NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Final Year Project

Project Number: SCSE22-0861

Game-Based Cryptographic Algorithms Learning

Supervisor:

Dr Smitha Kavallur Pisharath Gopi

Written by:

Malcolm Tan Wei Zhang

U2022160E

Abstract

In an ever-evolving landscape of digital communication, secure data transmission has become paramount. The art of encoding and decoding information plays a pivotal role in ensuring data confidentiality. This project introduces an innovative approach to cryptography education through a game-based application that explores classic encryption methods. This report serves as a comprehensive guide to the project, detailing software choices, components, and their functionality. It outlines the project's objectives, scope, and the broader context of cryptography education.

This project integrates modern user interface design principles to ensure a seamless and intuitive user experience while providing educational opportunities that sparks curiosity and engagement.

Acknowledgements

I extend my heartfelt gratitude to Dr. Smitha K G, my project supervisor, whose patience, guidance, and encouragement have been instrumental throughout the journey of this project.

Table of Contents

List of Figures

1. Introduction

1.1 Project Background

Technology has become increasingly reliant on online services due to the rapid growth of digitalization. Due to the Covid-19 epidemic, there has been a huge increase in new digital products that helps businesses overcome their contactless situation [1]. As online services continue to thrive and expand, cryptography algorithms have become vital in data protection and privacy. Some applications of modern cryptography in everyday life includes digital currency, e-commerce, social media applications, emails, passwords, and military operations.

Without proper protection, our personal information can be vulnerable to cyber-attacks and data breaches. These breaches can result in significant losses, including financial losses, identity theft, and reputational damage[2]. With how relevant cryptography knowledge is to developers, there is a lack of resources and tools to further educate themselves on software and computer security [3].

Cryptography is the continuous practice and evolution of techniques used for ensuring integrity, exchanging secret keys, authenticating users, and many more [4]. Each technique uses an encryption algorithm which convert plain text into unintelligible nonsense text, also known as ciphertext. Without such measures, the secure transmission and storage of sensitive information would not be possible.

The impact of cryptography is significant, and potential beneficiaries include individuals, businesses, and governments. Individuals can benefit from the increased security and privacy of their personal information, while businesses can protect their confidential data and trade secrets from competitors and cyber criminals. Governments can use cryptography to protect national security interests and sensitive information[5].

1.2 Project Importance

Traditional methods of teaching cryptographic algorithms have been limited in their effectiveness due to several factors. One of the main limitations of is the complexity of the algorithms. The abstract nature of the algorithms can make it difficult for students to understand their practical application[6]. Another limitation of traditional methods is the lack of interactivity and engagement. Traditional teaching methods, such as lectures and textbooks, can be dry and uninteresting, making it challenging to keep students engaged and motivated. This lack of engagement can lead to students losing interest in the subject matter and failing to grasp the core concepts[7].

The relevance of game-based learning in teaching cryptographic algorithms has been recognized as a potential solution to these limitations. Studies have shown that game-based learning can improve students’ problem-solving skills, increase motivation and engagement, and improve their retention of information[8]. Therefore, the niche that a game-based learning approach fills is the need for a more engaging and effective way of teaching cryptographic algorithms[9]. By using gamification to teach these concepts, we can overcome the limitations of traditional methods and provide students with a more interactive and engaging learning experience.

1.3 Project Objective and Scope

This project aims to provide developers an opportunity to learn cryptography algorithms through games and tutorials. The target audience for this project are beginners in software security who are not well-versed in the field of Cyber Security, Information Technology or Computer Science. To achieve this objective, the project will accomplish four tasks:

1. Design and implement a user-friendly and responsive web application with interactive cryptography interface.
2. Start with game-based tutorials that teaches the fundamentals of simple cryptography.
3. Gamify encryption algorithms using puzzles with increasing levels of difficulty.
4. Assess the effectiveness of the web application.

2. Literature Review

In this section, we delve and understand more about specific cryptography algorithms, a fundamental aspect of information security. Historically, the evolution of cryptography has branched into two significant segments: symmetric and asymmetric (or public key) cryptography.

2.1 Symmetric Key Cryptography vs Public Key Cryptography

Cryptography involves the application of mathematical principles and algorithms to encode messages, making them unintelligible to unauthorized recipients. A plain message, called Plaintext, would enter Encryption to form Ciphertext, which in turn, would enter Decryption to form Plaintext.

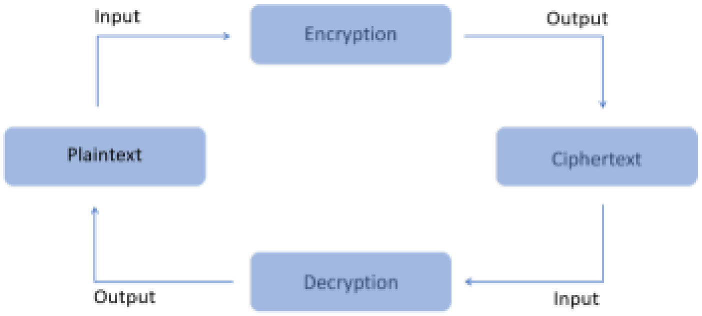


Figure 1: Cryptography Example

With symmetric key cryptography, a singular key is employed for both the encryption and decryption process. This provides an inherent advantage for symmetric keys as it has straightforward mathematical operations that improves its speed and efficiency. However, it grapples with a fundamental challenge known as key distribution problem. Since both sender and receiver require access to the same secret key, securely transmitting this key is extremely difficult.

Fortunately, a solution was found with public key cryptography, also known as asymmetric cryptography. It operates on a pair of keys: a public key, which can be disseminated openly, and a private key, which remains confidential to its respective owner. The public key is used to encrypt messages, while the private key is used to decrypt them. The advantages of public key cryptography is being the solution to symmetric key distribution issue. Since the public key can be openly shared without compromising security, there is no need for a secure transmission channel for key exchange. However, this method requires higher computational power and thereby makes it slower and less efficient than symmetric key cryptography.

2.2 Symmetric Key Cryptography

In this section, we will further delve into more classical symmetric key cryptography like Caesar, Beaufort, Vigenère, and Columnar Transposition. These historical ciphers are not secure by today’s standards, but they are excellent for teaching the basics of encryption and introducing the concept of symmetric cryptography.

2.2.1 Caesar Cipher

The Caesar Cipher exemplifies one of the earliest instances of cryptographic techniques. This cipher operates on letter substitution, which each letter in the plaintext is shifted a fixed number of positions down the alphabets. This shift value, called ‘key’, would determine the degree of encryption.

Suppose we would like to encrypt ‘HELLO’ using Caesar Cipher with a key of 3. In this case, each letter in the plaintext will be shifted 3 positions down the alphabets.

Figure 2: Caesar Cipher Example

In a mathematical perspective,

Let P be the position of a letter in the alphabet (A-Z, 0-25) respectively.

C be the position of the corresponding encrypted letter.

K be the number of positions to shift, known as the key.

Therefore, the Caesar Cipher encryption would be determined by $C = (P + K) \% 26$.

In modern cryptography, the Caesar Cipher is considered quite weak and is mainly used for educational purposes towards more complex algorithms, which is perfect for the project.

2.2.2 Beaufort Cipher

The Beaufort Cipher shares similarity with the Caesar Cipher but approaches encryption in a slightly different matter. While Caesar Cipher shifts their letters forward, the Beaufort Cipher shifts them backwards based on the key.

Suppose we want to encrypt ‘HELLO’ using Beaufort Cipher with the key ‘KEY’. In this case, each letter of the plaintext would be shifted back by the numerical value of the letters in the key. Since our key is shorter than our plaintext, we would extent it by repeating the letters, therefore ‘KEY’ in this example, would become ‘KEYKE’.

Figure 3: Beaufort Cipher Example

In a mathematical perspective,

Let $Ppos$ be the position of a letter in the alphabet (A-Z, 0-25) corresponding to our plaintext.

$Kpos$ be the position of a letter in the alphabet (A-Z, 0-25) corresponding to our key.

Therefore, calculate the ciphertext letter C using the shifted position and the alphabets:

$$C = alphabet[\ (Kpos - Ppos)\ \%\ 26]$$

While this method introduces more variation into the encryption, it remains easily exploitable and vulnerable to brute-force decryption or frequency of pattern recognition to deduce the encrypted message.

2.2.3 Vigenère Cipher

The Vigenère Cipher is an evolution of the Beaufort Cipher due to its approach of shifting individual letters of the plaintext using each letter of the key. However, the value of the shift would be based on the key, and the plaintext would be shifted towards the right. For instance, if the keyword is ‘KEY’, and the first plaintext letter is ‘H’, the letter would be shifted 10 to the right.

Figure 4: Vigenère Cipher Example

In a mathematical perspective,

Let P be the position of a letter in the alphabet (A-Z, 0-25) corresponding to our plaintext.

K be the position of a letter in the alphabet (A-Z, 0-25) corresponding to our key.

C be the ciphertext letter encrypted using P and K.

Therefore, the Vigenère Cipher encryption would be determined by $C = (P + K) \% 26$. This would be done for each letter individually.

In conclusion, the Vigenère Cipher can be seen as a combination of both Caesar Cipher and Beaufort Cipher in their method of encryption. This is a significant step forward for classical cryptography by incorporating dynamic keyword-driven shifting of letters. While it overcomes many of the weaknesses of the previous two ciphers, it still requires careful key management and consideration of potential vulnerabilities.

2.2.4 Columnar Transposition Cipher

The Columnar Transposition Cipher is a departure from the previously mentioned substitution ciphers, focusing on permutation and transposition. The plaintext is arranged in a grid, and letters are reordered based on a keyword, creating a complex and irregular pattern.

Figure 5: Columnar Transposition Cipher Example

The security of Columnar Transposition Cipher is improved due to the arrangement of letters in the grid and the keyword-based reordering of columns to create a complex and irregular pattern. However, a short or easily guessable keyword will still weaken the security of the cipher. While simpler than some other ciphers, it introduces users to the concept of transposition in cryptography.

2.3 Public Key Cryptography

In this section, we will further discuss more into how public key cryptography employs their private and public keys to enhance security. Each of the algorithms below has its own unique way of fortifying the infrastructure of public key cryptography, ensuring confidentiality, authenticity and integrity.

2.3.1 Diffie-Hellman Algorithm

The Diffie-Hellman algorithm was created to solve the secure key exchange dilemma. The process starts with two participants, User A and User B, agreeing to a prime number and a base value. Furthermore, each participant would choose a private number and a public number for themselves. Upon exchanging the public number to each other, both participants would use their private number to independently derive a shared secret. This shared secret can be used for encryption in subsequent communications. The figure below shows the process described.

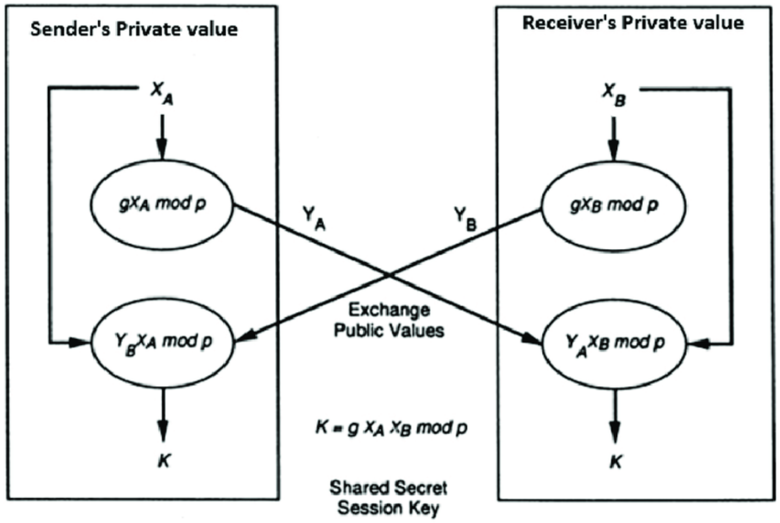


Figure 6: Diffie-Hellman Algorithm Diagram [10]

This algorithm is well-versed against security breaches due to its logarithmic key exchange system. However, this cause the algorithm to be computationally expensive. As this algorithm is a fundamental concept in public key exchange, this is a great topic to teach to beginners in cryptography.

