Privacy is a UX Problem

David Dahl, SpiderOak, Inc.

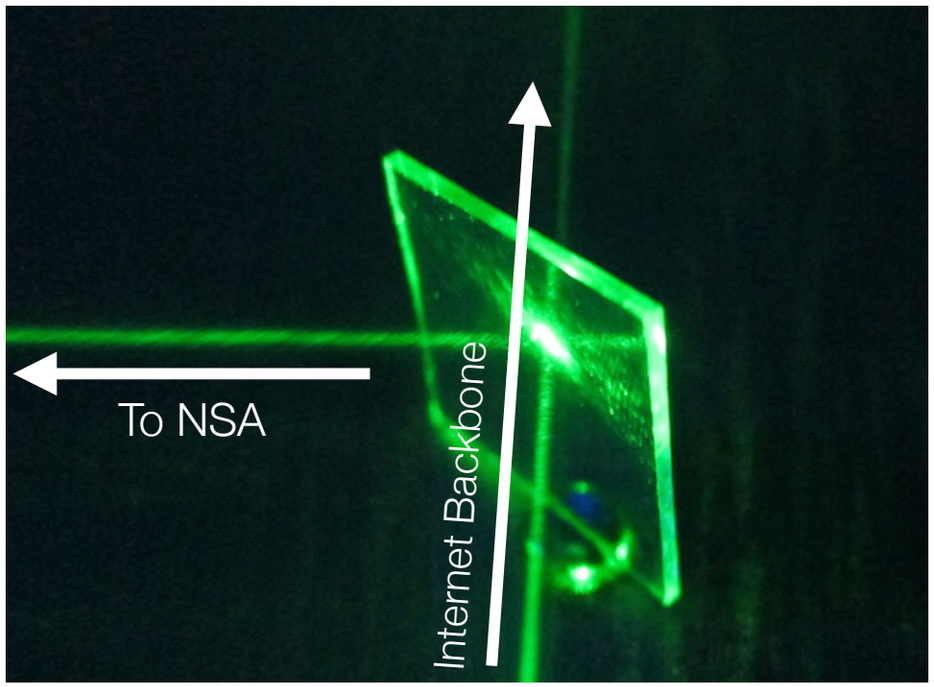
Future Insights Live

Las Vegas, NV June 3rd, 2015

David Dahl

- Privacy Enthusiast
- Director, Crypton - SpiderOak
- Former Privacy Engineer, Mozilla
- ILM, DOE, consultant
- @deezthugs

