# The State of HTTPS Configuration of Public Websites Around the World

Bugra Veysel Yildiz
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
b.v.yildiz@student.utwente.nl

## ABSTRACT

HTTPS is a crucial element in the world of online communication, transaction and data exchanges which have significant values in daily life. It ensures enhancing privacy and security of the internet users. To date, many researchers, engineers, organizations and browser companies worked in spreading the HTTPS to provide a secure connection between the web servers and clients. However, there is a variation in the adoption and configuration of HTTPS across different countries and regions of the world. This research seeks an outcome for determining the adoption rate of HTTPS in the world by investigating differences in public websites between developing and developed countries. Additionally, the vulnerabilities associated with the different HTTPS configurations are important to address. This research will use a combination of both quantitative and qualitative methods by involving scanning the HTTPS configurations of the domains and analyzing the case studies and literature review. The results have significant consequences on improvement of security and privacy in public websites and promoting the adoption of best practices. The output of this study indicates the difference of the HTTPS configurations in developing and developed countries. These can be enhanced according to the best practices for secure communication. Organizations may efficiently focus on these best practices and provide resources to solve any possible vulnerabilities.
.

### Keywords

HTTP, HTTPS, web-crawling, Secure Sockets Layer (SSL), Transport Layer Security (TLS), protocols, cipher strength

## 1. INTRODUCTION

Nowadays, people interact with each other, communicate and conduct business digitally via the internet. Due to the high connectivity, there is a substantial risk of data breaches, Man-in-the-middle attacks and other cyber threats. The highly effective way for securing online traffic HTTPS is the secure version of HTTP which runs on top of SSL/TLS [1]. It encrypts the data that is being exchanged between the web servers and clients, which makes it difficult for unauthorized parties to access the communication between other two parties which are client and server. In this project, we aim to investigate the adoption rate of HTTPS in public websites of developed and developing countries. Besides that, how different security threats can occur from different HTTPS configurations will be examined.

## 2. RESEARCH QUESTIONS

This project aims to analyze and evaluate the differences and similarities between the implementation of HTTPS in developed and developing countries. This helps us to provide valuable insights into the global landscape of online security. By identifying the HTTPS configurations of most popular websites of each country and specific security threats, policymakers who build regulations and guidelines can be enlightened to encourage to build more robust web security. Besides that, it can be insightful for the administrators who configure the secure and reliable web servers. In addition, understanding those vulnerabilities and weaknesses can help people that work in the industry about best practices and guiding principles of using HTTPS.

The following research questions can be retrieved:

1. How does the adoption and implementation of HTTPS in public websites differ between developing and developed countries?

2. What are the security risks associated with the HTTPS configurations of the public websites?

# 3. BACKGROUND

## 3.1 SSL/TLS Overview

SSL/TLS is the standard protocol for providing authenticity and confidentiality on top of TCP connections [2]. It can be built into the applications to protect the HTTP communication between client and server. This is where the "S" takes the role in the word "HTTPS".

The communication with SSL/TLS enabled applications starts with a handshake between client and the server. The handshake is required for the client to authenticate the server before the communication starts. The process of how SSL/TLS work is as follows; Client first sends a hello message to the server. This message includes the parameters as TLS/SSL versions, cipher suites, session ID, supported extensions and compression methods. Server responds to the client with a hello message specifying the parameters that are going to be used in the following communication and a digital certificate. This certificate states the server's public key, validity period, owner and the issuer information. Upon the client's successful verification of the certificate, the session keys which are required to encrypt and decrypt data exchanged during the SSL/TLS session as well as substantiate the message integrity are being introduced between the client and server [3].

This was the complete process of how the client and the server is being introduced to each other and how encrypted communication is supplied. Once the communication is complete or the session times out, the session is also being terminated with the session keys. This will make sure that a new session key is provided in every different session.

## 3.2. Investigating SSL/TLS Parameters: Understanding Key Factors in Secure Communication

This research aims to examine the key factors and parameters of SSL/TLS parameters. Specifically, TLS protocol version, cipher suites and the certificate validity lifespans are key features in those parameters that are being focused. This will give in depth analysis on the importance in ensuring robust security measures in online connections.

