Is the national prohibition of strong encryption feasible, and would it prove effective in the fight against crime?

Awn Umar

April 28, 2017

Contents

Abstract

This paper will attempt to highlight the feasibility, effectiveness, and consequences of a ban on strong encryption. It will go over notable instances in history (recent or otherwise) of attempted regulation and their respective successes. It will also explore a hypothetical modern world where strong encryption is prohibited, and the measures a criminal could take in such a world. By doing this, we hope to make the point that if encryption is outlawed, only the outlaws will have encryption.

1 Introduction

Cryptography is a silent guardian for most people. We don't think about it much in our everyday lives—I would say that many of us don't even know what it is—but without it, many of the things we take for granted online would not exist[1]. However, there are a number of parties that seek to either regulate or prohibit the use of strong cryptography, or that have something to gain from that outcome.

The main, and most vocal, opponents of encryption are governments. The UK and US have repeatedly come out against it, stating that it provides "safe-havens" for criminals and terrorists[2]. After the former British Prime Minister David Cameron promised to ban strong encryption, there was public outcry. This prompted the new Prime Minister, Theresa May, to attempt clarification: "We believe encryption is important. We are not proposing to make any changes to encryption and the legal position around that." However, she went on to say, "Where we are lawfully serving a warrant on a provider [...] they are required to provide certain information to the authorities. The company should take reasonable steps to ensure they are able to comply with the warrant."

Theresa May's bill, the "Snooper's Charter"[3]—which has now passed into law—contains the following clause: "CSPs subject to a technical capacity notice must notify the Government of new products and services in advance of their launch, in order to allow consideration of whether it is necessary and proportionate to require the CSP to provide a technical capability on the new service." The UK government now has the power to demand companies to install backdoors in their services—only time will tell whether they aren't afraid to use it.

Governments are not the only ones with something to gain from citizens having inadequate protection. The people that would potentially benefit the most are criminals: the very people the authorities are trying to catch. Without the protection of cryptography, our personal information, banking details, and darkest secrets are open to theft. It's a somewhat ironic thing to note.

2 Historical Successes of Cryptography

2.1 Crypto Wars

Perhaps the most known examples of counter-cryptography occurred as part of the so-called *crypo-wars*[4]. The term is an unofficial name for the U.S. and allied governments' attempts to limit the public's and foreign nations' access to cryptography strong enough to resist decryption by national intelligence agencies[5].

2.1.1 Export of Cryptography

During the Cold War, the U.S. and its allies pushed into law a series of export controls mandating that any technologies classed as *critical* required a license[6][7]. In the immediate post-WWII period the market for cryptography was almost entirely military, so encryption technology was also included as a Category XIII (*Materials and Miscellaneous Articles*) item into the United States Munitions List[8]. This heavy regulation was perhaps a hint as to the fact that governments were scared of cryptography and of what it could do.

In 1991, Philip Zimmermann wrote the most widely used email encryption software in the world: PGP (Pretty Good Privacy). It allowed ordinary people to easily encrypt sensitive information and communications such that even powerful nation-states could not gain access. But the notable thing was not simply the fact that he had written it, but rather that he made the source-code open and available to the world.

This was momentous. Despite attempts to stop it from spreading, the software very rapidly acquired a considerable following around the world, attracting dissidents in totalitarian countries, civil libertarians in other parts of the world, and cypherpunks[9].

For some insight into why, we can look to the Streisand effect[10]. It is a phenomenon whereby an attempt to hide, remove, or censor a piece of information has the unintended consequence of publicising the information more widely. We live in the age of the internet, the communication of data is almost too trivial, and what data is more at home on the internet than what built it in the first place: code.

There were a few more regulations put into place later—notably *The Wassenaar Arrangement*[11]—but in 2000, the Department of Commerce (perhaps seeing the futility of the restrictions) relaxed the export-control regulations specified in *The Code of Federal Regulations* somewhat[12]—thereby simplifying the export of commercial and open-source cryptographic software.

Today, export controls do exist but are much more relaxed. The general public has access to strong encryption, and there haven't been any US prosecutions between 2009 and 2012 for the export of encryption software alone[13].

2.1.2 Clipper Chip

The Clipper chip was a chip-set that was developed and promoted by the NSA as an encryption device with a built-in backdoor, intended to be adopted by telecommunications companies for voice transmission[14]. It was based on the concept of key-escrow; although the Electronic Frontier Foundation preferred the term "key-surrender" to stress what, according to them, was actually happening.

In the factory, any device with a Clipper chip would be given a cryptographic key which would be provided to the government *in escrow*. If government agencies felt the need to listen to a communication, the key would be provided to them and they could subsequently decrypt all data transmitted by that particular device.

