

Animating Cryptographic Primitives

A dissertation submitted to The University of Manchester
for the degree of Master of Science in the Faculty of
Engineering and Physical Sciences

2017

Richard Lenin Leiva Atiaga

School of Computer Science

Contents

<i>List of Tables</i>	5
<i>List of Figures</i>	6
<i>Glossary</i>	9
<i>Abbreviations</i>	10
<i>Abstract</i>	11
<i>Declaration</i>	12
<i>Copyright</i>	13
Chapter 1: Introduction	14
1.1 Chapter Overview	14
1.2 Introduction	14
1.3 Project Aims	15
1.4 Project Objectives	15
1.5 Dissertation Outline	15
Chapter 2: Background and Theory	17
2.1 Chapter Overview	17
2.2 Cryptography	17
2.3 Classical Ciphers	19
2.4 Modern Ciphers	19
2.4.1 Symmetric Key Algorithms.....	19
2.4.1.1 AES	20
2.4.2 Asymmetric Key Algorithms.....	25
2.5 Animations in the teaching environment	26

2.6	Related Works	27
2.6.1	CrypTool.....	27
2.6.2	CryptoWorkflow.....	28
2.6.3	YouTube	29
2.7	Summary	29
Chapter 3: Research Methods		31
3.1	Chapter Overview	31
3.2	Website and Animation Technologies	31
3.2.1	3D Animation Tools.....	31
3.2.2	Flash (ActionScript 3.0)	32
3.2.3	HTML5, CSS3 and JavaScript	34
3.3	Evaluation Tools and Integrated Development Environments	35
3.3.1	Adobe Animate CC, Flash Develop and NetBeans IDEs	35
3.3.2	SWFScan.....	36
3.3.3	Adobe Scout CC.....	37
3.3.4	Browser Shots	39
3.4	Summary	39
Chapter 4: System Design and Implementation		41
4.1	Chapter Overview	41
4.2	Development Methodology	41
4.3	Design Considerations	43
4.4	Graphical User Interface	45
4.5	Audio files	47
4.6	Third Party Libraries	48
4.6.1	GSAP.....	48
4.6.2	AS3 Crypto.....	49
4.7	Implementation	50
4.7.1	Menus	51

4.7.2	Dynamic Screens	52
4.7.3	Static Screens	58
4.7.4	Size Optimisations.....	60
4.8	Summary.....	61
Chapter 5: Evaluation		62
5.1	Chapter Overview.....	62
5.2	Performance testing	62
5.3	Security testing.....	65
5.4	Compatibility testing	67
5.5	Comparison with Cryptool, CryptoWorkflow and YouTube.....	68
5.6	Questionnaire.....	69
5.7	Summary.....	81
Chapter 6: Conclusions and Future Work.....		82
6.1	Chapter Overview.....	82
6.2	Conclusions	82
6.3	Future Work	83
References.....		85
Appendix A: Important Code Snippets.....		92
Appendix B: Source Labs Results		96
Appendix C: Questionnaire		98

WORD COUNT: 17328

List of Tables

Table 2.1: AES Versions.....	20
Table 5.1: E-Learning Tools Comparison.....	69

List of Figures

Figure 2.1: Block representation of the plaintext.....	21
Figure 2.2: AES encryption and decryption structure.....	22
Figure 2.3: AES Substitute Bytes step	22
Figure 2.4: AES Shift rows step	23
Figure 2.5: AES Mix columns step.....	23
Figure 2.6: Simplified Key Expansion Algorithm	24
Figure 2.7: Screenshot of the AES animation in the CrypTool software	27
Figure 2.8: CryptoWorkflow screenshot.....	28
Figure 3.1: Movies created with Blender.....	32
Figure 3.2: SWFScan Example	37
Figure 3.3: Animate CC Screenshot	38
Figure 3.4: BrowserShots Screenshot.....	39
Figure 4.1: Steps involved in each iteration	42
Figure 4.2: Website Home Page.....	43
Figure 4.3: Use Case diagram for the AES animation	44
Figure 4.4: Navigation Menu	45
Figure 4.5: Animation Home Screen	46
Figure 4.6: Hover Image.....	47
Figure 4.7: Plugin registration in GSAP	49
Figure 4.8: Project Organisation	50
Figure 4.9: Type of Screens.....	51
Figure 4.10: Menus of the program.....	51
Figure 4.11: History and Structure.....	52
Figure 4.12: Sample Timeline code.....	52
Figure 4.13: Key Expansion	53
Figure 4.14: Encryption Animation Popup.....	54
Figure 4. 15: Pre-round animation (Encryption).....	55

Figure 4.16: Sub Bytes and Shift Rows.....	56
Figure 4. 17: Mix Columns and Add Round Key.....	56
Figure 4. 18: Final explanations of the encryption process.....	57
Figure 4. 19: Decryption.....	57
Figure 4.20: AES Solver	58
Figure 4.21: AES Solver Popup	59
Figure 4. 22: Evaluation	60
Figure 5. 1: History and Structure Analysis.....	63
Figure 5. 2: Key Expansion, Encryption and Decryption Animation Analysis	64
Figure 5.3: Initial security analysis	65
Figure 5. 4: Final security analysis.....	66
Figure 5.5: Browser Shots results	67
Figure 5.6: Question 1 Results	70
Figure 5.7: Question 2 Results	71
Figure 5.8: Question 3 Results	71
Figure 5.9: Question 4 Results	72
Figure 5.10: Question 5 Results	73
Figure 5.11: Question 6 Results	73
Figure 5.12: Question 7 Results	74
Figure 5.13: Question 8 Results	74
Figure 5.14: Question 9 Results	75
Figure 5.15: Question 10 Results	76
Figure 5.16: Question 11 Results	76
Figure 5.17: Question 12 Results	77
Figure 5.18: Question 13 Results	77
Figure 5.19: Question 14 Results	78
Figure 5.20: Question 15 Results	79
Figure 5.21: Question 16 Results	79
Figure 5.22: Question 17 Results	80
Figure 5.23: Question 18 Results	80
Figure A.1: Method used to sync the audio with the animation.....	92
Figure A.2: Methods that show the benefits of manipulating a master timeline instead of	

having to deal with individual timelines	92
Figure A.3: AS3 Crypto library code used to encrypt a block of code using the AES cipher ...	93
Figure A.4: Similar to A.3 but with changes to accommodate to the needs of the solver.....	93
Figure A.5: Main Menu Event Listeners.....	93
Figure A.6: Use of the 'to' method of the TimelineLite class.....	94
Figure A.7: Extract of the method used to create the reusable XOR animations in the whole program.....	95
Figure B.1: Internet Explorer 11 running in a Windows 8.1 Machine	96
Figure B.2: Microsoft Edge 15 running in a Windows 10 Machine	96
Figure B.3: Internet Explorer 11 running in a Windows 7 Machine	97
Figure C.1: Questionnaire, Part 1.....	98
Figure C.2: Questionnaire, Part 2.....	99
Figure C.3: Summary of the results - Question 1 to 6	100
Figure C.4: Summary of the results - Question 7 to 12	101
Figure C.5: Summary of the results - Question 13 to 18	102

Glossary

- Plaintext Text to be encrypted
- Ciphertext Encrypted text
- AES State Array Copy of the plaintext which is used in all the rounds of the algorithm. This array is set up as a 4x4 grid where each cell contains 1 byte.
- Small Web Format Executable Adobe Flash file format.

Abbreviations

- AES Advanced Encryption Standard
- DES Data Encryption Standard
- S-Box Substitution Box
- IDE Integrated development environment
- GUI Graphic User Interface
- MAC Message authentication codes
- UX User Experience
- UI User Interface
- AS ActionScript
- JS JavaScript
- CSS Cascading Style Sheet
- HTML HyperText Markup Language
- SWF Small Web Format
- W3C World Wide Web Consortium
- FPS Frames Per Second
- GUI Graphical User Interface
- NIST National Institute of Standards and Technology

Abstract

Cryptography is a fundamental part in the cyber security world. Nowadays, where millions of people surf the internet and perform various transactions online, cryptographic algorithms play a major role in assuring the security of the communications. One of these algorithms is the Advanced Encryption Standard which was approved by the National Institute of Standards and Technology as the replacement for the insecure DES cipher. Because cryptography encompass various fields like mathematics, networking, computer science, among others, it is usually hard to teach. Therefore, this dissertation focuses on providing an interactive and intuitive way of teaching the AES cipher. In order to achieve this, a didactic program that took into consideration various pedagogical factors to enhance the learning experience of the students was built using ActionScript 3. This program consists of an animation that was embedded in a website so that it was easily accessible to students. The AES animation comprises four parts that teach the student all the intrinsic aspects of the algorithm. The animation covers the history and structure of the algorithm, as well as the encryption, decryption and key expansion processes. Furthermore, users can play with the encryption and decryption of messages by inputting messages and keys, and seeing how the values change through the different rounds of the algorithm. Students are also able to self-assess their knowledge by completing an evaluation which allows them to auto-assess their progress. Finally, the compatibility, performance and security tests showed that the implementation of the program was appropriate. Additionally, a questionnaire that was filled out by the target group of this dissertation which are MSc. ACS students was performed and the results obtained exposed the teaching benefits of the animation and an exceptional user satisfaction rate.

Declaration

I confirm that no part of the work in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

- The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.
- Further information on the conditions under which disclosure, publication and exploitation of this dissertation, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University’s policy on presentation of Theses.

Chapter 1: Introduction

1.1 Chapter Overview

This chapter briefly introduces the importance of learning about security mechanisms in today's world and states the aims and objectives of the project.

1.2 Introduction

Cryptography has been around for thousands of years. It is the art of turning information from something readable and understandable to something that looks random and is incomprehensible unless the user possesses some secret information to decode it. Nowadays, in a digitalized world, cryptography is even more important than it was before. According to the United Nations, more than three billion people around the world access the internet (International Telecommunication Union, 2015). This puts a great pressure on security mechanisms because people need to feel and be safe while using the internet or performing any kind of electronic transaction. Therefore, these transactions need to be secure, not only by hiding the content of the transaction from unauthorized users, but also by assuring the receiver that the sender is who it claims to be (message authentication) and that the message has not been altered (integrity) (Katz & Lindell, 2015). Cryptography plays a big role in achieving this type of secure communications. These secure communications are needed in a variety of fields, for example: storage of medical records, communication between a self-driving vehicle and the GPS satellites, electronic purchases on the internet, transmission of information between a home security camera and the user's phone, among many others.

Usually trying to teach cryptography to people without any previous knowledge is challenging, especially, if the time span is short. Cryptography is not only a computer science subject, it is closely related with "number theory, abstract algebra, ..., probability statistics, and some knowledge of communications and network fundamentals" (Song & Deng, 2009) which makes it a complex subject to learn. For this reason, this project will focus on building an animation that aims to help students to understand how the symmetric key encryption

algorithm called Advanced Encryption Standard (AES) works.

1.3 Project Aims

The project aim is to create a piece of software that helps professors teach the Cryptography class at the Master's level in such a way that MSc. ACS students can learn about the AES cipher in a didactic and interesting way.

1.4 Project Objectives

The following objectives were proposed in order to achieve the aim of the project:

- Investigate the theory behind cryptography.
- Investigate about AES.
- Build a website that includes an interactive animation of the AES cipher, this animation consists of:
 - An explanation of the algorithm's history, its phases and how it works.
 - An explanation of the encryption, decryption, and key expansion processes.
 - A solver, which allows the user to input values and get the encrypted message, the user is also able to decrypt messages.
 - An evaluation to assess if the user understood the inner workings of the algorithm.
- Evaluate the security, compatibility and performance of the animation.
- Develop a questionnaire for MSc. Advanced Computer Science students.
- Evaluate and analyse the effectivity of the animation with the questionnaire.

1.5 Dissertation Outline

In order to achieve the objectives and aim of this dissertation, it has been divided into six chapters, all of which help to explain the AES cryptosystem and how the developed animation behaves and helps in the teaching of the algorithm.

- Chapter one gives a brief introduction to the security topic and states the aims and objectives of the project.

- Chapter two describes the cryptography world (its background and theory); it delves into the theoretical aspects of the symmetric and asymmetric key algorithms, and also analyses previous works that are related to the animation created for this project. This chapter also states the importance of using an animation to teach the AES cipher.
- Chapter three outlines the research methods; this chapter critically evaluates the technologies that could have been used and the technologies that were used to develop the website and the animation.
- Chapter four explains the design considerations, the third-party libraries that were used, the Graphic User Interface (GUI) and the technical considerations of the development.
- Chapter five contains the evaluations of the system; these evaluations comprise security, compatibility, performance and comprehensibility/effectivity factors.
- Chapter six contains the conclusions of the project and the future enhancements that the website/animation could implement.

Chapter 2: Background and Theory

2.1 Chapter Overview

Chapter two describes the cryptography world (its background and theory). This chapter delves into the theoretical aspects of the symmetric and asymmetric key algorithms, and also analyses previous works that are related to the software created for this project. This chapter helps to understand the theory behind the software created and how an animation helps in the teaching of the algorithm.

2.2 Cryptography

The Cambridge Dictionary defines cryptography as “the practice of creating and understanding codes that keep information secret” (Cryptography, n.d.). This concept is related to the classical problem of cryptography in which a person A wants to send a message to person B without letting an attacker C to be able to read the content of the message (Bellare, Desai, Jookipii, & Rogaway, 1997). Maintaining certain information as confidential as possible has been a necessity for centuries. Julius Caesar himself was aware of this, and that is why he started using a simple cryptographic algorithm to encrypt his messages (Dooley, 2013). However, over the years cryptography has evolved to not only tackle the confidentiality of information but also to handle data integrity, message authentication, among others (Al-Vahed & Sahhavi, 2011).

Hieroglyphics are considered the oldest form of encryption known to mankind, so it could be said that the history of cryptography goes back to Egypt in 3000 B.C (Singh, 2001). Even though the principle behind hieroglyphics might not have been keeping the messages secret, they were impossible to decipher until the 19th century (Thawte, 2013). In the current era, the Romans were avid users of cryptography. Sadly, cryptography was vanished from the Western world when the Roman empire fell in 476 (Dooley, 2013). Cryptography has evolved drastically since then and it has played a major role in the wars that the world has gone through. During the World War 1 (WW1), the decipherment of the Zimmerman Telegram

changed the outcome of the war since it was the breaking point that led the United States to attack Germany (McDonald, 2010). Not only the final result of WW1 was influenced by the decipherment of a cryptographic algorithm, the same thing happened in the Second World War. The breaking of the Enigma machine (used by the Germans to encrypt their messages) by the British gave a great advantage to the allied armies which ended up winning the war (Gaj & Orłowski, 2003).

As discussed above, cryptography has been around for a long time. However, its usage has never been as big as it is nowadays. This is because: “during this time ... the Internet provides essential communication between tens of millions of people and is being increasingly used as a tool for commerce, therefore, security becomes a tremendously important issue to deal with.” (Devi.T, 2013). This security issue can be handled with different features of cryptographic algorithms (Tripathi & Agrawal, 2014). Nowadays not only secrecy is important, that is why the goals of crypto algorithms have evolved through time. The main goal of cryptography is to provide: data integrity, confidentiality, authentication (Ayushi, 2010) and non-repudiation (Devi.T, 2013).

- Confidentiality: This is achieved with encryption. The whole purpose is to not let unauthorized users to read the contents of a message that is supposed to be kept secret.
- Data Integrity: This is achieved with hash functions or message authentication codes (MAC). The purpose is to give the possibility to check if the contents of the message have been altered or tampered with.
- Authentication: This is achieved with public key crypto systems or MACs. The purpose is to let the receiver know that the person that sends a message is who he claims to be.
- Non-Repudiation: This is achieved with digital signatures. The purpose is to not let a user to deny the transmission or reception of a message.

In order to understand how these goals are achieved, it is necessary to classify the ciphers. The cryptographic world can be divided into classical encryption techniques and modern cryptographic ciphers (public key, private key, hash functions) (Singh, 2003).

2.3 Classical Ciphers

Classical encryption techniques have been used for thousands of years, for instance, in the Roman empire Julius Caesar used a very simple substitution cipher later known as the Caesar's cipher. This cipher basically substitutes the letters in the plaintext with subsequent letters in the alphabet, where the number of shifts correspond to the private key (Rocca, 2014). People then started using a similar approach but instead of having only one alphabet they started using multiple alphabets; these are known as polyalphabetic ciphers. All these ciphers that rely only in the substitution of letters are known as substitution ciphers. Other classical ciphers that rely on the change of position of the letters are known as transposition ciphers. However, none of these ciphers will be discussed in this dissertation because its focus lies on the AES cipher, which is part of the modern ciphers. As Stallings mentions, all the classical ciphers would now be part of the private key ciphers because they only use one key (2014).

2.4 Modern Ciphers

The modern cryptography field is composed by the symmetric ciphers also known as private key ciphers, asymmetric ciphers also called public key ciphers (Singh, 2003) and data integrity algorithms (Stallings, 2014). The asymmetric ciphers appeared on the 1970s (McDonald, 2010) and they make use of two keys: one public and a one private. The private key is used to decrypt messages while the public key is used to encrypt information. Asymmetric ciphers use trapdoor one way functions which are mathematical functions that are easy to calculate in one way (encryption) and hard to calculate the other way around (decryption) unless extra information is used (key) (Catalano, Fiore, Gennaro, & Vamvourellis, 2015). To the contrary, symmetric ciphers only use one key to encrypt and decrypt the messages, that is why this key should be kept secret and should only be known to the sender and the receiver.

2.4.1 Symmetric Key Algorithms

Symmetric key ciphers were the only type of encryption algorithms available before the creation of asymmetric ciphers (Stallings, 2014). The key characteristic of these algorithms is that they only use one key to encrypt and decrypt messages, however, not all the algorithms

process the data in the same way; the data can be processed by blocks or by streams (Sharif & Mansoor, 2010). Block ciphers process a group of bits at a time while stream ciphers process only one bit at a time (Singhal & J.P.S.Raina, 2011).

Private key algorithms are very important because they tend to be faster than asymmetric ciphers, this is because the latter algorithms use computationally expensive mathematical operations to encrypt the data (Minaam, Abdual-Kader, & Hadhoud, 2010). Because of this, symmetric ciphers are usually preferred over asymmetric ciphers when the amount of resources of the device is limited like in mobile devices (Elminaam, Kader, & Hadhoud, 2010). Typical examples of symmetric ciphers are DES, RC4, AES, among others. However, AES is the only cipher explained in detail because that is the core algorithm this dissertation is based on.

2.4.1.1 AES

This is an iterative block cipher that relies on the use of substitutions and permutations over a certain number of rounds to encrypt the plaintext and decrypt the ciphertext (Stallings, 2014). AES takes blocks of 128 bits of plaintext and converts them into 128 bits blocks of ciphertext (128 bits is equal to 16 bytes or 4 words). The AES cipher, also known as Rijndael, was invented in Belgium by Joan Daemen and Vincent Rijmen. Rijndael was born as the replacement cipher for DES because the latter was no longer secure due to its short key size. AES consists of 3 versions: AES-128, AES-192, AES-256, all of them perform the same operations but the key sizes and the number of rounds vary (Biryukov, Dunkelman, Keller, Khovratovich, & Shamir, 2009).

	AES-128	AES-192	AES-256
Key Size	128 bits	128 bits	128 bits
Plaintext Block Size	128 bits	128 bits	128 bits
Number of Rounds	10	12	14

Table 2.1: AES Versions

Adapted From: (Stallings, 2014)

This table shows the differences between the versions of the AES cipher, even though all the versions deal with 128 bit blocks, they use different key sizes and different number of rounds.

As it can be seen in Table 1, the difference between the AES versions is subtle. Therefore, only the AES-128bit version consisting of 10 rounds will be analysed, which can be used to understand any other AES version. Before starting, it is necessary to transform the plaintext to a 4x4 byte grid structure. For instance, if the plaintext is “This is a test” which consists of 14 bytes, it is necessary to pad the message with empty spaces so that it has 16 bytes (128 bits), so the plaintext would end up being “This is a test ” (two empty spaces at the end). The plaintext example represented in a grid structure is depicted in the following figure:

String				Hex			
T		a	s	54	20	61	73
h	i		t	68	69	20	74
i	s	t		69	73	74	20
s		e		73	20	65	20

Figure 2.1: Block representation of the plaintext

Adapted From: (Selent, 2010)

The figure shows the grid representation for the plaintext “This is a test ” with its hexadecimal representation (also known as the state). Each cell in the grids is one byte, which is equivalent to 8 bits. The only purpose of working with a grid like structure is to facilitate the understanding of the algorithm.

The encryption process consists of a pre-round + 10 rounds, where the pre-round only consists of an XOR operation between the original key and the state. Each round has 4 steps:

1. Substitute bytes.
2. Shift rows.
3. Mix columns.
4. Add round key.

Rounds 1 to 9 are the same, however, round 10 is slightly different because it does not make use of the Mix Columns step. This can be seen in the following figure:

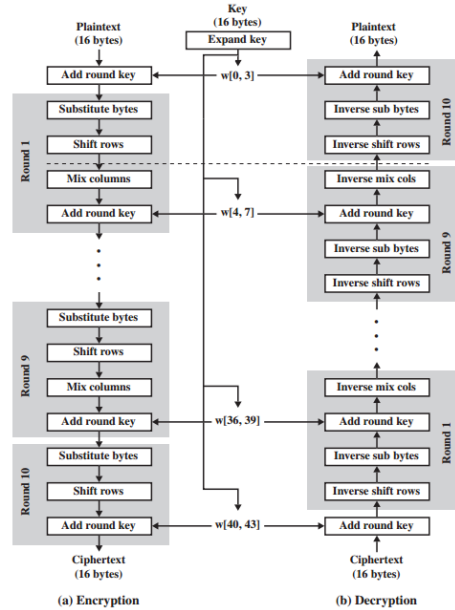


Figure 2.2: AES encryption and decryption structure

Source: (Stallings, 2014)

This figure shows the structure of the AES algorithm for both the encryption and decryption processes. In both encryption and decryption rounds 1 to 9 are the same but round 10 is different because it does not perform the Mix Columns step.

The Substitute bytes round is a simple substitution using the AES Substitution box (S-Box). This box serves as a lookup table to simply substitute the state array bytes for the bytes that exist in the S-Box (Granado-Criado, Vega-Rodriguez, Sanchez-Perez, & Gomez-Pulido, 2010). The substitution is done byte by byte, where the first 4 bits represent the row and the last 4 bits represent the column in the S-Box. This is the step that provides confusion in the algorithm (Fahmy, Shaarawy, El-Hadad, Salama, & Hassanain, 2005) and can be seen in Figure 2.

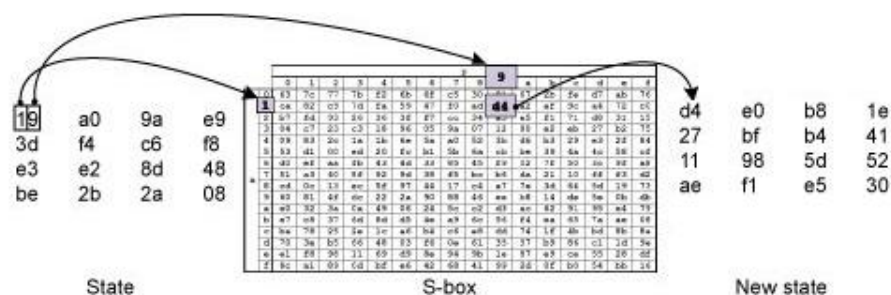


Figure 2.3: AES Substitute Bytes step

Source: (Masood, 2012)

This figure shows how the bytes in the state matrix are substituted using the S-Box

The shift rows step is simple, it consists of circular left byte shifts of the state matrix. The first column row remains intact, the second column has 1 left byte shift, the third column has 2 left byte shifts and the fourth column has 3 left byte shifts as can be seen in figure 3. This step provides the diffusion property (Fahmy, Shaarawy, El-Hadad, Salama, & Hassanain, 2005).