2.4 User Interface

The user interface (UI) of a webpage plays a pivotal role in determining user experience and engagement. It serves to bridge the user and the education functionalities of the project, to further improve the learning experience. In this section, we delve into the UI design principles, along with specific considerations for a navigational bar and side bar design.

2.4.1 UI Design principles

Effective UI design principles are crucial to creating a seamless and user-friendly experience. The aim of good UI design is to create an environment that removes potential stumbling and confusion from the user. For this project, we will be following 6 golden rules of UI Design by Ben Shneiderman [11].

1) **Consistent user interface.**

   Consistency is a fundamental aspect in UI Design. This approach involves employing similar design patterns, consistent terminology in prompts and standardized action commands throughout the website.

2) **Facilitate efficient navigation and shortcuts.**

   Regular users and frequent visitors are expected for proper learning experience. Therefore, easily accessible shortcuts and navigation would allow users to quickly return to their previous learning point and enhance their user experience.

3) **Offer informative feedback.**

   Providing feedback based on user input is key in creating an interactive game-based platform. Small examples of this would be when clicking on buttons or prompts, the site should provide some visual indication based on the user's input.

4) **Design dialog to yield closure.**

   Organizing sequences of actions into coherent groups with a clear beginning, middle, and end provides user with a sense of accomplishment and prepares users for further development.

5) **Reduce errors.**

   Strive to create a UI that minimizes the likelihood of user errors. Implement practices whereby any buttons that redirects user to another webpage should have a backup error page that allows users to return to the home page. User testing would also enable us to assess the functionality and user-friendliness of the project.

6) **Enable easy reversal of actions.**

   Ensure that users would have uncomplicated and apparent means of reversing their erroneous actions. As Shneiderman suggest, this feature alleviates user anxiety and encourages exploration within the webpage.

2.4.2 Navbar Design

The Navigation Bar (Navbar) is situated at the top of the interface and serves as a critical navigation element for the user. It should contain the most useful navigation to enhance user accessibility and adherence to the golden rules of UI Design.

1) **Branding**

   Incorporating the application logo within the Navbar maintains brand consistency throughout and reinforces the users' sense of place within the system.

2) **Navigational Links**

   Includes essential navigation links to different sections of the application to help user move quickly between relevant areas.

3) **Simplicity**

   To avert excessive clutter and information saturation, the navigation bar should embody simplicity and efficacy in communicating information to the user.

2.4.3 Sidebar Design

The Sidebar would only be used in our learning tutorial process pages. This navigation bar would be located on the left side of our page, which complements the Navbar by offering additional navigation options within our tutorial pages. The goal of this navigation bar would be to provide user an option to access any cipher that they prefer to learn through at any time, adhering to the golden rules of UI Design.

1) **Menu items**

   Structuring clear and concise menu items with relevant icons aids user in understanding the available sections or features at a glance.

2) **Collapsibility**

   Incorporating a collapsible design prevents overwhelming users with information and conserves space within the screen itself.

3) **Hierarchy**

   Organizing menu items in a hierarchy manner helps users understand the flow of the tutorials and allow users to guide themselves naturally and efficiently.

2.5 Software Choice

The chosen platform for developing this educational application is a web-based framework utilizing HTML, CSS, and JavaScript. Our choice would allow us to use React, a famous front-end JavaScript library, that allows us to create meaningful and interactive UI/UX components for the application. We would also be using Material-UI (MUI) for some simple icons and components used within the application.

Figure 7: Software Library used.

2.6 Conclusion

In conclusion, the UI design principles, along with thoughtful Navbar and Sidebar designs, collectively contribute to a user-centric interface that promotes efficient navigation and an enjoyable user experience. By adhering to these principles, we aim to create a system that not only fulfills user needs but also enhance the learning experience.

3. Software Development

The following section presents a comprehensive overview of the game-based design made for an educational application focused on ciphers. This application aims to engage users through interactive and informative experiences that teaches them various encryption techniques.

3.1 Hub Screens

The hub screens serve as navigational points of entry to various sections of the application. Each screens serves different purposes that improves user experience and ensure that new users do not feel lost within the application.

3.1.1 HomeScreen.js

The Home Screen is the welcoming interface of our application. Designed with user-friendliness in mind, it greets every visitor with a warm message, setting the tone for an immersive educational journey.

Central to this screen are two prominently placed buttons:

1. **'Start Learning!'**

   This button is strategically designed to attract newcomers, gently ushering them into the world of knowledge contained within the app.

2. **'Test your Skills!'**

   Aimed at seasoned users, this button serves as an invitation to challenge and validate their understanding.

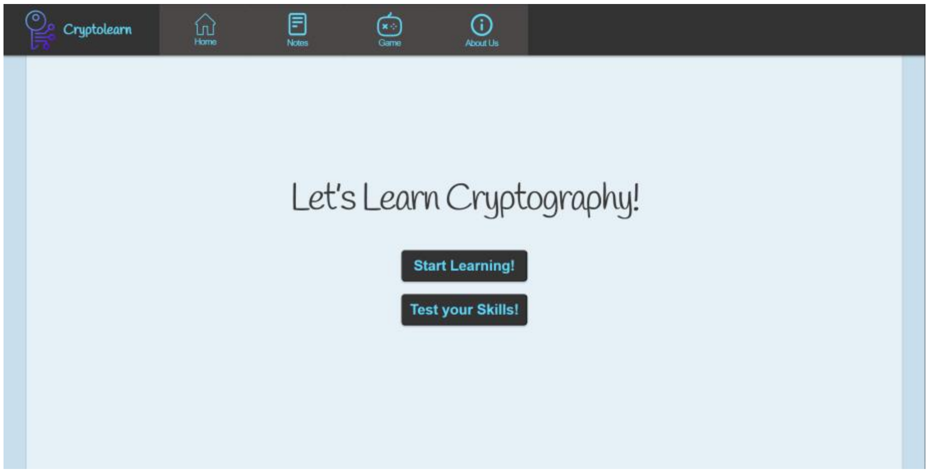
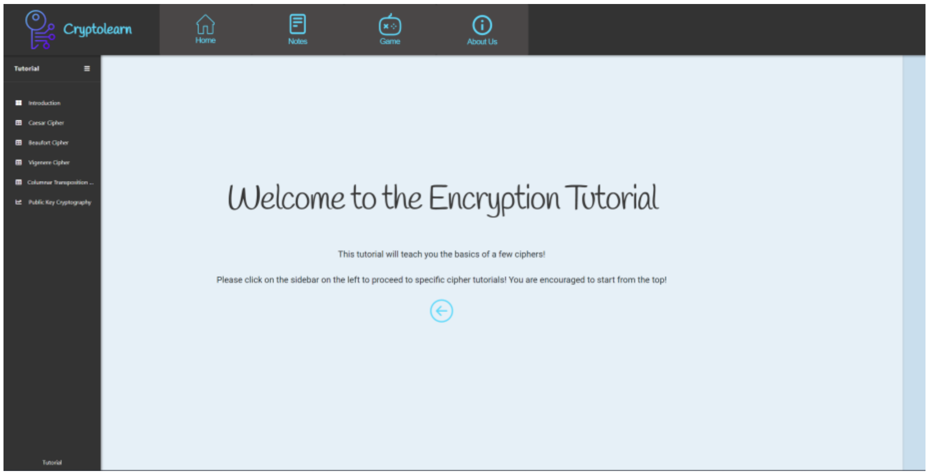


Figure 8: HomeScreen.js

The navigation bar will consistently appear across the application, ensuring users have a familiar point of reference regardless of their location within the interface. Additionally, both the 'Start Learning!' and 'Notes' buttons on the navigation bar will direct users to the same screen. Similarly, the 'Test your Skills!' and 'Game' buttons will lead to an identical screen experience. By ensuring such consistent navigation pathways, we aim to create an environment where learning is not just effective but also effortlessly enjoyable.

3.1.2 NotesScreen.js

When a user selects either the ‘Start Learning!’ or the ‘Notes’ buttons in the application, they are ushered into the introduction page of the tutorial screens. This introductory page is designed to orient users to the Sidebar, which is consistently positioned on the left of all tutorial screens. The Sidebar is the primary navigation tool for users to choose and delve into different ciphers.



Welcome to the Encryption Tutorial

This tutorial will teach you the basics of a few ciphers!

Please click on the sidebar on the left to proceed to specific cipher tutorials! You are encouraged to start from the top!

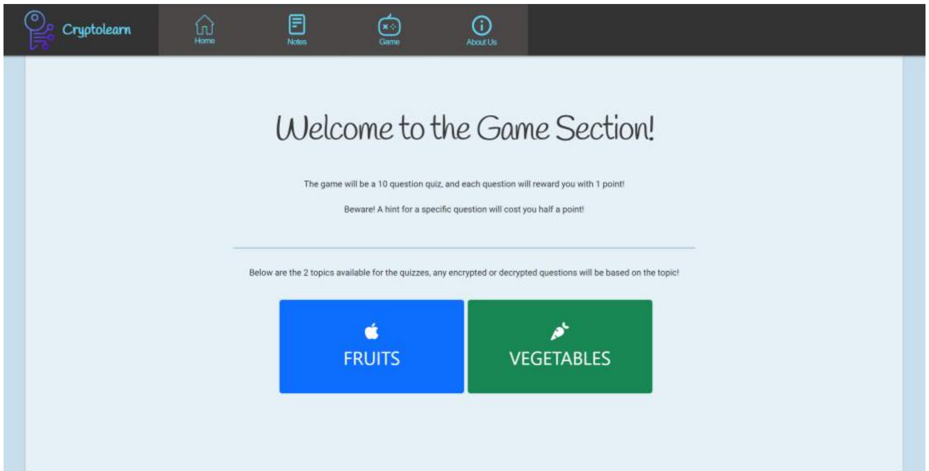
Figure 9: NotesScreen.js

As illustrated in Figure 9, the main screen intuitively directs users to select their desired cipher from the Sidebar. For those unfamiliar with the ciphers, we recommend starting from the top and progressing downward. This sequence is intentional, as the list is organized in increasing order of cipher complexity or uniqueness. Our design philosophy prioritizes user-friendliness, and this approach is consistently applied throughout the tutorial screens.

An added advantage of our Sidebar design is its modularity. As we introduce new ciphers to the application, they can be seamlessly integrated into the existing navigation structure, ensuring that updates are both efficient and non-disruptive for users.

3.1.3 GamesScreen.js

Upon entering the Games Introduction screen, users are greeted with an insight into the heart of our application: the game quizzes.



Welcome to the Game Section!

The game will be a 10 question quiz, and each question will reward you with 1 point!

Beware! A hint for a specific question will cost you half a point!

Below are the 2 topics available for the quizzes, any encrypted or decrypted questions will be based on the topic!

