Critical Review of Imperfect Forward Secrecy

Eyal Ronen, Adi Shamir
Computer Science department
Weizmann Institute of Science
Rehovot, Israel
{*eyal.ronen,adi.shamir*}*@weizmann.ac.il*

***Abstract*—We reviewed the claims made by Adrian et al., in [1] that precomputation of specific DH groups provides the NSA with the ability of mass surveillance of Internet communications and passively eavesdropping to large percentage of IPSec and TLS connections. We have independently reached essentially the same conclusions as Wouters in [2], namely, that the success rate of a hypothetical NSA attack on IPSec, would be much lower than the original estimate which appeared in [1]. More interestingly, we also checked the claimed statistics in [1] related to HTTPS connections (which was not checked by Wouters). These connections use the TLS protocol (or its predecessor SSL) to securely access HTTP servers on the Internet, and is of great interest to intelligence services since it protects access to search engines (such as GOOGLE), to email (such as GMAIL), to social networks (such as FACEBOOK), to financial information (such as CITIBANK and VISA) and to various services (such as ordering books on AMAZON or reserving airline tickets on EXPEDIA). Our independent tests show that even massive precomputation applied to all DH groups will have very limited success in passively breaking interesting HTTPS connections, which are used to access the most popular sites on the web. We have also found several methodological problems in the current implementation and interpretation of Internet wide HTTPS scans by based on the ZMap scanning software [3] used for generating the statistics shown in [1] and in the DROWN attack paper by Aviram et al., [4]. For example not implementing support of URL redirection caused many sites to be excluded from the scans, while not implementing host name validation cause mis-configured sites to be considered as valid sites that will be trusted by browsers.**

We have also reviewed the Snowden documents referenced by [1]. In those and other Snowden documents we have found clues that the NSA use other attack vectors to decrypt IPSec and TLS traffic. The recent Juniper and Cisco back-door revelations support this hypothesis.

1. Introduction

Documents leaked by Snowden have shown us the extent of the NSA's surveillance program and it's ability to decrypt data encrypted by many modern protocols, including the widely used IPSec protocol used for protecting VPNs (Virtual private network) and TLS protocols used to protect HTTPS Internet traffic to web servers. Although the extent of the surveillance was made clear, the specific methods that enabled it was not explained. Adrian et al., [1] claimed that a plausible explanation is a NSA ability to break the discrete logarithm DH (Diffie Hellmann) protocol [5] based ephemeral key exchange. The DH key exchange is used by many protocols such as SSH, IPSec and TLS. The mentioned protocols allow the user to configure the preferred key exchange algorithm (for example DH or ECDH - Elliptic Curve DH) and the specific cryptographic parameters of the algorithm such as the DH group. Adrian et al., have claimed the NSA can use massive precomputation of DH groups of up to 1024 bits sizes to efficiently compute discrete log in the group and passively break the key exchange protocol. They have demonstrate their ability to precompute and then break 512 bit size groups almost in real time, and estimated that 768 and 1024 bit size can be broken by the NSA. Although it was shown how to use this ability for active MITM (Man In The Middle) downgrade attacks, they claim that a large percentage of popular HTTPS sites can be passively broken as they used DH algorithm with groups of up to 1024 bits size by as their default chosen algorithm. In order to test this claim they used the ZMap scanning software [3] to . ZMap can efficiently connect using a chosen protocol to a large number of IP addresses, and save the results of the handshakes. This allows the user to gather statistics on servers algorithm support, specific configuration and vulnerabilities in implementations of security protocols. This can be repeated on a daily basis to show trends and changes over time. For example [6] stores the results of HTTPS handshake TLS for the Alexa top 1 million sites on a daily basis.

In this paper we review the claims made in [1] about the affect of massive precomputation on IPSec and TLS connections, and show that the effect exist primarily at the least popular sites in the Alexa list.