### 3.2.1. TLS Protocol Versions

TLS has been evolving and becoming more secure by solving or mitigating such security problems through continuous version upgrades. POODLE, BEAST and Lucky Thirteen can be given as examples to those security problems as we explained below [4].

TLS 1.0 was the first version of the TLS protocol which came out in 1999 as a minor modification to the SSL 3.0 protocol. However, TLS is not a perfectly secure protocol because it has exposed various vulnerabilities such as weak cryptographic algorithms and attacks that have been discovered over time. BEAST [4], POODLE [5] and Lucky Thirteen [6] are the examples for these attacks which compromise the security protocol. Weak cipher suites are also one of the aspects of why TLS 1.0 is insecure. It supports old and weak cipher suites compared to the later versions, there is a lack of support for more secure algorithms [7], [8].

These vulnerabilities identified in TLS 1.0 lead formation of TLS 1.1. In this version, there are new cipher suites identified. AES is an example of a new algorithm that leads to new cipher suites. Besides that, HMAC-based construction has been used instead of checksum-based construction that has been used in TLS 1.0. Backward compatibility is another point of development of TLS which enables the use of different TLS versions in the communications [9].

TLSv1.2 has been introduced as an upgrade to the previous version. There was limited cipher suite support, backward compatibility and performance in TLSv1.1. TLS 1.2 launched to address these problems and provide enhancements. It extended its cipher suite capabilities and brought more secure encryption and key exchange algorithms [1].

Lastly, TLS 1.3 has been introduced with significant improvements. Due to the slow key exchange that occurred in TLS 1.2, the new version came with a faster handshake protocol with reduced round trip times (RTT). Besides that, the weak cipher suites and legacy features have been removed to ensure higher level security. Perfect Forward Secrecy (PFS) feature which allows the confidentiality of past communications became mandatory to use in TLSv1.3.

The specifications have always been shaped for better and more secure protocols. According to Sullivan, TLS 1.3 is one of the best recent examples of how it is possible to take 20 years of deployed legacy code and change it on the fly, resulting in a better internet for everyone [10].

### 3.2.2. TLS Cipher Suites

A cipher suite is a set of cryptographic algorithms that has been used in the SSL/TLS protocol for secure network communication. It defines algorithms for the tasks such as key exchange, encryption and message authentication. These algorithms work together to provide robust and secure communication.

The combination of these components' algorithms generates the cipher suite and it is used in the handshake process of the communication between client and server. They negotiate and agree on the cipher suite that is mutually supported. The client sends a list of cipher suites it supports to the server. Then the server selects the most appropriate one from the list. This supported set of algorithms then used to preserve the confidentiality of the communication [11]–[13].

### 3.2.3. Certificate Validity Lifespan

An SSL certificate's validity lifespan is the time it shows the date that the SSL certificate is set to expire and the date that the SSL certificate is issued. It is one of the aspects of recalling whether a certificate is valid or invalid. A valid certificate can ensure the trustworthiness and authenticity of the website by revealing that it is issued and trusted by a Certificate Authority (CA) [14]. There are some reasons why certificate lifespan is not infinite. One of the essential reasons is because of the advancements in technology. These innovations can leave algorithms and cryptographic key sizes vulnerable to attacks. By setting expiration to the certificates and renewing them, certificates can make sure that they are updated with the certain keys and secure algorithms [15]. Additionally, confidentiality and integrity of the data transfer are the other factors that are protected by the validity of the certificate. It helps to reduce the security risks by warning users and browsers about the expired certificates, thereby preventing potential insecure connections.

## 4. RELATED WORK

In order to collect some related information in the research domain, Google Scholar, IEEE and ACM Digital Library is planning to be used for literature review for both HTTPS adoption rate and the security threats of different configurations.

To be able to get some insights on the adoption rate of the HTTPS, many studies have been conducted in various regions in both developed and developing countries. An existing study conducted by Google shows that in 2017, 64% of web pages loaded by Chrome users in the United States used HTTPS, while the adoption rate was significantly lower in other regions in developing countries such as Africa and South America [16].