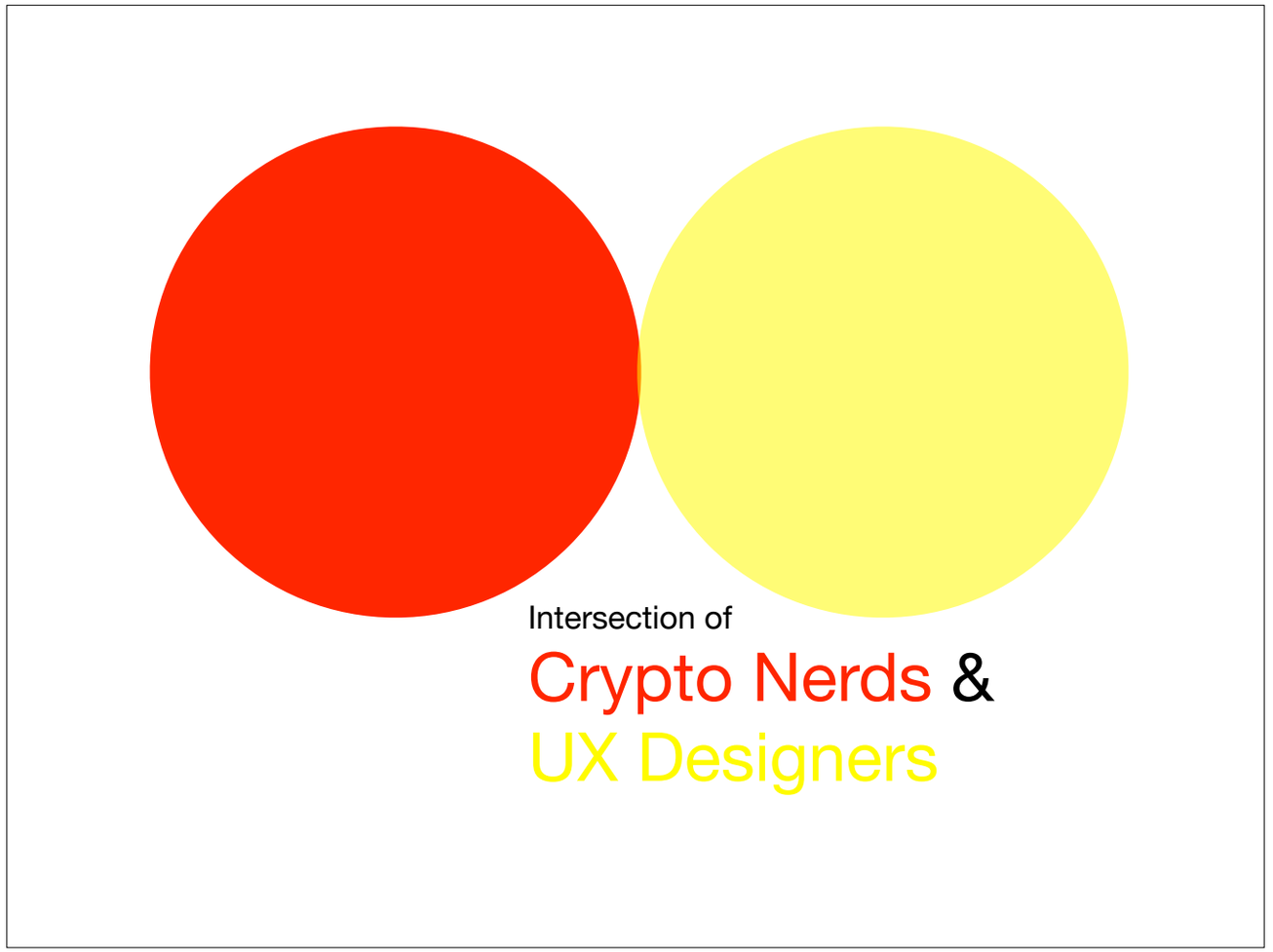
mozilla



A beam splitter

In 2006, a whistle-blower came forward from ATT named Mark Klein. He gave us a heads up that our personal communications were being siphoned off of the internet via a literal prism that split the internet backbone fiber optics line in San Francisco to be copied, searched sorted, sliced and diced. “Former director of the NSA's World Geopolitical and Military Analysis Reporting Group, William Binney, has estimated that 10 to 20 such facilities have been installed throughout the United States.” https://en.wikipedia.org/wiki/Room_641A

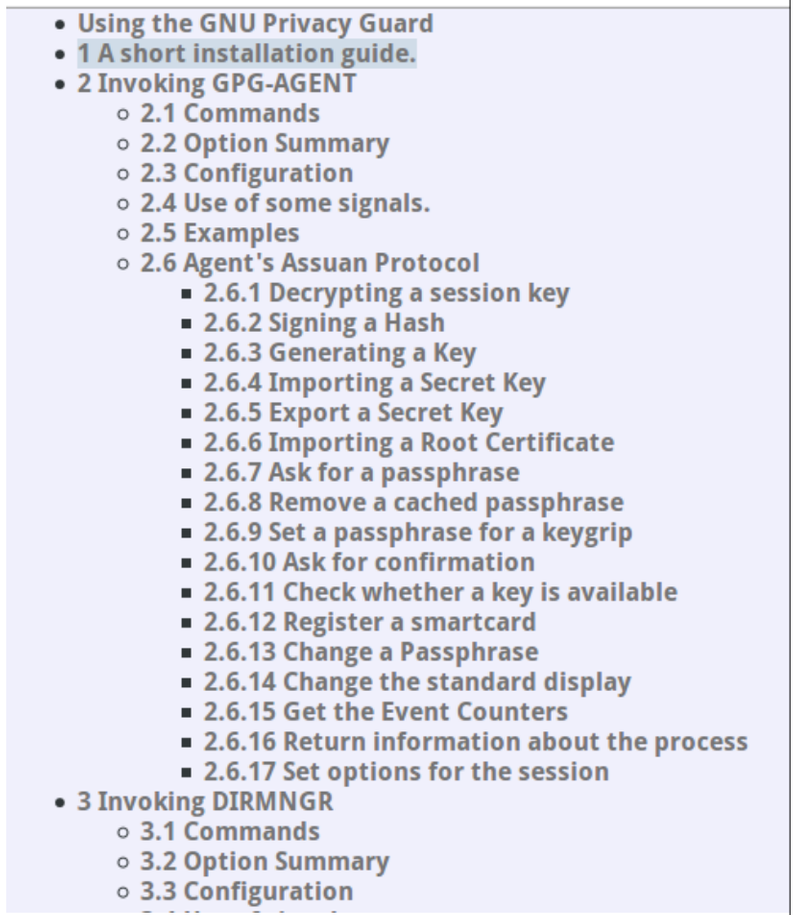
I find this very disturbing. Also, how difficult is it to send a truly private message to someone online? For most users this has been until recently effectively impossible. Naturally this got me thinking about Silicon Valley companies and what they actually do - and the fact that so few of them provide privacy with their services, least of all privacy from *them*



The number 1 reason for this talk is in this Venn diagram. Do you see any orange in this Venn diagram? A sliver?
When I was at Mozilla I remember bending our UX designers' ears about privacy and crypto and User Interface design. It was a bit baffling for both of us to try and understand each other. There really was too little interaction between privacy & crypto engineers and UX. Most of the software designed for preserving online privacy suffers from this

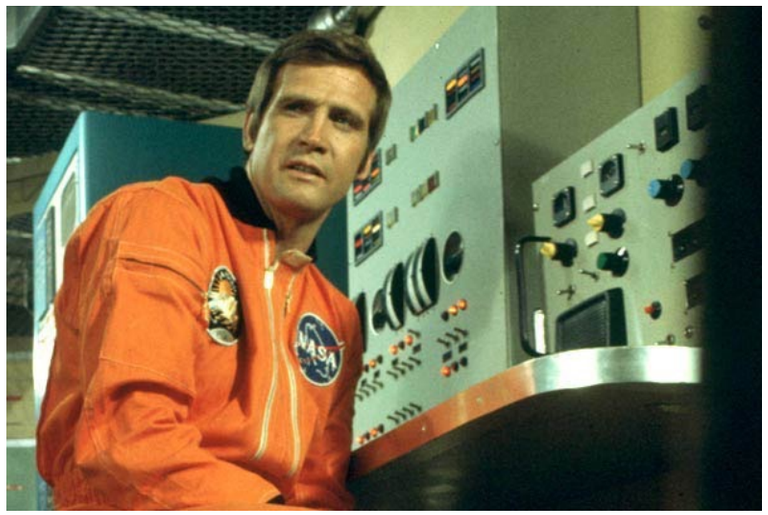
Wait, privacy is
just
a UX problem?

Well, kind of…



Who has heard of GPG or PGP? Used GPG or PGP? This image is a screenshot of the first page of the GPG user guide. Clearly, GPG documentation is dense, technical and long winded. This is not going to work for the majority.
In this talk I want to navigate through concepts, issues and pitfalls of using apps built for privacy. I will also cover newer advances in privacy UX. Please take this talk as a call to action to familiarize yourself with privacy techniques and tooling - you *can* make it better

We have the technology!