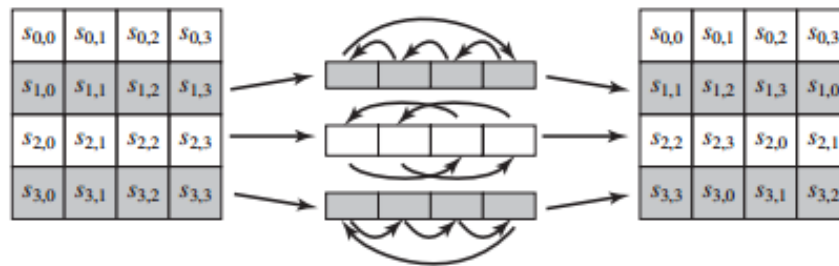


Figure 2.4: AES Shift rows step

Source: (Stallings, 2014)

This figure shows the shifts of each of the rows. One shift for row two, two shifts for row three and three shifts for row four.

The Mix columns step is a matrix multiplication in the Galois Field 2^8 of the state array with the mix columns matrix (Chitu & Glesner, 2005); this matrix is a constant that was designed to provide maximal diffusion properties. The following figure shows the matrix:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} :$$

Figure 2.5: AES Mix columns step

Source: (Stallings, 2014)

This figure shows the mix columns matrix on the left side and the state matrix on the right side. The output of this step is the multiplication of both matrixes in the Galois Field 2^8 .

The add round key step is a XOR operation between the state array and the round key. This is the only phase in a round where a key is used.

The decryption is slightly different to the encryption. The decryption process follows

the same steps, however, the order of the operations in each round change (Stallings, 2014):

- 1st: Inv Mix Columns - The inv mix columns step uses the inverse of the matrix used for the encryption process.
- 2nd: Add round key - The add round key is the same step for encryption and decryption, because the inverse of an XOR operation is the XOR operation itself. However, the sub keys are used in the inverse order.
- 3rd: Inv Sub Bytes - For the inv substitute bytes phase, the S-Box used is an inverse of the original S-Box.
- 4th: Inv Shift Rows - The inv shift rows phase instead of performing left shifts performs right shifts.

As it was mentioned earlier, each encryption/decryption round uses a different round key. These sub keys are generated using a key expansion algorithm that takes the original key as input. The key expansion algorithm is shown in the following figure:

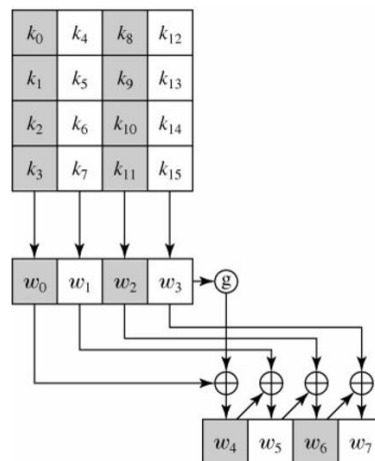


Figure 2.6: Simplified Key Expansion Algorithm

Source: (Stallings, 2014)

This figure shows how the key expansion algorithm works. To obtain the words that are multiple of four, the previous word has to go through a G function, however, to obtain the words that are not multiple of four a simple XOR operation is necessary. This figure only shows the first 7 words for simplicity purposes, but the algorithm produces 44 words in total.

The purpose of the key expansion algorithm is to produce a different round key for each of the rounds of the encryption process (the decryption process uses the same keys in inverse order). For this reason, the algorithm takes the original key as input and creates 10 more keys

(44 words in total). The following steps that explain how the key expansion algorithm works were adapted from Mangard (2002) and Stallings (2014):

- The original key is converted to its hexadecimal grid representation (like the plaintext in Figure 1).
- Each column in the key grid is treated as 1 word (16 bytes). To obtain the words that are multiple of four ($w_4, w_8, w_{12}, \dots, w_{40}$), the previous word goes through a function G , which does the following:
 - 1 byte circular left shift, where the first byte in the word is moved to the position where the last byte is, and every other byte is moved to the position of the previous byte (Similar to the one byte circular shift in Figure 4).
 - Each byte in the word is substituted using the S-Box.
 - The word is XORed with a constant called RCON. There is a unique RCON value for each word.
 - The word is XORed with the word 4 positions before (e.g if we are obtaining w_8 , we have to XOR it with w_4)
- To obtain the words that are not multiple of four ($w_5, w_6, w_7, w_9, \dots, w_{43}$) an XOR operation between the previous word and the word four positions before is necessary (e.g to obtain w_5 we need to XOR w_4 with w_1).

2.4.2 Asymmetric Key Algorithms

The main difference between asymmetric and symmetric key algorithms is that asymmetric ciphers use two keys (private and public). The issue with having only one key to encrypt/decrypt data is that the key should be transmitted between the sender and the receiver of the message in a secure way. This problem is solved by using asymmetric key ciphers in which the public key (available to everyone) is used to encrypt messages and the private key (only available to the intended receiver) is used to decrypt the message (Kumar, Munjal, & Sharma, 2011). This type of encryption is usually used to securely send the private key for a symmetric cipher and then a symmetric encryption scheme is used to encrypt the message (because of the performance benefits) (Tripathi & Agrawal, 2014). That is the reason why Stallings (2014) mentions that public key cryptography complements private key

cryptography instead of replacing it. Asymmetric ciphers also play a fundamental role in the digital signature world because a user is able to encrypt a message with his private key (digital signature creation) and then every user in possession of the public key can verify the digital signature. Examples of asymmetric encryption algorithms are El Gamal, RSA, among others.

2.5 Animations in the teaching environment

Since the goal of this dissertation is to build an animation that helps to teach the AES cipher, the importance of animations in the teaching environment will be analysed. There have been several studies that state the importance of animations in the learning process, some of these researches mention that animations are useful while others mention they are not (Höffler, Prechtel, & Nerdel, 2010). Nonetheless, some of the arguments that support the view that animations are not helpful are based on assumptions that do not necessarily apply to this project. For instance, Ainsworth and VanLabeke (2004) mention that animations do not provide permanent information. However, with the facilities provided in the developed animation, the user can freely navigate through the animation going back and forth to any content that the user wants to revisit.

Even though the authors Tversky, Morrison, & Betrancourt mentioned that in some cases the use of animations is not justified since the learning outcomes are not different from the ones gotten with normal presentations (2002), some studies have taken the research further. These researches not only analyse the quality of the teaching process by using animations, but they also consider the capacity of students to retain the information that was taught. One of these researches show that the information was better retained by students that used animations instead of static content (O'Day, 2007). Also, another study confirms that complicated topics can be more easily taught by using visual aids like animations (Janitor, Jakab, & Kniewald, 2010), description which exactly shows the necessity to build an animation to teach an advanced topic such as cryptography.

Despite the fact that some researches argue about the utility of animations as teaching tools, the audience of the animations has not been analysed. The target audience of the developed animation are MSc. ACS students, and as O'Day (2007) states, today's students prefer animations over textbooks because they have grown up in an electronic world. Also, after analysing the questionnaires filled out by the students there is no doubt that the use of

the AES animation was useful to them (refer to Chapter 5).

2.6 Related Works

Some software applications have already been developed to tackle the issue of helping people to better understand cryptographic algorithms. Two of these programs are CrypTool and CryptoWorkflow, however, students nowadays find that YouTube has vast teaching material as well. As Roodt & Peier (2013) said, “it is believed that the use of YouTube as a teaching tool could have an effect on the level of student engagement”. Today’s students “consider Web-based technologies integral to the information gathering process” (Buzzetto-More, 2015). That is why it is important to analyse YouTube as well as the other two cryptographic-teaching programs mentioned before in order to assess the effectiveness of the developed animation.

2.6.1 CrypTool

CrypTool is a free e-learning tool that helps people to understand how cryptography works. This open source program built using C++ is only available on Windows and has around 60 volunteers that work on it regularly (Hick, Esslinger, & Wacker, 2012). CrypTool has a wide variety of algorithms to play with, it supports classic and modern cryptographic ciphers. This program allows the users to test several algorithms and analyse the results obtained after running simulations (Loussios, 2014). As Kaushik & Singhal (2012) mention, this is not only a teaching tool, it can also be used as a simulator to run different experiments. Figure 7 shows a screenshot of the AES animation in the CrypTool software.

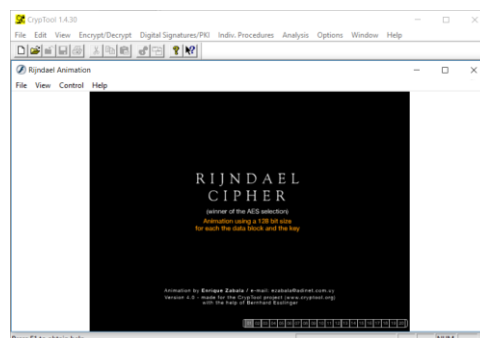


Figure 2.7: Screenshot of the AES animation in the CrypTool software

This screenshot shows the AES animation inside the CrypTool Software. As it can be seen on the top of the screen, the software has a variety of options (encryption, decryption, digital signatures, etc.) that help the user to understand cryptographic algorithms.

CrypTool has an animation that explains the intrinsic works of the algorithm and it also has an Inspector which allows the user to input values and see how they change through the different encryption rounds of the algorithm. The animation of the algorithm delves into the encryption and key expansion processes, however, the decryption and the structure of the algorithm itself are not explained in detail which weaken the teaching ability of this tool.

2.6.2 CryptoWorkflow

CryptoWorkflow “is a tool for learning the intricacies of cryptographic algorithms” (CryptoWorkflow, n.d.) and can be run in multiple Operative Systems (Windows, Linux, OSX). CryptoWorkflow is a much simpler program than CrypTool; out of the box it only supports three algorithms which are DES, Vigenere and AES. The AES version presented in this application is a simplified version that only allows the user to see an example of how the algorithm works (it is not a complete implementation). As it was mentioned earlier, AES has three versions consisting of different numbers of rounds, however, this program uses a simplified version of the AES-128 that consists of only 3 rounds. The simplified version can be seen in the following figure:

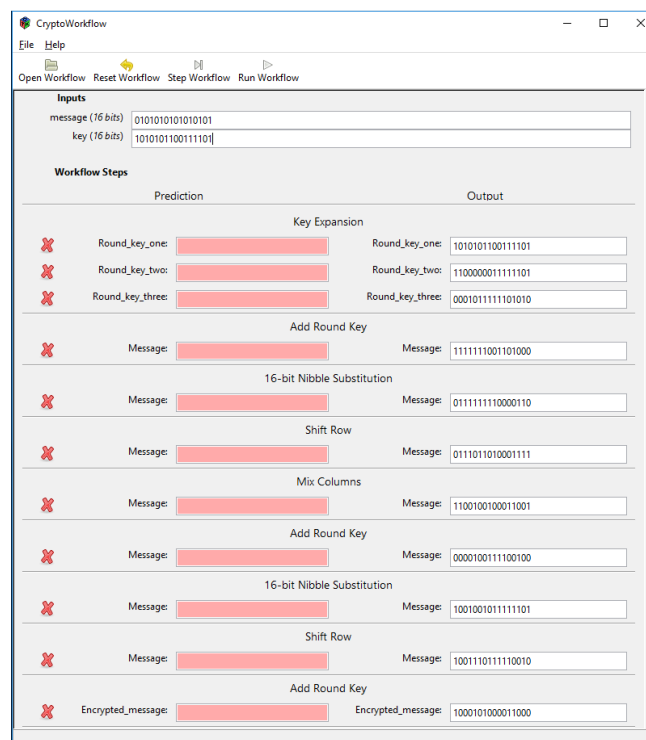


Figure 2.8: CryptoWorkflow screenshot

This figure displays a screenshot of the CryptoWorkflow tool, specifically, of the simplified AES-128 version consisting of only 3 rounds. This figure shows that none of the user predictions were correct.

This software claims to help its users learn how cryptography works, however, it does not explain how the ciphers work. It appears that the sole purpose of this software is to let the user predict values and then compare them with the correct values after each round of the AES encryption process (if the prediction is correct a green box will be displayed, if not, a red box will show up as in Figure 8). Not only the lack of theoretical explanation is the problem, but based on the screenshot in Figure 8, the user interface (UI) is not friendly. The UI is not friendly not only because the user needs to input binary values to start the process (which make it confusing for people without computer science background), but also because it is confusing for the user to understand how to interact with the application.

2.6.3 YouTube

YouTube was created in 2005 and is the most famous platform for sharing videos on the internet (Lee & Lehto, 2013). As it is mentioned by Duffy (2008) “YouTube is increasingly being used by educators as a pedagogic resource”. This platform has hundreds of videos related to cryptography. One of the most famous cryptography channels is Christof’s Paar channel who has around 17 thousand subscribers. Another channel that publish a significant amount of content related to cryptography is the Khan Academy channel, with more than 3 million subscribers. These two channels teach how cryptographic algorithms work and how they are used, however, since they are videos they lack the interaction with the user. Interaction is important in online learning because “interaction is seen as central to an educational experience” (Garrison & Cleveland-Innes, 2005). YouTube channels provide in depth theoretical knowledge about the AES cipher, however, the videos are lengthy (usually more than one hour) and the user is not able to auto-assess his learning.

2.7 Summary

This chapter briefly explained the history of cryptography and the usefulness of it. The AES algorithm, which is the core algorithm of this dissertation, was explained in detail giving the reader a good understanding of what the animation work is based on. Some related works were also analysed: on the one hand, the interesting part about CrypTool and CryptoWorkflow is that the user can see step by step how the algorithm works, but they lack the theoretical part of the teaching process. On the other hand, the interesting part about the YouTube videos is that they provide strong theoretical knowledge but they lack the

interaction with the user. For this reason, the animation created for this dissertation has both, the theoretical teaching and the user interaction which enhance the learning process of the student.

Chapter 3: Research Methods

3.1 Chapter Overview

This chapter critically evaluates the technologies that could have been used and the technologies that were used to develop the website and the animation. The decision on which technologies were better for this kind of project was made based on multiple factors that are explained throughout this chapter.

3.2 Website and Animation Technologies

In today's world, teaching cryptography with blackboards is not enough if better resources can be used. Animations are important resources to educators because they improve the quality of the education process (Robling, Schuler, & Freisleben, 200). There is a vast amount of animation tools available, however, most of these tools are complex and difficult to learn (Davis, Colwell, & Landay, 2008). Because of this, the learning curves of the tools were considered before deciding which technologies were going to be used. The tools that were analysed were 3D animation programs, Adobe Flash and web solutions.

3.2.1 3D Animation Tools

Three-dimensional animation tools are very popular for modelling and animating virtual objects. Some of the tools used to create 3D animations are Blender, Maya, among others. Proprietary tools like Maya tend to be expensive, according to its web page, the price is around £200 per month. Because the price of the animation tools is a constraint for this project, this type of tools was not considered to be suitable. However, the 3D tool "Blender" is free so the feasibility of using that software was analysed.

Blender is a free software that allows to create models and animations in three dimensions. It is an open source tool that runs in Windows, Linux and OSX, and relies on the support of hundreds of people that help to develop the software (Blender, n.d.). Blender has a variety of features, it allows users to: model, rig, simulate, animate, video edit, create

games, among others. Because of all these features, Blender is a complete animation suite and a very big program; which means that it requires a lot of time to master and a lot of time for new users to be able to create good quality animations (Flavell, 2010).



Figure 3.1: Movies created with Blender

Source: www.blender.org/features/projects/

This figure shows posters with the advertisements of some movies created with the Blender Software.

Blender is a powerful software which has been used to create many movies and games (see Figure 9). Even though blender is a very useful tool, it is known to have a “steep learning curve” (Flavell, 2010) which renders it difficult to use for this project that has time constraints. In the end, the final decision was to not use 3D animation tools because of the price restriction and the difficulty of learning how the tools work in a short period of time.

Besides of the previously mentioned limitations, it is important to mention that another reason why 3D tools were not used was because this project focuses on animating the AES cipher, which can be better explained by using a two-dimensional animation. Implementing a 3D animation for this type of algorithms is an overkill, and would increase not only the complexity of the animation but would also make it harder for the students to understand the concepts of the algorithm. By implementing a 2D animation, the observer can concentrate on understanding the concepts that really matter instead of losing focus because of the unnecessary spatial animations that are achieved with 3D animations.

3.2.2 Flash (ActionScript 3.0)

Macromedia was the former owner of Flash, however, in 2005 Adobe decided to buy the company and all the products associated with it. Adobe Flash is a technology that allows developers to create dynamic content for the web such as: animations, games,

advertisements, among others. It uses vector and raster images to build animations (Russel, 2006) which can be run by special programs (Adobe Animate, Swiff Player, etc.), within a web browser with the Flash plug-in, and as a standalone executable file. Flash has been known to have security issues that allow remote attackers to access a victim's computer, nevertheless, adobe tries to patch the bugs as soon as possible. Despite of Adobe's best efforts, some companies prefer to use HTML5 instead of Flash because of security concerns. For instance, Apple mobile devices do not provide support for flash websites or applets, however, Phung, Monshizadeh, Sridhar, Hamlen, & Venkatakrishnan (2015) mention that around 23% of all the websites in the internet still use Flash. Even though 23% is not a high percentage, it is worth mentioning that big companies like CNN, Spotify, Hulu, HP, Cisco, among others, still use flash for some of the features they offer. Also, 24 out of the top 25 games on Facebook are developed using Flash technologies (Adobe, n.d.). The ease to learn how the Flash tools and IDEs work makes it a good technology for short animations and games. According to Adobe, more than three million developers use Flash technologies to create web content (Adobe, n.d.).

ActionScript (AS) is the object-oriented language used to build Flash applications and its smooth learning curve and powerful animation frameworks is what made this technology interesting for the development of the animation for this project. As Reimers and Stewart (2015) mention, the advantages of using flash are that it is still a commonly used technology, it can be used to implement complex programs and that it produces cross-platform animations specifically built for web environments. Because of these reasons, Flash was the chosen technology to build the AES animation.

The animation was developed using some libraries that helped to move and transform the objects on the stage, and another library that assisted with the AES encryption and decryption processes. The animation library that was used is called GSAP which is a famous JavaScript (JS) and ActionScript library used to build simple and complex animations. The company that owns the library is GreenSock, which has companies like Adobe, YouTube, Ford, Samsung, Google, McDonald's, among others, within its customer base. On the other hand, the library that was used for the encryption and decryption processes is called As3 Crypto, which according to his author "is a cryptography library written in ActionScript 3 that provides several common algorithms" (Hurlant, n.d.). One of the algorithms provided by this library is

AES, however, multiple tweaks were necessary to be able to achieve the results needed for the animation (refer to Chapter 4).

3.2.3 HTML5, CSS3 and JavaScript

The developed website uses HTML5 as the markup language, CSS3 for the styling of the site and JavaScript to make the webpages dynamic. These technologies were chosen because they are the mainstream technologies on web site development that allow to create eye-catching sites and embed Flash animations in an easy way.

HTML is the acronym for HyperText Markup Language which is the standard for website development. HTML5 is the latest version of this standard and is maintained by the World Wide Web Consortium (W3C) (Anthes, 2012). The goal of HTML5 is to move away from proprietary technologies such as Adobe Flash, Quicktime, among others (Vaughan-Nichols, 2010). Before HTML5, proprietary software gained a lot of popularity because HTML was not flexible enough, for instance, webpages were forced to use third party extensions to play audio and video (Reimers & Stewart, 2015). HTML was used as a text presenting technology; however, its latest version provides developers with great resources to build complex websites and animations with the help of CSS and JavaScript. The main benefit of HTML5 is that it is open source and the contents (static or animated) created with that technology do not need anything else than a web browser to be accessed.

HTML sites need to be styled so that they look appealing to the users which is why Cascading Style Sheets (CSS) are needed. The latest version is CSS3 which is a standard maintained by the W3C. CSS allows developers to position, animate, and transform elements in a website, giving developers the freedom to explore their imagination. Having dynamic content on a website betters the user experience, therefore, the use of JavaScript is necessary. JS is a weakly typed language used to make the content in a website dynamic. JavaScript, with the help of the Canvas HTML5 tag, provide developers with a powerful tool to “incorporate graphics, video, and animations” (Vaughan-Nichols, 2010) in a website without the need to use third party plugins. However, when building animations with HTML5, CSS and JS usually not all the animations are interpreted in the same way by every browser, so animations can end up looking and behaving differently in different browsers. That is why one of the advantages of working with Flash is that developers know that the animation will

look the same independently of the browser the animation is played with.

3.3 Evaluation Tools and Integrated Development Environments

In order to build the Flash animation and the website, the use of the Adobe Animate CC, Flash Develop and NetBeans IDEs was necessary. These IDEs facilitated the coding because they provide autocomplete features, code snippets, templates, among other features that were useful while developing the final product for this dissertation. Also, in order to assess the quality of the animation the SWFScan and Adobe Scout tools were used. These tools helped to assess the security and the performance of the animation; moreover, the assessment of the website was done with a tool called Browser Shots which helped to determine compatibility issues among different browsers and different browser versions.

3.3.1 Adobe Animate CC, Flash Develop and NetBeans IDEs

The Adobe Animate CC and Flash Develop IDEs were used to create the animation, while the NetBeans IDE was used to deal with the website. The creation of the visual elements and objects were done with the Animate CC platform and the code was written using the Flash Develop IDE. Even though Animate CC allows to modify the ActionScript code within its own editor, the separation between the visual elements and the code was necessary because each of the IDEs previously mentioned possess different strengths. For instance, FlashDevelop has a great auto completion feature that Animate CC lacks, however, FlashDevelop does not offer the possibility to visually see which elements are on the stage which can be done using Animate CC.

Adobe Animate CC is the official IDE to develop Flash applications. This is a proprietary animation suite that costs £20 per month which is an affordable price for this project. According to Adobe, Animate CC is the “leading vector animation toolset” (Adobe Animate CC, n.d.), which is true because the tools and the features that the suite possess are incomparable. This tool allows users to work with different technologies like HTML5, OpenGL, Flash, among others. When working with Flash, Animate CC lets users to visually manipulate, animate and transform objects on the stage. Animate CC works with frames and layers, giving the developer the flexibility to create simple and complex animations as well as the ability to

modify vector objects. The debugger in this tool permits the developers to take control of the animation step by step, which is useful when an error occurs in the middle of an animation and the developer has no idea about what is wrong.

Flash Develop is a free open source software that allows to manipulate Flash (.fla) files. This IDE contains autocompletion features that save a lot of time while coding in ActionScript 3. Instead of having to type complete sentences or method calls, the IDE guesses what the developers are going to type and complete the sentence for them. Another important feature is the pre-set code snippets it offers, it saves time when the developer wants to add the code to create a new function or class, and when trying to add the code to subscribe to an event (e.g. `MouseEvent`).

NetBeans is an IDE developed in Java and is one of the most used IDEs for Java development. In 2010 Netbeans became part of the Oracle family after the purchase of Sun Microsystems. Besides being a free Java IDE, this tool also offers good support for web development which involve the creation/edition of HTML, CSS and JS files. As any other modern IDEs, it has an autocomplete feature that helps developers to code faster and without syntax errors. An interesting feature is that it has a plugin for most modern browsers (Edge, Mozilla, Chrome, etc.) that, together with the IDE, allows the user to profile and debug errors in the webpage being developed.

3.3.2 SWFScan

For the vulnerability assessment of the application the SWFScan application was used. This application was developed by Hewlett Packard, and even though it is no longer supported by that organization, it is still a great tool to detect security flaws in a Flash application because it “is a tool that decompiles a Flash application and performs static analysis to detect possible vulnerabilities.” (Acker, Nikiforakis, Desmet, Joosen, & Piessens, 2012). The cause of these vulnerabilities is usually poor coding practices and the purpose of using this program is to not let attackers take advantage of those coding errors.

Figure 10 shows how SWFScan works with an initial version of the developed animation. Most of the vulnerabilities that were found in that version were path disclosure vulnerabilities which can be fixed by removing the path strings created by the debugging tool of Adobe

Animate CC. One important feature of this security application is that besides giving the user information about the vulnerability it tells him what the possible fix is.

