LESSON

Warm Up: The Avengers have successfully collected all six of the Infinity Stones and are celebrating on Earth. All is well, until Thanos launches a surprise attack on them in an attempt to steal the stones! Unprepared for the attack, the Avengers flee to the corners of the Marvel Universe. All six of the stones have were gabbed by some of the Avengers before they fled, but which ones?

Your mission is to communicate with your fellow Avengers to uncover the owner and location of each of the Infinity Stones, and to report that information to the Gatherer. Beware though! Thanos has been left on Earth with access to all of the Avengers communication lines. He can see everything that you say. Could there still be a way to communicate securely to find all of the Infinity Stones before he does?

Lesson:

Student Walkthrough

Your students will need to understand what chaffing and winnowing is and how it is beneficial to concealing information. The idea comes from the agricultural meanings of either word. Chaff is defined as the extra husks or unwanted seeds that are mixed in with the desired seeds being harvested at a given time. Winnowing is the process of blowing air through the mix to remove the chaff. In computer science terms, this is the process of adding extra, fake information to data (chaffing) being shared between authorized entities, and then sifting through all said data (winnowing) to find the correct information. This technique is used when trying to conceal the true message from a third party who is not authorized to have access to the data.

Another crucial concept is hashing, which is a way of uniquely identifying some data. A hashing algorithm will always produce hashes of the same length for any amount of data supplied. Even the smallest change in the supplied data will result in a major change in the hash. There are many different algorithms for creating hashes. Some algorithms are weaker, and some algorithms are stronger. Some algorithms are keyless, and some algorithms are keyed. In the case of the game, a key is used in conjunction with the message.

Note: Stronger is not always better. More complex algorithms may require more computation and more power from the device performing the work. A software solution must be paired with the appropriate hardware and achieve operational goals to be successful.

DATA + KEY → [HASHING ALGORITHM] → HASH→ [HASHING ALGORITHM] → HASH

As stated before, the smallest change in the data, or key for that matter, will result in a completely different hash. By using the key, we can ensure a completely unique hash for the data. This key is shared among authorized entities only, and therefore any third party without the key will be unable to successfully verify the data.

For example, say Alice and Bob are trying to communicate but don't want Eve to know what they are saying.

Alice wants to send Bob the message "Meet me at the theater next Friday at noon".

First the message is broken up into parts:

(Meet me) (at the theater) (next Friday at noon)

The message is then chaffed with extra information:

(Meet me) (at the theater) (next Friday at noon)
(Meet Josh) (at the lake) (next Sunday at 8)

Each portion of the message is hashed using the key that both Alice and Bob share, but the fake message hashes are computed with the data, the key, and additional random data that will alter the hash from what it would have been with just the original data and key. Now the message is ready to be sent.

Alice sends:

Data	Hash
(Meet me)	RCZJbSs07H
(Meet Josh)	DPFB4rSwkR
(at the theater)	rdJOSDcQMw
(at the lake)	qoGgd4da4K
(next Friday at noon)	ZYgxtTQz69
(next Sunday at 8)	wod6QU0xND

When Bob and any others listening receive this message, they will not be able to understand what the true message is right away since it is mixed in with the fake message. However, since Bob has the key, he can winnow by computing the hashes for each of the pieces of data sent. Next, he can compare the hashes generated to those that were sent from Alice, and if they match, he knows that the data is a part of the true message. However, because Alice computed the hashes for the fake data with an extra, random component that was not shared with Bob, the hash that he computes for any of the fake data will not match the hash provided by Alice for that same data. This lets Bob know that this bit of data is not a part of the true message.

Bob receives:		**Bob computes:**	
Data	**Alice Hash**	**Bob Hash**	**Real or Fake**
(Meet me)	**RCZJbSs07H**	**RCZJbSs07H**	**Real**
(Meet Josh)	**DPFB4rSwkR**	**BRqdHhjpkk**	**Fake**
(at the theater)	**rdJOSDcQMw**	**rdJOSDcQMw**	**Real**
(at the lake)	**qoGgd4da4K**	**KFljykBmJg**	**Fake**

(next Friday at noon)	**ZYgxtTQz69**	**ZYgxtTQz69**	**Real**
(next Sunday at 8)	**wod6QU0xND**	**UeFtNbWeIT**	**Fake**