Figure 10: GamesScreen.js

This is what they can expect:

1) Purpose

This screen serves as a gateway to our quizzes. Whether a user clicks on 'Test your Skills!' or 'Game' from the main menu, they will be navigated to this introduction page.

2) Points System Overview

Before diving into the quizzes, users are briefed about the point system, helping them understand how their performance will be evaluated.

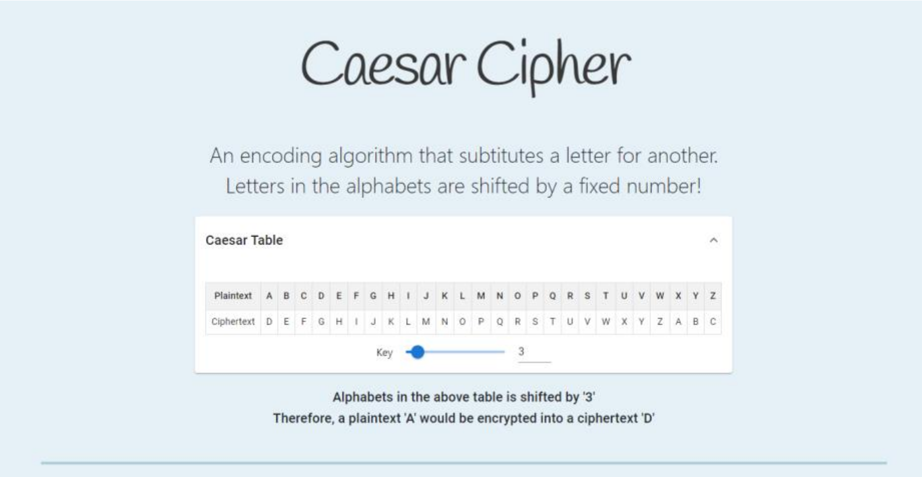
3) Topic Selection

We believe in giving our users choices. Therefore, on this screen, they can select from two distinct topics for their quiz: fruit names and vegetables names. These topics are chosen due to being universally recognized and simple to grasp. It ensures that users of all ages and backgrounds can enjoy the game without any prerequisites.

3.2 Caesar Tutorial

The Caesar Cipher is the first selection in our tutorial screen and is meant to be the first cipher taught to new users and beginners in cryptography. Therefore, it is important for the project to create a baseline knowledge for new users to work with.

A quick tutorial is given to the user at the start of the Caesar Cipher page as seen in Figure 11. The example below shows an alphabet shift for rotational value of 3. The default rotation value of the Caesar Table component is set to 3 for this example, but users can adjust the key by either dragging the bar or typing in their desired value. This would be a key component used in further game-based designs as it provides new user with an interactable interface specially created for Caesar Cipher.



Caesar Cipher

An encoding algorithm that subtitutes a letter for another.
Letters in the alphabets are shifted by a fixed number!

Caesar Table

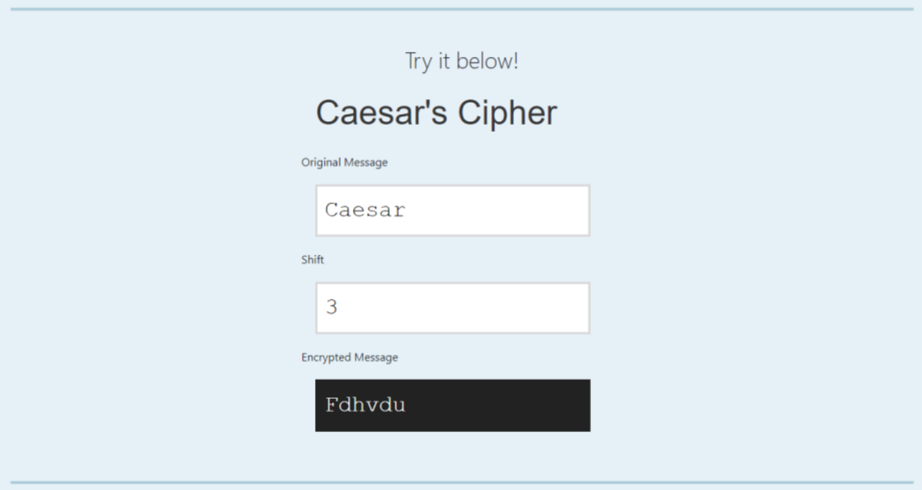
Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Key 3

Alphabets in the above table is shifted by '3'
Therefore, a plaintext 'A' would be encrypted into a ciphertext 'D'

Figure 11: Caesar Cipher Introduction

On the same screen, a component for Caesar Cipher cryptography encryption is included, allowing users to experiment with their own messages and shift values. A default value of 'Caesar' with a shift of '3' is provided to familiarize users with the example they previously saw on the screen.



Try it below!

Caesar's Cipher

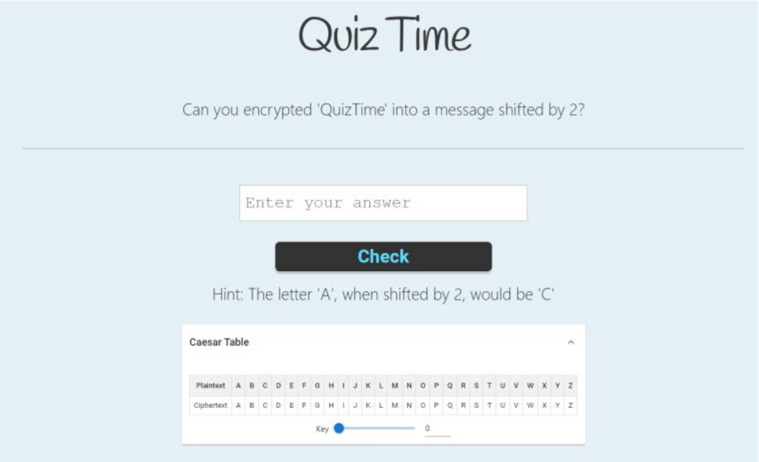
Original Message: Caesar

Shift: 3

Encrypted Message: Fdhvdu

Figure 12: Caesar Cipher Encryption Component

Now that users have been introduced to how Caesar ciphers work, they can proceed to the game-based section. Here, we evaluate how well they've retained the information from the previous page. As a checkpoint, users are tested on the encryption of the plaintext 'QuizTime' using a shift value of 2. To aid them, a hint is provided, referencing the first example they encountered. This hint demonstrates the result of shifting the letter 'A' by the specified shift value. Additionally, users have access to the Caesar Table component, which enables them to easily encrypt values using an interactive table. The figure below shows this implementation.



Quiz Time

Can you encrypted 'QuizTime' into a message shifted by 2?

Hint: The letter 'A', when shifted by 2, would be 'C'

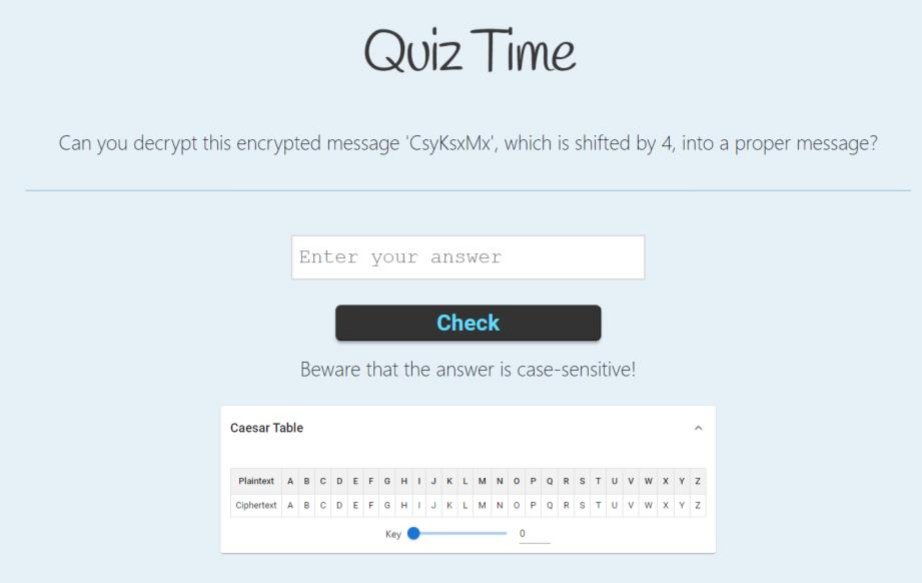
Caesar Table

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Key 0

Figure 13: Caesar Cipher Encryption Checkpoint

The next step involves testing the user on Caesar Cipher Decryption. This checkpoint mirrors the one in Figure 13, where the user is presented with a question to answer. They are provided with an encrypted message and a shift value and are tasked with decrypting the message to reveal the plaintext. Importantly, the message incorporates both uppercase and lowercase letters. This serves as a subtle knowledge check, reminding users that both uppercase and lowercase letters share the same encryption and decryption values in Caesar Cipher.



Quiz Time

Can you decrypt this encrypted message 'CsyKsxMx', which is shifted by 4, into a proper message?

Beware that the answer is case-sensitive!

Caesar Table

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Key 0

Figure 14: Caesar Cipher Decryption Checkpoint

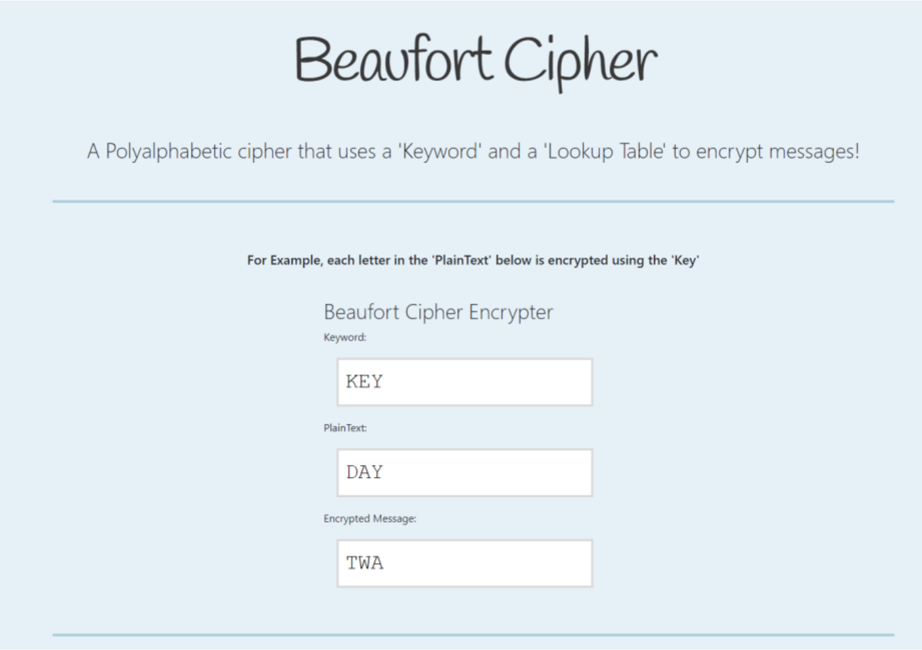
Once the user completes this checkpoint, they have successfully finished the Caesar Cipher Tutorial. This preparation ensures they are well-equipped for the Caesar Cipher Game Quiz or the Final Quiz, both of which are presented in the Game Section.

In summary, the Caesar Cipher tutorial serves as a foundational introduction to the realm of cryptography for newcomers. By incorporating interactive components and game-based checkpoints, users are not only educated but also tested on their comprehension of the cipher's mechanics. This immersive learning approach ensures that participants are actively engaged and effectively prepared for subsequent challenges. The Caesar Cipher, as the first cipher presented, establishes a solid baseline, setting the stage for more complex cryptographic concepts that users might encounter in the future.

