AN IXIA RESEARCH REPORT

The State of the Internet

Encryption—What's Hiding in Plain Sight.

ixia

A Keysight Business

CONTENTS

TABLE OF CONTENTS

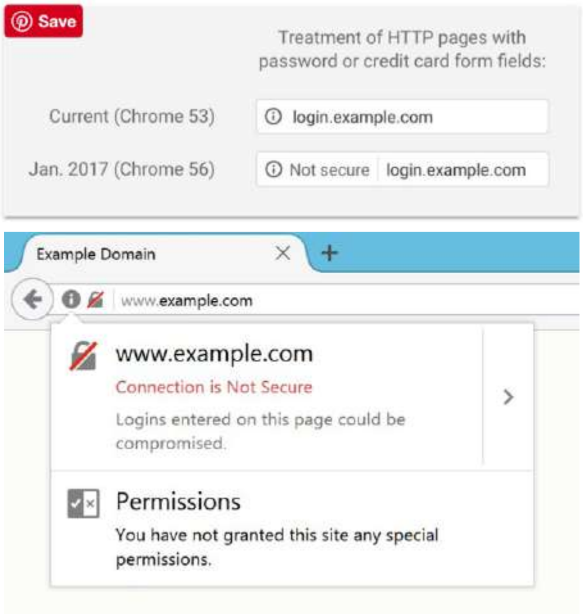
ENCRYPTION, ENCRYPTION EVERYWHERE

The Internet has adapted to encryption at a breakneck speed. Multiple studies indicate a rapid increase of encrypted Internet traffic. According to Mary Meeker’s 2019 Internet Trends report, 87 percent of Web traffic was encrypted at the start of 2019, compared to just 53 percent in 2016.

The technology industry is pushing for encryption over the last few years. Google Chrome and Mozilla Firefox have been calling out any website without encryption as “Not Secure.”

“Rise of encrypted browsing shows no signs of slowing” **- Forbes**

“With corporate demand for data security growing, full data encryption is becoming a default feature in the mobile space” **– Network Computing World**



Treatment of HTTP pages with password or credit card form fields:

Current (Chrome 53)	ⓘ login.example.com
Jan. 2017 (Chrome 56)	ⓘ Not secure login.example.com

www.example.com

Connection is Not Secure

Logins entered on this page could be compromised.

Permissions

You have not granted this site any special permissions.

Figure 1

Chrome and Mozilla are calling out any unsecured browsers.

Additionally, services like Lets Encrypt, an open-certificate-authority maintained by the non-profit Internet Security Research Group (ISRG), are propagating encryption initiatives. By being a certificate authority and providing certificates to interested websites, it reduces the overall cost of encryption for a website considerably. All these together ensure that, sooner than later, almost all relevant traffic will be encrypted and the ones that will not be encrypted will mostly be irrelevant (in a grand scale of the Internet) websites that don’t have the need to encrypt.

But in the frantic drive to encrypt, equipment makers and enterprises are finding a lack of the data needed to properly understand and thereby handle the encryption flood. This starts with vendors who need to design the right products that can effectively handle encryption across cloud-scale networks. It also includes enterprises and service providers who both need to upgrade their hardware and software infrastructures and tune their network and server/service policies to better handle the encrypted traffic.

TLSV1.3 -THE NEW SECURITY KID IN THE BLOCK

The IETF released a new version of its encryption standard called RFC 8446 (transport layer security (TLS) protocol version 1.3) in the latter half of 2018. This is a newer version of the original IETF secure socket layer (SSL) protocol first issued in 1995. While the name was changed a few years, the goal is still the same—encrypt IP traffic. TLS 1.3 brings in some of the most revolutionary concepts in TLS like reducing one round trip time for encryption, a zero-round trip (0-RTT) session resumption, and improved security with the enforcement of the use of ephemeral keys and the support of a much smaller list of ciphers and keys. All these steps were taken to ensure that while TLS 1.3 gives higher performance, it doesn't get any less secure than its predecessor and while doing all this should also remain easy to configure.

Public key exchange has a major impact on encryption performance and inspection strategies

HTTP2.0 IS CHANGING THE INTERNET ONE WEBSITE AT A TIME

HTTP2.0 solved several issues of HTTP1 apart from adding some of its own enhancements. Although derived from the much lesser popular predecessor SPDY, HTTP2.0 is gaining rapid prominence due to several advantages like multiplexing of some requests over a single TCP/Encrypted session, HTTP2.0 server push, and data compression of HTTP headers. The fact that it is a binary protocol as compared to HTTP1.X also helps Improve latency. The improvements that come with HTTP2, along with major browser support for it, have caused its rapid adoption in servers.

THE CASE OF THE MISSING INFORMATION

There are many papers, studies, and research that suggest the meteoric rise of encryption. However, there is little information about the types of encryption, popular ciphers and their distributions, key sizes negotiated, or changes in the ciphers over last few years. This is the basic data needed to make informed decisions when manufacturing, purchasing, installing, or monitoring SSL-aware infrastructure. To fill the information gap, we will dig deeper into these topics, research the state of encryption, and share our findings. ***In fact, our analysis of the data that follows indicates the urgent need for equipment vendors to change the way they build and optimize hardware and software and for enterprises and service providers to re-design encrypted data inspection and monitoring strategies and set new requirements for future procurement of network infrastructure.***

Let's first discuss the type of data we've set out to discover. This is not a report on encryption types, so we will spare those details, but cover the points that are important for a better understanding of the report data.

