# Rebel Alliance Tech Manual

Dated: Jan 12th, 2017

There are many texts that list off a few tools or programs they advise people to use while trying to hide context and considerations so as to not overwhelm. These texts usually target "normal people" and or professionals (like journalists and lawyers) who tend to be entitled and adverse to learning new things. Instead this text is geared towards activists and dissidents who are already self-motivated and want to be fully prepared for serious repression from authoritarian regimes. Specifically it intends to address common usecases and threats radical activists and dissidents face and answer every question commonly asked at trainings for radical activists and dissidents.

This is meant neither as a trivial introduction nor as a comprehensive education, but as something that bridges the gap between the two and provides conceptual understandings in addition to advice and best practices. While we will attempt to be acessible to everyone we will not lie to you "for your own good", we will attempt to be provide a map. Our goal is not to scare, but to be honest about the scary reality and the work that is necessary (and doable!) to achieve a significant measure of security.

We, the authors, have decades of experience as both hackers and anarchists, and we've personally assisted and trained hundreds of other activists and dissidents.

# Table of contents

# Note about the zine version

This zine was created in 2021 from the online version of the *Rebel Alliance Tech Manual*, taken from https://github.com/rebel-tech/Rebel-Alliance-Tech-Manual. We are not the original authors of this manual. This zine was made in order to make the information contained in the manual more readable and available in a printable format.

Apart from a few formatting tweaks, most things remain identical to the online version. Notable changes include :

- Installation instructions for the Tails operating system were removed, as they were outdated as of 2021. A link to the current, official installation instructions is available instead.
- Warnings were put at the beginning of sections that refer to Thunderbird's plugin *Enigmail*, which is now outdated – the PGP functionality it provided has migrated into Thunderbird in 2020.

Stay safe out there and good luck,

*The zine authors*

# First, the basic things

## Install Signal (whispersystems.org)

Signal is an text messaging app for Android and iPhones. It replaces the app you currently use to send text messages. When you text someone who likewise has Signal on their phone the text messages you send to one another will be encrypted every step of the way. Texts that were encrypted will have a small padlock icon next to them.

## Download TorBrowser (torproject.org)

TorBrowser is an enhanced version of Firefox that provides some annonymity to your internet browsing. It connects you to the Tor Network (a subset of the servers that relay information across the internet), and makes your connection to whatever websites you're visiting hard to track back. This obscures your location from the site you're visiting. It also obscures what website you're visiting from anyone watching your connection.

We will explain more about these two tools later. But they are simple to pickup and use immediately.

# Introduction

## The use of technology

The modern tools we use are complicated and many of us don't quite understand what goes on beneath the surface. This can lead us to make inaccurate models in our heads or treat technologies with either dismissiveness or superstition.

In particular many activists gravitate towards a kind of security nihilism, presuming that all our communications technologies are insecure and nothing can really be done to secure them so there's no point.

This creates a default mode where folks either refrain from using technologies when they would be better off using them, OR they use them with careless abandon, assuming that if they haven't faced state oppression yet, they won't in the future.

In the opposite direction many activists default to trusting treating applications or service providers like they would a person. In this view either a tool is trustworthy (ie totally has your back) or it's not. This sort of thinking obscures what exactly a tool is, what precisely you can trust it to do, and why. For example, people think that using an email provider like Riseup.net that's run by widely trusted and respected anarchist activists and technologists means that no one can see your emails, when this is largely untrue without using the additional tool of PGP. Indeed, no matter how good the email provider, email is almost as public and insecure as a postcard when not using PGP.

Before we finish this introduction we want to go over why using secure tools is important and sometimes much better than trying to avoid technology altogether. Don't be too alarmed, we will explain everything in detail later.

## 1st Example: The Raid

Many authoritarian governments like to raid and arrest dissidents just for belonging to a subculture or holding certain views. In the United States it's been common for the police to raid the houses of punks and activists before protests or just in periodic coordinated crackdowns on the community.

Let's say you come around a corner and see a SWAT team pouring into your house. You duck out of sight and immediately turn off all your devices and discard any that you can't also take the battery out of (good move). You may even head over to another friend's house and see the cops in front of their house too. What do you do? Who do you go to? Who do you contact? How do you even figure out the situation without putting yourself at greater risk or putting other friends at risk? How do you skip town if that's something you choose to do? How do you set up the things you need to set up?

If you know a little about how communications technologies work and what tools are available you could be in much better situation. With the right tools and some preparation before hand you could still be able to contact your lawyer and friends in a way the state can't trace or read.

One solution here is to learn and adopt PGP with your friends beforehand and then use PGP from a computer that you boot from a USB with the Tails Operating System on it. PGP would encrypt your communications and Tails would mask your location. If you don't carry your laptop

around regularly you could prepare by creating a TAILS USB and a USB with your PGP key on it and then sticking both in a sealed bottle and burying it somewhere away from your house where you can always retrieve them and then use at a public computer.

## 2nd Example: The Meeting Scheduling

Many activist groups and individual organizers face persistent survellience from the police. Police infiltrate a huge number of groups from grandmas knitting sweaters in protest of the war to the cops' infamous campaign to infiltrate vegan potlucks. When they can't assign an undercover or informant to sit through another boring meeting about banner colors they will often try to bug the spaces activists meet in beforehand and track your phones. They do this to gather intel on activist groups and their actions. Even planning a tame protest can result in preemptive (unwarranted) raids and arrests if the cops know when you're doing it, so as to save a corporation from the loss of profit that might result from a picket (often labeled "economic terrorism").

Let's say that your group is organizing a project or a protest and needs to meet repeatedly. You're all close friends or highly vouched for so the odds of infiltration are low. You either meet at houses where the odds of police getting a bug in are likewise low, or you plan on meeting up somewhere and then decide on a second location together and all travel there for the actual meeting. You may decide a new meeting place and time for the next meeting every time. And if you use a regular meeting spot you all never bring your phones when traveling to it so that the location can't be tracked and identified.

But! Then something changes and you suddenly need to schedule an emergency meeting at a time and place that hasn't been decided upon prior. You can try to literally drive around from house to house doing meetings one-on-one and relaying information piecemeal about best times that work for individual people. But now you're introducing all kinds of additional work that makes arranging the meeting between everyone almost impossible. So your group either doesn't meet in a timely fashion. Or you say "fuck it all internet security is impossible" and just text message or call one another the normal unencrypted way. However this increases the chance that the cops will infiltrate.

On the other hand you could use Signal. The cops could figure out that you're all talking together. And there's a chance they could backdoor one of your phones (in addition to tracking it), but the odds of that are much lower, and so there's a good chance you could speak both semi-securely and productively about meeting locations and times.

## Broken tools

We give these examples to encourage you to learn more about the lay of the land when it comes to electronic security. Simplistic paranoia is highly counterproductive. Activists can and do use technology to successfully augment their work and resistance. You just need a little bit of understanding first and we will provide it here.

That said, let us be clear that almost every tool or program a normal person uses by default is horribly broken. Often this is by design. While the internet and communications technologies have immense potential for liberation and can be leveraged to great effect, our digital infrastructure is itself a site of political conflict. For decades authoritarians have sought to normalize insecure tools and standards.

Many people are surprised to learn that it's trivial for the police to access everything you've put on your social media profile or sent in messages from it, regardless of your "security settings." Similarly the "incognito mode" advertised on your browser is basically meaningless; the sites you visit, as well as your identity and location are all exposed to the world, and certainly to the government or any diligent adversary. To make matters worse the padlock that sometimes appears the corner of your browser's address bar means very little and certainly doesn't protect you against the government.

Most applications out there that advertise as offering encryption are either snake oil, unverifiable, or have not gotten sufficient attention by experts. There is however a very large and diverse community of activists, cryptographers, and security experts who have come to a consensus on some best tools and practices.

# Core concepts

## Encryption

This is the matter of hiding *what* is being communicated -- not necessarily who is communicating with whom, where they are, or that they're communicating using a crypto tool. Communicating over the internet is a lot like passing a note to someone in class; encryption is sealing the note in a magically strong envelope that can't be opened by anyone but the recipient. But with encryption *alone* everyone along the way can still see that a note is getting passed and who it's addressed to. Additionally someone else could stick a note in an envelope, address it to your crush, and pretend that it was from you. Or someone could just snatch the note up and refuse to deliver it. Encryption alone only makes the content of the note unreadable to third parties, other tools are necessary to deal with all the other potential problems.

### Normal Encryption

Many of you were as kids familiar with the "ceasar cipher" where you change each letter in a message by a number of places in the alphabet. So for exampe the letter "A" might be moved two places and become the letter "C", similarly the letter "H" would become "J", and the letters at the end looping back around so "Y" would become "A". Under this change "HELLO ALICE" becomes "JGNNQ CNKEG". The problem with this sort of ancient approach is that if you do this sort of encryption long enough it becomes obvious by what number you're moving all the letters forward in the alphabet. One simple trick is to count the letters, and since "E" is the most common letter in normal texts of english the letter that's the most common in the encrypted text is probably "E". Now the *totally unbreakable* way to do cryptography is to choose a new random number per each character. So if the number for one "E" is 7 then it becomes "L" but maybe for another "E" it's 3 and thus "H". The problem here is that you have to have as many random numbers as you have letters that need to be encrypted. You can't reuse your set of numbers or else a fancy adversary with a very smart computer could start to find the patterns again. So you and whoever you want to communicate with have to first have a shared long series of numbers. This is unweildy. You'd have to have a "key" as long as the message.

The way that modern crypto works is by using fancy mathematics to turn a very small "key" into a very big "key" without just repeating it, but also making it impossible for someone to pick out patterns. These methods can even turn a small key into an infinitely large key through a process that is just repeated as much as needed. The precise mathematical procedures are very interesting but irrelevant to this guide.

One way this can work is to take something very small (like a password) and turn it into a very large key capable of encrypting and unencrypting a whole harddrive. Note that with this conventional form of encryption -- just as with the "ceasar cipher" -- the key you use to encrypt the message is the same key someone needs to decrypt the message. This is intuitive to a lot of people. But there's a different category of encryption that only became possible in the 1970s....

### Public Key Encryption

In public key encryption the key that encrypts a message is a different key from the key necessary to unencrypt it. Instead there's a pair of keys: the public key and the private key. The

public key can encrypt, but only the private key can decrypt. (Although of course anyone using the public key knows what the message they encrypted with it is.)

You can think of this being like having a lockbox with a slit and a key. Anyone can stick something inside the lockbox, but only the person with the key can open it. Or you can think of it in terms of magical envelopes. You can hand out as many copies of your magical envelope as you like. But once someone has sealed a note within it only you can open it.

Public key encryption is the bit of mathematical magic that the internet and our entire modern economy is based around. It works because in mathematics that there are problems you can do in one direction but are extraordinarily hard to do in the reverse direction. We know it works not just because the math checks out and thousands of expert mathematicians have checked, but also because if anyone could break it they could immediately get rich.

If you want to get a better grasp on how public key encryption works in all our fancy programs read on. However if you're happy with the above explanation then feel free to skip the next section of elementary math.

One common example of a one-way function is factorization; it's relatively easy to take a set of prime numbers (numbers that can't be divided by anything besides themselves and 1) and multiply them together ( 37 x 41 = 1517 ) but it's a lot harder to take a large number and find out what prime numbers have to be multiplied together to make it. You might think of yourself having to do trial-and-error dividing the number by all possible primes again and again...

One of the most common and famous examples of asymmetric encryption is actually something that can be understood if you know how to do exponentiation. It goes something like this:

1. I come up with a number X, and publicly tell you it. [Let's say X = 7]
2. I also come up with a number A, but don't tell you it. [Let's say X = 4]
3. You come up with a number B, and don't tell me it. [Let's say X = 3]
4. You figure out what X^B is and send me THAT number. [ 343 ]
5. I figure out what X^A is and send you THAT number. [ 2401 ]
6. You take the number that I sent you (X^A) and then raise it by your secret number B. So now you have (X^A)^B. [ 13841287201 ]
7. I likewise take the number that you sent me and raise it by my secret number A. So now I have (X^B)^A. [ 13841287201 ]

Note that these two resulting numbers are the same number. Because the way exponentiation works -- if you've forgotten -- is that raising a number to two numbers in a row is the same as raising that number by the multiple of those two numbers. (X^A)^B = (X^B)^A = X^(BA) So we end our conversation with the same secret number [ 13841287201 ].

In our example numbers all that we've shared are the numbers 7, 343, and 2401. In order for someone overhearing us to get the same number they'd have to factor both 343 and 2401 and find what powers of 7 they are. This is somewhat easy in our example because we picked small numbers (3, 4) but you can imagine it can get much harder if we'd chosen larger numbers.

However the real challenge for computers is when we add one more trick: an agreed upon maximum number after which the count resets. Like how on a clock after you reach 12 the next number is 1 again. For our max number we might choose 20000. Then we can think of the

number 13841287201 going around the clock (passing multiples of 2000) many times and finally comes to rest on 7201 (the remainder of 13841287201 divided by 20000). This added trick is sufficient to make the reverse engineering of our shared secret number from our public numbers incredibly hard for computers (provided of course that the numbers involved are unbelievably bigger than the examples we've given).

In the example we gave the public keys would be 343 and 2401. By exchanging these public keys over the public internet to one another we're able to encrypt respooses to one another and arrive at a common shared key. And no one just passively observing what we passed to one another in the beginning would be able to figure out the shared encryption key we ended up with that then allows us to use normal old symmetric encryption.

Again you might ask, "this seems simple, surely someone smarter than me could figure out a hole or someone with a large enough supercomputer could do the hard problem of reversing?" But many mathematical geniuses have attacked this and the way the mathematics works out as the size of the public numbers grow the hardness of the problem grows much much faster. We know a lot about what sort of technology is on the market. And we know from the Snowden leaks and many other analyses that the NSA doesn't have some secret massive infrastructure that has been building supercomputers light-years beyond anything anyone knows about. We also know they don't because folks have repeatedly been able to pull one over on them while using public key encryption.

## Authentication

Authentication is the issue of making sure that the person you're communicating with is actually the person you think they are. To return to the example of the note passed in class, authentication is like putting a magical seal with your markings on the envelope that only you could put on it.

It's important to send messages that are both encrypted AND authenticated.

You could attempt to communicate through encryption with your friend Tom on the internet, but as you both attempt to make an encrypted connection with one another online an NSA goon listening in could jump on the network inbetween your two computers and pretend to be each person.

So you would have an encrypted connection to the NSA goon, and Tom would have an encrypted connection to the NSA goon too. Then as you message Tom the NSA goon could just take that message from the encrypted connection he has with you, read the message you unknowingly encrypted to the NSA goon, and then pass that message along to Tom in the *different* encrypted connection that the NSA goon and Tom share.

Both you and Tom see that you have an encrypted connection, and you see the messages you're passing one another, but you don't see that the encrypted connections you have are *different*. Your messages are not being passed over a one You-Tom connection but a You-NSA connection then an NSA-Tom connection.

This is called a "man-in-the-middle" attack and it's relatively easy to automate. Folks have also discovered the NSA and other governements doing this to millions of people.

Encryption is not authentication.

So how do you authenticate? Well you check with one another (off the internet or through other channels that are unlikely to be intercepted) to see that the fancy numbers that represent each other's public keys or your mutual connection are the same.

Every encryption program of any value provides a way to authenticate. In PGP you collect public keys for each person you communicate with, and then when you're both in person you can compare the public keys you have on file for one another. In Signal every conversation provides the option to display a set of secret numbers for that conversation, and in person you can both look at each other's phones and see that the numbers match (or scan each others' barcodes).

In general if program offering "encrypted" communication has no clear option for users to also do authentication, it's a bad program and you shouldn't use it.

## Untraceability, Anonymity

> "We kill people based on metadata" --Michael Hayden, Director of the NSA (1999-2005)

When you pass a note in class or address a package to be mailed it tends to have both a destination address and origin address. Even if you've encrypted what's inside you still have to make sure that it arrives at its intended destination. While an observer may not be able to tell WHAT you're sending they can still tell that you sent something, and who you sent it to.

Any information sent with modern communications technologies likewise inherently involves such "metadata." And it's not just who sent what and who recieved. It can be things like WHEN and WHERE both of you were located. As you can imagine sometimes this metadata alone is sufficient to make encryption meaningless.

Phones calls are a good example, with metadata alone:

- They can tell you rang a phone sex service at 2:24 am and spoke for 18 minutes. But not what you talked about.
- They can tell you called the suicide prevention hotline from the Golden Gate Bridge. But not the topic of the call.
- They can tell you spoke with an HIV testing service, then your doctor, then your health insurance company in the same hour. But not what was discussed.

The exact same sort of problems arise with the internet and encrypted applications:

- In 2012 police raided an anarchist activist and charged them with every bank vandalism in the pacific northwest over a two year period. This was of course preposterous. But the two pieces of evidence brought against them were: 1) that they had pink hair and a member of a black block that had vandalized an ATM had likewise had a tuff of pink hair poke out from underneath their hoodie, and 2) that they had sent an encrypted text message using Signal from their phone just before a bank vandalism and in the same neighborhood as it. The FBI couldn't decrypt the message itself, but they argued that the timing and location constituted evidence. The activist in question ultimately ducked the 72 felony charges in court, but the case is illustrative of the problem of metadata. Even though they couldn't decrypt the messsage itself the police knew a message had been sent from *their* phone, they knew *when* it had been sent, and they knew *where* it had been sent.