3.3 Beaufort Tutorial

The Beaufort Cipher is the second among the four symmetrical key ciphers presented in this application. While it maintains some similarities with the Caesar Cipher, particularly in using a rotational value for encryption, it introduces users to a nuanced method of key-to-letter cryptography. Unlike the Caesar Cipher, which applies a single rotational value to the entire plaintext, the Beaufort Cipher rotates each letter in the plaintext based on the specific key provided.

In the Beaufort Tutorial, users are guided on how to utilize the Beaufort Lookup Table to encrypt each letter according to its corresponding key. Initially, Figure 15 demonstrates the process of encrypting plaintext using a key within the Beaufort Cipher. The Beaufort Encryption component processes user inputs of a plaintext message and a keyword to generate an encrypted output based on the Beaufort cipher. This component aims to highlight the influence of keywords in the encryption and provide users with a hands-on experience of the Beaufort cipher.



Beaufort Cipher

A Polyalphabetic cipher that uses a 'Keyword' and a 'Lookup Table' to encrypt messages!

For Example, each letter in the 'PlainText' below is encrypted using the 'Key'

Beaufort Cipher Encrypter

Keyword:

KEY

PlainText:

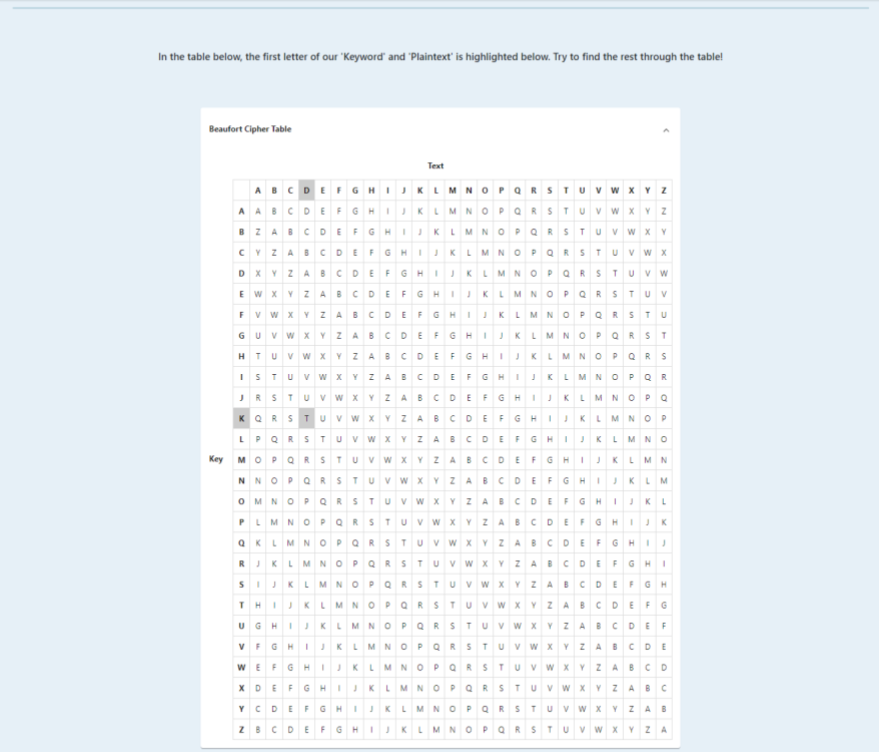
DAY

Encrypted Message:

TWA

Figure 15: Beaufort Cipher Encryption Component

This component would work in tandem with the next component shown below in Figure 16. This interactable table enables users to encrypt each letter according to its associated key and serves as a versatile tool for any encryption or decryption tasks related to the Beaufort Cipher. It dynamically selects the appropriate letters based on the user input.



In the table below, the first letter of our 'Keyword' and 'Plaintext' is highlighted below. Try to find the rest through the table!

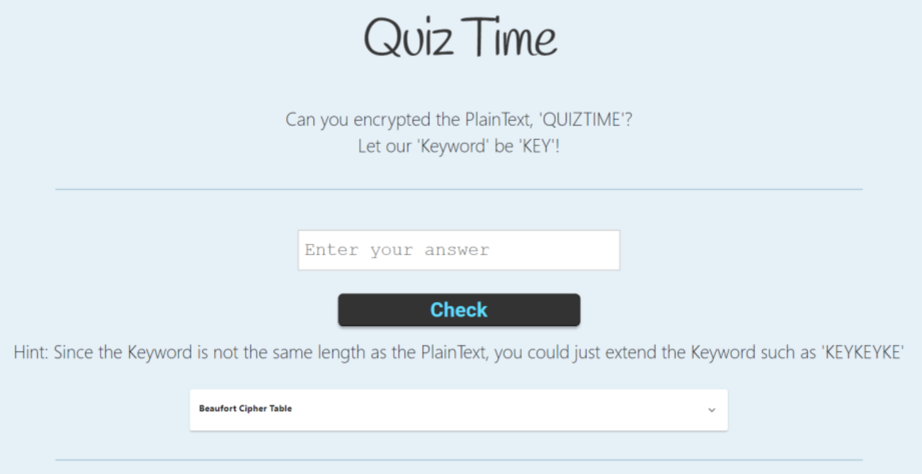
Beaufort Cipher Table

Text

Key	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
C	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
D	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
E	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
F	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
G	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
H	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
I	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
J	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
K	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
L	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
M	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
P	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
Q	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
R	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
S	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
T	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
U	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
V	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
W	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
X	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
Y	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
Z	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Figure 16: Beaufort Cipher Table Component

In our implementation, the default key and text correspond to the first letter from the example in Figure 15. We prompt users to complete the example using the table. Through this interactive exercise, newcomers learn how to employ the Beaufort Cipher Table for both encryption and decryption tasks, using a provided key. Once users have familiarized themselves with this concept, we will move on to the Beaufort Encryption checkpoint.



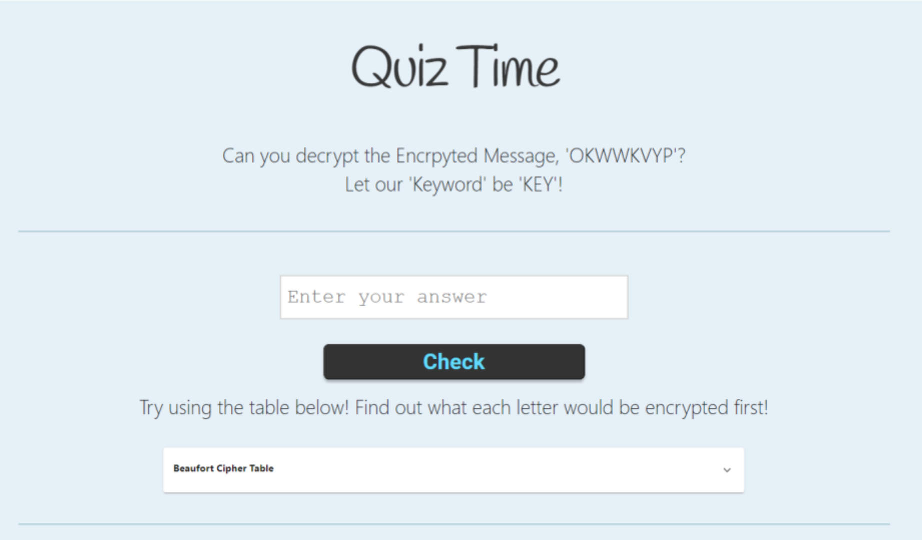
Quiz Time

Can you encrypted the PlainText, 'QUIZTIME'?
Let our 'Keyword' be 'KEY'!

Hint: Since the Keyword is not the same length as the PlainText, you could just extend the Keyword such as 'KEYKEYKE'

Figure 17: Beaufort Cipher Encryption Checkpoint

In Figure 17, we challenge users with a Beaufort encryption test. While it uses the same key as the example, the plaintext is notably longer. This introduces a common problem for beginners: what to do when the plaintext exceeds the key's length. To guide users, a hint is provided, illustrating how to extend the keyword to encrypt longer plaintexts. For further assistance, the Beaufort Cipher table is conveniently provided below the challenge. This checkpoint would reinforce the knowledge retained from the previous page and prepare them for the next checkpoint which would test them on Beaufort decryption.



Quiz Time

Can you decrypt the Encrpyted Message, 'OKWWKVYP'?
Let our 'Keyword' be 'KEY'!

Try using the table below! Find out what each letter would be encrypted first!

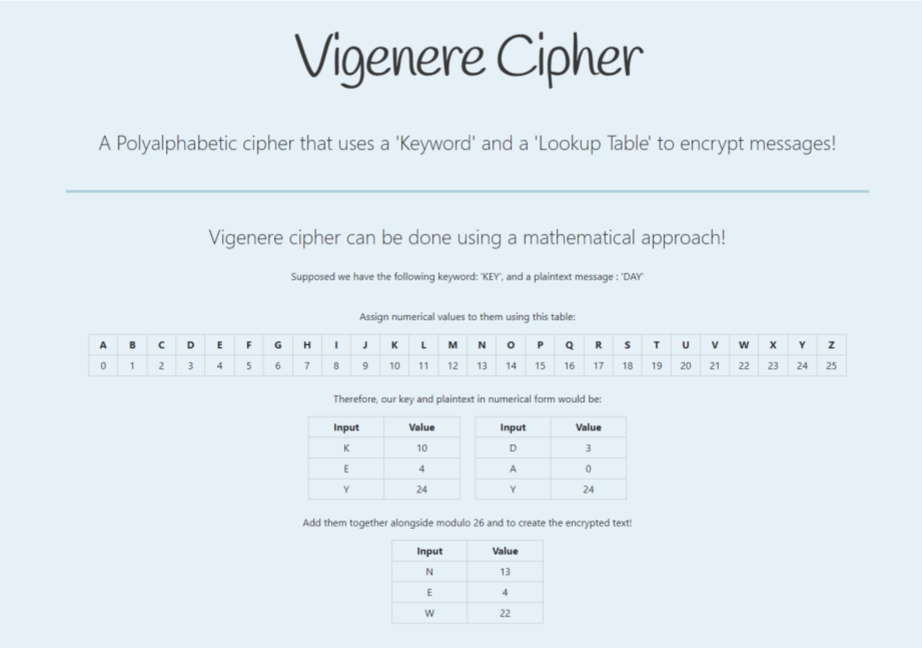
Figure 18: Beaufort Cipher Decryption Checkpoint

In Figure 18, we present a challenge similar to the previous encryption checkpoint. However, this time, instead of encrypting plaintext, users are tasked with decrypting an encrypted text. This exercise enforces the symmetrical nature of Beaufort ciphers. As always, the Beaufort Cipher table is provided to assist users throughout the process.

In conclusion, the Beaufort Cipher tutorial seamlessly guides users from foundational knowledge to hands-on application. Through interactive components, users are introduced to the unique key-to-letter encryption method of the Beaufort Cipher, distinguishing it from the more familiar rotational encryption like that of the Caesar Cipher. Challenges presented in the tutorial not only test users on encryption but also decryption, emphasizing the symmetrical nature of the cipher. By providing users with practical tools like the Beaufort Cipher table and offering hints for extended application, the tutorial ensures that learners are both educated and equipped to handle a range of cryptographic scenarios involving the Beaufort Cipher.