Many people saw this as gross violation of privacy and there was considerable backlash from the Electronic Privacy Information Center and the EFF, amongst others. They claimed that it would subject citizens to increased and potentially illegal government surveillance. Another concern was that the strength of the Clipper chip's encryption could not be evaluated by the public (as its design was classified as *Secret*) and that therefore individuals and businesses could be stuck with an insecure communication system.

This triggered the creation of several strong cryptographic software packages such as Nautilus, PGP, and PGPfone. The logic was that if strong cryptography was freely available to the general public, the government would be unable to prevent its use.

However, what finally killed off the Clipper chip was not alternative cryptographic suites, but rather a demonstration of how broken it was. One attack, [15], allowed the Clipper chip to be used as an encryption device without the key escrow capability, and another, [16], bypassed the escrow system in real time.

The entire project is now scrapped and defunct. It is important to note, however, that the predominant forms of voice communication today are not strongly encrypted. So even though the cypherpunks won this battle, who exactly is winning the war?

2.1.3 Encryption in Smartphones

Smartphones are amongst our most used computing devices, but also the ones that are most targeted. It is not a surprise that this is the case: terrorists use mobile phones to spread news and propaganda, drug dealers use them to communicate, and paedophiles use them to store illegal materials; they are an appealing target.

On the other hand, journalists, whistle-blowers, activists, and even ordinary people like you and I—we all benefit from this technology. It seems as though, like many things, they are a tool with no inherent good or evil, but the capacity for both. The question we should all ask ourselves is if we should allow the few to decide what the many should be permitted.

That is not to say that there are no defences in place. Post-Snowden, we have seen a significant increase[17] in the number of chat applications using end-to-end encryption. The former Director of National Intelligence in the US, James R. Clapper, said, "As a result of the Snowden revelations, the onset of commercial encryption has accelerated by seven years." This change is reflected in software too: in the last few years multiple (very) popular services have employed end-to-end encryption, essentially making it impossible for them to comply with requests from the authorities.

But why are companies intentionally designing their products so that they themselves cannot access them? The only probable answer is because consumers want it. That is the reason for any innovation, is it not? Demand.

This massive spike in privacy-protecting technologies has led to governments all over the world calling for a backdoor or a ban. They can no longer spy on their citizens with impunity, and this hurts them.

But governments do not necessarily need to force private-sector companies to install backdoors in their services, they can and will[18][19] attempt to insert their own backdoors (or even exploit existing vulnerabilities) covertly in commercial products. This, quite predictably, leads to massive fallout *when* something leaks[20].

There is also a constant trend of underestimating criminals. One thing needs to be understood: *people who want something to remain private will not use backdoored or proprietary technologies to do so.* If a backdoor is inserted into a consumer application, criminals will migrate to something that is not backdoored. If a new encryption standard with built-in key-escrow is introduced, criminals will ignore it and continue using strong encryption. There is no use in naively believing criminals will use what we want them to. All this approach does is harm consumers and gives us a false sense of security[21].

2.2 The Onion Router

For another example of how extremely difficult it is to block software, we should take a look at the onion router (or Tor for short).

Tor is a protocol that allows the user to conserve privacy, anonymity, and confidentiality, whilst browsing the web. Put *very* simply, it works by routing the client's connection through three separate nodes before it reaches its destination. The first "entry" node knows only the identity of the client and the address of the next "relay" node. The relay node knows only the address of the previous and next "exit" node, and the exit node knows only the address of the previous node and the destination.

The technology is used by countless groups of people[22]; from journalists to activists; law-enforcement to whistle-blowers; businessmen to bloggers; to normal people; and everything in between. All of these people derive huge value from Tor, but—inevitably—so do some criminals.

According to the Snowden leaks, the NSA—and other "spooks" such as GCHQ—invested a lot of resources into breaking Tor[23]; and failed[24]. The few attacks that they did come up with relied on end-point exploitation where