We have incredibly fast and capable devices today. There is no reason we cannot also have privacy in all of our communications.

We have very fast, tiny computers

in our pockets

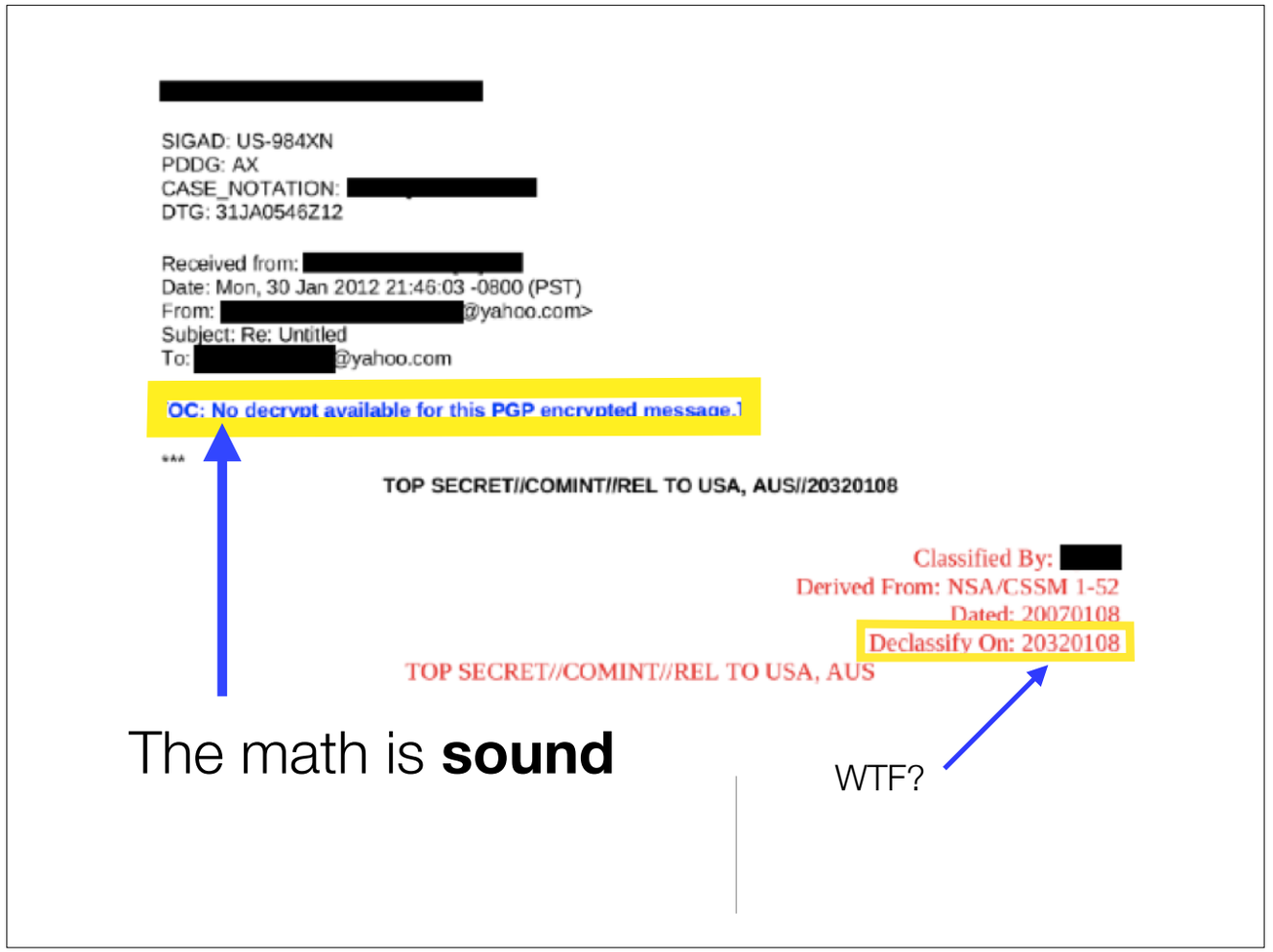
All of us do. Some even have 2.

…and on our wrists.

