[9144965]

Abstract

[This report will look at the importance of cryptography for meeting the security objectives of the Zambesi website.]  
  
Word count = 1586

[Cryptography]

[A report on the efficacy of using cryptography to protect the Zambesi online store]



Contents

[Figures 1](#_Toc57389082)

[Introduction 2](#_Toc57389083)

[Logic 2](#_Toc57389084)

[Cryptography and its Primitives 3](#_Toc57389085)

[Symmetric-Key Cryptography 5](#_Toc57389086)

[Asymmetric Encryption 6](#_Toc57389087)

[Hashing and MAC 8](#_Toc57389088)

[X.509 Certificate 8](#_Toc57389089)

[TLS 10](#_Toc57389090)

[UML 12](#_Toc57389091)

[Conclusion 13](#_Toc57389092)

[References 14](#_Toc57389093)

# Figures

[Figure 1: A logic table using XOR 2](#_Toc57389073)

[Figure 2: Multiple Venn diagrams showing set theory operations. 3](#_Toc57389074)

[Figure 3: Cryptographic primitives’ 4](#_Toc57389075)

[Figure 4: Showing a basic XOR Cipher 5](#_Toc57389076)

[Figure 5: Diffie-Helman exchange explained with colours. 7](#_Toc57389077)

[Figure 6: A SHA-256 hash of a textbook and the letter ‘a’ 8](#_Toc57389078)

[Figure 7: A partial X.509 certificate. 9](#_Toc57389079)

[Figure 8: TLS 1.2 Handshake 11](#_Toc57389080)

[Figure 9: UML Diagram showing the implementation of encryption and TLS 1.2 12](#_Toc57389081)

# Introduction

60% of small businesses shutdown 6 Months after a cyber-attack (Galvin, 2018). Cryptography is vital in nullifying the effect of a data breach and for meeting the security objectives of the Zambesi website. This report will explain the different cryptographic protocols, primitives and how they fit into the CIA triad (Confidentiality, Integrity, Authentication). It will show how symmetric and asymmetric encryption can be used to ensure that Zambesi’s customer data is secured from attackers. As well as securing the data, it is also vital that the data is legitimate and beyond repudiation. This requires an overview of hashing and its methods. Lastly are certificates, these employ different cryptographic methods to help customers verify that they are on the official Zambesi website.

# Logic

Logic translates human language into mathematics and is the core of not only cryptography but computers. A computer cannot understand a simple sentence such as “cryptography is hard”. For a computer to understand a statement it must use a proposition and then use propositional logic to resolve the statement to true or false. For example, “Yorkshire is a region of England”, this can be resolved to true. There are many connectives that are used with logic e.g. AND, OR, XOR. XOR is an exclusive or.

Table

Description automatically generated

Figure 1: A logic table using XOR

Predicate logic uses quantifiable variables to further expand upon propositional logic. For example, all Yorkshiremen have whippets (A). Bob does not own a whippet (B). Therefore, Bob is not a Yorkshireman. Set theory is a way of creating sets or groups using propositional logic. If you had two sets of data one set is filled with people the other set is filled with whippets, by using the ‘AND’ operator you can then see the Yorkshiremen. In the Venn diagram below Yorkshiremen would be the yellow part under the ‘AND’ operator (Ben-Ari, 2003, p. 3).