The improper application and configuration of SSL during the implementation of Android apps can cause vulnerabilities such as Man-In-The-Middle (MITM) and phishing attacks. A tool called DCDroid has been designed to detect the vulnerabilities using static and dynamic analysis techniques. In the examination of 2213 apps from Google Play Store and 360app, it has been discovered that 457 application (20.65%) contains vulnerabilities to MITM and phishing attacks due to their SSL configurations. These findings take attention to the necessities in the implementation of SSL for securing HTTPS connections [17] .

## 5. METHODOLOGY

The research methodology that has been used in this study involves an approach with mixed-methods to analyze and examine the HTTPS configuration of public websites in developing and developed countreis. This approach has been described in a variety of ways which can make it a difficult concept to understand. However, collecting and analyzing both the qualitative and quantitative data in a pre-defined order covers more aspects of the inquiry. Some researchers believe that this approach provides researchers with opportunities to "… compensate for inherent method weaknesses, on inherent method strengths, and offset inevitable method biases". Creswell and Plano Clark comment that this approach enables a greater degree of understanding to be formulated than if a single approach were adopted to specific studies [18]–[22].

Firstly, a representative random sample of countries has been selected from both developed and developing countries. Then the top 100 domains from those countries have been collected. Cloudflare Radar is the resource that is going to be used in this research. Cloudflare offers a service where they identify the top most popular domains that reflect how people use the Internet globally and per country [23]. The developing and developed countries have been selected according to the ranking of their Human Development Index (HDI) value. Along with the research, 5 countries from each category have been selected randomly. Selected developed countries are Australia, Germany, Japan, Norway and Singapore. Morocco, Nigeria, Pakistan, Sudan and Syria have been selected for developing countries. Following this, top 100 popular domains for each country have been analyzed.

After collecting the pile of websites, website scanning and analysis has been made to those websites. OpenSSL has been employed as a crawling tool to assess the HTTPS configuration of the websites. It is an open source software that provides a robust, commercial-grade and full featured toolkit for general purpose cryptography and secure communication. The command "*openssl s_client -connect "$host":443 -servername*" initiates a SSL connection with the specified host for HTTPS port which is 443. SSL/TLS certificate data and the evaluation of the HTTPS configuration can be retrieved with the execution of this command. Based on this output, websites' supported TLS protocol versions, supported cryptographic algorithms (cipher suites) and validity of the certificates are the points that have been collected for analysis.

The collected data and the qualitative data has been used in the analysis to detect any significant difference or similarities between the configuration of the HTTPS in developing and developed countries. Based on the output, it has been examined to determine whether different configurations introduce any security vulnerabilities.

## 6. WEBSITE SCAN AND ANALYSIS

Once all the top 100 website domains from both developed and developing countries have been collected, TLS configurations for those domains have been retrieved with OpenSSL "s_client" tool. This command implements a generic SSL/TLS client which connects to a remote host using SSL/TLS [24]. This process has been done with a bash script. The script allowed me to get the configuration for all the domains in parallel.

After we combined the outputs, for each category, there were some duplicated domains. For instance, domains such as "google.com", "youtube.com", "facebook.com", etc. These are the most popular websites for almost all countries in the world. That's why those duplicated domains have been removed. There were also some domains that do not respond to the OpenSSL s_client call. The domains which I was not able to crawl the TLS configuration have also been removed. After the data cleaning, the number of domains that I could analyze was decreased from 1000 to 227. 110 domains for the developed countries and 117 domains for the developing countries.

When we analyzed the TLS Protocol Version rate, it was found that both developed and developing countries have similar rates. TLSv1.1 has not been encountered for any countries. TLSv1.3 is the version that is mostly being used. The rate was 66.1% for developed countries and 69.8% for developing countries. It can be seen from Figure 1 below.