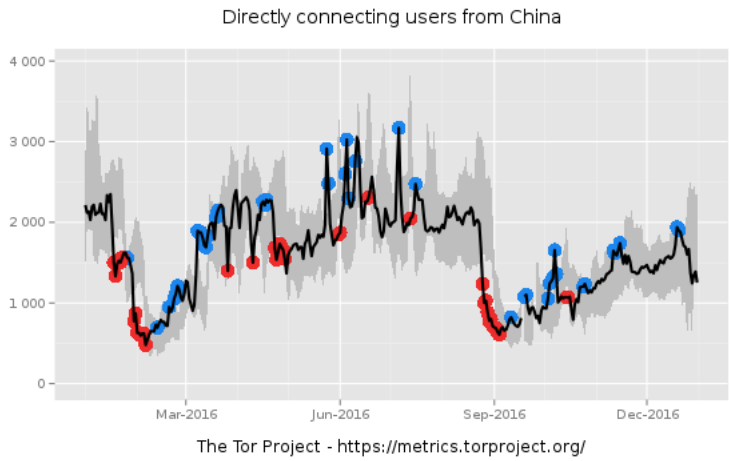

Figure 1: Tor users in China during 2016

they attack the user's system directly, bypassing Tor. These attacks are usually ephemeral unless they're very careful about their use. It's a Cornelian dilemma: if they use the exploits freely then someone will notice and fix the vulnerability, and if they don't use it freely then they don't catch as many people as they want to. Regardless, the protocol itself has mostly resisted exploitation, and where it hasn't, patches are quick.

One of these vulnerabilities that the NSA has access to led to a surprising turn of events. During March of 2017, in the case of *United States v. Jay Michaud*, the Department of Justice dismissed a case against a child-porn suspect in order to keep the details of a Tor vulnerability private[25]. In the motion to dismiss, [26], the federal prosecutor, Annette Hayes, says, "The government must now choose between disclosure of classified information and dismissal of its indictment. Disclosure is not currently an option."

What does this case say about the war our governments are waging against mathematics? It is a strange world that we live in where the value of some source-code trumps the prosecution of a *child-porn* suspect.

The other major party interested in Tor is China, and this is the interesting one because they not only try to actively probe and block Tor from working[27], but they also censor online services that allow you to download Tor. It's quite a parallel to what would be required if banning encryption.

Despite China's best efforts, fig. 1 shows us that there are plenty of users in China that manage to access the Tor network just fine. You can even see possible censorship attempts where the number of users drops dramatically, but each time, Tor fights back and the graph recovers.

Now, while Tor does make extensive use of encryption, it is not exactly the same. In the end it is just software: a collection of source-code that can be shared in any medium that allows it, and backed up by people all over the

world that fight for it. Encryption alone on the other hand is simply pure mathematics. If powerful agencies and nations are unable to beat Tor, which is magnitudes easier to censor than cryptography, then how do they expect to enforce a ban on a huge branch of computer science and mathematics?

It is like trying to ban an idea. It's completely infeasible.

3 The Hypothetical Authoritarian State

In this section, we will set the scene on a hypothetical modern world in an authoritarian state where strong encryption is prohibited. We will play the role of Alice: a criminal in such a place. Alice will attempt to encrypt a message for her friend, Bob, such that no one is able to access the plaintext.

3.1 Mission Objective

The objective is to encrypt the message, "Attack at dawn", with strong, modern encryption. The adversaries are the government and the authorities, who will be referred to collectively as Eve. Eve should not be able to access the secret plaintext at any time.

The encrypted message needs to be exchanged with Bob. Once the exchange is complete, the following must hold as true:

1. Both Alice and Bob must be confident that only the two of them know the contents of the secret message.
2. Bob must be confident that the message he is reading was sent by Alice in its entirety.

Keeping the fact that the communication took place a secret is out of scope in this mission, as it is in the real world.

3.2 Rules and Restrictions

Since this is an authoritarian state, there will be some restrictions. These measures were designed by Eve to make it impossible for an exchange of information to happen such that Eve does not have access to the information. The resources that you have access to are as follows:

1. A computer running Linux. Even if this was banned, having it anyway is not unlikely since it is relatively easy to smuggle contraband into and out of a country. Think about drugs, weapons, et al.
2. An internet connection through a standard internet service provider.
3. Access to a communication channel provided by Eve. This is considered to be the standard legal method of "secure" communication—representing a backdoored communication service in the real world. Eve, of course, can read and modify any data sent on this channel.

There are some restrictions on these resources:

1. Known online sources of encryption tools and libraries are blocked at the ISP level.

2. There is a ban on services that allow circumvention of censored content. This is enforced by blocking lists of known VPNs and proxies.

3. Big sites and services like Amazon, Twitter, and Google are left unblocked as blocking these would incite significant outrage by the general public.

3.3 The Execution

3.3.1 Preparation

Before we begin exchanging encryption-keys, we must acquire the software and tools required to do so.

Firstly, language choice. There are a multitude of different languages that we could choose for this task; capability and ease-of-use are the main deciding factors. For a nice balance, we will be using Go. Its syntax is relatively readable and easy to understand, and its standard library is very powerful.