2. Effect of DH precomputation on IPSec

Wouters [2] gave a detailed account of different biases resulting from the scanning methodology employed in [1] for IPSec connections. The main conclusion that we have also reached independently is that it is impossible to determine the percentage of IPSec connections using the DH group of 1024 bit and under, without wide access to real Internet traffic. One reason is that in many cases correctly configured IPSec endpoint will not be detected by the scans made in [1]. For example, large organization may use MPLS VPN (MultiProtocol Label Switching Virtual Private Network) and will not have an address in the public IP domain. Another example is IP white-listing that might be used to prevent access from unauthorized IP addresses. As stated in [2] the majority of IPSec endpoints detected were omitted from the result, as they responded with a NO-PROPOSAL-CHOSEN. A response that many securely configured IPSec endpoint will reply. Combining those factors causes a strong statistical bias towards less secure configurations.

Although we can not determine the actual statistics of various configurations, we can look at the configurations that are recommend by the vendors of IPSec VPN, and assume the most security aware users follow this advice. For example, in a user guide from 2012 [7] Cisco recommends configuring her IPSec VPNs to use 2048 bits or larger DH or ECDH.

3. Effect of DH precomputation on TLS

We have conducted several scans of the Alexa Top 1 million sites for TLS support, from November of 2015 till March of 2016. Our goal was to independently verify the results shown in [1] and to examine the proper methodology for conducting wide range tests and interpreting and displaying the results.

3.1. Usage of DH in TLS/SSL

DH is one of the key negotiation algorithms supported by TLS along with RSA and ECDH. DH and ECDH have the prefect forward secrecy property that protects the current communication from future key theft. In the TLS protocol the client sends the server a list of supported ciphers suite, usually in order of preference. The server can choose any one of them, or abort. The standard does not state how the server should choose the cipher suite, and he can ignore the preference order sent by the client. The TLS server also determine the cryptographic parameters of the chosen cipher suite. To be more specific, if the server choses to use DH as key negotiation algorithm, he sends to the client his choice, along with the specific parameters of the DH group, such as size and modulus. In the case of active attack, it is enough for the client to support a weak cipher suite, or weak parameters, to allow an attacker to perform a MITM attack (as been done in the active attack described in [1]). As we are testing the claim for the possibility of passive breaking of the connection, the choice of algorithms and parameters by the server from the lists send by modern browsers, will decide if a passive attack is possible or not.

3.2. Internet wide TLS scans

Internet wide scans were used in [1] to demonstrate that precomputation on DH groups will allow the NSA to passively decrypt large percentage of the Internet's TLS traffic. To prove their claim they chose to sample all the sites in Alexas Top 1 million most popular sites list. We believe there is a problem in the statistic measured and shown in [1], and the methodology used to test them.

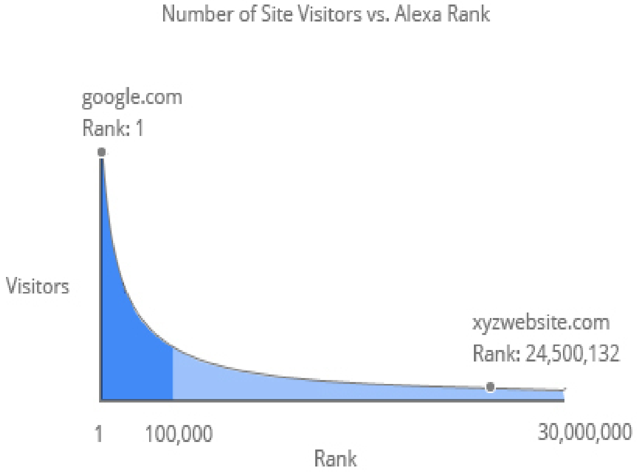


Figure 1. Approximation of visitors number Vs. rank

TABLE 1. EXAMPLES OF SITES' TRAFFIC

Site rank	Site	Hits per day	Visitors per day
1	google.com	12 billion	610 million
5	yahoo.com	1.3 billion	160 million
10	twitter.com	470 million	93 million
50	google.es	250 million	21 million
101	google.co.kr	94 million hits	8.7 million
1002	pbs.org	4.3 million	1.7 million
9980	ncaa.com	310000	160000 visitors
100009	cryptosam.com	43000	17000
1000000	decoandkids.com	8000	1600