3.4 Vigenère Tutorial

The Vigenère cipher stands as the third of the four symmetrical key ciphers featured in this application. Within this tutorial, we delve into two distinct methods for encryption and decryption using the Vigenère cipher: the mathematical approach and the lookup table approach. As depicted in Figure 19, we employ tables and columns to illustrate the process of encrypting plaintext via the mathematical method, using a specified key.



Vigenere Cipher

A Polyalphabetic cipher that uses a 'Keyword' and a 'Lookup Table' to encrypt messages!

Vigenere cipher can be done using a mathematical approach!

Supposed we have the following keyword: 'KEY', and a plaintext message : 'DAY'

Assign numerical values to them using this table:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Therefore, our key and plaintext in numerical form would be:

Input	Value
K	10
E	4
Y	24

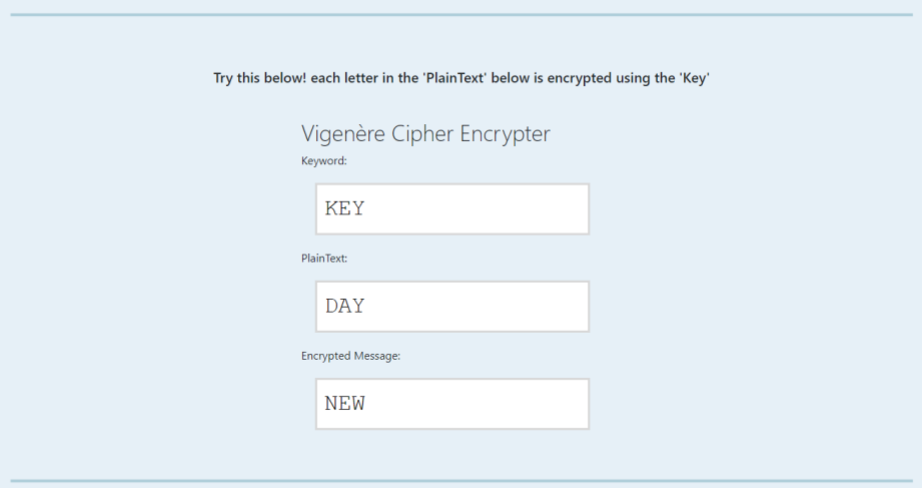
Input	Value
D	3
A	0
Y	24

Add them together alongside modulo 26 and to create the encrypted text!

Input	Value
N	13
E	4
W	22

Figure 19: Vigenère Cipher Mathematical Approach

In Figure 20, we introduce a Vigenère cipher encryption component, granting users the freedom to experiment with their chosen keywords and plaintext. Both the mathematical method and this encryption component default to the same keyword and plaintext, ensuring a consistent and familiar experience for users as they navigate the page.



Try this below! each letter in the 'PlainText' below is encrypted using the 'Key'

Vigenère Cipher Encrypter

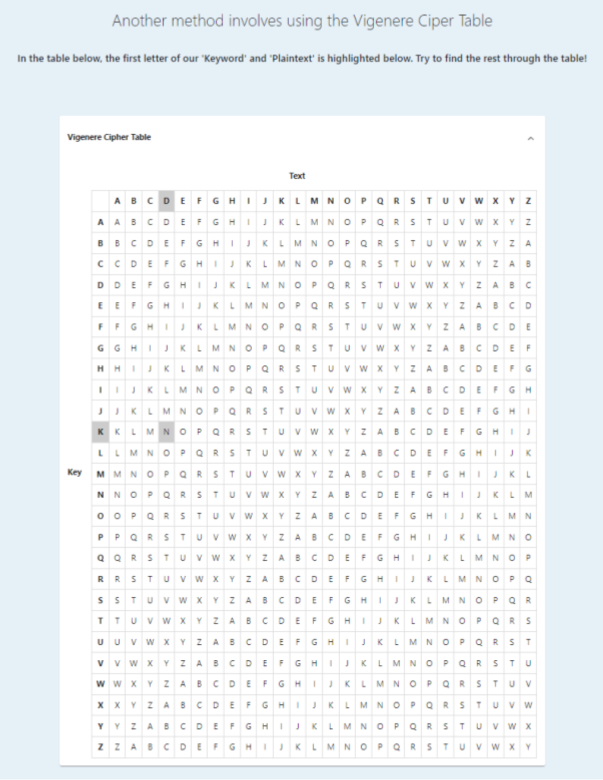
Keyword: KEY

PlainText: DAY

Encrypted Message: NEW

Figure 20: Vigenère Cipher Encryption Component

An alternative method for encryption and decryption using the Vigenère cipher is through a dedicated lookup table. As illustrated in Figure 21, users are guided to employ the provided Vigenère cipher table to encrypt each letter, referencing the example above. By maintaining consistency in the keyword and plaintext across methods, we aim to facilitate a seamless learning experience, capitalizing on the familiar yet highlighting the nuanced differences between the two approaches.



Another method involves using the Vigenere Ciper Table

In the table below, the first letter of our 'Keyword' and 'Plaintext' is highlighted below. Try to find the rest through the table!

Vigenere Cipher Table

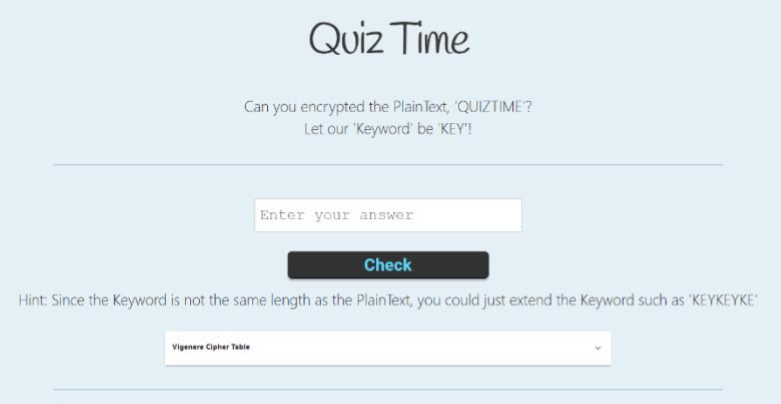
Text

Key

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 21: Vigenère Cipher Table Component

Following the instruction on the two approaches, we transition to the Vigenère encryption checkpoint. As with the Beaufort Cipher, this challenge examines the user's ability to handle encryption with a lengthier plaintext and a shorter keyword. The rationale for this repetition is to accommodate the application's flexible design, where users have the liberty to choose the order of cipher tutorials. Thus, it is possible for users to arrive at this checkpoint without having experienced the Beaufort Encryption Checkpoint. For those preferring the lookup table method, the Vigenère cipher table is readily available, providing an alternative to the mathematical approach. This checkpoint is depicted on Figure 22.



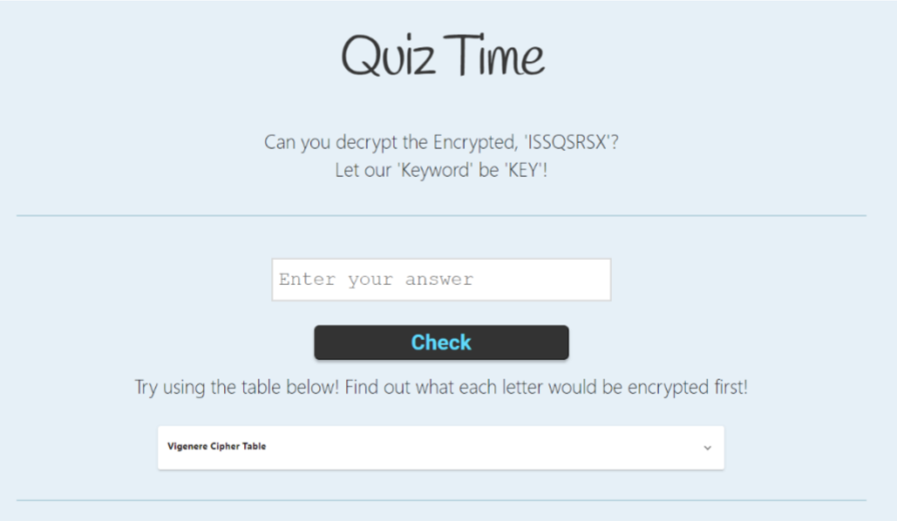
Quiz Time

Can you encrypted the PlainText, 'QUIZTIME'?
Let our 'Keyword' be 'KEY'!

Hint: Since the Keyword is not the same length as the PlainText, you could just extend the Keyword such as 'KEYKEYKE'

Figure 22: Vigenère Encryption Checkpoint

Upon successfully completing the Vigenère encryption checkpoint, users progress to the Vigenère decryption challenge. Mirroring previous decryption checkpoints, our aim is to assess their understanding of symmetrical key encryption and decryption, with a focus on the nuances of the Vigenère cipher. The Vigenère Cipher Table is added as always for users who prefer the lookup table approach. This is depicted in Figure 23.



Quiz Time

Can you decrypt the Encrypted, 'ISSQSRSX'?
Let our 'Keyword' be 'KEY'!

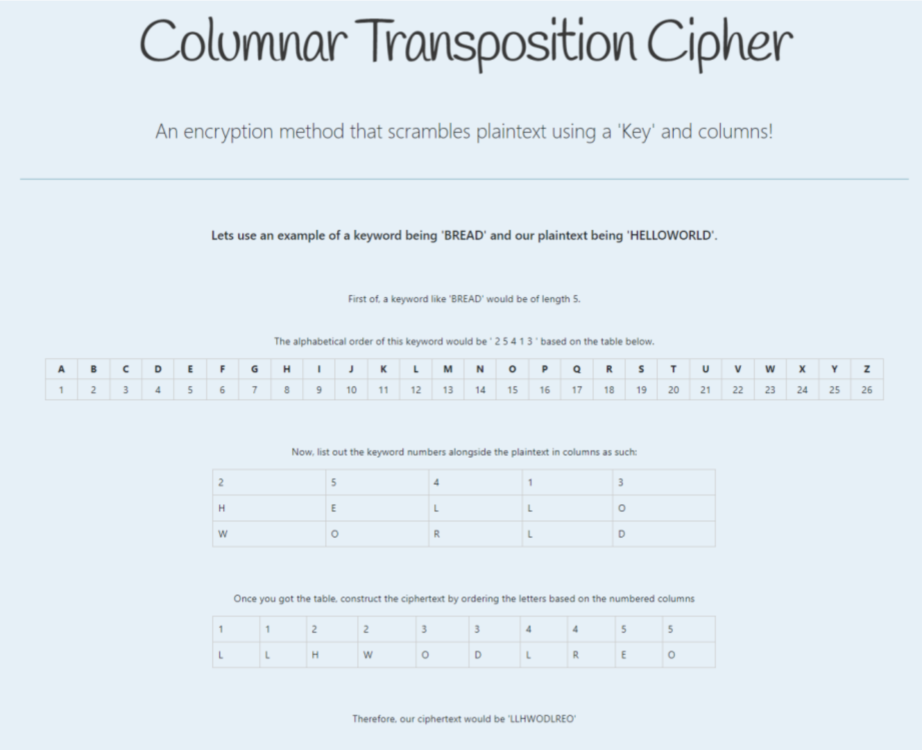
Try using the table below! Find out what each letter would be encrypted first!

Figure 23: Vigenère Decryption Checkpoint

In conclusion, the Vigenère cipher tutorial offers users a comprehensive exploration of this cryptographic method. By presenting two distinct approaches, the mathematical method and the lookup table, the tutorial caters to varied learning preferences and styles. Challenges, both in encryption and decryption, are designed not just to test users' grasp of the Vigenère cipher but also to emphasize the broader principles of symmetrical key ciphers. By ensuring consistent tools, like the Vigenère Cipher Table, and familiar setups across challenges, the tutorial aims for a seamless learning curve. Whether users approach the ciphers in sequence or jump directly to the Vigenère cipher, the tutorial is structured to ensure a solid understanding and application of its principles.