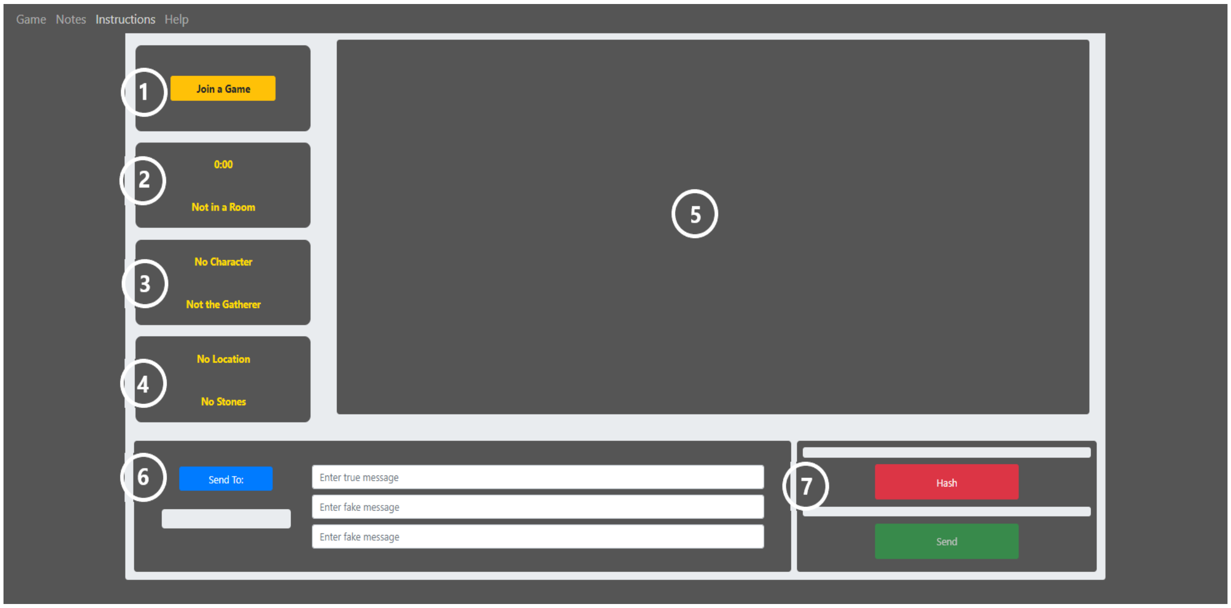
Bob now has successfully winnowed away the fake data to arrive at the real message. Any third party receiving this data will have no way of winnowing, and as such, will not be able to separate the real message from the fake.

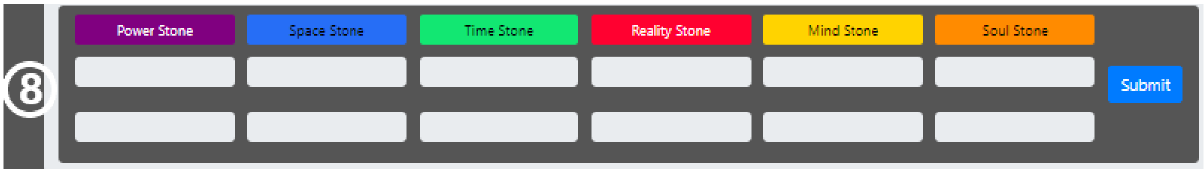
Instructions

The Game Board:

After powering on and logging into your machines, your first step will be to open up a web browser (Google Chrome is preferred). Once your browser is open, you will need to go to https://stonehunt.tntech.edu to visit the home page for Stonehunt!

At the top of this page you will see a navigation bar. Select 'Play' and you will be brought to the page where you will get to play the game! It will look like the following:





Along the top of the page you will see a second navigation bar. Clicking between the options will show the game, notes, instructions, and a help page for the game. During the game, you may switch freely between these tabs. Below that is the actual game board. Each section is numbered and described below.

1) The 'Join a Game' button is, of course, for joining a game. More on that later.

2) This is your game timer which shows how much time is left in your game. Note that you cannot send messages before the timer starts, or after the timer ends! Next is your Room name (you will not need to remember this for the game).

3) Following that is your Character which you will be allowed to choose. Below that you will see a message saying whether you are the Gatherer or not (If you are the Gatherer, you will also receive a prompt when the game starts).

4) Last but definitely not least, you can see the location and stone(s) that have been assigned to you for the game. These will be very important!

5) This is your message window where you will be able to view all of your communications with the other Avengers!

6) This is where you will construct your messages. On the left you have a button to select an Avenger to talk to and a label of the currently selected Avenger. Nest to that are the message inputs where you will enter in your messages.

7) After constructing messages, this is where you will hash and send them on to your teammates!

