THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

CS203- Lab 13 (*The last one!*)

Encryption & Git

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

Encryption

What is encryption

- Encryption is the process of converting data into a cipher(code) to prevent unauthorized access
- It’s important to protect you or your customers data!
- The basic idea is taking some string or data(input), applying a mathematical formula, and saving the hash(output)
- For instance, “Hello world!” encrypted with AES-256-CBC and the key(passphrase) “computerscience!” looks like this:
  JhGeaSRhZ6ZLZBLsfytFJQ==

Why do we need it?

- **Data Privacy:** Ensures that only authorized parties can access sensitive information
- **Confidentiality:** Protects the content of messages or data from being understood by unauthorized individuals
- **Integrity:** Ensures that data remains unaltered during transmission or storage
- **Authentication:** Verifies the identity of communicating parties to prevent impersonation

Modern Encryption Methods

- **AES (Advanced Encryption Standard):**
  - **Application**: Data transmission, file encryption, and disk encryption
  - **Strengths:** Widely adopted for its speed and security. Commonly used to encrypt sensitive data.
- **RSA (Rivest–Shamir–Adleman):**
  - **Application:** Key exchange, digital signatures, and securing communications over the internet.
  - **Strengths:** Provides a secure method for key exchange and digital signatures.
- **TLS/SSL Encryption:**
  - **Application:** Secure communication over the internet, such as HTTPS.
  - **Importance:** Ensures the confidentiality and integrity of data during transmission.
- **SHA-256 (Secure Hash Algorithm 256-bit):**
  - **Application:** Generating fixed-size hash values from variable-size data.
  - **Strengths:** Resistance to collision attacks, used in blockchain and digital signatures.

Choosing the Right Encryption

- **Security Requirements:** Different scenarios may require different levels of security
- **Key Management:** How keys are generated, distributed, and managed
- **Performance:** Balance between security and computational efficiency

Char[] c = str.tochararray()

Caesar's Cipher

- One of the earliest known substitution ciphers, it is attributed to Julius Caesar
- How it works: Shift each character by a fixed number of positions up/down the alphabet
- This is what we will use for the assignment this week!
- Example: If the shift is 3 'A' becomes 'D', 'B' becomes 'E' etc.

Caesar’s Cipher Cont.

Pros:

- Simple and easy to understand
- Quick encryption process

Cons:

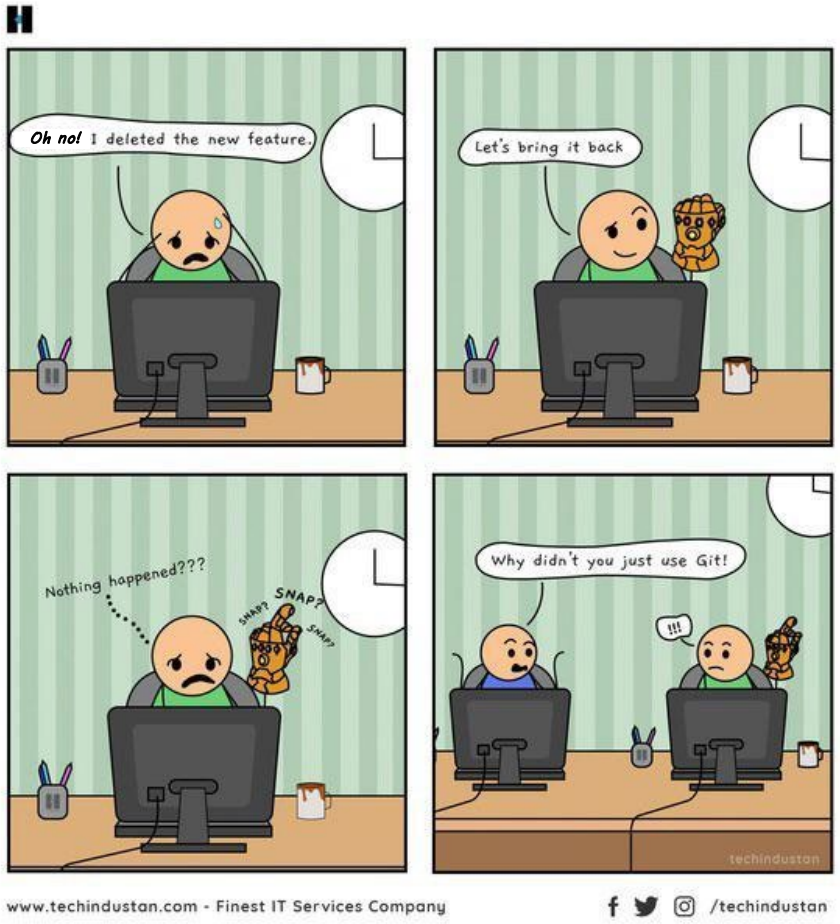
- Vulnerable to brute force attacks due to a limited key space
- Lacks security for modern applications

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM.