3.5 Columnar Transposition Tutorial

The Columnar encryption screen introduces users to the Columnar transposition cipher, which focus on permutations and reordering of characters, and is the final symmetric key cipher taught with the application. Being completely different from the previously taught ciphers so far, we decided to provide a more concrete example to help new users familiarize themselves with how the cipher functions. We also provided another encryption component that allows users to input their own plaintext and key which would reinforce the learning process and the example given. This process is shown in Figure 24.



Columnar Transposition Cipher

An encryption method that scrambles plaintext using a 'Key' and columns!

Lets use an example of a keyword being 'BREAD' and our plaintext being 'HELLOWORLD'.

First of, a keyword like 'BREAD' would be of length 5.

The alphabetical order of this keyword would be ' 2 5 4 1 3 ' based on the table below.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Now, list out the keyword numbers alongside the plaintext in columns as such:

2	5	4	1	3
H	E	L	L	O
W	O	R	L	D

Once you got the table, construct the ciphertext by ordering the letters based on the numbered columns

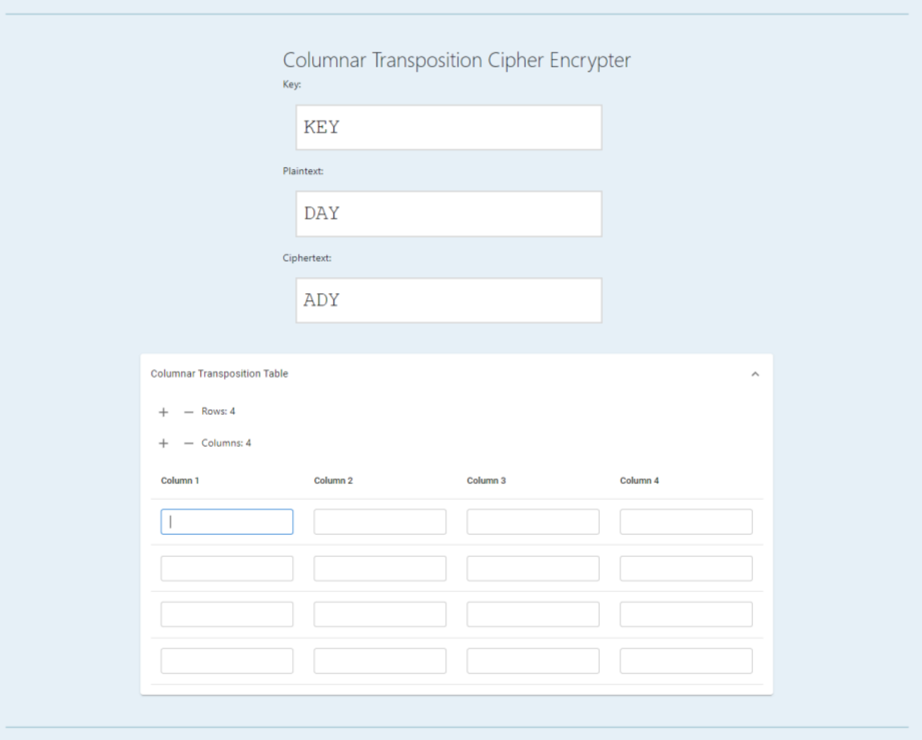
1	1	2	2	3	3	4	4	5	5
L	L	H	W	O	D	L	R	E	O

Therefore, our ciphertext would be 'LLHWODLREO'

Figure 24: Columnar Transposition Cipher Example

In Figure 25, we showcase the Columnar encryption component. Users input a plaintext message and a keyword, and the component produces an encrypted output using the principles of the Columnar Transposition cipher. This hands-on interaction offers users valuable insight into the mechanics of how the cipher rearranges characters based on the keyword's specific ordering, thereby deepening their understanding of alternative encryption strategies.

Furthermore, users are encouraged to experiment with their chosen plaintext and keyword. To aid in this exploration, an empty Columnar Transposition Table is provided. This interactive template allows firsthand execution of columnar transpositions. Notably, the table is designed with adaptability in mind: users can adjust its rows and columns to fit their needs. Such flexibility ensures its applicability across all variations of the columnar transposition cipher.



Columnar Transposition Cipher Encrypter

Key: KEY

Plaintext: DAY

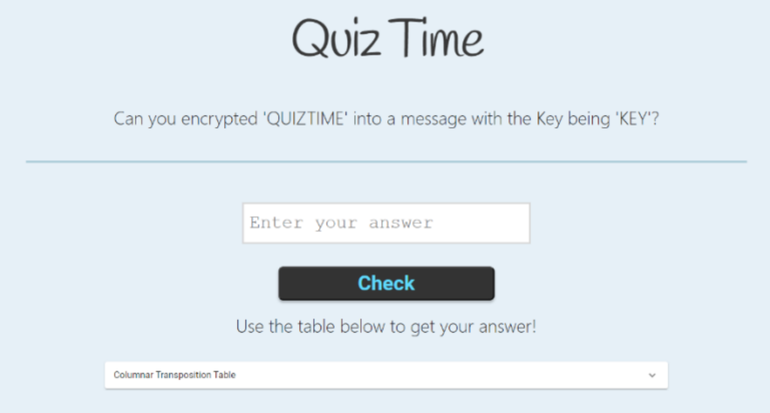
Ciphertext: ADY

Columnar Transposition Table

Column 1	Column 2	Column 3	Column 4

Figure 25: Columnar Transposition Encryption Component

Having acquainted users with the mechanics of columnar transposition encryption, we can move towards a game-based test component. In line with the assessments for other symmetrical key ciphers, we challenge users with an encryption task using a specified plaintext and keyword. To support their problem-solving, the columnar transposition table component is provided, offering a visual and interactive platform for users to map out their thought process. This is illustrated in Figure 26 below.



Quiz Time

Can you encrypted 'QUIZTIME' into a message with the Key being 'KEY'?

Use the table below to get your answer!

Figure 25: Columnar Transposition Encryption Checkpoint

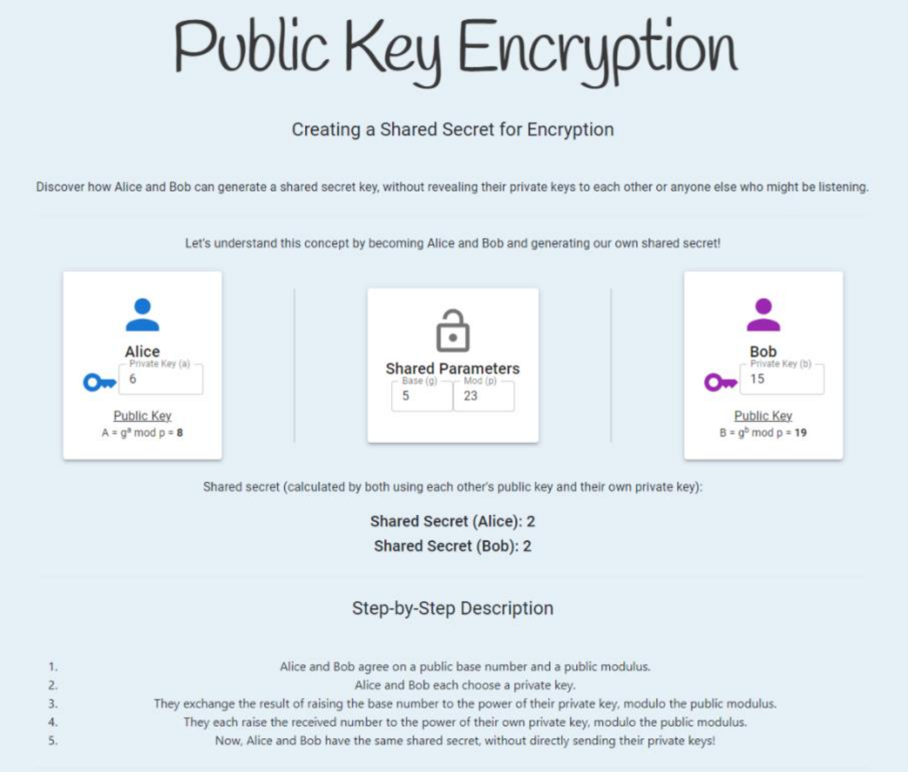
In conclusion, the Columnar Transposition cipher tutorial offers users a deep dive into this unique encryption method, differentiating it from other symmetrical key ciphers. By enabling hands-on experimentation through the encryption component and the Columnar Transposition Table, users gain both theoretical knowledge and practical experience. These tools serve as invaluable aids in allowing users to visualize the intricate process of rearranging characters based on keyword ordering. Through challenges and interactive components, the tutorial ensures that learners not only understand the cipher's principles but can also apply them effectively. The Columnar Transposition cipher tutorial is used to engage users in a comprehensive and engaging cryptographic education.

3.6 Diffie-Hellman Tutorial

The Diffie-Hellman algorithm stands distinctively amidst the cryptographic methods presented in this application, not as a cipher, but as a groundbreaking protocol for secure key exchange. Originating from the pioneering work of Whitfield Diffie and Martin Hellman, this protocol allows two parties to generate a shared, secret key over an insecure channel without having previously shared any secrets.

In Figure 26, a visual representation of this key exchange is illustrated between two parties, Alice and Bob, which they established their private and public keys and subsequently generating a shared secret. Users can customize this experience by editing the base, modulus and private keys in the respective input boxes. This interactable interface allows users to experiment their own parameters and see the changes between Alice and Bob.

A simple Step-by-Step description is provided for users to help them fully understand the process of the Diffie-Hellman algorithm. However, as public key encryption is quite complicated, the Diffie-Hellman algorithm is a great first step in understanding how to solve the logistics dilemma of a symmetric key encryption.



Public Key Encryption

Creating a Shared Secret for Encryption

Discover how Alice and Bob can generate a shared secret key, without revealing their private keys to each other or anyone else who might be listening.

Let's understand this concept by becoming Alice and Bob and generating our own shared secret!

Alice
Private Key (a): 6
Public Key
$A = g^a \bmod p =$ **8**

Shared Parameters
Base (g): 5
Mod (p): 23

Bob
Private Key (b): 15
Public Key
$B = g^b \bmod p =$ **19**

Shared secret (calculated by both using each other's public key and their own private key):

Shared Secret (Alice): 2

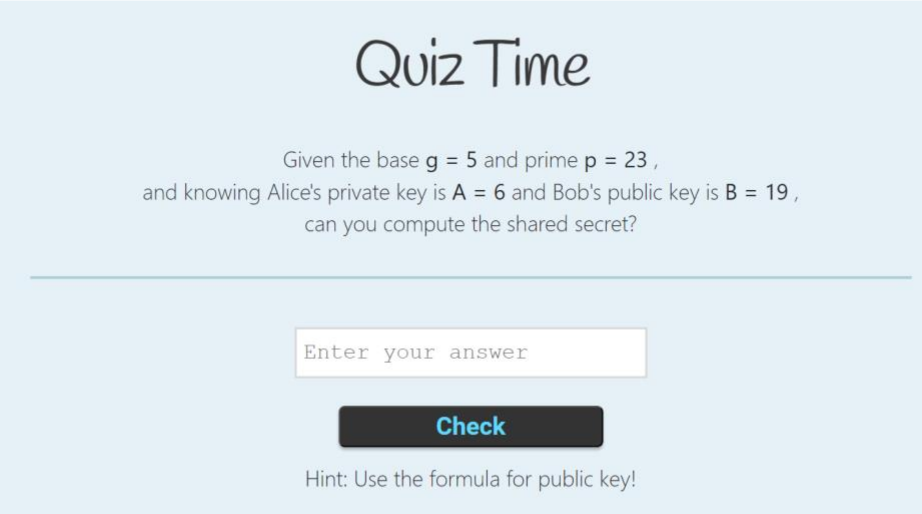
Shared Secret (Bob): 2

Step-by-Step Description

1. Alice and Bob agree on a public base number and a public modulus.
2. Alice and Bob each choose a private key.
3. They exchange the result of raising the base number to the power of their private key, modulo the public modulus.
4. They each raise the received number to the power of their own private key, modulo the public modulus.
5. Now, Alice and Bob have the same shared secret, without directly sending their private keys!

