Tor vs the NSA

Nicky van Rijsbergen[1] and Kevin Valk[2] *

[1] Radboud University Nijmegen (s4062833)
n.vanrijsbergen@student.ru.nl
[2] Radboud University Nijmegen (s3052273)
kevin@kevinvalk.nl

Abstract

Governmental and powerful organizations' digital spying is exactly what privacy advocates have been warning about for many years. However, the scale of the revealed program by the National Security Agency (NSA) is rather scary, even for experts.

Edward Snowden showed us that the NSA is actively collecting as much information as possible and uses any means possible to achieve this goal. They have, for example, actively influenced various companies' implementations and standards to make sure that backdoors and weak implementations would exist. After the disclosures made by Snowden, many people seek a way to protect their privacy. However, what tools can we trust in the light of recent events?

In this paper we investigate the anonymization tool The onion router (Tor). We looked at how much privacy Tor can provide against an adversary as powerful as the NSA. We looked at available attacks on Tor and how the NSA can use those attacks. We also investigated how likely it is that the NSA can brute force the ciphers that Tor uses. To find an answer, we have made an estimation of how powerful the NSA can be, given certain assumptions.

Keywords: Tor, AES, RSA, ECC, NSA

*This work is supervised by Anna Krasnova anna@mechanical-mind.org.

1 Introduction

In an ideal scenario everyone may decide for themselves whether they want their privacy violated or not. In such a scenario people should be at least aware of any violation of their privacy and are able to stop this violation whenever they want to. Everyone is in control of their own privacy and is able to trust anyone with their privacy. This scenario is ideal because privacy is a right for everyone [45] and on top of that privacy is important to all of us [37].

Unfortunately, the reality is not much like the ideal scenario. Thanks to whistle blower Edward Snowden, it has become more and more clear that the NSA is spying on citizens in ways many people did not expect [1]. At first, people were unaware of this privacy violation. Now they are aware of it, they want to reclaim a part of their privacy, but are often unsure how to do so. One reason for this is because it is unknown what the NSA is capable of. One way many people try to retake some part of their privacy is by using anonymization tools; An example of such a tool is Tor, which is with more than three million users, one of the most used anonymization tools [42].

In order to make the reality more like the ideal scenario, we have researched whether Tor is able to provide privacy against an adversary as powerful as the NSA. We state the following research question: *To what extent can Tor provide protection against the NSA?*.

To get an answer to this question, we have first researched the NSA. What is the revealed power of the NSA (what have they done) and what is the expected power of the NSA (what can they possibly do). The revelations of Edward Snowden were a great help. We investigated what the NSA can do against Tor. The NSA can attack Tor users in two ways, targeted, mostly when there are already reasons to suspect a person, and through mass surveillance, when the NSA just wants to collect information. We have researched several vulnerabilities existing in Tor that the NSA can exploit in targeted attacks. We also looked at whether the cryptography used in Tor is sufficient if the NSA wants to attack it.

This paper is organized as follows. An introduction to Tor is given in section 2. Section 3 gives a global overview of the NSA and explains why the NSA is considered as an adversary. Currently known attacks on Tor will be described in 4. This section will also give an overview of what ciphers Tor uses and will give a detailed analysis of how secure these ciphers are. Section 5 concludes this paper.

2 The onion router

The onion router (Tor) is open source software that is capable of keeping anonymity while using the internet. Tor was originally designed, implemented, and deployed as a third-generation onion routing project of the Naval Research Laboratory with as primary use protecting government communication. Nowadays everyone can use Tor, i.e. journalists, military people, activists, whistle-blowers, criminals and normal people. Tor has over three million users as of December 2013 [42]. Tor was developed by the U.S. Naval Research Laboratory [44], and is currently sponsored by

various companies and over 4600 personal donors [43].

Tor bounces traffic throughout the Tor network to hide the real sender. Outsiders see only a big Tor network and they do not know where packets that come out of the network came from.

2.1 Global overview of Tor

Tor provides anonymity by routing user's traffic through several Tor nodes, usually three. Tor nodes are computers running Tor software. Every Tor node does not know either the origin of the message, the destination of the message or both. The entry node knows who the originator is, the exit node knows who the receiver is and the relay node knows neither. The endpoints in the path (namely the sender and the end point in the Tor network) encrypt the original packet with X layers of encryption. Where X is equal to the path length. This is the main reason why Tor is called the onion router [20]. Because of this only the endpoints know the original content. Tor used to use RSA encryption, but the newest version of Tor uses elliptic curve cryptography (ECC) [30]. For efficiency, the Tor client uses the same path for connections that happen within the same ten minutes. Later requests are given a new path, to keep adversaries from linking your earlier actions to the new ones.