Just like pink hair poking out from underneath incomplete block bloc attire, metadata can end up helping to expose critical information. Metadata can be added "helpfully" by for example the program you used to create an image or document ("this document created by the user 'Eric Malatesta' on 'my computer'"), but it's also a baked into the very manner by which information flows through most of our communications networks. Even when we manage to encrypt our messages, encryption doesn't hide who's sending the message and who's intended to recieve it.

There is a sneaky way around this however. The idea is basically to put addressed envelopes inside of other addressed envelopes. So you might want to secretly communicate to Jake, but you stick the sealed envelope to him inside another envelope that you address to your friend Sam. Then when Sam gets the envelope from you he opens it, finds the envelope addressed to Jake and forwards it along. This way the busybody sitting next to you that you have to first hand your envelope to only sees that you're sending something to Sam.

This whole game of putting sealed envelopes insides other sealed envelopes is basically how modern anonymity software works.

The single-hop approach where you only stick a single envelope inside another one -- as with Sam above -- is how modern VPNs work (albeit with a whole internet connection). This can be useful in a small number of cases. For example, the busybody in the seat next to you in class, or a hacker sitting on the same wifi as you in a cafe. But the VPN approach is insufficient for most activists and dissidents. It requires you to place your trust in someone (Sam!) and it's relatively easy for powerful states to watch the messages being passed around and decipher who originally sent what.

Anonymity networks like Tor and I2P go several steps further, akin to nesting even more envelopes insides of envelopes. This makes watching everything much harder and makes it impossible for a single person to rat you out. Instead every single person you address one of the envelopes to would have to collude together, or some superpower would have to get very lucky watching all the flows of traffic and tracing everything. Tracing everything is quite computationally hard -- too hard experts think for even hypothetical NSA supercomputers and surveillience machines. And these anonymity networks work to try and assure that most of the servers are run by activists or are otherwise uncompromised. The main trick however is having the relaying servers all run by *different* people. So long as superpowers like the US, China, and Russia remain competative and don't collaborate then there's a good bet even if they each compromise some relay servers they won't share their information, and thus won't be able to de-anonymize anyone.

However unlike encryption this kind of anonymity is not a sure thing. We have good reason to believe that it almost always works, but there *are* rare exceptions. And activists can still screw themselves over if they're not careful.

If you use something like Tor to check a Facebook account under your own name then obviously anyone at the end of that connection can see that you're online, that you're connecting to Facebook and probably what you're doing on Facebook. What they probably won't see however is *where* you're connecting to Facebook from. So for example activists on the run from the Syrian government could still check Facebook without revealing what city they were staying in as they fled.

Similarly if you use Tor to look up weather in your city then someone watching could easily figure out that you are a Tor user in that city, but if you're not doing anything connected to your

personal identity they may not know *who* you are. Tor Browser separates your connections to Facebook and the weather site by using different paths through the Tor Network for each one, making it relatively safe to do both at the same time.

However no technology is a magic bullet for anonymity. If you use Tor to post a statement and sign your name on that statement that statement is obviously not anonymous. Similarly if you create and upload an image in that statement, but it was created on your computer your image software may have automatically added your username to that image "to be helpful!" thus possibly revealing who you are. Your prior posts online may even be useable to identify you by your writing style. If you're on the run and you don't log in to any account connected to you but you do keep the same browsing habits (eg visit the same unique set of webcomics) they could concievably identify you by them. We don't know of any activist anywhere who's been decloaked by writing style or favorite webcomics, but those are the sort of possibilities to keep in mind.

## Steganography, Obfuscation & Censorship Circumvention

Often the problem isn't even hiding who you are, it's just finding any way to get online past a censor -- or *hiding* that you're doing anything encrypted. Encrypted communications are usually quite obviously encrypted and sometimes you can't risk anyone seeing that you're using encryption.

Steganography is different from encryption, whereas encryption hides the content of the message, steganography attempts to hide that there was a message at all. You can think of spies exchanging messages in discarded coffee cups at a park bench. The idea is to hide from an observer that they were exchanging anything at all.

One classic example of steganography in the electronic era is to make small changes to images posted online. So you might set up a cute kittens or meme page and upload images that sometimes introduce slight pixel changes on the edges. You could think of someone grabbing an image that involves grey clouds at the top and altering the top line of pixels in the image to alternate different shades of grey according to morse code. This -- incidentally -- would be a BAD form of steganography, but it's pretty close to what some terrorists have been caught doing.

There are at present a huge array of tools online claiming to provide steganography. We advise extreme caution about almost all of them. None of these tools is fully mature, but they are also probably the future of struggle between users and authoritarian governments.

TorBrowser provides the option of some limited obfuscation to hide that you're using Tor from censors and firewalls that try to spy on your traffic to the internet. Note that this just hides your connection to the Tor network, it does not hide that you're using Tor from the sites you visit! To enable such "pluggable transports" click "configure" when setting up TorBrowser and select yes under "Does Your ISP block Tor?" then select a type of bridge and click connect.

# Tools

We recommend as a bare minimum for activists and dissidents that you do *all* of the following:

1. Use full disk encryption on your computer, tablets and phones. To prevent police or other theives from reading the contents of your devices if they seize them while turned off.
2. Use a key manager like KeePassX to store unique passwords for every account you use; share encrypted databases of accounts and passwords within your groups and organizations; and keep encrypted databases with accounts and instructions for lawyers, friends and support groups in case of your imprisonment or death.
3. Never hold sensitive discussions while near a phone. Always turn them off and leave them in a location where your discussion will be inaudible. Never bring a phone with you while visiting secret locations or engaging in an action.
4. Use Signal in all situations when you are chatting casually or trying to schedule meetingss, but nevertheless try to avoid having sensitive conversations over Signal.
5. Use TorBrowser when visiting sites when you want your connection to that site to be visible, when you want to comment or browse anonymously, or when you want to hide your current location.
6. Learn how to encrypt emails with PGP using Enigmail and Thunderbird. Set up your own PGP key on at least one email account you use and get in the habit of using it. This is more difficult, but can be critically useful in a broader set of contexts than Signal.

We will cover many other tools, advice, and situations. But these constitute the bare minimum.

Depending on your precise context you will probably want to go beyond this minimum in certain directions.

## Signal

Signal is a free and open source software application for Android, iOS, and Desktop that employs end-to-end encryption, allowing users to send end-to-end encrypted group, text, picture, and video messages, and have encrypted phone conversations between Signal users. Although Signal uses telephone numbers as contacts, encrypted calls and messages actually use your data connection; therefore both parties to the conversation must have Internet access on their mobile devices. Due to this, Signal users don’t incur SMS and MMS fees for these type of conversations. On Android, Signal can replace your default text messaging application, so within Signal it is still possible to send unencrypted SMS messages.

Signal is a very easy to use app, and because of that ease many activists tend to fall into over using it. We advise strongly against that. While it's *far* better to use Signal to communicate electronically than to use nothing secure, Signal has limits.

- Smartphones are not as securable as normal computers. Indeed smartphones are shockingly insecure and easy to hack. If your phone is infected with malware then that malware can compromise Signal. And phones are pretty easy to attack and infect. Additionally phones are often more likely to be seized and many people will not have time to turn off their phones before the police take them (if Signal is open when the cops take your phone then whatever local encryption your phone might have is meaningless). So

turn on your phone OS's encryption, screenlock your phone, and turn it off entirely if you're getting pulled over or raided. Also delete old conversations. But also realize that the people you're talking to might *not* be taking such precautions, thus endangering exposing your conversations with them.

- Signal messages leak metadata. Your phone leaks *when* you message and *where* you message from.

It is often preferrable to use appropriate tools on a regular computer instead of Signal. And despite Signal being better than nothing it's often *far* preferrable to say nothing at all about certain things. Don't fall into the trap of activists so enthused by Signal's easiness and security that they say anything and everything through Signal.

Some additional notes:

- While Signal messages are encrypted user-to-user without trusting anyone in-between, the Signal app itself requires a centralized architecture that could be shut down. If the state raids their servers and smashes them then the Signal apps everyone downloaded becomes useless.
- Your Signal app gets updates from the folks who write the Signal codebase. So you have to trust the developers and the other hackers checking their public code. However the Signal devs are a bunch of anarchists with long histories and wide networks of friends you're probably connected to, and the code they write has been checked and enthusiastically signed off on by all the top experts.
- Signal's encryption uses "perfect forward secrecy" which means that if they get your private encryption keys months later by seizing your unlocked device they still can't retroactively decrypt prior conversations that they may have recorded in encrypted form. Of course if those past conversations haven't been deleted on your device and are still visible then they can still read them.

## Installation

On your iphone or Android device, enter the App Store or the Google Play store and search for “Signal.” Select the app Signal by Open Whisper Systems. After you tap “Install,” you’ll see a list of Android functions that Signal needs to be able to access in order to function. Click “Accept.” After Signal has finished downloading, tap “Open” to launch the app. You will then need to register and verify your phone number. In order to verify your phone number, you will be sent an SMS text with a six-digit code. Since Signal can access your SMS text messages, it will automatically recognize when you’ve received the code and complete your registration. After this process is complete, you'll be asked if you want Signal to be your default SMS app. This can be useful to keep all your messages in one place. Be aware that if you accept this, messages sent to contacts that do not have Signal installed (even if you send them from within the Signal app) will not be encrypted.

## Basics

In order to send encrypted messages or have encrypted phone calls with Signal the person that you are messaging or calling must also have Signal installed. If you try to send a message to someone using the Signal app and they do not have Signal installed, it will send a standard, non-encrypted text message. If you try to call the person, it will place a standard phone call.

Note that Open Whisper Systems, the anarchist makers of Signal, use other companies' infrastructure to send its users alerts when they receive a new message. It uses Google on Android, and Apple on iPhone. That means information about who is receiving messages and when they were received may leak to these companies.

To get started sending encrypted messages, tap the pencil icon in the lower-right corner of the screen.You will see a list of all the registered Signal users in your contacts. You can also enter the phone number of a Signal user who isn’t in your contacts. When you select a contact, you'll be brought to the text-messaging screen for your contact. Note that for Signal users, you'll see the text "Send Signal Message" - this means that the message will be encrypted. On this screen, the "phone" icon in the upper right corner of the screen will indicate that you can make an encrypted voice call using Signal as well. From this screen, you can send send end-to-end encrypted text, picture, or video messages. For users that do not have Signal installed, you'll see the text "Send unsecured SMS", which will not send the message with encryption. On this screen, the "phone" icon in the upper right corner of the screen will make a regular, unencrypted phone call.

There will be a padlock icon under messages that are sent encrypted. And the phone icon at the top of the app will have a padlock icon if that contact likewise has Signal and thus can recieve an encrypted call.

## Authentication

It's important to verify the identity of the person you're messaging with, to ensure that their encryption key wasn't tampered with or replaced with the key of someone else when your application downloaded it.

Verifying is ideally a process that takes place when you are physically in the presence of the person you are talking with. First, open the screen where you're able to contact someone, or just an existing conversation with them. From this screen if you're on Android tap the three dots in the upper-right corner of the screen and select "Conversation settings." If you're using an iPhone you will have to tap or hold down the name of the contact (at the top of the screen), people often have trouble getting this part to work on their iPhones. From the following screen, tap "Verify safety numbers." You will now be brought to a screen which displays a QR code and a list of "safety numbers." This code will be unique for every different contact you are conversing with. Have your contact navigate to the corresponding screen for their conversation with you, so that they have a QR code displayed on their screen as well. On your device you can tap on your QR code which will use the camera to scan the QR code that is displayed on your contact's screen. Align your camera to the QR code and after a second or two hopefully the camera will scan the barcode and display a green checkmark. (However if one of you is using an out-of-date version of Signal then there might arise problems here.) If the barcode doesn't match a red X will show instead of the green checkmark (if this happens the two of you should avoid from communicating over Signal and you should contact your friendly neighborhood hacker).

## Disappearing Messages

Signal also has a feature called "disappearing messages" which ensures (to some degree) that messages will be removed from your device and the device of your contact some chosen amount of time after they are seen. To enable "disappearing messages" for a conversation, open the screen where you are able to message your contact. From this screen if you're on

android tap the overflow icon (the three dots in the upper-right corner of the screen) and select "Disappearing messages."

### Phonecalls

To initiate an encrypted call to a contact, select that contact and then tap on the phone icon. You'll know that the contact can accept Signal calls if you see a small padlock icon next to the phone icon. Once a call is established, both parties to the call will be shown a random pair of words. This word pair will allow you to verify your identity and keys with the other user—also known as key verification. You can read the words aloud if you recognize the caller's voice, although very sophisticated attackers might be able to defeat this if they needed to. The word pair must be identical on both users' phones for you to be sure your message is not being intercepted.

## Tor

Tor is a volunteer-run service that provides both privacy and anonymity online by masking who you are and where you are connecting. Traffic through Tor is shuffled (in encrypted layers) between different servers on the Tor network until finally exiting into the normal internet. Volunteers and activist groups set up Tor relays and other volunteers and some paid staff at the Tor Project collaborate to write the open source software that runs Tor.

Tor is frequently the subject of misinformation and paranoia. The core idea of Tor (and other anonymity networks like I2P) was first dreamed up by scientists working for the US navy. The hackers who worked as early developers of the Tor project took grant money from the US government before releasing all the code as open source. Hackers pooled together and founded The Tor Project as a nonprofit. There's no doubt that there's overlap between the interests of some parts of the US government (and other governments) and the activists and anarchists that run The Tor Project. And The Tor Project has continued to take grant money from organizations that take money from the US government, although they've worked to raise other sources of money and refuse to accept money with strings attached. Certain portions of the US military industrial complex want a way for spies to communicate anonymously, this only works if everyone uses it and there's no backdoors. There is of course a tension within the US government between those wings that benefit from Tor, and those wings that suffer from it.

Tor is a benefit to dissidents around the world, both dissidents against regimes the US dislikes, and dissidents against the US and regimes it's allied with. It's also a used by journalists, activists, criminals, and anyone in fear of being de-anoynymized. The US government does not like the function of "hidden services" (websites only accessible through Tor thus hiding who hosts them and where), but The Tor Project continues to support this feature.

Tor is regularly rumored to be broken, insecure, and hacked. Most of these rumors should be taken with a large helping of salt.

The NSA has thrown piles of money at breaking Tor, and yet their internal documents as leaked by Snowden reveal that they're largely frustrated. "Tor Stinks!" "We will never be able to de-anonymize all Tor users all the time" "With manual analysis we can de-anonymize a very small fraction of Tor users, however NO success de-anonymizing a user in response to a request"

Since then law enforcement agencies have nonetheless occasionally managed to deanonymize or hack a small number of Tor users. However this wasn't thanks to a flaw in Tor itself, but a flaw

in Mozilla Firefox, the web browser that Tor Browser is built around. Typically someone uses Tor to visit a hidden service that has been successfully hacked by a government and the compromised site sends a web page to the Tor Browsers of visitors designed to hack them. This hasn't happened that frequently, but has happened. Usually Tor developers are very quick to patch Tor Browser to protect from such attacks the moment they're discovered.

Users who disabled javascript and flash in Tor Browser's settings were *not* compromised by these attacks. It's strongly recommended you turn Tor Browser's security settings to "High" which would protect you against such attacks (although it may prevent the loading of some video and make certain sites load ugly or without some features).

Academics write papers regularly exploring ways that the Tor network itself could be compromised (or further secured) and some academics have collaborated with the FBI in trying to attack the Tor network itself. These instances are much more rare and usually defended against relatively quickly.

Tor is less-than-perfect software constantly besieged by the nations of the world, but it's designed and maintained by hundreds of committed anti-authoritarian activists and they're usually winning more than they're losing.

We *strongly* recommend using Tor and Tor Browser for anonymity online. But don't literally bet your life on it unless you have no better choice.

## Tor Browser

To download Tor Browser go to torproject.org

Do not use any other source, and if you are prompted to accept alternative HTTPS (SSL/TLS) security certificates, do not proceed.

Tor Browser is the most common and simple way to access and use the Tor network. When you go to torproject.org the big "Download" button will be for Tor Browser and will send you to a screen featuring versions for your operating system and computer (if prompted with versions for "32-bit" and "64-bit" know that most modern computers are "64-bit"). Click the download button. If you're on windows you'll get an EXE file that you can install quickly. On Macs and Linux you'll get a compressed file that you can unzip/extract it andthat will create a folder with some files in it (you can just double click on the "start tor" icon inside every time you want to run it, no installation required).

The first time Tor Browser starts, you'll get a window that allows you to modify some settings if necessary. You might have to come back and change some configuration settings, but go ahead and connect to the Tor network by clicking the Connect button. After clicking “Connect,” a new window will open with a green bar that will get longer as the Tor software starts up.

You can verify that you are connected to the Tor network by visiting check.torproject.org. If you are connected the website it will say “Congratulations. This browser is configured to use Tor.”

### *Tips for using Tor Browser*

1. Use Tor Browser Tor does not protect all of your computer's Internet traffic when you run it. Tor only protects your applications that are properly configured to send their Internet traffic through Tor. To avoid problems with Tor configuration, we strongly recommend you use the Tor Browser. It is pre-configured to protect your privacy and anonymity on the

web as long as you're browsing with Tor Browser itself. Almost any other web browser configuration is likely to be unsafe to use with Tor.

