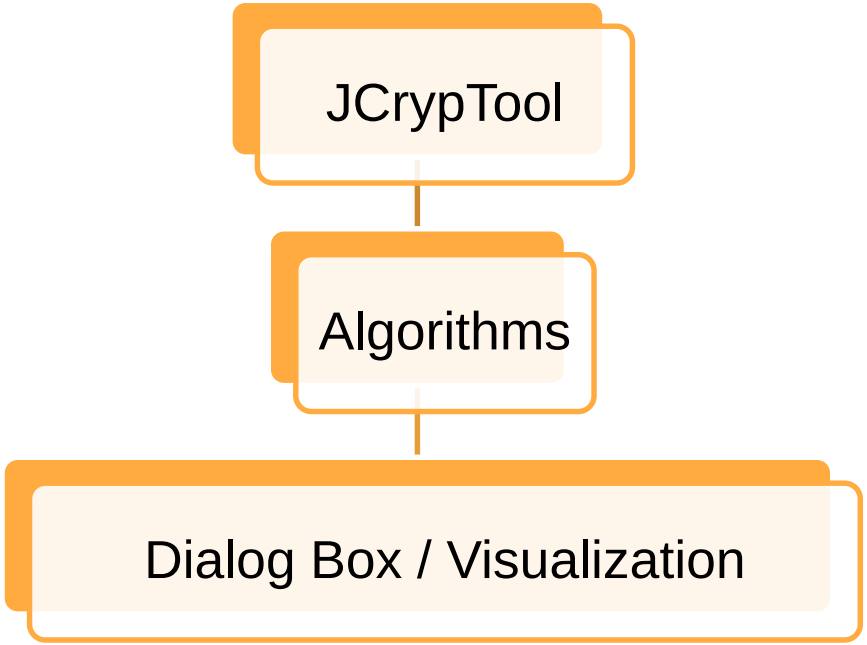
Cryptology

Dec 12 2016

GitHub

Trello

System Diagram



Product Highlights

1. Provide users with more encryption algorithms, which are candidates for SHA3 in the last round, to choose when using JCrypTool.
2. Design a better dialog box for users to choose different algorithms and read the output more easily.
3. Visualized Blake algorithm.

Demonstration

Problem with Algorithms

The algorithm doesn’t output the correct answer.

Test case(for Groestl):

1. Abcd: D8CD6BE396A8F029BC46E48367D3D84150776C10B7A6AFFEDB19E8D0D175A708
2. Aefg: D8CD6BE396A8F029BC46E48367D3D84150776C10B7A6AFFEDB19E8D0D175A708

Problem 1: Unless we change the first character of the input, the output will be the same.

Problem 2: Unless we change the length of the input, the output will be the same for most cases.

Problem solved:

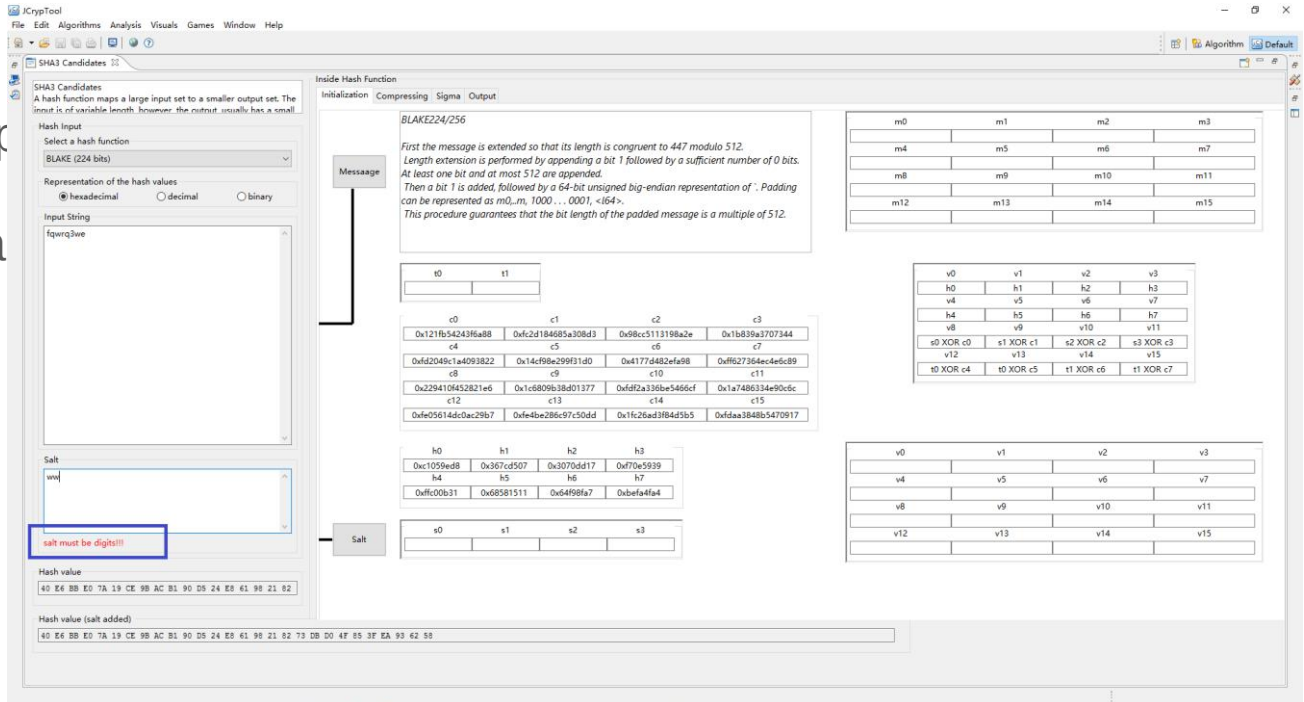
1. Abcd: 4A639E2F274A8B6A4D3AC957456F8FAE8DF80E2AEA89C19276BDACA5586B5F99
2. Aefg: EDFF182DB0D66874D155CA79CD37FF3F625C822B37AE66E3C1903D50AB5C6C3A

