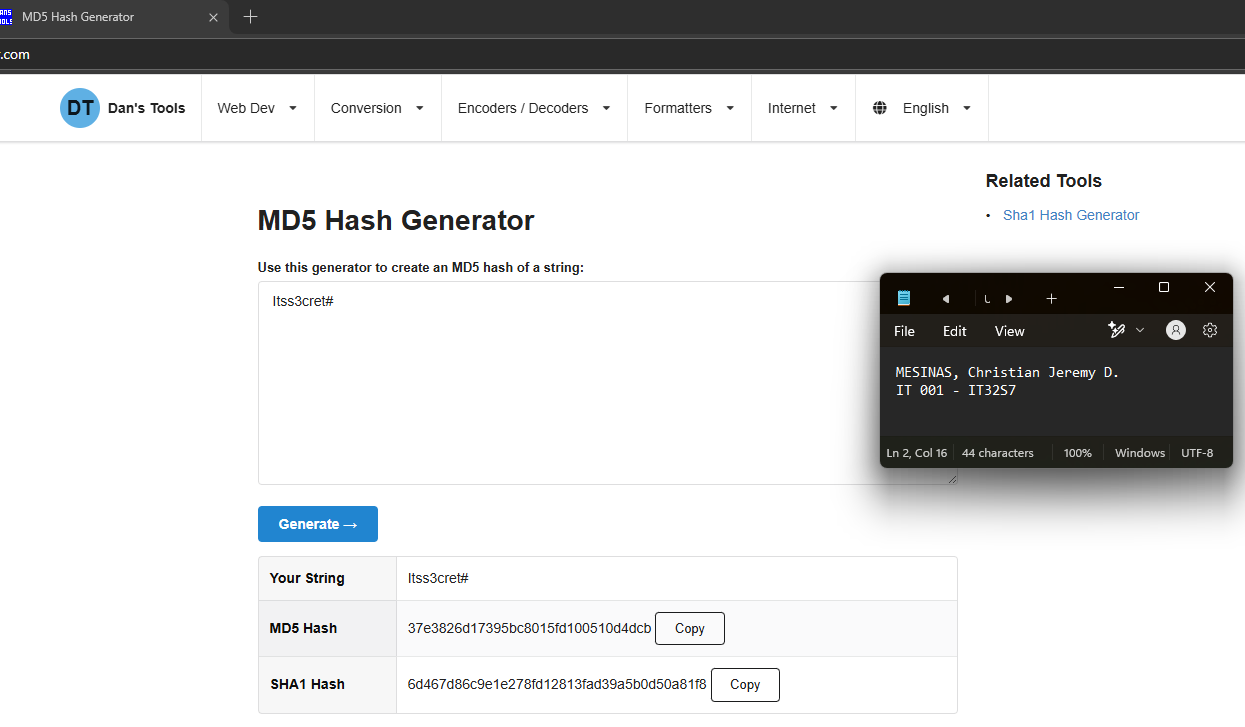
**MESINAS**

MD5 Hash Generator

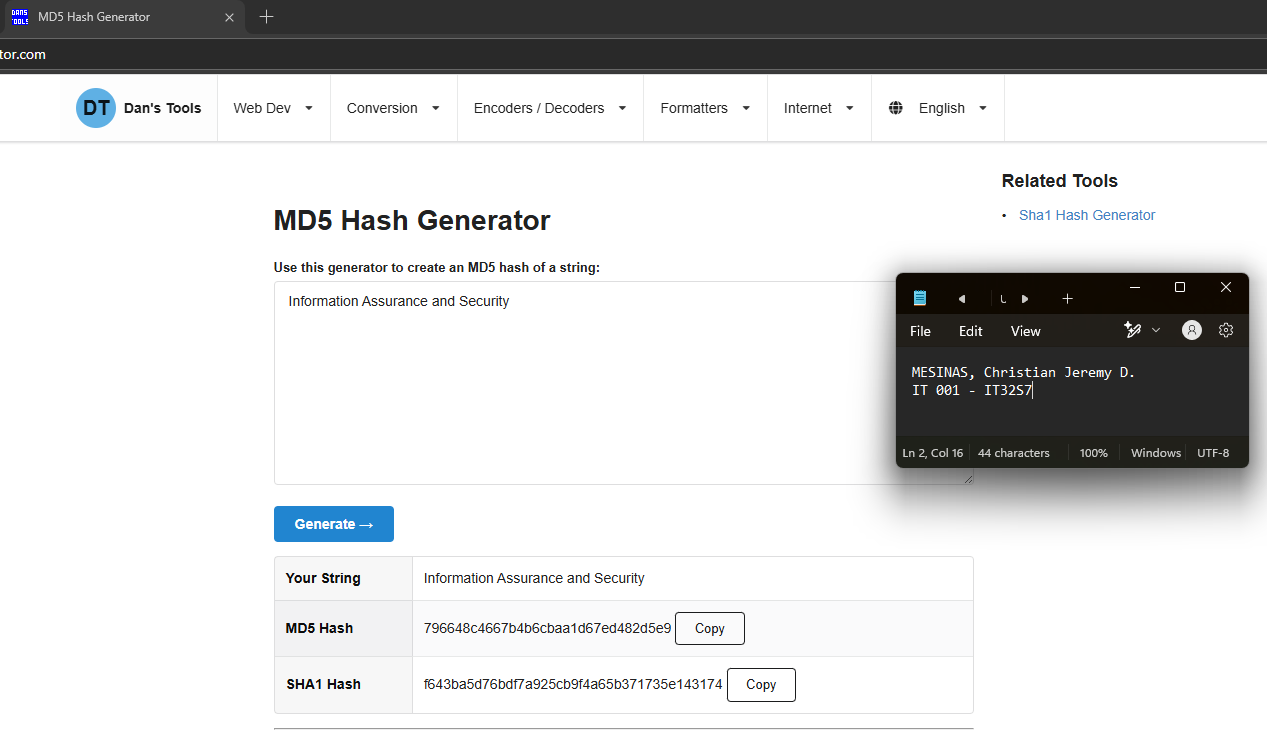
Full Name (Christian Jeremy Del Rosario Mesinas)



10 Character Random Password (Itss3cret#)

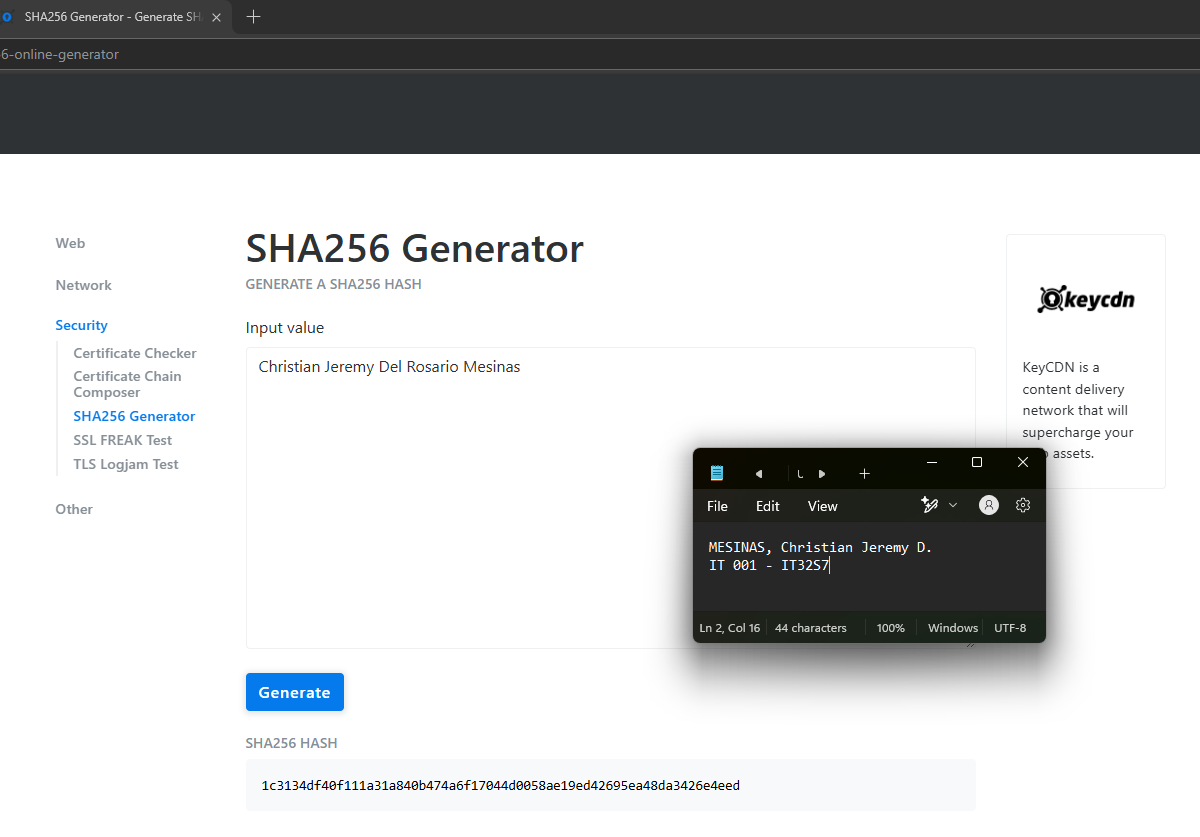


The phrase: *Information Assurance and Security*.