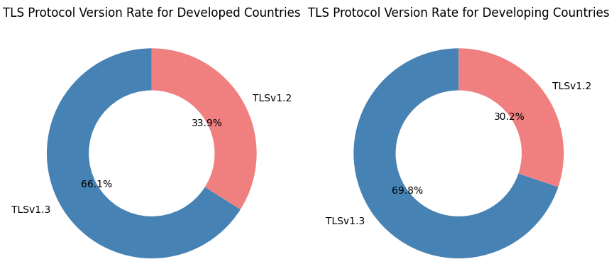


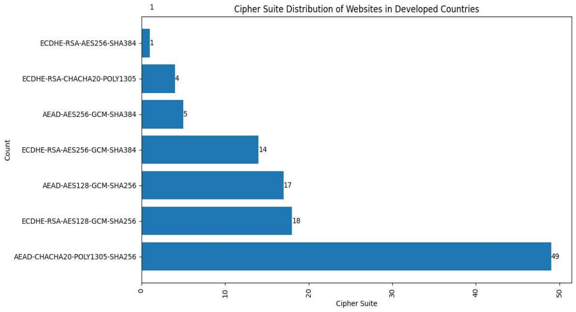
Figure 1: TLS Protocol Verison Rates

This means that there is a high rate of adoption of the latest security protocols in public websites around the world. It is one of the positive signs that indicates a focus on security, strong encryption and compliance with industry standards. It would not be correct to state the strengths and weaknesses of these results without analyzing each domain specifically with all its aspects. Usage of both TLSv1.2 and TLSv1.3 can be secure if they are both configured correctly.

Another point that has been analyzed in this research is the cipher suites. As pointed out previously, some of the cipher suites which have been supported in the non-latest TLS protocol versions are compromised. RC4, DSA, MD5, DH, ECDH are insecure ciphers that could be used in TLSv1.2. All the occurrences of cipher suites used with TLSv1.2 can be found in the Appendix A. Moreover, in the configuration of the HTTPS, TLSv1.2 support must be enabled since TLSv1.3 cipher suites are not compatible with older TLS protocol versions [25]. In the collected data, no instances of insecure cipher suites were found among developing and developed countries. The comparison of the cipher suites has been made with the secure cipher suites as it has been listed in Appendix B.

The number of occurrences of the cipher suites has been drawn in Figure 2. In that figure, it can be observed that the ECDHE key exchange is widely adopted. It indicates the global trend towards stronger security practices. There is also one occurrence of DHE key exchange observed in developing countries.

Servers use DHE key exchange to support forward secrecy. The majority of DHE-enabled servers used DH parameters that are weaker than their RSA signature strengths. This makes the sessions more vulnerable to a brute-force cryptanalysis attack than servers using the RSA key exchange which is without forward secrecy [26]. The reason for this weakness is the possible selection of insecure DH parameters. Apart from the length of the parameters, the security of DH also depends on the choice of the prime modulus of the generator [27]. On the other hand, ECDHE uses (algebraic) elliptic curves to generate the shared key. This shorter key length offers benefits over modular algorithms such as lower computational requirements while maintaining the security [28].



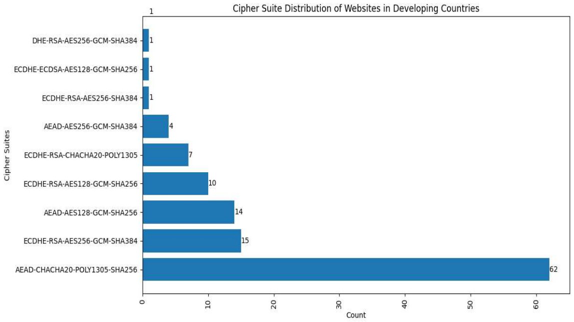


Figure 2: Cipher Suite Occurences

Validity lifespan of the certificates that are being used in the popular domains is the last point that has been analyzed in this website scan. According to the Federal PKI Policy Authority (FPKIPA), which is an authority for the US government, 1 year of a device certificate lifetime is recommended where permitted by operational considerations. Agencies should set a certificate lifetime of 1 to 3 years, depending upon the level of human interaction required to renew their device certificates [15]. This leads to filtering the domains whose lifespan is between 1 to 3 years. Among the examined countries, it is concluded that the developed countries tend to exhibit a slightly higher number of best practices in terms of certificate lifespan. According to the data, the number of best practices of certificate lifespan in developed countries is 60 while it is 56 in developed countries as shown in Figure 3. Besides that, we can affirm from the figure that in other examples of intervals which are not best practice, the number of usage in developing countries is higher than the developed countries.