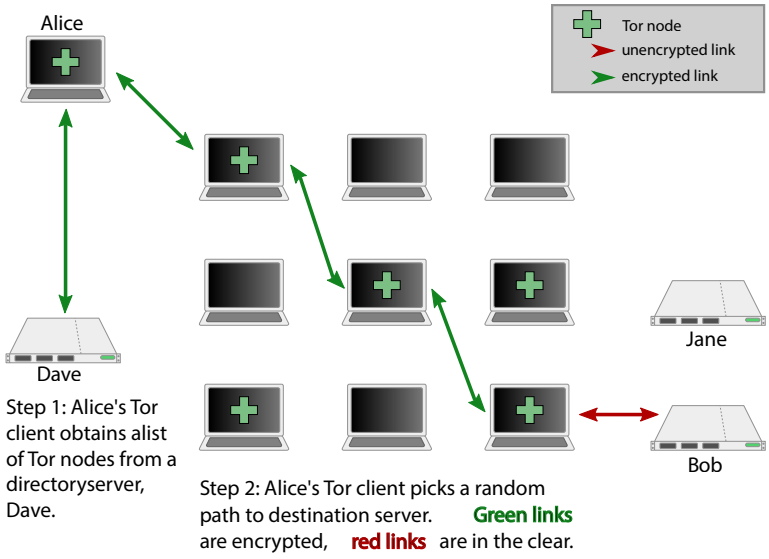

Figure 1: Overview of Tor
From Justin Findlay (2013) [16]

Tor also makes it possible for users to hide their locations while offering various kinds of services, such as web publishing or an instant messaging server.

Landscape of attacks on Tor

Tor focuses only on protecting the transport of data, so basically prevents an outsider from knowing who is talking to who. Tor can not provide full

privacy. When a subject uses software that sends messages containing identification without encrypting them end to end, it still reveals information about the subject.

Tor has recently released version 0.2.4 in which 256-bit ECC is the standard as opposed to RSA-1024 in older versions. The reason for this is that RSA-1024 is not considered secure anymore. However, not all Tor clients are updated to the newest version which means these old Tor clients may be vulnerable.

Other attacks on Tor exist too. For example traffic/timing analysis attacks [14, 12], which are used to correlate traffic between two known endpoints. Bad apple attacks [5] are another example, these make use of insecure applications that reveal user information such as the public IP address and of the fact that Tor puts all streams of a user on the same circuit. Finally an adversary can correlate a $(m/n)^2$ of the traffic if he controls n nodes [14].

3 National Security Agency

The National Security Agency (NSA) is the central producer and manager of signals intelligence for the United States. It is estimated to be one of the largest of U.S. intelligence organizations in terms of personnel and budget. There is a lot of speculation about what the NSA can do and what it has done.

Thanks to Edward Snowden we do no longer need to base our information on rumors about the NSA. Edward Snowden has so far leaked approximately 50 to 200 thousand secret documents [26] from the NSA and there are many more to come. These documents describe numerous spying activities done by the NSA. A few examples of these spying activities are:

- Get access to lots of data by influencing standards [40, 29].
- Receive copies of all internet traffic of companies that help the NSA, for example AT&T [15].
- Use mass surveillance programs to monitor internet traffic [22].
- Spy on friendly countries such as France, Britain and Mexico [24, 27].
- Eavesdrop on 35 world leaders [3].
- Access databases of Google and Yahoo via PRISM [18].

These are only a few of the spying activities of the NSA that have been revealed so far. Many of these activities violate European privacy rights [10].

Whistle-blower Edward Snowden woke us up and made us realize that the NSA is targeting citizens and friendlies, which means that the NSA is an adversary that we should be beware of.

Another reason why the NSA should be seen as an adversary of Tor, is because the NSA sees Tor as a high priority target. This becomes clear from top-secret slides of the NSA that Edward Snowden has leaked. These slides describe how the NSA plans to attack Tor [25].