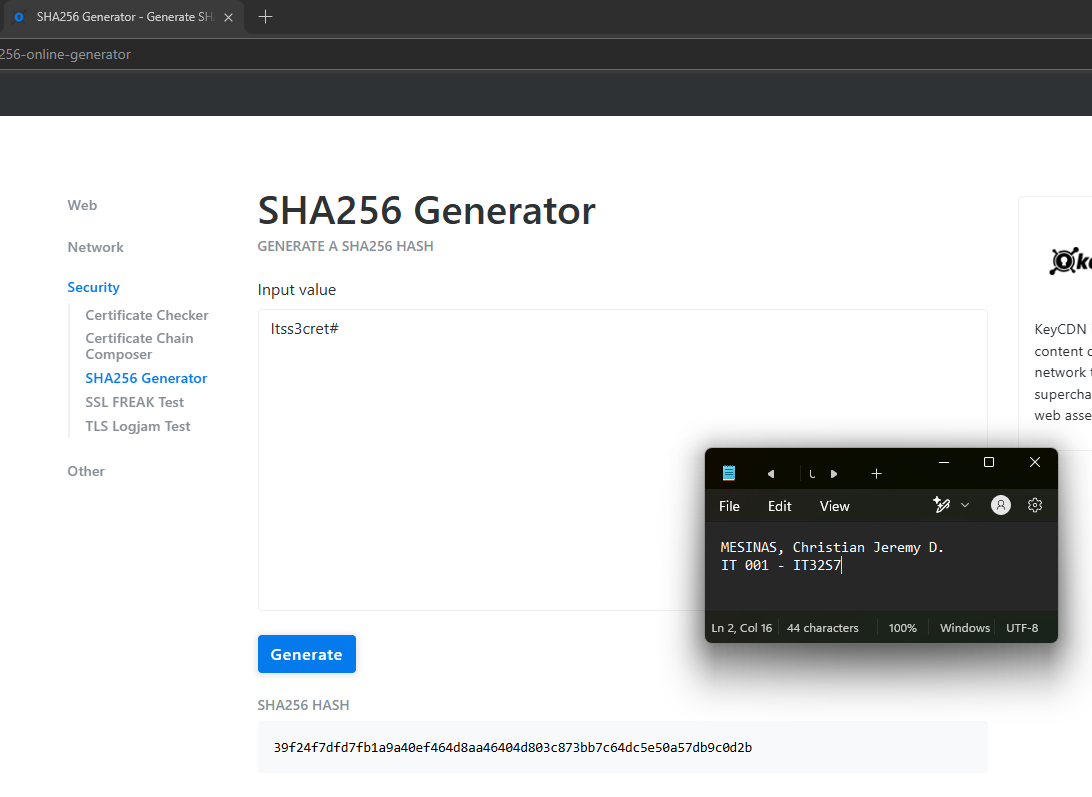


SHA256 Online Tool

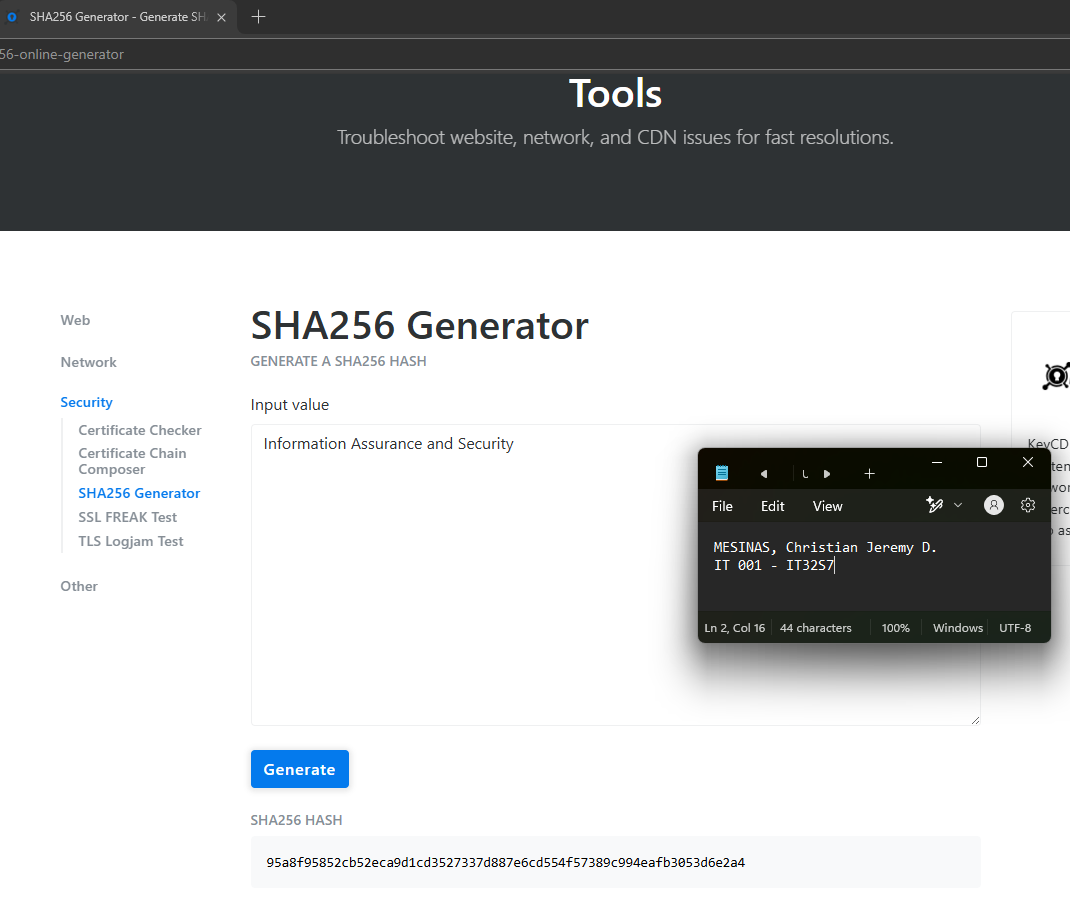
Full Name (Christian Jeremy Del Rosario Mesinas)