All generic web encryptions have two phases:

1. A public key exchange that results in both the client and server deriving a shared secret
2. The use of the shared secret for bulk encryption and decryption of application data

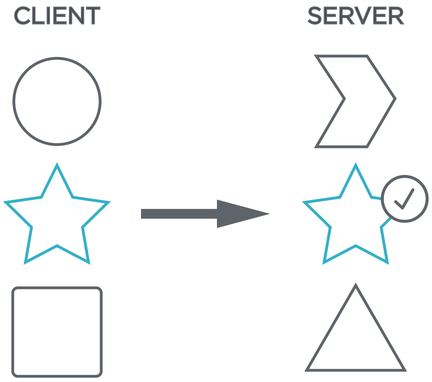
Public key exchange has a major impact on encryption performance numbers and inspection strategies, which is why this report will focus mainly on public key exchange ciphers. Almost all public key exchange is performed through one of the two ciphers, RSA (named for the algorithm makers, Rivest, Shamir, Adleman) or DHE/ECDH (Diffie Hellman Exchange/Elliptic Curve Diffie Hellman).

A major difference between these two is how often the server generates its private keys. RSA generally has a common private key that it couples with some random numbers and uses a specific function to generate the public key for each connection. For DHE or its more efficient Elliptical Curve (EC) successor ECDHE, the server has a unique private key per connection. This fundamental difference between the two major public key exchange protocols completely changes encryption behavior, performance numbers, and inspection strategies depending on which cipher is selected.

For example, if the public key exchange is always RSA, one can easily deploy a passive SSL inspection technique where the long-term private key is fed into the inspection device, and thus the inspection device gets the power to decrypt and encrypt any traffic without terminating and re-initiating a secured connection (passive inspection). However, the same inspection strategy can't be employed for DHE/ECDHE as the private key changes for each session. The only alternative for the inspection device is to terminate and re-initiate each connection (active inspection).

Figure 2

The client sends all its available ciphers and the server should select the common cipher that is strongest amongst them.



The research involved more than 130 client browsers and their different variants. To make it comprehensive and truly pan-international, the analysis included browser versions that came out in between 2012 to 2019, on different platforms/flavors, etc.

The significant difference in the public key exchange between the two ciphers makes it extremely important to understand the results of the popularity contest between them. This information will help both vendors and end users optimize and publish relevant performance numbers of the popular ciphers, deploy the right inspection methods, and right-size their investments in encryption infrastructures. As a corollary, this research also throws light on browsers and servers that may be propagating stronger or weaker security postures.

COLLECTING PUBLIC KEY EXCHANGE INFORMATION

The straight-forward way is to monitor all traffic passing through the Internet and monitor all encryption exchange to determine the cipher used in the exchanges. Of course, this straight-forward way is also understandably impossible, which is why we looked at other possible methods to reach closer to the truth. It turns out that there is another way that doesn't require us to monitor any network. The solution lies in the rules of public key exchange. In the first step in a key exchange, the client sends a "hello" that provides all its cipher proposals in order of preference. The server generally SHOULD (it's not a MUST) pick the strongest proposal that they both have in common.

This behavior ensures that if we analyze the cipher preferences of major clients (browsers) and popular servers, we will have a pretty accurate idea of the state of encryption. Based on this assumption, we conducted an experiment where we decided to explore the state of TLS by surveying a subset of the Cisco Umbrella 1 Million (a free list of the top 1 million most-popular domains).

This research was done from the perspective of the client, to see which parameters end up being negotiated upon successful connection to actual websites.

The experiment was conducted on the top 1 million host names. Using the latest OpenSSL binary packaged with AWS lambda and leveraging the Kubernetes infrastructure seemed like a good way to run multiple short queries to get the most up-to-date information in a fast and cost-effective way.

While we were at it, it also made sense to do some additional queries to understand other interesting facts like the adoption rate of HTTP2.0 and the proliferation of QUIC as a transport.

SERVER CIPHER PREFERENCE

In a perfect world, a server will always select the strongest amongst the list of cipher proposals that the client has sent, provided they support that cipher. That means our analysis of preference of ciphers, their rankings, and other inferences that we did earlier will hold true every time a browser interacts with the server. However, this is not a perfect world and not every server will do the "right" thing. There are many motivations for the servers to deviate and select a lower-preference cipher suite from the client list, as there is no fixed rule that is stopping them.

No matter what the browser's preference are, it is the servers that will determine if the world is moving towards a more encrypted Internet.

Servers can have many motivations to prefer one cipher over other. For example, an institute that has heavily invested in passive SSL inspection infrastructure would like their servers to negotiate TLS_RSA always to ensure it can still employ passive inspection. Similarly, a server that is running low on resources may force the use of weaker cipher suites or smaller key sizes to save computational bandwidth. Sometimes the upgrade cost of encryption infrastructure may prove to be a hindrance for a server to use better ciphers.

This is where the second part of research comes in, where we analyze the top 1 million websites, find out their cipher preferences, and the key sizes that finally get negotiated. This will also expose general website strategies on selecting the ciphers.

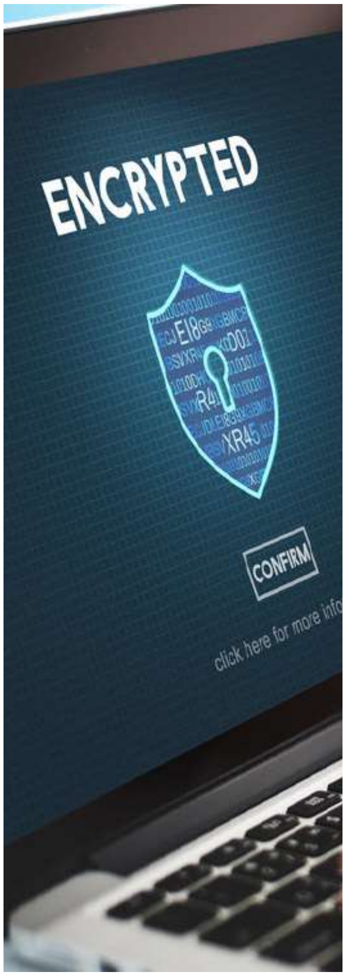
Similarly, the server will have motivation to select HTTP versions based on their infrastructure, investment, and business priorities. HTTP2.0 provides additional performance improvement, reduced latencies, and better security, which should be enough impetus to switch to HTTP2.0. However, it is interesting to understand if the websites of the world have taken the step to move to HTTP2.0.