NSA's black budget

The NSA should not be considered just an adversary, but a very powerful adversary thanks to their huge black budget. Their black budget for the year 2013 is estimated at 10.8 billion US dollars [31]. A black budget is a budget that is allocated for classified and other secret operations of a nation. Approximately 5.6 billion of this total amount goes to data collection, data processing and exploitation and data analysis. Also an estimated total of one billion US dollars is invested into cryptanalysis and exploitation.

Another major expenditure of the NSA is the building of a large data center which finished in 2013 [4]. This data center is called "Utah Data Center" but is also known as "Intelligence Community Comprehensive National Cybersecurity Initiative Data Center". This data center is around one million square feet big and should give the NSA storage capacity between three and twelve exabytes.

The precise goal of this data center is of course unknown. Probably it is used for storing and processing massive amounts of data. Thanks to Edward Snowden we have a general idea what kind of data this is.

4 Tor vs the NSA

In this section we will look at several attacks on Tor that are described in the literature. We also look at how the NSA can use these attacks to nullify the anonymization that Tor provides. Finally we will analyze what ciphers Tor uses and how strong these ciphers are currently considered.

4.1 Attacks on Tor

Tor has some unresolved weaknesses [14]. We will summarize the most pressing problems in the following sub sections.

4.1.1 Traffic analysis attack if the two endpoints are known

If an adversary is watching both the endpoints of a Tor path, then he can learn whether two parties are talking with each other. This means that if an adversary is watching Alice (or the first hop in the path) and Bob (or the last hop in the path) then the adversary can learn whether Alice and Bob are communicating, by making use of statistics. An adversary could for example simply count the packets that Alice sends into the Tor network and how many packets Bob receives from the Tor network. An adversary could also keep up the time when Alice sends a message and when Bob receives it (also called a timing analysis attack). After some time the adversary will have a good idea of whether Alice and Bob are communicating [14, 12]. Good places to start monitoring traffic are internet exchanges, as a lot of Tor circuits use one [32].

If the adversary actually controls the entry and the exit node, then he could launch a tagging attack. This basically means that he tags the data that comes into the Tor network and tries to find the tag once it comes at the exit node that he also controls [14]. An example of a tagging attack that has been tested is on the protocol level where Tor uses AES in

counter mode. The adversary could alter a message in such a way that the normal counter is disrupted. Once the message arrives at the exit node, this exit node will have problems decrypting as the counter is not correct anymore. The adversary will notice this if he controls the exit node [17].

Tagging attacks are however riskier for the adversary, as there is a higher chance that the adversary is noticed. This is because he makes actual changes in the data stream, where passive traffic analysis does not. Additionally, tagging attacks are expected to be less effective as for example timing attacks. If a timing attack results in suspecting that two persons are talking with each other, then this is often the case. In other words, time attacks have almost no false positives [35].

The NSA can use this attack, but they will have to do it manually, assuming that the NSA does not control any Tor nodes. They need to already suspect that two persons are talking with each other and this attack will then verify whether they actually are.

4.1.2 Bad apple attack

If a user wants to do something anonymously on the internet, then a user can do this by using Tor. However, if the user is running an insecure application while using Tor, then his session can be compromised. This is achieved by exploiting Tor's design with the so called bad apple attack [5]. With this attack one could learn who the person that is using the insecure application is.

The attack consists of two steps:

1. Exploiting an insecure application on user's machine to reveal the source IP address, or trace of a Tor user. BitTorrent is such an insecure application that can be used to start this attack. A Tor user that uses BitTorrent can use his public IP/port to establish P2P connections for efficiency, hereby sending his public IP/port in the clear.
2. Exploiting Tor to associate the usage of a secure application with the IP address of a user (revealed by the insecure application). Once an adversary knows the IP address of a BitTorrent user, it is easy to reveal the IP address of streams in the same circuit. As Tor multiplexes all traffic from one user in one circuit, an adversary can associate all these streams with the same IP address.

   Revealing IP addresses of streams in different circuits is a bit harder, but can also be done. BitTorrent creates a peer identifier, which is unique for every user. If an adversary finds this peer identifier in a different circuit, then he knows that this circuits has the same source. A problem arises when communication between peers is encrypted. In this case the adversary can link two circuits if a peer initiates communication with an IP/port contained in the tracker response of a different circuit [5].

Tor is not responsible for the first part of the attack as Tor does not provide protection against application-level attacks. The second part is however an attack against Tor.

Using this method the researchers of this attack were able to reveal 10.000

IP addresses within 23 days while having control of only six exit nodes. This means that the NSA, with their huge budget, is able to identify the IP address of almost all BitTorrent Tor users, as they can make a lot of Tor exit nodes.

