



Enhancing Cryptography Education Using Collaborative Visual Programming

Sherif Abdelhamid, Sarah Patterson, Blain Patterson

Virginia Military Institute

abdelhamidse@vmi.edu, pattersonse@vmi.edu, pattersonba@vmi.edu

Overview

- Cryptography is the science of securing sensitive information and ensuring that only the intended recipients can access and process the encrypted data.
- Internet shopping, online payments, and social networking websites have become increasingly popular with the advancement of the internet.
- Protection of data has become increasingly more important than before.

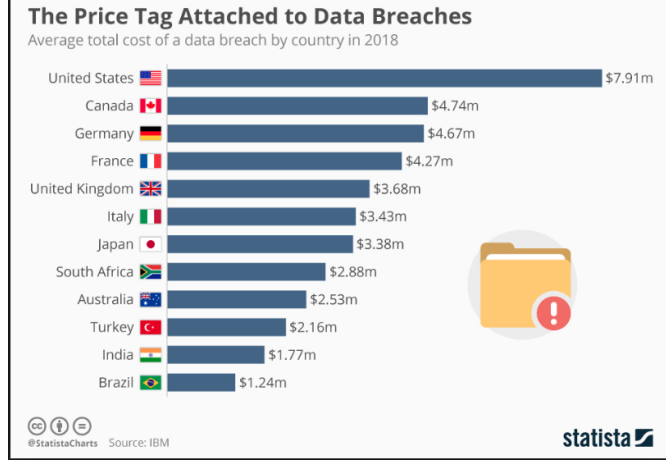


Problem

- Hackers are getting more skilled than before to exploit existing vulnerabilities and attack these websites.
- According to the Identity Theft Resource Center's (ITRC) data breach analysis, there were **1,291 data breaches through September 2021**. This number indicates a **17% increase** in data breaches in comparison to breaches in 2020, which was 1,108.



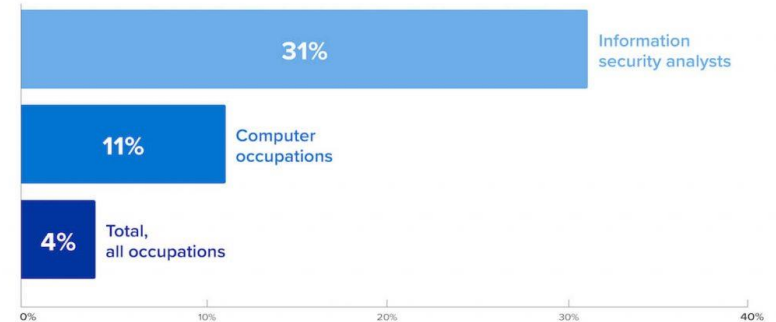
Source: Cybersecurity Ventures, 2020



Solution

- It is important to introduce the science of cryptography to future generations, at a younger age, in straightforward and more engaging ways.

Projected Employment Growth for Information Security Analysts, 2019-2029





Challenges

- Students need to acquire multidisciplinary skills in mathematics, information theory, and software programming to achieve this goal.
- In addition, students have to receive formal training in software engineering, testing and big data analysis.
- learners need to think in an organized and procedural way about the problems they are attempting to solve .

These requirements might create a barrier for students and scientists to develop and implement novel encryption algorithms or enhance existing ones.

Methodology

Explore Related Work

Qualitative manual literature review approach
(Provides in-depth detailed view on specific relevant research works)

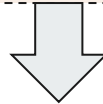
Subjective research aims to manually inspecting sources to identify key patterns that is not necessarily numbers

Relevant new keywords

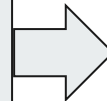
Overview directions and insights

Quantitative automated bibliographic analysis
(Provide global overview and main research areas and trends)