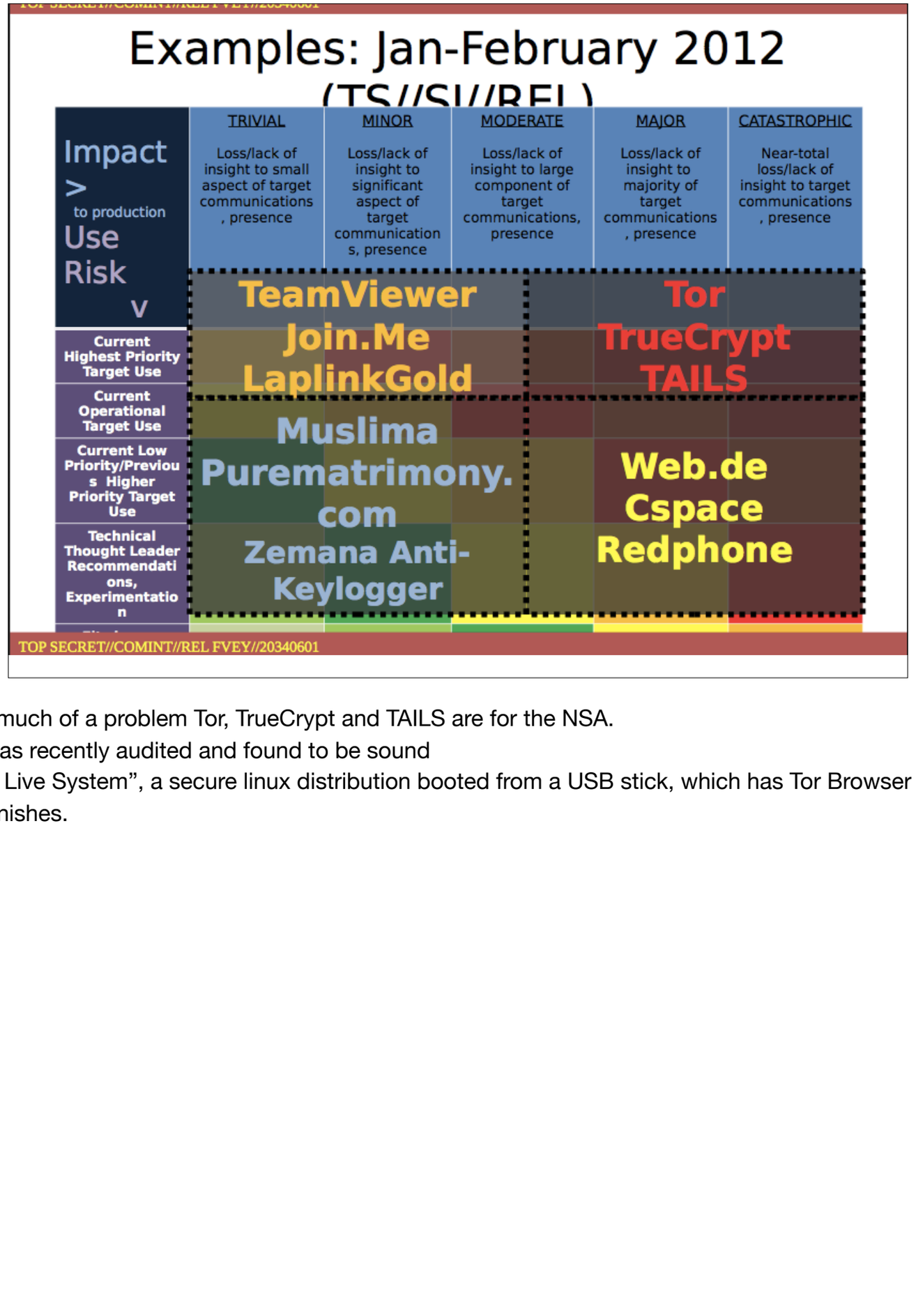
These are devices are capable of decrypting real-time streaming video & audio data! Ok, well, maybe not this device, but one a bit like it.

We have very fast networks connected to these tiny computers

Not to get out in the weeds, but, some of our network operators are hostile to privacy, which presents further problems.



Strong crypto DOES work. This is a page from Snowden's files. Even the NSA cannot decrypt a PGP message. Note: this does not stop the NSA from storing this message until the computer is broken into and the private key is taken. Unless… January 2032 arrives and they delete or reclassify this message! Srsly?



This Snowden slide illustrates just how much of a problem Tor, TrueCrypt and TAILS are for the NSA.
TruCrypt is a disk encryption tool and was recently audited and found to be sound
Tails stands for “The Amnesic Incognito Live System”, a secure linux distribution booted from a USB stick, which has Tor Browser and other tools ready to use. Once you shutdown all traces of your activities vanishes.

Privacy nerds have an **embarrassment** of riches in crypto libraries

[NaCl, BoringSSL, SJCL, LibGCrypt, NSS & others in many languages]

Not that ~~m~~any of them are *easy*
for developers to actually use

Crypto libraries must be used with care. When I worked at Mozilla and implemented window.crypto.getRandomValues it took an inordinate amount of time to land a feature that “just” produced random numbers.

You've probably noticed a lack of **web** applications

Web pages are not considered safe for cryptography & privacy applications. You know, those pages that load JS files from CDNs that can be anywhere. JavaScript source files are not verified at all, they are just downloaded and immediately run.

This problem is not going away any time soon. Browser vendors are slowly working towards signed and verified web apps, but it is such a vast change in the way browsers work.

Naturally, you can build HTML5 apps that can be packaged and signed and not load external content. This does work. I am doing it today.

Pitfalls

Common issues in creating and using private applications

Lets talk about some of the pitfalls we run into with privacy apps. Some of these issues are related to GPG & PGP. But even some newer applications suffer from some notable problems.

Understanding Cryptography concepts is a **requirement** for using many privacy applications

Some apps ask the *user* to choose algorithms & key lengths (whatever that means!) to use

---

GPG / PGP, which **does work** to defend people against dragnet surveillance can be horribly difficult to even install

That being said, there are some new takes on PGP: Yahoo Mail has a browser extension, Whiteout.io is a new email client that supports PGP with any eye toward better UX, and there are others as well

Jargon

keys

fingerprints

signatures

Jargon and comcepts in Crypto is dense and complicated. A Math degree is required to truly understand it. For instance, a “key” is used to unlock data but it is also used to identify you. Fingerprints make it easy to verify keys. Signatures verify that the sender is the originator of data - or has access to the private key that created the encrypted message. The “signature” is the best real-world analog here.

Side note, most everyone who you think *might* be a crypto nerd will never admit to it. Some of the top engineers in Silicon Valley who do crypto every day will not admit they “know crypto”. It is a dark art. I only claim to be able to use APIs, with guidance from professionals.

Keys

Most applications have a step where “keys” are generated

Do users really need to know about this?

Knowing about key generation adds complexity to the mental model a user is used to in a communications app. Hiding this step is worth considering. I would prefer to not refer to keys at all. Creating new metaphors for these crypto-specific operations is a better idea. A “key” really represents a user - or the ability to converse with said user. A peer’s keys represent them - can’t we just use an “account” & “contacts” model here and do away with all of this key jargon?