4.1.3 Amount of nodes per circuit and profiling

An adversary controlling m out of n nodes can correlate at most $(m/n)^2$ of the traffic. An adversary can however attract more traffic to one of his m nodes and thereby correlate a higher amount of traffic [14]. Tor uses entry guards to counter this [41], this means that a Tor client has a small list of relay nodes that the client will use as entry nodes. If the adversary does not control any of these entry nodes, then it is not possible to correlate any of the user's traffic. If the adversary does have control of one of the entry nodes, he can correlate as much traffic as the setup without the entry guards. So if the adversary has control of one of the nodes, the user is still vulnerable to profiling, but because the user uses only a few nodes as entry nodes, he will now have a $((n-c)/n)$ chance to avoid profiling, where c is the number of relays the adversary can control and or observe.

A way to decrease the amount of profiling an adversary can do is to increase the amount of nodes per circuit. If n in $(m/n)^2$ becomes larger, you will need a higher m as well to have the same amount of traffic correlation. Increasing the size of the circuits would however also decrease the performance of the Tor network and can thus result into less users using Tor. Less users would result in less traffic and in order to have strong anonymity in an anonymity system, you need a lot of traffic [2].

It is not only a trade-off between performance and anonymity, but also a trade-off between anonymity by high amounts of traffic and anonymity by a longer path. The trade-off depends a lot on how many Tor nodes are expected to be compromised. This does not include the total amount of nodes that are compromised, but the amount of nodes that are being controlled by a certain company or group that work together to correlate traffic on the Tor network.

Another interesting fact about Tor is that it allows users to choose which entry and exit nodes they trust. So, if a user is aware of an adversary, then he can choose some entry and exit nodes of which he is sure that they are not compromised.

The NSA can use this attack quite well. They will need to obtain a lot of Tor nodes, to make sure that they often control two or more nodes in the circuit. This will require money, but that is something the NSA has. However, controlling so many nodes will raise suspicions.

4.2 Cryptography

4.2.1 Cryptography used in Tor

The source code of Tor is open source, making it possible to analysis the cryptographic functions used in Tor. In table 1 a summary of all cryptographic function used in the source code is given (accessed on 13 November 2013).

crypto.c	**aes.c**
DH key	EVP when possible
RSA	AES CTR (EVP_aes_ctr128)
DER (enc/decode)	
ASN.1 (enc/decode)	**General**
PKCS1 padding (signature)	TLSv2
SHA1 (signature)	SSL3
SHA256	
HMAC-SHA-256	
base32 (enc/decode) (RFC 4648)	
base64 (enc/decode)	
RFC2440 iterated-salted S2K	
curve25519	
3DES (unused)	

Table 1: Cryptographic function referenced in the source code

The protocol

In this section we will analyze when the above named cryptographic schemes are used in the Tor protocol [13].

Before we go deeper into the protocol, it is important to note that each Tor node has three keys: a connection key which is used to set up TLS connections, an onion key which is used to encrypt or decrypt a layer of encryption and an identity key which is used to sign documents and certificates.

The first part of the protocol is setting up a connection between two nodes or between a node and a client. All implementations of Tor must support SSLv3 and should support TLSv2. To set up a connection a TLS handshake is used, during which the two sides agree upon what cipher suite they will use. Once both sides have agreed upon the cipher suite, the authentication part is started for which both sides use SHA256 and SHA256-HMAC.

The second part of the protocol is creating a circuit. During this part SHA-1 is used over the identity key of a node to ensure integrity. Currently there are two handshakes available. Ntor is the newest handshake and is used by default in the newest version of Tor. TAP is used by default in all older versions of Tor. Ntor uses 256-bit ECC and TAP uses RSA-1024 for exchanging keys.

The third part of the protocol is to create session keys based on the key that was exchanged in the second part. Once both clients have created their session keys, they can communicate securely by making use of AES-128. When the circuit has lived long enough, which is ten minutes in the current version of Tor, it will be torn down.

4.2.2 The security of the ciphers used in Tor

AES

The Advanced Encryption Standard [11] is, like the name implies, the standard for encryption since National Institute of Standards and Technology (NIST) announced so in 2001. AES runs with three different key sizes, namely: 128-bit, 192-bit and 256-bit.