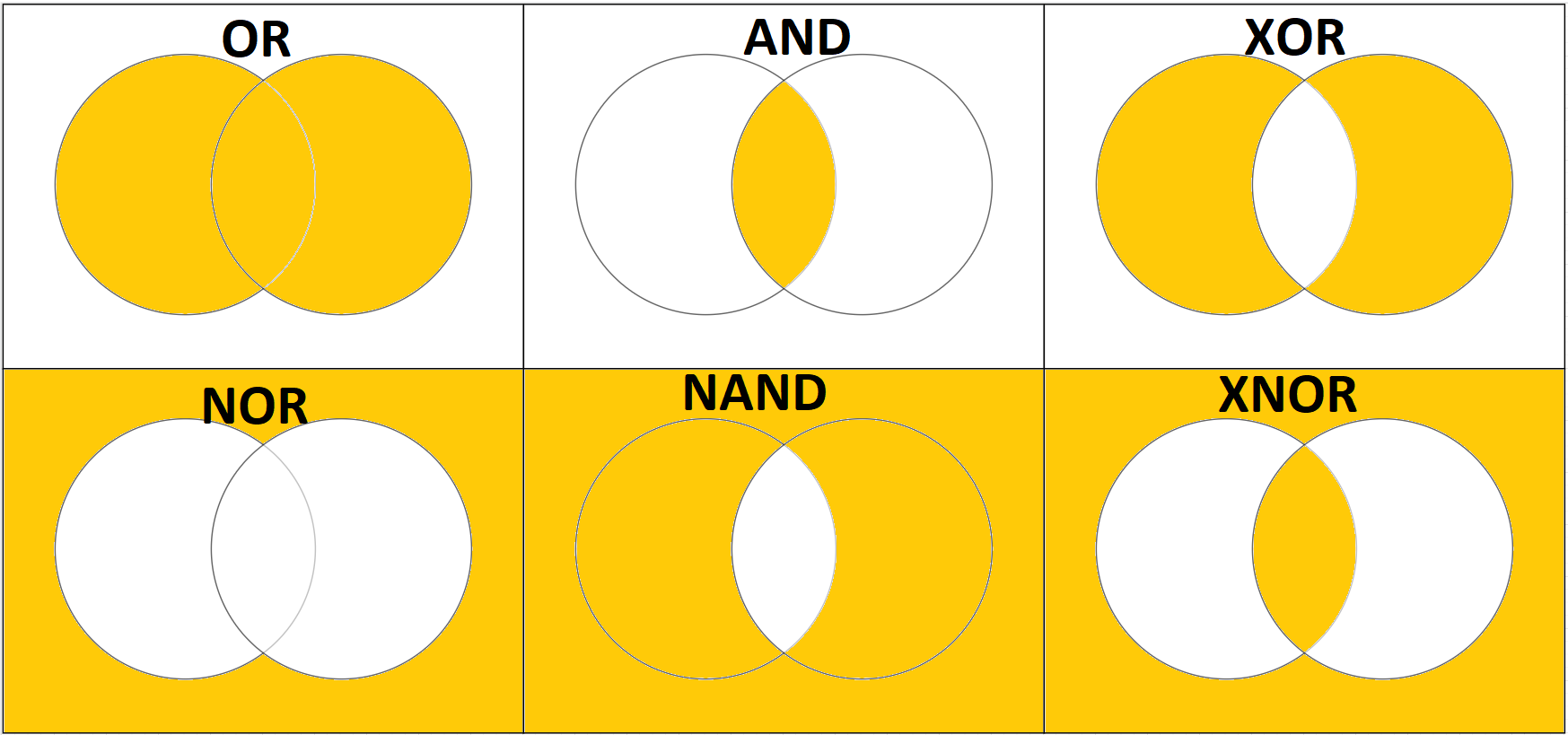


Figure 2: Multiple Venn diagrams showing set theory operations.

# Cryptography and its Primitives

Cryptography is a method of obfuscating data using algorithms, to keep it secure. There are many different methods of cryptography each serving its own purpose.

The basic tools of cryptography are primitives. Primitives are low level algorithms which are used as building blocks to create protocols. Primitives can also be used to build other primitives. For example, a symmetric-key is a primitive which contains other primitives such as block ciphers and stream ciphers.

An important aspect of cryptography is the bit size. This is the size of the cryptographical key. A 4-bit (24) key would have 16 possible key combinations and thus be easy to brute force. A 256-bit key has 78 digits. It would take one of the world’s fastest super computers millions of years to brute force this key (Nohe, 2019).

Due to the complexity of the subject the primitives and associated protocols will be explained first and then how they all come together to create a secure environment. The primitives discussed in this report will be symmetric-key cryptography, asymmetric-key cryptography, hash functions and certificates. (Menezes et al., 1997, p. 5)

Diagram

Description automatically generated

Figure 3: Cryptographic primitives’

Note. From *Handbook Of Applied Cryptology (p. 5),* by Menezes, A., Vanstone, S. and Van Oorschot, P., 1997, CRC.