2. Don't torrent over Tor Torrent file-sharing applications have been observed to ignore proxy settings and make direct connections even when they are told to use Tor. Even if your torrent application connects only through Tor, you will often send out your real IP address in the tracker GET request, because that's how torrents work. Not only do you deanonymize your torrent traffic and your other simultaneous Tor web traffic this way, you also slow down the entire Tor network for everyone else.
3. Don't enable or install browser plugins Tor Browser will block browser plugins such as Flash, RealPlayer, Quicktime, and others: they can be manipulated into revealing your IP address. Similarly, we do not recommend installing additional addons or plugins into Tor Browser, as these may bypass Tor or otherwise harm your anonymity and privacy.
4. Use HTTPS versions of websites Tor will encrypt your traffic to and within the Tor network, but the encryption of your traffic to the final destination website depends upon on that website. To help ensure private encryption to websites, Tor Browser includes HTTPS Everywhere to force the use of HTTPS encryption with major websites that support it. However, you should still watch the browser URL bar to ensure that websites you provide sensitive information to display a blue or green URL bar button, include https:// in the URL, and display the proper expected name for the website. Also see EFF's interactive page explaining how Tor and HTTPS relate.
5. Don't open documents downloaded through Tor while online Tor Browser will warn you before automatically opening documents that are handled by external applications. DO NOT IGNORE THIS WARNING. You should be very careful when downloading documents via Tor (especially DOC and PDF files, unless you use the PDF viewer that's built into Tor Browser) as these documents can contain Internet resources that will be downloaded outside of Tor by the application that opens them. This will reveal your non-Tor IP address. If you must work with DOC and/or PDF files, we strongly recommend either using a disconnected computer, downloading the free VirtualBox and using it with a virtual machine image with networking disabled, or using Tails. Under no circumstances is it safe to use BitTorrent and Tor together, however.
6. Use bridges and/or find company Tor tries to prevent attackers from learning what destination websites you connect to. However, by default, it does not prevent somebody watching your Internet traffic from learning that you're using Tor. If this matters to you, you can reduce this risk by configuring Tor to use a Tor bridge relay rather than connecting directly to the public Tor network. Ultimately the best protection is a social approach: the more Tor users there are near you and the more diverse their interests, the less dangerous it will be that you are one of them. Convince other people to use Tor, too!

## Tails

Tails is a complete operating system designed to be used from a DVD, USB stick, or SD card independently of the computer's original operating system. Tails comes with several built-in applications pre-configured with security in mind: web browser, instant messaging client, email client, office suite, image and sound editor, etc.

Besides ideally leaving no mark on your computer to show that you ever ran Tails on it rather than your normal operating system, Tails ensures that all software on it is configured to connect to the Internet through Tor and if an application tries to connect to the Internet directly without using Tor, the connection is automatically blocked for security.

### *Installation*

Complete, friendly and detailed instructions can be found at https://tails.boum.org/doc/.

### *Limits to Tails*

- Tails does not protect against compromised hardware
- Tails can be compromised if installed or plugged in untrusted systems
- Tails does not protect against BIOS or firmware attacks -- if your computer is already severely compromised then Tails will be compromised
- Tor exit nodes can eavesdrop on communications
- Someone watching your network traffic can see that you are using Tor and probably Tails
- Tails doesn't encrypt your documents by default
- Tails doesn't clear the metadata of your documents for you and doesn't encrypt the Subject: and other headers of your encrypted email messages
- Tor doesn't protect you from a global adversary -- if all superpowers collude you can be decloaked.
- Tails doesn't magically separate your different contextual identities -- shutdown and restart Tails every time you're using a new identity, if you really want to isolate them better.
- Tails doesn't make your crappy passwords stronger
- Tails is a work in progress

## Using Tor on Android phones (Orbot, Orweb, and Gibberbot)

You can install the free app "Orbot" (from the GooglePlay store, the F-Droid repository, or from https://guardianproject.info/releases). This will establish a connection to the Tor network on your phone (when the app is run), allowing other apps like Orweb (for webbrowsing) or Gibberbot (for instant messaging) to use the Tor network. Your phone's other apps will not be routed through Tor.

## Sharing Files over Tor (Onionshare)

It's a generally bad idea to attempt to pipe torrents over the Tor network, however the situation frequently arises where you need to send someone a file anonymously or while cloaking both of your locations. Onionshare is a a cute little tool that you can download from onionshare.org that enables one person to share their file over Tor. However, note that you will have to keep your computer on and connected to the Tor network so that the other person can access the file (if you suspend your laptop, for example, the URL won’t work until you get back online). And also note that Tor is slower than the normal internet and so the download may take a very long time, depending on its size. The moment the download is complete, OnionShare shuts down the web service, the URL no longer works, and the files you shared disappear from the internet.

When open OnionShare you can drag some files into it and click the “Start Sharing” button. After a moment, OnionShare gives you URL that looks something like http://4a7kqhcc7ko6a5rd.onion/logan-chopin. You send this URL to someone you’d like to share files with, and they load it using Tor Browser, downloading the file from you.

### *What OnionShare protects against*

- Third parties don't have access to files being shared. The files are hosted directly on the sender's computer and don't get uploaded to any server. Instead, the sender's computer

becomes the server. Traditional ways of sending files, like in an email or using a cloud hosting service like Dropbox or Google Drive, require trusting the service with access to the files being shared.

- Network eavesdroppers can't spy on files in transit. Because connections between Tor onion services and Tor Browser are end-to-end encrypted, no network attackers can eavesdrop on the shared files while the recipient is downloading them. If the eavesdropper is positioned on the sender's end, the recipient's end, or is a malicious Tor node, they will only see Tor encrypted traffic.
- Anonymity of sender and recipient are protected by Tor. OnionShare and Tor Browser protect the anonymity of the users. As long as the sender anonymously communicates the OnionShare URL with the recipient, the recipient and eavesdroppers can't learn the identity of the sender.
- If an attacker enumerates the onion service, the shared files remain safe. There have been attacks against the Tor network that can enumerate onion services. If someone discovers the .onion address of an OnionShare onion service, they still cannot download the shared files without knowing the full URL, and OnionShare has rate-limited to protect against attempts to guess the URL.

### *What OnionShare doesn't protect against*

- Communicating the OnionShare URL might not be secure. The sender is responsible for securely communicating the OnionShare URL with the recipient. If they send it insecurely (such as through an email message, and their email is being monitored by an attacker), the eavesdropper will learn that they're sending files with OnionShare. If the attacker loads the URL in Tor Browser before the legitimate recipient gets to it, they can download the files being shared. If this risk fits the sender's threat model, they must find a more secure way to communicate the URL, such as in an encrypted email, chat, or voice call. This isn't necessary in cases where the files being shared aren't secret.
- Communicating the OnionShare URL might not be anonymous. While OnionShare and Tor Browser allow for anonymously sending files, if the sender wishes to remain anonymous they must take extra steps to ensure this while communicating the OnionShare URL. For example, they might need to use Tor to create a new anonymous email or chat account, and only access it over Tor, to use for sharing the URL. This isn't necessary in cases where there's no need to protect anonymity, such as coworkers who know each other sharing work documents.

## Leaking To Journalists Using Tor (SecureDrop)

Most major newspapers and media outlets provide a software service called "SecureDrop" for people to leak information to in a way that strenuously protects the identity of the leaker. In short they run a Tor Hidden Service.

For example if you go to washingtonpost.com/securedrop they advise you to instead open Tor Browser and go to vbmwh445kf3fs2v4.onion Other examples are The Intercept (firstlook.org/theintercept/securedrop) with their hidden service: y6xjgkgwj47us5ca.onion And Lucy Parsons Labs (ucyparsonslabs.com/securedrop) with their hidden service: qn4qfeeslglmwxgb.onion

A larger list of newspapers and organization can be found by going to securedrop.org/directory

### Ricochet and Torbirdy

Ricochet is a cross-platform chat program that uses Tor, Ricochet uses the Tor network to reach your contacts without relying on messaging servers. It creates a hidden service, which is used to rendezvous with your contacts without revealing your location or IP address. Instead of a username, you get a unique address that looks like ricochet:rs7ce36jsj24ogfw. Other Ricochet users can use this address to send a contact request - asking to be added to your contacts list. Ricochet is likewise a tool in Beta development and *is not recommended for anything serious whatsoever*.

TorBirdy is an extension for Mozilla Thunderbird that configures it to make connections over the Tor network. Think of it as Torbutton for Thunderbird. Unfortunately while this is necessary technology for many activists, TorBirdy is poory maintained and has known issues.

## Anonymous Email Addresses & Burner Phones

It's not trivial to set up a truly anonymous and persistent email account with popular providers of free email. If it was too easy they'd face problems with spammers automatedly creating infinite email accounts. But also, as more and more of our lives depend on accounts registered under an email address there's legal pressure on companies to not just hand them out without attaching something tied to you.

Many providers will either block Tor users from starting an account or they will require you to tie the account to a phone number. Sometimes they will merely ask for another email account, in which case you can use one of the many free online services that provide disposable temporary email accounts. However the disposable email account will eventually be unreachable and it may in some cases arise that your new email provider will lock you out of your account and say that they've sent an email to the other address you provided (which, being temporary, is then no longer accessible).

There are some apps that promise the ability to create temporary phone numbers but we largely don't trust them, as they seem to keep identifying information like your phone's actual number on file, tying it to the burner number.

If you need an anonymous phone number we recommend buying a cheap prepaid burner phone. Pay with cash and activate it without using a phone or internet connection tied to you. Activation can be a problem as some prepaid vendors sporadically block Tor traffic, but you can use public wifi at a cafe, public internet (e.g. at a library), or even one of the few remaining public phones. However!

Note that even if a burner phone is not tied to your name, internet, or finances, you may still be identifiable by *where* you use it. It can be a good practice to turn off your burner phone when you're not using it (ideally also taking out the battery if it allows it). In the past activists have avoided raids by using burner phones away from their homes and then turning them off and hiding the phone somewhere (like in a ziplock bag under some rocks), before returning home. Remember that even a burner phone is a tracking device and try not to tie it to yourself. It is advisable to not bring your personal phone with you while you use the burner phone, so as to avoid showing that the two move as one.

# GPG email

PGP (essentially the same thing as "GPG" and "GnuPG") is a small bit of software that constituted the first major encrpytion program on the early internet. Today it's most commonly used to encrypt emails, and most often with the program "Mozilla Thunderbird" and a plugin "Enigmail".

GPG is an ancient and somewhat hard to use tool. Many users find it very confusing first use. It has engendered a lot of hate, most of it entirely deserved. Nevertheless GPG is an unparalleled and critical tool and activists who avoid learning it often severely hurt themselves as a consequence. We strongly advise you to use it.

GPG is a frequently superior encryption tool for a few reasons:

1. It runs exclusively on your computer rather than your phone and computers are more securable than phones are.
2. It does not require the other person you're communicating with to be online at the same time as you are.
3. It doesn't depend on a single centralized system, but runs on a preexisting and relatively decentralized network (email).
4. It's relatively easy to set up anonymous email accounts and attach unique GPG keys to those accounts.
5. It provides strong and clear ways to authenticate the people you're encrypting to.
6. It doesn't require you to constantly trust a provider every time you open it.

Signal fails to provide 1, 3, and 4. Other common encrypted messengers usually fail to provide 2 and often 5. Most "encrypted email" programs fail on 6. In practical experience this is often catastrophic for activists. When shit goes down and you need to send encrypted messages to others from something not tied to your or their identities (or revealing your location) you don't have the capacity to sit and wait online for both of you to come online at the same time. How do you even communicate to them *when* they should come online? We've watched this sort of situation prove catastrophic for activists who'd refused to learn GPG or forgotten. Similarly while the people who keep Signal afloat are heroes, it could easily be shut down at any point in a government crackdown.

We will teach you to use GPG with Mozilla Thunderbird, an email client program that performs a similar function to Outlook. You may have your own favorite email software program (or use a web mail service like Gmail or Outlook.com). This guide won't tell you how to use GPG with these programs. We are are still generally skeptical for various reasons of those alternative GPG programs.

In addition to installing the software needed to use it, you will also need to create a private key, which you will keep private. The private key is what you will use to decrypt emails sent to you, and to digitally sign emails that you send to show they truly came from you. Finally, you'll learn how to distribute your public key—a small chunk of information that others will need to know before they can send you encrypted mail, and that they can use to verify emails you send.

## Installation

### *GPG*

GPG is the core bit of software that does the encryption, but provides no frills or real interface. We will need it so that the other programs can interact with it.

If you're using Windows or a Mac you will first need to install GPG (also called GnuPG). If you are running Linux GPG may already be installed on your computer, but it may also be an outdated version. You can get GPG on Windows and Mac by going to gnupg.org/download/ and looking under the binary releases, download and then run the installer. On linux you want "gpg2" and you can install with "sudo apt-get install gnupg2" on common variants of Linux like Ubuntu and Mint.

### *Thunderbird*

Thunderbird is an application that allows you to access your email with it directly, without using a web browser. Older people might have already used Microsoft Outlook, a similar program.

You can download and install thunderbird from mozilla.org/en-US/thunderbird/

Most versions of Linux have thunderbird already installed.

When Mozilla Thunderbird launches for the first time, you will be asked whether you would like a new email address. Click the “Skip this and use my existing email” button. Now you will configure Mozilla Thunderbird to be able to receive and send email. If you are used to only reading and sending email through things like riseup.net, gmail.com, outlook.com, or yahoo.com, Mozilla Thunderbird will be a new experience, but it isn't that different overall.

You now have a choice as to what email account you'd like to connect to Thunderbird. Like most people you probably have several, and some may be more or less secure than others. For the purposes of this tutorial we advise you to connect to an email account that is tied to the identity you use most when on your computer -- not an email that you use only for secret activist work, for example. There will be opportunities later to set up PGP for that work.

Enter your name, your email address, and the password to your email account. Mozilla doesn't have access to your password or your email account, although it will be saved on your computer for the program itself to regularly use. Click the “Continue” button.

In many cases Mozilla Thunderbird will automatically detect the necessary settings.

In some cases Mozilla Thunderbird doesn't have complete information and you'll need to enter it yourself. Here is an example of the instructions Google provides for Gmail:

> Incoming Mail (IMAP) Server - Requires SSL imap.gmail.com Port: 993 Requires SSL: Yes Outgoing Mail (SMTP) Server - Requires TLS smtp.gmail.com Port: 465 or 587 Requires SSL: Yes Requires authentication: Yes Use same settings as incoming mail server Full Name or Display Name: [your name or pseudonym] Account Name or User Name: your full Gmail address (username@gmail.com). Google Apps users, please enter username@your_domain.com Email address: your full Gmail address (username@gmail.com) Google Apps users, please enter username@your_domain.com Password: your Gmail password

The difference between POP and IMAP is basically the difference between pulling down emails from the server (on Google's drives) to your computer alone, or leaving them up and just

reading them through Thunderbird. If you want to be able to check your email via webmail, or ever recover your emails if something happens to your computer, then you want IMAP. There are activists who use POP, under the premise that it's better to pull them down to their local computer with full disk encryption rather than leave their emails online and accessible by warrant. However since basically all email is insecure to begin with when not encrypted end-to-end as with PGP, a committed adversary like the NSA could still read any non PGP-encrypted emails. Indeed it's long been known that the NSA and other countries keep massive archives of all emails ever sent over the internet. Long story short: IMAP generally fits the needs of users better, but some technologically adept users may choose POP for some marginal benefits.

If you use two-factor authentication with Google (and depending on your threat model you probably should!) you cannot use your standard Gmail password with Thunderbird. Instead, you will need to create a new application-specific password for Thunderbird to access your Gmail account. See Google's own guide for doing this. support.google.com/mail/answer/185833

When all the information is entered correctly, click the “Done” button. Mozilla Thunderbird will start downloading copies of your email to your computer. Try sending a test email to your friends. (This email will not be encrypted! We still need one more piece!)

## Enigmail

*Note from the zine authors : some of the instructions below are out of date, as the PGP functionality of the Enigmail plugin has migrated into Thunderbird as of 2020. You should look for more recent guides with updated instructions.*

Enigmail is installed in a different way from Mozilla Thunderbird and GnuPG. As mentioned before, Enigmail is an Add-on for Mozilla Thunderbird. Click the “Menu button,” also called the Hamburger button and select “Add Ons.” Enter "Enigmail" into the Add-on search field to look for Enigmail on the Mozilla Add-on site. Enigmail will be the first option. Click the "Install" button. After the Enigmail add-on is installed Mozilla Thunderbird will ask to restart the browser to activate Enigmail. Click the “Restart Now” button and Mozilla Thunderbird will restart.

When Mozilla Thunderbird restarts an additional window will open up that will start the process of setting up the Enigmail add-on. Keep the “Start setup now” button selected and click the “Next” button. We believe Enigmail’s “standard configuration” option to be a good choice. Click the “Next” button.