Many researchers tried to break AES but none succeeded so far with any noticeable practical attack. So we can assume the only way to decrypt data is by brute forcing. Table 2 shows an approximation on how many possible keys there are for a given key size.

Key size	Number of keys	Avg. keys before hit
128-bit (AES)	$\approx 3.4 \cdot 10^{38}$	$\approx 1.7 \cdot 10^{38}$
192-bit (AES)	$\approx 6.2 \cdot 10^{57}$	$\approx 3.1 \cdot 10^{57}$
256-bit (AES)	$\approx 1.1 \cdot 10^{77}$	$\approx 5.7 \cdot 10^{76}$

Table 2: Number of keys per AES key size

The table illustrates that brute forcing an AES encryption with key size of 128 bits will take an average of 170 undecillion ($1.7 \cdot 10^{38}$) attempts. A very fast GPU implementation for AES takes per byte about 0.17 cycles for encryption and 0.19 cycles for decryption [7]. The world's fastest supercomputer, created by a company named NUDT, has a theoretical top speed of 54,902.4 TFlop/s [34]. Brute forcing a 16 byte block with 128-bit key size on this supercomputer takes around $\frac{0.19 \cdot 16 \cdot 1.7 \cdot 10^{38}}{5.49 \cdot 10^{16}} = 9.41 \cdot 10^{21}$ seconds or $2.984 \cdot 10^{14}$ years. Brute forcing 16 bytes with an 128-bit key size seems thus impossible.

If we assume that the NSA can do the decryption twice as fast as the fastest AES implementation, it still takes $2.984 \cdot 10^{14}$ years to decrypt a single 16 byte block with AES-128.

However there is a lot of work on quantum computing. L. K. Grover shows that with a quantum computer he could search a n sized list in $\sqrt{n}$ steps [23]. Assuming that there is a way to achieve this on the fastest GPU AES implementation it would mean that AES-128 is broken and AES-192 is in the danger zone. Table 3 shows an approximation on how much time it takes to brute force AES for each key size assuming we have a fully working quantum computer.

Key size	Math	Number of seconds
128-bit	$\frac{\sqrt{0.19 \cdot 16 \cdot 1.7 \cdot 10^{38}}}{5.49 \cdot 10^{16}}$	≈ 414
192-bit	$\frac{\sqrt{0.19 \cdot 16 \cdot 6.2 \cdot 10^{57}}}{5.49 \cdot 10^{16}}$	$\approx 2.5 \cdot 10^{12}$
256-bit	$\frac{\sqrt{0.19 \cdot 16 \cdot 1.1 \cdot 10^{77}}}{5.49 \cdot 10^{16}}$	$\approx 1.0 \cdot 10^{22}$

Table 3: Time to brute force AES assuming quantum computer

Concluding, we can say that it is near impossible for any one without a quantum computer to brute force any AES key size. Tor uses AES in 128

bit CTR mode so if one has a quantum computer they could hypothetical brute force the data in very little time. However there are no signs of the existence of a fully operational quantum computer.

RSA

RSA is a public-key crypto system which is widely used around the world. The smallest key for a secure setup is 1024 bits, although rumors indicate that this may no longer be the case. We will take a better look at how much security RSA-1024 really offers.

Brute forcing RSA-1024 requires approximately $1.88 \cdot 10^{302}$ $((2.76 \cdot 10^{151})/2 - 2.76 \cdot 10^{151})$ pair of two distinct primes, as there are $2.76 \cdot 10^{151}$ possible primes when using RSA-1024. The world's fastest supercomputer has a theoretical top speed of 54,902.4 TFlop/s [34]. It would take this super computer $1.88 \cdot 10^{302} / 5.49 \cdot 10^{16} = 3.42 \cdot 10^{285}$ seconds or $1.084 \cdot 10^{278}$ years, assuming that every float operation can do one modulus. Brute forcing RSA-1024 thus seems impossible

There are however other methods that are way more efficient than simply trying out all prime combinations. For example the General Number Field Sieve (GNFS) technology is one of these methods and is the fastest known method to compute discrete logarithms. Since GNFS there have been no other major breakthroughs if it comes to factoring numbers. Gordon shows that computing discrete logarithms can be done in $L_p[1/3; 3^{2/3}]$ conjectured running time [21]. An improvement has been proposed by Shirokauer [38], which runs in $L_p[1/3; (64/9)^{1/3}]$. Here $L(a, c)$ is used to describe the complexity of algorithms and is defined as: $L(a, c) = e^{(c+o(1))(\ln(n))^a (\ln(\ln(n)))^{1-a}}$ where c is a positive constant and a is a number between zero and one.