COLLECTING SERVER CIPHER INFORMATION

In the world of Internet, millions of websites operate and many are added each day. This extreme dynamism and breadth means that however accurate and detailed the analysis we do, it will still have some error margins and the results will change after a short time. Our purpose is to explore macro-level trends and gain an understanding of the immediate big picture. Analysis of the top-ranked websites will provide us a general idea of key trends and since the majority of the traffic is generated by the top few-thousand websites of the world, we can also assume that a trend that holds true for the first few hundred thousand should continue for the next several million.

As part of this exercise to periodically assess the state of the Internet in terms of supported protocols and cryptography suites, we ran a list of DNS, HTTP, and HTTPS queries against the Cisco Top 1 Million domains list by leveraging the Kubernetes, AWS Lambda, and the AWS EKS cluster infrastructure. We specifically looked at the responses from using HTTP curl to query 1m domains/hosts from the Cisco list. The results tell us:

- TLS version negotiated
- Cipher suites used
- Application-layer protocol negotiation (ALPN) support
- HTTP protocol version



As part of this exercise to periodically assess the state of the Internet in terms of supported protocols and cryptography suites, we ran a list of DNS, HTTP, and HTTPS queries against the Cisco Top 1 Million domains.

In this research, we decided to not call out any websites no matter how high their rank. It is also important to note that for the most part, it is neither wrong nor irresponsible to use TLS_RSA. In fact, NIST considers TLS_RSA with 2K key size to be secured till 2030.

TLSv1.3 has made massive strides and is catching up with TLS 1.2 one server at a time.

SERVER SIDE ANALYSIS

The first time we analyzed the data and were ready to view the result, this was definitely an anxious moment for us. All our previous research related to browsers pointed to the extensive popularity and strong preference for TLS_ECDHE for public key exchange. Now, if the websites didn't follow suit, then that would mean the Internet is not moving into the direction of modern, efficient, and at the same time highly secured servers. Thankfully that wasn't the case.

WHAT ARE THE MOST IMPORTANT TLS VERSION

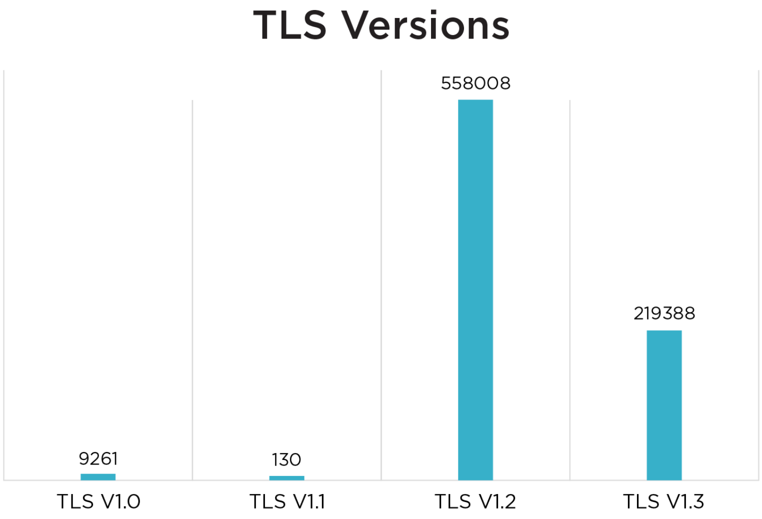


Figure 3: TLS v1.3 and 1.2 are the dominant TLS versions.

Our research found that not only the top websites are ensuring they have the more secured TLS versions (1.2 and higher), they are also rapidly moving towards TLSv1.3. This is indeed great news as far as Internet security is concerned. It is also an eye-opener for some vendors as they will need to support TLSv1.3 faster than previously anticipated. As the data shows, nearly 30% of the top 1M servers show preference towards TLSv1.3. Another heartening fact about this data is that the servers have decidedly rejected the older SSL versions like SSLv3 and SSLv2.