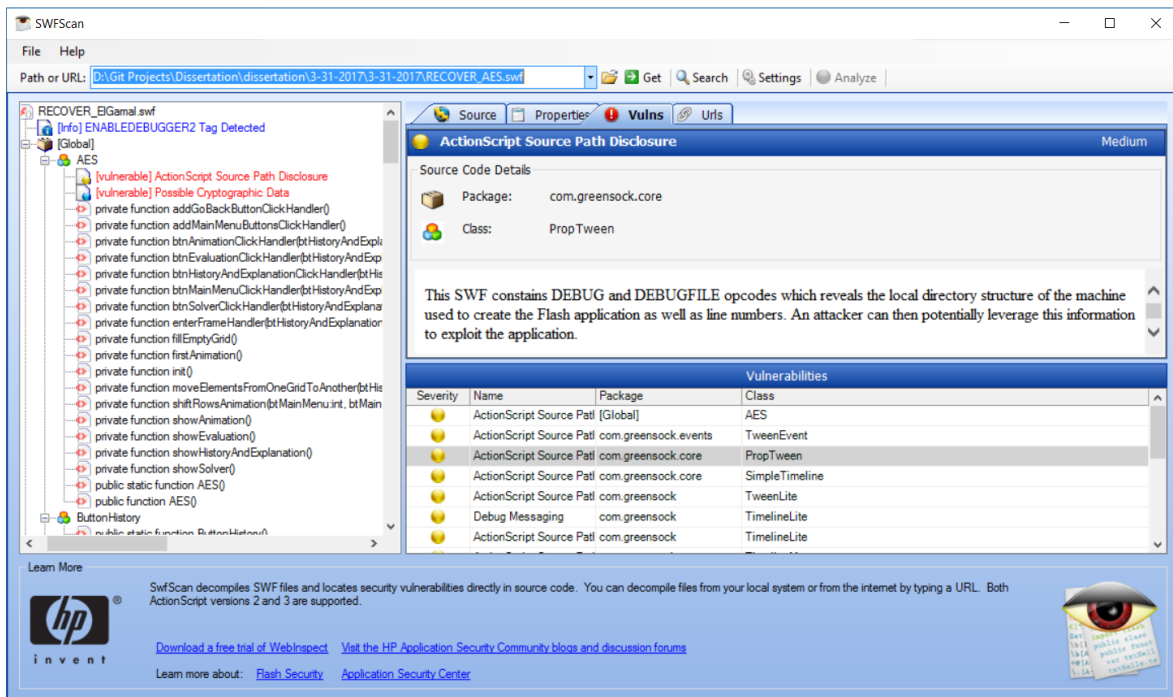


Figure 3.2: SWFScan Example

Screenshot of the vulnerabilities found in an initial version of the animation.

The UI of the program is simple, the user only needs to enter the location of the SWF file on the “Path or URL” field and click on the “Get” button, after the animation has been loaded, the “Analyze” button needs to be pressed. Once the analysis finishes, a detailed list with the vulnerabilities and possible fixes are shown to the user.

3.3.3 Adobe Scout CC

To analyse the performance of the application the Adobe Scout CC program was used. Scout CC is a SWF profiling tool that allows to detect issues that may cause performance malfunctions (Adobe, 2017). It provides memory analysis, CPU usage, network usage, method call examination, among others; all of them on a frame by frame basis. By using this application, developers can detect any bottlenecks in the animations and fix them.

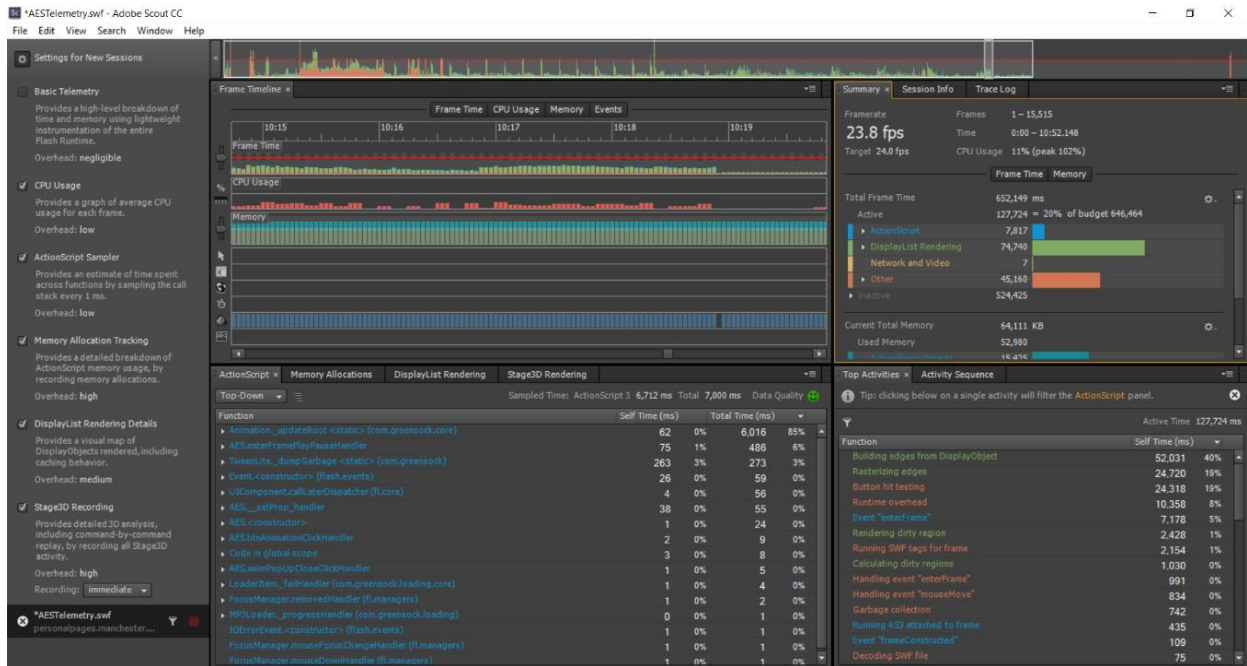


Figure 3.3: Animate CC Screenshot

This screenshot was taken after running the AES Encryption animation. As shown in the Figure, Animate CC provide a lot of information that can be used by developers to profile an animation and fix performance issues if necessary.

The UI of Scout CC is easy to understand. To start profiling, the user has to play a Flash animation and Scout CC will load it automatically and will start displaying information to the user in real time while the animation is running. Figure 11 shows how the AES Encryption animation performed in the Adobe Animate CC tool. The top part of the UI shows the timeline of the animation on a frame by frame basis. The red line on the timeline is the most important thing to consider because it represents the budget time, any frame that surpasses the budget time may cause stuttering and lower the performance of the animation (Adobe, 2017). On the top right corner, a summary of the whole animation is displayed that shows the average CPU usage, the desired/achieved frames per second (FPS) and which activities (Code execution, Object rendering, Network and Video, Other overhead) took the most time. The same information that appears in the timeline appears in the centre of the screen, the difference is that the information is organized in such a way that the developer has more control over what he wants to profile. The bottom part of the screen displays the method calls and activities for a certain frame or groups of frames, letting the user know which activities are taking the most time. It is important to mention that to be able to obtain detailed information like the one in

Figure 11, the animation must be published with the advanced telemetry option enabled in Adobe Animate CC, otherwise, only details about the CPU usage will be displayed (Adobe, 2017).

3.3.4 Browser Shots

Browser Shots is an online free tool used to test the compatibility of a website across several browsers. This tool tests a given website against more than 150 browser versions in Mac, Linux and Windows machines. The main browsers are: Chrome, Firefox, Safari, Opera, Lynx, Epiphany, ELinks, Konqueror, Luakit, among others. To use this tool, the user has to input the url of the website to be analysed, and select the desired browsers. The tool then displays a series of screenshots of each of the selected browsers so that the user can see if there are any compatibility issues, Figure 12 shows an example of how the tool works.

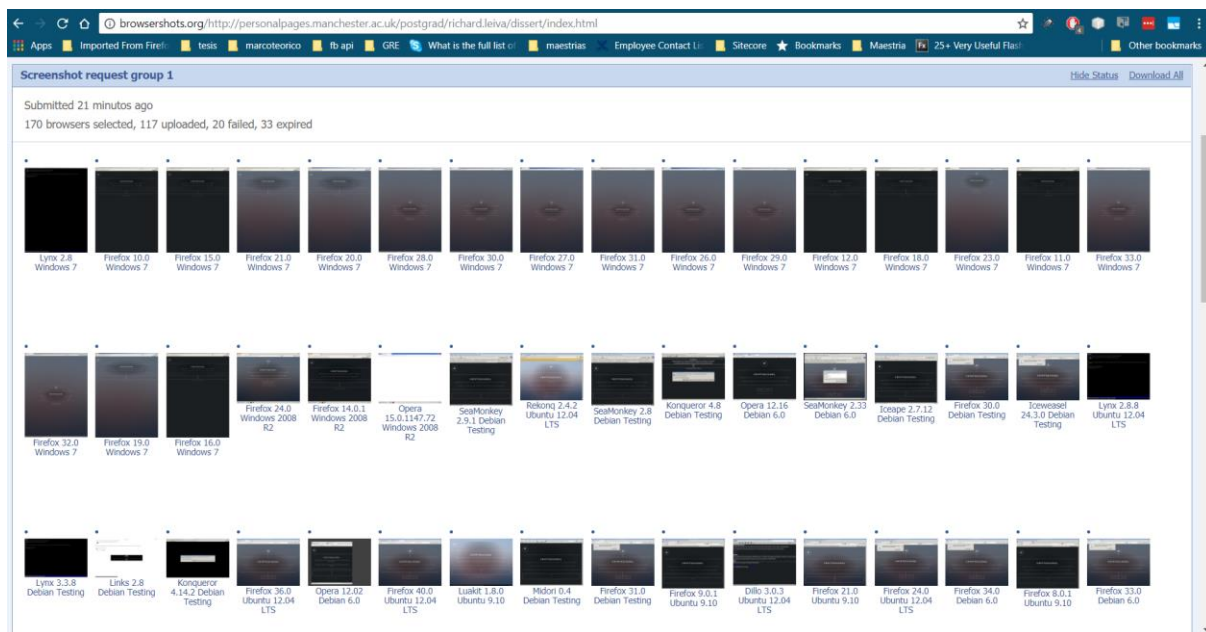


Figure 3.4: BrowserShots Screenshot

This Figure show several screenshots of <http://personalpages.manchester.ac.uk/postgrad/richard.leiva/dissert/index.html>.

3.4 Summary

This chapter explained the technologies that were used to develop and assess the developed product for this dissertation. Because of the factors explained in this chapter, the final decision was to create the animation using Adobe Flash. In order to create the animation,

the Adobe Animate CC and the Flash Develop IDEs were used, and the security assessment and performance tests were done using SWFScan and Adobe Scout CC respectively. On the other hand, the website uses HTML5, CSS3 and JS. The files were manipulated using the Netbeans IDE and the compatibility tests were performed using the online tool BrowserShots.

Chapter 4: System Design and Implementation

4.1 Chapter Overview

This chapter focuses on explaining the intrinsic aspects of the animation created for this project. These aspects include the methodology used, the different sections the animation has, the libraries employed, the GUI, important code snippets, among others. Also, throughout this chapter, the animation will be analysed based on how it helps in the teaching of the AES cipher by using pedagogical practices found on the literature.

4.2 Development Methodology

Even though a specific development methodology was not employed, the use of an agile development mindset was helpful throughout the development of this project. In a project of this size with only one developer and with a short time frame, the use of a specific agile methodology like Scrum or XP was considered to be inappropriate. Those methodologies put great effort on how people interact in the project (Rasnacisa & Berzisa, 2017) which is something that cannot be applied in a one-person development. Because of this a specific methodology was not used, however, the principle of short iterations with clearly defined deliverables used in agile methodologies was useful to achieve the construction of the program in a timely and orderly manner. The development process started on the first week of May, and the iteration length was set as two weeks. Each iteration had clearly defined tasks, the first iteration tasks were focused towards setting up the website and having it online so that new versions of the animation could have been uploaded as soon as they were finished. Every other iteration corresponded to the development of new sections of the animation.

This project could have used a waterfall approach because of its static requirements (the main requirement was to create an animation for the AES cipher). However, the final

decision was to use an agile methodology because the design of the project was not as static as the requirements; and, as stated by Huo, Verner, Zhu, & Babar (2004), waterfall methodologies are good for projects with well-defined requirements and designs. The design of the animation was being developed and changed continuously because the animation had to clearly convey the concepts of the cipher and the user interface needed to be easy to use; therefore, the animation was being continuously assessed by third persons after every iteration to see how the design could have been improved.

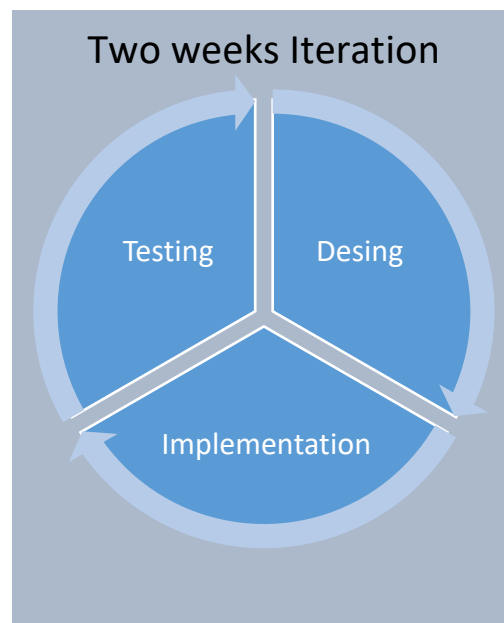


Figure 4.1: Steps involved in each iteration

This Figure shows each of the steps involved in a normal iteration, the length of the iteration was two weeks and the process started with the design phase.

Figure 13 graphically shows the steps that were involved in each iteration:

- The design phase was used to create the blueprints of the animations on paper, they contained elements of the UI and had a draft of what the animations were going to be like.
- The implementation phase involved the coding of the animation in AS3 and the use of third-party libraries.
- The testing phase included correctness and usability tests (the latter were done by third-persons not involved in the project).

4.3 Design Considerations

The animation was designed in such a way that allows users to navigate through the application in a very intuitive manner. A decision was made to embed the animation in a website to allow users to easily access the animation; it can be seen as long as the browser has the Flash plugin installed. The url of the website is the following:

<http://personalpages.manchester.ac.uk/postgrad/richard.leiva/dissert/index.html>

The website is called CryptoSchool and uses a very minimalistic design to allow the users to really focus on what is important, which is the AES animation. Minimalism is important because it impacts how students perceive the material shown to them, when an environment (in this case a website) is cluttered with too much information students can become overwhelmed, therefore, suppressing their cognitive capacity (McMains & Kastner, 2011). The minimalistic design of the website was based on the 'HTML5Up' designs that lets users use its code for any purpose. The design of the website is fully responsive, which means that the website can be seen in most screen sizes without losing quality, however, because the animation was developed using flash, that part of the website will not render appropriately on Apple mobile devices (because of the limitations stated in Chapter 3).

Following the minimalistic principle, the website only contains two tabs which are: Introduction and AES. The first tab gives a brief summary of the author of the dissertation and why the animation was built, and the second tab contains the animation developed in Adobe Flash; a screenshot of the home screen is shown in the following figure:

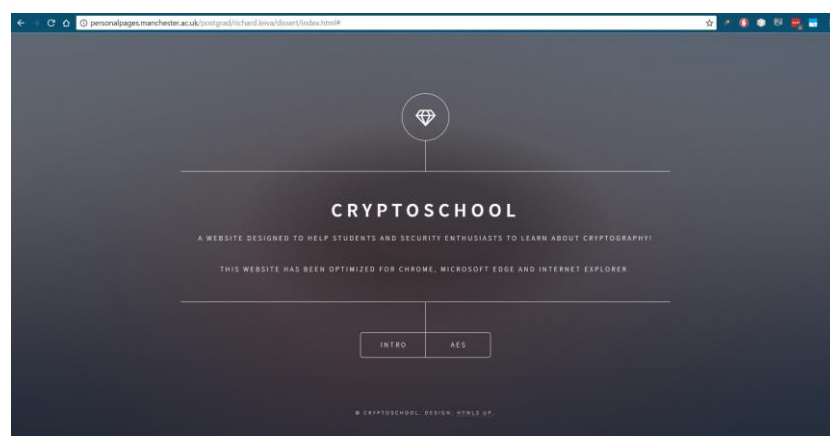


Figure 4.2: Website Home Page

This Figure shows the home page of the website created to embed the AES Animation.

In order to explain the AES algorithm, a detailed animation running at 24 frames per second was built. This frame rate was selected because it allowed to have smooth animations without putting too much pressure on the processor and avoiding ending up with a heavy SWF file (Adobe, 2017). The animation consists of four sections which can be seen in the following use case diagram:

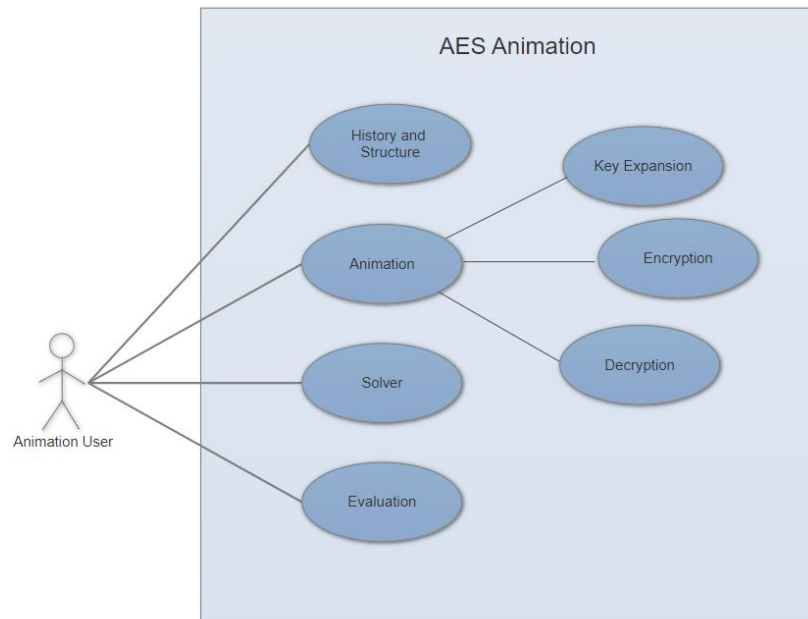


Figure 4.3: Use Case diagram for the AES animation

This use case diagram shows the functions available to the user in the AES animation. As it is depicted above, the user can do 4 things in the animation: Learn about the history and structure of the algorithm, see the encryption, decryption and key expansion animations, interact with a solver and auto-assess his new knowledge with an evaluation.

The 'History and Structure' section allows the student to learn why AES was necessary, and how it works. It explains the user the amount of rounds the AES-128 use and what each round does. The 'Animation' section is divided into three subsections (depicted in figure 4.3) and is the core of the program, because it explains in a detailed manner every step of the AES encryption, decryption and key expansion processes. The 'Solver' lets the user play with the AES encryption and decryption processes, allowing the user to input values and see how they change through each of the rounds of the cipher. The last section is an evaluation that lets users auto-assess their knowledge to see if they understood the key aspects of the cryptographic algorithm.

4.4 Graphical User Interface

Since Adobe Flash support the use of vectors, all the images in the animation are vector graphics. The advantage of working with vector images is that they can be resized without losing quality and that they are easy to manipulate (Liao, Forsyth, Yu, & Hoppe, 2012). Some of the vectors in the animation were obtained from Freepik, which is an online library that contains thousands of free vector images.

The GUI of the animation was developed taking into consideration the usability of the application as the most important feature. Therefore, the final decision was to have the animation organised as if it was a video but with the advantages of letting the user interact with it. The purpose of building the navigation bar as if it was the navigation bar of a video was to allow the user to be able to use it in a natural way. According to a study of 310 internet users, 98% had already watched videos online (Truong, 2009). Another study by CISCO (2017) states that by 2019 video will account for the 80% of the internet traffic. Therefore, it can be generalised and stated that nowadays almost every internet user is comfortable, or at least familiar, with this type of user interface. The navigation bar can be seen in the following figure:



Figure 4.4: Navigation Menu

The navigation bar allows the user to navigate through the animation, it consists of 6 buttons and a slider.

As it can be seen in the previous image, the navigation bar resembles the navigation bar of a YouTube video with a few tweaks:

1. Button one allows the user to go to the home screen of the animation.
2. Button 2 allows the user to go to the next animation. For instance, if the user is in the history and structure animation, by pressing that button, the user will be redirected to the key expansion animation, and so on.
3. Button 3 allows the user to activate or deactivate the sound in the animation.

4. Button 4 allows the user to play or pause the animation.
5. Button 5 allows the user to navigate to the previous step within the current animation.
6. Button 6 allows the user to navigate to the next step within the current animation.
7. The slider allows the user to see the progress of the current animation and it also lets the user to navigate through the timeline of the animation by dragging and dropping the slider.

The navigation bar is only visible on the screens that use animations. There are three static screens that do not have any animation involved unless there is some sort of user interaction, these screens are: Home Screen, Solver, and Evaluation. For instance, the buttons in the home screen are static unless the user clicks on them, that is what triggers an animation. A screenshot of the home screen is depicted in the following figure:

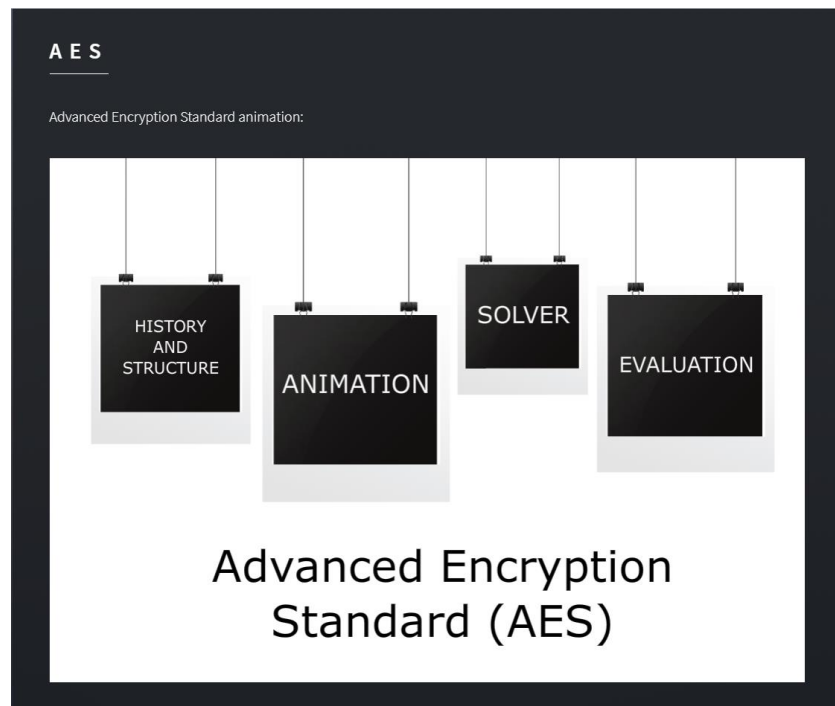


Figure 4.5: Animation Home Screen

This figure shows the home screen of the animation. The home screen is static and starts animating once the user clicks on one of the options provided.