The hardest part of the GNFS method, is the sieving step. A hardware implementation has been proposed, which significantly increases speed. The device is called The Weizmann Institute Relation Locator (TWIRL) and uses dedicated hardware to achieve a very high level of parallelism. This machine (as far as publicly known) is fictional. If the NSA would create a dedicated hardware implementation, then they could use TWIRL as a starting point. A TWIRL device can have 79 independent sieving devices on a 30cm single silicon wafer. It is expected that a TWIRL device can calculate RSA-1024 keys if it runs for about a year. A TWIRL device would cost approximately ten million dollars [39].

In 2007 National Institute of Standards and Technology (NIST) published RSA key-length recommendation that advise the phasing out of 1024-bit RSA keys by the end of 2010 [28].

Concluding, we can say that it is near impossible for the average hobbyist to break RSA-1024. However, if a large specialized organization with plenty of money, i.e. the NSA, wants to break RSA-1024, then they could theoretically break it in at most a year time and maybe even faster. Also because of the advise to stop using RSA-1024 we can conclude that RSA-1024 should not be used anymore.

ECC

Elliptic curve cryptography is like RSA a public-key crypto system. The advantage of ECC over RSA is that it can reach the same level of security, with smaller keys. The largest key length for ECC that has been brute forced, is 109 bit in 2002 [9]. We will take a closer look at how secure 256-bit ECC is, as this is the key length Tor uses with ECC.

The most efficient way to brute force ECC is Pollard's Rho method [6]. This method has various advantages of which full parallelization is the most important one. Thus in contrast with the GNFS method for brute forcing RSA, this method can have any computer contribute to the attack. The attack in 2002 had 10^4 computers for the attack, the attack took 549 days. The currently fastest implementation of Pollard Rho on elliptic curves is focused on 131-bit ECC. It is expected to have an overall run time of 10^5 years on an AMD phenom 9500 quad-core 64-bit processor with a clock frequency of 2.2GHz.

Using this information we can extrapolate to find the run time of this implementation on 256-bit ECC. The difficulty of the attack will be increased by $(\frac{256}{131} \cdot 2^{\frac{256-131}{2}})$. Multiplying this by the amount of years that would be necessary for ECC-131 will give us the amount of years necessary for 256-bit ECC. The result is approximately $1.2745 \cdot 10^2 3$ years or $1.2745 \cdot 10^1 4$ billion computers with above said processor, running for approximately one year.

NIST has proposed a Dual Elliptic Curve Deterministic Random Bit Generator (Dual_EC_DRBG) which can be used by the ECC to generate random numbers. Dual_EC_DRBG has a way higher efficiency than other known random number generators [8]. As with many crypto systems, ECC is only secure if the random number generator provides enough randomness.

The NSA has, however, deliberately influenced standards or created weakened standards themselves [33]. They did this with Dual_EC_DRBG [36] and this was not an exception. In this case it is for example possible to create a distinguisher that will give an adversary an advantage on finding the "random" numbers used [19].

Concluding we can say that brute-forcing 256-bit ECC is not an option. This would mean that a correct implementation of 256-bit ECC is secure.

5 Conclusion

Thanks to the disclosure of Edward Snowden, we now have a better idea of what the NSA is up to. We know that they want to collect as much information as possible and that they have 10.6 billion dollar to realise this goal.

We have seen that several attacks on Tor exist and that they are able to nullify the anonymization Tor provides. These attacks however are only possible, if two targeted nodes from a path are owned by the NSA or if a large part of the Tor network is in hands of the NSA.

Although RSA-1024 is not considered insecure, there are reasons to not use it anymore. 256-bit ECC and AES-128 can still be considered secure. The uncertainty is however about the NSA. Slowly we are getting more information about what the NSA can do, but we do not know whether they are more technologically advanced than the academic world. Neither do we know how many Tor nodes the NSA controls.

As a final conclusion we would like to say that Tor will provide protection against the NSA if it comes to mass surveillance, assuming that the NSA are as technologically advanced as the academic world. If the NSA however targets a person specifically then Tor alone will not provide enough protection.

References

[1] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel. FPDetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, CCS '13, pages 1129–1140. ACM, November 2013. http://www.cosic.esat.kuleuven.be/publications/article-2334.pdf.