Objective research aims to identify relationship between variables and interested in numbers and trends



vizLab tool design and Implementation



Testing and integration within courses



Literature Review

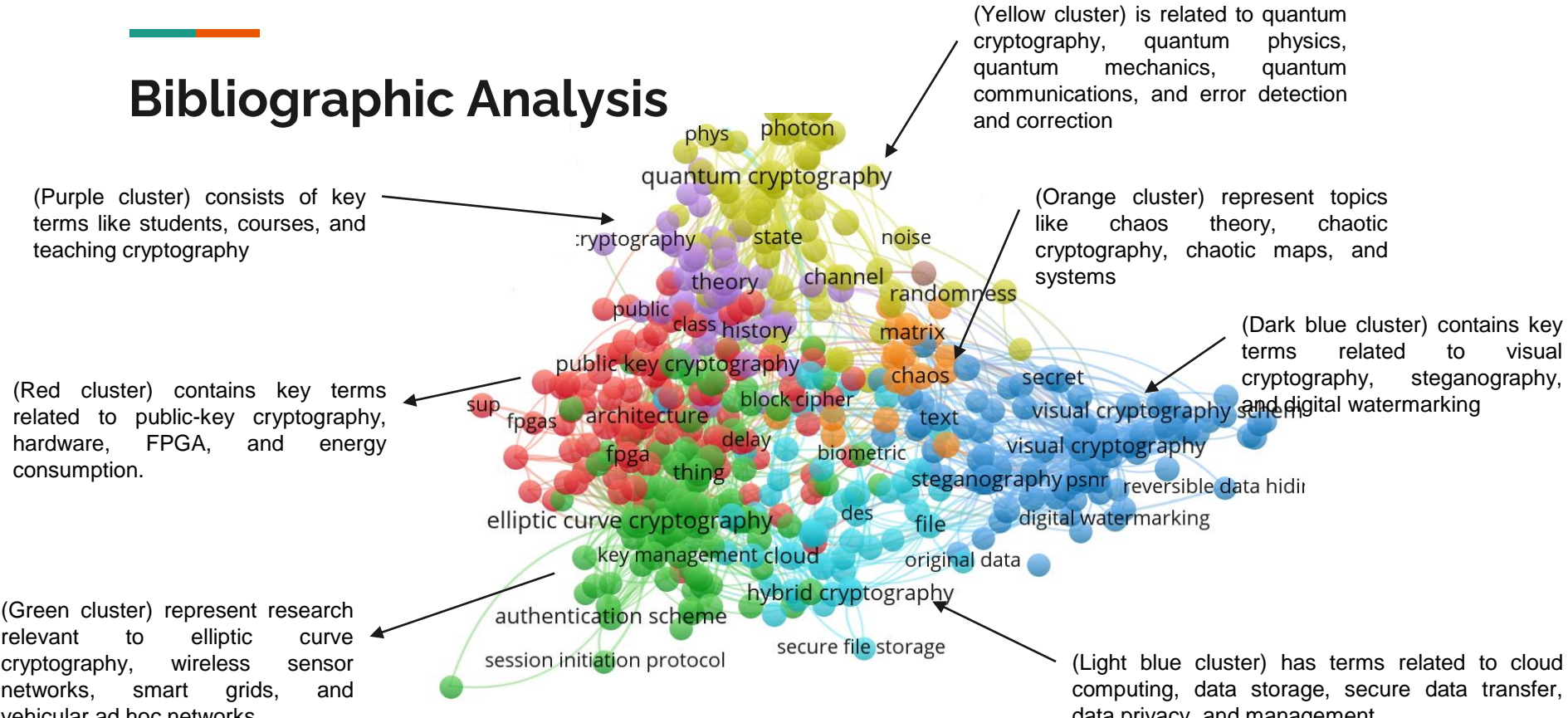
- Integrate security concepts into existing courses
- Create separate security coursework or programs and concentrations
- Implementing active learning techniques
- Use of real-world scenarios
- Algorithm illustration with animations
- Use of interactive visualizations
- Building intelligent tutoring systems



Bibliographic Analysis

- Conducted bibliometric analysis to identify research themes and explore the academic landscapes related to cryptography education.
- Collected ~10k publications from **Crossref database**, a not-for-profit association of publishers, including both commercial and not-for-profit organizations.
- From the retrieved publications, we **key extracted terms** (with minimum occurrences of **ten**) from the abstract and title, resulting in **468 keywords**.
- Based on the **co-occurrences of the terms** within the same title or abstract, we constructed the term **co-occurrence network**, which consists of **14937 edges/links** and **468 nodes**.
- Each node represents a term, and each edge represents a co-occurrence relationship.