Problem with Visualization

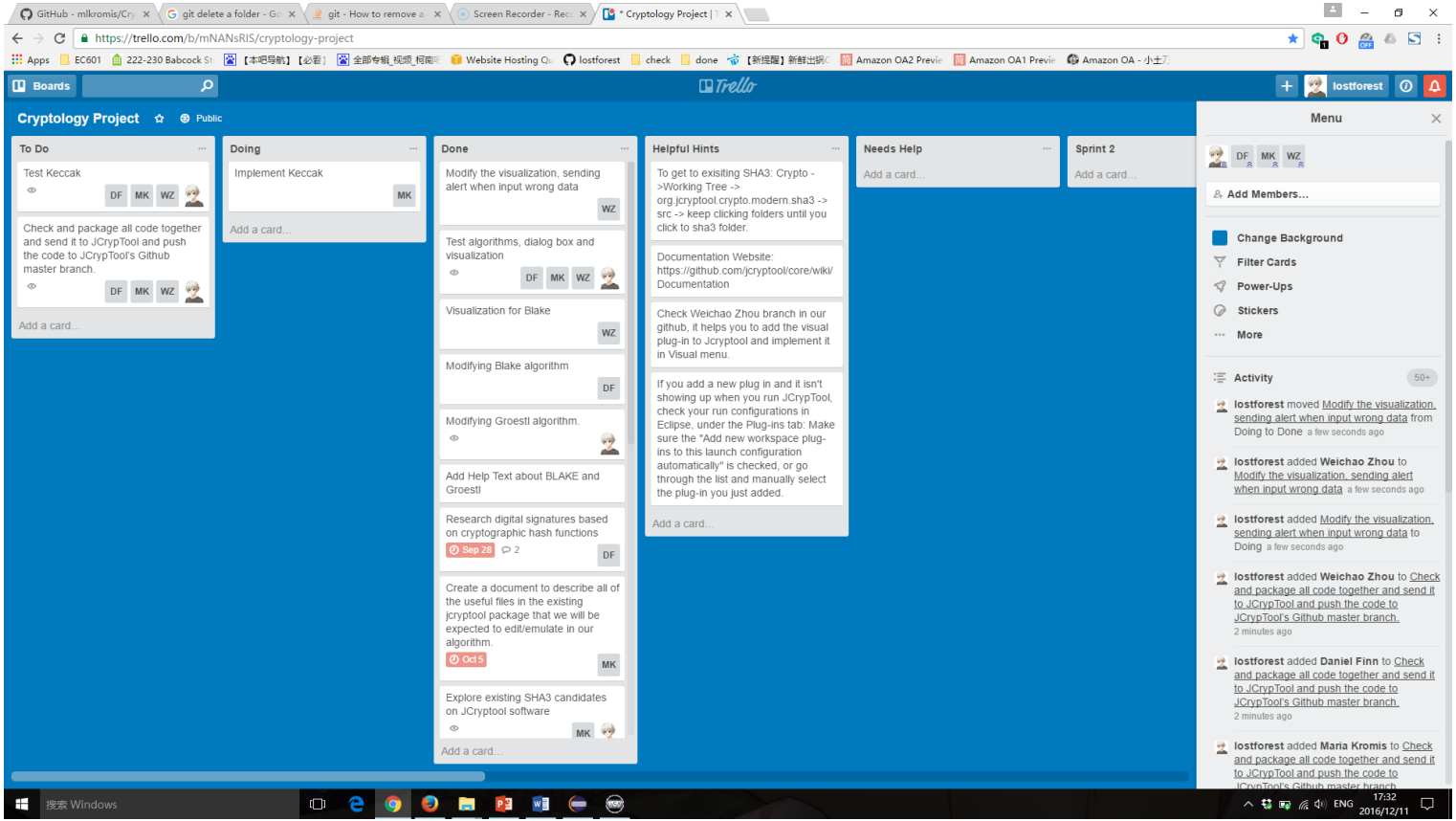
Visualization cannot jump automatically when clicking at the button.