# Symmetric-Key Cryptography

A symmetric-key cipher uses the same key to both encrypt and decrypt the data. A simple symmetric cipher is the Caesar cipher. The Caesar cipher would shift the letters along 3, so ‘hi’ would become ‘kl’. Knowing the key is +3 you can just shift the letters back 3. The symmetric-key ciphers explained in this report will be block ciphers as used by AES and stream ciphers as used by ChaCha20.

A stream cipher encrypts the data one bit or byte at a time. A basic example is the XOR stream cipher below. In this example you compare the original bits and the key using XOR. If there is a 1 and a 0 then the encrypted bit is a 1. If both bits are the same, then it is a 0. This will encrypt the data. In order to decrypt the data, you repeat the process using the key and the encrypted text.

Table

Description automatically generated

Figure 4: Showing a basic XOR Cipher

The main advantage to using a stream cipher is its speed, the primary disadvantage is they can be vulnerable to malicious code injection. A way to help counter injection is by using a hash-based message authentication code (HMAC) to verify the integrity of the transmitted data (Aumasson, 2018).

Unlike a stream cipher which encrypts individual bytes, a block cipher encrypts the data in blocks. For example, AES encrypts the data in blocks of 128 bits in a 4x4 array. If the data being encrypted is only 50bits then a block cipher will pad the remaining bits. AES uses four functions to transform the data from plaintext to cipher text:

* SubBytes: Substitutes bytes from one array using a substitution box, which is a 16x16 array.
* ShiftRows: Shifts the bytes in each row by a set amount.
* MixColumns: Mixes the bytes in the columns.
* AddRoundKey: Adds a key for each round.

For an AES-128 bit key, it repeats this process for 10 rounds. To create a 256-bit key it repeats the process 14 times, creating a key with high entropy values (Rijmen & Daemen, 2001).

Symmetric-key encryption ensures the data is confidential and protected however it does not authenticate or verify the integrity of the data.

# Asymmetric Encryption

Asymmetric encryption also known as public-key encryption is a method of authenticating users by sharing a private key using public means. As of 2018, Diffie-Helman is the only key exchange protocol used in TLS 1.3 (Rescorla, 2018).

An oversimplified explanation of a Diffie-Helman key is. Alice and Bob have a private key, 1 and 3 respectively, they both agree to a shared key 5. Alice adds their private key (1) to the shared key (5) this creates Alice’s public key (6). Bob does the same (8). They then swap their public keys. Alice now adds her private key (1) to Bobs public key (8) to give Alice the secret key (9). Bob adds his private key (3) to Alice’s public key (6). This will then give them the same secret key (9). By doing it this way both Alice and Bob keep their private keys secure and private (Diffie & Hellman, 1976).

Asymmetric encryption can be used to authenticate data but cannot guarantee confidentiality or the integrity of the data.