To install Go normally would be a simple case of grabbing it from our package manager, but let's assume that this is blocked. We must therefore find a way to bypass these blocks. This is a lot easier than it sounds—any school student can teach you how to do it.

In our case, VPNs and proxies are not suitable since they are killed as soon as they're discovered so, even if we do find a working one, it's likely to be ephemeral. Instead we will use Tor—routing our connection through Google. This is brilliant because, to Eve, it will look like we are simply accessing an allowed service. Also, countries like China spend a lot of time and resources actively fingerprinting and blocking Tor, and yet still the citizens of said country are able to use it. Hence, this scenario is not implausible.

Now that we have circumvented Eve's blocks, let's grab and install Go:

```
$ pacman -S go
```

The easiest and most powerful encryption library is undoubtedly NaCl. With this, we can abstract away the intricate details of the cryptographic algorithms and instead treat them as "black-boxes". Installing NaCl is just as easy as installing Go:

```
$ go get golang.org/x/crypto/nacl/box
```

Now we are ready to begin.

You might not believe me when I say this was the hard part, but it's true—the most difficult step is getting your hands on encryption software in the first place. The rest is trivial.

In our case we managed to bypass ISP blocks with Tor but, if this were not viable, we could have easily smuggled the same software into the country on a physical drive. If that too failed, we could have sat outside with a Geiger counter and generated random values using background radiation, and then use them in a one-time-pad[28].

3.3.2 Key-exchange

There are generally two types of ciphers: symmetric and asymmetric. They each have their own benefits and drawbacks, but the basic idea is that symmetric ciphers use the same key for both encryption and decryption, while asymmetric ciphers use one key for encryption and another for decryption. Most cryptographic protocols in the real world use a combination of both—NaCl does the same, implementing a system that uses Curve25519XSalsa20Poly1305 to achieve confidentiality, integrity, and authenticity.

Generating the key-pair is not difficult:

```
publicKey, privateKey, _ := box.GenerateKey(rand.Reader)
```

One thing to keep in mind is that Eve can modify data on the channel. Consider the following:

1. Alice sends her public-key to Bob over the insecure channel.
2. Eve intercepts and saves it. She then forwards her own public-key to Bob instead of Alice's.
3. Bob sends back his own public-key and Eve repeats the same ritual.
4. Alice encrypts a message with the public-key she believes to be Bob's and sends the ciphertext through the channel.
5. Eve intercepts it, decrypts it, and reads it. She then re-encrypts it—possibly even modifying it first—with Bob's real public-key and forwards the new ciphertext to Bob.
6. Both Alice and Bob believe they have communicated securely when in actual fact Eve saw everything.

The solution to this is *fingerprinting*. After exchanging public-keys, if both Alice and Bob generate a unique value based on their own key and the key they believe to belong to the other party, they can compare this value to check if Eve tampered with the exchange.

To generate the fingerprint, Exclusive-OR (XOR) is computed with the two public-keys as the operands, and then that output is hashed. Symbolically this is identical to:

$$sha256(PK_{\text{Alice}} \oplus PK_{\text{Bob}})$$

Since XOR is commutative, the order of the keys is irrelevant.

The next step is to compare this value securely. There are a whole load of ways to do this; insert it into the Bitcoin block-chain; post it on Twitter; write it in the sky with jet planes, et al. But to keep things simple, we will have Alice and Bob simply meet in person and compare the value they computed. They could of course simply exchange the public-keys themselves in public and skip the fingerprinting step altogether; both are viable options.

3.3.3 Encryption and Communication

This is the easy part—which is not something Eve would want a criminal to say. Firstly Alice encrypts the message with Bob's public-key, using her own private-key to sign the message:

```
ciphertext := box.Seal([]byte(""),
                       []byte("Attack at dawn"),
                       &nonce,
                       bobPublicKey,
                       alicePrivateKey)
```

This yields the following ciphertext:

```
zfCWUMfm4gFvFbjwYGUpq1KHJrNO6KNI9v6VhpPk
```

Alice sends this to Bob through the insecure channel. Bob then decrypts it with his private-key, using Alice's public-key to verify that it came from her:

```
plaintext, _ := box.Open([]byte(""),
                         ciphertext,
                         &nonce,
                         alicePublicKey,
                         bobPrivateKey)
```

Giving the plaintext string:

```
Attack at dawn
```

And there we have it: in an authoritarian state where encryption is outlawed and heavily regulated, Alice and Bob were still able to communicate securely—and extremely easily—with Eve being none the wiser...

4 Logical Arguments

Previously we have concentrated on why an encryption ban is a bad thing.