### Creating a Public Key and Private Key

*Note from the zine authors : some of the instructions below are out of date, as the PGP functionality of the Enigmail plugin has migrated into Thunderbird as of 2020. You should look for more recent guides with updated instructions.*

(If you ever get lost and want to return to this, or need to set up a public/private key on a different email account, you can go to the "Enigmail" dropdown in Thunderbird and click the "Startup Wizard".)

Unless you have already configured more than one email account, Enigmail will choose the email account you've already configured. The first thing you'll need to do is come up with a strong passphrase for your private key.

You are generating a public/private key pair, you want to share the public key with others, but keep the private key secret. The private key enables the decryption of every message encrypted

with your public key, so it's critical. If someone breaks into your computer you want your private key *itself* to be encrypted so they can't use it. Thus we're creating a password to encrypt your private key. Note that the password itself won't work to decrypt your encrypted emails without the private key, you will need both. Choose a good passphrase, at least five random dictionary words.

When you are finished entering your password and click the "Next" button you will be shown a screen with a slow progress bar, as it takes a while for your keys to be first generated.

Note: Your key will expire at a certain time; when that happens, other people will stop using it entirely for new emails to you, though you might not get any warning or explanation about why. So, you may want to mark your calendar and pay attention to this issue a month or so before the expiration date. It's possible to extend the lifetime of an existing key by giving it a new, later expiration date, or it's possible to replace it with a new key by creating a fresh one from scratch. Both processes might require contacting people who email you and making sure that they get the updated key; current software isn't very good at automating this. So make a reminder for yourself; if you don't think you'll be able to manage it, you can consider setting the key so that it never expires, though in that case other people might try to use it when contacting you far in the future even if you no longer have the private key or no longer use PGP.

Enigmail will generate the key and when it is complete, a small window will open asking you to generate a revocation certificate. This revocation certificate is important to have as it allows you to make the private key and public key invalid. It is important to note that merely deleting the private key does not invalidate the public key and may lead others to sending you encrypted mail that you can't decrypt. Click the “Generate Certificate” button. First you will be asked to provide the passphrase you used when you created the PGP key. Click the “OK” button. A window will open to provide you a place to save the revocation certificate. While you can save the file to your computer we recommend saving the file to a USB drive that you are using for nothing else and storing the drive in a safe space. We also recommend removing the revocation certificate from the computer with the keys, just to avoid unintentional revocation. Even better, save this file on an encrypted disk. Choose the location where you are saving this file and click the “Save” button.

Finally, you are done with generating the private key and public key. Click the “Finish” button.

## Getting Other People's Public Keys

*Note from the zine authors : some of the instructions below are out of date, as the PGP functionality of the Enigmail plugin has migrated into Thunderbird as of 2020. You should look for more recent guides with updated instructions.*

You might get a public key sent to you as an email attachment. If so you can just click on the "Import Key" button.

It's possible that you get a public key by downloading it from a website or someone might have sent it through chat software. Open the Enigmail Key Manager and click on the “File” menu. Select “Import Keys from File.”

It's also possible to get a public key from copying the key as a raw block of text (eg on someone's website). First select the key itself (any raw text beginning with "-----BEGIN PGP PUBLIC KEY BLOCK-----" and ending with "-----END PGP PUBLIC KEY BLOCK-----") and copy it. Open the Enigmail Key Manager and click on the "Edit" menu. Select "Import Keys from

Clipboard." Beware that if you select slightly the wrong set of text, like miss a "-" or highlight an extra space at the end then this import will fail.

It's also possible to get a public key by downloading it directly from a URL. Open the Enigmail Key Manager and click on the “Edit” menu. Select “Import Keys from URL.” Enter the URL. The URL can have several forms. Most often it is likely a domain name ending in a file.

The last way to get a public key is by asking a keyserver. Keyservers are servers set up by volunteers that collect and display every public key sent to them. From the Key Management interface click the “Keyserver” menu and select “Search for Keys.” A small window will pop up with a search field. You can search by a complete email address, a partial email address, a name, or a Key ID (the last 8 or 16 characters of a key's fingerprint, more on that in a moment). A larger window will pop up with many options. You might notice some keys that are italicized and grayed out, these are keys that have either been revoked or expired on their own. Select the key you want from this list (if exists).

## Sending Your Public Key

*Note from the zine authors : some of the instructions below are out of date, as the PGP functionality of the Enigmail plugin has migrated into Thunderbird as of 2020. You should look for more recent guides with updated instructions.*

You can send your public key to people several ways. You can upload it to a keyserver by right clicking on it inside "Key Management" and selecting "Upload Public Keys to Keyserver". (Note that it may take a while for the keyserver to process and start giving away your key when people search for it).

You can copy-paste your key (right click on the key and select "Export Public Key to Clipboard") and post it somewhere online. Facebook even provides a place in your facebook profile for this!

Or you can send an email to someone and attach your public key. Fill in an address and a subject, perhaps something my “my public key,” click the “Attach My Public Key” button. If you have already imported a PGP key for the person you are sending the PGP key to, the Lock icon in the Enigmail bar will be highlighted. As an additional option, you can also click the Pencil icon to sign the email, giving the recipient a way to verify the authenticity of the email later. A window will pop open asking you if you forgot to add an attachment. This is a bug in the interaction between Enigmail and Mozilla Thunderbird, but don’t worry, your public key will be attached. Click the “No, Send Now” button.

## Authenticating!

*Note from the zine authors : some of the instructions below are out of date, as the PGP functionality of the Enigmail plugin has migrated into Thunderbird as of 2020. You should look for more recent guides with updated instructions.*

Just because you got a public key id for someone doesn't mean you truly got it *from* them and that it's the correct public key. As always with encryption it's critical that you likewise authenticate that the key you have from them is the key they sent. Note that in all of the above steps relaying public keys a malicious adversary could have intercepted and replaced the public key being sent. Also *anyone* can upload keys claiming to be from *anyone* to public key servers.

The way to do this with PGP is to check the "key fingerprint". Go to Enigmail's "Key Management" and right click on the key you'd like to inspect, then select "Key Properties".

You want to convey your fingerprint in person to other people and get fingerprints from them. One way to do this is to physically write down and hand out the fingerprint to the key associated with your relevant identity/email. If the fingerprint someone give you matches the fingerprint of the key you have for them then you can trust encrypted messages to and from them.

You can sign their key by right clicking it in "Key Management" and selecting "Sign Key". A window will pop up with their fingerprint and they key of yours to be used for signing.

It's possible to upload your signature to a keyserver or send it to someone else who has authenticated your key. This will enable someone to trust a key they get for someone else, without actually meeting them in person. If Sam has checked fingerprints with Julie and Julie has checked fingerprints with Charles, then if Charles gets a copy of Sam's public key and sees that Julie has signed it, he can trust it (insofar as he trusts Julie to have correctly and honestly authenticated).

However note that putting this information online publicly can help reveal your social network. It's polite and good security to ask before sharing with the world that you've signed someone's public key. Once you've gotten consent you can right click on their key (after you've signed it) and select "Upload Public Keys to Keyserver". This will update the copy of the key the keyserver has to add your signature. (It's only possible to add signatures, not remove them.)

## Sending Encrypted Emails

*Note from the zine authors : some of the instructions below are out of date, as the PGP functionality of the Enigmail plugin has migrated into Thunderbird as of 2020. You should look for more recent guides with updated instructions.*

Once you and the person you're communicating with have both gotten PGP runing and have exchanged and authenticated public keys it's time to send your first encrypted email!

In the main Mozilla Thunderbird window click the “Write” button. A new window will open. Write your message, and enter a recipient. For this test, select a recipient whose public key you already have. Enigmail will detect this and automatically encrypt the email (the padlock icon should turn yellow). If you don't have a key for someone you can click the padlock icon to turn it yellow and before the email sends you will be prompted to search for their key on public key servers.

NOTE THAT THE SUBJECT LINE WON'T BE ENCRYPTED.

## Receiving Encrypted Emails

*Note from the zine authors : some of the instructions below are out of date, as the PGP functionality of the Enigmail plugin has migrated into Thunderbird as of 2020. You should look for more recent guides with updated instructions.*

When you open Thunderbird to find new mail and click on it, you will get a prompt to enter your password to your private key if the email is encrypted. Thunderbird will remember your password for a few minutes so that you can read multiple encrypted emails without having to constantly enter your password.

### Signing Your Emails

GPG provides the capacity to authenticate emails without encrypting them. What happens is it takes the text of your email and uses your private key to create a cryptographic signature that someone with your private key can see would only be creatable by someone with your private key and that text.

This enables you to send unencrypted emails that unambiguously came from you and not someone else. Someone will still be able to read them without PGP, but if they later get PGP and your public key they will be able to confirm those emails came from you and were not altered.

## Signing Pseudonymous Communiques

One major use that GPG can have for activists is to certify which actions were actually committed by a group. (To avoid being falsely accused of other actions.)

You can sign the text of communiques using GPG without using email. And there are tools like gnu privacy assistant and cleopatra on mac that provide graphical interfaces for GPG to generate keys and manually sign (or encrypt) texts you want to be released. This provides your activist group with strong authentication of their communiques or statements.

## Proving You Did Or Knew Something

It's sometimes the case that you want to prove that you or your organization knew about something before it officially happened, without actually telling people beforehand. Your group may be secretive and you don't want to give press conferences, but you also don't want other people to be able to undertake actions and blame them on you or falsely take credit for actions your secretive group took.

To do this you can use a cryptographic tool to create a "hash" of a message, publish the hash to a site that adds timestamps, and then later publish the message itself. So for example you could post to twitter or a pastebin style site the hash "22a4e73e917bcb383466ddad4181b2f49c17827d0ddb907c853cccc23c7d8a56" a week before your bakesale and then after you've done the bakesale post "We, the housewives of Toledo, claim responsibility for the bakesale that will happen tuesday".

Via command line on a mac or linux computer this hashing would be done like so:

> echo -n "We, the housewives of Toledo, claim responsibility for the bakesale that will happen tuesday" | sha256sum

This will return the hash:

> 22a4e73e917bcb383466ddad4181b2f49c17827d0ddb907c853cccc23c7d8a56 -

A brief explanation of what's happening here: The first word "echo" tells linux to run the program "echo" with everything that follows that word until the "|" character (not the letter "l" or "i" but a distinct character called "bar" or "pipe" that is found on your keyboard typically on the same button as the "" character (requiring you to press shift to get the bar instead). The "-n" is a setting for the program "echo" that tells it not to add a newline to your text, and everything within the quote marks ("We, the housewives of Toledo, claim responsibility for the bakesale that will

happen tuesday") is what "echo" will run. Echo is a simple program that just returns what you put into it. We then use the "|" character to assign the output of "echo" as the input to the program "sha256sum" which is what creates the hash.

The trick here is that the hash returned by sha256sum isn't something that can be easily reversed. It's quite hard to look at a hash and figure out what message creates it. But once you have the message you can quickly verify that it indeed was what created that hash.

This is a great way to make statements in advance of underground actions and prove what your org did or didn't do. If the action doesn't work you don't have to reveal the message that you were hashing. And if the authoritarian regime that has outlawed your anonymous bakesales is in danger of hightening security ever time your org releases a hash you can fuck with them by sometimes posting hashes of knock-knock jokes.

## VPNs

There was a long time in which Tor was effectively blocked in China. The Chinese government had complete control over traffic entering and exiting the country and carefully monitored all packets and connections for anything that looked remotely close to a connection to the Tor network and blocked them. Eventually the Tor project managed to create steganographically hidden connections that would allow Chinese dissidents who'd been passed secret entry servers into the Tor network to connect without being blocked. But in the meantime Chinese users continued to find ways around The Great Firewall. Most of the time they used VPNs, despite them being lower quality. Many were detected, many were backdoored, but some were not.

While VPNs don't provide real anonymity they can be useful.

If you're going to torrent something claimed as "intellectual property" you want to hide your IP Address, but your heavy traffic would slow down the Tor network and (despite the similar names) Tor isn't built for torrents.

If you're trying to hide your location but you want to check your bank account attempting to login via Tor will get your account frozen (requiring you to visit a branch in person!). Websites know when a user is visiting via Tor. They don't necessarily always know when you're visting via a VPN.

It's impossible to make any "best VPN" list because the whole utility of VPNs is their relative obscurity. However free VPNs are generally suspect. The best way to determine a good VPN is via word-of-mouth. Ideally if that VPN starts betraying their clients to the government folks will spread the warning.

If you don't have the money for a VPN we can however recommend using Riseup.net's free VPN service. It's better than nothing! Currently their Bitmask VPN only works for Linux and Android, but they promise versions for Mac and Windows soon...

## Keepassx

Keepass is an open-source program that allows you to create encrypted databases of account information (including passwords), it's useful in a wide variety of contexts for activists and is *highly* recommended.

KeePassX saves many different information e.g. user names, passwords, urls, attachments and comments in one single database. For a better management user-defined titles and icons can be specified for each single entry. Furthermore the entries are sorted in groups, which are customizable as well. KeePassX also offers a password generator.

You can download and install KeepassX for Windows and Mac at keepassx.org/downloads and it's available in the repositories of almost all linux variants ("sudo apt-get install keepassx" on Ubuntu).

Besides keeping track of your various accounts and passwords (and thus enabling you to use different lengthy secure passwords on each account), you can also create a KeepassX database to securely share group resources (like your organization's social media accounts). It can also be used to securely organize information you're researching with other people.

And KeepassX can be used as a secure way to backup critical information for yourself or for other people you trust. You can leave a copy with certain accounts to people you expect to do jail support for you (giving them the password only once you're imprisoned), or you can use it as a (not legally binding) Last Will And Testament, giving one person you trust the database and the other (distant) person the password, trusting both of them to not seek the other out unless you've died.

Remember to use *VERY* long and secure passwords for your KeepassX databases because it's a password that secures everything else. We recommend seven *random* dictionary words and some characters or numbers. Yes, really. Look you only have to memorize that single password, once you have it memorized you can create infinite different really long passwords for every other account you use and just have keepassx remember them for you.

KeepassX also provides some limited security against keylogger malware, because if you just copy-paste the password from KeepassX into your browser there's nothing to record from your keyboard. KeepassX also removes the copy-pasted passwords from your computer's "clipboard" so someone can't just click "paste" a minute later and recover it.

Note that KeepassX by default keeps a history log of every password you've saved for various accounts. You can delete these by going to the history folder. This is important if you're making a copy of a KeepassX database but deleting a few accounts before you share it with someone. You will also need to delete the entries under the history tab. Probably better to delete all the history entries before sharing, just to be safe.

## Generating Passphrases

When creating passphrases people often pick some phrase from pop culture — favorite lyrics from a song or a favorite line from a movie or book — and slightly mangle it by changing some capitalization or adding some punctuation or using the first letter of each word from this phrase. Some of these passphrases might seem good and entirely unguessable, but it’s easy to underestimate the capabilities of those invested in guessing passphrases.

It would not be that difficult for your adversary to take the lyrics from every song ever written, the scripts from every movie and TV show, the text from every book ever digitized and every page on Wikipedia, in every language, and use that as a basis for their guess list. Will your passphrase still survive?

If you created your passphrase by just trying to think of a good one, there’s a pretty high chance that it’s not good enough to stand up against the might of a spy agency. For example, you might come up with “To be or not to be/ THAT is the Question?” If so, I can guarantee that you are not the first person to use this slightly mangled classic Shakespeare quote as your passphrase, and attackers know this.

The reason the Shakespeare quote sucks as a passphrase is that it lacks something called entropy. You can think of entropy as randomness, and it’s one of the most important concepts in cryptography. It turns out humans are a species of patterns, and they are incapable of doing anything in a truly random fashion.

Even if you don’t use a quote, but instead make up a phrase off the top of your head, your phrase will still be far from random because language is predictable. As one research paper on the topic states, “users aren’t able to choose phrases made of completely random words, but are influenced by the probability of a phrase occurring in natural language,” meaning that user-chosen passphrases don’t contain as much entropy as you think they might. Your brain tends to continue using common idioms and rules of grammar that reduce randomness. For example, it disproportionately decides to follow an adverb with a verb and vice versa, or, to cite one actual case from the aforementioned research paper, to put the word “fest” after the word “sausage.”

Passphrases that come from pop culture, facts about your life, or anything that comes directly from your mind are much weaker than passphrases that are imbued with actual entropy, collected from nature.

We STRONGLY recommend using a sequence of six or seven random dictionary words (nothing like "the"). Opening a dictionary at random and selecting a word and then opening it again would work. Or you could everyone in your group think up random words, put them in a hat and select seven randomly selected from them. NEVER say your passphrase or the words aloud. If someone does then that word is invalid.

At one trillion guesses per second — per Edward Snowden’s January 2013 warning — it would take an average of 27 million years to guess the passphrase “bolt vat frisky fob land hazy rigid”. A five-word passphrase, in contrast, would be cracked in just under six months and a six-word passphrase would take 3,505 years, on average, at a trillion guesses a second. Keeping in mind that computers are constantly getting more powerful, and before long 1 trillion guesses a second might start looking slow, so it’s good to give your passphrases some security breathing room.

Almost everyone upon recieving this advice rolls their eyes in exasperation at how extreme it sounds, but we cannot emphasize this enough: we're not being extreme, we're being absolutely practical. Just because your passphrase "ittakesstr3ngthtob3g3ntleandkind" hasn't involved you getting obviously hacked yet, doeesn't prove anything. Just because some website's automated password checker said your password was "strong" doesn't mean it actually is. We know what the fuck we're talking about. Your friend and that article you read once are wrong.