10 Character Random Password (Itss3cret#)

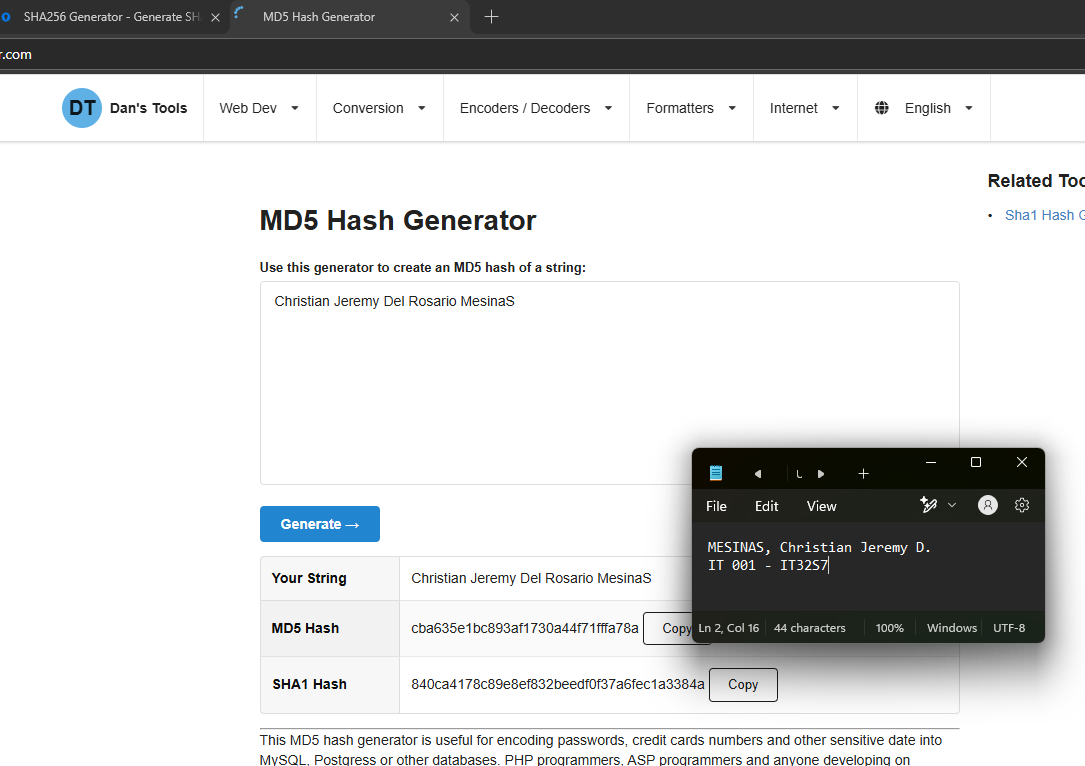


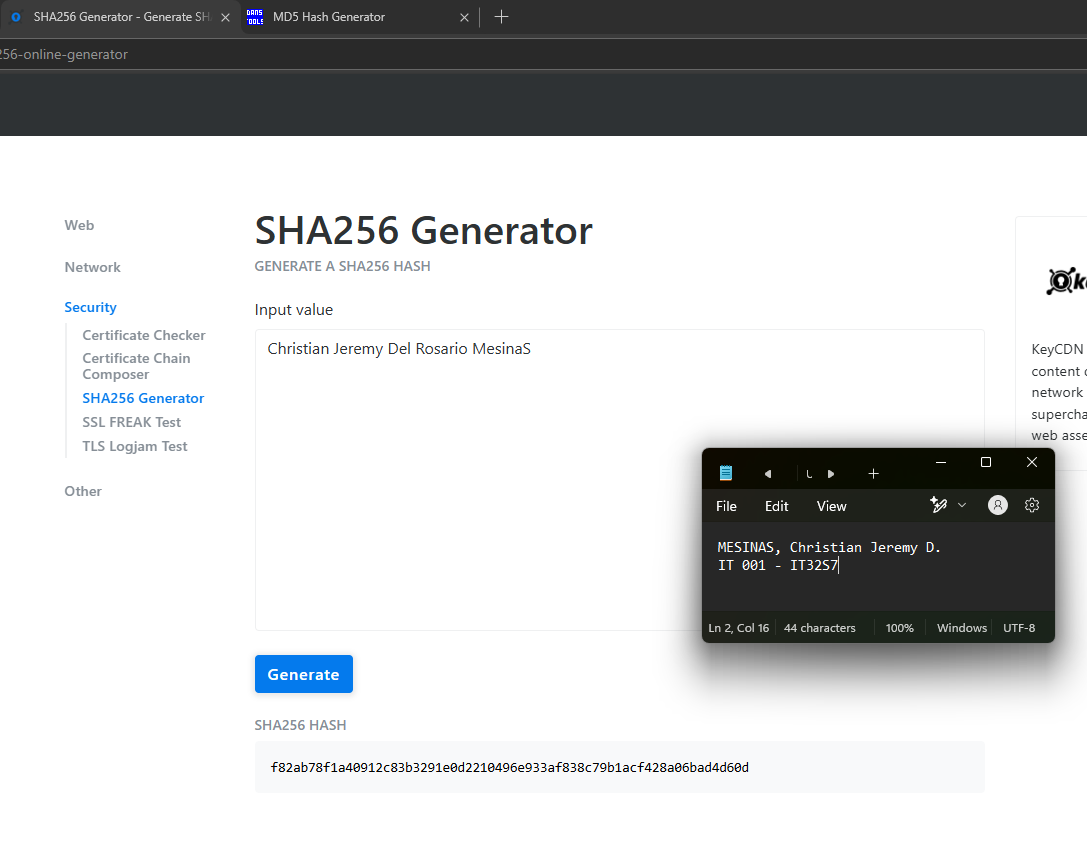
The phrase: *Information Assurance and Security*.



Compare the outputs and observe the differences when altering input data.

The MD5 algorithm generates a fixed 32-character hash of the input string, regardless of the plaintext’s length. Likewise, the SHA256 generates one with 64 characters. Altering input data will directly change the entire hash. The succeeding screenshot capitalizes the last letter of the student’s full name.





What happens when even one character of the input changes?

As seen in the previous two screenshots, changing even a single character in the plaintext will **significantly change** the hash output while the hash length remains unchanged.

What are the advantages and limitations of these hashing algorithms?

| **Algorithm** | **Advantages** | **Disadvantages** |
| --- | --- | --- |
| **MD5** | * Generates a robust 16-byte password * Low memory required * Simple understandability * Faster hash production compared to advanced algorithms | * Neither asymmetric or symmetric * Susceptible to brute force attacks * Susceptible to collisions wherein two distinct strings produce the same hash |
| **SHA - 256** | * Fast hash production due to simple structure. * Currently has no known vulnerabilities. * Currently effectively deters tampering and is applied in data integrity critical tasks such as backups and forensics. | * Vulnerability to length extension attacks. * Susceptible to quantum attacks. |

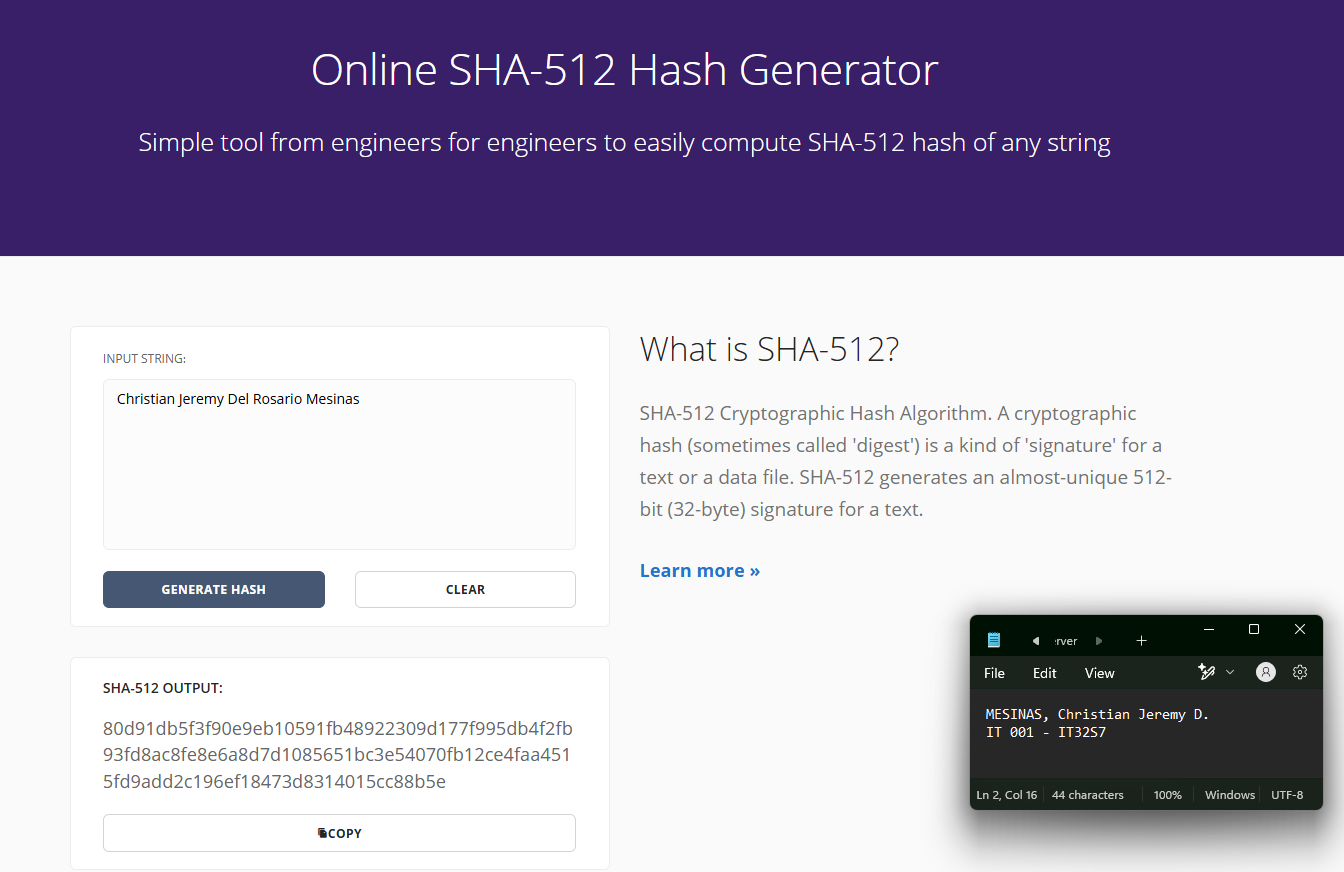
Search for and identify at least two other online tools that support hashing (e.g., SHA1, SHA512).

1. SHA-512 Hash Generator: <https://sha512.online/>
2. SHA 1: <https://emn178.github.io/online-tools/sha1.html>

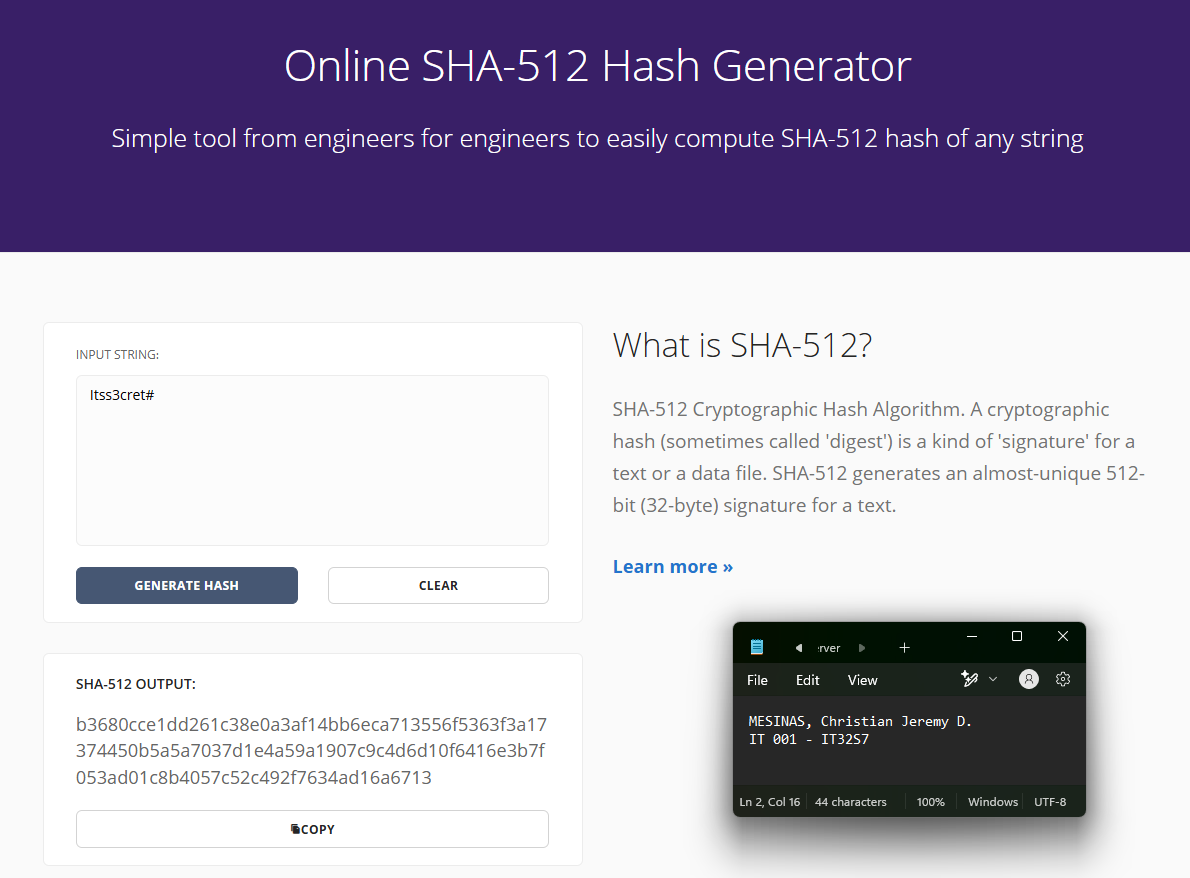
Test these tools using the same inputs as above.