3.2.1. The long tail of insecurity . As seen in figure 1 taken from [8], The number of visitors per site drops exponentially vs the site's rank. In Table 1 we can see the number of visitors and page hits of different ranking site estimated by Alexa in January of 2016. In [9], Alexa claims that There are limits to statistics based on the data available. Sites with relatively low measured traffic will not be accurately ranked by Alexa. We do not receive enough data from our sources to make rankings beyond 100,000 statistically meaningful. The long tail of lower ranking site has a very low visitor number. Although a specific user may be effected by the long tail (for example if he reuses his password), those sites are probably less of prime target for organizations like the NSA. One can also assume that smaller sites, will have less means or incentive to invest in security, and keep their TLS servers up to date with the best practices. This assumption is supported by our independent scans.

3.2.2. Scan statistics by site rank. We have conducted independent scans to verify the results of [1]. We have conducted the first scan in November of 2015, shortly after the publication of [1]. As the TLS servers' eco system takes

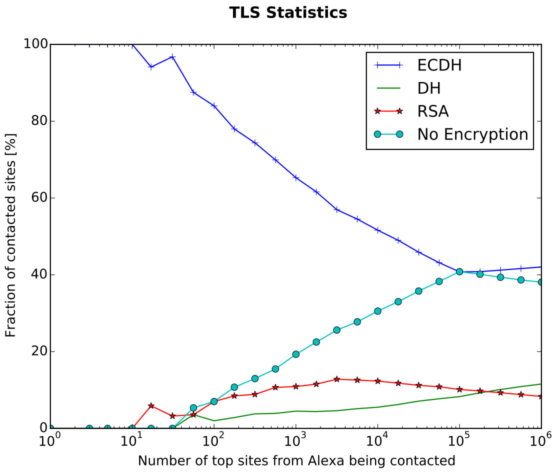
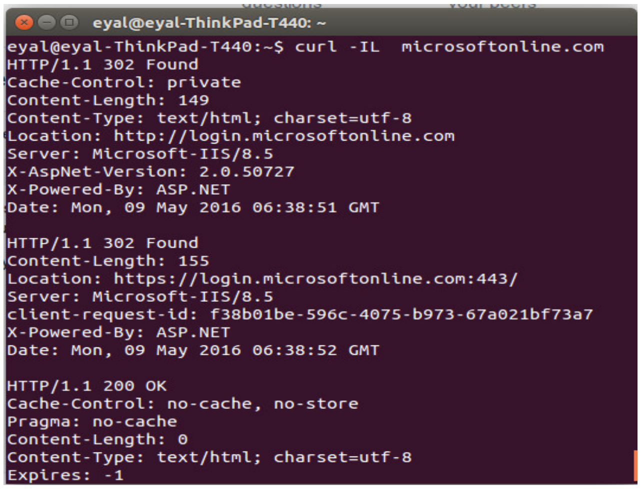


Figure 2. Measured ciphers distribution Vs. rank



```
eyal@eyal-ThinkPad-T440:~$ curl -IL  microsoftonline.com
HTTP/1.1 302 Found
Cache-Control: private
Content-Length: 149
Content-Type: text/html; charset=utf-8
Location: http://login.microsoftonline.com
Server: Microsoft-IIS/8.5
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Mon, 09 May 2016 06:38:51 GMT

HTTP/1.1 302 Found
Content-Length: 155
Location: https://login.microsoftonline.com:443/
Server: Microsoft-IIS/8.5
client-request-id: f38b01be-596c-4075-b973-67a021bf73a7
X-Powered-By: ASP.NET
Date: Mon, 09 May 2016 06:38:52 GMT

HTTP/1.1 200 OK
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Length: 0
Content-Type: text/html; charset=utf-8
Expires: -1
```

Figure 3. HTTP headers of URL redirection

a long time to adopt security configuration recommendations, and upgrade to newer versions, we believe this time difference to be too short for large changes in the TLS statistics.