In this section—building upon the points made in the previous sections—we will instead prove that it is impossible. The arguments are centred around the fact that there is no point of having a non-enforceable law; especially when the law's very existence has disastrous consequences for government, businesses, and normal citizens; and zero consequences for the people that law is targeting: criminals.

4.1 Successfully Banning Encryption is Impossible

This is a very short deductive argument on the impossibility of banning encryption. A deductive argument is very simple—a number of premises are setup, after which a conclusion is given. The truth of the premises *guarantees* the truth of the conclusion.

1. The only way to prevent an individual (read: criminal) from using strong encryption is to take away access to the tools required to implement strong encryption.
2. From section 3, it is not possible to take away the tools required to implement strong encryption.
3. Therefore it is not possible to prevent an individual from using strong encryption.

4.2 Banning Encryption Only Affects Civilians

This is a short inductive argument on why a ban on encryption will not affect criminals, but rather will only have catastrophic consequences for the rest of us.

An inductive argument differs from a deductive one in that the premises do not guarantee the conclusion, but instead strongly imply it.

1. From section 4.1, it is impossible to prevent an individual (read: criminal) from using strong encryption.
2. Overcoming an attempted ban on encryption is nontrivial and requires effort.
3. It is unlikely that ordinary civilians will invest a lot of time and effort into overcoming the ban.
4. Therefore ordinary civilians—and not the true targets of the ban—will be the only ones to be affected by the aforementioned ban.

5 Proposed Alternative Solutions

Thus far when we have discussed "banning" encryption, that has included banning it outright, only allowing backdoored encryption tools, or limiting bans to strong encryption. In this section we will go over and evaluate the merits, drawbacks, and effectiveness of alternatives to banning encryption, as in measures the authorities could take to try to improve the game against criminals while preserving the security of ordinary citizens.

5.1 Key Escrow

Key-escrow systems are discussed a lot, and to many people they seem like the best option. We have already touched upon a (failed) implementation of a key-escrow system in section 2.1.2, but in this section we will discuss them more generally.

There are some other arguments concerning the problems with implementing these kinds of systems, but for this section we will concentrate on the effectiveness instead. For more information on that, source [29] is a report by many prominent cryptographers and security experts on the inherent risks of key-escrow systems. It ends with, "Key recovery systems are inherently less secure, more costly, and more difficult to use [...] the breathtaking scale and complexity that would be required for such a scheme is beyond the experience and current competency of the field".

Key-escrow is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorised third party may gain access to those keys. In our context the third party is the government, and the "certain circumstances" are something like a court-order.

It should be noted that implementing this kind of system almost by definition prohibits certain cryptographic best-practices. For example, forward-secrecy is a feature that ensures that the breach of an encryption key for a certain message does not impact the security of subsequent communications. This would not be possible with a key-escrow system in place.

Another thing to consider is that there is one important prerequisite to being able to catch criminals with key-escrow systems: the target should be using the system that the government has access to. This is a lot harder to guarantee than you would expect. Of course for this to happen, he or she must not suspect that the encryption tool that is being used is giving up keys to the authorities.

Broadly speaking, there are two main types of encryption tools; those that handle stored data and those that handle communications traffic. In the following sections we will discuss the practicality of hiding the key-escrow systems' existence for both of these categories.

5.1.1 Stored Data

Stored data encompasses all data that is encrypted for secure storage. Examples are local file encryption, full disk encryption, et al. Encryption software is used to accomplish this, and this is the key point, it's all local.

Firstly, an important point to make is that the most highly trusted encryption utilities are all open-source and free for everyone to inspect. An adversary therefore has two options; try and secretly add some code to existing software and probably get caught immediately; or release their own closed-source encryption utility that no one would trust and so would never be used, at least by their targets.

Another property of local encryption is that the key-generation, encryption, everything; it all happens on the user's own system. Therefore, if the software

suddenly decides to send the encryption keys to a remote server, the user could easily detect it and block it, or simply use a different software.

5.1.2 Communications Traffic

Communications traffic entails any message or information that is sent to a remote party. Real world examples of this could include emails, phone calls, and text messages. When this data is encrypted, the key-escrow system would need to be able to get the key to where it is to be held in escrow.

This is miles easier than it is for locally stored data, in some cases. Some things are unaffected, for example properly encrypted emails are encrypted locally using PGP—and so are subject to the limitations discussed in section 5.1.1—but other services such as end-to-end encrypted chats like WhatsApp and Signal have a whole key-exchange system and networked security negotiations.

Normally this would be of no concern, but WhatsApp is closed-source and so are many other similar services. The same open-source argument comes into play: if you are unable to view and audit the security software yourself, then how can you trust it?