You CAN memorize the seven randomly selected words even though there's no ostensible emotional connection for you. Work at it. You can even write notes about the password down and keep it in your wallet at the start if you're not in immediate danger of being raided or detained. Train yourself on it repeatedly for a couple days and then burn any papers you wrote it down on.

# Full Disc Encryption

You know that password you use to log into your phone or computer? It's almost worthless. It may stop your friend from reading your emails when you go to the bathroom but that's about it. If someone steals your computer or an officer just takes it out of your bag for a brief period they can read the contents of your harddrive and copy basically everything off of it. And this even includes material that you've "deleted".

That email you wrote to a friend years ago? That sexy picture you sent to your ex back when you were dating? That one website you visited accidentally? All visible. The moment someone gets physical access to your computer or phone every single aspect of your digital life on it becomes visible in its entirety. All of it twistable into "evidence". All of it useable for blackmail. All of it usable to fool your friends into thinking someone else is you.

It may be the case that the programs or operating system that you use may encrypt some select files, passwords, keys, but why pin your hopes on that? Why not just Encrypt Everything!?

The use of full disk encryption on all ones' devices and drives should be considered absolutely required for all activists and dissidents of any stripe. Thankfully it's dead simple for most people to enable full disk encryption from within the default options of their operating systems.

Please note that you're picking something of a master passphrase that you will need to enter every time you start your computer. Since everything hinges upon it you should pick a long passphrase, with over 25 characters, hopefully with every type (lowercase, uppercase, numbers, symbols). If you forget this passphrase you will be incapable of recovering anything from your computer. It's adequate to choose a passphrase that's a full sentence or two, although don't use any popular lyrics or anything already publicly associated with you. Feel free -- if you're not in immediate danger -- to write it down and refer to it for a few weeks until you've completely memorized it.

Most major Linux distros now offer full disk encryption as a checkbox when first installing the operating system. For example when installing the popular Linux variant "Ubuntu" you just check, “Encrypt the new Ubuntu installation for security” and enter the passphrase twice. It's suggested that at this point you also check "Overwrite empty disk space."

To enable full disk encryption on modern Windows computers, first sign in with an administrator account. Go to Start, enter "encryption", and select "Change device encryption" settings from the list of results. Select "Manage BitLocker", select "Turn on BitLocker", and then follow the instructions.

To enable full disk encryption on a Mac, go to "System Preferences" from the Apple menu, then click Security & Privacy. Click the FileVault tab. Click the Lock Locked button, then enter an administrator name and password. Click Turn On FileVault and follow the instructions. When FileVault setup is complete, your Mac restarts and asks you to log in with your account password. Your password unlocks your disk and allows your Mac to finish starting up.

To enable full disk encryption on an iphone running any version of the operating system between iOS 4 and iOS 7, you go to the General settings, and choose Passcode (or iTouch & Passcode). As for iOS 8-9, Passcode (or “Touch ID & Passcode”) has its own section in the Settings app. Follow the prompts to create a passcode. You should set the “Require passcode” option to “Immediately,” so that your device isn't unlocked when you are not using it. Disable Simple Passcode so that you can use a code that's longer than 4 digits. If you choose a

passcode that's all-numeric, you will still get a numeric keypad when you need to unlock your phone, which may be easier than typing a set of letters and symbols on a tiny virtual keyboard. You should still try to keep your passcode long even though Apple's hardware is designed to slow down password-cracking tools. Try creating a passcode that is more than 6 digits. Once you've set a passcode, scroll down to the bottom of the Passcode settings page. You should see a message that says “Data protection enabled.”

To enable full disk encryption on an Android head into the Settings menu and tapping on “Security,” although the wording may be slightly different. If your device is already encrypted, it will show up here. Some devices will also allow SD card contents to be encrypted, but by default Android just encrypts on-board storage. If the device isn’t encrypted, you can start the process by tapping the “Encrypt phone” option.

**PLEASE NOTE :**

While full-disk encryption provides security against someone physically accessing, stealing or confiscating your turned-off devices, *it does not help if someone gets physical access to your device while it's turned on.* Once you power on your phone or computer and enter your passphrase it continues to store in RAM the key necessary unencrypt the drive. Your devices needs to be turned entirely off (shutting your laptop's screen won't help) for the encryption to be truly unreadable.

Nor does full-disk encryption provide security against someone who repeatedly physically accesses your device without your knowledge. While they would not be able to decrypt your drives the first time they get your device they could nevertheless add a physical bug to your computer that logs your password when you turn on the device and unencrypt it, or even injects malware once it's turned on. If someone you don't trust gets physical access to your phone or computer you consider never using it again. Or at least know that the next time you turn it on and log in you could be compromising everything inside. We recognize that it's often financially infeasible to tell activists to simply scrap any computer seized by the police, but one thing you could do is simply overwrite the encrpyted drive with a fresh operating system and then use it for necessary life tasks, while treating it as completely compromised.

Nor does full-disk encryption do anything to protect you against malware and hacking. Someone might still be able to attack your computer once it's turned on and connected to a network. If you use full disk encryption but then open a PDF file from a weird email they could put a backdoor on your computer and have it transmit the contents of your drive, without having to ever have physical access.

To encrypt USBs or individual files or folders we enthusiastically endorse Veracrypt, an open source program that has passed multiple serious security reviews and runs on all major operating systems. Veracrypt can be downloaded here: veracrypt.codeplex.com

To make Veracrypt work Mac users will also need to download and install FUSE located here: osxfuse.github.io

Veracrypt can be used instead of Bitlocker to encrypt entire Windows machines. To encrypt a system partition or entire system drive, select System > Encrypt System Partition/Drive and then follow the instructions in the wizard. To decrypt a system partition/drive, select System > Permanently Decrypt System Partition/Drive. Note that VeraCrypt can encrypt an existing unencrypted system partition/drive in-place while the operating system is running (while the system is being encrypted, you can use your computer as usual without any restrictions).

Likewise, a VeraCrypt-encrypted system partition/drive can be decrypted in-place while the operating system is running. You can interrupt the process of encryption or decryption anytime, leave the partition/drive partially unencrypted, restart or shut down the computer, and then resume the process, which will continue from the point it was stopped.

But Veracrypt's best function is that it can be used to hide multiple operating systems.

It may happen that you are forced by somebody to decrypt the operating system. There are many situations where you cannot refuse to do so (for example, due to extortion or torture). VeraCrypt allows you to create a hidden operating system whose existence should be impossible to prove (provided that certain guidelines are followed). Thus, you can "unencrypt" your computer upon demand and show the cops nothing but adorable kitten pictures, while your real operating system and files remain hidden in a second (or third...) operating system on the same device.

The VeraCrypt wizard helps you create a second encrypted operating system, so-called decoy operating system, during the process of creation of a hidden operating system. A decoy operating system must not contain any sensitive files. Its existence is not secret (it is not installed in a hidden volume). The password for the decoy operating system can be safely revealed to anyone forcing you to disclose your pre-boot authentication password. There will be two pre-boot authentication passwords — one for the hidden system and the other for the decoy system. If you want to start the hidden system, you simply enter the password for the hidden system in the VeraCrypt Boot Loader screen (which appears after you turn on or restart your computer). Likewise, if you want to start the decoy system (for example, when asked to do so by an adversary), you just enter the password for the decoy system in the VeraCrypt Boot Loader screen.

## Buying Devices

Always purchase your computers or devices with cash in person.

If you can't manage that then get a non-political friend to place the online order for you and pay using their card.

We know from the NSA papers leaked by Snowden that as of 2013 the US government already had a rather automated procedure by which they bug and backdoored the new computers and phones purchased online by certain dissidents. Basically if you use your credit card online to order a device they'd intercept or reroute the package momentarily, insert a bug, and then repackage it and send it along to you. We expect that they've only scaled this operation up.

Another problem are backdoors at the point of production. One of the biggest concerns has long been that the US and other governments would oblige chip producers (either publicly or secretly) to install backdoors in everyone's computers. We haven't seen indication that this is widespread. However the US government forbids Lenovo laptops from certain government facilities because they're produced in China and could have secret backdoors.

Lenovo laptops have definitely at points come with obvious and simple spyware installed. But in fact many hackers and dissidents in the west use Lenovo computers nonetheless because it's possible to remove this spyware just by deleting the default Windows installation and installing Linux instead. The reasoning of these people is that it's better to have a computer that the chinese possibly backdoored than one the NSA has possibly backdoored.

Dealing with device security all the way from the origin is a complex topic. There are a handful of companies offering fully open source computers and phones, but nothing has gone into true mass production, many of these devices are still not completely friendly to novice users or capable of running applications many users consider important. Still they do constitute a gold standard and a major hope for the future. However what devices are being offered by these small projects tends to change rapidly so we're not going to recommend any here because our recommendations might become quickly outdated.

## Deleting Data

Most of us think that a file on our computer is deleted once we put the file in our computer's trash folder and empty the trash; in reality, deleting the file does not completely erase it. When one does this, the computer just makes the file invisible to the user and marks the part of the disk that the file was stored on as "available"—meaning that your operating system can now write over the file with new data. Therefore, it may be weeks, months, or even years before that file is overwritten with a new one. Until this happens, that “deleted” file is still on your disk; it’s just invisible to normal operations. And with a little work and the right tools (such as “undelete” software or forensic methods), you can even still retrieve the “deleted” file. The bottom line is that computers normally don't "delete" files; they just allow the space those files take up to be overwritten by something else some time in the future.

The best way to delete a file forever, then, is to make sure it gets overwritten immediately, in a way that makes it difficult to retrieve what used to be written there. Your operating system probably already has software that can do this for you—software that can overwrite all of the "empty" space on your disk with gibberish and thereby protect the confidentiality of deleted data.

Note that securely deleting data from solid state drives (SSDs), USB flash drives, and SD cards is very hard! The instructions below apply only to traditional disk drives, and not to SSDs, which are becoming standard in modern laptops, USB keys/USB thumb drives, or SD cards/flash memory cards. As a result your best bet in terms of protection is to use encryption—that way, even if the file is still on the disk, it will at least look like gibberish to anyone who gets ahold of it and can’t force you to decrypt it. At this point in time, we cannot provide a good general procedure that will definitely remove your data from an SSD.

On Windows and Linux, we currently suggest using BleachBit. BleachBit is a free/open source secure deletion tool for Windows and Linux, and is much more sophisticated than the built-in “shred.” BleachBit can be used to quickly and easily target individual files for secure deletion, or to implement periodic secure deletion policies. It is also possible to write custom file deletion instructions. Please check the documentation for further information.

Mac users are in less luck. On OS X 10.4 to 10.10, you can securely delete files by moving them to the Trash and then selecting Finder > Secure Empty Trash. The Secure Empty Trash feature was removed in OS X 10.11, because Apple felt that it could not guarantee secure deletion on the fast flash (SSD) drives that most of its modern models now use. If you use a traditional hard drive, and are comfortable with the command line, you can still use the Mac's srm command to overwrite the file.

### A Warning About the Limitations of Secure Deletion Tools

First, remember that the advice above only deletes files on the disk of the computer you’re using. None of the tools above will delete backups that were made to somewhere else on your

computer, another disk or USB drive, a “Time Machine,” on an email server, or in the cloud. In order to securely delete a file, you must delete every copy of that file, everywhere it was stored or sent. Additionally, once a file is stored in the cloud (e.g. via Dropbox or some other file-sharing service) then there’s usually no way to guarantee that it will be deleted forever.

Unfortunately, there’s also another limitation to secure deletion tools. Even if you follow the advice above and you’ve deleted all copies of a file, there is a chance that certain traces of deleted files may persist on your computer, not because the files themselves haven't been properly deleted, but because some part of the operating system or some other program keeps a deliberate record of them.

There are many ways in which this could occur, but two examples should suffice to convey the possibility. On Windows or Mac OS, a copy of Microsoft Office may retain a reference to the name of a file in the "Recent Documents" menu, even if the file has been deleted (Office might sometimes even keep temporary files containing the contents of the file). On a Linux OpenOffice may keep as many records as Microsoft Office, and a user's shell history file may contain commands that include the file's name, even though the file has been securely deleted. In practice, there may be dozens of programs that behave like this.

It's hard to know how to respond to this problem. It is safe to assume that even if a file has been securely deleted, its name will probably continue to exist for some time on your computer. Overwriting the entire disk is the only way to be 100% sure the name is gone. Some of you may be wondering, "Could I search the raw data on the disk to see if there are any copies of the data anywhere?" The answer is yes and no. Searching the disk (e.g. by using a command like grep -ab /dev/ on Linux) will tell you if the data is present in plaintext, but it won't tell you if some program has compressed or otherwise coded references to it. Also be careful that the search itself does not leave a record! The probability that the file's contents may persist is lower, but not impossible. Overwriting the entire disk and installing a fresh operating system is the only way to be 100% certain that records of a file have been erased.

## Secure Deletion When Discarding Old Hardware

If you want to finally throw a piece of hardware away or sell it on eBay, you'll want to make sure no one can retrieve your data from it. Studies have repeatedly found that computer owners usually fail to do this—hard drives are often resold chock-full of highly sensitive information. So, before selling or recycling a computer, be sure to overwrite its storage media with gibberish first. And even if you're not getting rid of it right away, if you have a computer that has reached the end of its life and is no longer in use, it's also safer to wipe the hard drive before stashing the machine in a corner or a closet. Darik's Boot and Nuke is a tool designed for this purpose, and there are a variety of tutorials on how to use it across the web (including here).

Some full-disk encryption software has the ability to destroy the master key, rendering a hard drive's encrypted contents permanently incomprehensible. Since the key is a tiny amount of data and can be destroyed almost instantaneously, this represents a much faster alternative to overwriting with software like Darik's Boot and Nuke, which can be quite time-consuming for larger drives. However, this option is only feasible if the hard drive was always encrypted. If you weren't using full-disk encryption ahead of time, you'll need to overwrite the whole drive before getting rid of it.

## Stripping Metadata

It has been the case countless times that a quasi-underground group or an activist organization facing severe repression will release a statement, a zine, or a blog post, only to inadvertantly expose the names of the people involved and the computer software they use. This can result in personal death threats, attacks, firebombings of their house, etc. Abortion rights activists for example, have to be very careful. Even a technophilic biblethumping nutter can right click on an image and read the metadata.

This is because every PDF, document, audio file, video, or image you create or save from the internet is marked with additional information. Your programs do this "to be helpful!" which is really incredibly odious. People go along with this because our society is brainwashed by the ideology of intellectual property. They just assume that everyone would want the world to know who created a file or edited it. So instead we're forced to live in a world where almost every program on your computer will mark a file with additional information about you, the user. It helps to never give your name when setting up your computer and registering a user account.

But just to be safe before posting any file publicly activist organizations should strip them of this metadata. Unfortunately there are a huge number of ways to do this and little consistency across file types and operating systems. We could easily fill this entire zine twice over giving examples and guides. You pretty much have to search online for the ways to remove metadata in your specific situation. If you're using Linux though you're in luck because the Metadata Anonymisation Toolkit (mat) comes packaged in Debian and Tails -- this is the best option for serious security concerns.

Microsoft offers a "Document Inspector" for removing "personal or sensitive information" before you share an Office file. Windows lets you view and delete metadata from a file via the Properties dialog box. The quickest way is to click Properties > Details > Remove Properties and Personal Information > "Create a copy with all possible properties removed."

Metadata even exists in physical media like on printed pamphlets. Printers are built to add tiny unique dots of almost invisible (usually light yellow) ink to what you print. These dots are arranged in patterns that link a document to a printer. This was originally added as part of a deal between the US Secret Service and printer manufacturers to help track down counterfeiters. We suggest getting a printer from a yard sale or recycling space and buying ink cartridges for it in cash.

## OTR messengers (the shit talk section)

There are many applications that attempt to provide encrypted chat on your computer. For the most part we advise against using them.

The most popular was Pidgin (in addition to its OTR plugin) and Adium on Macs, we strongly advise against using Pidgin and Adium because they both depend upon a piece of software called "libpurple." We condemn libpurple in no uncertain terms. The developers have become pariahs in the crypto community for repeatedly refusing to fix deep and dangerous vulnerabilities in their software. We'll go further than the rest of the crypto community and say, fuck those traitorous bastards and may they drown in the blood no doubt spilled as a result of their complicity with the NSA and authoritarian regimes abroad.

Another messaging app that achieved some prominence was Cryptocat. We likewise don't endorse using Cryptocat. While the core developer has recieved unfair hate, he has a terrible

track record. He focused on hyping his unaudited software without adequate warnings and exposed countless activists over several months when an update he made rendered Cryptocat's encryption moot. He's also violently hostile to anarchists and antiauthoritarian activists, and has met in person with members of state security services.

There are countless other OTR messengers but few have been both highly audited and made useable.

## Video Conferencing

Don't fucking do it.

If you're going to do it for the love of all that is holy don't ever use Skype. Never install Skype.

Skype is one of those programs that is actively and openly hostile encryption and user security. While it claims to offer some "encryption" it happily backdoors connections for local regimes. Because money is far far more important to them than activist lives and human rights. At the start of the Syrian Civil War the government quickly worked to identify all dissidents and murder them, or abduct, torture and then murder them. Skype was central to how they did this. The Syrian cops would grab a dissident, take their laptop and get their contacts list, then they'd infect all those people when they logged onto Skype and get their contact lists, and so on.