Another important feature of the GUI are hover images and popups (the latter are discussed in section 4.7). The need to use hover images arose with the necessity to let users know what the next section was going to be. Since the AES animation is composed of various

sub-animations, it was necessary to have a pre-established sequence so that users knew which way to go. Even though the user is free to navigate through the application as it best suits him, the application sets a guideline (with the hover images and the next section button). A screenshot of the animation showing the hover image displayed when hovering the mouse on the next section button is displayed in the following figure:

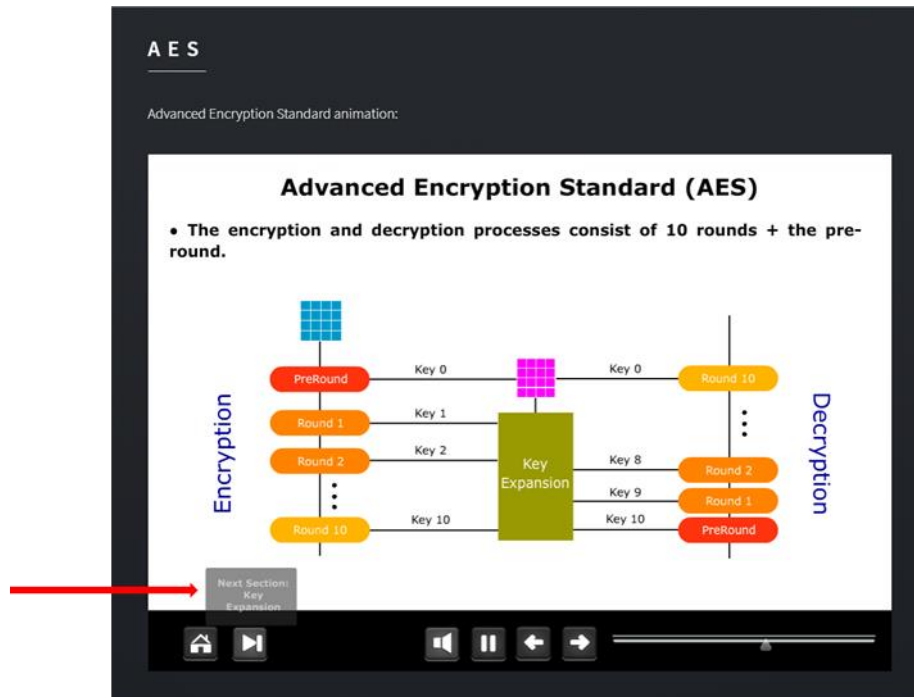


Figure 4.6: Hover Image

This figure shows a screenshot of the History and Structure section. As it can be seen in the navigation bar, when the user hovers over the next section button, a hover image appears telling the user what the next section of the animation is. This feature is especially useful when the user wants to skip sections.

4.5 Audio files

As Koroghlanian & Klein mention (2004), having text and audio in a presentation enhance the learning outcomes of students. Because of the previous statement, audio files were included as part of the initial design of the animation. In total 4 audio files were recorded using the Audacity application which is a free open source audio edition software available on Windows, Linux and Mac computers.

The audios explain (in collaboration with the text on screen) what is happening on the animation; the sections that contain audios are: History and Structure, Key Expansion Encryption and Decryption which are the screens that contain animations. The audio files

were integrated with the Flash animation by using the MP3Loader library from the GSAP animation platform. The use of that library made the manipulation of MP3 files straightforward because only two lines of code were necessary to load the file into the flash animation. Besides that, the library provides developers with a lot of methods to manipulate audios like: play, pause, rewind, fast forward, change volume, among others. These methods were useful to make some items in the animation menu work, specially the “mute/unmute” button and the progress slider. The progress slider was challenging to integrate with the audio because the sound needed to be in sync with the animation when a user rewound or fast forwarded, however, with the facilities provided by GSAP an efficient solution was found (See Appendix A.1).

4.6 Third Party Libraries

The two ActionScript libraries used were GSAP and AS3Crypto. The GSAP library allowed to create rich animations with good performance and the AS3crypto helped to implement the AES algorithm for the solver.

4.6.1 GSAP

The GSAP library was used throughout the whole program. Its great features allow developers to create complex and easy animations in a standardized form. The most used features of the library in this project were: Timelines and Tweens. Tweens let developers interact with the elements on the stage, for instance coders can move, transform and modify the properties of any object. Tweens and Timelines work together because timelines allow to have a sequence of tweens so that complex animations can be created.

For all the animations, the concept of a master timeline was used. When building complex animations, it may be necessary to have multiple timelines because multiple animations with different sequences might occur simultaneously. The use of a master timeline allowed to abstract the concept of using different timelines and permitted to treat all the animations as if they were contained in only one timeline. Appendix A.2 shows some code snippets that depict the previously mentioned statement, where the benefits of using a master timeline instead of dealing with individual timelines can be seen.

GSAP also allows for the creation or use of custom plugins. Out of the box, GSAP offers

38 plugins but for this project only three were necessary. The following image shows the necessary code to tell the ActionScript compiler that GSAP is going to use some custom plugins:

```
TweenPlugin.activate([BezierPlugin, ColorTransformPlugin, HexColorsPlugin]);
```

Figure 4.7: Plugin registration in GSAP

The figure shows how to register to use the BezierPlugin, ColorTransformPlugin and HexColorsPlugin using the GSAP library.

The plugins used were: Bezier, Color Transform and Hex Colors:

- The BezierPlugin was used to create complex animations that did not move along a straight line. This plugin permitted to move objects along a Bezier curve which was fully customizable and allowed to create eye-catching animations like the add round key or the shift rows animations in the encryption process (refer to section 4.7).
- The ColorTransform plugin was used to modify the colours of the backgrounds on the encryption and key expansion animations. This plugin allows to create smooth progressive colour transformations instead of just swapping colours.
- The HexColors plugin was useful to manipulate colours in terms of their hexadecimal value instead of having to edit the Red, Green and Blue (RGB) values.

4.6.2 AS3 Crypto

This library helped to implement the backend code for the solver which allows the user to input a message and encrypt or decrypt it. The code of the library had to be modified because the way it was implemented did not allow to print the values after each step of the encryption / decryption processes which was a required feature for the solver (the original code can be seen in Appendix A.3). In order to be able to accomplish this, the signature of the encrypt and decrypt methods had to be changed to be able to return an object containing the encrypted or decrypted values after each step of the processes. Not only the signature of the methods had to change, but also some semantic aspects of the methods. For instance, the Crypto library used to perform the sub bytes and the shift rows operations all in one method (the same happens with the inv sub bytes and inv shift rows methods for the decryption process), which was not what was needed for the solver. Therefore, the methods had to be

changed and split into separate functions to meet the needs of the solver (the modified code is depicted in Appendix A.4).

After some analysis, it was understood why the library implemented the code in such a way; by mixing the sub bytes step with other operations the memory load decreases which leads to an improvement in the performance of the algorithm (Didla, Ault, & Bagch, 2008). However, because the calculator does not deal with high amounts of data (only one block of 128 bits at a time) the necessity to have the optimized implementation was not justified, consequently, it was safe to perform the above-mentioned changes to the library.

4.7 Implementation

Before starting to explain what each animation does and how they work, it is necessary to understand how the program was organised. The following figure shows which screens are present in the program and what is the order in which the users should see the animations:

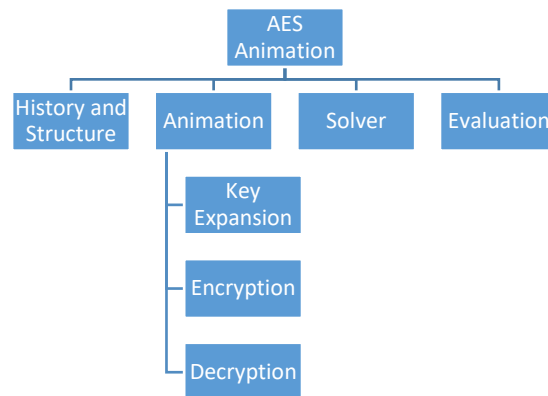


Figure 4.8: Project Organisation

This figure shows an overview of the program and how it was structured

Despite the fact that users can see the animations in any order they want, it is recommended that they start with the History and Structure animation because it gives a brief overview of the algorithm and sets some basic guidelines about how the animations are going to behave. The user should then go to the Key Expansion, Encryption and Decryption animations, and finish with the solver and with the evaluation to auto-assess his knowledge.

Even though every screen in the program contains animations, a division between static and dynamic screens can be made. The difference between static and dynamic screens is that

static screens do not contain the navigation menu (Figure 4.4) because the animations only happen when a user clicks on the buttons present on the GUI. The following figure shows the different type of screens the program has:

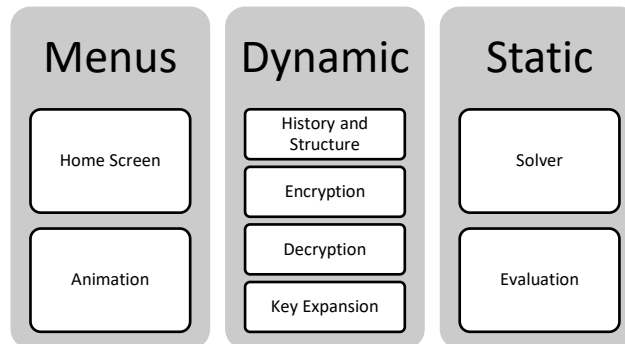


Figure 4.9: Type of Screens

This figure shows the screens present in the program. In total, 8 screens were developed.

4.7.1 Menu

The program has two menus, the main menu (or homepage) which lets the user select four main options (History and Structure, Animation, Solver and Evaluation) and a sub menu which allows the user to select 3 options (Encryption, Decryption and Key Expansion). The menus can be seen in the following figure:

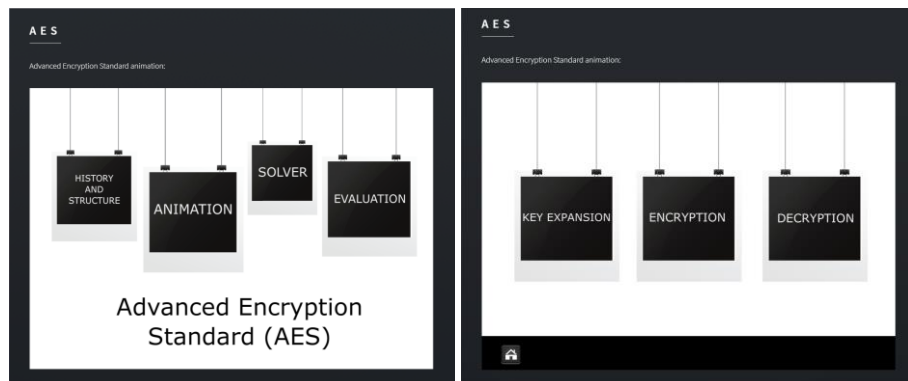


Figure 4.10: Menu of the program

This figure shows screenshots of the main menu (left) and the animation menu (right)

These are the least complex screens in the program and they only animate when a user clicks on one of the buttons. The code developed for these screens mostly uses Event Listeners to be able to detect when a user clicks on the buttons and react accordingly, a sample code can be seen on Appendix A.5.

4.7.2 Dynamic Screens

As shown in Figure 4.9, there are 4 dynamic screens in the program. The first one is History and Structure and corresponds to an animation that gives a broad overview of the AES cipher. This screen shows the user who the inventors of the cipher are, why it was necessary to replace the previous standard (DES) and what type of cryptographic algorithm AES is. It also lets the user know what is the structure of the different versions of the cipher and focuses on explaining the AES-128 version which is the main version discussed in the whole program.

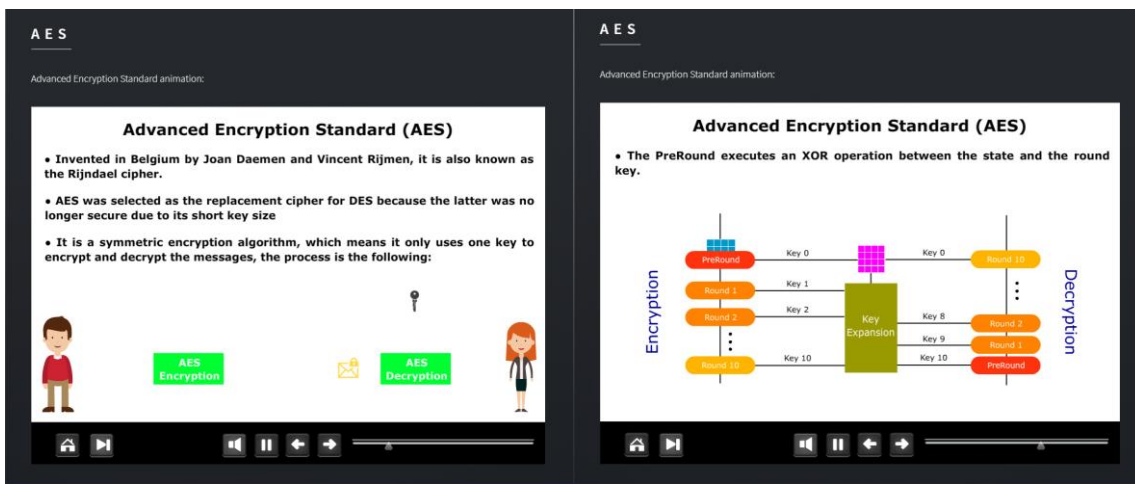


Figure 4.11: History and Structure

This figure shows two screenshots of some of the animations present in the History and Structure section of the program

This screen is animated as if it was a PowerPoint presentation and dynamically displays a variety of information and animations. According to Collins (2004), in order to give an effective presentation, the presenter needs to introduce the subject and state the objectives, which is exactly what the History and Structure section does. This and all the other dynamic screens make a heavy use of the Timeline method “to” which is shown in the following figure:

```
var t1 = new TimelineLite();
t1.to(this.txtHistory1, 3, {alpha:1}, "0");
```

Figure 4.12: Sample Timeline code

Example of a basic animation using the ‘to’ method of the TimelineLite class.

As shown in the previous figure, the signature of the ‘to’ method, which is one of the

most used methods throughout the whole program, contains four arguments. The first argument is the target object that is going to be animated, the second argument is the duration of the animation, the third argument is an array of elements and represent one or more attributes of the object that will be animated, and the last argument corresponds to the position of the animation in the timeline (if the last argument is left empty, the animation will happen at the end of the previous animation). With the 'to' method complex animations can be created by nesting different elements and timelines, one of these complex animations can be seen in Appendix A.6.

The second section of the program is the Key Expansion animation. This section explains how the key expansion algorithm works and the user is able to visually see what are the operations that the algorithm uses to expand the original key so that each round of the encryption/decryption processes use a different subkey. Some screenshots of this section are shown in the following figure:

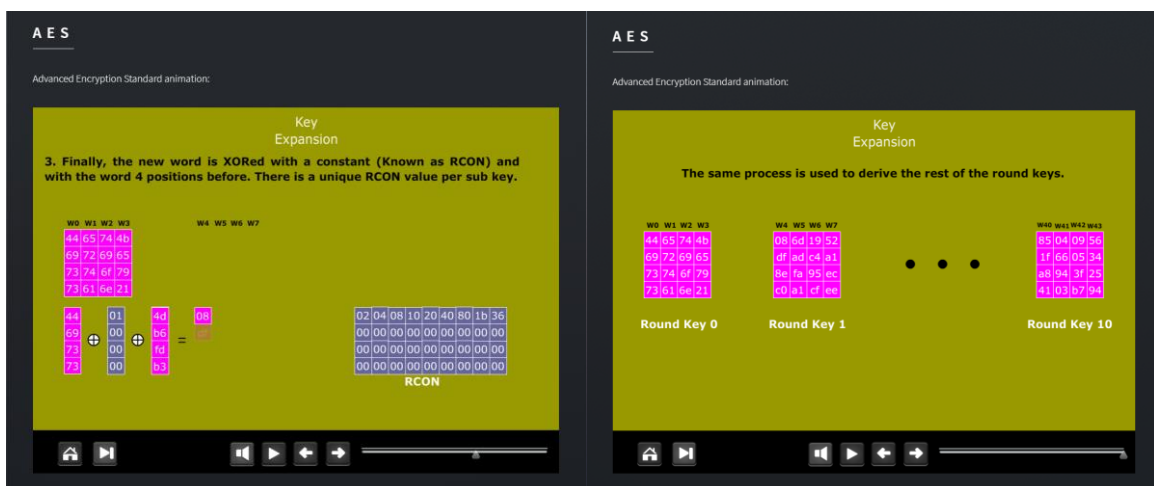


Figure 4.13: Key Expansion

This figure shows two screenshots of the Key Expansion animation section

Figure 4.13 only shows a couple of the operations involved in the key expansion process, a lot of the animations involved the use of Bezier paths, linear movements, alpha modifications and voice instructions. The third section of the program is the encryption animation which is the longest and the most complicated animation developed for this project. The first thing the user notices when going to this animation is a popup which can be seen in the following figure:

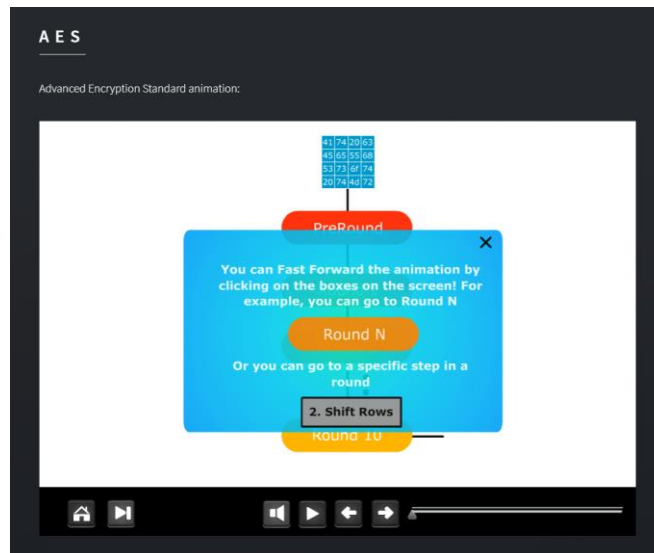


Figure 4.14: Encryption Animation Popup

This figure shows a screenshot of the Encryption section that shows the use of a popup window to give directions to the user.

This popup is very useful because it lets the user know how he can interact with the animation. In the encryption animation, the user is able to click on multiple places to skip or return to some sections, therefore, it was necessary to let the user know which the clickable items were. The clickable items in this section are: the round buttons that represent each round and the squared buttons that are used to specify the steps in each round (see figure 4.15 and 4.16). This screen has a variety of animations for each of the encryption steps and involve the use of: voice instructions, Bezier path animations, transparencies, spatial reordering, among others. As soon as the user clicks on the close button of the popup window the animation starts playing. Throughout the encryption process the user is able to see how the values in the state (blue grid in Figure 4.14) change through the different rounds of the algorithm. The first animation is the pre-round which consists of an animated XOR operation between the state and the key matrices; some screenshots are depicted below:

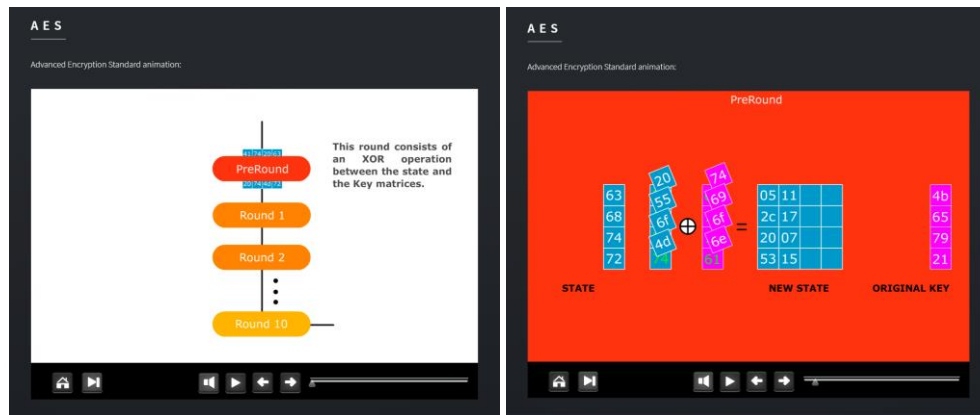


Figure 4. 15: Pre-round animation (Encryption)

This figure shows two screenshots of the pre-round animation. This is a complex animation built using Bezier paths.

Because the animations needed to be reusable, the modularity of the code was very important. For instance, the code used to build the pre-round animation (specifically the XOR animation) was needed for the Add Round Key step and for the Key Expansion Algorithm as well. For this reason, it was important to build high quality code that easily allowed to reuse animations previously built which helped to save time and effort while coding (Appendix A.7 shows an extract of the code that was used to create the reusable XOR animations).

The next step in the process is the animation of round 1. Round 1 animation consists of 4 parts which are: Sub Bytes, Shift Rows, Mix Columns and Add Round Key. The Sub Bytes animation is the longest animation in the program because it visually shows how each byte in the state is substituted with a byte in the AES S-Box. Since it is very lengthy, a user might want to skip a part of this animation which can be easily done by clicking on the square buttons containing the name of the steps, or by clicking on the next button (besides the play button), or by dragging and dropping using the progress bar. The user has a lot of options to interact with the animation so that he can focus on the parts that he really needs to focus on, therefore, the user is able to exploit to the maximum the teaching capabilities this program offers. The following figure shows some animations that take place in the Sub Bytes and Shift rows sections:

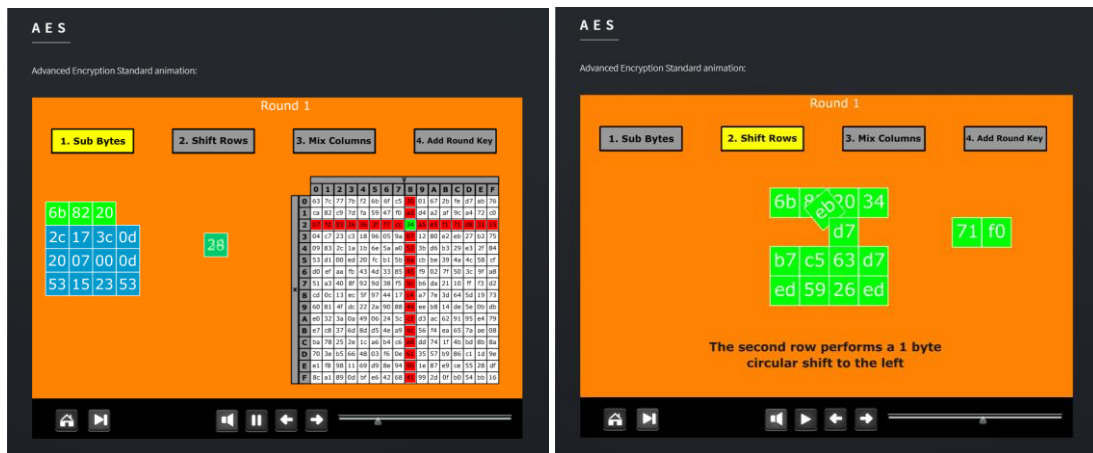


Figure 4.16: Sub Bytes and Shift Rows

This figure shows screenshots of the Sub Bytes (left) and Shift Rows (right) steps.

As shown in the previous figure, the colours used for the animations are vivid. The reason behind the choosing of the colours was that they needed to be self-explanatory. As stated by Brown & Hershberger (1992), colours are very important because they help to communicate information, especially, for animations of algorithms. With the help of colours, smooth transitions between the different animations were achieved. Also, colours aided to highlight important activities which required immediate attention of the viewer. Since the first section (History and Structure), the same colours were used to represent each round, and the key and state matrices, allowing the users to start relating colours to objects which improves the learning outcomes of the students. The figure below uses the same colour principles and show the Mix Columns and Add Round Key animations.