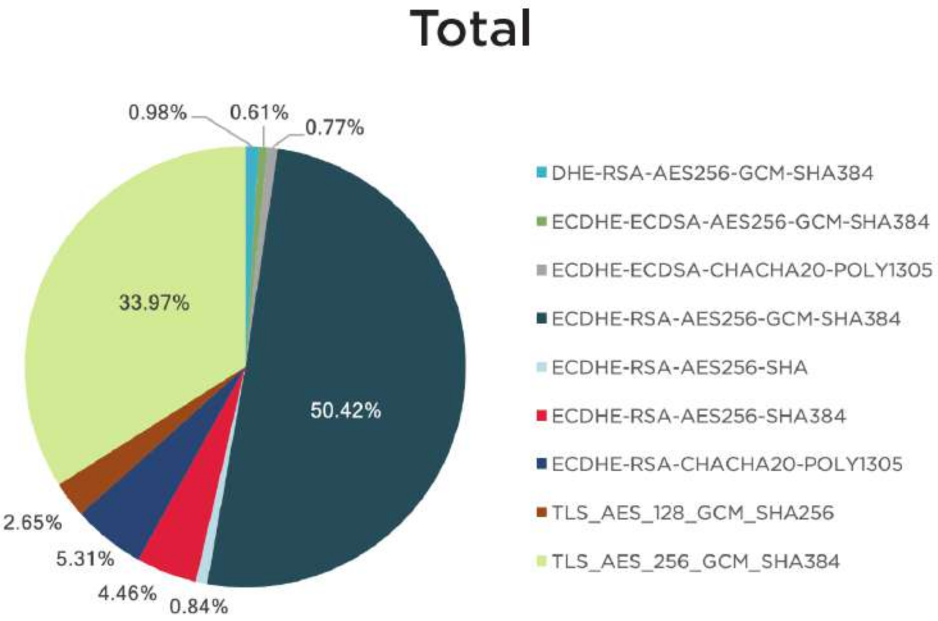


Figure 4: Cipher preference shows focus on just two in the vast range of options.

This is where things get interesting. If we make the connection between the previous chart where we compare the TLS versions, it's quite clear that when the servers select TLSv1.2, they will go for the ECDHE-RSA-AES256-GCM-SHA384 ciphers, and when they select the TLSv1.3, the preferred cipher is TLS_AES_256_GCM_SHA384. This basically means, overwhelmingly, the servers have focused in on just a few ciphers in the vast range of options available to them. This is also a good model to follow for the architects designing their encryption strategies and for vendors for publishing their performance numbers.

From the sea of encryption choices, only TWO ciphers are being preferred by more than half a million of world's top websites.

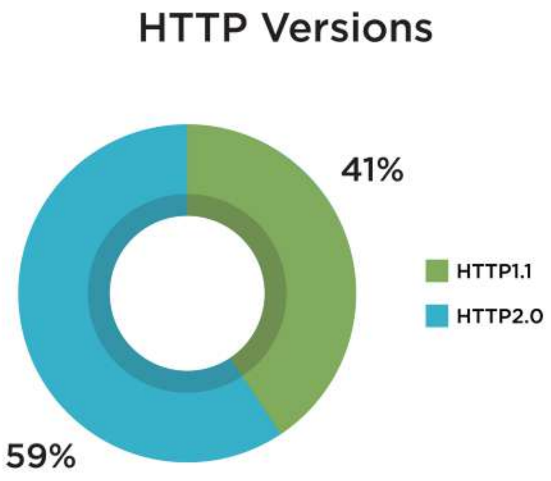


Figure 5: HTTP2.0 unexpectedly passed HTTP1.1 for server preference.

This is by far the most interesting chart for all of us. We were understanding that HTTP2.0 has been gaining prominence, however we didn't expect it to pass HTTP1.1 for servers preference.

It looks like this trend has been continuing for some time, where more websites are moving to HTTP2.0. This makes sense as websites have tremendous impetus to move to HTTP2.0 as it improves latencies along with providing security.

This also is a great data point as both network and security infrastructure teams need to configure devices to handle traffic sent over the HTTP2.0. Similarly, network and application tool vendors need to tune their devices for better HTTP2.0 performance. Overall, this also shows that the top websites are quite aligned with the latest Internet trends and are willing to move to it faster than ever before.

The meteoric rise of HTTP2 indicates the top websites are moving quickly to improve security.

CLIENT CIPHER PREFERENCE—ANALYSIS AND RESULTS

The second part of the research is carried forward from our 2017 report, checking the most common ciphers from a browser perspective. The overall trend from our 2017 study is still relevant as it highlights ECDHE overtaking RSA ciphers. Any time we found a cipher proposal in a particular client, we incremented the count. This provided us a good idea about how common certain ciphers are in a browser's proposals.

Most Popular Ciphers

Ciphers	Occurrence Frequency
TLS_RSA_WITH_3DES_EDE_CBC_SHA	98%
TLS_RSA_WITH_AES_128_CBC_SHA	95%
TLS_RSA_WITH_AES_256_CBC_SHA	91%
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA 116 4	89%
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA 113 5	87%
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA 112 6	86%
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA 110 7	85%
TLS_DHE_RSA_WITH_AES_128_CBC_SHA 92 8	71%
TLS_RSA_WITH_RC4_128_MD5	69%
TLS_RSA_WITH_RC4_128_SHA	69%

This doesn't come as a surprise. RSA has been in existence for the past thirty years, so it's understandable that most browsers support TLS_RSA. In this data, we see not one, but three variants of the TLS_RSA type of cipher is present in almost all the client browsers. This also means that, if the server really wants to use RSA for public key exchange, then there's nothing that can stop it (at least for now). However, as explained earlier, being present in the list of client ciphers doesn't necessarily mean it will get selected. It's the order of preference in both client and server that would determine its selection. For this, we need to dig a level deeper and understand preferences of these ciphers.

RSA was definitely the most common public key exchange, but was it also the most popular?

BEING COMMON DOESN'T NECESSARILY TRANSLATE INTO BEING POPULAR

As we looked deeper into individual clients and their cipher preferences, some interesting and startling facts came forward. We ranked a cipher list of each browser according to their preferences and then graded using a weighted average for each cipher according to their popularity across different browsers and browser versions. Below are few key points of the criteria.