SHA 512

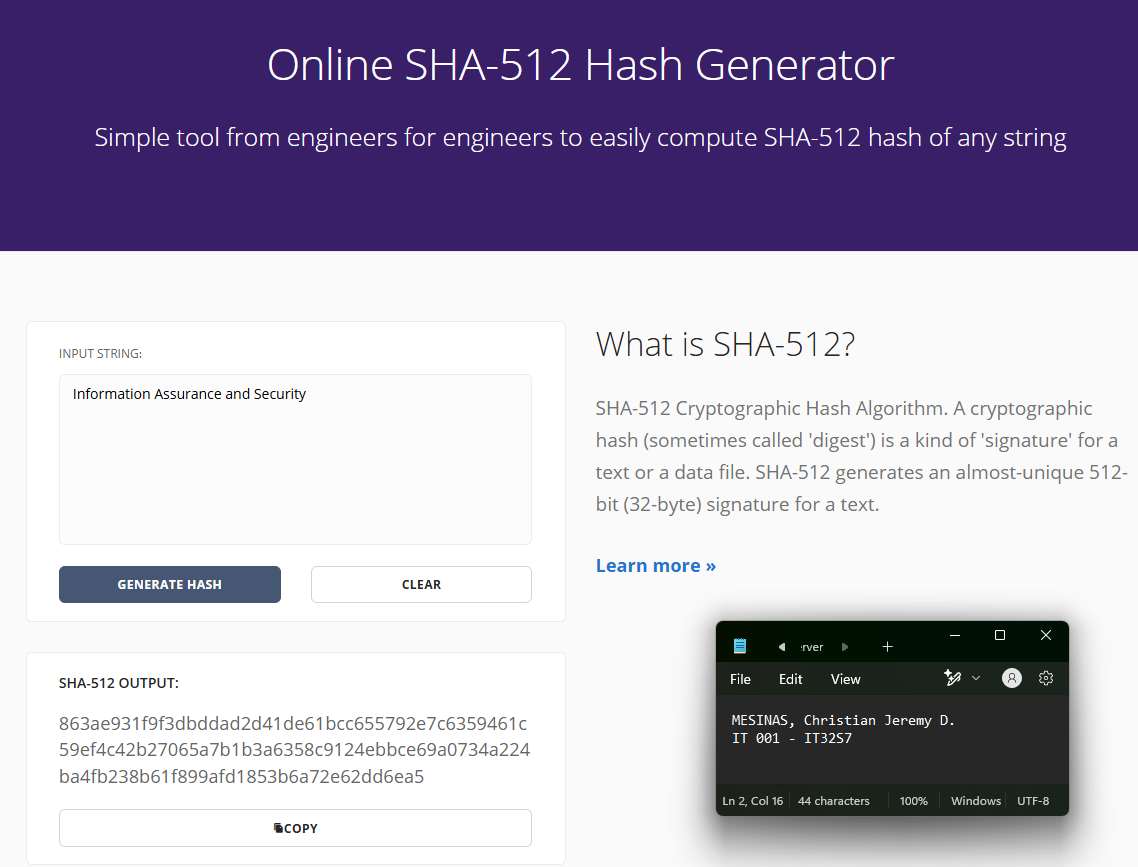
Full Name (Christian Jeremy Del Rosario Mesinas)



10 Character Random Password (Itss3cret#)

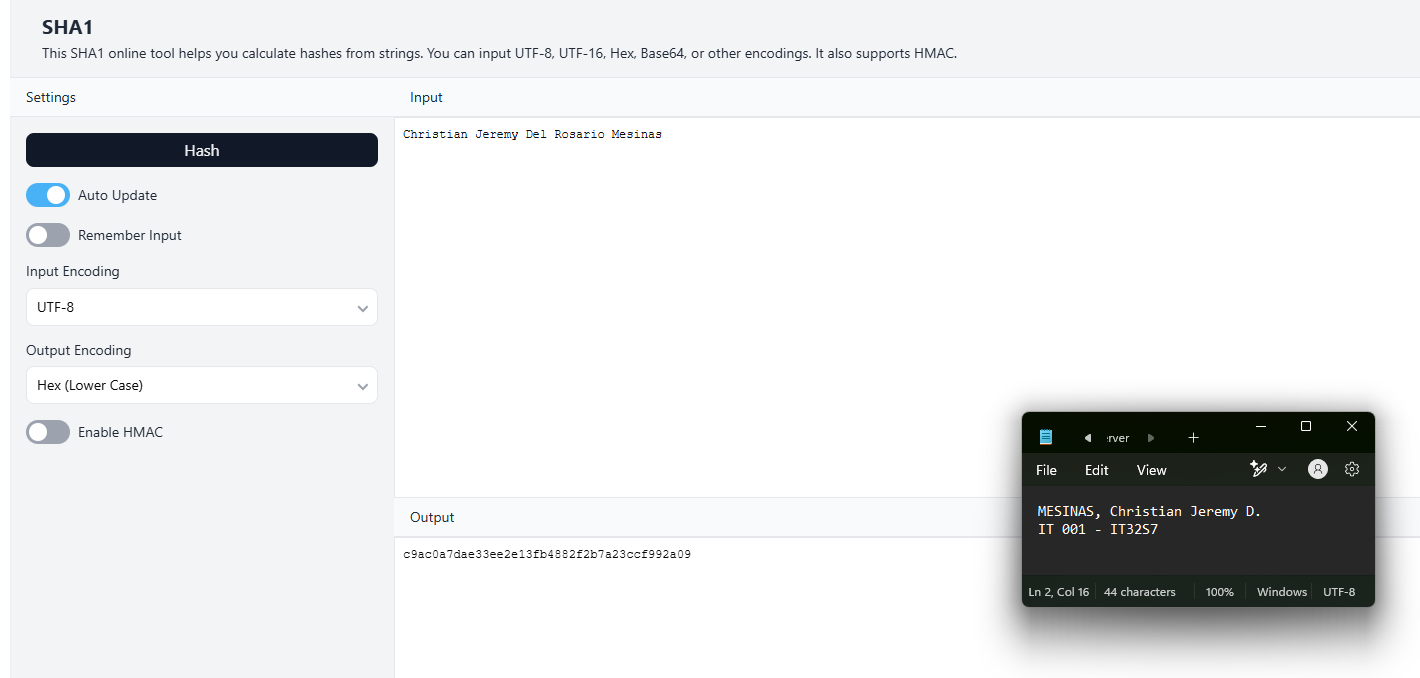


The phrase: *Information Assurance and Security*.