Git & GitHub

GitHub & Git

- GitHub is a web-based platform that allows users to collaborate, share, and manage code
- GitHub uses **Git,** a version control system(VCS), to track and manage changes to files.



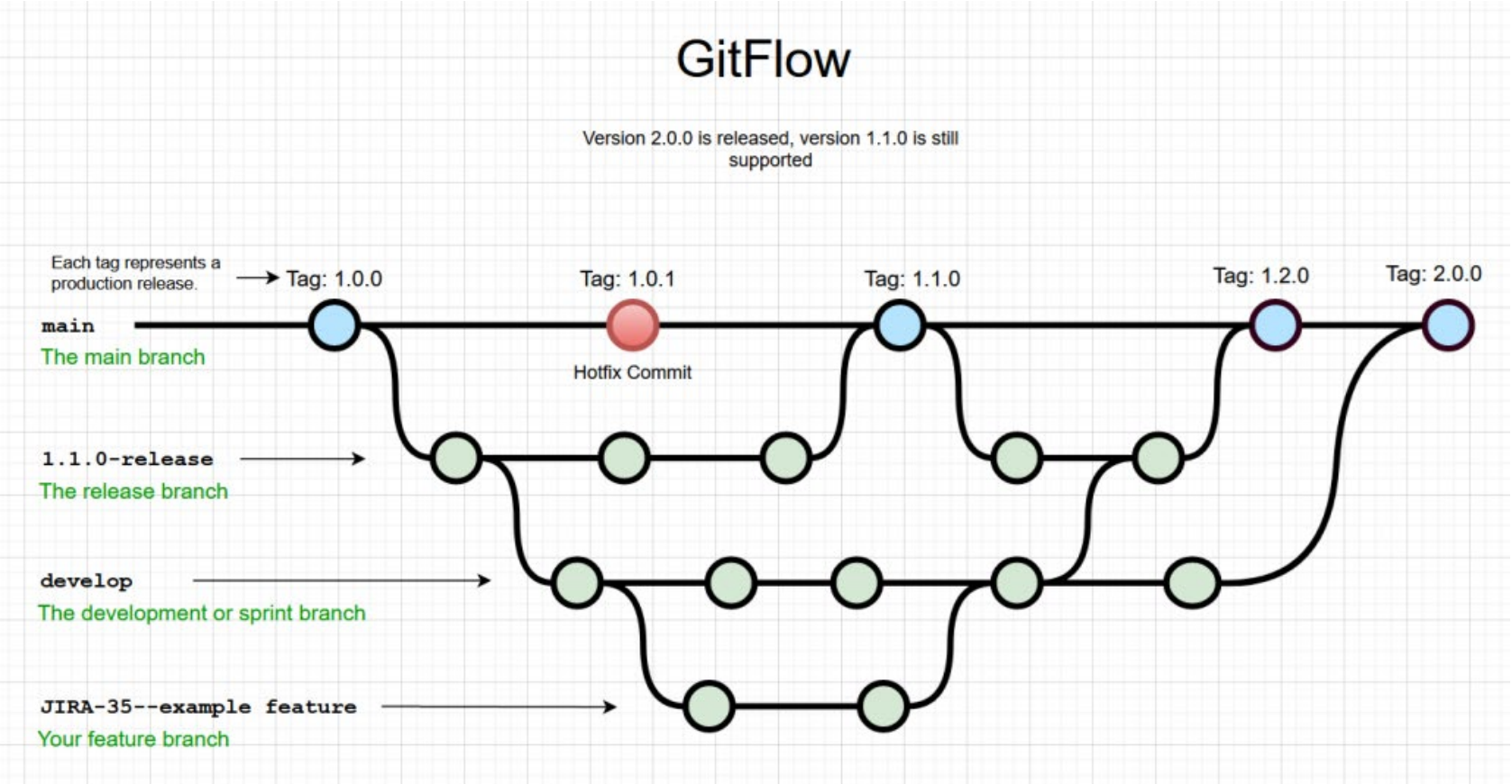
Why use GitHub?