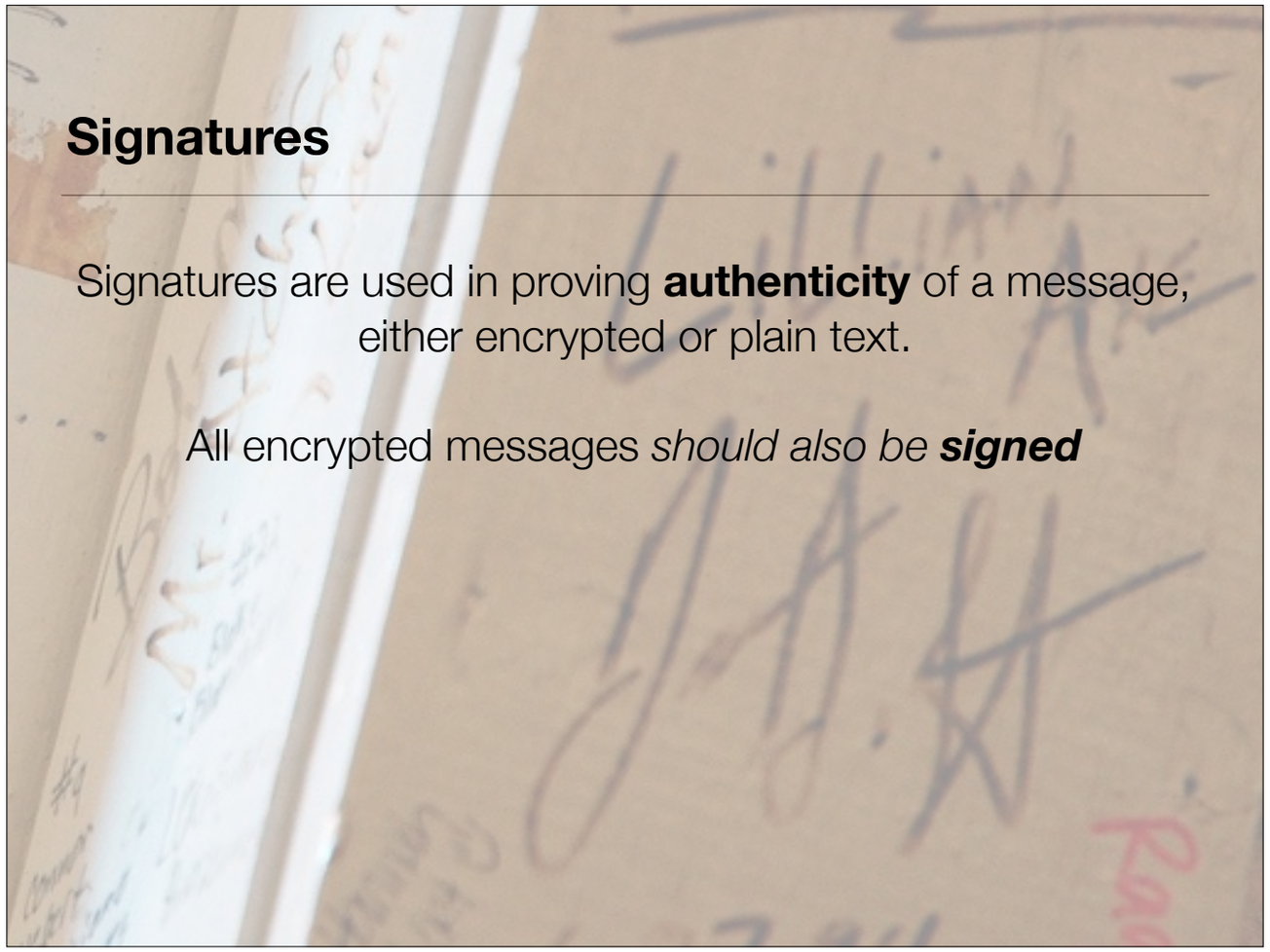
Fingerprints

A **fingerprint** is a shorter string derived from and which helps identify a longer key

My GPG fingerprint is:

094A 590E 099D 4621 A7DB 440A 8425 DACF 4F19 5F87

Before you collect my public key - and save it into your “key database”, you should compare my fingerprint to the one that you will get with my public key or via a key server lookup.
I publish my fingerprint on my business card in order to hand contacts an out of band copy of the fingerprint. This goes a long way towards really verifying my public key and my messages to contacts



Signatures

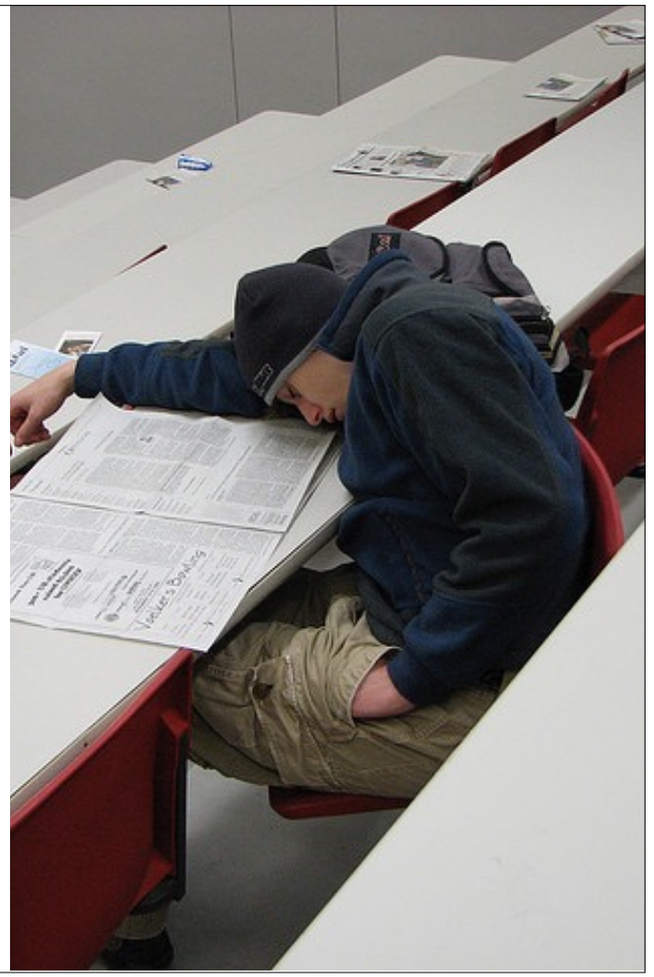
Signatures are used in proving **authenticity** of a message, either encrypted or plain text.