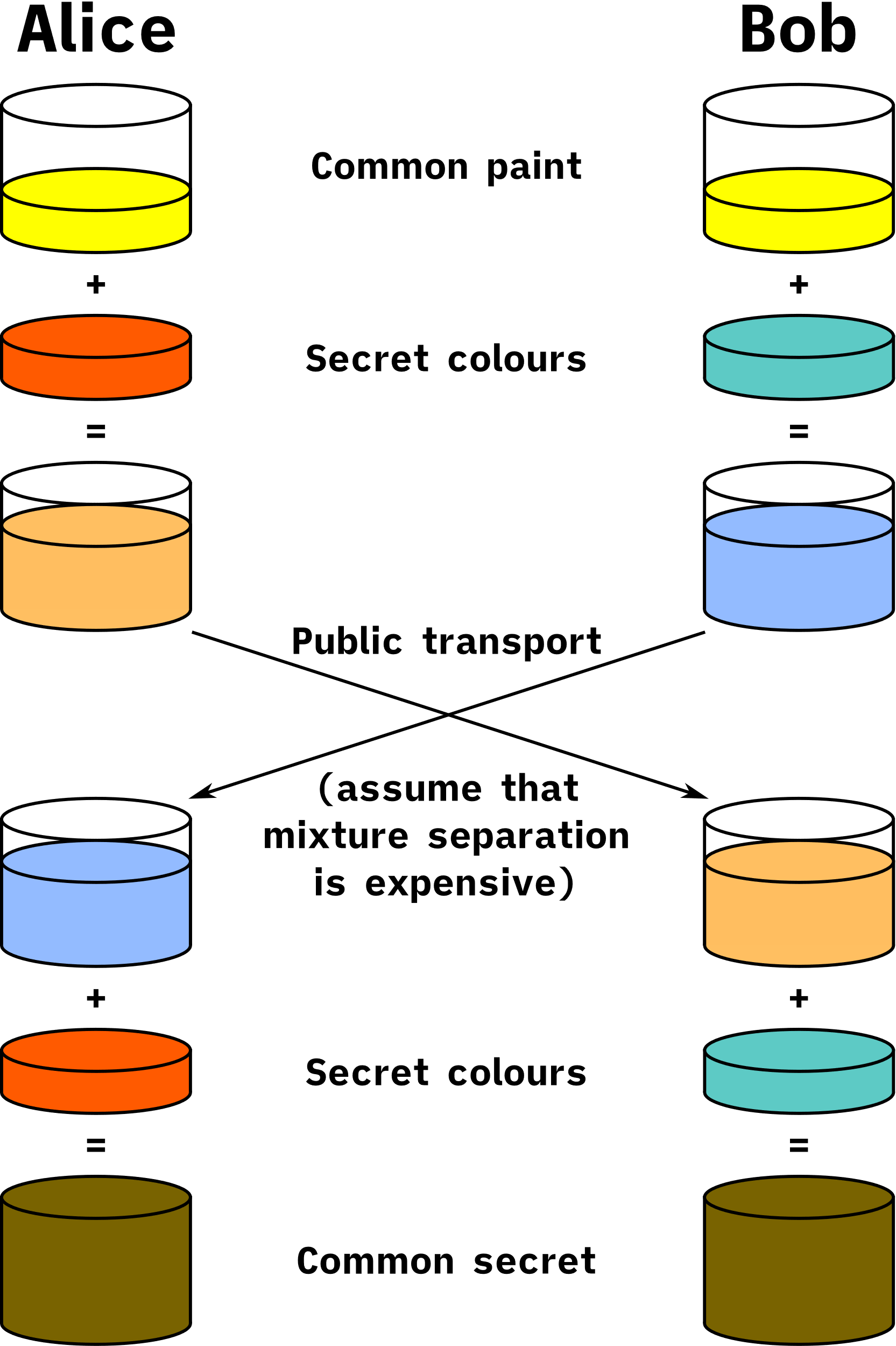


Figure 5: Diffie-Helman exchange explained with colours.

Note. *What is the Diffie–Hellman key exchange and how does it work?*, by J. Lake 2019 Comparitech (<https://www.comparitech.com>)

# Hashing and MAC

Unlike encryption hashing is designed not to be reversible. Hashing is a one-way function that creates a hash value of a fixed length from data of any size. Hashes are used as a method to verify the integrity of data. Whether you hash a textbook or the letter ‘a’ it will give you the same length hash. A hash with a private key is used to create a digital signature.