1. For each occurrence of the cipher, we checked if it ranked between 1 to 20 in a browser's preference (for a server to not select them would mean they would have to reject the first 20 proposals).
2. We rejected the one-off ciphers by having the criteria that across all the browsers the cipher should appear at least 20 percent of the time. Unfortunately, this may eliminate newer and upcoming ciphers like TLS_GREASE, but we do not mind as we are going for the big picture in this report.
3. There was also another restriction imposed that it should have a preference between 1 to 20 at least 20 percent of the time out of its total occurrence. This would eliminate the not-so-popular or the one-off exception ciphers from the analysis.
4. Based on the above three elimination criteria, the ciphers were given weighted ranks across browsers and then aggregated, scored, and converted into percentile.

[Note: There might be other statistical methods to form a more accurate calculation, but since this paper is not concerned about granularity of cipher types, a simple statistical average was enough to showcase the big picture.]

It's fairly difficult to know the ratio of different types of browsers in use at any point of time. However, using some intelligent guesses, giving higher weight to popular browsers like Mozilla, Chrome, and Safari, we created the ranks. We understand that there is a significant number of users who may be still use several-year-old browsers (WannaCry exposed the rampant use of end-of-support Windows XP and Internet Explorer). The first aggregation of ciphers was done for all browser versions between late 2011/early 2012 until the first half of 2017. However, to see more-granular trends, we also did aggregations between 2012 to 2014 and another one between 2015 to first half of 2017.

[Note: In all the proceeding graphs and tables, a higher numerical value of rank signifies lower preference.]

The survey included both older and newer browsers. Attacks like Wannacry exposed the rampant use of older operating systems and possibly older browsers as well.

Analysis of All Browser Versions Aggregated Between 2012-2017

Ciphers	Percentile Score	Relative Rank
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	100.00	1
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	93.65	2
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	93.59	3
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	88.41	4
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	86.60	5
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	86.60	6
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	84.45	7
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	81.54	8
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	77.69	9
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	77.17	10
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	72.28	11
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	72.28	12
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	71.75	13
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	69.19	14
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	68.26	15
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	65.81	16
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	62.49	17
TLS_ECDHE_RSA_WITH_RC4_128_SHA	60.10	18
TLS_RSA_WITH_RC4_128_SHA	59.41	19
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	58.88	20

ECDHE is definitely the most popular key exchange method.

After compilation of the large chunk of data, the end results were a big surprise for us. We did expect ECDHE to rank quite high, but we had not thought of RSA's aggregated rank being placed as low as it is in the tables with the first RSA proposal getting a rank of 19.

[Note: This table is an approximation of preferences over a large set of values. Meaning, the cipher at Rank 1 doesn't necessary mean it is at Rank 1 for all the browsers. However, it does signify that whenever this cipher appeared, more often than not it had a higher preference than most of the other ciphers.]

LET'S DIG DEEPER

RSA has been a very important cipher for public key exchange, so one needs to be doubly sure before declaring that it has run out of favor amongst browsers. This is why a little in-depth analysis was necessary on both TLS_RSA and TLS_ECDHE/TLS_DHE to get closer to the real picture. The last time TLS_RSA had a preference between one to five was back in 2014.

Browser Version	Ciphers	Rank	Year
Chrome 32	TLS_RSA_WITH_AES_128_GCM_SHA256	4	2014
Internet Explorer 11	TLS_RSA_WITH_AES_128_CBC_SHA256	1	2013
Firefox 21	TLS_RSA_WITH_AES_128_GCM_SHA256	4	2013
Opera 17	TLS_RSA_WITH_AES_256_CBC_SHA	5	2013
Android_2.3.7	TLS_RSA_WITH_RC4_128_MD5	1	2011

RSA has been a very important cipher for public key exchange, so one needs to be doubly sure before declaring that it has run out of favor amongst browsers.

We also looked at some prominent TLS_RSA cipher suite preferences (lower number means higher preference), along with different versions of the same browser and checked the trends.

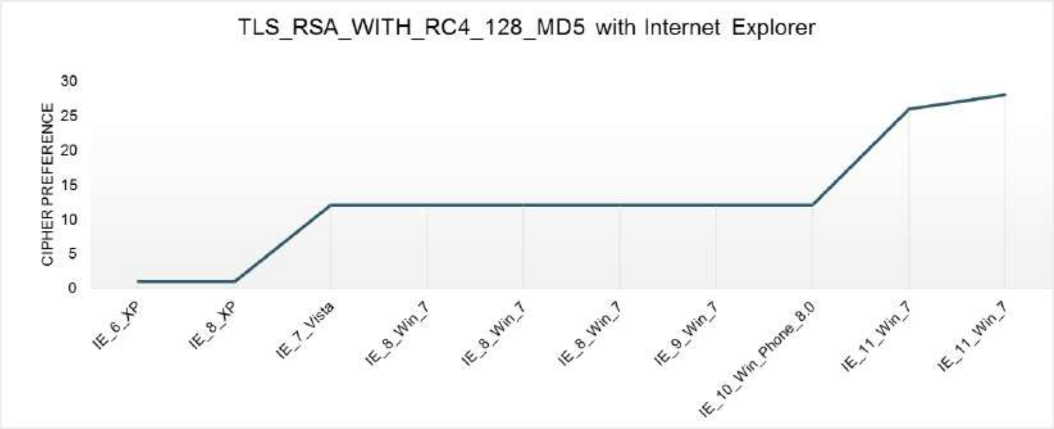


Figure 6

The decreasing preference of RSA on Internet Explorer.

Internet Explorer always had a preference for TLS_RSA, however, at later versions the RSA priority did take a nose dive. This became more profound with Edge, where all types of RSA ciphers ran out of favor with "TLS_RSA_WITH_AES_256_GCM_SHA384," making it rank 15 for Edge version 13.