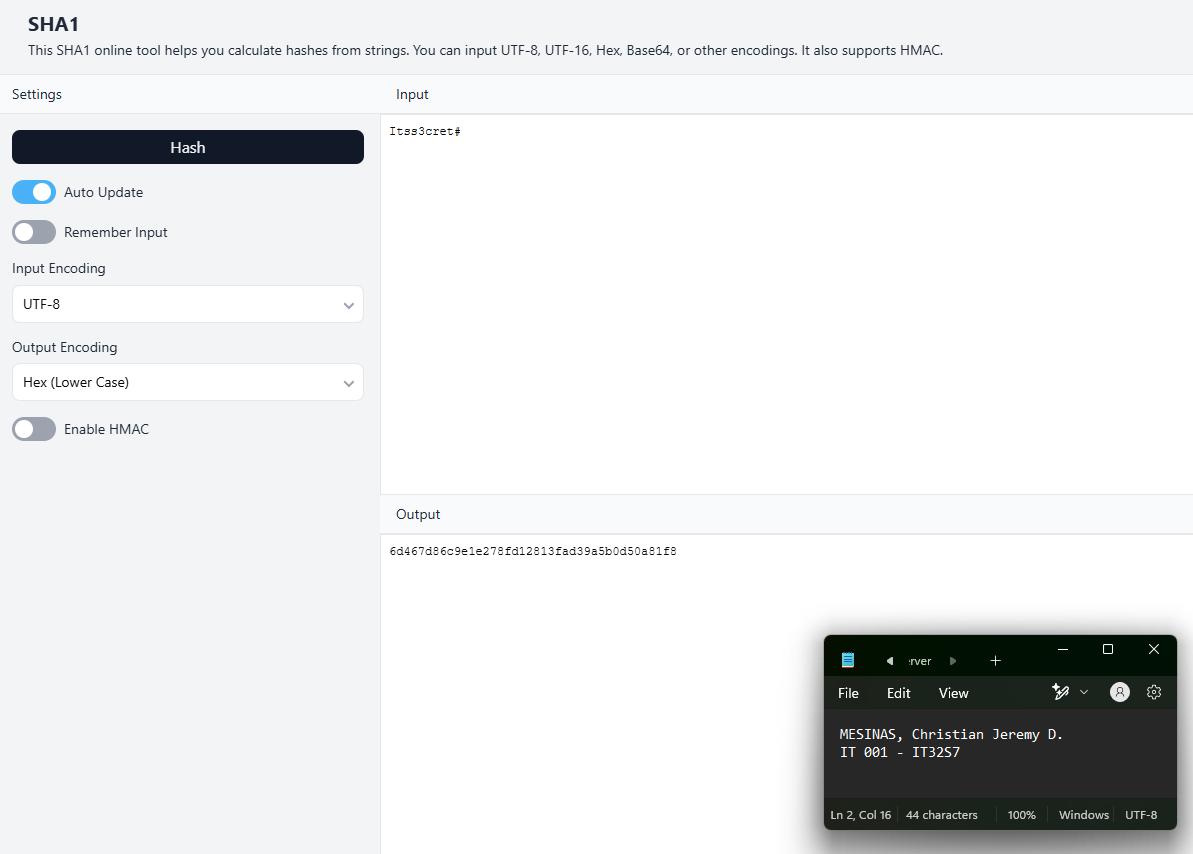


SHA 1

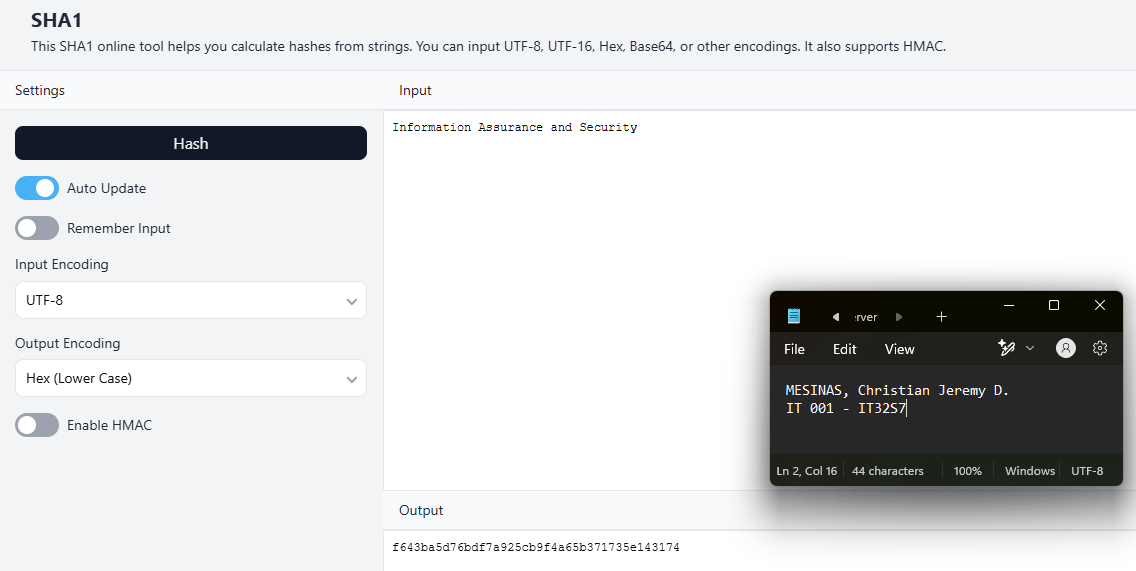
Full Name (Christian Jeremy Del Rosario Mesinas)



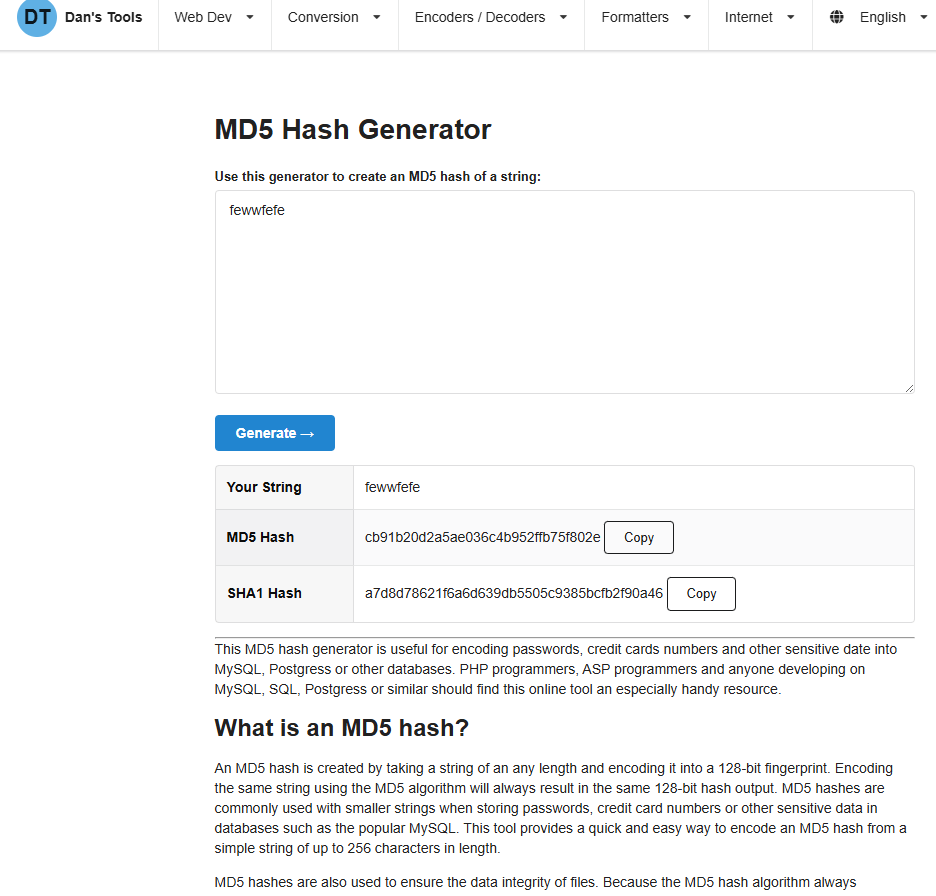
10 Character Random Password (Itss3cret#)



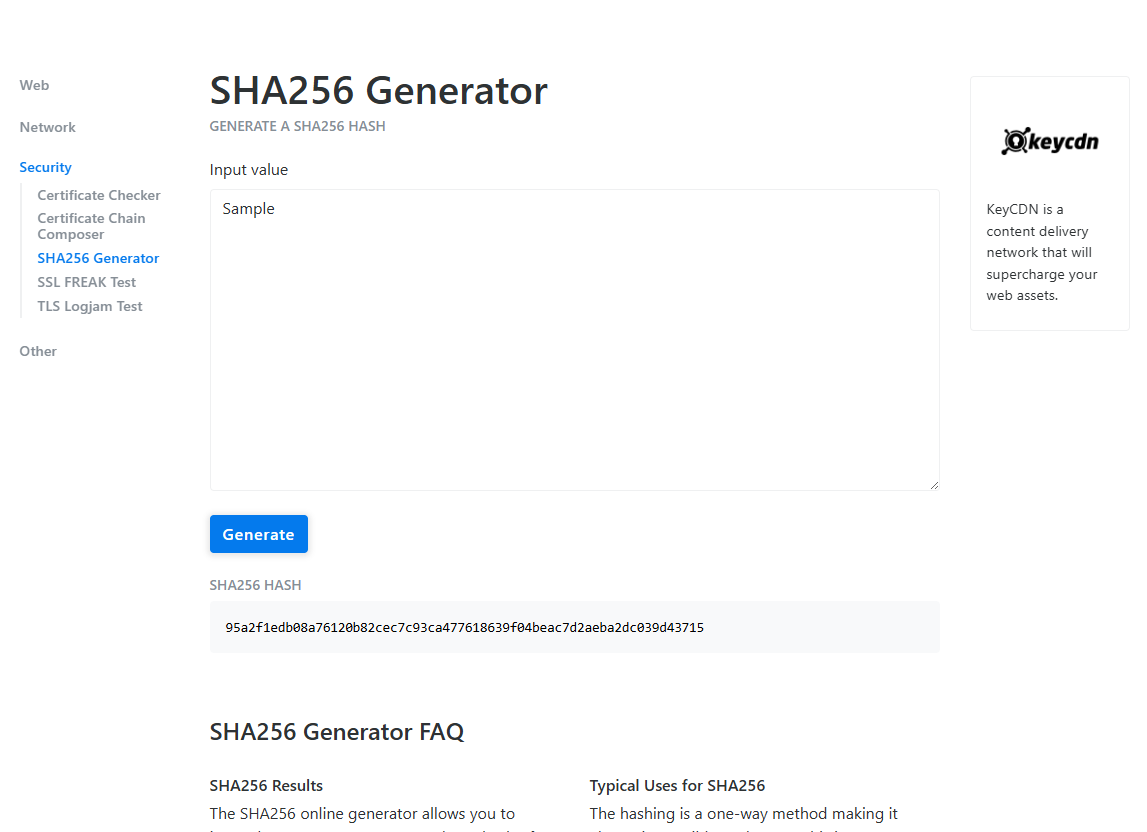
The phrase: *Information Assurance and Security*.



Document the features of each tool (e.g., speed, user interface, supported algorithms). Write a brief paragraph comparing the tools used.