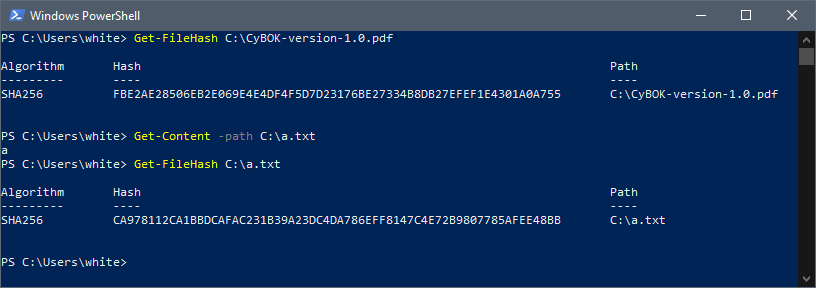


Figure 6: A SHA-256 hash of a textbook and the letter ‘a’

Despite the wide variance of data that can be hashed, collisions are highly improbable. Secure Hash Algorithm (SHA) is the predominant hashing algorithm used. SHA-256 creates a 256-bit hash value which is 64 hexadecimal characters in length (Stallings, 2014, p 330). This makes the chance of a collision with SHA-256, 4.3\*10-60 (Ramirez, 2015).

Message authentication code (MAC) is a method of verifying the authenticity of data using a shared symmetric-key (Stallings, 2014, p 357). Hash-based message authentication code (HMAC) takes a mac and hashes it this ensures both the authenticity and the integrity of the data (Krawczyk et al., 1997).

# X.509 Certificate

It is important that when a customer visits the Zambesi website, they can see that it is both secure and an official website. This can be achieved with certificates.

X.509 is an IETF standard for public-key certificates. A certificate is a means to confirm whether a public key belongs to the website. It contains the registration details of the website, its public key and the hash signature for the key (Cooper et al., 2008). Certificates are handed out by a certificate authority (CA). In the example below the certificate is issued by DigiCert Inc.



Figure 7: A partial X.509 certificate.

# TLS

This report has shown the importance of individual cryptographic methods in protecting data. What Transport Layer Security (TLS) does is bring all these methods into one protocol.

The main protocol for encrypting data between a website and its users is TLS. The first part of TLS is the handshake. The user sends a request to a website for their X.509 certificate. This is done with a ‘ClientHello’ message, this message contains the client’s version of TLS and the clients available cypher suites e.g. ChaCha20 and AES. The server will respond with a ‘ServerHello’, the x.509 certificate and its public key. The client will decrypt the server’s digital certificate with its public key and then verify the TLS certificate with the issuing CA. Using Diffie-Helman key exchange the client and server exchange keys to create a session key. Once done, the connection is now secured using symmetric encryption and the integrity of the data sent over the connection is guaranteed by HMAC (Dierks & Rescola, 2008).

A picture containing graphical user interface

Description automatically generated

Figure : TLS 1.2 Handshake

Note. *TLS 1.3 Handshake: Taking a Closer Look. By J. Thakkar 2018 The SSL Store (* <https://www.thesslstore.com>)