Solved: Inp

Also add a



Trello


Cryptology Project
Public
To Do
Test Keccak
Check and package all code together and send it to JCrypTool and push the code to JCrypTool's Github master branch.
Add a card...
Doing
Implement Keccak
Add a card...
Done
Modify the visualization, sending alert when input wrong data
Test algorithms, dialog box and visualization
Visualization for Blake
Modifying Blake algorithm
Modifying Groestl algorithm.
Add Help Text about BLAKE and Groestl
Research digital signatures based on cryptographic hash functions
Create a document to describe all of the useful files in the existing jcryptool package that we will be expected to edit/emulate in our algorithm.
Explore existing SHA3 candidates on JCryptool software
Add a card...
Helpful Hints
To get to exisiting SHA3: Crypto -->Working Tree -> org.jcryptool.crypto.modern.sha3 -> src -> keep clicking folders until you click to sha3 folder.
Documentation Website: https://github.com/jcryptool/core/wiki/Documentation
Check Weichao Zhou branch in our github, it helps you to add the visual plug-in to Jcryptool and implement it in Visual menu.
If you add a new plug in and it isn't showing up when you run JCrypTool, check your run configurations in Eclipse, under the Plug-ins tab: Make sure the "Add new workspace plug-ins to this launch configuration automatically" is checked, or go through the list and manually select the plug-in you just added.
Add a card...
Needs Help
Add a card...
Sprint 2
Add a card...
Menu
Add Members...
Change Background
Filter Cards
Power-Ups
Stickers
More
Activity
lostforest moved Modify the visualization, sending alert when input wrong data from Doing to Done a few seconds ago
lostforest added Weichao Zhou to Modify the visualization, sending alert when input wrong data a few seconds ago
lostforest added Modify the visualization, sending alert when input wrong data to Doing a few seconds ago
lostforest added Weichao Zhou to Check and package all code together and send it to JCrypTool and push the code to JCrypTool's Github master branch. 2 minutes ago
lostforest added Daniel Finn to Check and package all code together and send it to JCrypTool and push the code to JCrypTool's Github master branch. 2 minutes ago
lostforest added Maria Kromis to Check and package all code together and send it to JCrypTool and push the code to

Test Case 1

1. Different input with same length with 256 hash bit length.

Input1: ABCDefgh

Input2: ABCDijkl

Groestl:

Output1:
AA7FF5E8167C47B451C75852A2CF77362708F9D9C15CE349E26D81FB8266B085

Output2:

CE4D4AE59B17F7441F638E519D8A24BE737CDAF9E6A944BF7BF4A609E184A7BF

Blake:

Output1:
E6A2B048BAB77FC6798346A34FE91635507DFABF21CF904D63DD12933675C6DB

Output2:
0FFFB62363102F94C5FA4791F565CA93FAD8BE51EC2067E16F2BED91310E8C7B

Test Case 2

2. Different input with only ONE character different

Input1: uihkjllihjkl1oiuhjnkml7liu

Input2: uihkjllihjkd1oiuhjnkml7liu

Groestl:

Output1:
EE2BE9B9D85A5F72C11AA258742E1B7E8742C1395966B4CF9D5F66F7780AFA79

Output2:
99FDA91895657FFE1D19F4821945D5750CB9C8F19065F2E413D2EF7A1B904AD4

Blake:

Output1:
C054E0ECCC6AD3DE0E539B9F04904C07B6B75ABF528E7A2644F790C4D67B5447

Output2:

42632FE4B23C9C8DCB2C40503D16674EC315DA0BD5EFB1180A985D86D282DF79

Top three things that worked well

1. Algorithm implementation for SHA3 candidates.
2. Design the dialog box.
3. Visualized the Blake algorithm.

Top three things that we have learned

1. Learn how to develop a plugin based on a project.
2. Java knowledge, especially the basic Java data structure.
3. Learn how to use Github, Trello and Slack.

Future plan

1. Implement SHA3 candidates left.
2. Build more visualization for SHA3 candidates.
3. Implement the visualization by import the package other than have a copy for the visualization
4. Check and package all code together and send it to JCrypTool and push the code to JCrypTool’s Github master branch.

Thank You