There is an open source application made by people who aren't monsters that offers encrypted video, talk, and messaging. It's called Jitsi. And while it is probably the best of the OTR messengers, it has not had sufficiently rigorous auditing in our opinion and remains, to be frank, ugly and unusable. Few users persist in using it. However at present it remains the best thing out there.

## Filming & Livestreaming

Freedom of information is a noble ideal and the ultimate aspiration of anarchists, but if you're the French Resistance fighting the Nazis you don't line up in town square with your masks off afterward.

So long as our oppressors have a giant police state there will remain a huge difference between whistleblowing and snitching. Turning your camera on the cops is a very different act than turning your cameras on protesters. Sure the police have cameras already and film protests, but they can't film every angle and it has repeatedly been the case that video recorded with the intent of exposing the cops has been used to identify and gin up charges against protesters. Under authoritarian regimes protesters and their families are regularly subject to repraisals, sometimes even paramilitary assassination. In the US extrajudicial repraisal is relatively common in communities of color. And the scope of what's considered "assault on an officer" or "resisting arrest" is absurd. Everyone's heard of protesters charged with "assaulting officers' fists with their faces" but there's instances of being charged with assault for sneezing in the vicinity of an officer or for bleeding on an officer's uniform while being beaten. Mere video evidence of someone being in the way of a charging cop can be enough to convict someone of assault on an officer.

The fact that your video evidence would expose and convict an officer in the eyes of any reasonable person doesn't actually mean it wouldn't instead expose and convict the protester being beaten. We strongly urge videographers to consider not immediately publishing their

video, but encrypting it and waiting for a lawyer to review individual selections before publication.

There is sometimes a place for livestreaming but livestreamers should exercise extreme caution to avoid incriminating protesters and themselves.

There are presently a number of applications designed to record interactions with the police and put them online, but most publish immediate, few if any use encryption and all depend on trusting third parties to hold the video. If you want to use these, and there are situations where they're called for, then we recommend doing your own research.

If you're a videographer we recommend rather than streaming to the public via some site, setting up a stream of your video to a private and encrypted server, preferably in an inconvenient foreign jurisdiction. Another option is to get an "eye-fi" SD card and set up a constant transfer from your camera to another device on you like an encrypted laptop or smartphone and delete the copy from the SD card's local storage. You could even have the smartphone or laptop then encrypt the material locally and upload it pre-encrypted.

If you don't have an external connection (because for example the police/military have shut down cellphone access or your country's internet) or don't want to run the risks of such connections then you should find other ways to regularly relay video. At protests folks have historically set up systems of runners, people who don't look like they're involved in the protest, who USBs can be handed off to discretely. Try to keep as little on you as possible as police regularly seize the data and and smash the cameras of videographers and journalists covering protests.

Finally remember that police aren't the only ones with the capacity to bug people or set up surreptitious surveillence devices. It's possible to buy very tiny cameras online (at for example supercircuits.com although look around obviously), and plant them in rooms you expect cops and politicians to enter. Of course there are many complex laws around filming people without their consent that vary by region, so read up on that sort of thing if you care about the law.

## Email, Listservs & Other Services

There are many radical activists who provide email accounts or other services on their servers.

Riseup.net is the most famous and technologically advanced. However their servers are based in the US and Canada and it's a bad idea to centralize everyone into using a single service (that a fascistic government can just shut down). Riseup has turned the user information over to the FBI in response to a warrant when Riseup deemed those users to be violation of their Terms and Service or using their services for "criminal" rather than "activist" ends.

Some other radical providers are:

**Europe**: actiu.info aktivix.org autistici.org Artikel-140.nl boum.org espiv.net Framasoft immerda.ch indivia.net nadir.org nodo50.net sindominio.net so36.net squat.net systemausfall.org systemli.org shelter.is

**North America**: espora.org flag.blackened.net hackbloc.org mutualaid.org riseup.net resist.ca tao.ca

**South America**: entodaspartes.org GuardaChuva.org Sarava.org

**West Asia**: 404team.org

Again be aware that an email provider can't truly protect your email in transit between email servers (when you send emails), or protect your email when you login and view it via the web. They may use HTTPS and that's better than nothing but HTTPS is not bulletproof and attacks are common.

## Backup Services

It's important to note that when you allow someone else to back your data up for you on their devices (because that's what "the cloud" is, just other people's computers), you're giving them a hell of a lot of trust.

We would advise against using another party's service to backup your computer. But we also recognize that many people are insistant upon using one. So if you're going to use a backup service, Spideroak is the best. Their program claims to encrypts your data on your own computer before sending to their servers and never sends the key. But Spideroak is not fully open source and this is concerning given that you're trusting them with everything.

## Collaborative Editing Software

There is at present no good option for activists to edit documents securely over the internet. Every option has downsides.

1. Use Sandstorm, an open source set of productivity tools. However this will require you to have someone you trust with technical skills set up a copy on a server for you.
2. Use etherpad, there's a publicly hosted version at pad.riseup.net. If you enter a sufficiently random and long enough URL when creating your etherpad then it might be hard for some random visitor to Riseup.net to guess your pad. Since pad.riseup.net is served over HTTPS you get some very mild level of encryption -- an observer that can't break HTTPS or give itself an X509 Certificate wouldn't be able to see precisely *what* page or URL you were visiting at pad.riseup.net, however again basically any government can trivially break HTTPS (and a lot of non government actors too).
3. Use Google docs. Ooph. Trusting Google is obviously not fucking remotely optimal. If you're gonna do this we suggest everyone using a gmail account they set up completely anonymously over Tor for the exclusive purpose of collaborative editing. And only access the google doc and that email over Tor. Still if some NSA goon ran an algorithm to find suspicious content or identify some keywords in your document they could get Google to insert malware into the page so when you access the google doc they try to hack you through your Tor browser. And obviously you can't use a Google doc with javascript turned off in Tor.
4. Use SpiderOak. If you're already using Spideroak to backup your files online with some measure of security then this is an easy choice. You can create a ShareRoom by choosing any number of folders from several of your computers. A ShareRoom may be accessed as a unique web URL or by entering a user's ShareID and RoomKey on the SpiderOak homepage – easily allowing people you invite to view your documents, pictures, movies, and so on. As you make additions or edits to the folders within a ShareRoom (no matter what computer those changes are made on), the changes are automatically updated to the ShareRoom. However there may be conflicts if multiple people are editing a file at the same time.

5. Use SpiderOak Semaphor. This is supposedly a software platform SpiderOak has developed for business teams to collaborate. It sounds promising but is still very new and insufficiently tested.
6. Use OwnCloud. If someone in your group has serious technical skills they can set up a private server and run an open source program called OwnCloud that offers collaborative document editing over encryption. However this is extremely new and unvetted software. https://owncloud.org/
7. Use Gobby. If everyone in your group has some technical skills then you can download the open source program Gobby which provides a client (an app on your computer) and a server. Everyone downloads a text editor and one person sets up a server, then folks connect and edit within the text editor app, presumably securely. However this is extremely new and unvetted software. https://gobby.github.io/

We wish there were better or more developed options. Absolutely do not use any of these if you are facing high security threats (ie likely FBI investigation). Remember that the ease of a tool can be seductive and encourage bad security decisions.

## Infection

The best way to deal with a malware attack is to avoid getting infected in the first place. This can be a difficult feat if your adversary has access to zero day attacks—attacks that exploit a previously-unknown vulnerability in a computer application. Think of your computer as a fortress; a zero day would be a hidden secret entrance that you do not know about, but which an attacker has discovered. You cannot protect yourself against a secret entrance you don’t even know exists. Governments and law enforcement agencies stockpile zero day exploits for use in targeted malware attacks. Criminals and other actors may also have access to zero day exploits that they may use to covertly install malware on your computer. But zero day exploits are expensive to buy and costly to re-use (once you use the secret tunnel to break into the fortress, it increases the chances that other people may find it). It is much more common for an attacker to trick you into installing the malware yourself.

There are many ways in which an attacker might try to trick you into installing malware on your computer. They may disguise the payload as a link to a website, a document, PDF, or even a program designed to help secure your computer. You may be targeted via email (which may look as if it’s coming from someone you know), via a message on Skype or Twitter, or even via a link posted to your Facebook page. The more targeted the attack, the more care the attacker will take in making it tempting for you to download the malware.

For example, in Syria, pro-Assad hackers targeted members of the opposition with malware hidden in fake revolutionary documents and a fake anti-hacking tool. Iranians have been targeted using malware hidden in a popular censorship-circumvention program. And in Morocco, activists were targeted with malware hidden in a document made to look as if it had been sent by an Al-Jazeera reporter, promising information about a political scandal. We know of other examples where the FBI pretended to be a reporter writing a story about the activists and looking for feedback on the attached story.

Don't open emailed files, especially PDFs. If you're going to open a document sometimes it's better to do so in a program that "sandboxes" the file (attempts to confine it from the rest of your system).

Never ever ever use Adobe products to open files recieved over the internet. In fact try not to ever use Adobe products, they're bloated and constantly top the annual lists of programs responsible for getting the most amount of people hacked. Adobe Acrobat (for reading PDFs) and Adobe Flash Player (for videos on websites), are consistently responsible for hackings.

It's very common for malicious emails to come from email addresses that look very similar to actual domains you trust. Someone might register a domain "mail-riseup.net" but that's different than "mail.riseup.net". The way that domain names work is that they form a hierarchy. The ".net" server certifies that you reach the person who registered as "riseup.net" and the person who registered as "riseup.net" can in turn tell you what computers handle things at "mail.riseup.net" or "lists.riseup.net". Note that these are seperated by periods, not dashes. Any random person can register the domain "mail-riseup.net" or "something-riseup.net" and the ".net" server will be okay with it. Similarly it's common for hackers to use misleading top-level domains like "riseup.nettery" which is totally different than "riseup.net". And lastly hackers will sometimes use domain names that look very similar but are actually misspelled.

Another thing you can do to protect your computer against malware is to always make sure you are running the latest version of your software and downloading the latest security patches. As new vulnerabilities are discovered in software, companies can fix those problems and offer that fix as a software update, but you will not reap the benefits of their work unless you install the update on your computer. It is a common belief that if you are running an unregistered copy of Windows, you cannot or should not accept security updates. This is not necessarily true. Microsoft provides sincere updates even to pirates to help stop mass outbreaks.

You can mitigate against a large class of attacks by disabling flash video in your browser and limiting or disabling javascript.

Common browser extensions like AdBlock can also help prevent common attacks from advertisements. We recommend installing extensions like NoScript or uMatrix that allow you to selectively block what scripts are run on a webpage and from what origins (unfortunately modern webpages often embed and run scripts from other websites).

While HTTPS is insecure it can be slightly harder for state adversaries to inject scripts on a site that uses HTTPS rather than plain HTTP. Using browser extensions like HTTPSeverywhere that check if a site has an HTTPS version and make sure you're using that can thus help prevent infection.

Ultimately the solution to malware will require restructuring how software is produced and checked, radically simplifying the currently complex, arcane and frequently rotting software ecosystem. We need entirely open source software, but also a culture with lots of eyes on a single piece of code, with limited languages and frameworks designed for security from the ground up. And we need comprehensive and well checked code-distribution with reproducible builds and very clear and explicit dependencies of trust.

Since we're not going to get that any time soon the best we can hope for are programs that isolate data and programs, working hard to stop a compromise of one program to spread to other programs or compromise the machine itself. The linux-based operating systems SubgraphOS and QuebesOS are the near-future of security. However they are not yet mature and we do not recommended for users without strong technical skils.

Activists facing significant security concerns have adopted to it by running TailsOS on an unwriteable USB stick, thus making it much harder for malware they might download in one

session to persist and infect them the next time they boot their computer. Other approaches have been to use Virtual Machines (a way of simulating a computer within your computer and running an operating system on that). These are not ideal and place some annoying limits on what you can really do, but they do constrain malware's capacity to infect you.

### What Should I do if I Find Malware on my Computer?

If you do find malware on your computer, unplug your computer from the Internet and stop using it immediately. Every keystroke you make may be being sent to an attacker. You may wish to take your computer to a security expert, who may be able to discover more details about the malware. If you've found the malware, removing it does not guarantee the security of your computer. Some malware gives the attacker the ability to execute arbitrary code on the infected computer—and there is no guarantee that the attacker has not installed additional malicious software while in control of your machine.

Log into a computer you believe is safe and change your passwords; every password that you typed on your computer while it was infected should now be considered to be compromised.

You may wish to reinstall the operating system on your computer in order to remove the malware. This will remove most malware, but some especially sophisticated malware may persist. If you have some idea of when your computer was infected, you may reinstall files from before that date. Reinstalling files from after the date of infection may re-infect your computer.

## Phone Interception

Here's a list of potential signs to look out for:

1. Apparent connectivity, but unable to transmit/receive or unusual delay in calls/texts (bars, but service not normal)
2. Unexpected loss of mobile signal (no bars)
3. Sudden mobile phone battery draining
4. Unexpected downgrading in cellular network (4G to 3G, 3G to 2G, etc.)
5. IMSI catcher evidence as detected by software (e.g. AIMSICD, Snoopsnitch)

## Browser

A lot of the time on your personal device you want to use something faster than Tor to browse kitten gifs and don't care about anonymity.

By now almost everyone under 50 knows not to use Internet Explorer or trust Microsoft, but there are tradeoffs between the two major browsers Mozilla Firefox and Google Chrome.

Firefox is fully open source and made by the largely positive Mozilla foundation. Chrome is made by the Dread Empire Google. But sadly Chrome is generally more secure against adversaries that aren't Google itself. Chrome benefits from being a younger browser that was developed more recently and was structured from the start for security, Firefox has lagged. Google has poured endless money into Chrome to secure it against bugs, Mozilla has comparably limited funds. There are many bugs constantly being found in Firefox and exploited by hackers to inject malware onto users' computers. This happens less frequently with Chrome.

However using Chrome innately involves trusting Google, a company that happily turns over users to oppressive governments on the regular. Google Chrome also spies on what you're doing and visiting within your browser and sends it all home to Google for future use. There is a purely open source version of Chrome called Chromium, that doesn't spy on you quite as much for Google, but it's less well supported and sometimes can't run various plugins and addons.

This is not a pretty trade off.

With Firefox you're more likely to get hacked by random hackers or get caught in a mass-hacking attempt by the government targetting thousands of users to backdoor their computers (this absolutely happens). However you're slightly less likely of having the Mozilla Foundation comply with a warrant to backdoor your computer when your browser updates and Mozilla isn't spying on you as much in your everyday use. On the other hand Firefox is pretty much the same thing as Tor Browser, so if you're using Tor Browser regularly it's like you're already using Firefox so maybe just stick to the dangers you're already facing and don't hand your information over to Google.

It's generally best to just choose one and stick to it. If you were visiting the same website with one browser one day and the other browser the next day you'd be exposing yourself to the worst of both worlds. If they only put up malware for one browser you'll absolutely end up getting it on your computer if you use both.

Note that "incognito mode" just means the browser clears some session data and cookies. The only person this might protect you against is your low-tech partner trying to quickly check your browser history. Firefox is seeking to eventually integrate Tor Browser into itself by default and allow you to turn on Tor with a button, but until then such claims mean almost nothing.

## Desktop and Laptop Operating Systems

When it comes to operating systems on computers that you're using repeatedly it's a bad idea to continue using Windows. Better to use a Linux variant. Best to use Qubes OS (although as of the most recent edit of this document Qubes still requires signficant technical skill to run).

Windows is closed-source, is in many regards highly vulnerable as well as widely targetted, and Microsoft has a long history of collaborating with governments to imperil their users. Because Windows is closed source you don't have full control over your computer and its code (as well as updates) can't be checked by programmers in the public for vulnerabilities. Microsoft has dragged its feet on upgrades to security that would help users, and famously the stubborn refusal to make Skype even remotely secure (because governments had asked them not to, so that they could better surveil their citizens) led to the deaths of countless Syrian dissidents early in the Syrian civil war. The Assad Regime used the fact that Skype was popular in Syria to let dissidents and activists communicate with one another until it had identified a large number of them, whereupon it murdered or imprisoned them. They also found it easier to deliver malware to activists with Windows computers.

However if you *are* going to use Windows get a recent version that is still being serviced by Microsoft and keep it up-to-date. If you put for example a Windows XP computer on a network it's beyond trivial to hack it and basically anyone can (and will). Of course even with best practices you're still critically trusting Microsoft and your antivirus sofware, and even then a stray image or PDF or word document may end up compromising everything.

While using Linux as your main operating system doesn't in any sense fully protect you it's a step up. If you're getting started with Linux, don't worry, while it comes in different flavors (called "distros"), there is a fairly well established and user-friendly version of linux called Ubuntu that is used by tens of millions of people. You can go to Ubuntu.com to download a copy, which can then be put on a USB. You can boot from this USB and try Ubuntu without installing to get a feel for it, and when you're ready you can choose the option to install. Installing an operating system will typically wipe everything on your drive, so backup your files beforehand. Not every program you use in Windows will also run on Linux, nor is there always another program of a different name that does the same thing, but you'd be surprised at how much does.