# UML

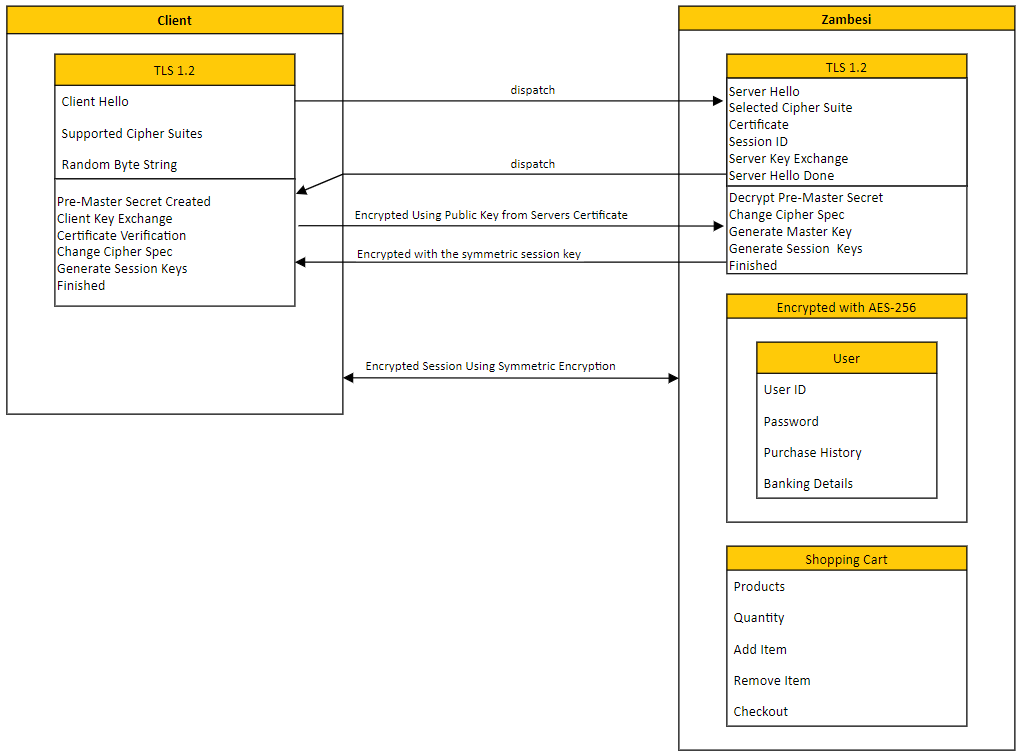


Figure : UML Diagram showing the implementation of encryption and TLS 1.2

# Conclusion

Losing a customer’s data can have a catastrophic effect on a business. Encrypting client data and using TLS will meet the security needs required to secure both the Zambesi website and Zambesi’s client data.

When holding a customer’s data, it is vital to follow the CIA triad. It is Zambesi’s responsibility to ensure the customers data is secure. Cryptography can enable Zambesi to follow the CIA triad. Symmetric encryption can be used to ensure confidentiality of customers data. Hash algorithms guarantee the integrity of the data and public-key encryption helps to authenticate the users accessing the data.

This report has shown without enacting encryption protocols such as TLS, Zambesi cannot achieve the basic security objectives required to secure customers data.

# References

Aumasson, J. (2018). *Serious Cryptography*. (1st ed.). San Francisco: No Starch Press, Inc.

Ben-Ari, M. (2003). *Mathematical Logic For Computer Science*. (2nd ed.). London: Springer

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008). RFC 5280: Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile. *IETF, May*.

Dierks, T., & Rescorla, E. (2008). RFC 5246-the transport layer security (TLS) protocol version 1.2. *The Internet Engineering Task Force (IETF)*.

Diffie, W. and Hellman, M. (1976) "New Directions In Cryptography". *IEEE Transactions On Information Theory* 22 (6), 644-654. <https://www.doi.org/10.1109/TIT.1976.1055638>

Galvin, J. (2018). *60 Percent of Small Businesses Fold Within 6 Months of a Cyber Attack. Here's How to Protect Yourself.* Inc.. <https://www.inc.com/joe-galvin/60-percent-of-small-businesses-fold-within-6-months-of-a-cyber-attack-heres-how-to-protect-yourself.html>

Krawczyk, H., Bellare, M., & Canetti, R. (1997). RFC2104: HMAC: Keyed-hashing for message authentication.

Lake, J. (2019). *What is the Diffie–Hellman key exchange and how does it work?*. Comparitech. <https://www.comparitech.com/blog/information-security/diffie-hellman-key-exchange/>

Menezes, A., Vanstone, S. and Van Oorschot, P. (1997). *Handbook Of Applied Cryptology (1st ed.)*. [Place of publication not identified]: CRC

Nohe, P. (2019). *How strong is 256-bit Encryption?*. The SSL Store. <https://www.thesslstore.com/blog/what-is-256-bit-encryption>

Ramirez, G. (2015). *MD5: The broken algorithm. Avira.* <https://www.avira.com/en/blog/md5-the-broken-algorithm>

Rescorla, E. (2018). Rfc 8446: The transport layer security (tls) protocol version 1.3. Internet Engineering Task Force (IETF).

Rijmen, V., & Daemen, J. (2001). *Advanced encryption standard*. Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology, 19-22.

Stallings, W. (2014). *Cryptography And Network Security.* (6th ed.). Boston: Pearson

Thakkar, J. (2018). *TLS 1.3 Handshake: Taking a Closer Look. The SSL Store.* https://www.thesslstore.com/blog/tls-1-3-handshake-tls-1-2/