[2] A. Acquisti, R. Dingledine, and P. Syverson. On the Economics of Anonymity. In *Financial Cryptography*, volume 2742 of *Lecture Notes in Computer Science*, pages 84–102. Springer, 2003. http://freehaven.net/doc/fc03/econymics.pdf.

[3] J. Ball. NSA monitored calls of 35 world leaders after US official handed over contacts, Oktober 2013. http://www.theguardian.com/world/2013/oct/24/nsa-surveillance-world-leaders-calls.

[4] J. Bamford. The NSA Is Building the Country's Biggest Spy Center (Watch What You Say), March 2012. http://www.wired.com/threatlevel/2012/03/ff_nsadatacenter/all/1.

[5] S. L. Blond, P. Manils, A. Chaabane, M. A. Kaafar, C. Castelluccia, A. Legout, and W. Dabbous. One Bad Apple Spoils the Bunch: Exploiting P2P Applications to Trace and Profile Tor Users. *USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 4, March 2011. https://www.usenix.org/legacy/events/leet11/tech/full_papers/LeBlond.pdf.

[6] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery. On the security of 1024-bit rsa and 160-bit elliptic curve cryptography. *IACR Cryptology ePrint Archive*, 2009. http://lacal.epfl.ch/files/content/sites/lacal/files/papers/ecdl2.pdf.

[7] J. W. Bos, D. A. Osvik, and D. Stefan. Fast implementations of aes on various platforms. *IACR Cryptology ePrint Archive*, 2009. http://eprint.iacr.org/2009/501.

[8] D. R. L. Brown. Conjectured security of the ansi-nist elliptic curve rng. *IACR Cryptology ePrint Archive*, 2006. http://www.math.ntnu.no/~kristiag/drafts/dual-ec-drbg-comments.pdf.

[9] Certicom. Certicom Announces Elliptic Curve Cryptosystem (ECC) Challenge Winner, 2002. http://www.certicom.com/2002-press-releases/340-notre-dame-mathematician-solves-eccp-109-encryption-key-problem-issued-in-1997.

[10] Council of Europe. European Convention on Human Rights. November 1950. http://echr-online.com/art-8-echr/introduction.

[11] J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002. http://www.springer.com/computer/security+and+cryptology/book/978-3-540-42580-9.

[12] G. Danezis. The Traffic Analysis of Continuous-Time Mixes. In *Proceedings of the 4th International Conference on Privacy Enhancing Technologies*, PET '04, pages 35–50. Springer, 2004. http://freehaven.net/anonbib/cache/danezis:pet2004.pdf.

[13] R. Dingledine and N. Mathewson. Tor protocol specification. November 2013. https://gitweb.torproject.org/torspec.git/blob_plain/ac3824c45dde87849e531e75248e7495e0047028:/tor-spec.txt.

[14] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. 2004. http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA465464.

[15] EFF. NSA Spying on Americans, 2013. https://www.eff.org/nsa-spying.

[16] J. Findlay. Tor website PNG diagrams as SVG. November 2013. https://lists.torproject.org/pipermail/tor-dev/2013-November/005762.html.

[17] X. Fu and Z. Ling. One Cell is Enough to Break Tor's Anonymity. *Black Hat DC*, 2009. https://www.blackhat.com/presentations/bh-dc-09/Fu/BlackHat-DC-09-Fu-Break-Tors-Anonymity.pdf.

[18] B. Gellman and A. Soltani. NSA infiltrates links to Yahoo, Google data centers worldwide, Snowden documents say, October 2013. http://www.washingtonpost.com/world/national-security/nsa-infiltrates-links-to-yahoo-google-data-centers-worldwide-snowden-documents-say/2013/10/30/e51d661e-4166-11e3-8b74-d89d714ca4dd_story.html.

[19] K. Gjøsteen. Comments on dual-ec-drbg/nist sp 800-90 draft december 2005. March 2006. http://www.math.ntnu.no/~kristiag/drafts/dual-ec-drbg-comments.pdf.

[20] D. Goldschlag, M. Reed, and P. Syverson. Onion Routing for Anonymous and Private Internet Connections. *Communications of the ACM*, 42(2):39–41, February 1999. http://www.onion-router.net/Publications.html#CACM-1999.

[21] D. M. Gordon. Discrete Logarithms in $GF(P)$ Using the Number Field Sieve. *SIAM Journal*, 6:124—-138, 1992. http://www.ccrwest.org/gordon/log.pdf.