If the government asked a service provider to covertly implement a key-recovery system in their service, odds are that they would catch very few or even no criminals. That is because they would all be using something like Signal: which is fully open sourced. And if the government, again, attempted to add their own code to an existing open-source service, they would probably be caught immediately.

The fatal flaw with key-escrow systems, as with most of the things we have discussed, is that they naively assume that their targets will use the system. In reality, this just isn't true.

5.2 Commercial Backdoors

Commercial backdoors are backdoors in commercial products and services that would allow a third-party access to the data of its users. For example; Facebook allowing access to Messenger chat logs; Google allowing access to emails; or internet service providers allowing access to history logs[3].

Implementing legislation to force companies to add backdoors to their services is perhaps the easiest option for the government. If you think about it, all they would have to do is pass a bill and it would all work out, right? Well, there are a few problems that need to be solved first.

For a start, companies will be *very* hesitant to comply with such legislation. Post-Snowden, they have been trying to distance themselves from anything that would support government surveillance, and have been leaning towards privacy-conserving features. For example, as discussed in section 2.1.3, smartphones now have been designed so that even the manufacturer themselves cannot access the user's data that is stored on them.

If companies suddenly started turning back and being complicit in government-sponsored surveillance, it would likely damage their image and would weaken consumer trust in them. Hence, enforcing this legislation would be extremely difficult, if not impossible. The laws of a country do not extend beyond its borders, so companies are more likely to simply pull out instead of designing a custom device to appease the government, and that very few consumers are likely to buy anyways.

Considering other implications of forced commercial backdoors, there is a paper, [30], by more than a dozen prominent experts who state that requiring government access to all data and communications is "mandating insecurity". They also say, "The complexity of today's Internet environment, with millions of apps and globally connected services, means that new law enforcement requirements are likely to introduce unanticipated, hard to detect security flaws."

Backdoors are also seen as privacy invading. From a user's perspective, if their data is being logged for the government to look at, what's to stop the company from selling it to other parties? This problem is perhaps solved with more privacy-conserving legislation.

The biggest problem however cannot be solved with any amount of legislation: criminals will not use these backdoored services. Lord Paddick, Liberal Democrats spokesman, said regarding the Investigatory Powers Bill, "Now that we have a professional estimate that it would cost well over £1 billion in set-up costs alone and could be easily circumvented by criminals for just a few pounds a week, apart from anything else, this represents appallingly bad value for money."[31]

5.3 Backdoored Ciphers

Some people suggest that consumers should use backdoored ciphers, that is, encryption algorithms that have been designed to remain secure in all cases except for when the government wants access. This is a fairy tale, it is not possible. This kind of "solution" is usually sprouted by people that have a very limited understanding of cryptography. Encryption is either secure for everyone all of the time, or never secure for anyone. There is no middle ground.

The way that ciphers are designed is really quite interesting. An individual or group carefully design and propose an algorithm, expecting it to be quickly broken. If it survives the initial analysis by experts, then it is laid out to the world for years, while no one uses it and lots of very smart people try to break it and get recognition for doing so.

After maybe five to ten years of it remaining secure against sophisticated attacks, people start trusting it and using it. A great example of this process is the AES[32] and SHA3[33] competitions run by NIST (National Institute of Standards and Technology).

That is what it takes for someone to trust a cipher. It has to be open and withstand attack by the academic community for a very long time. Therefore, the *only* ciphers that a security-concerned individual would use is those that have gone through this process.

See where we're going? Back to the same argument of the actual targets of these measures not being affected at all.

5.4 Targeted Government Attacks

Some people have advocated for allowing the government to target and attack individuals that are suspected of committing a crime, with proper oversight and a court-order of course.

This lawful hacking would allow the authorities to use existing vulnerabilities to obtain evidence instead of creating new backdoors. This is pretty much employing the same techniques as criminals, but if there is proper judicial oversight then it may be a viable option.

Another benefit of going down this route is that it is expensive and time-consuming. This means that the authorities would hopefully limit its use to where it is actually needed. This, alongside the oversight, would ensure that the power was employed responsibly.

However, since these vulnerabilities affect public, commercial products, the government should carefully weigh each one and consider disclosing it responsibly. If they are not fixed, criminals could find the same vulnerabilities and wreck havoc with them.

Another drawback is one of trust. The general public has a sense of mistrust towards the government and its surveillance programs, ever since the Snowden leaks. And these concerns are somewhat justified; if the government has been abusing their power and spying on us with impunity for years, what's to stop them from doing the exact same thing again?

5.5 Improve Tech-Literacy

We should take measures to improve tech-literacy amongst the authorities.