All encrypted messages *should also be* ***signed***

Many GPG front-end applications default to NOT signing each message. Why this is I do not know. It is less safe to send an unsigned message. We end up with an extra step for users to do - and many may not even know to do this.

Are your eyes glazing over yet?

Just wait!



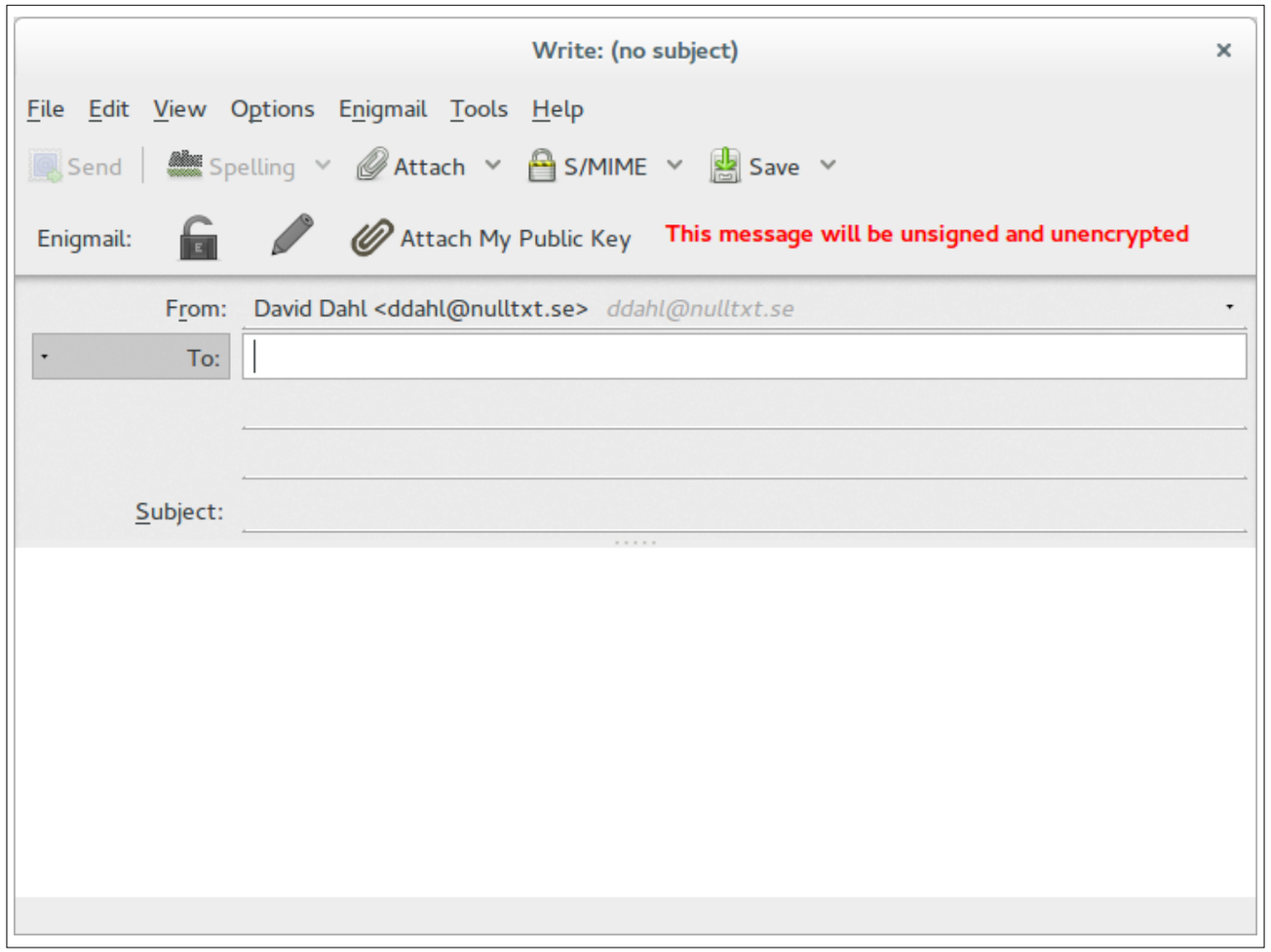
All these extra bits needed to make software more private need to be turned around and used - with new metaphors and mental models - to make the software *feel private but not be a chore to operate*