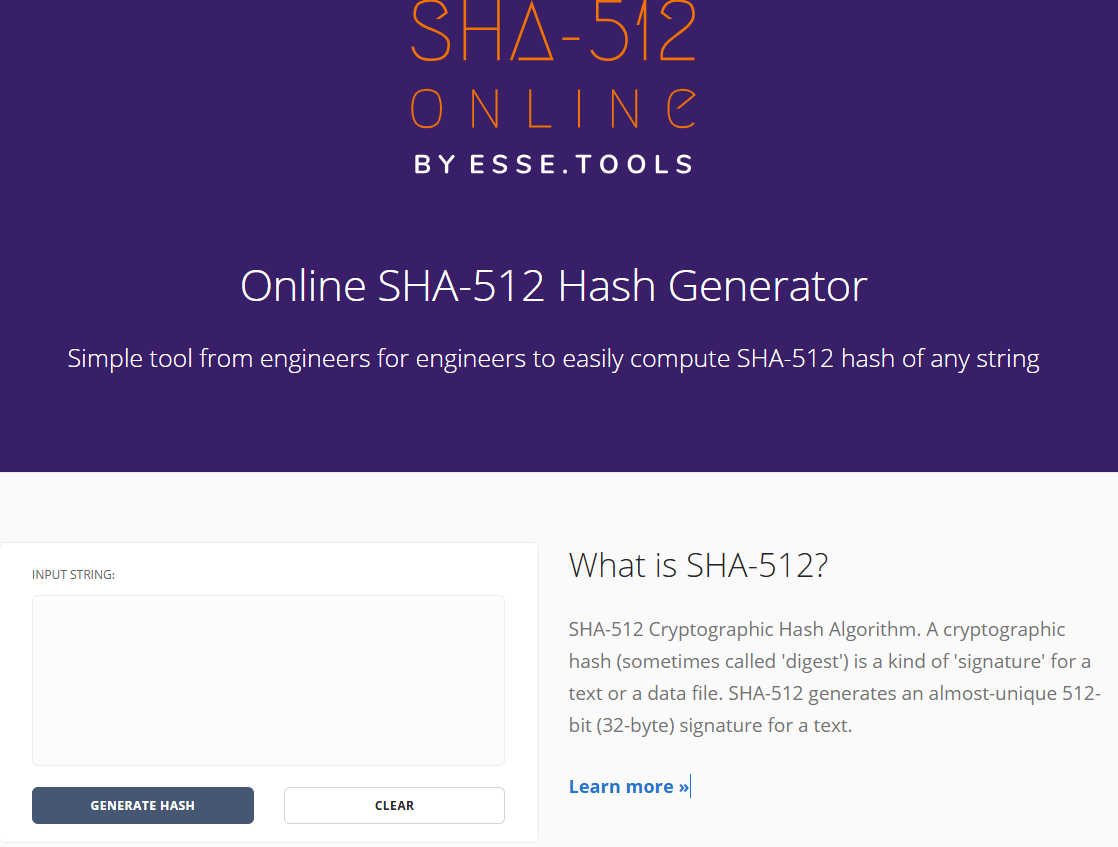


For the first tool, the MD5 hasher from the webpage “www.md5hashgenerator.com” takes input in the text area and requires a button click to generate the hash. The page requires a full reload before the hash is generated. They also provide a SHA1 hash along with the intended MD5 hash. Moreover, providing information about the hash function enables users to understand the algorithm.

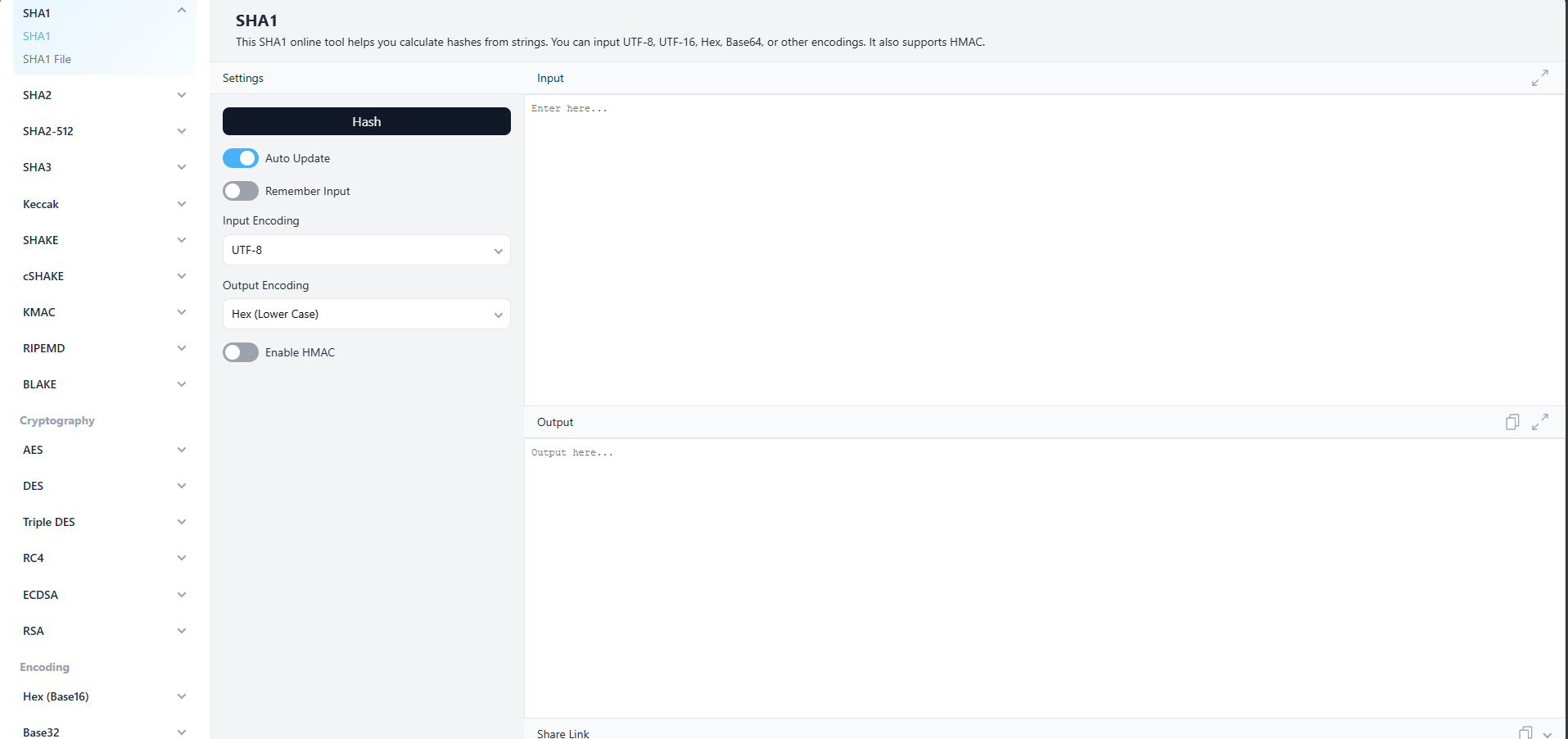


Next, the SHA256 tool provided by the website “tools.keycdn.com” takes an input value and a button trigger to generate a hash. Similar to the MD5 hasher, the page reloads before the hash is generated. The platform offers four service categories, including web, security, network, and other.

In terms of speed, the hash is immediately shown after a full, successful reload.



The third tool, a SHA-512 Hash Generator by the website “<https://sha512.online/>,” provides a brief description of the algorithm, similar to the tools above. However, it appears that SHA-512 is the sole algorithm provided by the platform. In terms of user experience (UX), I found it helpful that a clear button was implemented for faster text testing. Similarly, the platform also requires a full reload before generating the hash.



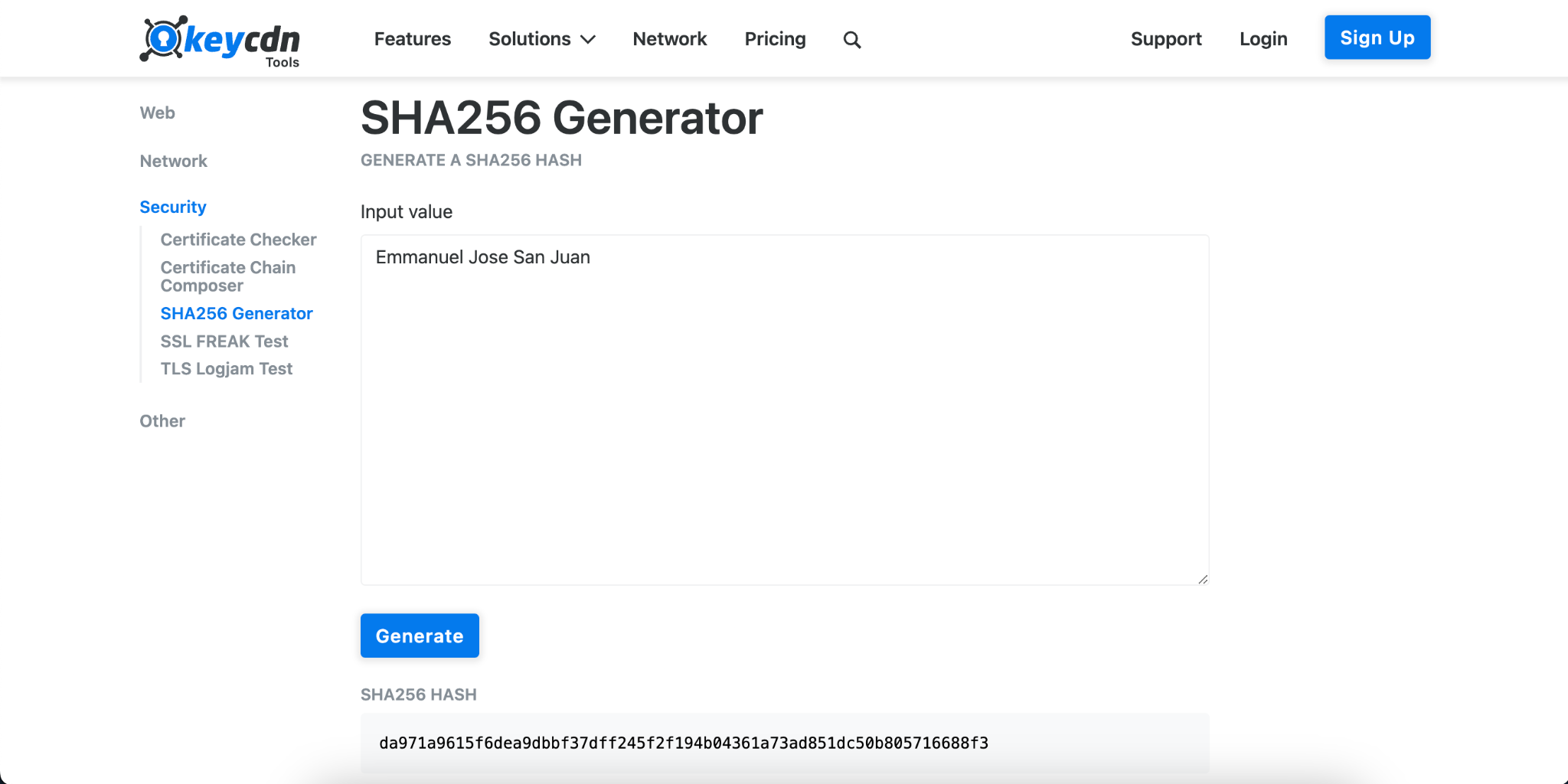
Lastly, the SHA-1 generator provided by “<https://emn178.github.io/online-tools/sha1.html>” immediately generates the hash without reloading. The sidebar also asks for input and output encoding according to user preference. A wide range of hashing algorithms were also provided, from SHA1 to SHA3. Alternatively, a SHA256 algorithm is also available through this platform. However, no informative description of the algorithms was provided, unlike the previous tools.