This "going-dark" scare isn't something new. In the 1990s, organised-crime suspects started using disposable phones that hampered wire-taps but, regardless of this setback, the enforcement of the law and the prosecution of suspects continued[34]. Source [35] shows that even when encryption is used by suspected criminals, in the vast majority of cases the prosecution are still able to get enough evidence for a conviction.

Running into encrypted communications does not necessarily mean an evidence trail will go cold. Encryption can occur on a device, as the data is transmitted and when it is stored in the cloud. A dead-end due to encryption in one avenue does not necessarily mean the other avenues will also be encrypted.

For example, Apple can access the contents of an encrypted iPhone if it has been backed up to iCloud. Recognising how and when encryption occurs—and the different security offerings of service providers—may help law enforcement.

There is some acknowledgement of this lack of education, and its importance. The House Judiciary Committee & the House Energy and Commerce Committee—in their end-year report[36]—noted that there is "a significant gap in the technical knowledge and capabilities of the law enforcement community

[...] This results in a range of negative consequences that not only hinder law enforcement's ability to pursue investigations but also contribute to its tension with the technology community."

The report also noted that "encryption [...] is a global technology that is widely and increasingly available around the world", "there is no one-size-fits-all solution to the encryption challenge", and "any measure that weakens encryption works against the national interest". This is consistent with our views throughout this paper.

6 Conclusion

Keeping the conclusion short and sweet; if encryption is outlawed, only the outlaws will have encryption. Everyone loses in this scenario, including the government. The only winners would be criminals, who can carry on just as they had before.

Thus, the government should persue alternatives such as the ones that were discussed in section 5.4 and section 5.5.

The national prohibition of strong encryption is not feasible, and it would prove ineffective in the fight against crime.

References

[1] S. Simpson. *Cryptography in Everyday Life*. July 14, 1997. URL: http://www.laits.utexas.edu/~anorman/BUS.FOR/course.mat/SSim/life.html.

[2] M. Castell. *The devastating effects of irretrievable encryption*. 1999. URL: http://www.fipr.org/rip/Mark%20Castell%20NCIS%20-%20Devastating%20effects%20of%20irretrievable%20encryption.htm.

[3] *Investigatory Powers Act*. 2016. URL: http://www.legislation.gov.uk/ukpga/2016/25/contents/enacted/data.htm.

[4] R. Anderson. *The Crypto Wars Are Over!* May 25, 2005. URL: http://www.fipr.org/press/050525crypto.html.

[5] *The Crypto Wars: Governments Working to Undermine Encryption*. Nov. 12, 2016. URL: https://www.eff.org/document/crypto-wars-governments-working-undermine-encryption.

[6] *International Traffic in Arms Regulations*. Apr. 1, 1992. URL: https://epic.org/crypto/export_controls/itar.html.

[7] *Export of cryptographic items*. Dec. 12, 2012. URL: https://www.gov.uk/guidance/export-of-cryptographic-items.

[8] *The United States Munitions List*. Oct. 12, 2016. URL: http://www.ecfr.gov/cgi-bin/text-idx?node=pt22.1.121.

[9] T. Rid. *The cypherpunk revolution*. July 20, 2016. URL: http://passcode.csmonitor.com/cypherpunk.

[10] T.C. *What is the Streisand effect?* Apr. 16, 2013. URL: http://www.economist.com/blogs/economist-explains/2013/04/economist-explains-what-streisand-effect.

[11] *Export Controls for Conventional Arms and Dual-Use Goods and Technologies*. Dec. 1, 2015. URL: http://www.wassenaar.org/wp-content/uploads/2016/05/Stand_Alone_Munitions_List_WA.pdf.

[12] *Revised U.S. Encryption Export Control Regulations*. Jan. 10, 2000. URL: https://epic.org/crypto/export_controls/regs_1_00.html.

[13] J. Gordon. *Encryption, Open Source and Export Control*. Dec. 4, 2014. URL: https://www.thoughtworks.com/insights/blog/encryption-open-source-and-export-control.

[14] P. Reuvers and M. Simons. *Clipper Chip: Cryptographic Key Escrow*. July 26, 2016. URL: http://www.cryptomuseum.com/crypto/usa/clipper.htm.

[15] M. Blaze. *Protocol Failure in the Escrowed Encryption Standard*. Aug. 20, 1994. URL: http://www.crypto.com/papers/eesproto.pdf.

[16] Y. Frankel and M. Yung. *Escrow Encryption Systems Visited: Attacks, Analysis and Designs*. Aug. 31, 1995. URL: https://books.google.co.uk/books?id=Q-6qCAAAQBAJ&pg=PA222&ots=TWgiD_DVlp&sig=yjqTOoEdjChOVSo0JS5WKX0mOqs.