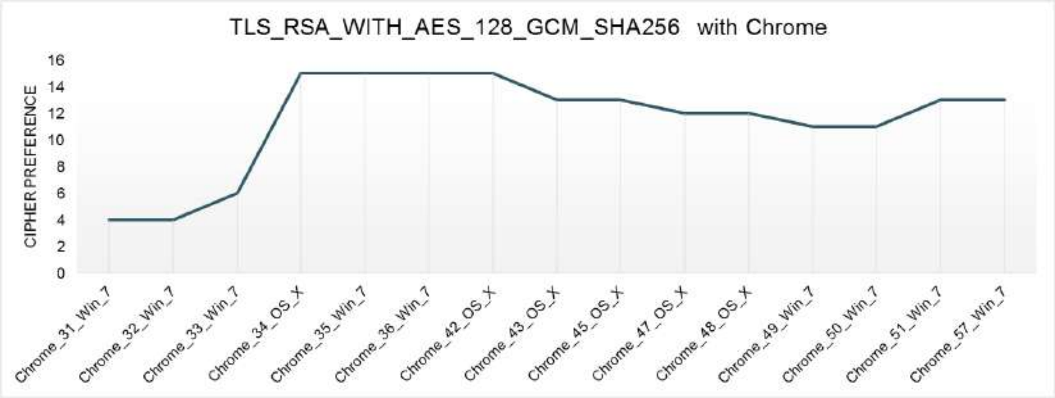


Figure 7

The initial decline was rapid for the RSA cipher in Chrome, followed by a plateau.

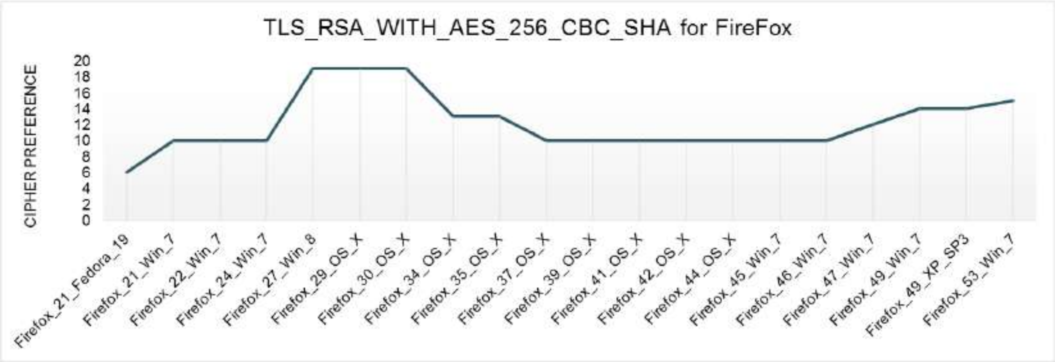


Figure 8

RSA didn't do too well with Firefox, with RSA running out of favor in later versions.

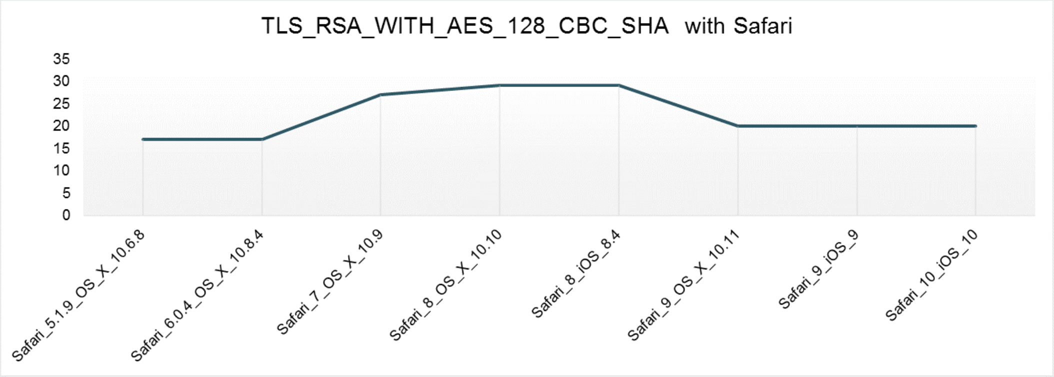


Figure 9

Safari cipher preference for another prominent RSA cipher suite.

For Safari, Chrome, or Firefox, the trends are pretty consistent. A few one-off versions had the TLS_RSA public key exchange as a preference, but for the most part, TLS_RSA had a lower preference and it either decreased or plateaued over time

Now, if the overall encryption has gone up but there is a decline in TLS_RSA, this would mean that there is something else that is increasing in use. It turns out that TLS_ECDHE is this something that has replaced TLS_RSA. Different variants of TLS_ECDHE are consistently making the top ranks, so as far as the browsers are concerned, they certainly have picked their favorites.

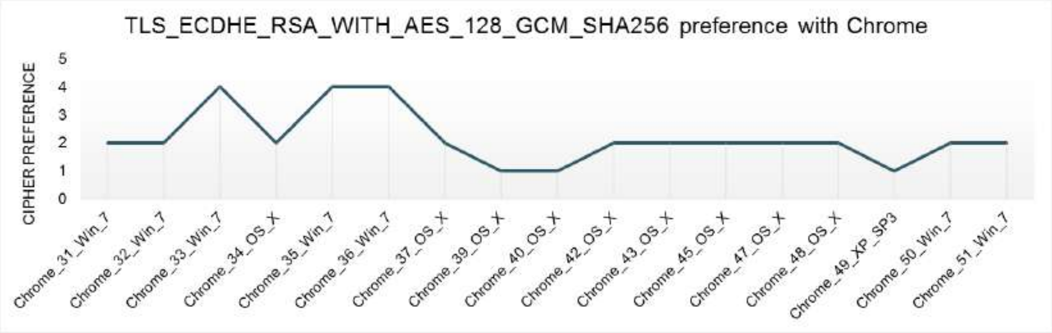


Figure 10

Unlike RSA, prominent ECDHE cipher suites consistently have better preferences.