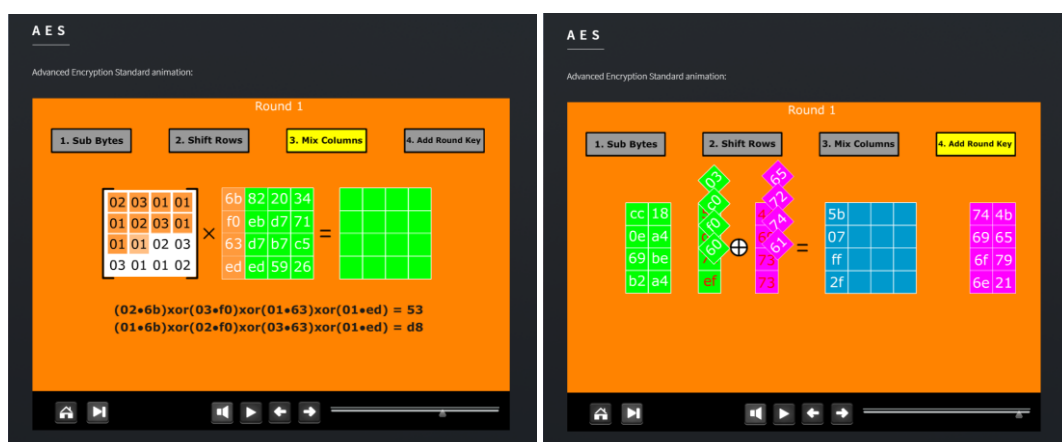


Figure 4.17: Mix Columns and Add Round Key

This figure shows screenshots for some of the animations in the Mix Columns and Add Round Key sections

The last part of the encryption animation explains the difference between round 1 and round 10. Basically, all the rounds are the same except round 10 which omits one of the encryption/decryption steps (explained in chapter 2). The following figure shows the final explanations of the encryption animation:

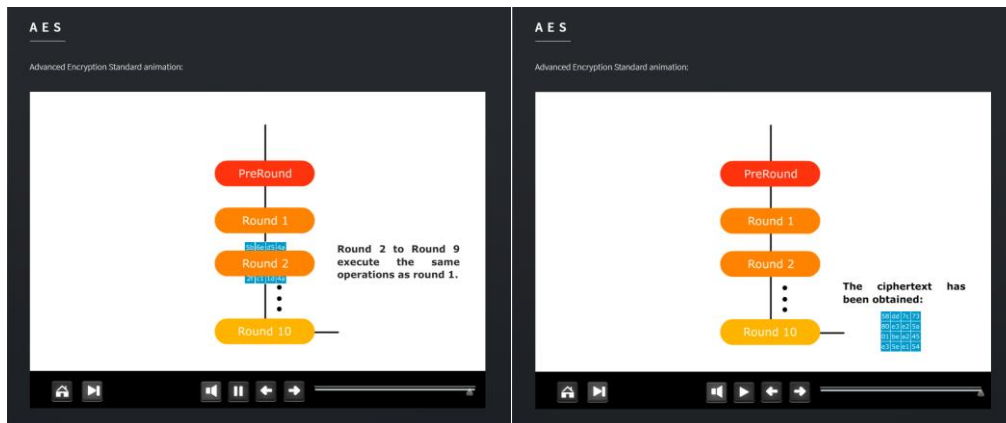


Figure 4. 18: Final explanations of the encryption process

This figure shows the final parts of the encryption animation and shows the user the ciphertext obtained after the state went through all the encryption rounds of the AES algorithm.

Only round 1 was animated in detail because there was no point in explaining the other rounds since they perform the same exact steps as round 1, except for round 10 that ignores one of the steps which is explained in the animation. The last dynamic screen is the decryption animation. Since the encryption and decryption processes are very similar, this animation focuses on explaining the differences between them (refer to chapter 2.4.1.1).

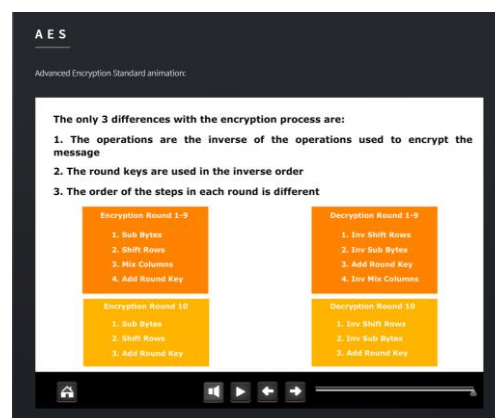


Figure 4. 19: Decryption

Screenshot of the decryption animation that describes the differences between the AES encryption and decryption.

It is important to mention that none of the tools already available (refer to related works chapter 2.6.1 and 2.6.2) explain the decryption process because of its similarities with the encryption process. However, since the goal of this project was to help students really understand the AES algorithm, the differences of encryption and decryption were clearly described with the aid of text and audio guidelines.

4.7.3 Static Screens

The program consists of two static screens which are the AES solver and the user evaluation. These screens are different from dynamic screens in the sense that they are not interactive animation movies. These screens allow the user to enter data to the system and receive information back. The solver is a calculator for the AES cipher that uses a tweaked version of the AS3 Crypto library (refer to chapter 4.6.2) and allows the user to enter string or hexadecimal values to encrypt or decrypt messages. This AES calculator allows to process one block of 128 bits at a time (remember that the AES cipher is a block cipher that process blocks of 128 bits as explained on Chapter 2). The user is able to select which kind of data he wants to input and according to that the character limit varies. On the one hand, if the user selects to input strings the maximum amount of characters allowed is 16. On the other hand, if the user chooses to input hexadecimal values the character limit is 32 (16 string characters or 32 hex values correspond to 128 bits which is the size of the AES block).

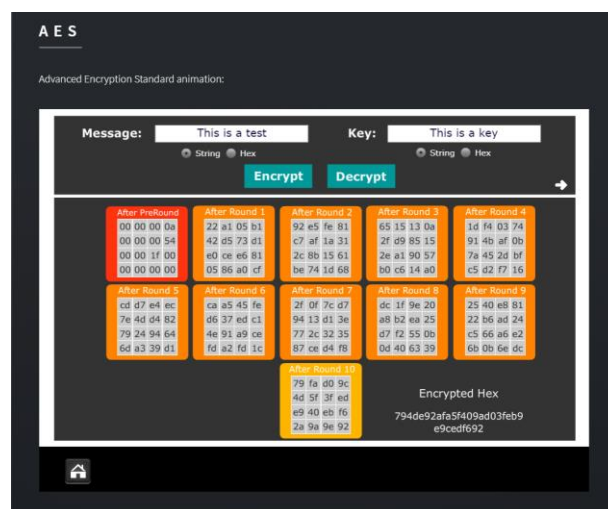


Figure 4.20: AES Solver

This figure shows a screenshot of the AES solver after entering “This is a test” as message and “This is a key” as the encryption key.

If the user does not input a value in the message or key boxes, or if the inputted value is not exactly 16 chars or 32 hex values, the solver automatically pads the message/key with empty spaces. Once the user clicks on the encrypt or decrypt button, the system shows the user how the values of the message change after each round of the algorithm. Because of the screen size limitations, a decision was made to use popup windows to display more detailed information of each round. If the user clicks on the image of a round, a popup will appear displaying what the round key for that particular round is, and what are the values after each step of the round, the following figure shows an example:

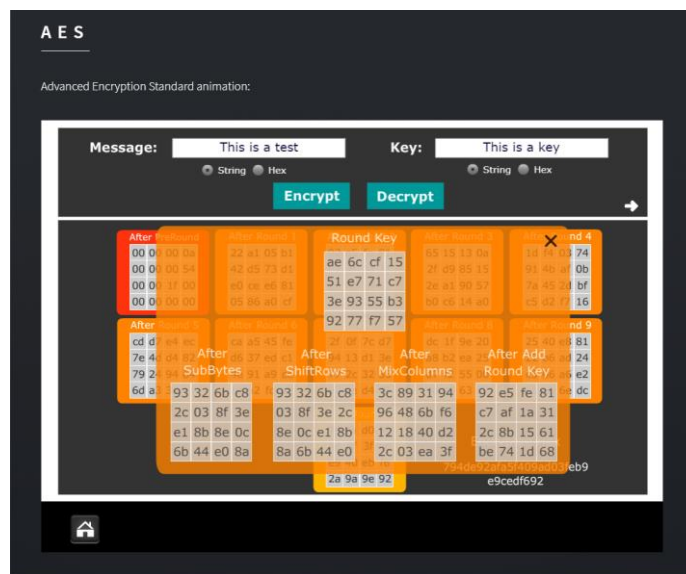


Figure 4.21: AES Solver Popup

This figure shows the popup after clicking in one of the rounds. The popups show the values of the message after each of the four steps of a round.

The second static screen is the evaluation. The purpose of the evaluation is to let the user self-assess his learning after using the animations and playing with the solver. In an education environment, self-assessment is important because “it enhances student motivation by providing a sense of ownership and responsibility” (McMillan & Hearn, 2008). This screen consists of 8 multiple choice questions that test the student’s reasoning and knowledge regarding the AES algorithm. Different messages are shown based on the score that the student received, if the student gets:

- 8 correct answers the message is “You nailed it!”
- 5 or more correct answers the message is “Good enough!”

- Less than five correct answers the message is “Seems like you need more practice!”

After seeing the score the solver provides, the user can also see which errors he made. Some screenshots of the evaluation are shown in the following figure:

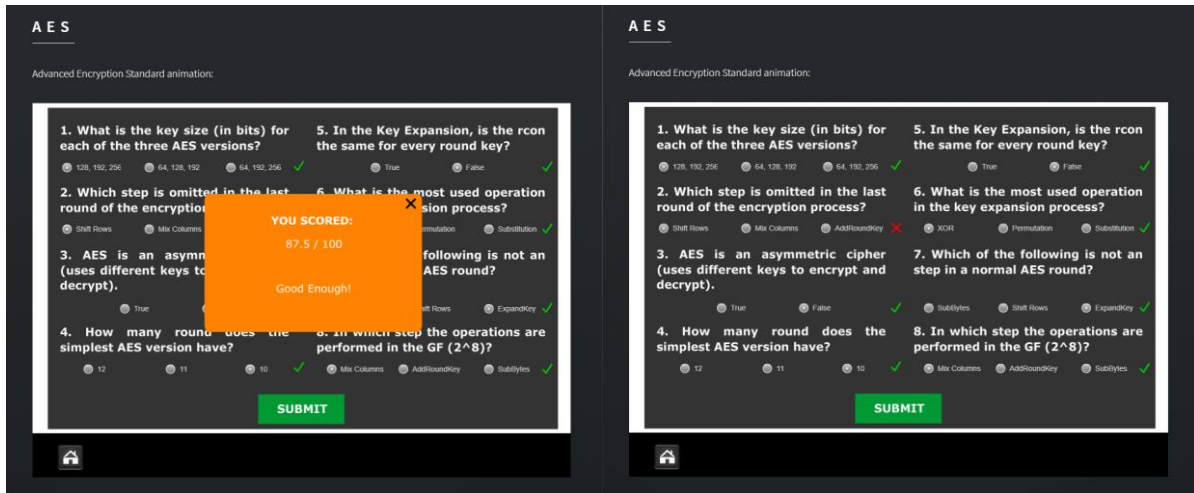


Figure 4. 22: Evaluation

This figure shows the evaluation screen which lets the user to self-assess his knowledge regarding the AES cipher.

4.7.4 Size Optimisations

Even though the use of the GSAP library facilitated the creation of the animations and object manipulation, because of the length and complexity of the whole program, the final product ended up having more than 2700 lines of code written explicitly for this project. It is worth mentioning that all the animations were implemented with ActionScript code and not using the visual aids provided in Adobe Animate CC.

Since the program is shown inside a webpage, its final file size matters because the heavier it is the more time it takes to load on a website. The final size of this program was 264KB which is a light size for an animation with the features explained in the previous chapter (4.7.3). The reason behind such a small file size (taking into consideration the amount of animation time and functionality the program has) is because GSAP is a lightweight library that allowed to create complex animations without increasing the flash file size significantly. Also, the use of MP3 files for the audio in the animations allowed to have shorter file sizes in comparison to other audio formats like WAV (which can be up to 10 times heavier) which

helps to load the animations inside the website faster.

4.8 Summary

This chapter showed that even though a formal development methodology was not used, the application of an agile mindset was beneficial to achieve the goals of the project in a timely manner. As mentioned, the screens were divided into static and dynamic screens. On the one hand, dynamic screens were organised as if they were interactive animation movies which means that they have a navigation bar and make use of audio files to enhance the learning process of the student. The dynamic screens explain the concepts of the algorithm in depth, giving users all the knowledge they require regarding the AES cipher. On the other hand, static screens allow the user to input values and answer questions. The solver, which is a static screen, lets the user play with different inputs and see how the values change throughout the different encryption/decryption rounds of the algorithm. The second static screen is the evaluation which lets users answer questions and self-assess the learning outputs of the program. This chapter explained why the website and the animation were built the way they were built, backing up the decisions with literature material that shows how the design considerations taken helped to improve the teaching capabilities of the developed software.

Chapter 5: Evaluation

5.1 Chapter Overview

Chapter 5 analyses the developed software in terms of its performance, security and compatibility. This chapter also assesses the usefulness of the animations by analysing the results of a questionnaire and comparing the program with similar products that are already available on the market. By performing these analysis, the effectiveness of the project can be assessed which is useful to determine if the developed product would be useful in a real teaching environment.

5.2 Performance testing

It is important to assess the performance of the developed animation because only by having smooth and fluent animations users are able to really take advantage of the program. Having stuttering animations lowers the quality of the user experience and may cause users to stop believing in the usefulness of the program. Therefore, the purpose of this test is to measure how optimal was the implementation of the code and how fluent the animations are. To assess the performance of the animation, the Adobe Scout CC program was used. The History and Structure animation was the first to be analysed and the results can be seen in figure 5.1.

As it was explained on chapter 3.3.3, the most important part of the Adobe Animate CC results is the red line on the top part of the screen. As long as the bars do not surpass the red line it can be assured that the animation is fluid because all the activities going on in the animation do not surpass the budget time. As shown in figure 5.1, the only time where the budget time is surpassed is at the beginning of the animation which is justified because that is when all the objects are loaded (but not necessarily displayed) into the stage. Even so, the surpass of the budget time only occurs in one frame, therefore, it is imperceptible to the user.

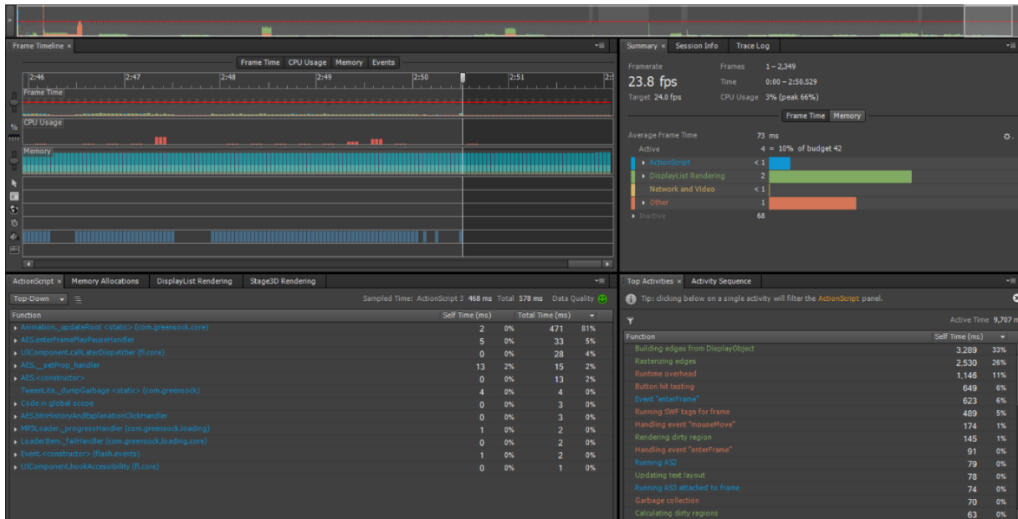


Figure 5. 1: History and Structure Analysis

This figure shows the Adobe Scout CC IDE running with the History and Structure animation.

The target fps for this animation was 24 (refer to chapter 4.3) and the average fps obtained was 23.8 which basically means that the animation runs fluently. The average CPU usage was 3% which shows that this animation represents a very little load to the CPU. On the top right side of the screen, under the fps section, 4 bars (blue, green, yellow and pink) can be seen. These bars represent the amount of time the flash animation spent performing a certain activity:

- Blue bar: ActionScript execution
- Green bar: Object rendering
- Yellow bar: Network and Video events
- Pink bar: Other events like garbage collections and other overhead.

As shown in Figure 5.1, the most time-consuming activity is the object rendering (green bar) which is understandable because the animations use a lot of texts and vector images (cached as bitmaps to improve the performance) that need to be loaded. The least time-consuming activity (without considering the yellow bar because no network or video features are used in this program) is the code execution. This means that the developed code works in optimal conditions and that is because good quality code was written with the help of the GSAP library.

The same analysis was performed for the rest of the animations (Key Expansion, Encryption and Decryption) and is shown in figure 5.2. As it can be seen, all of the animations

reach at least 23.4 fps which is a good frame rate that allow to have smooth animations. The three animations surpass the budget time in the beginning of the timeline, this happens because of the same reasons explained above (all the objects are loaded in the beginning). However, the encryption animation has a few other frames were the budget time is surpassed. Even though the effect on the performance is not noticed by the user because the surpassing of the budget time only occurs in individual and isolated frames, an examination was done. The reason why the budget time is surpassed on those frames is because complex object transformations with audio instructions are taking place. This is understandable because in the beginning of this transformations (when the budget time is surpassed) is when flash starts performing calculations and buffering the audio files. The CPU usage ranges between 8% and 11% on average and the least time-consuming task for all the animations is the code execution (blue bar), which means that the code is optimal.

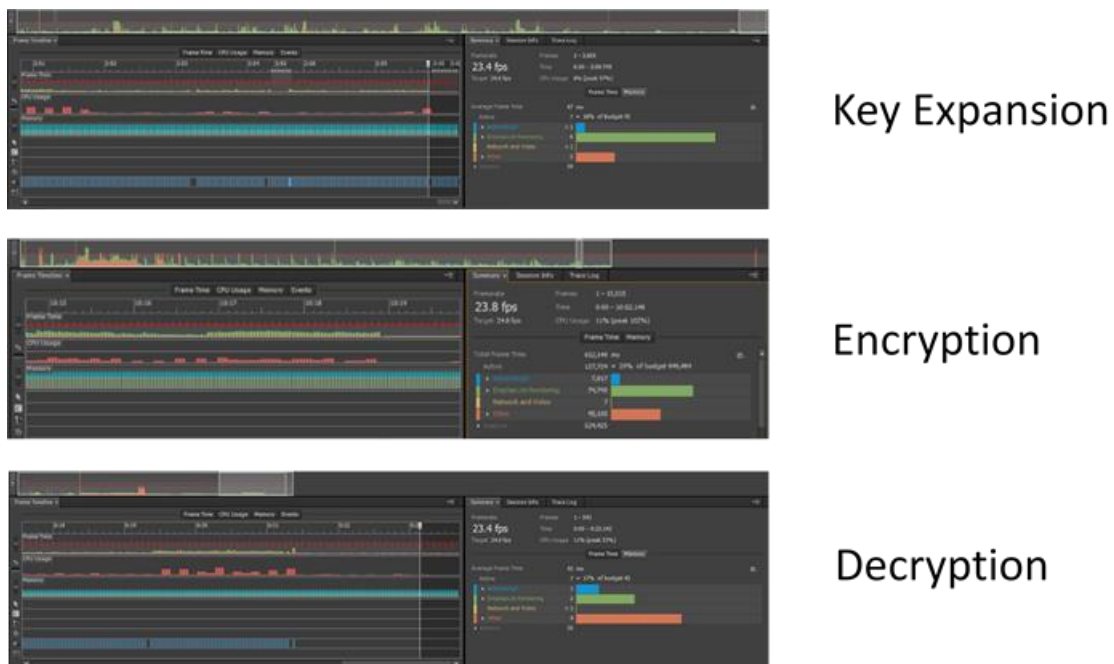


Figure 5. 2: Key Expansion, Encryption and Decryption Animation Analysis

This figure shows the Adobe Scout CC analysis for the key expansion, encryption and decryption animations. As it can be seen, the implementation of all the animations was efficient and optimal.

As shown in the analysis, the code implemented with the GSAP library is optimal and the user is able to interact with a fluent animation with smooth transitions (23.4 - 23.8 fps) without putting much load on the processor (7% - 11% on average). It is important to mention

that the performance tests will vary depending on the computer where the program is running. For the purpose of this assessment, a Dell laptop with 16GB of RAM and an Intel Core i7-7700 running at 2.8GHz was used.

5.3 Security testing

Adobe Flash is widely known to have had security issues in the past, therefore, it was necessary to evaluate the security of the developed animation to make sure that it is hard for an attacker to exploit a vulnerability in the code. To assess the security of the developed animation the SWFScan program was used (refer to chapter 3.3.2). The following figure depicts the initial analysis of the animation:

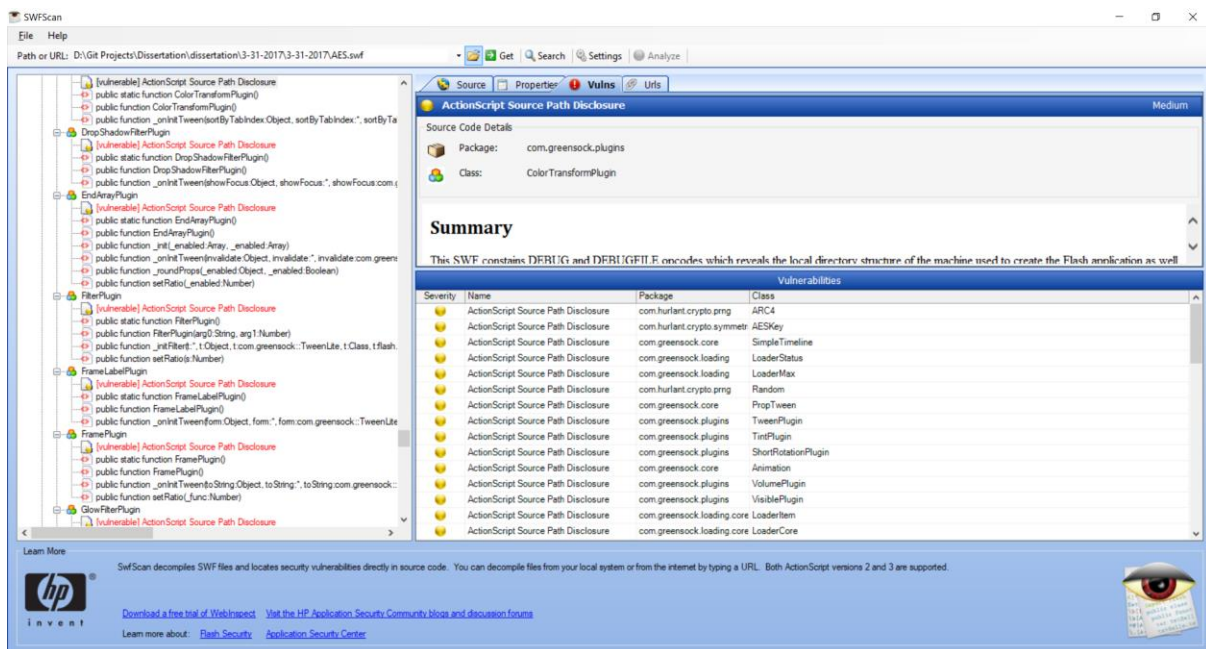


Figure 5.3: Initial security analysis

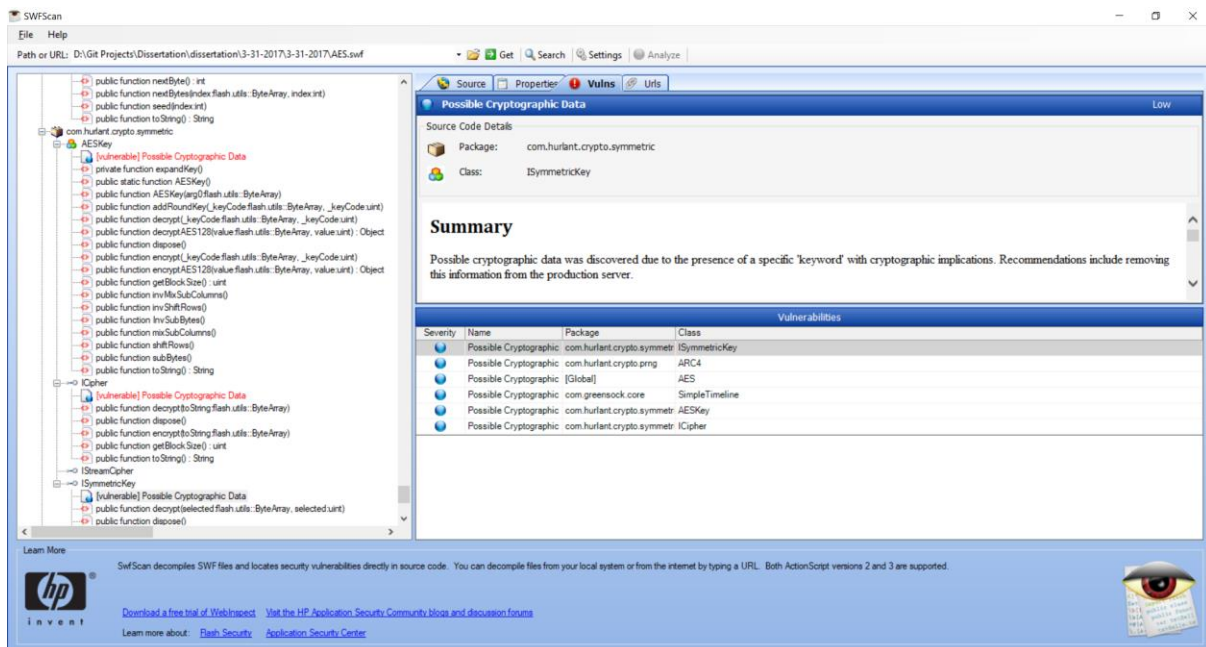
This figure shows a screenshot of the SWFScan tool after the analysis of the animation was performed.