Bibliographic Analysis

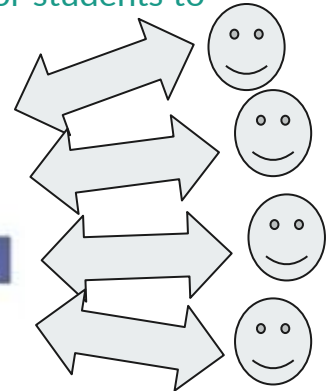
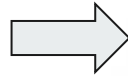


Outcome

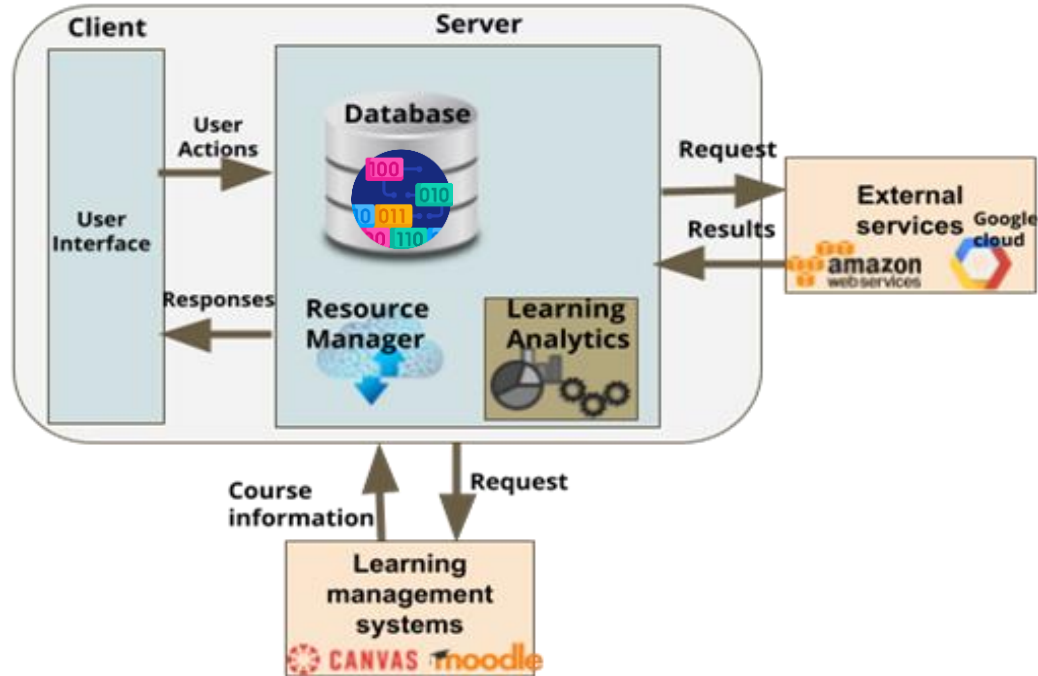
Few studies within cryptography education have considered the programming challenges during cryptography algorithm implementation.

Most of the studies focus on providing means to simplify the understanding of complex math concepts and algorithms but with little attention to the software development aspect.

As a response, in this work, we implemented vizLab to provide an effective way for students to implement novel encryption algorithms or enhance existing ones.



System Overview



System Overview

The screenshot displays a block-based programming environment with a sidebar on the left, a central workspace, and a right-hand panel. The sidebar lists categories: Logic, Loops, Math, Text, Lists, Colour, Variables, Cryptography, and Functions. The central workspace contains a script with the following blocks:

- set alphabet to make list from text " a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,..."
- set message to " meeting "
- set shift to 1
- set encryption to ""
- for each item i in list message do
 - set pos to in list alphabet find first occurrence of item i
 - to encryption append text in list alphabet get # remainder of
- print encryption