Our full scan of all the top one million websites showed that DH-based connections were established to around 18.7% of the sites that accepted HTTPS. This is a slightly smaller percentage than the 23.9% described in [1], which refers to the fraction of HTTPS connections that use the ten most popular 1024-bit DH groups (note that this discrepancy could be due to natural trends in the use of cryptography on the Internet, due to the time difference between the two scans, or to the slightly different sampling methodology). However, our main claim is that most of these 1 million web sites have a relatively small amount of traffic and are of little interest to intelligence services, and thus this statistics is not the right one to use when trying to estimate the possible success rate of a hypothetical NSA attack. We thus compiled the fraction of the different cipher selected by the servers among the attempted connections to the top k web sites for all values of k smaller than one million as seen in figure 2. For example, when we restrict our attention to the top 1000 web sites, the percentage of sites whose connection used a DH handshake drops to 4.5%, and if we consider only the top 100 web sites (which include most of the interesting sites mentioned above), the percentage drops to just 2%. In other words, the most popular web sites are the least likely to negotiate a DH handshake when the client is not actively modified or influenced by the NSA.

3.2.3. TLS scanning methodology. Our first methodology was to use OpenSSL version 1.0.1e-fips from 11 Feb 2013 in its standard configuration (the version installed on our cluster computer at the Weizmann Institute). We tried to open a secure connection to every single site in the Alexa list of the top one million web sites, and recorded the TLS handshake. Since many sites either declined a TLS connection, we also heuristically tried to add either the ”www.” or ”login.” prefix to all of the sites on the list. We analysed our results, and compared them to the results obtained by using ZMap as was done in [1], and the way browsers handle selected sites. For our comparison with ZMap, we examined the HTTPS top 1 million sites scan results saved in [6], dating to the 9th of March 2016, and found several issues, that can affect the statistics. We found that many sites were not included in the results (over 30% of the top 1 million and 22 of the top 100). 17 out of those 22 were found to support encryption but were omitted from the ZMap gathered statistics. On the other hand we found sites that were falsely sampled as browser trusted (not browser trusted site will complete the TLS handshake but will cause a warning in the browser due to certification issues). We have found some overlooked issues in the ZMap scanning methodology that explain some of the discrepancies we found in the results.

3.2.4. Support of URL redirection. When trying to access a site like microsoftonline.com, the browser sends a query to the DNS server, and receives the site IP address. After that he sends a HTTP GET request to that IP address. In many cases the site will replay with a 302 HTTP response code and will redirect the browser to another address, this may happen multiple times. In our example microsoftonline.com will be redirected to http://login.microsoftonline.com and then redirect again to https://login.microsoftonline.com, as can be seen in figure 3. Another example is soso.com that redirects to http://www.sogou.com. URL redirection is used in many different ways. In many cases a server will only respond on a URL with the www prefix, URL redirection helps the site support users accessing the site without the prefix. The site can use URL redirection to redirect browsers to HTTPS address and in that way instruct the browser to use encrypted communication. Many of the sites will not respond to a TLS request on the IP address of the URL listed in the Alexa top 1 million list. Although trying the two prefix of www and login heuristically solved many of the issues in our first tests, the proper way is to simulate the actions taken by browsers. This should be done by first sending a HTTP GET request and follow URL redirection. Some sites use the URL redirection to redirect to webpages that were customized for specific browsers, it is important to include a User-Agent string in the HTTP GET request, that is typical of a modern browser. We manually checked the 22 missing sites in the top 100 (for example microsoftonline.com) with modern browsers (Chrome and Firefox), and then tried connecting to the redirected address using OpenSSL. Out of the missing 22 sites, 12 of them after answered after redirection or by adding the login prefix. Some of those sites even use encryption by default (redirection to HTTPS address).