Figure 5.3 shows that the analysis of the application found 52 medium-risk and 6 low-risk vulnerabilities. On the one hand, the medium-risk vulnerabilities can be categorised as 'ActionScript Source Path Disclosure' and 'Debug messaging'. On the other hand, the low-risk vulnerabilities were of the type 'Possible Cryptographic Data'.

Source Path disclosure and Debug messaging vulnerabilities can be dangerous because they can give information about the intrinsic workings of the application to an attacker who

can use that knowledge to find loopholes in the application or reverse engineer it. These vulnerabilities were fixed by removing all the ‘trace’ method calls that were used to debug the application and also by exporting a new version of the animation using the release mode instead of the debug mode.

Possible Cryptographic Data vulnerabilities are not real issues in the case of this project. The static analysis mentioned that the AES algorithm was being used and that it was explicitly mentioned in the code. However, this would be a real issue if the program was using the AES cipher to encrypt messages between two parties, but the use of the algorithm in this application is for teaching purposes only. Therefore, letting an attacker know that the AES cipher was used in this animation is not a real vulnerability.



The screenshot shows the SWFScan application interface. The left pane displays a tree view of the source code, with several instances of '(vulnerable) Possible Cryptographic Data' highlighted. The main pane shows a detailed view of a 'Possible Cryptographic Data' vulnerability, including source code details (Package: com.hurlant.crypto.symmetric, Class: ISymmetricKey) and a summary stating that possible cryptographic data was discovered due to the presence of a specific 'keyword' with cryptographic implications. Below this, a table lists the remaining vulnerabilities:

Severity	Name	Package	Class
Low	Possible Cryptographic Data	com.hurlant.crypto.symmetric	ISymmetricKey
Low	Possible Cryptographic Data	com.hurlant.crypto.prng	ARC4
Low	Possible Cryptographic Data	[Global]	AES
Low	Possible Cryptographic Data	com.greensock.core	SimpleTimeline
Low	Possible Cryptographic Data	com.hurlant.crypto.symmetric	AESKey
Low	Possible Cryptographic Data	com.hurlant.crypto.symmetric	ICipher

Figure 5. 4: Final security analysis

This figure shows the security analysis for the application after the medium-risk vulnerabilities were fixed.

Figure 5.4 shows the final results of the security analysis performed to the latest version of the program. As it can be seen, the 52 medium-risk vulnerabilities are no longer present and the only vulnerabilities left are the ‘Possible Cryptographic Data’ vulnerabilities, which as previously mentioned, are not real threats.

5.4 Compatibility testing

The animation was embedded in a website to allow students to be able to easily access it. Therefore, it was necessary to analyse if the website was rendered correctly in different browsers and operative systems so that users can take full advantage of the developed product in different environments. The tool used to perform this analysis was Browser Shots (refer to chapter 3.3.4) and the results obtained were favourable. Out of the 150 browsers that could have been tested, only the latest version of the major browsers (Chrome, Firefox, Safari and Opera) were tested. The test consisted of 17 browsers running in Linux, Windows and Mac machines, and the results can be seen in the following figure:



Figure 5.5: Browser Shots results

This figure is a screenshot of the results of the compatibility test for the website in which the animation is embedded.

Figure 5.5 shows that the website was correctly rendered in all the tested browsers. These browsers were:

- Windows: Chrome 51, Chrome 39, Firefox 38, Firefox 40, Opera 36, Opera 35
- Linux: Opera 12.02, Opera 12.16, Firefox 53, Firefox 52, Chrome 60, Chrome 57
- Mac: Firefox 48, Firefox 47, Safari 9.1, Chrome 48

Since Browser Shots does not support Internet Explorer or Microsoft Edge browsers, a different tool was used for that analysis. It was important to analyse Microsoft's browsers because they still have a big number of users and because, in the website development world, these browsers are known to cause compatibility issues. The latest versions of Edge and Explorer were tested in Windows 7, 8.1 and 10 using the online automated testing platform SauceLabs. The obtained results can be seen in Appendix B which show that the animation is

correctly rendered in the previously mentioned browsers and operative systems. As shown in the results of Browser Shots and SourceLabs analysis, the website renders correctly in the latest versions of the most famous and widely used browsers in the world which is beneficial to students because they can access the software from a variety of environments. As discussed in chapter 3.2.2, the support of Flash applications in mobile devices is poor; that is why compatibility tests were not done in this kind of devices because CryptoSchool was not meant to work on those devices from the beginning of the project.

5.5 Comparison with Cryptool, CryptoWorkflow and YouTube

There is a variety of tools that already tackle the problem of teaching cryptography to students, therefore, it is important to compare the most important and well-known platforms that teach the AES cipher with the program developed for this project (CryptoSchool). The following table helps to understand the benefits of each platform because it summarizes and compares key aspects of each program.

	CryptoSchool	Cryptool	CryptoWorkflow	YouTube
User Interaction	Yes	Yes	Yes	No
Theoretical teaching	Text and Audio	Text	None	Text and Audio
Animations	Yes	Yes	No	Yes
Algorithm Practice	Solver	Calculator	Simulation	None
Self-Assessment	Evaluation	None	None	None
Supported Operative Systems	Windows, Linux, Mac. (If the Adobe Flash plugin is installed)	Windows	Windows, Linux, Mac	Windows, Linux, Mac.
Supported Algorithms	AES	Multiple	AES, DES, Vigenere (more can be added)	Multiple
Languages	English	English, German, Spanish	English	Multiple

Needs installation	No	Yes	Yes	No
Mobile devices support	No	No	No	Yes

Table 5.1: E-Learning Tools Comparison

This table summarizes the key features of each e-learning platform that help students learn the AES cipher.

As it is shown in the table above, CryptoSchool has many advantages over the other e-learning tools. The main characteristic that the rest of the tools lack and CryptoSchool offers is the self-assessment feature which allow students to measure how much they know about the AES cipher. However, CryptoSchool has one disadvantage in comparison with YouTube, and it is the fact that it cannot be used in Apple mobile devices. An important attribute that is only present in CryptoSchool and YouTube is that they do not need to be installed in order to be used. The other programs need to be downloaded and installed in each machine where the program is going to be run which demands extra effort by the users before being able to use the e-learning tool. Also, since this project focuses on the AES cipher, the CryptoSchool program only teaches the mentioned algorithm; however, the other tools support more algorithms that could be added to CryptoSchool in the future.

5.6 Questionnaire

A questionnaire intended to be filled out by MSc. ACS students at the University of Manchester was created to assess if CryptoSchool is useful to students or not. In order to know the number of students that was needed, the formula found in Montgomery & Runger (2014) was used and modified to fit the current study:

$$n_0 = \frac{Z^2 * p * (1-p)}{e^2} = \frac{1.645^2 * 0.5 * 0.5}{0.15^2} \approx 30$$

$$n = \frac{n_0 N}{n_0 + (N-1)} = \frac{30 * 90}{30 + (90-1)} \approx 23$$

Equation 1: Finite Population Sample Size formula

The previous equation determined that 23 students (sample size) were necessary to have a representative population so that the results obtained from the analysis of the

questionnaire could be trustworthy. To obtain the sample size, some assumptions were made:

- The total population of MSc. ACS students was 90 (N)
- The chosen confidence level was 90% whose associated Z-value is 1.645
- The margin of error (e) accepted was 15%
- The percentage of the studied population (p) was unknown, so 50% was chosen

It was explicitly mentioned in the questionnaire that the participation in the survey was completely anonymous and voluntary. The questionnaire consisted of 18 closed questions (multiple choice) that evaluated some aspects of the participants as well as the experience they had while using the CryptoSchool website / animation and can be seen in Appendix C (a summary of the results is detailed in Appendices C.3, C.4 and C.5). Some of the questions used a linear scale (answers from 1 to 5) in which 1 was the best possible answer and 5 was the worst (e.g. 1 fluent - 5 Not fluent, 1 Excellent - 5 Very poor), and other questions had yes/no answers. Knowing important characteristics about the participants was useful to understand some of their answers and also to know that the target audience of the project was being evaluated. The questions that asked about characteristics of the respondents were:

- Are you an MSc. ACS student?

One hundred percent of the people that participated in the survey were MSc. ACS students since that was the target group of this project, this can be seen in the following figure:

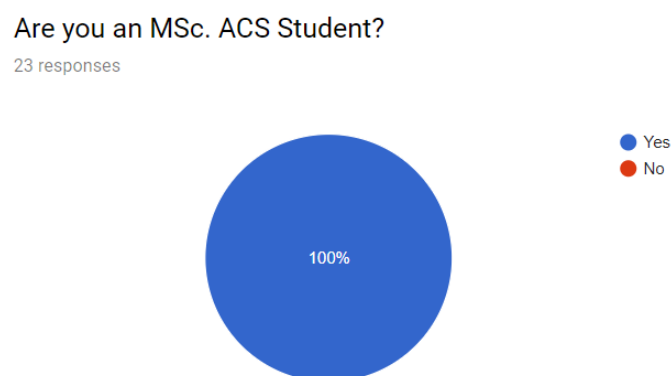


Figure 5.6: Question 1 Results

- How fluent are you in English?

This question was important because all the audios and texts are in English. This question

allowed to find out that not all the respondents were fluent in English, the variety of results show that 60.9% have good English knowledge while 39.1% have an average or below English fluency. This is because most of the students enrolled in the MSc. ACS at the University of Manchester are international students.

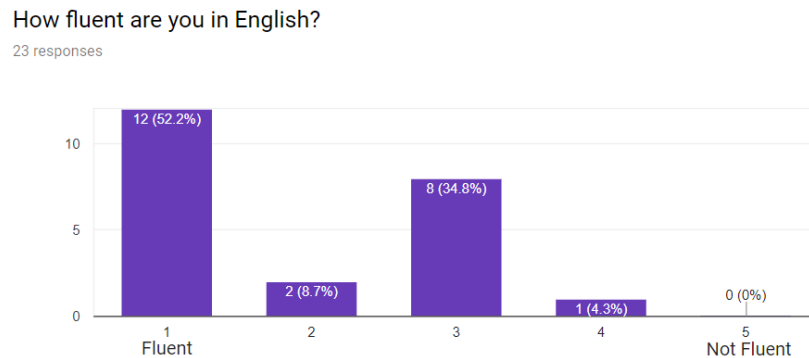


Figure 5.7: Question 2 Results

- How good was your knowledge regarding the AES cipher before using CryptoSchool?

The answers to this question indicate that the majority of the respondents (60.8%) had poor or almost no knowledge about the AES cipher before using the tool developed for this project. Only 26% of the students had good or excellent knowledge of the cipher. These answers make sense because even though all the respondents are MSc. ACS students, they all come from different backgrounds in which they might or might not had cryptography lectures. Also, this question was especially important because the tool is aimed to help people with or without knowledge of the cipher. Therefore, if most of the evaluated population was new to the AES cipher, the teaching capabilities of the developed animations could be more critically evaluated.

How good was your knowledge regarding the AES algorithm before using CryptoSchool ?

23 responses

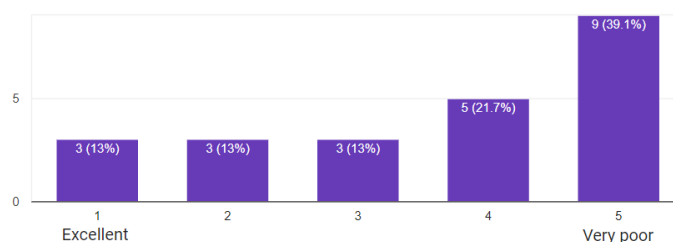


Figure 5.8: Question 3 Results

The rest of the questions were more focused towards finding what was the experience the users had whilst using the application. This part consisted of 15 questions that evaluated the ease of use and teaching capabilities of the developed product, the questions are:

- How easy was it to use the website?

As shown in figure 5.9, all the participants agreed that the website was easy / really easy to use. This is because the website used a minimalistic approach (discussed in chapter 4) which allowed to have a very simple and easy to use design. By only having two tabs (which can be expanded to more tabs if new algorithms are added), the user clearly knows how to navigate the site and is able to focus on what really matters, which is the AES animation.

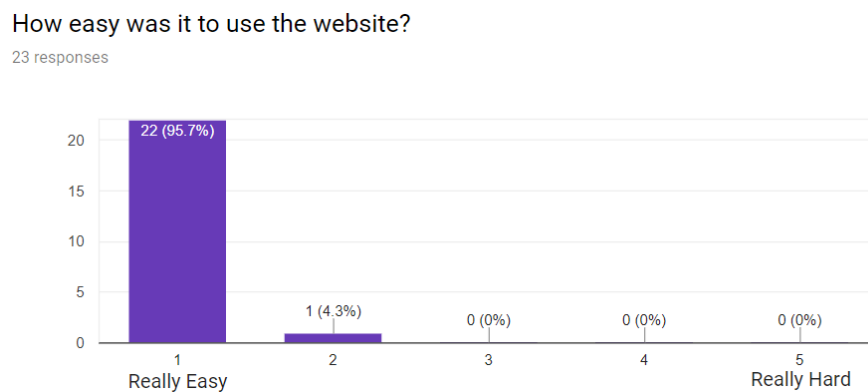


Figure 5.9: Question 4 Results

- How easy was it to use the AES animation?

One hundred percent of the participants agreed that the AES animation was really easy to use, as it is shown in figure 5.10. As discussed in chapter 4, the main characteristic considered when building the animation's GUI was the usability. Based on the results of this question, it can be said that the approach of building the animation in a similar way to what a YouTube video looks like worked perfectly, because the users were able to intuitively navigate through the animation. Also, the fact that users were guided through the application (there was a button to go to the next section/animation) might have helped them to perceive the flow of the application as smooth and simple.

How easy was it to use the AES animation?

23 responses

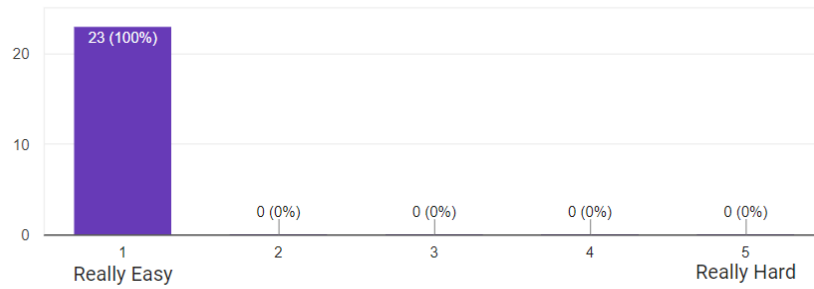


Figure 5.10: Question 5 Results

- Did you experience any performance issues while using the website/animations?

As shown in figure 5.11, none of the users experienced performance issues. This was an expected result because the performance of the animations was evaluated before using the Adobe Animate CC software. The previous analysis (discussed in chapter 5.2) showed that there were no leak of resources and that the desired frame rate was being met which allowed to have smooth animations. Even though in the Animate CC analysis there were some frames where the budget time was not being met, it was not perceptible to the users because those issues only happened in isolated frames. Therefore, none of the users experienced any kind of stuttering while using the application as shown in the following figure:

Did you experience any performance issues while using the website/animations?

23 responses

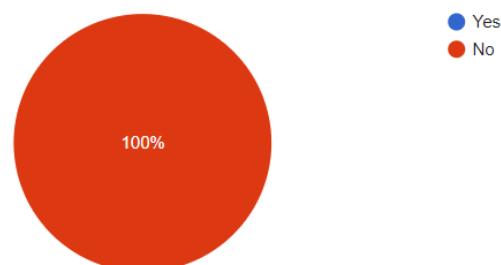


Figure 5.11: Question 6 Results

- The History and Structure section helped you to get an overview of how the AES cipher works

This question indicates that the majority of the respondents agreed that the History and Structure section was useful to understand the basics of the AES algorithm. Only the 4.3% did not agree or disagree with the fact that this section helped to get an overview of the AES cipher. Overall, it can be said that the History and Structure section was properly executed and achieved its goal which was to introduce the users to the AES algorithm.

The History and Structure section helped you to get an overview of how the AES cipher works

23 responses

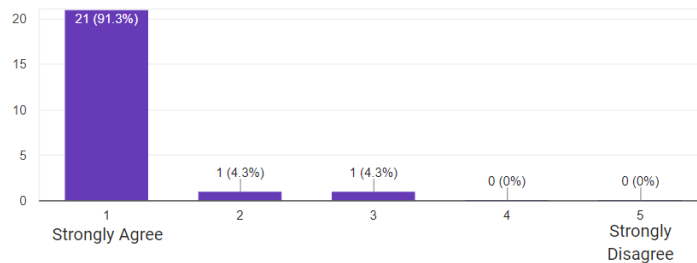


Figure 5.12: Question 7 Results

- The Key Expansion animation allowed you to understand the inner workings of the process

The Key Expansion animation is the second longest animation in the program and, based on the results found in Figure 5.13, this animation achieved the goal of explaining how the key is expanded in the AES algorithm. The results obtained are according to what was expected, because the operations involved in the key expansion algorithm are easy to perform and, therefore, are easy to explain to the viewers.

The Key expansion animation allowed you to understand the inner workings of the process

23 responses

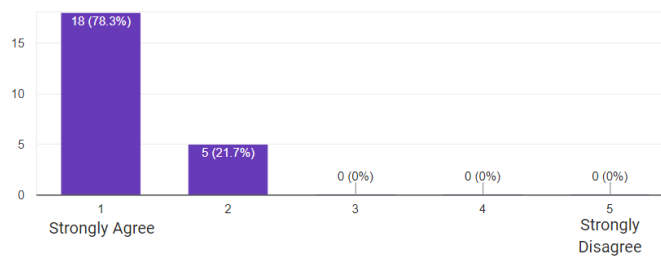


Figure 5.13: Question 8 Results

- The Substitute Bytes step was clear and allowed you to really understand the process

The Substitute Bytes step was an animation inside the encryption animation. This animation explained how the S-Box was used to substitute the bytes in the state. As shown in figure 5.14, the majority of respondents agreed that the animation allowed the viewers to understand how the substitution process works. Only the 4.3% of respondents could not agree or disagree with the fact that this animation was useful, but based on the rest of the answers, it can be said that the animation achieved its goal which was to explain how the substitute bytes step works in the AES cipher.

The substitute bytes step was clear and allowed you to really understand the process

23 responses

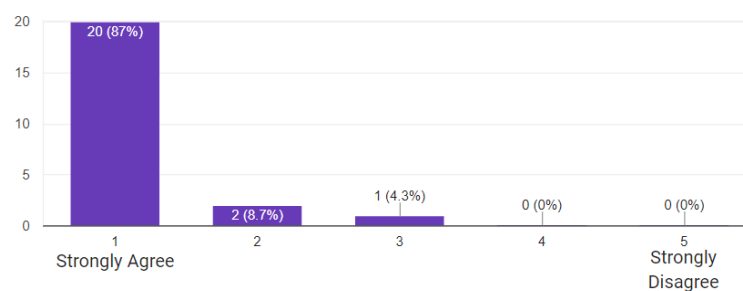


Figure 5.14: Question 9 Results

- The Mix Columns step was clear and allowed you to really understand the process

This is probably the most complicated step to explain because of its mathematical background. Because of this, it was expected that this was going to be the step which had the lowest user acceptance. However, based on figure 5.15, the majority of respondents (56.5%) agreed that this animation was useful to understand the mix columns procedure. Also, some interesting findings could be drawn. The majority of people that had excellent or good knowledge about the AES cipher (Question 1), before using the CryptoSchool program, strongly agreed that this animation was helpful and most of the participants with average fluency level in English (Question 2) could not agree or disagree with the fact that this step was useful. The 4.35% of the 8.7% that disagreed about the usefulness of this step were not fluent in English. Therefore, it can be generalised and said that the people that had most

issues with this step were the ones with an average or below fluency level in English and the participants that were new to the AES cipher. As a result, this step will need to be revisited in the future and changed accordingly so that different texts/audio are used, and maybe different animations can be implemented so that users can take full advantage of the AES animation.

The mix columns step was clear and allowed you to really understand the process
23 responses

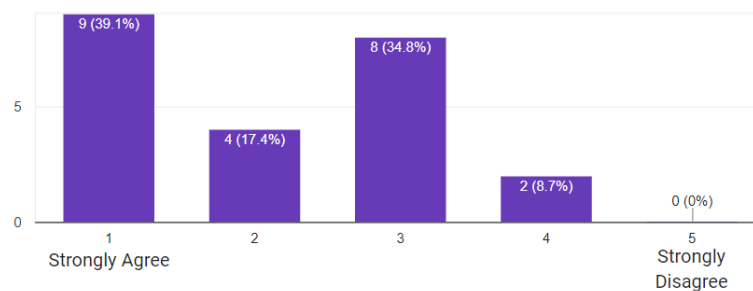


Figure 5.15: Question 10 Results

- The Shift Rows step was clear and allowed you to really understand the process

The shift rows step explains the circular byte shift performed in the encryption process. Because of the nature of this step, it is simple to explain with an animation, and therefore, the 95.7% of respondents found this animation useful. As a result, it can be said that the goal of explaining how the shift rows step works in the AES cipher was met because the majority of participants agreed that they understood the process as shown in the figure below:

The shift rows step was clear and allowed you to really understand the process
23 responses

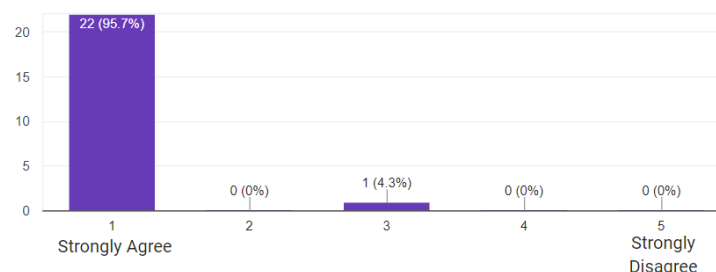


Figure 5.16: Question 11 Results

- Did you learn the differences between the encryption and the decryption processes?