GPG Usage

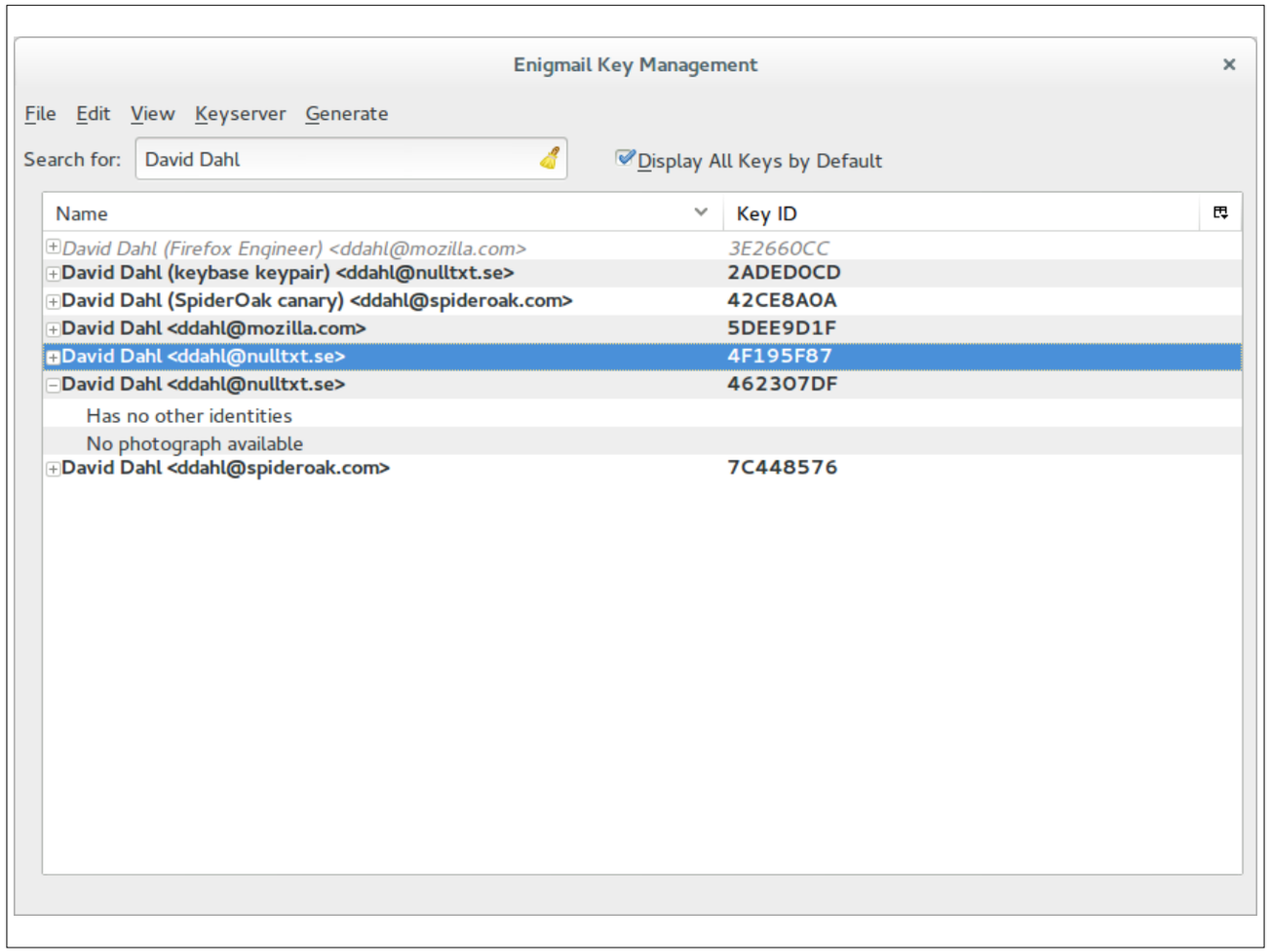
Enigmail on Thunderbird

- Install Thunderbird
- Install GPG
- Install Enigmail
- Try not to Fail
- There *is* Hope here…

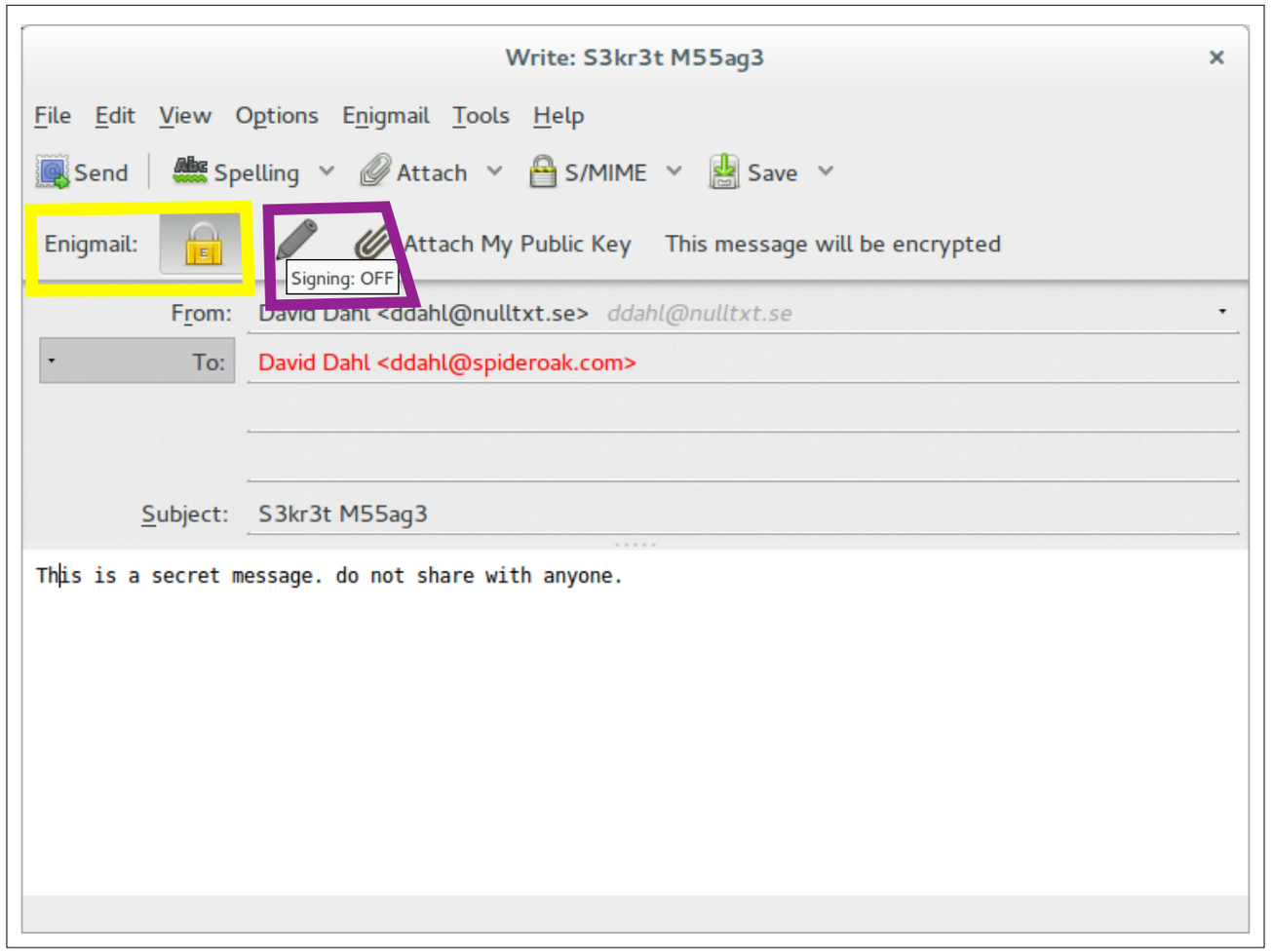
To get GPG email going - in this case - you have to install 3 different products! Then you walk through a messy “Wizard” to generate keys and publish your public key to a “keyserver”. After all of this you must be aware that by default, all email is sent in plain text and you usually have to be proactive about making sure your message is going to the right person using the correct key and the message is both *encrypted* and *signed*. Also, your subject line is never encrypted.