Below the workspace is a console area with a 'Run' button, 'Save Blocks', and a 'Project name' field. The console shows the following Python code:

```
alphabet = None
message = None
shift = None
encryption = None
i = None
pos = None

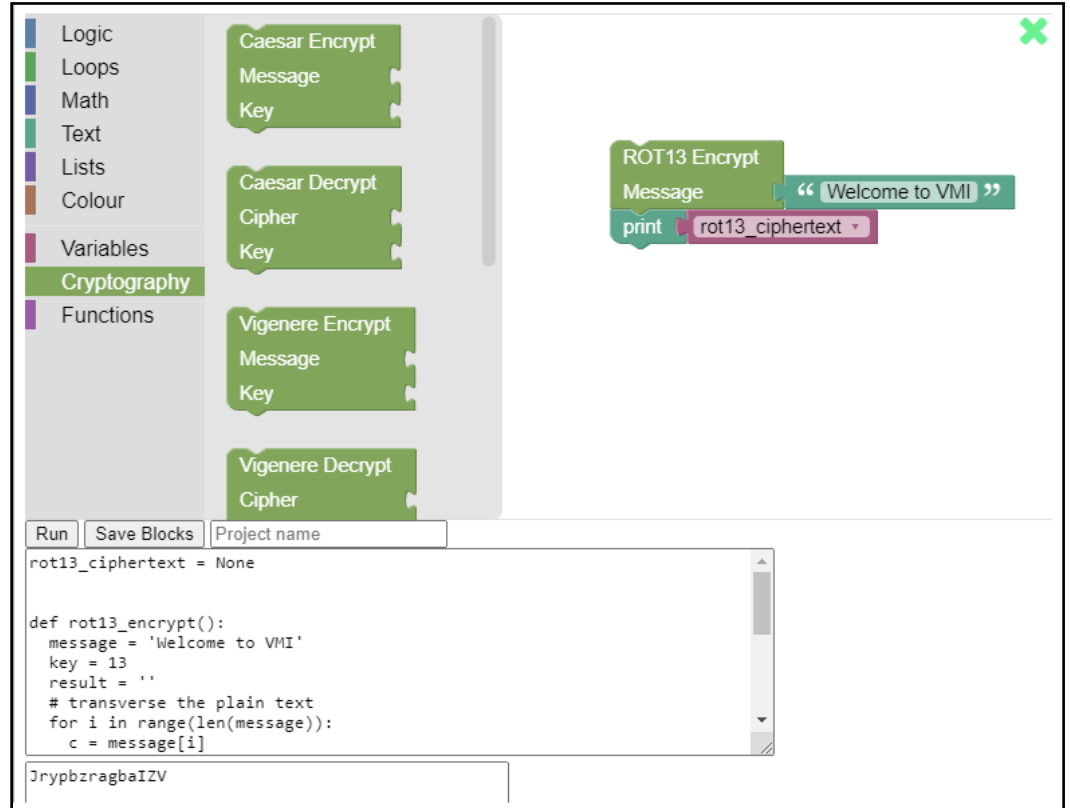
def first_index(my_list, elem):
    try: index = my_list.index(elem) + 1
    except: index = 0

liddshmf
```

The right-hand panel is titled 'My Projects' and lists several projects: loop demo1, Caesar, caesar_cipher, and Rot13_demo. Below this is a 'Shared Projects' section with 'loop demo1'.

System Overview

vizLab provides a ready-to-use library of cryptography blocks. For example, students here use the ROT13 block to encrypt an input message.



The screenshot displays the vizLab interface, which includes a block palette on the left and a workspace on the right. The block palette is organized into categories: Logic, Loops, Math, Text, Lists, Colour, Variables, **Cryptography**, and Functions. Under the Cryptography category, several blocks are visible: Caesar Encrypt, Caesar Decrypt, Vigenere Encrypt, and Vigenere Decrypt. Each of these blocks has input fields for 'Message' and 'Key'. The workspace shows a 'ROT13 Encrypt' block with a 'Message' input field containing the text "Welcome to VMI". This block is connected to a 'print' block, which has a dropdown menu set to 'rot13_ciphertext'. Below the workspace, there is a control bar with 'Run', 'Save Blocks', and 'Project name' buttons. A code editor window is open, showing the following Python code:

```
rot13_ciphertext = None

def rot13_encrypt():
    message = 'Welcome to VMI'
    key = 13
    result = ''
    # transverse the plain text
    for i in range(len(message)):
        c = message[i]
```

The output of the code is displayed as 'JrypbzragbaIZV'.



Conclusion

In this research work, we presented **vizLab**, a web-based programming learning tool that helps students avoid the programming challenges during cryptography algorithm implementation. vizLab promotes collaborative learning and engagement.

Our future work will involve integrating vizLab into an undergraduate course(s).

In addition, we will collect and analyze data about students' conceptual understanding and their ability to construct cryptography algorithms using surveys and more class observations.



Questions Please...