The AES animation explains the encryption process in detail, however, because of the similarities with the decryption process' operations, the individual steps involved in the decryption were not animated. Nevertheless, the differences were highlighted and shown to the user because it is important to understand how different the encryption and decryption processes are even though the operations are basically the same (using the inverse of the operations). As shown in figure 5.17, 100% of the participants agreed that they learned what the differences between encryption and decryption are.

Did you learn the differences between the encryption and decryption processes?
23 responses

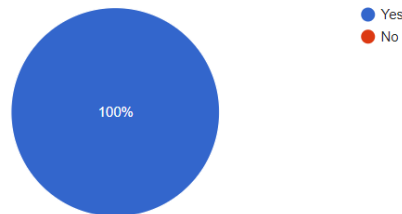


Figure 5.17: Question 12 Results

- Did the colours used help you to better understand the animations?

The colours used in the animations were an important feature that was considered when designing the application. The colours used throughout the whole application helped 100% of the users to better understand the animations (as shown in figure 5.18). This was an important characteristic to analyse because, as it was mentioned in chapter 4.7, colours help to better communicate information. This affirmation was based on the literature and was confirmed with the results shown below:

Did the colors used help you to better understand the animations?
23 responses

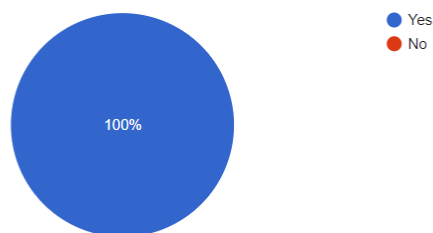


Figure 5.18: Question 13 Results

- The texts on screen helped to better understand the animations

As shown in figure 5.19, 100% of the respondents agreed that the texts on screen helped them to better understand the animations. This is a logic result because without some sort of explanation (either text or voice) the viewers would not have been able to really understand what the animations were doing; especially users that were new to the algorithm would have been confused.

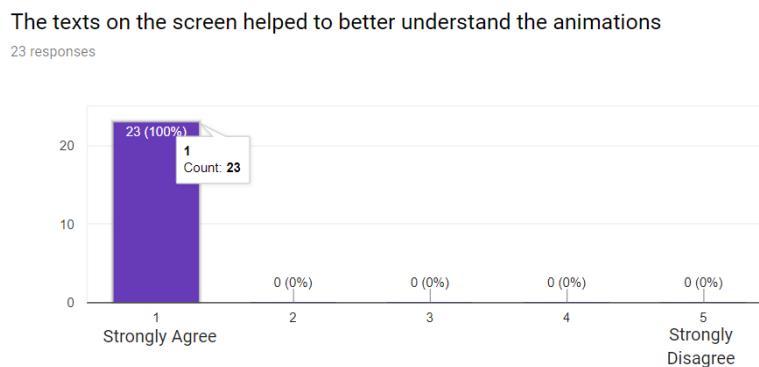


Figure 5.19: Question 14 Results

- Were the audio explanations in the AES animation useful?

This question, as opposed to the previous question, has a variety of results which were not expected. Even though 60.9% of the population agreed that the audio explanations were very useful, 21.7% percent could not agree if it was useful or not, and the other 17.4% agreed that it was useful (option 2 in the scale of 1 to 5). Therefore, a deeper analysis was performed. It was found that out of the 39.1% that did not choose the 'very useful' option, 66.7% had an average fluency level in English or less than average. This raises a warning, because probably the audio did not have a neutral English accent, or probably it was too fast, therefore, some changes might need to be done in future versions of the application. Nonetheless, having a 60.9% of participants that agreed on the fact that the audios were really useful means that the goal of using audios to increase the learning experience worked for the majority of the population, but there is still room for improvement. The results can be seen in figure 5.20:

Were the audio explanations in the AES animation useful?

23 responses

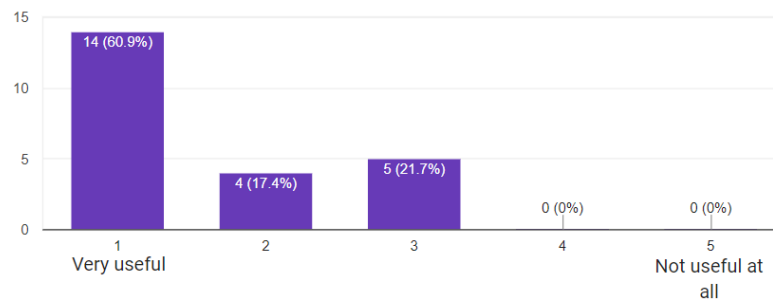


Figure 5.20: Question 15 Results

- Did the evaluation helped to self-assess your knowledge regarding the AES cipher?

The self-assessment feature is one of the main characteristics of the program because it is not present in any other of the related e-learning programs that were analysed. The questionnaire found that 91.3% strongly agreed with the fact that the evaluation helped the users to self-assess their knowledge and the 8.7% agreed that it was useful (2 in a scale of 1 to 5). Therefore, having this feature was useful and helps the developed program to differentiate from the rest of the cryptographic teaching tools on the market.

Did the evaluation helped to self-assess your knowledge regarding the AES cipher?

23 responses

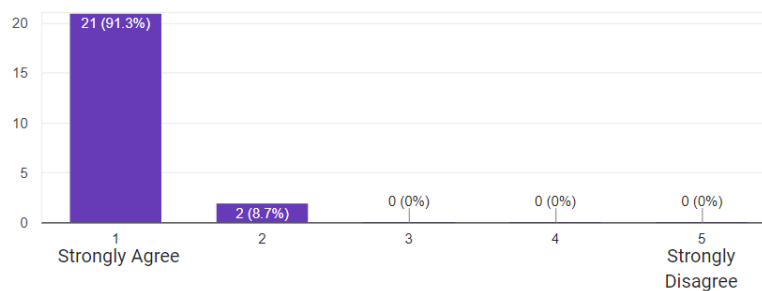


Figure 5.21: Question 16 Results

- Would you recommend CryptoSchool to someone who wants to learn the AES cipher?

This question of the survey asked the respondents to answer if they would recommend the use of CryptoSchool, and the 100% of the participants said Yes. If all the participants would

recommend this software to someone who wants to learn about the AES algorithm, it means that the respondents were pleased with the teaching capabilities of the program and that they learned how the AES cipher works. The results can be seen in the following figure:

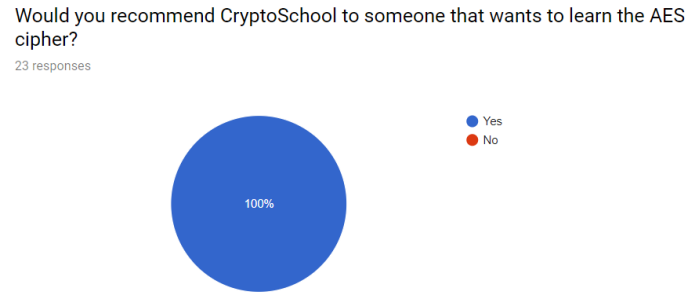


Figure 5.22: Question 17 Results

- After using CryptoSchool, how good is your knowledge regarding the AES cipher?

This was the last question in the survey and helped to validate the effectivity of the developed program. A similar question was asked to the respondents (question 3) but the results obtained were substantially different, which show CryptoSchool's success in the teaching of the AES cipher. Before, the 60.8% (based on question's 3 results) of respondents considered that they had a very poor or poor knowledge regarding the AES cipher. The interesting fact is that that percentage lowered down to 0% after the participants used CryptoSchool, which can be seen in figure 5.23. Also, after using CryptoSchool, the majority of respondents (73.9%) considered to have an excellent or good knowledge of AES cipher, percentage that before (question 3 results) was only 26%. This validates the fact that CryptoSchool is an excellent mechanism for students to learn about the AES cipher.

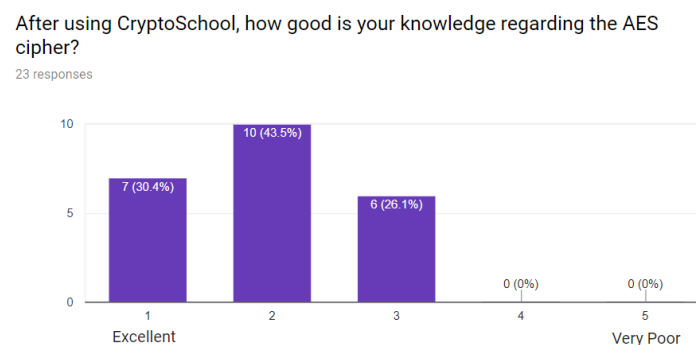


Figure 5.23: Question 18 Results

5.7 Summary

This chapter evaluated the developed tool in terms of its performance, security, compatibility and effectiveness. The results of the performance tests were successful and showed that the animations did not have any stuttering issues and that the desired frame rate was obtained. The results of the performance tests were corroborated with one of the questions in the questionnaire in which 100% of the respondents agreed that they did not suffer any performance issues. The security test showed that there were some medium and low risk security issues that were fixed in the final version of the program. Also, the compatibility tests revealed that the website can be correctly seen in the latest versions of the most popular browsers. Furthermore, a table showing the differences between the developed program and related e-learning tools was developed which helped to find some advantages and disadvantages of CryptoSchool. Lastly, the analysis of the questionnaire revealed interesting conclusions, which in summary state that the CryptoSchool tool is successful in the teaching of the AES cipher but some features could be improved.

Chapter 6: Conclusions and Future Work

6.1 Chapter Overview

This chapter focuses on summarising the work done for this project and also states the future improvements that can be done to enhance the teaching capabilities of the developed product.

6.2 Conclusions

The aim of the project was to create a software that helped MSc. ACS students to learn about the AES cipher. In order to achieve this aim, several objectives were established.

- Firstly, a deep literature review was performed to learn and understand about the history behind cryptography and the intrinsic aspects of the AES cipher. This was helpful because it allowed to have the theoretical knowledge required to create the software.
- Secondly, an analysis of the development tools and languages was performed, including the considerations on how an animation could help in the teaching of the cipher. This permitted to choose the most appropriate programming languages and tools to develop the animation and web site.
- Thirdly, the implementation took place. In this phase, a series of teaching features were analysed and implemented in the developed program to improve the student's learning process. The development methodology and designs were examined and detailed explanations of the program were specified. The final design of the website (<http://personalpages.manchester.ac.uk/postgrad/richard.leiva/dissert/index.html>) consists of two tabs (introduction and animation) and the animation consists of 4 parts, all of which help in the teaching of the AES cipher. The History and Structure

was the first section of the animation and gives the users an introduction to the AES cipher. The second part consists of the Key Expansion, Encryption and Decryption animations which help the user to understand the intrinsic aspects of each of the processes. The third section is the Solver, that allows the user to play with the encryption or decryption processes by inputting a message and a key and obtaining the encrypted or decrypted value. The last part of the program is the Evaluation, which allows the users to respond to 8 questions and self-assess their knowledge. All the features used like the colours, the voice instructions, the menus, the sections of the program, among others, were chosen based on the benefits that they give to the students in terms of the usability and learning capabilities, which proved to be useful based on the analysis of the questionnaire.

- Fourthly, the developed program was evaluated to assess the effectiveness of the project. These evaluations consisted of performance, security and compatibility tests, as well as the analysis of a questionnaire filled out by MSc. ACS students. All the tests passed successfully and this was confirmed with the questionnaire's results. With the results of the questionnaire it can be concluded that the user satisfaction is outstanding and that the teaching benefits of the software are clearly visible. The questionnaire also revealed some interesting facts that can be used to improve the animations and that will be discussed in chapter 6.3.

6.3 Future Work

Even though the results of the tests and the analysis of the questionnaire were successful, some of the features of the developed software can be improved. Some of the features that can be improved were found with the results of the questionnaire, and other features that can be implemented were considered in the beginning of the project, however, because of the time constraints imposed to this kind of projects, they were not implemented. Therefore, some recommendations for future works are detailed below:

- CryptoSchool can contain more animated algorithms than just AES. Because of the nature of the website, adding new algorithms is easy. The developer will only need to create a tab for the new algorithm and embed it in the web page (like it was done with the AES animation).

- Because the AES animation was built using Flash, the animations cannot be seen in Apple mobile devices. Therefore, if the user base expands and it is required to support these kind of devices, the animation will have to be migrated to HTML5 and JS.
- Based on the questionnaire analysis, the Mix Columns Step animation should be revisited. In order for users to really understand this step, it is necessary to give them an introduction to Galois Field arithmetic. Therefore, an introduction to this topic is necessary before the mix columns animations takes place, so that users can fully benefit from the visual aids that the animation provides.
- The audios used to explain the animations can be improved so that people that are not fluent in English can benefit more. These improvements can include making the audios have a more neutral English accent and using a slower pace.
- A focused teaching approach can be implemented, in which users get asked in the beginning of the animation about the current knowledge they possess. Depending on the user's knowledge, the GUI and functionality change, so that users with different expertise level can focus on some parts of the animation more than others, which will enhance the learning process of students.

References

- Acker, S. V., Nikiforakis, N., Desmet, L., Joosen, W., & Piessens, F. (2012). *FlashOver: Automated Discovery of Cross-site Scripting Vulnerabilities in Rich Internet Applications*. Leuven: Katholieke Universiteit Leuven.
- Adobe. (2017). *SWF Profiling Tool*. Retrieved April 07, 2017, from Adobe Scout CC: <http://www.adobe.com/uk/products/scout.html>
- Adobe Animate CC. (n.d.). *Buy Adobe Animate CC*. Retrieved July 15, 2017, from Adobe Systems Incorporated: <http://www.adobe.com/uk/products/animate.html>
- Adobe. (n.d.). *Statistics | Adobe*. Retrieved July 14, 2017, from Adobe: <http://www.adobe.com/products/flashruntimes/statistics.html>
- Ainsworth, S., & VanLabeke, N. (2004). Multiple forms of dynamic representation. *Learning and Instruction, 14*, 241–255.
- Al-Vahed, A., & Sakhavi, H. (2011, April). An overview of modern cryptography. *World Applied Programming, 1*(1), 55-61. ISSN: 2222-2510
- Anthes, G. (2012). HTML5 leads a web revolution. *Communications of the ACM, 55*(7), 16-17.
- Ayushi. (2010). A Symmetric Key Cryptographic Algorithm. *International Journal of Computer Applications, 1*(15), 1-4.
- Bellare, M., Desai, A., Jokipii, E., & Rogaway, P. (1997). A Concrete Security Treatment of Symmetric Encryption. *Foundations of Computer Science, 394-403*.
- Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., & Shamir, a. A. (2009). *Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds*. Cryptology ePrint Archive.

- Blender. (n.d.). *About*. Retrieved July 7, 2017, from Blender: <https://www.blender.org/about/>
- Brown, M., & Hershberger, J. (1992). Color and Sound in Algorithm Animation. *IEEE Computer*, 25(12), 52-63. doi:10.1109/2.179117
- Buzzetto-More, N. (2015). Student Attitudes Towards The Integration Of YouTube In Online, Hybrid, And Web-Assisted Courses: An Examination Of The Impact Of Course Modality On Perception. *MERLOT Journal of Online Learning and Teaching* , 55-73.
- Catalano, D., Fiore, D., Gennaro, R., & Vamvourellis, K. (2015). Algebraic (trapdoor) one-way functions: Constructions and applications. *Theoretical Computer Science*, 143–165.
- Chitu, C., & Glesner, M. (2005). An FPGA implementation of the AES-Rijndael in OCB/ECB modes of operation. *Microelectronics Journal* , 139–146.
- CISCO. (2017). *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*. California: CISCO Visual Networking Index.
- Collins, J. (2004). Education Techniques for Lifelong Learning. Giving a PowerPoint Presentation: The Art of Communicating Effectively. *RadioGraphics*, 24(4), 1185-1192.
- Cryptography. (n.d.). In *Cambridge Dictionary*. Retrieved June 21, 2017, from <http://dictionary.cambridge.org/dictionary/english/cryptography>
- CryptoWorkflow. (n.d.). *CryptoWorkflow*. Retrieved March 29, 2017, from Google Code: <https://code.google.com/archive/p/cryptoworkflow/>
- Davis, R. C., Colwell, B., & Landay, J. A. (2008). K-Sketch: A “Kinetic” Sketch Pad for Novice Animators . *SIGCHI Conference on Human Factors in Computing Systems*, 413-422.
- Devi.T, R. (2013). Importance of Cryptography in Network Security. *International Conference on Communication Systems and Network Technologies*, 462-467.
- Didla, S., Ault, A., & Bagch, S. (2008). Optimizing AES for embedded devices and wireless sensor networks. *Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*. Innsbruck, Austria: Center for Wireless Systems and Applications (CWSA).

- Dooley, J. F. (2013). *A Brief History of Cryptology and Cryptographic Algorithms*. Galesburg: SpringerBriefs in Computer Science. doi:10.1007/978-3-319-01628-3_2
- Duffy, P. (2008). Engaging the YouTube Google-Eyed Generation: Strategies for Using Web 2.0 in Teaching and Learning . *Electronic Journal e-Learning*, 06(02).
- Elminaam, D. S., Kader, H. M., & Hadhoud, M. M. (2010). Evaluating The Performance of Symmetric Encryption Algorithms. *International Journal of Network Security*, 10(3), 213–219.
- Fahmy, A., Shaarawy, M., El-Hadad, K., Salama, G., & Hassanain, K. (2005). A Proposal For A Key-Dependent AES. *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications* .
- Flavell, L. (2010). *Beginning Blender: Open Source 3D Modeling, Animation, and Game Design* . New York: Apress.
- Gaj, K., & Orłowski, A. (2003). Facts and Myths of Enigma: Breaking Stereotypes. *International Conference on the Theory and Applications of Cryptographic Techniques*, 106-122. doi:10.1007/3-540-39200-9_7
- Garrison, R., & Cleveland-Innes, M. (2005). Facilitating Cognitive Presence in Online Learning: Interaction Is Not Enough. *The American Journal of Distance Education*, 19(03), 133-148.
- Granado-Criado, J. M., Vega-Rodríguez, M. A., Sanchez-Perez, J. M., & Gomez-Pulido, J. A. (2010). A new methodology to implement the AES algorithm using partial and dynamic reconfiguration. *INTEGRATION, the VLSI journal*, 72-80.
- Hick, S., Esslinger, B., & Wacker, A. (2012). Reducing the complexity of understanding cryptology using CrypTool. *In Proc of the 10th International Conference on Education and Information Systems Technologies and Applications*.
- Höffler, T. N., Prechtel, H., & Nerdel, C. (2010). The influence of visual cognitive style when learning from instructional animations and static pictures. *Learning and Individual Differences*, 20, 479–483.

- Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004). Software Quality and Agile Methods. *Proceedings of the 28th Annual International Computer Software and Applications Conference* .
- Hurlant. (n.d.). *AS3 Cryptography Library*. Retrieved April 06, 2017, from As3Crypto: <http://crypto.hurlant.com/>
- International Telecommunication Union*. (2015). Retrieved March 19, 2017, from United Nations: <https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf>
- Janitor, J., Jakab, F., & Kniewald, K. (2010). Visual Learning Tools for Teaching/Learning Computer Networks . *Sixth International Conference on Networking and Services*, 351-355.
- Katz, J., & Lindell, Y. (2015). *Introduction to Modern Cryptography, Second Edition*. Florida: CRC Press.
- Kaushik, S., & Singhal, A. (2012). Network Security Using Cryptographic Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 02(12), 105-107.
- Koroghlanian, C., & Klein, J. (2004). The Effect of Audio and Animation in Multimedia Instruction. *Journal of Educational Multimedia and Hypermedia*, 13(1), 23-46.
- Kumar, Y., Munjal, R., & Sharma, H. (2011). Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures. *IJCSMS International Journal of Computer Science and Management Studies*, 11(3), 60-63.
- Lee, D. Y., & Lehto, M. (2013). User acceptance of YouTube for procedural learning: An extension of the Technology Acceptance Model. *Computers & Education*, 61, 192-208.
- Liao, Z., Forsyth, D., Yu, Y., & Hoppe, H. (2012). A Subdivision-Based Representation for Vector Image Editing. *IEEE Transactions On Visualization and Computer Graphics*, 18(11), 1858-1867.
- Loussios, M. K. (2014). Cryptool 2 in Teaching Cryptography. *Journal of Computations &*

Modelling, 349-358. ISSN: 1792-7625

- Mangard, S. (2002). A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion. (Springer, Ed.) *Information Security and Cryptology — ICISC, 2587*, 343-358.
- Masood, J. (2012). *Protect your data at the speed of light with gKrypt*. IBM. DeveloperWorks.
- McDonald, N. G. (2010). *PAST, PRESENT, AND FUTURE METHODS OF CRYPTOGRAPHY AND DATA ENCRYPTION*. Logan: University of Utah.
- McMains, S., & Kastner, S. (2011). Interactions of top-down and bottom-up mechanisms in human visual cortex. *Journal of Neuroscience*, 587-597.
- McMillan, J., & Hearn, J. (2008). Student Self-Assessment: The Key to Stronger Student Motivation and Higher. *Educational Horizons*, 87(1), 40-49.
- Minaam, D. S., Abdual-Kader, H., & Hadhoud, M. M. (2010). Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types. *International Journal of Network Security*, 11(2), 78–87.
- Montgomery, D., & Runger, G. (2014). *Applied Statistics and Probability for Engineers* (6th ed.). Hoboken: John Wiley & Sons.
- O’Day, D. H. (2007). The Value of Animations in Biology Teaching: A Study of Long-Term Memory Retention. *CBE - Life Sciences Education*, 6, 217-223. doi: 10.1187/cbe.07–01–0002
- Phung, P. H., Monshizadeh, M., Sridhar, M., Hamlen, K. W., & Venkatakrishnan, V. “. (2015). Between Worlds: Securing Mixed JavaScript/ActionScript Multi-Party Web Content. *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, 443-457.
- Rasnacisa, A., & Berzisa, S. (2017). Method for Adaptation and Implementation of Agile Project Management Methodology. *Procedia Computer Science*, 104, 43 – 50.
- Reimers, S., & Stewart, N. (2015). Presentation and response timing accuracy in Adobe Flash and HTML5/JavaScript Web experiments. *Behavior Research Methods*, 47(2), 309–327.

- Robling, G., Schuler, M., & Freisleben, B. (200). The ANIMAL algorithm animation tool. *5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education*, 37-40.
- Rocca, C. F. (2014). Mathematics in the History of Cryptography. *Cryptologia*, 232-243. doi:10.1080/01611194.2014.915254
- Roodt, S., & Peier, D. (2013). Using Youtube© in the Classroom for the Net Generation of Students. *Issues in Informing Science and Information Technology*, 10, 473-488.
- Russel, K. (2006). Flash. *ComputerWorld*, 32.
- Selent, D. (2010). Advanced Encryption Standard. *Rivier Academic Journal*, 1-14.
- Sharif, S. O., & Mansoor, S. (2010). Performance analysis of Stream and Block cipher algorithms . *3rd International Conference on Advanced Computer Theory and Engineering*, 522-525.
- Singh, S. (2001). *The Code Book*. New York: Delacorte Press.
- Singh, S. (2003). *The History of Cryptography: How the history of codebreaking can be used in the mathematics classroom with resources on a new CD-ROM*. United Kingdom Mathematical Association.
- Singhal, N., & J.P.S.Raina. (2011). Comparative Analysis of AES and RC4 Algorithms for Better Utilization. *International Journal of Computer Trends and Technology*, 177-181.
- Song, X., & Deng, H. (2009). Taking Flexible and Diverse Approaches to Get Undergraduate Students Interested in Cryptography Course. *Education Technology and Computer Science*, 490-494. doi:10.1109/ETCS.2009.371
- Stallings, W. (2014). *Cryptography and Network Security Principles and Practice*. New Jersey: Pearson Education.
- Thawte. (2013). *History of Cryptograpy - An Easy to Understand History of Cryptography*. Thawte.
- Tripathi, R., & Agrawal, S. (2014). Comparative Study of Symmetric and Asymmetric

Cryptography Techniques. *International Journal of Advance Foundation and Research in Computer*, 1(6), 68-76.