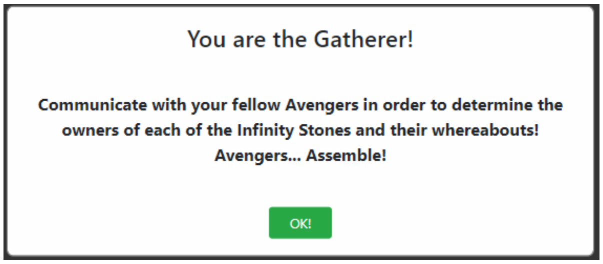
8) When making a guess, areas 6 and 7 will be replaced with this window where you can select an owner and location of each of the Infinity Stones and submit a guess.

9) **How to Play:**

In each game, the Avengers will face off against Thanos. In order to win, the Gatherer must guess the owner and location of more stones correctly than Thanos, both of whom will be randomly selected at the beginning of the game.
To join a game, first click the 'Join a Game' button. You will be asked to enter a username. This is not used in the game, but it will allow you to join the game again in the case that your computer dies, you accidentally leave the page, etc. Once you have entered a valid username, you will see a button appear below which will let you select a character.

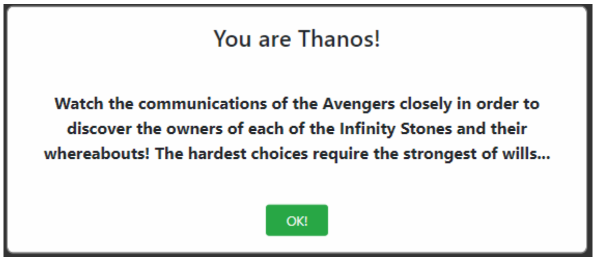
When you click the 'Select a Character' button, you will see a popup appear with all of the remaining Avengers in this game. Select one, and once all of the other players have selected one as well, the game will start.

If you have been selected as the Gatherer or as Thanos you will see one of the following popup messages:



You are the Gatherer!

Communicate with your fellow Avengers in order to determine the owners of each of the Infinity Stones and their whereabouts! Avengers... Assemble!

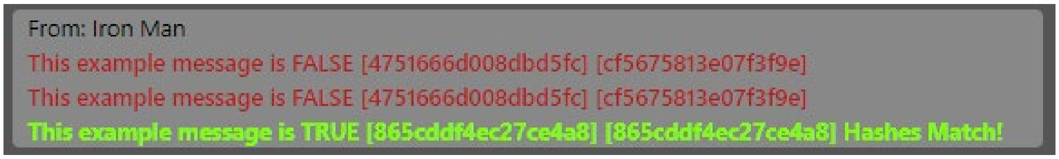


You are Thanos!

Watch the communications of the Avengers closely in order to discover the owners of each of the Infinity Stones and their whereabouts! The hardest choices require the strongest of wills...

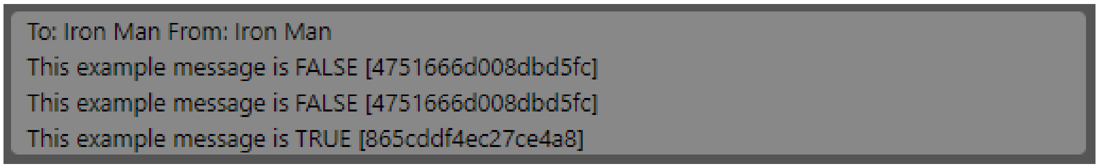
Thanos cannot send messages, but the Gatherer and the rest of the Avengers can. To send a message, you can first select who to send to by clicking the 'Send To' button and selecting a character from the popup window. Next, you can enter in your true message into the top message input. The below two inputs are for the 'chaff' data which we discussed earlier. Now you can hash your messages by clicking the 'Hash' button and send them by clicking the 'Send' button.

When you receive a message, it will appear as the following:



```
From: Iron Man
This example message is FALSE [4751666d008dbd5fc] [cf5675813e07f3f9e]
This example message is FALSE [4751666d008dbd5fc] [cf5675813e07f3f9e]
This example message is TRUE [865cddf4ec27ce4a8] [865cddf4ec27ce4a8] Hashes Match!
```

Notice that the messages have been rearranged. This is to prevent Thanos from immediately knowing the top message is true. Also notice that as an Avenger, when you receive a message that the 'chaffed' message will appear red, and the true message appears green. This is because the message has already been winnowed for you. Since Thanos does not have access to the key that all the Avengers share. As such, his messages will appear as follows:



```
To: Iron Man From: Iron Man
This example message is FALSE [4751666d008dbd5fc]
This example message is FALSE [4751666d008dbd5fc]
This example message is TRUE [865cddf4ec27ce4a8]
```

This makes the process of discovering the actual true message extremely difficult.

The Avengers have until time runs out to report their information to the Gatherer. When the time runs out, the only step left is to wait for the Gatherer and Thanos to submit their guesses.