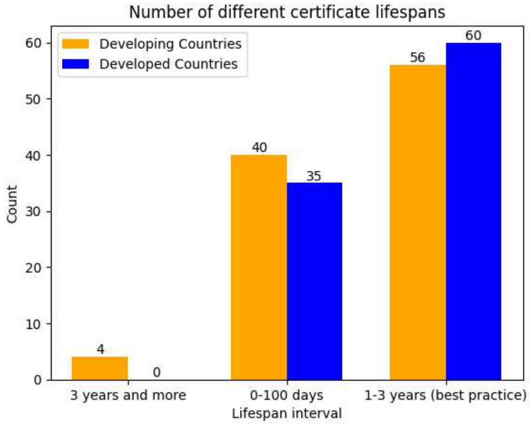


Figure 3: Number of occurrences of certificate lifespan

## 7. POSSIBLE VULNERABILITIES BASED ON HTTPS CONFIGURATION

In the overall analysis, it is observed that there is a high rate of adoption of the latest TLS protocol version. It shows that there is an emphasis on robust security measures.

There were no insecure cipher suites found in the examined domains. However, DHE cipher suites in developing countries can be a sign for possible vulnerability if it is configured with insecure DH parameters.

Developed countries have moderately higher adherence to best practices for the certificate lifespan. There was no configuration with an insecurely short lifespan such certificates that are valid for a few minutes, hours or days and no configurations with expired certificates. Those certificates are not considered as secure and should be renewed to support confidentiality and integrity. Despite that, there were 4 domains in developing countries which had an excessively long lifespan of more than 3 years while there were no domains found in developed countries. Certificates which have long lifespans increase the risk of potential compromise by enabling attackers to have extended time to exploit any potential vulnerability.

## 8. FUTURE WORK

Based on the findings of this study, numerous directions for further investigation can be identified. Despite this research demonstrating an enhanced sense of coherence, the relationship between the key sizes and validity periods for the certificates has not been integrated into the study. The higher the key size, the more secure the certificate is from attackers, but will require more processing to use. According to the RSA research, The matrix that has been presented is:

- Key length of 1024: Validity period = not greater than 6-12 months
- Key length of 2048: Validity period = not greater than 2 years
- Key length of 4096: Validity period = not greater than 16 years

In further analysis, these requirements must be considered to improve the study and provide an outcome about whether there is any vulnerability and any difference in the domains for developing and developed countries regarding the key lengths. It was not possible to retrieve the key size of the certificates efficiently with the OpenSSL s_client tool. That's why the use of another HTTPS configuration crawling tool is recommended for future work.

Another point for future work can be examining the DH parameters in DHE-enabled servers. The ephemeral Diffie-Hellman (DHE) key exchange is being used by many TLS servers to support forward secrecy. However, there is a research that has been conducted that mentions 82.9 percent of the DHE-enabled servers use weak DH value and this results in a false sense of security [28]. Therefore, further investigation into the selection and utilization of the DH parameters can be made to enlighten the vulnerabilities and potential improvements for robust TLS configurations.

Due to the restricted time limit in this research, the sample size that has been analyzed in this research is low. Due to some unresponded calls and data duplications, we had to get rid of 77.3% of the data. This could be improved by analyzing more countries. This would increase the accuracy of the results.

## 9. CONCLUSION

To bring it all together, this study has been focused on the adoption of HTTPS in public websites of both developing and developed countries. By the use of a mixed-methods approach, which combines quantitative analysis from the TLS configuration scanning and qualitative analysis of case studies and literature reviews, the similarities and differences of the HTTPS configurations and potential vulnerabilities of them has been enlightened.

The results indicate that the HTTPS is widely used in both developed and developing countries. The configurations express the significant adoption of the latest TLS protocol version which is TLSv1.3. Besides that, stronger encryption and forward secrecy are being prioritized globally. The analysis of the cipher suites shows the broad use of the secure key exchange algorithm, ECDHE. There were no instances of insecure cipher suites found.

Furthermore, along the study, the validity lifespan of the certificates in popular dominas were examined. It has been found that the best practice lifespan interval is higher in developed countries. This draws attention to how crucial it is to have up-to-date certificates in the TLS configuration for trustworthiness, authenticity and secure data transfer.