[22] G. Greenwald. XKeyscore: NSA tool collects 'nearly everything a user does on the internet'. July 2013. http://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data.

[23] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996. http://arxiv.org/abs/quant-ph/9605043.

[24] Guardian. British spy agency taps cables, shares with nsa -. June 2013. http://www.reuters.com/article/2013/06/21/usa-security-britain-idUSL5N0EX39I20130621http://www.reuters.com/article/2013/06/21/usa-security-britain-idUSL5N0EX39I20130621.

[25] T. Guardian. 'tor stinks' presentation. October 2013. http://www.theguardian.com/world/interactive/2013/oct/04/tor-stinks-nsa-presentation-document.

[26] M. Hosenball. NSA chief says Snowden leaked up to 200,000 secret documents, November 2013. http://www.reuters.com/article/2013/11/14/us-usa-security-nsa-idUSBRE9AD19B20131114.

[27] M. R. Jens Glüsing, Laura Poitras and H. Stark. Fresh Leak on US Spying: NSA Accessed Mexican President's Email, October 2013. http://www.spiegel.de/international/world/nsa-hack-email-account-of-mexican-president-a-928817.html.

[28] A. Juels. RSA-768 Factored, January 2010. https://blogs.rsa.com/rsa-768-factored/.

[29] J. Larson and J. Elliott. Government standards agency "strongly" discourages use of NSA-influenced algorithm, September 2013. http://arstechnica.com/security/2013/09/government-standards-agency-strongly-suggests-dropping-its-own-encryption-standard/.

[30] Lunar. Tor Weekly News, September 2013. https://blog.torproject.org/blog/tor-weekly-news-%E2%80%94-september-11th-2013.

[31] D. Matthews. America's secret intelligence budget, in 11 (nay, 13) charts. August 2013. http://www.washingtonpost.com/blogs/wonkblog/wp/2013/08/29/your-cheat-sheet-to-americas-secret-intelligence-budget/.

[32] S. J. Murdoch and P. Zieliński. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies*, PET '07, pages 167–183. Springer, June 2007. http://www.cl.cam.ac.uk/~sjm217/papers/pet07ixanalysis.pdf.

[33] J. L. Nicole Perlroth and S. Shane. Secret Documents Reveal N.S.A. Campaign Against Encryption, September 2013. http://www.nytimes.com/interactive/2013/09/05/us/documents-reveal-nsa-campaign-against-encryption.html?_r=0.

[34] NUDT. Tianhe-2 (MilkyWay-2), November 2013. http://www.top500.org/system/177999.

[35] L. Øverlier and P. Syverson. Locating Hidden Servers. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, SP '06, pages 100–114. IEEE Computer Society, May 2006. http://www.onion-router.net/Publications/locating-hidden-servers.pdf.

[36] N. Perlroth. Government Announces Steps to Restore Confidence on Encryption Standards, September 2013. www.nytimes.com/interactive/2013/09/05/us/documents-reveal-nsa-campaign-against-encryption.html?_r=0.

[37] J. Rachels. Why Privacy is Important. *Philosophy & Public Affairs*, 4(4):323–333, 1975. http://www.jstor.org/stable/2265077.

[38] O. Schirokauer. Discrete Logarithms and Local Units. *Theory and Applications of Numbers without Large Prime Factors*, 345(1676):409—423, 1993. http://www.jstor.org/stable/54275.

[39] A. Shamir and E. Tromer. On the Cost of Factoring RSA-1024. 2729:1–26, 2003. http://link.springer.com/chapter/10.1007%2F978-3-540-45146-4_1.

[40] A. Solani. NSA influences standards, September 2013. https://twitter.com/ashk4n/statuses/375725741830598657.

[41] Tor. Tor FAQ: Entry guards. November 2013. https://www.torproject.org/docs/faq.html.en#EntryGuards.

[42] Tor. Tor metrics portal: Users. December 2013. https://metrics.torproject.org/users.html.

[43] Tor. Tor: Sponsors. 2013. https://www.torproject.org/about/sponsors.html.en.

[44] U.S. Navel Research Laboratory. Onion Routing. March 1997. http://www.onion-router.net/Archives/onions-1997.txt.

[45] S. D. Warren and L. D. Brandeis. The Right to Privacy. *Harvard Law Review*, 4(5):193–220, December 1890. http://www.jstor.org/stable/1321160.