Ending the Game:

The game ends when both the Gatherer and Thanos have submitted their guesses. They are only allowed to submit one guess, so make sure that you have a good guess! Note that they can submit at any point during the game, not just after the time runs out.

The results are based on the selected difficulty of the game (Easy, Medium, or Hard). To win, The Avengers must earn enough points while also preventing Thanos from getting too many points. A single point is awarded when the owner and location of a stone is correctly guessed, allowing for a total of 6 points per team. The point schemes are as follows:

	Gatherer	Thanos
Easy	4 or more	3 or less
Medium	All 6	3 or less
Hard	All 6	1 or less

Once both of the guesses have been submitted, all players will receive the results and the game will be over!

Points to ponder

- **What is a hash? What is a key?**
    - When hashing, certain algorithms allow for the use of shared keys, but some do not. In this program we use md5 and append our key to the data to be sent which allows us to create unique IDs, or hashes, of our messages. Why have we done this? Why do we need a key?
        - A hash is a unique identifier for some data. The data is run through a hashing algorithm that produces a completely unique hash. Any different data run through the same algorithm will produce dramatically different hashes, meaning that even with a small change in the data you will not be able to guess the hash.
        - In this case, a key is an agreed upon data like a password or passphrase that is shared among all authorized entities. This key is added to the data that is put through a hashing algorithm creating a completely different hash. In our case we need a key to allow the sender and receiver to hash data with confidentiality and integrity.

- **What is chaffing? What is winnowing?**
    - Why is this process useful? How is data hidden from untrusted parties?
        - Chaffing is the process of adding extra data to your communications in order to conceal your true data.
        - Winnowing is the process that the receiver goes through that takes away all of the chaff and reveals the true message from the sender.

- **How does chaffing and winnowing use hashes? Who has access to the key?**
    - We know that hashes act as unique identifiers for data, in this case messages.
        - The sender creates a hash based upon the data and key for the correct message(s) and computes a hash for the fake message(s) with the data, key, and extra, random data. The receiver knows that any fake message sent will have a hash that does not match the hash computed on the receiver side because of the extra, random data. Any unauthorized entity will not be able to see which hash matches, because they do not have access to the key that was used to compute the hash originally.

- **Is it necessary to make the fake/chaffed data look legitimate? Why or why not?**
    - In this game we are trying to hide a true message among other fake messages. Why would you not want to use generic or meaningless messages for your fake messages?
        - Yes, sending messages that are incoherent or not related to the message subject will make it that much easier for any unauthorized entity to guess the correct message. The best practice is to send fake messages that look like they could pass as the real messages.

- **Does this process provide confidentiality? Does it provide integrity? Why or why not?**
    - From the CIA model, what attributes apply to chaffing and winnowing? Confidentiality refers to keeping the intended message a secret from any unauthorized entity. Does chaffing and winnowing provide this? Integrity can come in two forms,

data and origin. Data integrity refers to knowing the data was not tampered with by an unauthorized entity, and origin integrity refers to knowing that the person who sent the message is really the person that sent the message. Does chaffing and winnowing provide this?

    - This does provide some level of confidentiality, even without encryption. Anyone who receives the messages can see the correct message, but will not be able to tell it apart from the other, fake messages unless they are the intended receiver.
    - Integrity is also provided for both data and origin. Because a key is used in hashing the data, the receiver knows that when a correct hash is found that the data has not been tampered with. Also, because only authorized entities have access to the key, the receiver knows that any correct hashes found means that you know that the sender is who they say they are.

- **Is this a simple and/or easy solution?**
    - Another GenCyber principle to consider is “Keep it simple”. Does chaffing and winnowing accomplish this? Have we used an efficient process to hide/conceal our data from unauthorized parties?
        - Yes, chaffing and winnowing is a good example of keeping it simple. Typically, to provide confidentiality data is encrypted on the sender side and decrypted on the receiver side. To bypass that entire process, we simply add extra data to the messages in a way that allows only the receiver to tell which data corresponds to the correct messages.

Resources

Chaffing and Winnowing

https://pdfs.semanticscholar.org/aaf3/7e0afa43f5b6168074dae2bc0e695a9d1d1b.pdf

https://link.springer.com/content/pdf/10.1007/978-1-4419-5906-5_184.pdf

https://www.sans.org/reading-room/whitepapers/vpns/review-chaffing-winnowing-876

https://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-5906-5_184

Hashing

http://orbit.dtu.dk/files/5025771/sst_thesis_v1.0.pdf

NodeJS

https://nodejs.org/en/

Socket.io

https://socket.io/