3.2.5. Support of encryption on subdomains . Some sites only offer encryption when user try to login, usually at some subdomain. Some of the more common examples we found are the login prefix (for example login.yahoo.co.jp), and the mail prefix (mail.qq.com). Those sites were missing from the ZMap statistics. As some sites use less common subdomains such as nid.naver.com it is not trivial to find those encrypted sub domains. One options is to download the main webpage of the site, and try and find links to HTTPS address of in a subdomain. However this might be very time consuming and error prone.

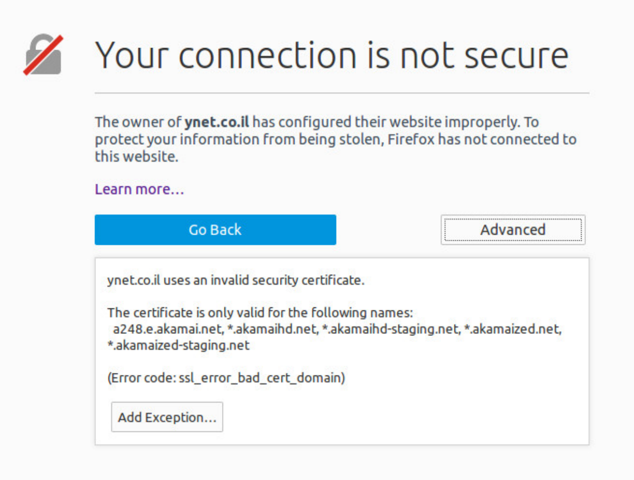
3.2.6. Host name validation. In the TLS protocol the server proves his identity to the client browser by providing a certificate. This certificate includes a chain of digital signatures (each link signs the public key and some meta data of the next link), that originates from a CA that the client knows his public key in advance. An outline of the verification process includes the following steps:

1) Find in the signature chain a CA whose public key you know and trust.
2) Verify the rest of the chain starting from the CA.
3) Verify that the site you are trying to access is the site that was signed in the certificate. This is called host name validation.

Without host name validation the entire security of TLS breaks. For example, I can do the following attack:

1) Buy the domain www.site-certificate-for-MITM.com.
2) Acquire a legitimate certificate for it from a well-known CA.
3) Hijack a connection of of a client to google.com, and provide the certificate valid for www.site-certificate-for-MITM.com in the TLS handshake.
4) If the client won't verify the site name he will thinks he is connected to google.com.

All modern browser perform this validation, and will give a warning to the user if it fails. Verifying the host name is complicated to do for various technical reasons, and it is sometimes overlooked (for example OpenSSL did not have built-in support for host name verification till January 2015 [10]). It seems that the scans made with ZMap do not perform this validation. In one example, in the ZMap's results the site ynet.co.il was marked as browser trusted. This is a newspaper site that does not employ TLS. It has a TLS server enabled (probably a default configuration) but the certificate domain common name is a248.e.akamai.net, and trying to surf to the site with HTTPS in a browser invokes a warning, as can be seen in figure 4.



Your connection is not secure

The owner of **ynet.co.il** has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

ynet.co.il uses an invalid security certificate.