The higher number of best practices in developed countries can be due to numerous reasons, including improved infrastructure, easy access to resources and higher level of institutional/governmental support for cybersecurity measures. On the other hand, developing countries may experience some challenges such as lack of resources, education and awareness about the best practices on HTTPS configuration procedures.

The outcomes of this research have a great impact on the improvement of security and privacy of the public domains. The result can promote the development of regulations and standards about robust web security measures in the world. The experts in the industry can adopt and encourage the best practices by understanding the weaknesses and vulnerabilities in the HTTPS configurations.

## 10. ACKNOWLEDGEMENT

We would like to express our appreciation to our supervisor Florian Hahn, who has been incredibly supportive and encouraged me throughout, and provided valuable and comprehensive feedback.

## 11. REFERENCES

[1] "https://www.ietf.org/rfc/rfc5246.txt#:~:text=The protocol allows client%2Fserver,%2C tampering%2C or message forgery.&text=The primary goal of the,integrity between two communicating applications." Accessed: May 23, 2023. [Online]. Available: https://www.ietf.org/rfc/rfc5246.txt

[2] D. Naylor *et al.*, "The Cost of the 'S' in HTTPS," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, Sydney Australia: ACM, Dec. 2014, pp. 133–140. doi: 10.1145/2674005.2674991.

[3] "SSL/TLS Handshake: Detailed Process and How does it Work," *SSL2BUY*. https://www.ssl2buy.com/wiki/ssl-tls-handshake-how-does-it-work (accessed May 23, 2023).

[4] "beast.pdf." Accessed: Jun. 20, 2023. [Online]. Available: https://tlseminar.github.io/docs/beast.pdf

[5] B. Möller, T. Duong, and K. Kotowicz, "This POODLE Bites: Exploiting The SSL 3.0 Fallback".

[6] N. J. Al Fardan and K. G. Paterson, "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols," in *2013 IEEE Symposium on Security and Privacy*, May 2013, pp. 526–540. doi: 10.1109/SP.2013.42.

[7] "ietf.org/rfc/rfc2246.txt." Accessed: Jun. 20, 2023. [Online]. Available: https://www.ietf.org/rfc/rfc2246.txt

[8] J. Lee, H. Lee, J. Jeong, D. Kim, and T. T. Kwon, "Analyzing Spatial Differences in the TLS Security of Delegated Web Services," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, Virtual Event Hong Kong: ACM, May 2021, pp. 475–487. doi: 10.1145/3433210.3453107.

[9] "https://www.ietf.org/rfc/rfc4346.txt." Accessed: May 24, 2023. [Online]. Available: https://www.ietf.org/rfc/rfc4346.txt

[10] "A Detailed Look at RFC 8446 (a.k.a. TLS 1.3)," *The Cloudflare Blog*, Aug. 10, 2018. http://blog.cloudflare.com/rfc-8446-aka-tls-1-3/ (accessed May 26, 2023).

[11] alvinashcraft, "Cipher Suites in TLS/SSL (Schannel SSP) - Win32 apps," Nov. 14, 2022. https://learn.microsoft.com/en-us/windows/win32/secauthn/cipher-suites-in-schannel (accessed May 29, 2023).

[12] P. Morrissey, N. P. Smart, and B. Warinschi, "The TLS Handshake Protocol: A Modular Analysis," *J. Cryptol.*, vol. 23, no. 2, pp. 187–223, Apr. 2010, doi: 10.1007/s00145-009-9052-3.

[13] J. A. Salowey, D. McGrew, and A. Choudhury, "AES Galois Counter Mode (GCM) Cipher Suites for TLS," Internet Engineering Task Force, Request for Comments RFC 5288, Aug. 2008. doi: 10.17487/RFC5288.

[14] T. Chung *et al.*, "Measuring and Applying Invalid SSL Certificates: The Silent Majority," in *Proceedings of the 2016 Internet Measurement Conference*, Santa Monica California USA: ACM, Nov. 2016, pp. 527–541. doi: 10.1145/2987443.2987454.