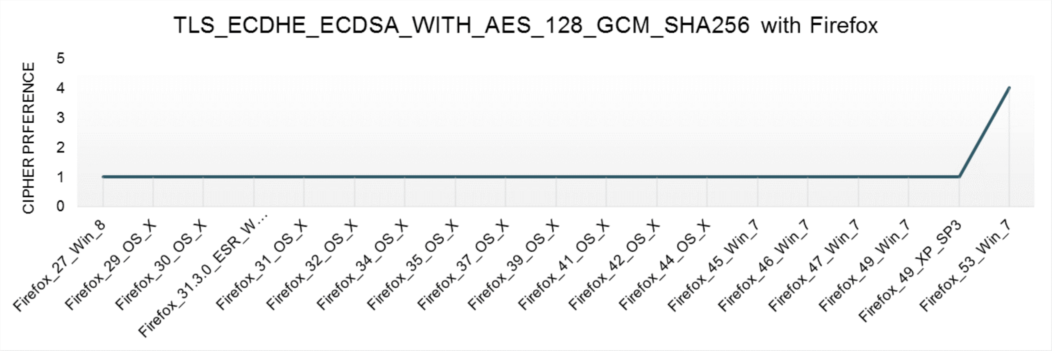


Figure 11

ECDHE_ECDSA is definitely the Firefox favorite.

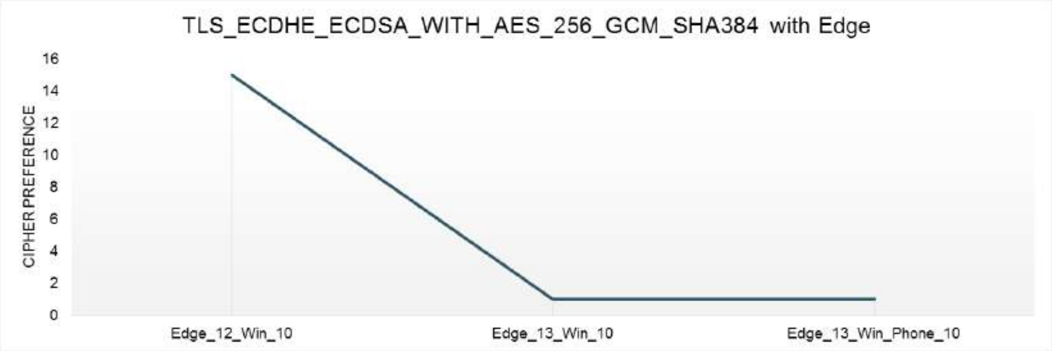


Figure 12

Edge browsers dumped RSA and replaced them with ECDHE.

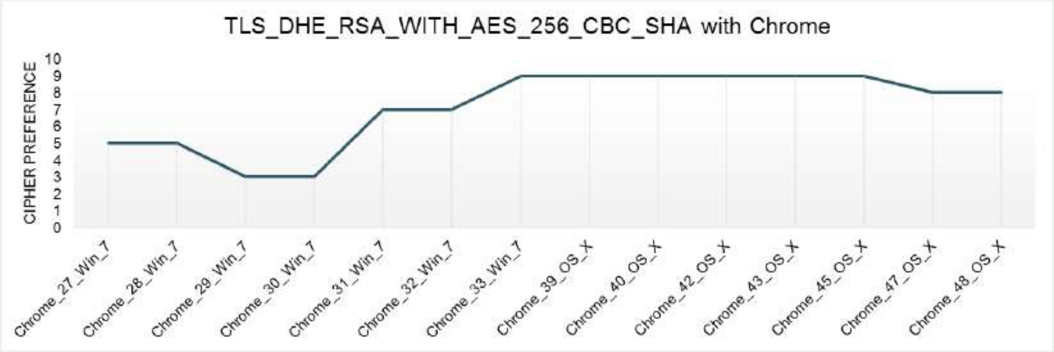


Figure 13

The less-efficient DHE doesn't share as much love, but it still is more popular than RSA.

WHY THE LOVE FOR TLS_ECDHE?

Elliptical Curve Deffie-Hellman Ephemeral (ECDHE) is a classic case of having your cake and eating it too. The Deffie-Hellman algorithm ensures forward secrecy by use of unique server private keys per session and the use of Elliptical Curve significantly reduces the size of the keys without compromising security. So overall, you are getting higher performance with higher security, a rare win-win in the world of security where normally an increase in security brings in additional delays in the business processes.

The one short coming is ECDHE's lack of signing algorithm within the cipher, which is why it has to use either RSA or DSA for signing. Based on the analysis, it seems that initially ECDHE was using RSA for signing, but in recent times it's increasingly using Elliptical Curve-DSA or ECDSA for signing. Coupled with the power of Elliptical Curve for both key exchange and signing TLS_ECDHE_ECDSA's steep rise in popularity isn't a surprise.

ECDHE's growing popularity can be attributed to it providing the dual advantage of higher performance along with higher security.

OTHER COROLLARIES OF THE BROWSER RESEARCH

SSLv2 Is Dead, RC4_MD5 Is Not

It can be said with strong conviction that SSLv2 (or older phased-out SSL versions and SSLv3) also has lost its prominence significantly. We can say this with confidence because even though a server may choose a weaker cipher suite from the list of proposals a client sends, it cannot choose a cipher suite that is not present in the clients list. Hence, by removing the vulnerable ciphers like SSLv2 from their list, browsers are ensuring they are phased out. This isn't the case for all vulnerable cipher suites. Even though the issues with TLS_RSA_WITH_RC4_128_MD5 are well proven, it's still found in more than 75 percent of the browsers that we polled.