[17] J. McLaughlin. *Spy Chief Complains That Edward Snowden Sped Up Spread of Encryption by 7 Years*. Apr. 25, 2016. URL: https://theintercept.com/2016/04/25/spy-chief-complains-that-edward-snowden-sped-up-spread-of-encryption-by-7-years/.

[18] J. Appelbaum, J. Horchert, and C. Stöcker. *Catalog Advertises NSA Toolbox*. Dec. 29, 2013. URL: http://www.spiegel.de/international/world/catalog-reveals-nsa-has-back-doors-for-numerous-devices-a-940994.html.

[19] J. Ball, J. Borger, and G. Greenwald. *Revealed: how US and UK spy agencies defeat internet privacy and security*. Sept. 6, 2013. URL: https://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security.

[20] J. Cox. *What We Know About the Exploits Dumped in NSA-Linked Hack*. Aug. 7, 2016. URL: https://motherboard.vice.com/en_us/article/what-we-know-about-the-exploits-dumped-in-nsa-linked-shadow-brokers-hack.

[21] B. Cahall et al. *Do NSA's Bulk Surveillance Programs Stop Terrorists?* Jan. 13, 2014. URL: https://www.newamerica.org/international-security/policy-papers/do-nsas-bulk-surveillance-programs-stop-terrorists/.

[22] The Tor Project. *Users of Tor*. Apr. 13, 2017. URL: https://www.torproject.org/about/torusers.html.en.

[23] B. Schneier. *Attacking Tor: how the NSA targets users' online anonymity*. Oct. 4, 2013. URL: https://www.theguardian.com/world/2013/oct/04/tor-attacks-nsa-users-online-anonymity.

[24] K. Fiveash. *Tor de farce: NSA fails to decrypt anonymised network*. Dec. 29, 2014. URL: https://www.theregister.co.uk/2014/12/29/nsa_gchq_internet_security_pet_hates/.

[25] C. Farivar. *To keep Tor hack source code secret, DOJ dismisses child porn case*. Mar. 5, 2017. URL: https://arstechnica.com/tech-policy/2017/03/doj-drops-case-against-child-porn-suspect-rather-than-disclose-fbi-hack/.

[26] A. Hayes. *United States v. Jay Michaud: Motion to Dismiss*. Mar. 3, 2017. URL: https://www.documentcloud.org/documents/3482329-Michaud-motion-to-dismiss.html.

[27] T. Wilde. *Knock Knock Knockin' on Bridges' Doors*. Jan. 7, 2012. URL: https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors.

[28] D. Rijmenants. *The One-time pad*. Feb. 2, 2017. URL: http://users.telenet.be/d.rijmenants/en/onetimepad.htm.

[29] H. Abelson et al. *The Risks of Key Recovery, Key Escrow, and Trusted Third Party Encryption*. May 27, 1997. URL: https://academiccommons.columbia.edu/download/fedora_content/download/ac:127128/CONTENT/paper-key-escrow.pdf.

[30] H. Abelson et al. *Keys Under Doormats: Mandating insecurity by requiring government access to all data and communications*. July 6, 2015. URL: http://hdl.handle.net/1721.1/97690.

[31] J. McCallion, J. Curtis, and A. Lee. *Investigatory Powers 'will cost UK £1 billion'*. Mar. 30, 2016. URL: http://www.itpro.co.uk/it-legislation/26034/investigatory-powers-will-cost-uk-1-billion.

[32] *AES: the Advanced Encryption Standard*. Jan. 27, 2014. URL: https://competitions.cr.yp.to/aes.html.

[33] National Institute of Standards and Technology. *Tentative Timeline of the Development of New Hash Functions*. Sept. 14, 2016. URL: http://csrc.nist.gov/groups/ST/hash/timeline.html.

[34] A. Segal and A. Grigsby. *How to break the deadlock over data encryption*. Mar. 13, 2016. URL: https://www.washingtonpost.com/opinions/how-to-break-the-deadlock-over-data-encryption/2016/03/13/e677fb78-d110-11e5-88cd-753e80cd29ad_story.html.

[35] K. Schmeh. *When Encryption Baffles the Police: A Collection of Cases*. Oct. 19, 2016. URL: http://scienceblogs.de/klausis-krypto-kolumne/when-encryption-baffles-the-police-a-collection-of-cases/.

[36] House Judiciary Committee and House Energy & Commerce Committee. *Encryption Working Group - Year-End Report*. Dec. 20, 2016. URL: https://judiciary.house.gov/wp-content/uploads/2016/12/20161220EWGFINALReport.pdf.