Figure 26: Diffie-Hellman Example

To gauge users' understanding, a challenge is presented in Figure 27. Users are tasked with determining a shared secret based on given public keys and given private key. As with other challenges, this exercise aims to test not just theoretical knowledge but practical application skills.



Quiz Time

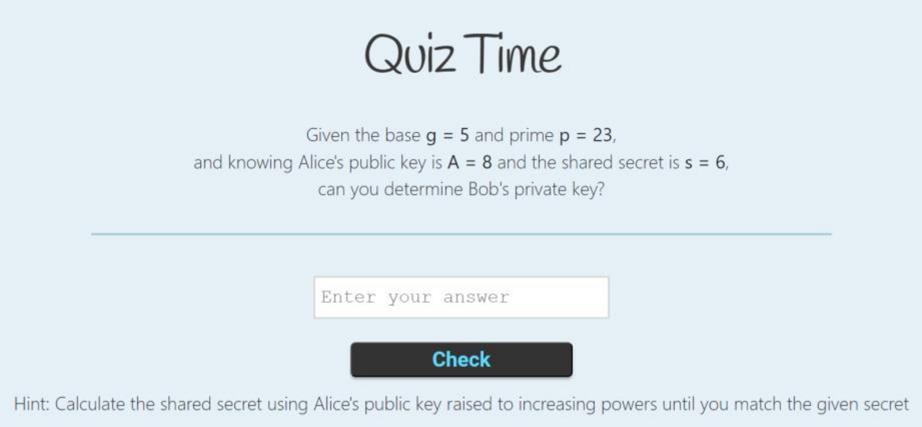
Given the base g = 5 and prime p = 23 ,
and knowing Alice's private key is A = 6 and Bob's public key is B = 19 ,
can you compute the shared secret?

Hint: Use the formula for public key!

Figure 27: Diffie-Hellman Shared Key Quiz

Now that users have gained a foundational understanding of the Diffie-Hellman algorithm, Figure 28 introduces a more advanced challenge. In this exercise, users are presented with the typical base and prime numbers. However, the twist lies in determining Bob's private key using only Alice's public key and the shared secret. This challenge simulates real-world scenarios where one might need to deduce certain cryptographic elements from limited information.

task serves a dual purpose: firstly, it tests users' comprehension and adaptability in applying the Diffie-Hellman principles; secondly, it emphasizes the importance of securing private keys in practical applications. Such a challenge not only deepens their grasp of the algorithm but also underscores the nuances and intricacies of secure key exchanges.



Quiz Time

Given the base g = 5 and prime p = 23,
and knowing Alice's public key is A = 8 and the shared secret is s = 6,
can you determine Bob's private key?

Hint: Calculate the shared secret using Alice's public key raised to increasing powers until you match the given secret

Figure 28: Diffie-Hellman Private Key Quiz

The Diffie-Hellman tutorial, with its blend of theoretical exposition and hands-on challenges, ensures that users walk away with a thorough understanding of secure key exchange protocols. The interactive components and challenges are testament to the application's dedication to making even complex cryptographic principles accessible and engaging.

3.7 Selection Game Screen

Upon selecting their topic of interest, whether it be fruit names or vegetable names, users are navigated to the quiz game screen. Here, they are presented with an array of quiz options, including the Caesar Quiz, Beaufort Quiz, Vigenère Quiz, Columnar Transposition Quiz, and the comprehensive Final Quiz, which amalgamates elements from all the previous quizzes. Each quiz is meticulously crafted to comprise 10 unique questions, with keys or rotational values randomly assigned to each, ensuring a dynamic and replayable gaming experience.

Moreover, users are afforded the flexibility to tailor their quiz experience according to their proficiency level. They can opt for level 1, which focuses exclusively on decryption questions, or level 2, which challenges them with a set of encryption questions. Regardless of the level or quiz type selected, users are guaranteed a stimulating and educational experience that tests their cryptographic skills and knowledge. This implementation is shown in Figure 29.

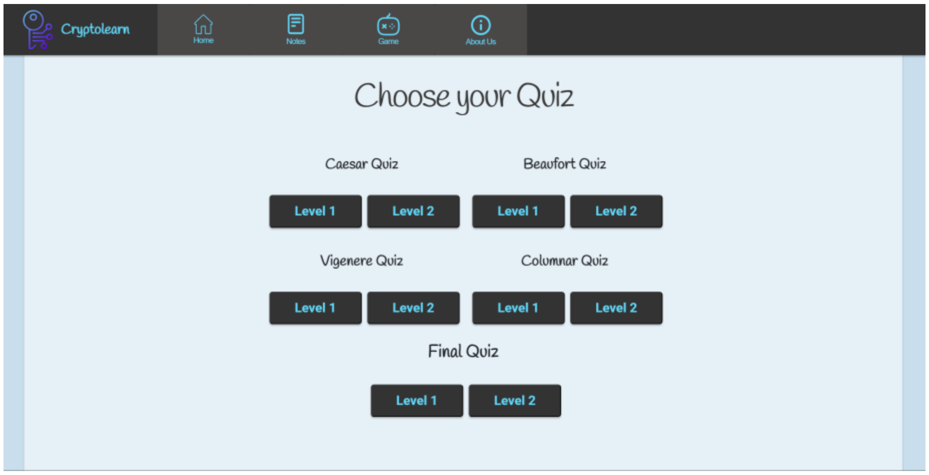
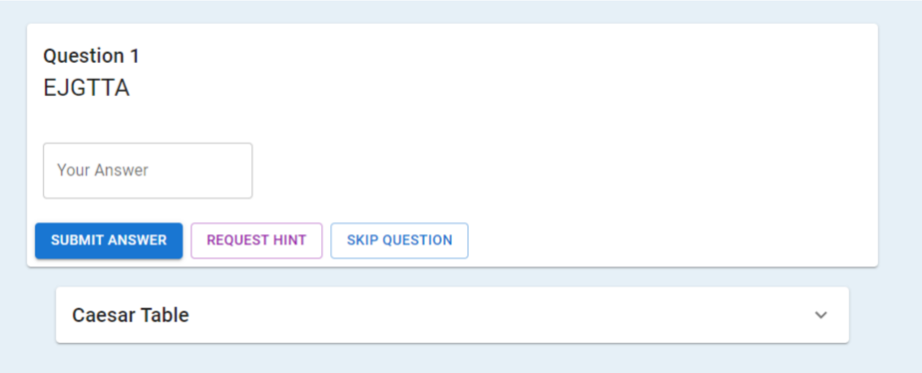


Figure 29: Selection Quiz Game Screen

3.8 Quizzes

After selecting their preferred topics, cipher types, and difficulty levels, users commence the quiz. Each quiz comprises 10 randomized questions, with keys or rotational values assigned based on the cipher type. As the formats across quizzes are largely uniform, the Caesar Cipher Quiz will be used as a demonstration in the figures below to illustrate the quiz section's mechanics, beginning with Figure 30.



Question 1

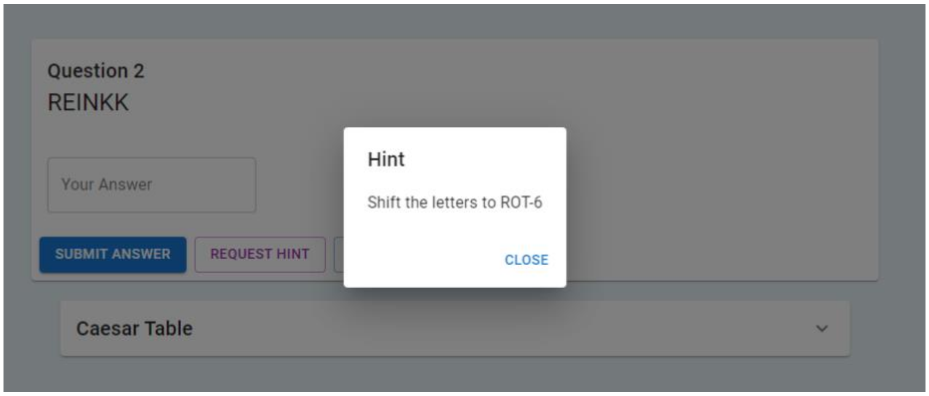
EJGTTA

Figure 30: Quiz Decryption Screen

In this segment of the application, users are presented with a series of encrypted texts. Their primary objective is to apply their knowledge from the tutorials to decipher these texts, submitting their plaintext values through an input box. The design choice to focus solely on uppercase values serves to create a more streamlined and focused learning experience. To aid users in their endeavors, each quiz is accompanied by a corresponding cipher table, located at the bottom of the screen. This table not only serves as a handy reference tool but also as a bridge connecting new knowledge with previous learning experiences from the tutorials.

In the quiz box, they are given 3 options to tackle this question. The 'Skip Question' button allows users to bypass challenging questions, fostering a more exploratory learning experience. As the questions are all randomized, we wanted to give users the chance to experience more questions instead of getting them stuck without a way forward. However, it is important to note that skipped questions do not contribute to the user's score.

The 'Request Hint' option provides assistance at the cost of half the points for that question. This trade-off ensures that users remain engaged and motivated to try for a correct answer without using the hint system. This implementation is showed in Figure 31.


Question 2

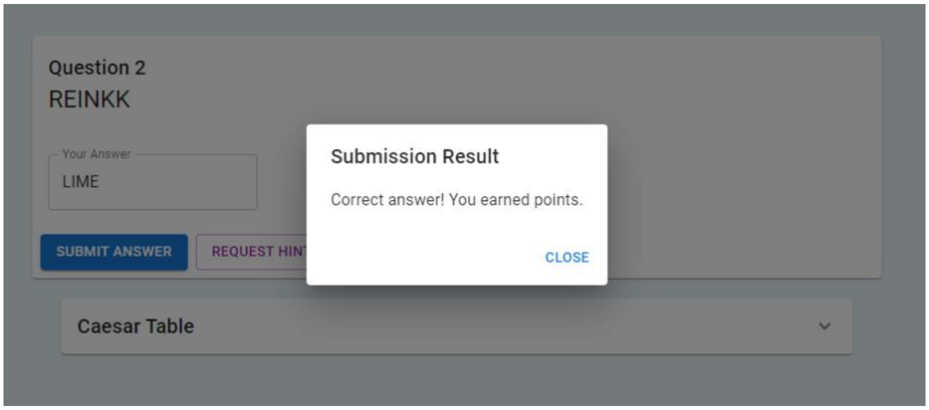
REINKK

Hint

Shift the letters to ROT-6

Figure 31: Hint Prompt

Regardless of using the hint or not, inputting the correct answer with the 'Submit Answer' button will give them the correct answer prompt. Once they get the correct answer, they will instantly be moved to the next question in the quiz and earn points with respect to whether they used the hint system or not. This prompt can be seen in Figure 32.