There are some slight benefits that can secure Windows users over default Linux users because unfortunately the Linux kernel developers have been refusing to implement certain security functions built by the Linux security community, while Windows *has* adopted these changes. However both operating systems require constant software updates and it's more likely that Microsoft will collaborate with an authoritarian regime to compromise a specific user than Ubuntu.

You can take additional measures to isolate programs you don't trust by only running them inside "virtual machines" on your computer (with VMware or Virtualbox). Whonix is an pair of operating systems designed to be run simultaneously using VirtualBox on your computer. One virtual machine operates as a Tor Gateway, and the other as a workstation. Unlike Tails, Whonix is not "amnesic"; both the Gateway and the Workstation retain their past state across reboots. Not being amnesic improves security on the Gateway, by allowing Tor's "entry guard" system to choose long-lived entry points for the Tor network, which has some security benefits. On the other hand, a non-amnesic Workstation could possibly allow attackers, especially operators of Web services, to inject state and associate a user's sessions with one another, despite the Tor Browser's safeguards; for some users, this could be a serious security exposure.

Again, in the long run the best way to handle security will generally be by running Qubes OS, which isolates every individual program on your computer so no single compromised program can compromise others.

## Mobile Devices

> "The only way to truly secure a cellphone is to put it in airplane mode, turn it off, take the battery out, and then chuck it into a river."

Phones are insecure for a lot of reasons. A standard smart phone is typically an amalgamtion of distinct small computers stuck together. They do this because there's different specialized functions that a phone has (like connecting to the celluar network) that are more efficiently handled using dedicated hardware. The problem is that much of this hardware and software is proprietary rather than open source and also typically written without security in mind. There's so many different bits to your average phone that it increases the number of ways hackers can get in. Indeed they're vulnerable to the point where police have historically used devices that could auto-hack almost any phone near them. However a phone gets hacked, once compromised it's a huge source of insecurity.

Yet even without being hacked, your phone is a huge source of insecurity, as it is almost always sending signals to the surrounding cellular networks, even when it appears that you have no

signal or you haven't paid for service. Even without being compromised your phone operates as a tracking device, with its location being constantly logged.

Turning your phone to airplane mode doesn't necessarily help as there may still be chips on some models transmitting. And even in airplane mode certain components of your phone may still be *listening.* So you may turn your phone to airplane mode and yet still get hacked.

Finally if and when your phone is hacked turning it off is no garantee that it won't continue listening in. A lot of malware will simply interrupt your attempt to turn the phone off while at the same time displaying what looks like a shutdown and then a black screen. In other words you think your phone is off, but it continues to run and record your conversations and/or broadcast its location.

Malware that does this is so widespread that it's a well known problem in corporate espionage and corporate executives will sometimes refuse to bring cellphones into a room, turned off or not.

Taking the battery out is ideal, however modern phones increasingly do not allow this without breaking the phone. And even worse? Some phones that do allow you to take out the battery actually have secret secondary batteries that sophisticated malware can take advantage of.

## Faraday Cages

While phones, tablets, laptops, and ebook readers are all in danger of broadcasting signals or recieving them when we don't want them to, thankfully physics has our back. All signals our devices transmit are essentially just radio signals, that is to say waves in the electromagnetic field. We can block such signals with conductive materials like metal sheets. When you fully block signals by surrounding something in metal sheets that's called a "faraday cage."

What happens is basically that the broadcast electromagnetic wave hits the metal and the loose electrons in the metal are so loose and easily moved by the push and tug of the electromagnetic wave that they all naturally rush around in ways that cancel it. For lazy enough electromagnetic waves the metal sheets can even have holes in them and still work -- this is why there's a metal mesh on the front of microwaves. The holes are small enough that the lazy "microwaves" get cancelled but highly energetic radio waves -- otherwise known as visible light -- can still sneak out through the holes, allowing people to see their hot pockets.

It's important to note that this cancellation effect only works if there's a non-conductive gap between the metal sheet and the device that's transmitting or recieving. If you touch your cell phone directly up against metal it'll just act as a bigger antenna!

As you can probably guess this is where the whole phenomenon of til foil hats comes from. Don't worry, we're not suggesting you need to protect your brain from government radio signals! But Faraday Cages can be quite useful.

Many retailers online sell faraday cage bags or wallets, although some are scams or only block some frequencies, always check the product reviews.

In a pinch you can use a things like fridges, ovens, or microwaves. Just place your devices inside, possibly putting them in a bag, but definitely making sure they're not directly touching the metal of the inside of the fridge/etc. It's worth test calling them or sending a signal to them and see if your appliance is actually blocking the relevant frequency.

Otherwise another approach is to just take a bunch of aluminum foil, fold it over itself several times so that it's thicker, stick your phone inside a canvass pouch or bag and then wrap that with the thick aluminum foil. Again, it's important to test your work. If you leave a hole you could just end up making a directed amplifier, or if the phone touches the foil you could make an antenna.

## Cryptocurrencies

Moving money is frequently a huge pain, and sometimes you want to hide your financial transactions from an authoritarian regime. For example sending money to the kurdish anarchists fighting ISIS might one day be enough to get you terrorism charges if the geopolitical calculus changes and western nations decide to call the kurdish anarchists "terrorists". Similarly if you're a reporter the run in Syria from a regime that has murdered your family and is actively trying to find and torture you, it would be nice to have a way to recieve and send money without going into a bank.

Cryptocurrencies have been an immense benefit to activists and dissidents and *can* provide a secure and anonymous way to transfer money, however note that cryptocurrencies like Bitcoin *are not anonymous by default*.

Bitcoin and similar cryptocurrencies were not originally created to create an anonymous way to transfer money, the intent was rather to create a way to create and sustain a currency that wouldn't allow a government to control it. Part of how Bitcoin does this is involves making every transaction *highly public.* Every single transaction made with Bitcoin is visible to everyone in the world. Bitcoin follows more closely to systems of money used by tribal peoples where "who has how much respect" is kept track of by everyone as public record. This is very different from later approaches to money introduced by early states that use depersonalized metals like gold to keep track of respect and value (allowing the historical source of someone's wealth to be hidden).

If you go to blockchain.info right now and you'll see every single bitcoin transaction that's occurring in real time with no exception. And a ledger that lasts forever.

Now, it not being perfect anonymity doesn't mean there isn't anything you can do to scrub the identity. Basically the chain goes Wallet > Hot Wallet > Tumbler > Market. There's a number of ways to obtain bitcoin in the first place. The easiest way is through what I'll call for lack of a better word a bitcoin bank. These services protect your bitcoins and insure them in a similar way to a bank. Also similar to a bank, they require your real information. This isn't necessarily a problem, because we're going to send those bitcoins through a series of washes or tumbles which will, to a very high extent, obscure where they came from. There is also the option of buying fake documents on the darknet, but because these banks try very hard to keep themselves from being shut down by the federal government, they try their best to comply with all laws, making lying to them a dangerous choice. There are also options to buy without giving your identity. While not technically illegal, the state doesn't like these services, so by the time you're reading about them, they may or may not exist. https://www.bitquick.co/ allows you to buy bitcoin by going to a bank or credit union and depositing cash.

Remember when you're purchasing bitcoin over the internet to use Tor, otherwise you'd be leaking your ip address to the site where you're buying. The Bitcoin core project has integrated Tor onion services to their core network daemon. If Tor is installed in the system, Bitcoin will automatically create an onion service and act as a Bitcoin node over Tor to avoid leaking the

real IP address of the node. However note that this requires an actual install of Tor, not just downloading a standalone folder with the Tor Browser in it (as with Macs or Linux).

There are tradeoffs between keeping your money (once translated to cryptocurrency) in a cryptocurrency bank, on a computer, or on your phone. Cryptocurrency banks do occasionally fail or get robbed, and sometimes they collaborate with governments. On the other hand keeping your money on your computer or on your phone is dangerous because any piece of malware could immediately steal it all. If having read this entire guide you believe you can secure your computer then it would be a good idea to keep your bitcoin on your computer or even on an encrypted USB. (Remember that all an attacker has to do is get access to any copy of the bitcoin wallet file you have to be able to transfer them to a wallet/address they control.) It's advisable not to keep much bitcoin on your phone, but there are phone apps that will enable you to spend bitcoin while out and about (like, for example, at a Bitcoin ATM).

If you're in a major city there are often Bitcoin ATMs that sometimes allow purchase or removal of Bitcoin up to a certain amount with ID. Again, it depends and sometimes changes. One venerable approach to anonymous Bitcoin is to buy with cash at an ATM, send to the recipient's secret address, and have them then withdraw from an ATM near them. This specific approach reveals to the world the cities and ATMs that the cash was deposited and withdrawn from, but not necessarily the individuals involved. Although as always note that CCTV cameras recording your visit to the ATM may be used against you.

An example interaction with a Bitcoin ATM (of a Genesis1 machine type) looks like this: Choose withdraw cash. Choose Bitcoin (these machines normally may support other cryptocurrencies). Choose amount to withdraw. Send bitcoins to given address QR code. Receive cash immediately as the bitcoin transaction is propagated on the network and the ATM then dispenses. ...However note that other ATMs may require a mobile phone number.

**Places to buy bitcoin with identity attached:** Coinbase, Circle

**Places to buy without identity attached:** bitquick.co, paxful.com

**Lists of darknet markets:**

- https://reddit.com/r/darknetmarkets/wiki/superlist
- https://www.deepdotweb.com/2013/10/28/updated-llist-of-hidden-marketplaces-tor-i2p/

There are some new cryptocurrencies being worked on (namely Zcoin and Zcash, implementing the ZeroCoin and ZeroCash protocols) that attempt to build anonymity into the currency from start. These are still fledgeling cryptocurrencies. Zcash conceals the amount of money sent in each transaction, whereas Zcoin does not. So Zcash is less prone to privacy timing attacks than Zcoin. On the other hand, this comes with a big tradeoff for Zcash, in the form of potentially undetected hyper-inflation in Zcash’s money supply. Zcash has the more famous developers.

## Dead drops & Geocaches

A geocache is something you physically buried somewhere secret that you can dig up later.

Contents of a geocache might be A USB with Tails on it. An encrypted USB with a GPG key, passwords to your accounts on it. Instructions to your prison support team. A bit of cash.

Stick them inside ziplock bags and seal those inside a bottle. Without taking your phone with you, bring your cache to some park or wilderness area or bit of shrubbery by a highway and dig a small hole that's very unlikely to be noticed, memorize it. Put the cache in and cover up.

These are useful in a *huge* variety of situations. You might want to leave an encrypted USB with some critical files and a KeepassX database of your accounts and passwords. You could leave this for yourself in case your house is raided and all your electronics are taken. Or you could create one geared towards helping a support group if you're imprisoned.

For example you're placed behind bars but you can call your friends or get visitations. In the first visitation or call you tell your friend where the USB is buried. In the second they confirm they got to it before the cops and passed it along or hid it somewhere themselves. If they did *then* you tell them the password, enabling them to unlock the USB or the KeepassX database and get access to your accounts (social media, financial, or other), even special instructions that you don't want to say while the cops can hear.

## CCTV and Other Camera Survellience

Black bloc (and we mean full black bloc, where there are no identifying features or skin shown and all the clothing is solid black, not "punk bloc" where someone wears some black clothes and a bandanna but still wears patches or exposes their hair or skin color) is good and all, but even done right it doesn't save you. A camera might catch someone's distinctive black boot, or someone's black hoodie might be slightly faded in a way distinguishable by going over the photography carefully.

## Security Culture

The central principle of all security culture, the point that cannot be emphasized enough, is that people should never be privy to any sensitive information they do not need to know.

The greater the number of people who know something that can put individuals or projects at risk—whether that something be the identity of a person who committed an act, the location of a private meeting, or a plan for future activity—the more chance there is of the knowledge getting into the wrong hands. Sharing such information with people who do not need it does them a disservice as well as the ones it puts at risk: it places them in the uncomfortable situation of being able to mess up other people’s lives with a single misstep. If they are interrogated, for example, they will have something to hide, rather than being able to honestly claim ignorance.

Don’t ask others to share confidential information you don’t need to know. Don’t brag about illegal things you or others have done, or mention things that are going to happen or might happen, or even refer to another person’s interest in being involved in such activities. Stay aware whenever you speak; don’t let chance allusions drop out thoughtlessly.

If someone challenges your credentials as a radical or activist let it slide off your back, don't feel obliged to allude to secret things. The results of your actions and projects should be sufficient reward. If you find yourself longing for public recognition meditate on your motivations.

Similarly don't work alongside people that brag or run their mouths, but don't turn rational precaution into toxic suspicion. Ideally security culture is a form of etiquette, a way to avoid needless misunderstandings and potentially disastrous conflicts.

It's easy to use "security concerns", secrecy and an air of paranoia to establish internal hierarchies and power structures. This can destroy a group or project just as much as an actual snitch. *Do* share non-sensitive information, context and skills that will empower people. Security culture is not never telling anyone anything or closely gaurding all information, it's just being mindful of what information might place people at risk. For example in some countries it's illegal to be identified as an anarchist or an atheist and so that information needs to be handled carefully, and you must avoid even revealing others' political affilliations. In other countries this is not the case and clamming up or getting accusatory when someone asks if the cute boy at the potluck is also an anarchist would be wildly overblown security theater.

Security concerns should never be an excuse for making others feel left out or inferior. Those who violate the security culture of their communities should not be rebuked too harshly the first time—this isn't a question of being hip enough to activist decorum to join the in-group, but of establishing group expectations and gently helping people understand their importance; besides, people are least able to absorb constructive criticism when they're put on the defensive. Nevertheless, such people should always be told immediately how they're putting others at risk, and what the consequences will be should they continue to. Those who can't grasp this must be tactfully but effectively shut out of all sensitive situations.

Note that merely having worked beside someone for a while is not sufficient reason to fully trust someone. And feelings are a relatively easy thing to fake. Undercovers and informants are generally good at faking emotional expressions. It's easy to present as passionate about hating the cops or wanting to save the environment. There are numerous cases of activists accidentally starting sexual and romantic relationships with cops. Any informant can memorize some cutting-edge vocabulary. And merely having committed an illegal act with someone before without getting busted is not sufficient proof they weren't establishing a record.

Widely accusing someone of being an informant or untrustworthy can be a hugely divisive and destructive act, it should only be done in situation with very clear proof. False accusations can destroy a community, leading to a storm of allegations and counter-allegations, and such division is often encouraged by undercovers. At the same time one shouldn't reject allegations as this can create just as toxic an environment. Rapists and abusers love to claim that accusations against them are just attempts to sabotage, but realistically there are usually far more rapists and abusers in a movement then there are undercover cops.

Trust is a complex thing. It shouldn't be seen as an all-or-nothing, or even one-dimensional. We trust people in various specific ways. We might trust someone never to talk to the cops, and also never to run their mouth and brag to friends, but we might also distrust their ability to maintain good electronic hygeine. They might for instance visit neonazi sites without using Tor Browser (leaking their ip addresss) because it's "such a bother to start a separate browser". Or they might download all sorts of junk on their computer and open random attachments on emails.

Similarly if you're collaborating with Alice, Bob, Carrie, and Darnell then while you may trust Alice and Darnell to not just encrypt but authenticate their communications with you, it may be the case that they don't authenticate when speaking about the same things to Bob and Carrie. Or that while Alice and Darnell always demand on authentication, Bob and Carrie don't use it when communicating about the same things with one another.

Psuedonyms, shorthand, or code words can sometimes be useful, but remember that your adversaries aren't idiots. They can put things together. And while you may think you have

established some legal "plausible deniability" a judge with some common sense and an authoritarian disposition may disagree. Remember that none of these should EVER be used as substitutes for proper encryption, authentication, and anonymization software. If you use them they should be used in addition to proper technology and/or avoidance of technology. So you have an encrypted channel but talk about "baking cakes" and "parties" over it rather than openly talking about political meetings or throwing blood on the local police precinct. But note that an internal lingo or code-language can provide one more barrier to newbies.

It's important not to let over enthusiasm for security turn into one more way to reinforce existing interpersonal hierarchies. A healthy movement is one that shares resources, contacts, friendships, and skills rather than greedily hording them to score points. Always avoid situations where a very small number of people are controlling the flow of vital information to everyone. It may makes sense to only have a few people with keys to a website's server, but if hundreds or thousands of people are depending upon that website then it may be worth replicating it or its function on other websites run by different people.

Finally however, don't be intimidated by bluffing on the part of your adversaries. Cops will almost always pretend to have evidence or testimonies they don't have.

# Threat Modeling

## Map out the threats you actually face

One of the most important things an activist can do is to try and comprehensively map out the threats they actually face. This can start with the things they're trying to accomplish, the context in which they're operating, the things upon which they're depending, and all the various ways an adversary could try to detect and/or interfere with things. The goal is that -- instead of visualizing our enemies as nebulous clouds of potential spying -- you should think through what you would do as a cop and what you might reasonably have available as a cop. Or even just what a particularly vicious stalker might be able to get access to.