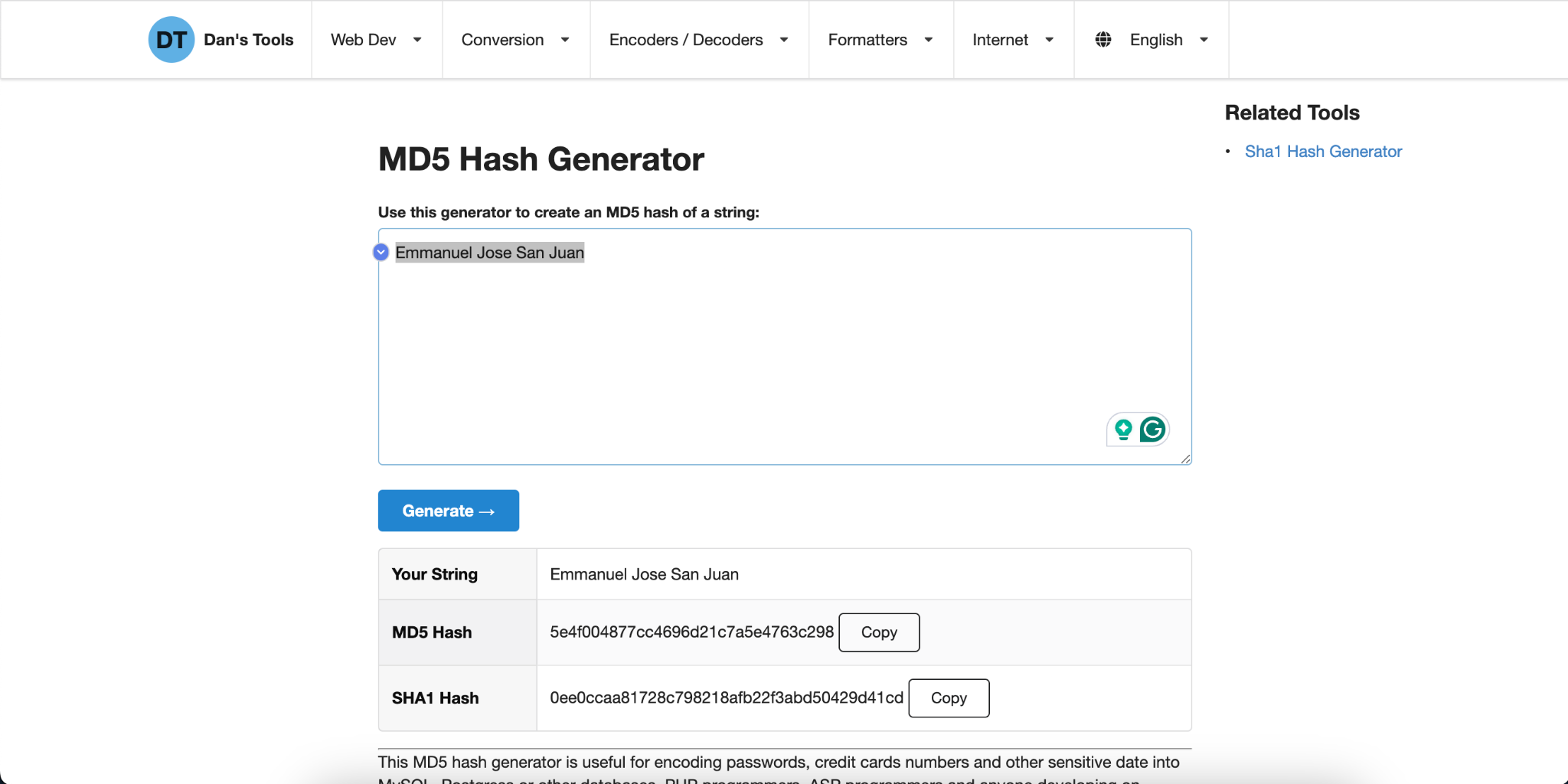
Which tool do you find the most user-friendly and why?

Subjectively, I found the SHA1 algorithm I researched to be the most user-friendly tool because of its real-time hash generation, its options to remember past inputs, and its consideration of input and output encoding. Moreover, the wide range of algorithms not limited to hashing helps end users with other tasks they may have. At the bottom of the page, JSON and XML converters were also provided.