The certificate is only valid for the following names:
a248.e.akamai.net, *.akamaihd.net, *.akamaihd-staging.net, *.akamaized.net, *.akamaized-staging.net

(Error code: ssl_error_bad_cert_domain)

Figure 4. Firefox's host name validation error message

3.2.7. SNI (Server Name Indication) support. A single TLS server with a single IP address may support several domains. For example Google may decide that one server will support both gmail.com and google.com, or a hosting service will use one server for multiple domains bought by multiple clients. The server may have a different certificate for each site (for example the different clients do not want to share the same private key). As described before, in the TLS protocol, the user opens a connection to a IP address he received from a DNS server. The specific URL he wants to get will be sent to the server after the TLS negotiation will be finished successfully. This means that the server cannot know the URL in advance, and will not know what certificate to present to the client. This may cause the host name validation on the client side to fail. This problem is solved by the SNI extension that allows the client to send the requested site in the TLS protocol. If a scan does not implement the SNI extension some sites may either reject the connection or present the wrong certificate and cause the host name validation to fail. As ZMap did not seem to implement host name validation, we do not know if this effected their measurements or not.

3.2.8. Phantom encryption. Some sites might allow a client to successfully negotiate a valid TLS connection when connected directly to their IP address, but a browser sending a HTTP request for a page will not be directed to use encryption. This may be caused for several reasons:

1) Some sites do no offer TLS encryption by default.

As mention before a site may redirect a browser from HTTP URL to HTTPS URL to force encryption. Not doing this will cause some browser to work unencrypted. Some extensions such as HTTPS Everywhere, were designed to solve this issue, but are not used by most users.

2) A site might have a configured support for TLS due to service provider or for future usage, but will not offer any pages encrypted. Trying to access any webpage in the domain under TLS connection will result in 404 page not found error.

Although a site like this will be recorded in the statistics, in reality it does not offer any encryption. We can assume that such sites will probably have lower level of security and their TLS software and configuration will not be updated regularly.

3.2.9. Unexplained discrepancies. While the points reviewed in the previous sections explain some of the discrepancies we have seen, they do not explain other examples (such as wordpress.com and wellsfargo.com) that were omitted from the ZMap statistics. Further checks should be made to explain them.

3.3. Weighted internet TLS statistics

When assessing the trends in the TLS eco system, we believe there should be some weighted averaging, as a problem in google.com has much more effect then one in a low ranking site. However determining the weights is non trivial. One simple method will be to use the visitor number as a weight, however this might be very inaccurate. As mentioned in the previous section, unless the site support encryption by default using URL redirection, we do not have a way to assess how many visitors if any accessed encrypted pages. Another method will be to assess the importance of the site's privacy and authenticity. Big social networks, search engines and banks will be of high weight while the connection to a site like Wolframs Mathematics will be of lower weight. However this assessment is difficult to do and is very subjective. Unless services like Alexa and Google Analytics will provide us with this kind of information, we believe our best option is to use our methods of displaying separate statistics for higher and lower ranking sites.

4. Analysis of possible NSA attack vectors

Recently discovered back-doors and the leaked Snowden documents give us a glimpse of some of the attack vectors that are used by the NSA, and suggest a more active approach by the NSA to break IPSec and TLS.

4.1. Known backdoors in PRNG

Over the years the NSA has been known to invest a lot of effort in PRNG back-doors. Such a back-door allows the attacker to determine the random noise used in the DH key exchange and passively break it. One well known example is the Dual-EC-DRBG (Dual Elliptic Curve Deterministic Random Bit Generator) back-door involving the RSA company and NIST, and published in the Snowden's leaked documents. A more recent discovery is a similar back-door in Juniper's OS. That back-door was most likely planted by the NSA before it was hijacked by some other unknown attackers.

4.2. Suggestions of active IPSec attack in the Snowden documents

We have reviewed the Snowden documents published in [11] and referred to in [1] to support their claim. We have found evidence that suggest active attacks by the NSA.