- **Version Control:** Allows you to keep track of every change made to code including who made those changes when working on projects together.
- **Collaboration:** Allows you to work on projects with teams without needed to physically be in the same space, and allows you to merge code created and edited by team members.
- **Project Management:** Keep work organized and manage large, complex projects.
- **Portfolio Building:** Showcase your projects to show for future opportunities
  - For any class assignments, make sure to ask professors if it is okay to include your assignments in public repositories!

GitHub Key Points

- **Repositories (Repos):** A place to store and manage your code and project files
- **Branches:** Create alternate versions of the source code with a shared origin
- **Merging:** Merge 2 branches together, maintaining the changes made to both branches
- **Commits:** Snapshots of changes to files in a repository, including a short description of what was changed.

GitHub Branching and Merging Diagram



Three States in Git

- **Modified:** file has been saved, but changes had not been committed (git has not been updated)
- **Staged:** the file is marked to be in the next commit
  - git add <filename>
- **Committed:** changes have been recorded
  - git commit

Working with Remote Repos

- "git pull" - saves the current state of the remote repo to the local repo as well as the workspace
- "git push" - saves the current version of your local repo to the remote repo
- "git fetch" - similar to git pull but only saves the remote files to the local repo
- "git clone" - initializes a local repo and then "git pull" from the remote repo (You will be cloning in today's lab)

Other things to know

- “git diff” - see what changes you’ve made before committing
- “git commit -a” - skips the staging area, adds and commits all files under the current directory
- “git rm <filename>” - removes a file from the project, this is done before a commit (when you are adding files to the staging area)
- “git mv file_from file_to” - renames a file
- “git checkout <branch>” - change branch, create new branch, or revert to a previous version. This has a lot of uses so make sure to read the docs before trying to use it.
- “git reset” - unstage all or a specific file
- “git restore” - restore all modified files to previous commit, BE CAREFUL you can lose data if used incorrectly
- “-h” - add this flag to almost any command to see how to use it

Getting started

- If you have not already, create a GitHub account.
- Download git onto your computer:
  - https://git-scm.com/downloads
  - After downloading, open your terminal and type “git” to check that it has been downloaded properly. If it has downloaded properly, you will see a help menu in your terminal.

Cloning A Repository

- Before cloning the repository from GitHub, Cd into your chosen directory :

```
PS C:\Users\SamMi> cd C:\Users\SamMi\OneDrive\Desktop\CS203
```

Cloning a Repository

- After you cd into the correct directory, visit the repository to be cloned and copy the URL
  - For this assignment, you should be copying:

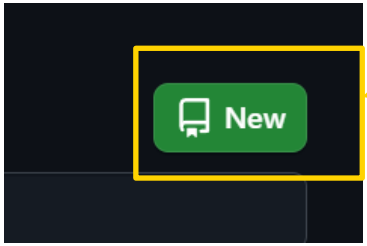
  https://github.com/SamMPhillips/Fa24_CS203_Lab13
- Return to your terminal and type:

  git clone https://github.com/SamMPhillips/Fa24_CS203_Lab13
- Check that the repository was cloned correctly by seeing if it is now in your directory.

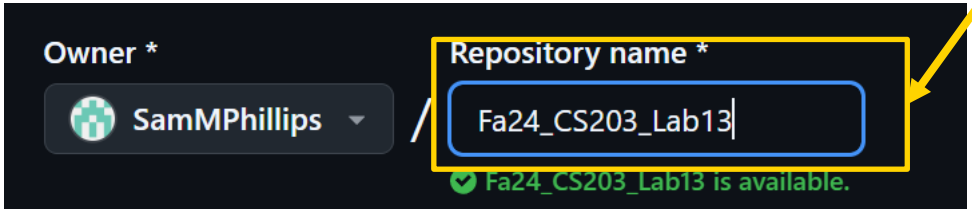
After Cloning

- You have now cloned the repository onto your local machine. You should be able to open the cloned repository in your chosen IDE to complete the assignment.
- Once completed, you will push your changes to a private repository that you will create.

