Transport Layer Security

Conner Brinkley

10.19.2020

Applied Cryptography
The University of Tennessee

Reviewed by: Regan Moreno

INTRODUCTION

The purpose of this project is to closely examine how different websites establish secure connections through various cipher suites and protocols. To accomplish this, data was collected from ten personally frequented websites using the **openssl s_client** command line interface (CLI) and arranged in a table for comparison. In general, there were more similarities than differences between all of the cipher suites, but there were still a few interesting finds, which left a couple lingering questions.

RESULTS

Website	Key Exch.	Auth	Encryption	Key Size	Mode	Message Auth	Protocol
twitter.com	ECDHE	RSA	AES	128	GCM	SHA-256	TLS v1.2
mail.google.com	ECDHE	RSA	ChaCha20	256	–	Poly1305	TLS v1.3
github.com	ECDHE	RSA	AES	128	GCM	SHA-256	TLS v1.3
myutk.utk.edu	ECDHE	RSA	AES	256	CBC	SHA-384	TLS v1.2
utk.instructure.com	ECDHE	RSA	AES	128	GCM	SHA-256	TLS v1.2
onlinebanking.regions.com	ECDHE	RSA	AES	256	GCM	SHA-384	TLS v1.2
maps.google.com	ECDHE	RSA	ChaCha20	256	–	Poly1305	TLS v1.3
linkedin.com	ECDHE	RSA	AES	128	GCM	SHA-256	TLS v1.2
netflix.com	ECDHE	RSA	AES	128	GCM	SHA-256	TLS v1.3
connerbrinkley.com	ECDHE	RSA	AES	128	GCM	SHA-256	TLS v1.2

Table 1. Cipher suites and other cryptographic information for 10 various websites.

The table above shows cryptographic information for ten of my most frequently visited websites, which has been interesting to collect and analyze since these websites contain a lot of my personal and financial information. The information included shows the cipher suites used for each website and the protocol for establishing a secure connection.

The first notable thing to mention is that all ten of the websites use the ephemeral elliptic curve Diffie-Hellman key exchange algorithm (ECDHE) to share a symmetric key for their encryption methods, and they are all signed with RSA for authentication.

Moving on, eight of the ten websites use block cipher encryption, and out of those eight, six of them use 128-bit AES in Galois/Counter Mode (GCM). The other two, Regions Bank and MyUTK, both use 256-bit AES encryption, but MyUTK is the only website that uses AES encryption in cipher block chaining mode (CBC). Although 128-bit AES encryption is sufficient enough to protect against brute force attacks, I suspect that Regions Bank and MyUTK likely use 256-bit AES because they both manage a lot of money and the larger number makes their customers feel safer.

After attempting to connect to the two Google subdomains (Gmail and Google Maps) using the **openssl s_client** CLI, I came across the ChaCha20 encryption algorithm. It seemed a bit weird not having a corresponding encryption mode, so after some light Googling, I discovered that the ChaCha20 algorithm is a stream cipher that uses a 256-bit key. According to Cloudflare, ChaCha20 has much better mobile browser performance over AES, which would make it a valuable alternative cipher suite under constrained conditions.

For message authentication, aside from the two Google subdomains that are using ChaCha20, all of the other websites are using a safe SHA variant. The ChaCha20 encryption methods both use the Poly1305 message authentication code algorithm, which is also a good alternative on mobile because of its superior performance. Thankfully, none of the websites are using SHA-1 for their MAC algorithm.

I had some issues getting the latest version of the **openssl** CLI to run on my MacBook, which was required to connect with TLS 1.3, so I instead used the developer tools in the Chrome and Firefox browsers to check what protocols were being used by default. Netflix, GitHub, Gmail, and Google Maps all established a TLS 1.3 connection, while the rest all used TLS 1.2. Interestingly, when using Chrome and connecting to the two Google subdomains, QUIC was the default protocol instead of TLS. This is likely a result of staying in the Google ecosystem by using Google's browser and connecting to Google-owned websites.

CRYPTOGRAPHIC GUARANTEES

The ephemeral Diffie-Hellman (ECDHE) variant ensures perfect forward secrecy, which means that all of the sites can be trusted to not leak any key exchange information in the future. In order to make sure that the key exchange is coming from a trustworthy source, RSA is paired with it for signing and authentication. Since all of the websites are using at least 128-bit AES once a secure connection is established, the encryption methods guarantee message confidentiality. Finally, all of the websites are using a safe hashing algorithm, if necessary, which means that

they are safe from collision or pre-image attacks and that the message source can most likely be trusted. Some of the cipher suites are probably better than others under certain conditions, but overall none of the websites jump out as totally insecure or untrustworthy; although, all of the ones using TLS 1.2 should upgrade to 1.3 for that extra bit of security.

LINGERING QUESTIONS

After also using the SSL Server Test tool from SSL Labs, I found that many websites have much more than just one or two cipher suites. I suspect this is due to optimizing under many different environments, such as mobile vs. desktop, Chrome vs. Firefox, or TLS v1.2 vs. v1.3, but I am not completely clear on that and what all the tradeoffs are.

Because all of the sites using AES are using GCM mode except one, I would be interested to learn more about this mode and how it compares to CBC.

I would also be interested to learn more about how the Poly1305 MAC algorithm works and how it differs from the HMAC algorithm we learned about in lecture.