Question 2

REINKK

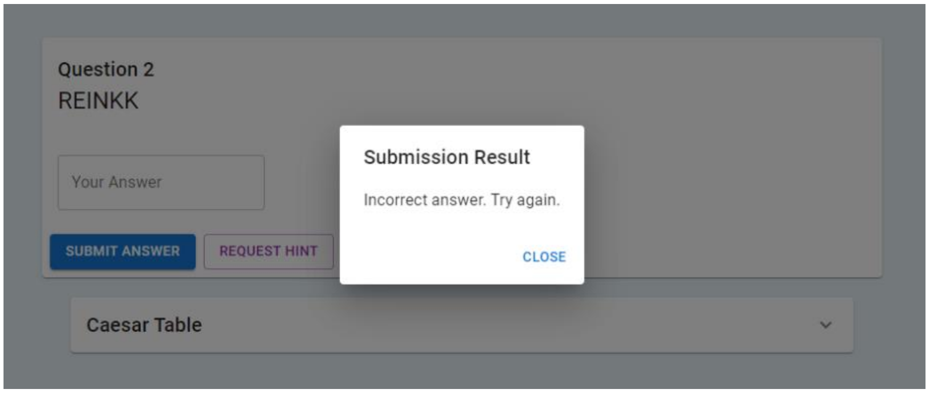
Your Answer: LIME

Submission Result

Correct answer! You earned points.

Figure 32: Correct Answer Prompt

However, if they input the wrong answer, they will be given the wrong answer prompt. The quiz would not push them to the next question, and instead encourages another attempt. There is no penalty for a wrong answer. This implementation can be seen in Figure 33.


Question 2

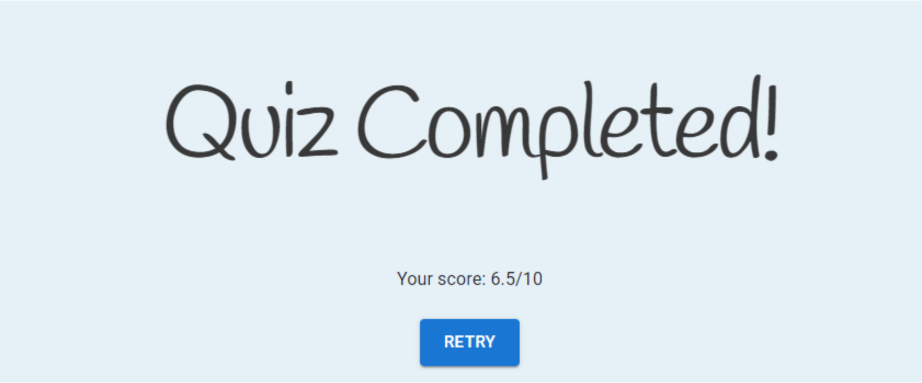
REINKK

Submission Result

Incorrect answer. Try again.

Figure 33: Wrong Answer Prompt

Lastly, once users have completed 10 questions, they will be led to the quiz completion result page. This will showcase their score based on how well they did in the questions above. Additionally, users are provided with options to either retry the quiz for a better score or explore other cipher quizzes. This can be seen in Figure 33.

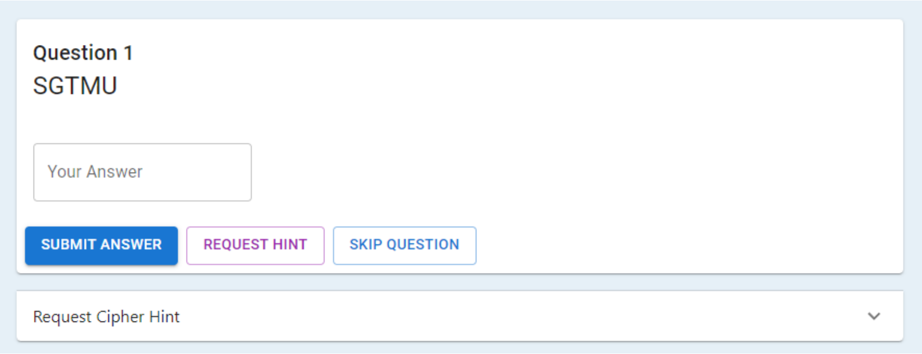


Quiz Completed!

Your score: 6.5/10

Figure 34: Quiz Result Page

Once users familiarize themselves with all the other quizzes, they can tackle the final quiz, which is slightly different from the quizzes before them. As the question are all randomized, even the ciphers type is completely random. Therefore, to level the playing field, users are granted the option to request a 'cipher hint'. This hint will not only provide the user with what type of cipher it is, but also the corresponding table that they will need to solve the question. However, this hint will also reduce the score this question by half. This balance ensures that users are both challenged and supported in this final learning journey. This implementation can be seen in Figure 35.



Question 1

SGTMU

Figure 35: Final Quiz Implementation.

The quiz section of the application offers a comprehensive and interactive learning experience, seamlessly integrating previously taught cryptographic concepts into a series of engaging challenges. Through a variety of quizzes, users can apply and solidify their knowledge, while having the flexibility to tailor the difficulty level and receive hints when needed. The intuitive design, coupled with instant feedback and supportive tools, ensures a balanced blend of learning and testing. Ultimately, whether users are navigating through specific cipher quizzes or taking on the final challenge, the application fosters a deepened understanding and proficiency in cryptography.

4. Testing

During the development phase of the application, we enlisted the assistance of several participants to interact with and assess the educational and gamified components of our project. Our primary goals were to evaluate the user-friendliness of the interface for individuals unfamiliar with cryptography, gauge the efficacy of the tutorial in imparting cryptographic knowledge, and understand the impact of the quiz feature on the participants' cryptography deciphering skills. Through this process, we aimed to test the application in hopes of holding up to its project objectives and scope.

4.1 User A

User A, with no substantial background in cryptography and not majoring in Computer Science or Computer Engineering, provided valuable insights for our project. His exposure to cryptography was primarily through informal settings like online forums and conversations with friends, making his feedback crucial to meeting the project's objectives.

User A was able to navigate through the interfaces properly and managed to start learning from the Caesar cipher tutorial section. He exhibited curiosity and engagement by exploring different cipher tutorials, although he encountered challenges, particularly with the Beaufort Cipher and the Diffie-Hellman Algorithm. His experience showed the need for additional support or simplified explanations in those sections. Fortunately, he was able to work with the game section by tackling the Caesar Cipher and using some hints and skipping some questions.

Despite the challenges encountered, User A finished his session with a newfound understanding of the Caesar and Vigenère ciphers and an ignited interest in the basics of cryptography. This outcome is particularly encouraging as it aligns well with the core goals of our project, showing that the application has the potential to make cryptography approachable and intriguing for beginners. The use of hints and the ability to skip questions played a significant role in ensuring that User A could complete the quiz without feeling the need to quit, highlighting the success of these features in enhancing the learning experience.

4.2 User B

User B is a student currently enrolled in Computer Engineering. With some knowledge of cryptography and the basic knowledge of classical ciphers, their insights will be extremely valuable to our project.

Navigating through the application's interfaces was a smooth process for User B, yet he offered constructive feedback on enhancing visual elements, pointing out areas where alignment, font choices, and graphical consistency could be improved. This feedback is crucial as it highlights opportunities to refine the user interface for a more cohesive and aesthetically pleasing experience.

Upon engaging with the tutorial section, User B suggested that the Diffie-Hellman algorithm page could benefit from a more seamless transition into the algorithm and a more interactive component to enhance the learning experience His experience and background allowed him to navigate through the classical ciphers with ease, but he faced challenges in the final quiz due to the lack of cipher identification. Despite scoring well, he noted that the ambiguity of cipher types added a layer of complexity to the quiz.

Overall, User B’s insights were invaluable in pinpointing specific areas for improvement and enhancement in the application. His feedback also affirms the strengths of the application in catering to users with a basic understanding of cryptography.

In conclusion, the valuable feedback and insights provided by User A and User B during the testing phase of our application have been instrumental in identifying both the strengths and areas for improvement in our project. User A’s experience, coming from a non-technical background, highlighted the application's ability to make cryptography accessible and engaging to beginners, aligning well with our project’s goals. On the other hand, User B’s background in Computer Engineering and prior knowledge in cryptography provided a different perspective, emphasizing the need for improved visual consistency and a more interactive learning experience, particularly in complex topics like the Diffie-Hellman algorithm.

5. Conclusion

In summary, our project’s primary aim is to facilitate the introduction of cryptography to newcomers, adopting a dynamic and engaging learning methodology. Leveraging game-based educational strategies, we were able to simplify cryptographic concepts into interactable components to enhance the learning journey. From the user testing sessions, we were able to affirm the positive impact of our approach, particularly in motivating beginners and ensuring relevancy for those new to the field. Ultimately, our project successfully achieves its intended objectives and scope, establishing itself as an enjoyable and immersive educational tool in the realm of cryptography.

References

[1] F. Almeida, J. [1]Duarte Santos, and J. Augusto Monteiro, “The Challenges and Opportunities in the Digitalization of Companies in a Post-COVID-19 World,” *IEEE Eng. Manag. Rev.*, vol. 48, no. 3, pp. 97–103, 2020, doi: 10.1109/EMR.2020.3013206.

[2] B. Cashell, W. D. Jackson, M. Jickling, and B. Webel, “The Economic Impact of Cyber-Attacks”.

[3] Y. Acar, C. Stransky, D. Wermke, C. Weir, M. L. Mazurek, and S. Fahl, “Developers Need Support, Too: A Survey of Security Advice for Software Developers,” 2017, pp. 22–26. doi: 10.1109/SecDev.2017.17.

[4] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. CRC Press, 2020.

[5] E. B. Barker, “Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms,” National Institute of Standards and Technology, NIST SP 800-175B, Aug. 2016. doi: 10.6028/NIST.SP.800-175B.

[6] M. G. Voskoglou and A.-B. M. Salem, “Benefits and Limitations of the Artificial with Respect to the Traditional Learning of Mathematics,” *Mathematics*, vol. 8, no. 4, Art. no. 4, Apr. 2020, doi: 10.3390/math8040611.

[7] B. D. Ruben, “Simulations, Games, and Experience-Based Learning: The Quest for a New Paradigm for Teaching and Learning,” *Simul. Gaming*, vol. 30, no. 4, pp. 498–505, Dec. 1999, doi: 10.1177/104687819903000409.

[8] G. Jin, M. Tu, T.-H. Kim, J. Heffron, and J. White, “Evaluation of Game-Based Learning in Cybersecurity Education for High School Students,” *J. Educ. Learn. EduLearn*, vol. 12, no. 1, Art. no. 1, Feb. 2018, doi: 10.11591/edulearn.v12i1.7736.

[9] M. A. Khan, A. Merabet, S. Alkaabi, and H. E. Sayed, “Game-based learning platform to enhance cybersecurity education,” *Educ. Inf. Technol.*, vol. 27, no. 4, pp. 5153–5177, May 2022, doi: 10.1007/s10639-021-10807-6.

[10] Figure 1. Diffie-Hellman key exchange structure. - researchgate, https://www.researchgate.net/figure/Diffie-Hellman-key-exchange-structure_fig1_334819575.

[11] Maze, “The 6 key principles of Ui Design,” Maze, https://maze.co/collections/ux-ui-design/ui-design-principles.