By removing the vulnerable ciphers like SSLv2 from their list, browsers are ensuring they are phased out.

Browsers Are Not Falling for Non-Ephemeral Ciphers Anymore

ECDH, which is the non-ephemeral version of ECDHE, had a fixed DH key. The thought was to combine the Elliptical Curve's highly secured short key with RSA's fixed private key to provide a double whammy on performance. However, the fact that the fixed key doesn't provide forward secrecy (compromise of the long-term DH key would result in someone decrypting all previous and future sessions) has made it quickly fall through the ranks and by now its average preference is at the south of 20. This would also suggest that the browsers are displaying strong preference towards forward secrecy and clearly understanding the advantages and client anonymity that it provides.

In Symmetric Encryption, the GCM Is Gaining Preference Over CBC

Although the report is geared towards the public key exchanges, we did take a look at symmetric encryption. Symmetric key cryptography is generally done using either of the two major algorithms, Cipher Block Chaining (CBC) or Galois/Counter Mode (GCM). Without going into the details of their workings, overall GCM is considered more secured and efficient as it provides encryption and integrity at the same time. Use of both ciphers is still rampant in the world of browsers. However, of late a certain preference for GCM has been noticed, with the browsers with the cipher suites having GCM in symmetric encryption having a higher chance of being picked than CBC. This also indicates intelligence on the browser's part in selecting the cipher suites that would provide better performance to the users without compromising security.

Key Size and Curves Follow NIST Recommendations

For RSA, 2K is the standard key size preferred by the clients, while for ECDHE, the P-256 curve is the most popular curve. This is well-aligned with National Institute of Standards and Technology (NIST) recommendations for the minimum key length.

Chrome Browsers Are the Bellwethers

Trends indicate that Chrome, very closely followed by Mozilla, will adapt to the newer ciphers first. It's also notable that most cipher suites become popular once either Chrome or Mozilla decide to adapt to and promote them.

Sometimes the upgrade cost of encryption infrastructure may prove to be a hindrance for a server to use better ciphers.

CONCLUSIONS AND RECOMMENDATIONS

With this data, we can accurately form several conclusions. RSA is fading and it's fading fast—and the browsers and the servers are hand-in-glove in making that change happen. With that, the state of encryption also is changing. This calls for a major shift in encryption strategies for those vendors and network procurement teams that want to optimize device and network performance and invest in the best encryption, firewall, and network monitoring infrastructure.

Encryption strategies must change for those who want to optimize device and network performance.

VENDORS

Optimize for ECDHE in Your Hardware and Software

Some crypto hardware and software has been optimized for RSA over time so that they are efficient in calculations that are involved with such large keys. ECDHE changes the paradigm completely. The keys are smaller, but now placed in a curve. Also, with no long-term keys being used, the server has to generate a private key every single time. The change in dynamics we've uncovered in this report shows an immediate need to calibrate both hardware and software to ensure the devices efficiently handle both ECDHE_RSA and ECDHE_ECDSA public key exchanges. Doing so will go far in helping your products outperform the competition for clients and servers that prefer the dominant cipher today (ECDHE_RSA) as well as the up-and-comer (ECDHE_ECDSA).

TLSv1.3 is here to stay with more servers and browsers adopting it at a breakneck speed. This makes ephemeral keys mandatory and the servers still using non ephemeral ciphers are a tiny, almost inconsequential, minority.

Most browsers support HTTP2.0 and more servers are showing affiliation with HTTP2.0 than HTTP 1.1.

Both of these mean that the customers will very soon demand HTTP2.0 and TLSv3 performance numbers along with TLSv1.2 performance numbers with ECDHE (ephemeral) type ciphers.

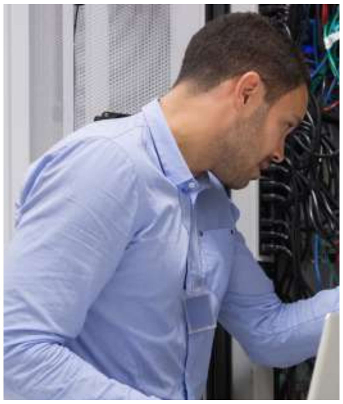
NETWORK OPERATORS

Re-Design Monitoring and Re-Think Procurement

Over time, passive monitoring has continued to flourish. Inputting long-term keys into monitoring devices means a monitoring device can decrypt traffic and send it to different tools without the need of terminating and re-initiating encrypted sessions. This was great for information security groups as they can deploy all aspects of monitoring without slowing down the critical business path.

With ECDHE, such long-term key techniques can't be applied and every session needs to be terminated and re-initiated. This causes delays in the business path. To ensure unchanged monitoring capacity with active encryption, the infrastructure will most definitely need to be upgraded and may need some re-designs.

A quick survey of available vendor data sheets indicates that most of the performance numbers for encryption are calculated with RSA-type ciphers. Unless your network is optimized for TLS_RSA, these performance numbers will not be relevant. You will need to check the performance with both ECDHE_RSA and ECDHE_ECDSA to gauge the actual performance of encrypted traffic on your particular network.



The world is getting encrypted and the encryptions themselves have changed. You need to update and optimize your infrastructure to keep up with the changes.

Learn more at: www.ixiacom.com

For more information on Ixia products, applications, or services, please contact your local Ixia or Keysight Technologies office. The complete list is available at: www.ixiacom.com/contact/info