Truong, Y. (2009). An Evaluation of the Theory of Planned Behaviour in Consumer Acceptance of Online Video and Television Services. *The Electronic Journal Information Systems Evaluation*, 12(2), 177-186.

Tversky, B., Morrison, J., & Betrancourt, M. (2002). Animation: Can it facilitate? *International Journal of Human Computer Studies*, 57, 247–262.

Vaughan-Nichols, S. J. (2010). Will HTML 5 Restandardize the Web? *IEEE Computer Society*, 43(4), 13-15.

Appendix A: Important Code Snippets

```
private function syncAndPauseAudio():void{
    var soundTime = this.timeSlider.value * mastertl.duration();
    sound.gotoSoundTime(soundTime, true);
    sound.pauseSound()
}
```

Figure A.1: Method used to sync the audio with the animation whenever the progress slider is changed by the user.

```
private function PlayButtonClicked(e:MouseEvent) : void{
    mastertl.play();
    sound.playSound();
}

private function PauseButtonClicked(e:MouseEvent) : void{
    mastertl.pause();
    sound.pauseSound();
}

private function ForwardButtonClicked(e:MouseEvent) : void{
    mastertl.seek(mastertl.getLabelAfter(), false).play();
    syncAndPlayAudio();
}

private function BackgroundButtonClicked(e:MouseEvent) : void{
    mastertl.seek(mastertl.getLabelBefore(mastertl.time() - 1), false).play();
    syncAndPlayAudio();
}

private function sliderValueChanged(e:SliderEvent):void{
    mastertl.progress(this.timeSlider.value).pause();
    syncAndPauseAudio();
}
```

Figure A.2: Methods that show the benefits of manipulating a master timeline instead of having to deal with individual timelines. In the Figure, the variable `mastertl` represents the master timeline.

```

public function encrypt(block:ByteArray, index:uint=0):void
{
    var round:uint;
    state.position=0;
    state.writeBytes(block, index, Nb*4);

    addRoundKey(key, 0);
    for ( round = 1; round < Nr + 1; round++ ) {
        if (round < Nr) {
            mixSubColumns();
        } else {
            shiftRows();
        }
        addRoundKey(key, round * Nb * 4);
    }

    block.position=index;
    block.writeBytes(state);
}

```

Figure A.3: AS3 Crypto library code used to encrypt a block of code using the AES cipher

```

public function encryptAES128(block:ByteArray, index:uint=0):Object
{
    var resultsArray:Array = new Array();
    var map:Object = new Object();
    var round:uint;
    state.position=0;
    state.writeBytes(block, index, Nb * 4);
    addRoundKey(key, 0);
    resultsArray.push(Hex.fromArray(state));
    map["round0"] = resultsArray;
    resultsArray = new Array();
    for ( round = 1; round < Nr + 1; round++ ) {
        subBytes();
        resultsArray.push(Hex.fromArray(state));
        if (round < Nr) {
            shiftRows();
            resultsArray.push(Hex.fromArray(state));
            mixSubColumns();
            resultsArray.push(Hex.fromArray(state));
        } else {
            shiftRows();
            resultsArray.push(Hex.fromArray(state));
        }
        addRoundKey(key, round * Nb * 4);
        resultsArray.push(Hex.fromArray(state));
        map["round" + round] = resultsArray;
        resultsArray = new Array();
    }

    block.position=index;
    block.writeBytes(state);
    return {Values:map,Key:Hex.fromArray(key)};
}

```

Figure A.4: Similar to A.3 but with some changes to accommodate to the needs of the solver.

```

private function addMainMenuButtonsClickHandler() : void
{
    this.btHistoryAndExplanation.addEventListener(MouseEvent.CLICK, this.btnHistoryAndExplanationClickHandler);
    this.btAnimation.addEventListener(MouseEvent.CLICK, this.btnAnimationClickHandler);
    this.btSolver.addEventListener(MouseEvent.CLICK, this.btnSolverClickHandler);
    this.btEvaluation.addEventListener(MouseEvent.CLICK, this.btnEvaluationClickHandler);
}

```

Figure A.5: Main Menu Event Listeners

```

private function historyFourthAnimation():void{
    TweenLite.set([this.txtHistory10, this.txtHistory11, this.txtHistory12, this.txtHistory13, this.txtHistory14], {x:20, y:70, alpha:0});
    var tl = new TimelineLite();
    tl.to(this.txtHistory9, 2, {alpha:0});
    tl.to(this.txtHistory11, 3, {alpha:1});
    mastertl.add(tl, "+=2");
    var tl2 = new TimelineLite();
    tl2.to(this.HistoryAESStructure.plaintext, 3, {y:"+=105", ease:Power3.easeInOut},1);
    tl2.to(this.HistoryAESStructure.k1, 3, {x:"-400", ease:Power3.easeInOut}, 1);
    mastertl.add(tl2, "HistoryAnimation4PreRound");
    var tl3 = new TimelineLite();
    tl3.to(this.txtHistory11, 2, {alpha:0});
    tl3.to(this.txtHistory10, 3, {alpha:1});
    mastertl.add(tl3, "+=2");
    var tl4 = new TimelineLite();
    tl4.to(this.HistoryAESStructure.plaintext, 3, {y:"+=78", ease:Power4.easeInOut},"a");
    tl4.to(this.HistoryAESStructure.k2, 3, {x:"-350", ease:Power4.easeInOut},"a");
    tl4.to(this.HistoryAESStructure.plaintext, 3, {y:"+=68", ease:Power4.easeInOut},"b");
    tl4.to(this.HistoryAESStructure.k3, 3, {x:"-350", ease:Power4.easeInOut}, "b");
    mastertl.add(tl4, "HistoryAnimation4Round1-9");
    var tl5 = new TimelineLite();
    tl5.to(this.txtHistory10, 2, {alpha:0});
    tl5.to(this.txtHistory12, 3, {alpha:1});
    mastertl.add(tl5, "+=2");
    var tl6 = new TimelineLite();
    tl6.to(this.HistoryAESStructure.plaintext, 3, {y:"+=120", ease:Power4.easeInOut},"a");
    tl6.to(this.HistoryAESStructure.k4, 3, {x:"-350", ease:Power4.easeInOut},"a");
    tl6.to(this.HistoryAESStructure.plaintext, 1, {alpha:0},"b");
    tl6.to(this.HistoryAESStructure.cyphertext, 1, {alpha:1}, "b");
    tl6.to(this.HistoryAESStructure.cyphertext, 3, {y:"+=80", ease:Power4.easeInOut});
    mastertl.add(tl6, "HistoryAnimation4Round10");
    var tl7 = new TimelineLite();
    tl7.to(this.txtHistory12, 2, {alpha:0});
    tl7.to(this.txtHistory13, 3, {alpha:1});
    mastertl.add(tl7, "+=2");
    var tl8 = new TimelineLite();
    tl8.to(this.HistoryAESStructure.cyphertext, 3, {x:this.HistoryAESStructure.plaintext2.x, ease:Power4.easeInOut});
    tl8.to(this.HistoryAESStructure.cyphertext, 3, {y:"-80", ease:Power4.easeInOut},"a");
    tl8.to(this.HistoryAESStructure.k5, 3, {x:"+=350", ease:Power4.easeInOut}, "a");
    mastertl.add(tl8, "HistoryAnimation4Decryption2");
    var tl9 = new TimelineLite();
    tl9.to(this.txtHistory13, 2, {alpha:0});
    tl9.to(this.txtHistory14, 3, {alpha:1});
    mastertl.add(tl9, "+=2");
    var tl10 = new TimelineLite();
    tl10.to(this.HistoryAESStructure.cyphertext, 3, {y:"-55", ease:Power4.easeInOut},"a");
    tl10.to(this.HistoryAESStructure.k6, 3, {x:"+=350", ease:Power4.easeInOut}, "a");
    tl10.to(this.HistoryAESStructure.cyphertext, 3, {y:"-55", ease:Power4.easeInOut},"b");
    tl10.to(this.HistoryAESStructure.k7, 3, {x:"+=350", ease:Power4.easeInOut}, "b");
    tl10.to(this.HistoryAESStructure.cyphertext, 3, {y:"-160", ease:Power4.easeInOut},"c");
    tl10.to(this.HistoryAESStructure.k8, 3, {x:"+=350", ease:Power4.easeInOut}, "c");
    tl10.to(this.HistoryAESStructure.cyphertext, 1, {alpha:0},"d");
    tl10.to(this.HistoryAESStructure.plaintext2, 1, {alpha:1}, "d");
    tl10.to(this.HistoryAESStructure.plaintext2, 3, {y:"-80", ease:Power4.easeInOut});
    mastertl.add(tl10, "HistoryAnimation4Decryption3");
}

```

Figure A.6: Code for a complex animation that demonstrate the use of the 'to' method of the TimelineLite class

```

private function shiftRowsAnimation(row:int, array:Array):TimelineMax{
    var tmpArray = array.concat();
    var tl = new TimelineMax();
    for(var k = row*4; k < row*4+4; k++){
        tl.to(array[k], 3, {x: array[k].x+200},0);
    }
    switch (row){
        case 1:
            tl.to(array[7],3,{bezier:{
                type: "thru",
                curviness: 2,
                values: [
                    {x:(array[7].x-tmpArray[6].x)/2, y:-50},
                    {x:tmpArray[6].x, y:tmpArray[6].y}
                ]
            }}, rotation:360
            ));
            .to(array[6],3,{bezier:{
                type: "thru",
                curviness: 2,
                values: [
                    {x:(array[6].x-tmpArray[5].x)/2, y:-50},
                    {x:tmpArray[5].x, y:tmpArray[5].y}
                ]
            }}, rotation:360
            ));
            .to(array[5],3,{bezier:{
                type: "thru",
                curviness: 2,
                values: [
                    {x:(array[5].x-tmpArray[4].x)/2, y:-50},
                    {x:tmpArray[4].x, y:tmpArray[4].y}
                ]
            }}, rotation:360
            ));
            .to(array[4],3,{bezier:{
                type: "thru",
                curviness: 2,
                values: [
                    {x:(tmpArray[7].x-array[4].x)/2, y:-50},
                    {x:tmpArray[7].x, y:tmpArray[7].y}
                ]
            }}, rotation:360
            ));
            break;
        case 2:
            tl.to(array[11],3,{bezier:{
                type: "thru",
                curviness: 2,
                values: [
                    {x:(array[11].x-tmpArray[9].x)/2, y:-50},
                    {x:tmpArray[9].x, y:tmpArray[9].y}
                ]
            }}, rotation:360
            ));
    }
}

```

Figure A.7: Extract of the method used to create the reusable XOR animations in the whole program. The reusability of the code allowed to save time and effort while developing the animations.

Appendix B: Source Labs Results

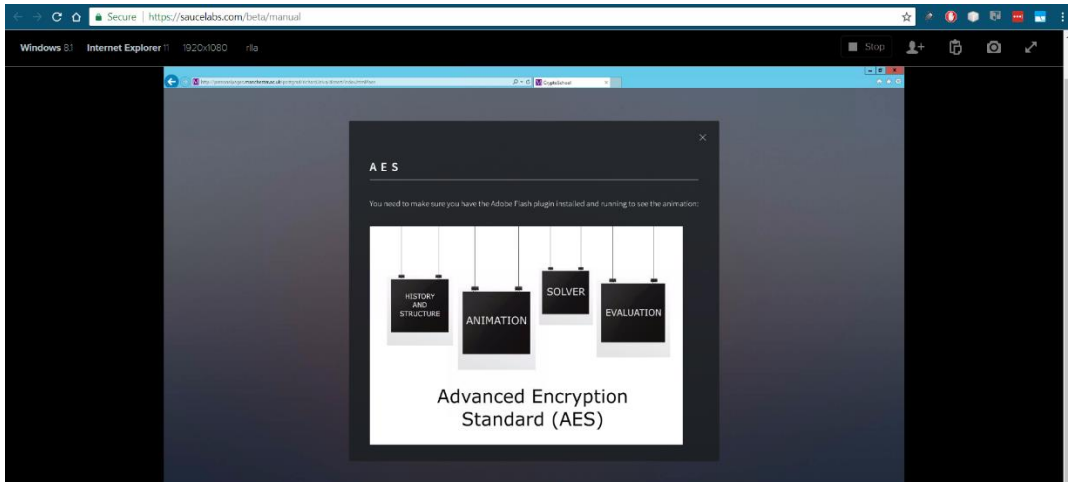


Figure B.1: Screenshot of how the animation looks on Internet Explorer 11 running in a Windows 8.1 Machine

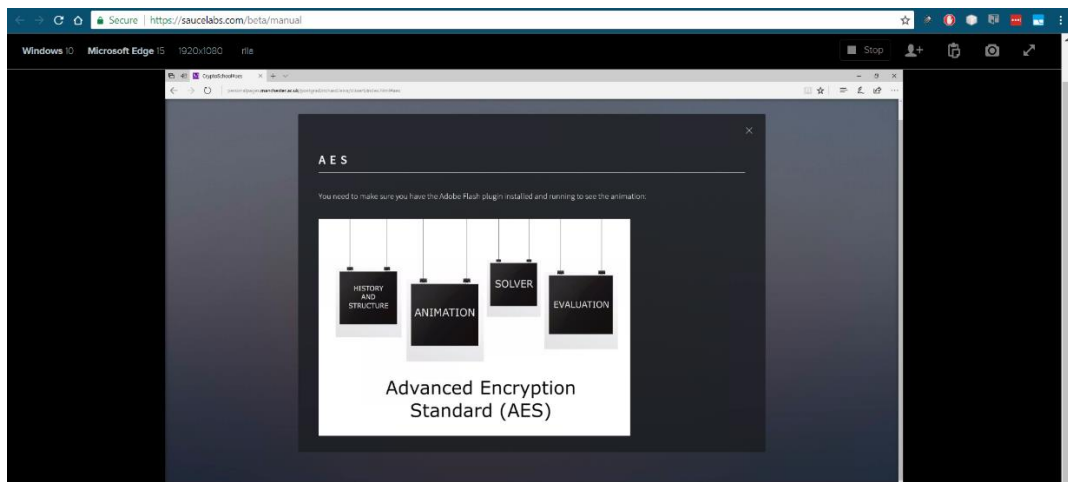


Figure B.2: Screenshot of how the animation looks on Microsoft Edge 15 running in a Windows 10 Machine

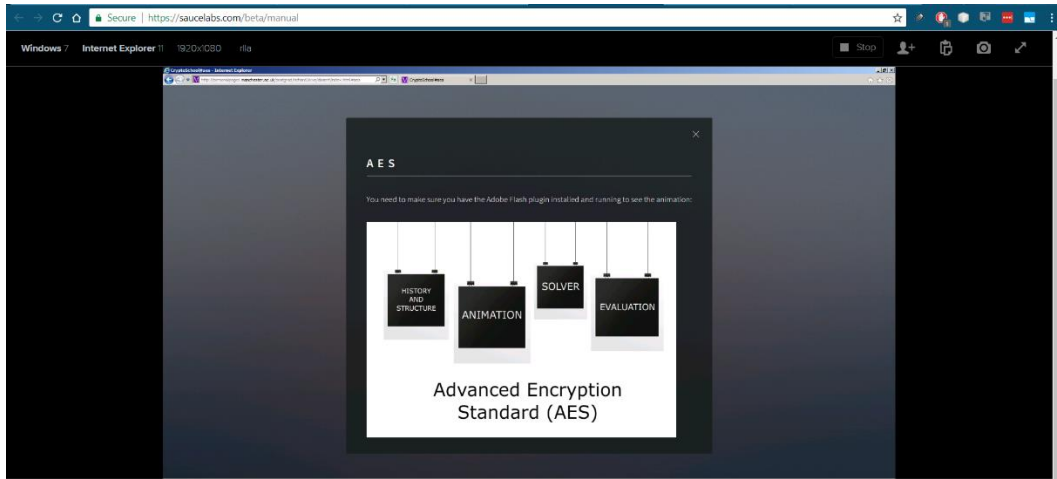


Figure B.3: Screenshot of how the animation looks on Internet Explorer 11 running in a Windows 7 Machine

Appendix C: Questionnaire

Animating Cryptographic Primitives

This is an anonymous and optional questionnaire that must be filled out only after using the CryptoSchool website
<http://personalpages.manchester.ac.uk/postgrad/richard.leiva/dissert/index.html>

* Required

Are you an MSc. ACS Student? *

Yes
 No

How fluent are you in English? *

1 2 3 4 5

Fluent Not Fluent

How good is your knowledge regarding the AES algorithm?

1 2 3 4 5

Excellent Very poor

How easy was it to use the website? *

1 2 3 4 5

Really Easy Really Hard

How easy was it to use the AES animation? *

1 2 3 4 5

Really Easy Really Hard

Did you experience any performance issues while using the website/animations? *

Yes
 No

The History and Structure section helped you to get an overview of how the AES cipher works *

1 2 3 4 5

Strongly Agree Strongly Disagree

The Key expansion animation allowed you to understand the inner workings of the process *

1 2 3 4 5

Strongly Agree Strongly Disagree

Figure C.1: Questionnaire to evaluate the user satisfaction of the developed software, Part 1

The substitute bytes step was clear and allowed you to really understand the process *

1 2 3 4 5

Strongly Agree Strongly Disagree

The mix columns step was clear and allowed you to really understand the process *

1 2 3 4 5

Strongly Agree Strongly Disagree

The shift rows step was clear and allowed you to really understand the process *

1 2 3 4 5

Strongly Agree Strongly Disagree

Did you learn the differences between the encryption and decryption processes? *

Yes

No

Did the colors used help you to better understand the animations?

Yes

No

The texts on the screen helped to better understand the animations *

1 2 3 4 5

Strongly Agree Strongly Disagree

Were the audio explanations in the AES animation useful? *

1 2 3 4 5

Very usefull Not usefull at all

Did the evaluation helped to self-assess your knowledge regarding the AES cipher? *

1 2 3 4 5

Strongly Agree Strongly Disagree

Would you recommend CryptoSchool to someone that wants to learn the AES cipher? *

Yes

No

After using CryptoSchool, how good is your knowledge regarding the AES cipher? *

1 2 3 4 5

Excellent Very Poor

Figure C.2: Questionnaire to evaluate the user satisfaction of the developed software, Part 2

Timestamp	Are you an MSc. ACS Student?	How fluent are you in English?	How good was your knowledge regarding the AES cipher before using CryptoSchool	How easy was it to use the website?	How easy was it to use the AES animation?	Did you experience any performance issues while using the website/animations?
8/1/2017 14:59:39	Yes	1	4	2	1	No
8/1/2017 17:09:42	Yes	1	2	1	1	No
8/1/2017 20:15:12	Yes	2	3	1	1	No
8/2/2017 10:11:50	Yes	4	5	1	1	No
8/2/2017 11:37:00	Yes	3	5	1	1	No
8/2/2017 12:13:41	Yes	3	3	1	1	No
8/2/2017 19:18:11	Yes	1	5	1	1	No
8/2/2017 23:06:56	Yes	3	3	1	1	No
8/3/2017 8:10:07	Yes	3	5	1	1	No
8/3/2017 12:10:26	Yes	1	1	1	1	No
8/3/2017 14:11:19	Yes	1	4	1	1	No
8/4/2017 22:10:46	Yes	1	5	1	1	No
8/5/2017 13:07:42	Yes	1	1	1	1	No
8/5/2017 20:32:09	Yes	1	2	1	1	No
8/6/2017 8:20:36	Yes	2	1	1	1	No
8/6/2017 13:05:56	Yes	3	5	1	1	No
8/6/2017 23:09:59	Yes	3	4	1	1	No
8/7/2017 10:15:40	Yes	3	4	1	1	No
8/7/2017 15:10:23	Yes	1	5	1	1	No
8/7/2017 15:36:13	Yes	1	5	1	1	No
8/8/2017 11:42:19	Yes	1	2	1	1	No
8/8/2017 16:39:20	Yes	1	5	1	1	No
8/8/2017 17:08:15	Yes	3	4	1	1	No

Figure C.3: Summary of the results - Question 1 to 6

Timestamp	The History and Structure section helped you to get an overview of how the AES cipher works	The Key expansion animation allowed you to understand the inner workings of the process	The substitute bytes step was clear and allowed you to really understand the process	The mix columns step was clear and allowed you to really understand the process	The shift rows step was clear and allowed you to really understand the process	Did you learn the differences between the encryption and decryption processes?
8/1/2017 14:59:39	1	1	1	1	1	Yes
8/1/2017 17:09:42	1	1	1	1	1	Yes
8/1/2017 20:15:12	1	2	1	2	1	Yes
8/2/2017 10:11:50	2	2	2	4	3	Yes
8/2/2017 11:37:00	1	2	1	3	1	Yes
8/2/2017 12:13:41	3	2	3	4	1	Yes
8/2/2017 19:18:11	1	1	1	1	1	Yes
8/2/2017 23:06:56	1	1	1	1	1	Yes
8/3/2017 8:10:07	1	1	1	3	1	Yes
8/3/2017 12:10:26	1	1	1	1	1	Yes
8/3/2017 14:11:19	1	1	1	3	1	Yes
8/4/2017 22:10:46	1	1	1	2	1	Yes
8/5/2017 13:07:42	1	1	1	1	1	Yes
8/5/2017 20:32:09	1	1	1	1	1	Yes
8/6/2017 8:20:36	1	1	1	3	1	Yes
8/6/2017 13:05:56	1	1	1	3	1	Yes
8/6/2017 23:09:59	1	1	1	3	1	Yes
8/7/2017 10:15:40	1	1	1	2	1	Yes
8/7/2017 15:10:23	1	1	1	1	1	Yes
8/7/2017 15:36:13	1	1	1	3	1	Yes
8/8/2017 11:42:19	1	1	1	1	1	Yes
8/8/2017 16:39:20	1	1	1	2	1	Yes
8/8/2017 17:08:15	1	2	2	3	1	Yes

Figure C.4: Summary of the results - Question 7 to 12

Timestamp	Did the colors used help you to better understand the animations?	The texts on the screen helped to better understand the animations	Were the audio explanations in the AES animation useful?	Did the evaluation helped to self-assess your knowledge regarding the AES cipher?	Would you recommend CryptoSchool to someone that wants to learn the AES cipher?	After using CryptoSchool, how good is your knowledge regarding the AES cipher?
8/1/2017 14:59:39	Yes	1	2	1	Yes	2
8/1/2017 17:09:42	Yes	1	1	1	Yes	1
8/1/2017 20:15:12	Yes	1	2	2	Yes	1
8/2/2017 10:11:50	Yes	1	3	1	Yes	3
8/2/2017 11:37:00	Yes	1	1	1	Yes	2
8/2/2017 12:13:41	Yes	1	1	1	Yes	2
8/2/2017 19:18:11	Yes	1	2	1	Yes	3
8/2/2017 23:06:56	Yes	1	3	1	Yes	2
8/3/2017 8:10:07	Yes	1	1	1	Yes	2
8/3/2017 12:10:26	Yes	1	1	1	Yes	1
8/3/2017 14:11:19	Yes	1	1	1	Yes	2
8/4/2017 22:10:46	Yes	1	1	1	Yes	3
8/5/2017 13:07:42	Yes	1	1	1	Yes	1
8/5/2017 20:32:09	Yes	1	1	1	Yes	1
8/6/2017 8:20:36	Yes	1	1	1	Yes	1
8/6/2017 13:05:56	Yes	1	3	1	Yes	2
8/6/2017 23:09:59	Yes	1	2	1	Yes	2
8/7/2017 10:15:40	Yes	1	3	1	Yes	2
8/7/2017 15:10:23	Yes	1	1	1	Yes	3
8/7/2017 15:36:13	Yes	1	1	2	Yes	3
8/8/2017 11:42:19	Yes	1	1	1	Yes	1
8/8/2017 16:39:20	Yes	1	3	1	Yes	3
8/8/2017 17:08:15	Yes	1	1	1	Yes	2

Figure C.5: Summary of the results - Question 13 to 18