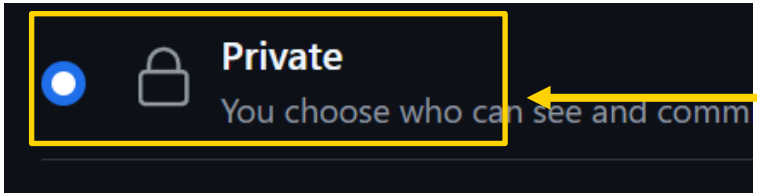
Create a New Repository



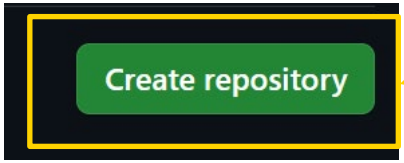
1. Click “New” in the upper left of your GitHub dashboard



2. Name the repository “Fa24_CS203_Lab13”



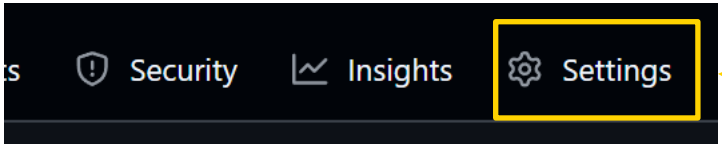
3. IMPORTANT: Set to Private



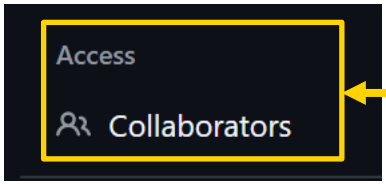
4. Click “Create repository”

Add Collaborators

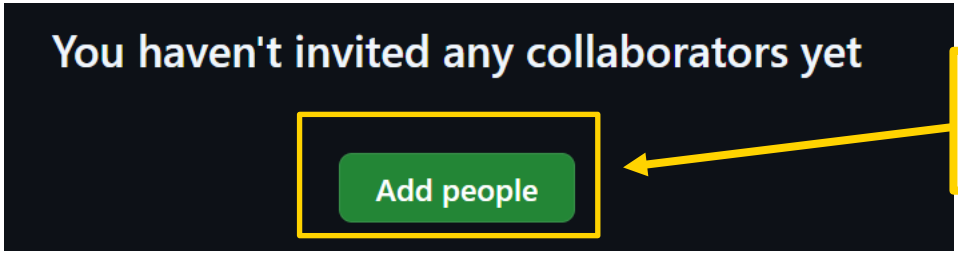
You *must* add the TAs as collaborators in your repository in order for your lab to be graded.



1. From your repository, go to “Settings” in the toolbar toward the top of the page



2. Click “Collaborators” in the toolbar on the left.



3. Scroll down and select “Add people”

4. Search for and add all 3 TA’s using the usernames:

SamMPhillips
Abbie-m
MichaelGathara

Pushing To Your Private Repository

- From your terminal, make sure that you are in the correct directory (the directory that was cloned)

```
PS C:\Users\SamMi\OneDrive\Desktop\CS203\Fa24_CS203_Lab13>
```

- Remove the old remote (the connection to the original repository) by typing:

"git remote remove origin"

- Copy the URL to your private repository and return to the terminal. Type:

"git remote add origin <your url>"

- To check that the remote was update properly, you can type:

"git remote –v"

Pushing To Your Private Repository

- After working on the assignment, you will need to push your changes to GitHub.

1. Return to the terminal and cd into the same directory as before. Type:

   "git add . "

   This will stage all changes for your next commit.

2. Type: "git commit –m "Your commit message" " Your commit message should be something meaningful and relevant. For example:

```
> git commit -m "Completed encryption.java"
```

3. Finally, you need to push your changes to the remote repository. Type:

   "git push"

After pushing, make sure to look at your repository in GitHub to make sure your changes were pushed correctly.

More Git/Github

- A free Udacity course: https://www.udacity.com/course/version-control-with-git--ud123
- UAB ACM has a Git/Github workshop and a link to the ACM GitHub: https://uabacm.org/
- UABACM first-contribution (a great guided way to make your first contribution on GitHub): https://uabacm.org/assignment

Lab 13 Submissions

- Your Canvas submission for Lab 13 will consist of only the link to your private repository (which must have all 3 TAs as collaborators)
- Inside of this repository, you should have:
  - your completed Encrypter.java
  - EncryptionTester.java
  - encryptMe.txt
  - Message.txt
- Remember, the only allowed resources are your lab/lecture notes, Zybooks or official Java documentation. If you use either Zybooks or official documentation, you must include a citation.

Good luck in your future courses!