What would the processes look like that end in the bad results you're trying to avoid? A couple cops are assigned to monitor the radical left in their town. What would you do if you were them? Try to out think yourself. But be reasonable about the limits to their budget and technology. Those cops may look online for anyone discussing your town and radical politics or arrest records at a recent protest. They may then use your name to look up your phone number. Immediately they have a map of everywhere you've gone in the last two years and everyone you've messaged or called. They may also try to find social media accounts associated with your name. You might have them under a different name but used your personal phone to register them, at which point the local cop can (if warrants or the like are easy to get there) just ask Twitter/Facebook/Tumblr/etc for any accounts that use that phone number. Now they have everything you've written or messaged on there, and your social graph of friends. Since they know your name and phone number they can also look up your address, maybe go through your trash and find financial statements or phone bills tied to other identities you use. ...Maybe if you're a high-value target and worth the money they move a mobile IMSI-catcher (stingray) near your house and intercept your cell connection. They could use this to try and hack your phone using the cheap software sold to law enforcement by unethical hackers. Maybe they park a car across from your house and set up a device to video record your front door or another device to try various passwords to access your wifi network. (All of the above have been used against activists/dissidents in just the US.)

Sequester your identities. Work out precisely what content touches what content. If you're in an organization trying to do underground work and you've set up one Signal group to only decide on meeting times/places and then another set of secret email accounts under psuedonyms only used for messaging one another about working group tasks, it would be a bad idea for someone to ask "hey what's your group email account again?" over Signal, much less for the other person to answer. You don't want the cops to hack or take someone's phone and learn that the group named "knitting circle" is actually tied to the group-only email accounts. And of course you don't want to use people's actual names over the group-only email accounts.

Of course varying types of activism or contexts of dissent require different levels of security.

If you're planning an action, begin by establishing the security level appropriate to it, and act accordingly from there on.

Learning to gauge the risks posed by an activity or situation and how to deal with them appropriately is not just a crucial part of staying out of jail; it also helps to know what you're not worried about, so you don't waste energy on unwarranted, cumbersome security measures. Keep in mind that a given action may have different aspects that demand different degrees of security; make sure to keep these distinct. Here's an example of a possible rating system for security levels:

1. Only those who are directly involved in the action know of its existence.
2. Trusted support persons also know about the action, but everyone in the group decides together who these will be.
3. It is acceptable for the group to invite people to participate who might choose not to—that is, some outside the group may know about the action, but are still expected to keep it a secret.
4. The group does not set a strict list of who is invited; participants are free to invite others and encourage them to do the same, while emphasizing that knowledge of the action is to be kept within the circles of those who can be trusted with secrets.
5. "Rumors" of the action can be spread far and wide through the community, but the identities of those at the center of the organizing are to be kept a secret.
6. The action is announced openly, but with at least some degree of discretion, so as not to tip off the sleepier of the authorities.
7. The action is totally announced and aboveground in all ways.

To give examples, security level #1 would be appropriate for a group planning to firebomb an SUV dealership, while level #2 would be acceptable for those planning more minor acts of property destruction, such as spraypainting. Level #3 or #4 would be appropriate for calling a spokescouncil preceding a black bloc at a large demonstration or for a group planning to do a newspaper wrap, depending on the ratio of risk versus need for numbers. Level #5 would be perfect for a project such as initiating a surprise unpermitted march: for example, everyone hears in advance that the Ani DiFranco performance is going to end in a "spontaneous" antiwar march, so people can prepare accordingly, but as no one knows whose idea it is, no one can be targeted as an organizer. Level #6 would be appropriate for announcing a Critical Mass bicycle ride: fliers are wrapped around the handlebars of every civilian bicycle, but no announcements are sent to the papers, so the cops won't be there at the beginning while the mass is still vulnerable. Level #7 is appropriate for a permitted antiwar march or independent media video screening, unless you're so dysfunctionally paranoid you even want to keep community outreach projects a secret.

It also makes sense to choose the means of communication you will use according to the level of security demanded. Here's an example of different levels of communications security, corresponding to the system just outlined above:

1. No communication about the action except in person, outside the homes of those involved, in surveillance-free environments (e.g. the group goes camping to discuss plans); no discussion of the action except when it is absolutely necessary.
2. Outside group meetings, involved individuals are free to discuss the action in surveillance-free spaces.
3. Discussions are permitted in homes not definitely under surveillance.
4. Communication by encrypted email or on neutral telephone lines is acceptable.
5. People can speak about the action over telephones, email, etc. provided they're careful not to give away certain details—who, what, when, where.
6. Telephones, email, etc. are all fair game; email listservs, fliering in public spaces, announcements to newspapers, etc. may or may not be acceptable, on a case-by-case basis.
7. Communication and proclamation by every possible medium are encouraged.

## Top Tier Security

Using Tails and Tor set up anonymous email addresses, not tied to any device or IP. Each individual creates a PGP key for that email address alone and keeps on an encrypted USB. In person check and compare everyone's PGP fingerprints. Only ever access these email addresses through Tor and Tails (and only using a public wifi away from cameras). Never through a persistant Operating System. Do not use personal names ever in these emails, only codenames. You might choose to also have a shared PGP key (kept on the same encrypted USB), that you never use for signing or encrypting emails, but that you use for signing Pastebin'd communiques. Never communicate over phones/smartphones at all. Never use the computer you plug Tails into for anything sketchy or download files from the internet. If possible buy it originally all of a sudden in person with cash without showing your ID.

## Things the police already do in the USA

- Follow you and your friends around for two years without your knowing.
- Tackle you, beat you up, and arrest you for jaywalking.
- Put five informants and undercovers from different law enforcement organizations in an peaceful aboveground organization with meetings of only twenty or so people.
- Have undercover police officers or paid informants enter into sexual relations with the people their spying on. Stoke an infatuation with them and then leverage that to set you up. Father a child with you.
- Murder you in your bed while you sleep.
- Plant a bomb in your car.
- Literally drop a bomb on your house and neighborhood.

...Just a reminder. It's important to neither be paralyzed with fear, but it's also important to be clear-eyed about the degree to which the threat can escalate.

# The "well actually" section

## Weird fucking edge case attacks

### Cold boot

The whole point of full disk encryption is that if you're computer is turned off no one should be able to read the contents of your drive. However this isn't quite true. If an adversary seizes your computer in a short period after you've turned it off it may still be possible to decrypt your drive provided the adversary has some advanced technology, luck and patience. The point is basically that the part of your computer that stores "active memory" when your computer is on includes the decryption key for your harddrive (so it can read and write to your harddrive without you constantly re-entering your password). When you turn your computer off this "active memory" quickly decays away and is lost. However if someone sprays liquid nitrogen on your computer's RAM soon after you've shut down they can freeze the bits of memory still in there and prevent them from decaying. Then they can plug the RAM into a bit of technology and read most of the bits off. They may not get them all, but if successful it'll be possible for them to guess or trial-and-error the rest. We haven't seen this attack very frequently in the real world and your average cops are highly unlikely to use it because it's burdensome. But the mitigation is to turn your computer off the first moment the cops come knocking or raiding.

### Van eck phreaking

This falls in the class of attacks where the adversary is sitting in the next room over or has put spy gear in the wall. Basically when your computer runs its electronics innately generates electromagnetic waves and these waves can sometimes be interpreted in ways that reveal what you're doing. The light and radio waves are both just instances of ripples in the electromagnetic field, caused by the movement of electrical charges. The electrical movement in your computer's central processor is too tiny and dense to generate a decipherable signal but things like the wires that deliver instructions to your screen, or the firings of wires from your keypad can sometimes generate signals that can be detected and decoded. This would require a lot of setup by your adversary and is very very rare because it's so difficult. The way to prevent this would be to only use your computer in new randomly chosen environments each time like a cafe, or to set up a giant faraday cage around your computer or the room you use it in.

### Sound pattern analysis

This is like a poor-man's Van Eck Phreaking, basically you just use some analysis software and a hidden spy microphone to make good guesses as to what someone is typing based on the sound of their key strokes. If you can't get access to the room they're in you could use a laser device from far away pointed at a window to the room and have that device read the sound variations on the window. Again this is rare. To defend against this we imagine that you could record random typing and then layer a bunch of tracks of that and then play it while you're typing. A new musical genre! Or you could poke the keyboard erratically. Again, this is a hyper-rare and rather involved surveillence technique.

### Satellite visual surveillance

Okay so what do you do if they're literally watching you from space? Well this is a low likelihood for activists in the present and the immediate future, basically because it costs a lot to use active spy satellites just to surveil individuals. But one could certainly imagine this cost decreasing as well as a modern fascist government turning its military spy satellites on domestic dissidents. The more reasonable usecase might be that they identify a house as being a site of dissident activism and then track people and cars going too and from it to identify them and their destinations. So even if you don't bring your phone you could still get tracked. We're necessarily forced into speculation as to best avoidance strategies, but in many ways this is just an extreme case of the normal CCTV camera surveillence networks. The same general obfuscation and counter-surveillence tactics apply. Trade cars in a parking garage. Go into a crowded public place and swap into a different change of clothes. Use a different style of walking. Be aware that it may not be enough to hide in the woods because your heat signature may be visible through the trees.

### Quantum Computers, Shor and Grover's Algorithms

Quantum mechanics isn't magic and almost everything said about it in popular layman discourse is wildly off basis, this includes claims that quantum computers will break everything. Although they will break some types of encryption. The "weirdness" of quantum mechanics is in essence just that we learned that contrary our intuitions and daily experience the universe actually calculates probability using complex numbers rather than just real numbers, every single "weird" thing in quantum mechanics directly arises from this single mathematical quirk. No wild philosophical conclusions required. Unfortunately it may soon (in matter of years) be possible to build a computer that operates on such a small scale as to take adantage of these probabilities that extend into the complex plane. When finally achieved there are a couple fancy algorithms that can take advantage of this unique behavior to speed up some specific computations. Shor's Algorithm makes factoring numbers much much easier and Grover's algorithm makes searching for an entry in a database faster. Basically Shor will break many popular classical ways of doing public key crypto, and Grover will make forms of symmetric key crypto behave as though their key was half as long. Symmetric key crypto is easy to save, just double the length of the encryption key! To save public key crypto we will need to adopt ciphers that rely upon hard mathematical problems that quantum mechanics don't make it a snap to solve. People are already implementing solutions to both. But the takeaway for activists is that -- worst case -- some of your present encrypted communications might be decryptable by the NSA in 10 years time anyway.

## Okay but what if they shut the whole internet down?

So thankfully there are deep economic costs to this, but sometimes a regime feels like that's a price worth paying. Or has already wrecked their economy so badly that it doesn't matter.

### First tier of censorship is a selective ban on certain websites

The most common way around this is to either use Tor or a VPN to connect to a server in another country where that site may still be reachable on the internet.

Sometimes the government will be so lazy that they merely ban anyone traveling to the domain of the site (ie "yahoo.com") rather than the actual underlying ip address of the site (ie

"13.112.2.45"). Then you can just type the numbers of the ip address (if you know them) into your browser's addressbar and go there directly.

Sometimes the government will grab control over the servers that your computer asks to look up which domains correspond to which ip addresses. You can get around this by tinkering with your computer to get it to ask different servers to do that lookup. This is a bit complicated though.

## Second tier of censorship is blocking certain protocols (through Deep Packet Inspection)

In this approach the state has servers on the network that peer into what information you transmit back and forth and block anything that looks like Tor, for example.

We get around this through "steganography" that is to say making one sort of thing look indistinguishable from another thing. Tor provides a number of "pluggable transports" that do precisely this.

## Third tier of censorship is just cutting the internet inside a nation off from the wider global internet

At this point the problem becomes establishing a connection with the outside. This can be done through physical connections that the state can't detect, like laser uplinks between two friends on either side of a border, or a connection to a statelite that the state doesn't control. Another option is to piggyback on whatever connections the state might leave open, like telephone lines or private financial connections like between Banks or retailers and credit card companies.

In all of these cases preparation is required to coordinate with individuals or devices located outside of your country before the walls slam down.

However once you personally get a connection out it may be possible to share with others, or to pool your multiple connections out, via a steganographically hidden secret network within your own country. Such networks would have to behave differently than Tor, however, because Tor makes every server publicly to everyone on the network.

## Fourth tier of censorship is a severe case of whitelisting

In this situation the internet ceases to be anything like network between equals and instead moves to merely a content delivery system where users can only connect to a limited number of approved content providers that the state has backdoored and controls with the threat of shutdown. Any connection to these approved servers must be logged and explainable, and servers are forbidden from relaying messages between users.

This is basically the internet reduced to literally nothing more than television + shopping. And is obviously the grand ambition of every ruler and authoritarian around the world.

Realistically however there would likely be deep organizational and economic problems forbidding all messages between users. And whatever window they leave open enables us to attempt steganography over it.

## Fifth tier of censorship is just shutting down the internet altogether

This could be done through a slow legal process, or armed raids or major infrastructure and eventually individuals, or a forced infrastructural/civilizational collapse that makes certain technologies materially infeasible to one's society, or finally EMPs that fry electronic devices.

# An Outline of Further Study

We're frequently asked by curious and industrious but confused young activists for a map of what to study to get a grasp on what goes on with the magic glowing rectangles that are now so central to our lives. The following is a very short map of stuff so that you don't feel completely lost in a sea of unknown unknowns. It is a map and not an actual explanation, so don't feel overwhelmed! Our purpose is to give you things to look into further on your own!

First there's the physical components of a modern computer: 1) The *processor* (often a central processing unit) is a chip that does the core thinking of a computer, it has sub-components that do arithmatic and logical operations, 'registers' that hold instructions and the data being manipulated, and cache memory that contains the data it needs to access most quickly. 2) The *RAM* (rapid acccess memory) is a component that stores data that may need to be accessed by the processor, like the contents of a file you've opened in a program. 3) The *hard drive* or disks, that store data in a permanent manner that persists even when the computer is turned off (unlike data in the processor or RAM). 4) Various "devices" like a monitor or keyboard or mouse or specialty chips like video processors (processors built to better handle video than central processors). 5) *The Motherboard* is a circuit board where the processor is often built into it and there are connectors between it and the various other components. ...Note that things can get a lot more complicated in practice, as with modern smartphones which are often really multiple computers combined into one.

These physical components wouldn't do anything without instructions, or software. The software of a computer ideally starts with the BIOS, a bit of software built into almost every computer at the lowest level. The BIOS helps the hardware run. So when you turn on your computer the first thing that runs is the BIOS which connects all the computer's physical components together and gets them working. Then the BIOS reads data from a disk (like a harddrive or a USB stick), and looks for a Boot Loader. The Boot Loader is a tiny bit of code that is pulled from the disk into active memory and run to then load the Operating System. If your computer has full disk encryption then really it's more like 99.99% disk encryption, with one tiny bit left unencrypted, the Boot Loader which contains the code to generate a key from whatever password the user enters and then try to use that key to unencrypt the rest of the disk. The Boot Loader also loads the Operating System, so it reads more data/code off of your disk and boots up a giant core program like Windows or Linux and can then load other programs as helpers. The core Linux OS is very small and everything you see on your screen when it "loads" is really other programs (a graphical interface program, a folder display program, etc). These programs often share common bits of internal code, called "libraries". For example a program to check your email and a program to browse the web might both use the same library to do encryption. Some big programs may have sub-programs that are optional, like "addons" and "plugins."

When information flows around the world it obviously does so through physical means, like wires, radio waves, or via lasers. But just pushing on and off fluctuations (1s and 0s) from one device to another doesn't necessarily mean much of anything on its own. The data that's relayed is structured. Often one structure of data will contain another structure of data within it. For example an "IP Packet" (internet protocol packet) is like an envelope, in addition to whatever is contained inside it has a source address and a destination addresss. There's also things like a version number and how big the contents are. You could imagine the packet as a raw stream of data like this "VERSION1.0|78 CHARACTERS|FROM:1.1.1.1|To:2.2.2.2|Hello Dave, How Are You Doing?" Except that in reality an IP packet involves 14 different types of

metadata and expresses them very succinctly. For more on the nesting dolls of packet protocols see the "OSI model."

The modern internet involves several types of connections to load a webpage. The first connection is a DNS lookup which hits a series of servers to match a domain "Riseup.net" to a computer-friendly address "198.252.153.69", this can involve multiple DNS servers in a hierarchy from a "root server" which returns where to look for the ".net" server which then in turn answers with the location for the "riseup.net" server (there are actually 8 steps to this and can be more if you look up mail.riseup.net for example). The second connection is an HTTP connection which loads the content of the page (often in the format of HTML for data, CSS for styles, Javascipt for responsive behavior, and whatever files like images). However if you load the page over HTTPS (eg "https://mail.riseup.net") then your computer will attempt to establish an "encrypted" connection with the server and pull the content (HTML, CSS, Javascript, images) over that encrypted connection -- which requires a whole other step. Your computer doesn't know how to trust that the server you're trying to set up an encrypted connection to is actually the correct server, so the server sends you an "x509 Certificate". This certificate is (ideally) signed by someone else (claiming to have legitimated that the Certificate is the correct one for that domain). And their capacity to certify may have been signed off on by someone else. On and on it goes until (ideally) some root Certificate Authority is reached that your browser already trusts because it was installed by default to trust that Certificate Authority. There are hundreds of such authorities. Yes, any one of them or the countless people they sign off on can pretend to be "riseup.net". Now you nderstand why the internet is broken.

# Endnote

This is still a somewhat incomplete and unfinished text, and will probably need to be updated. We happily encourage people to submit edits or concerns, either as pull requests or as issue tickets.