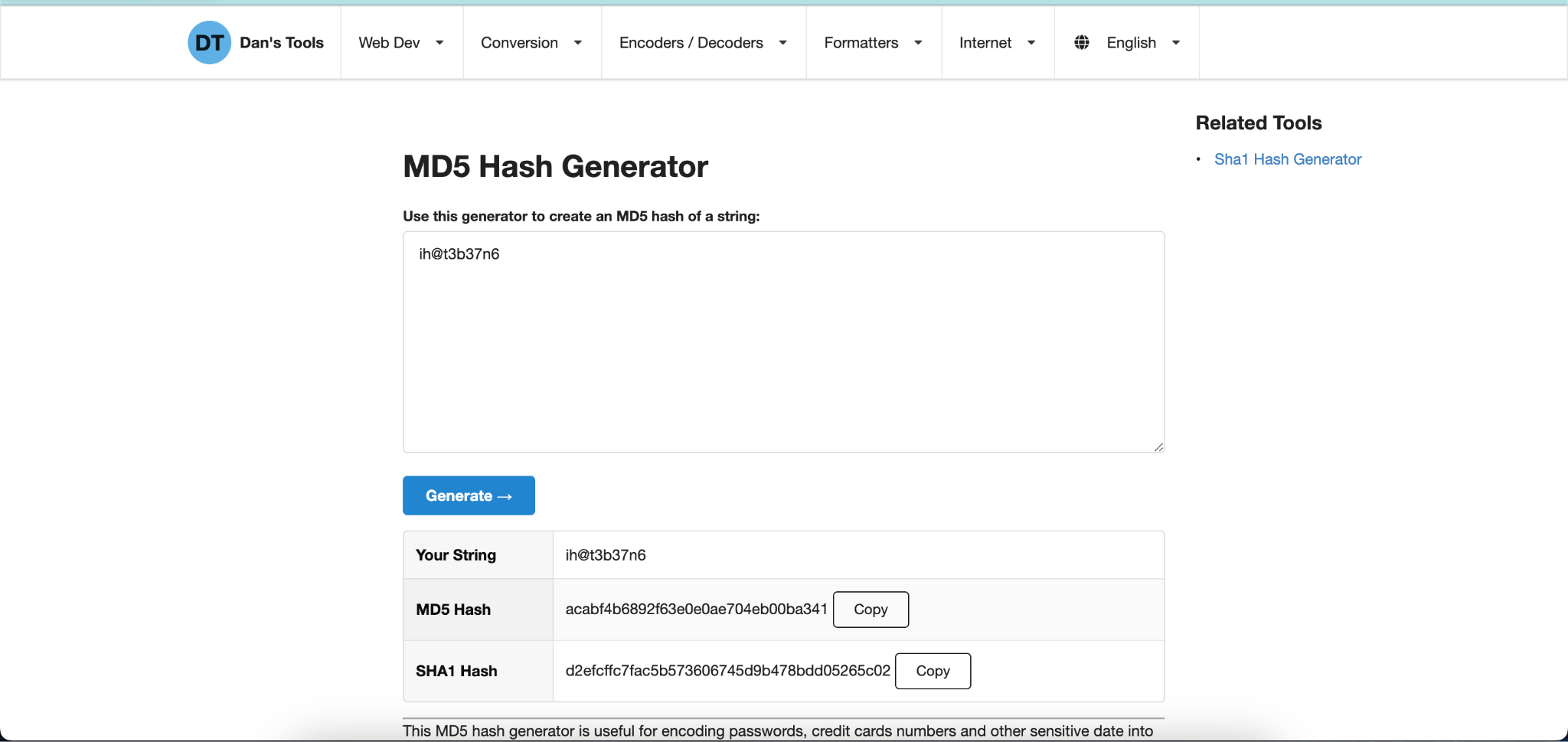
**SABINO**

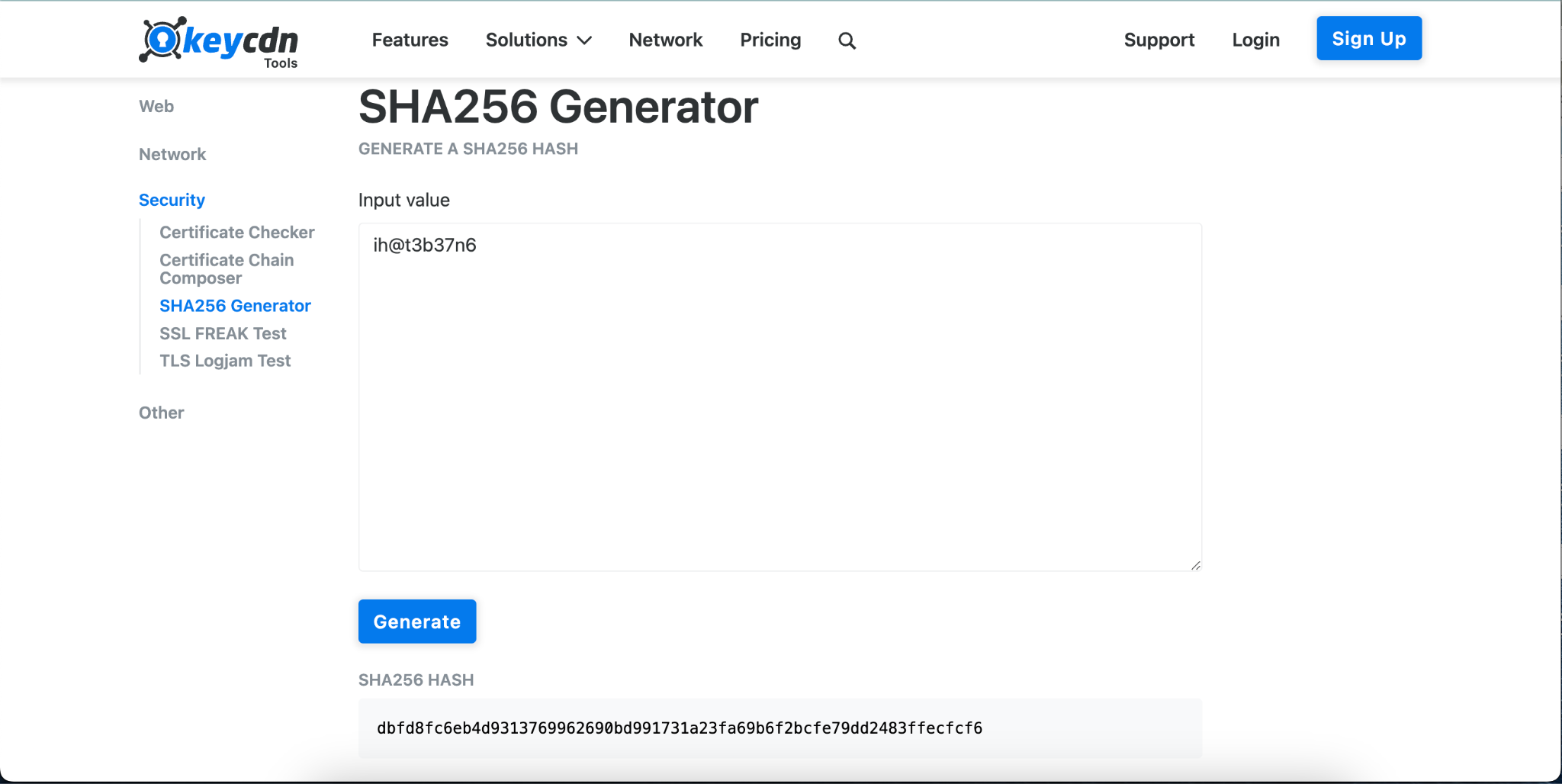
**SAN JUAN**

****

****

(Full Name: Emmanuel Jose San Juan)

****

(10 Character Password Used: ih@t3b37n6)  
  


**Compare the outputs and observe the differences when altering input data:**

MD5 and SHA256 are designed to produce a fixed-size output (hash value) from input data of any size. Even a small change in the input data (e.g., altering one character) will result in a completely different hash value. This property is known as the avalanche effect, also Secure Hash Algorithm has longer values than Message Digest Algorithm.

**What Happens When Even One Character of the Input Changes?**

The output hash changes completely due to the avalanche effect, This ensures that hashes are unique for different inputs, making them useful for verifying data integrity, However, this also means that hashing is a one-way process: you cannot reverse-engineer the original input from the hash.

**Advantages and Limitations of Hashing Algorithms**

Advantages:

Data Integrity: Hashes can verify if data has been altered (e.g., file checksums).

Fixed-Length Output: The hash output is always the same length regardless of input size.

Efficiency: Hashing is computationally fast for large datasets.

Uniqueness: Ideally, different inputs produce different hashes (though collisions can occur).

Limitations:

Collisions: Two different inputs may produce the same hash (more likely in weaker algorithms like MD5).

Irreversibility: Hashes cannot be reversed to retrieve the original input.

Vulnerability to Attacks: Older algorithms like MD5 are susceptible to collision attacks and are no longer considered secure for cryptographic purposes.  
 **Other Online Tools That Support Hashing  
  
Online Hash Tools:** [**https://onlinehashtools.com/**](https://onlinehashtools.com/)

**CyberChef** : [**https://gchq.github.io/CyberChef/**](https://gchq.github.io/CyberChef/)

**PART 3: Group Synthesis**

Hashing ensures that data has not been tampered with or altered. By comparing the hash of the original data with the hash of the received data, you can verify if the data is intact and unchanged. For example, Downloading a file from the internet. The website provides the file's hash (e.g., SHA256). After downloading, you hash the file and compare it to the provided hash. If they match, the file is authentic and unaltered, and ​​Hashing is used in digital signatures to ensure the authenticity of messages or documents. The sender hashes the message, encrypts the hash with their private key, and sends it along with the message. The recipient decrypts the hash with the sender's public key and cross-checks it with the hash of the received message.

From what we have researched, hashing is a one-way process of data that converts into a fixed-size sequence of characters known as a hash value. Hashing is irreversible, and because of that, it is a valuable security technique. Because of the way it processes the data since the original data cannot be retrieved from the hash, it ensures that sensitive data or information will be protected even though the hash will be exposed. It is an important method for maintaining the security and integrity of the data.

In real-world situations, the basic security that the group members have used is two-factor authentication with the use of email OTP and SMS OTP before logging in to an account. The access control may be the Google Docs and Google Drive access permission if the owner of the file decides if they will allow the file to be accessed by anyone or just by granting certain people access to it, whether it is by viewing or editing. Lastly, the encryption in Messenger is peer-to-peer messaging that whenever a user sends messages, it will surely be encrypted before sending the data to the other end user I messaged.

**References (APA):**

Cloudflare. (n.d.). What is SSL/TLS? Retrieved from https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/

Crane, C. (2021). What Is a Hash Function in Cryptography? A Beginner’s Guide. https://www.thesslstore.com/blog/what-is-a-hash-function-in-cryptography-a-beginners-guide

Encryption Consulting. (n.d.). What is SHA- 256? https://www.encryptionconsulting.com/education-center/sha-256/

Fromaget, P. (2023). 3 Reasons why MD5 is not Secure. https://infosecscout.com/why-md5-is-not-safe/

GeeksforGeeks. (2024). Cryptography Hash Functions. https://www.geeksforgeeks.org/cryptography-hash-functions/

GeeksforGeeks. (2024). Difference between Hashing and Encryption. https://www.geeksforgeeks.org/difference-between-hashing-and-encryption/

GeeksforGeeks. (2024). SHA256 and SHA3. GeeksforGeeks. https://www.geeksforgeeks.org/sha-256-and-sha-3/

GeeksforGeeks. (2024). What is the MD5 Algorithm? https://www.geeksforgeeks.org/what-is-the-md5-algorithm/

Grassi, P. A., et al. (2017). Digital identity guidelines: Authentication and lifecycle management (NIST Special Publication 800-63B). National Institute of Standards and Technology. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf

IBM (n.d.). What is cryptography? https://www.ibm.com/think/topics/cryptography

Investopedia. (2024). How cryptocurrencies work. Retrieved from https://www.investopedia.com/terms/c/cryptocurrency.asp

OWASP Foundation. (n.d.). Password storage cheat sheet. Retrieved from https://cheatsheetseries.owasp.org/cheatsheets/Password\_Storage\_Cheat\_Sheet.html

Spasojevic, A. (2024). What Is MD5 (Message-Digest Algorithm)? https://phoenixnap.com/glossary/md5-message-digest-algorithm

TechTarget. (n.d.). What is encryption and how does it work? Retrieved from https://www.techtarget.com/searchsecurity/definition/encryption#:~:text=Encryption%20ensures%20that%20unauthorized%20third,when%20transmitted%20across%20public%20networks.

The Investopedia Team. (2024, May 18). Cryptographic hash functions: Definition and examples. https://www.investopedia.com/news/cryptographic-hash-functions/

TutorialsPoint. (n.d.). Cryptography - Hash functions. Retrieved from https://www.tutorialspoint.com/cryptography/cryptography\_hash\_functions.htm

Yasar, K. (n.d.). Hashing. Retrieved from https://www.techtarget.com/searchdatamanagement/definition/hashing

**Honor Pledge**

*“We affirm that we have not given or received any unauthorized help on this assignment and this work is my own.”*