[15] "Wayback Machine," Feb. 15, 2013. https://web.archive.org/web/20130215040433/http://www.idmanagement.gov/fpkipa/documents/AgencyBestPracticesDeviceCerts.pdf (accessed Jun. 18, 2023).

[16] "HTTPS encryption on the web – Google Transparency Report." https://transparencyreport.google.com/https/overview?hl=en (accessed May 04, 2023).

[17] Y. Wang *et al.*, "Identifying vulnerabilities of SSL/TLS certificate verification in Android apps with static and dynamic analysis," *J. Syst. Softw.*, vol. 167, p. 110609, Sep. 2020, doi: 10.1016/j.jss.2020.110609.

[18] K. Niglas, "How the novice researcher can make sense of mixed methods designs," *Int. J. Mult. Res. Approaches*, vol. 3, no. 1, pp. 34–46, Apr. 2009, doi: 10.5172/mra.455.3.1.34.

[19] "Mixed Methods Research: A Research Paradigm Whose Time Has Come." https://journals.sagepub.com/doi/epdf/10.3102/0

013189X033007014 (accessed May 23, 2023).

[20] "Media Review: Greene, J. C. (2007). Mixed Methods in Social Inquiry. San Francisco: Jossey-Bass." https://journals.sagepub.com/doi/epdf/10.1177/1558689807314013 (accessed May 23, 2023).

[21] J. W. Creswell and V. L. P. Clark, *Designing and Conducting Mixed Methods Research*. SAGE, 2011.

[22] S. Almalki, "Integrating Quantitative and Qualitative Data in Mixed Methods Research—Challenges and Benefits," *J. Educ. Learn.*, vol. 5, no. 3, p. 288, Jul. 2016, doi: 10.5539/jel.v5n3p288.

[23] "Cloudflare Radar," Sep. 30, 2022. https://radar.cloudflare.com/ (accessed May 26, 2023).

[24] "/docs/man1.0.2/man1/openssl-s_client.html." https://www.openssl.org/docs/man1.0.2/man1/openssl-s_client.html (accessed Jun. 16, 2023).

[25] "Types of Ciphers and How to Create A Cipher Order?" https://crashtest-security.com/configure-ssl-cipher-order/ (accessed Jun. 18, 2023).

[26] L.-S. Huang, S. Adhikarla, D. Boneh, and C. Jackson, "An Experimental Study of TLS Forward Secrecy Deployments," *IEEE Internet Comput.*, vol. 18, no. 6, pp. 43–51, Nov. 2014, doi: 10.1109/MIC.2014.86.

[27] A. P. Kate, P. S. Kalekar, and D. Agrawal, "Weak Keys in Diffie-Hellman Protocol".

[28] H. Kario, "RSA and ECDSA performance," *securitypitfalls*, Oct. 05, 2014. https://securitypitfalls.wordpress.com/2014/10/06/rsa-and-ecdsa-performance/ (accessed Jun. 18, 2023).

## APPENDIX A

| **The number of different cipher suites and occurrences used with TLSv1.2** | | | |
|---|---|---|---|
| **Developed Countries** | | **Developing Countries** | |
| **Cipher Suite** | **Occurrence** | **Cipher Suite** | **Occurrence** |
| ECDHE-RSA-AES128-GCM-SHA256 | **18** | **ECDHE-RSA-CHACHA20-POLY1305** | **7** |
| **ECDHE-RSA-AES256-GCM-SHA384** | **14** | **ECDHE-ECDSA-AES128-GCM-SHA256** | **1** |
| **ECDHE-RSA-CHACHA20-POLY1305** | **4** | **DHE-RSA-AES256-GCM-SHA384** | **1** |
| **ECDHE-RSA-AES256-SHA384** | **1** | **ECDHE-RSA-AES256-SHA384** | **1** |
| | | **ECDHE-RSA-AES256-GCM-SHA384** | **15** |
| | | **ECDHE-RSA-AES128-GCM-SHA256** | **10** |

## APPENDIX B

**SECURE CIPHERS FOR TLSv1.2**

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305

### SECURE CIPHERS FOR TLSv1.3

- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256
- TLS_AES_128_GCM_SHA256
- TLS_AES_128_CCM_8_SHA256
- TLS_AES_128_CCM_SHA256