1) In [12] on page 3 it is written: ...CES generally requires the packets from both sides of an IKE exchange and knowledge of the associated pre-shared key in order to have a chance of recovering the key.... Note that in IPSec the pre-shared keys are only used for authentication, and not in the key derivation process. This fact strongly suggests an active exploiting process such as a MITM attacks.
2) In [13] at slide 40, a study case is described of exploiting IPSec using an implant in the device. With the implant it is only required to record the ESP data for decryption.
3) In [14] in slide 18 and onwards there is a detailed explanation on extracting configuration data, passwords and pre-shared keys from configuration files of different routers companies. Again this suggests an active cyber or MITM attacks on the VPN networks.

4.3. TLS statistics and attack vectors

It is interesting to note that some of the leaked Snowden documents suggests that DH-handshakes were rarely encountered by the intelligence services. One example is [15] that probably dates from the end of 2012. It explicitly states that at that time DH protocol was used in only 5% of the TLS traffic they observed. As for the relevance of active MITM, we believe it is much more cost effective to steal or forge a CAs private key or certificate. This will allow the attacker to perform MITM attack on all connections regardless of supported cipher suite on client or server. Techniques to prevent this type of attacks, such as certificate pinning, were not widely used till recently.

5. Future work

We should continue working on a proper methodology of understanding the impact of security vulnerabilities on the Internet. We currently do not have the required measurement metric. One option is to work with services like Alexa and Google Analytics to add a metric measuring the number of visitors and page hits using encrypted connections. Another option is to compile a list of interest sites divided by sectors. For example banking, mail services, and social media sites security are of high interest.

6. Conclusion

While we have no information about whether the NSA is able to break a few 1024 bit DH keys with a truly massive preprocessing, we can safely conclude that even if they were successful in preprocessing all the DH groups ever used on the Internet (of arbitrarily large sizes), they would be able to passively break only a few percent of the HTTPS connections that really interest them. We can collect statistics by scanning the entire IPv4 space, and thus we can understand the possible impact of newly discovered weaknesses and to track the reaction time it takes to fix them. However, we should be careful with the methodology we use to gather these statistics, and the way we interpret them. Since it involves many subtle points which can affect the collected data and it's analysis.

References

[1] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta *et al.*, "Imperfect forward secrecy: How diffie-hellman fails in practice," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 5–17.

[2] P. Wouters. (2015) 66https://nohats.ca/wordpress/blog/2015/10/17/66-of-vpns-are-not-in-fact-broken/

[3] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications." in *Usenix Security*, vol. 2013, 2013.

[4] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni *et al.*, "Drown: Breaking tls using sslv2."

[5] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.

[6] Z. Durumeric. (2016) Alexa top 1 million https handshakes. [Online]. Available: https://scans.io/series/443-https-tls-alexa_top1mil

[7] Cisco. (2012) Configuring internet key exchange for ipsec vpns. [Online]. Available: http://www.cisco.com/en/US/docs/ios-xml/ios/sec_conn_ikevpn/configuration/15-2mt/sec-key-exch-ipsec.html

[8] Alexa. Whats going on with my alexa rank? [Online]. Available: https://support.alexa.com/hc/en-us/articles/200449614

[9] ——. How are alexas traffic rankings determined? [Online]. Available: https://support.alexa.com/hc/en-us/articles/200449744-How-are-Alexa-s-traffic-rankings-determined-

[10] O. Wiki. (2015) Hostname validation. [Online]. Available: https://wiki.openssl.org/index.php/Hostname_validation

[11] S. Staff. (2014) Prying eyes: Inside the nsa's war on internet security. [Online]. Available: http://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html

[12] NSA. Turmoil/apex/apex high level description documnet. [Online]. Available: http://www.spiegel.de/media/media-35513.pdf

[13] ——. (2010) Intro to the vpn exploitation process. [Online]. Available: http://www.spiegel.de/media/media-35515.pdf

[14] ——. What your mother never told you about sigdev analysis. [Online]. Available: http://www.spiegel.de/media/media-35551.pdf

[15] GHCQ. (201X) Tls trends at ghcq. [Online]. Available: http://www.spiegel.de/media/media-35512.pdf