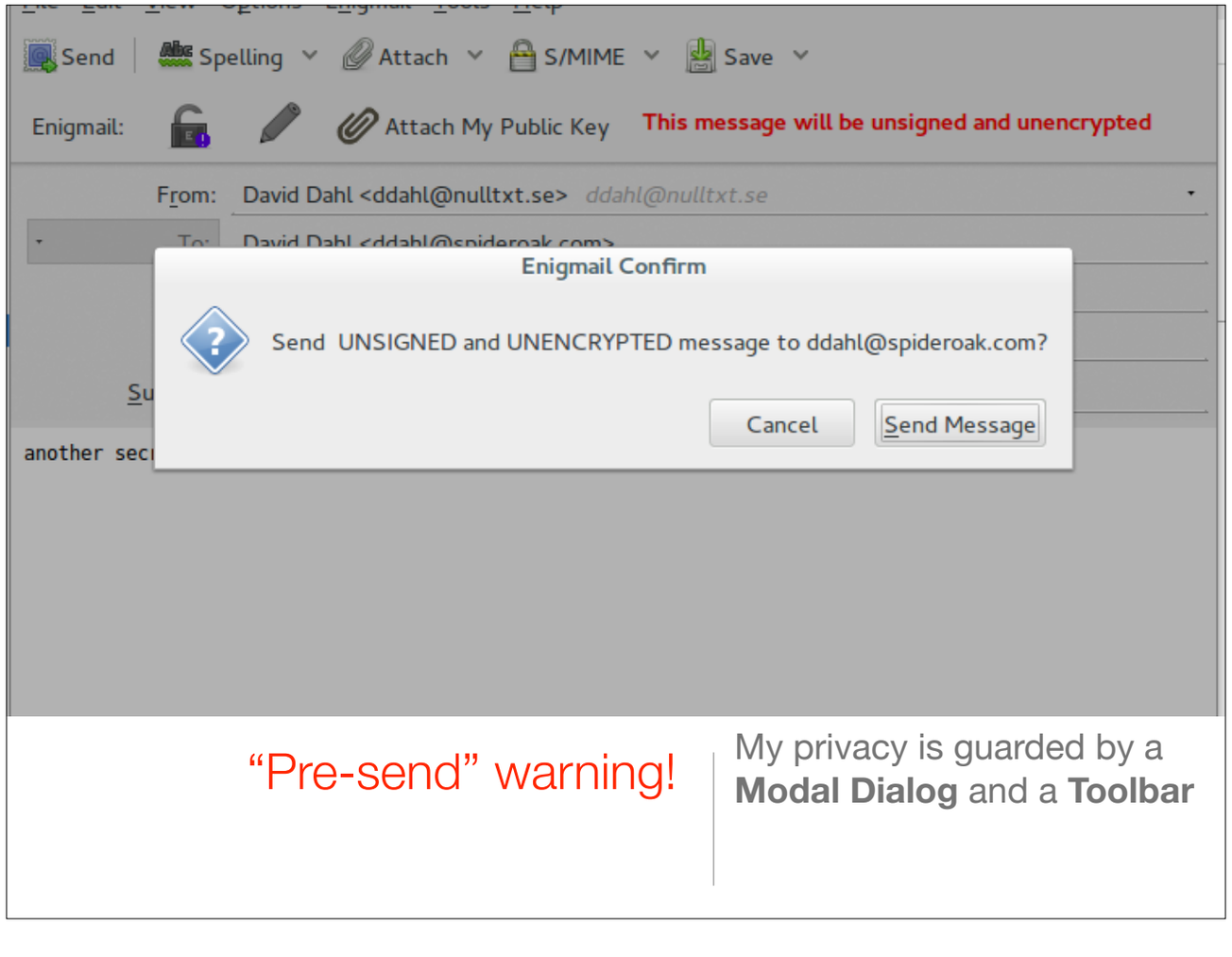
Lets send a GPG message via Thunderbird & Enigmail. Now, don’t get me wrong, I think this all works great - for me, a privacy nerd - but this is not going to work to make privacy universally usable. The first thing I notice here is that the message WILL NOT be encrypted & signed. That is OK, as most email is not, but this sometimes will allow you to send what you wanted private in the clear! Let’s pick a recipient - but I am not sure if I have the public key…



It turns out my recipient has more than one key! We have wandered into bad UX so quickly here. This is unusable by the vast majority of internet users.



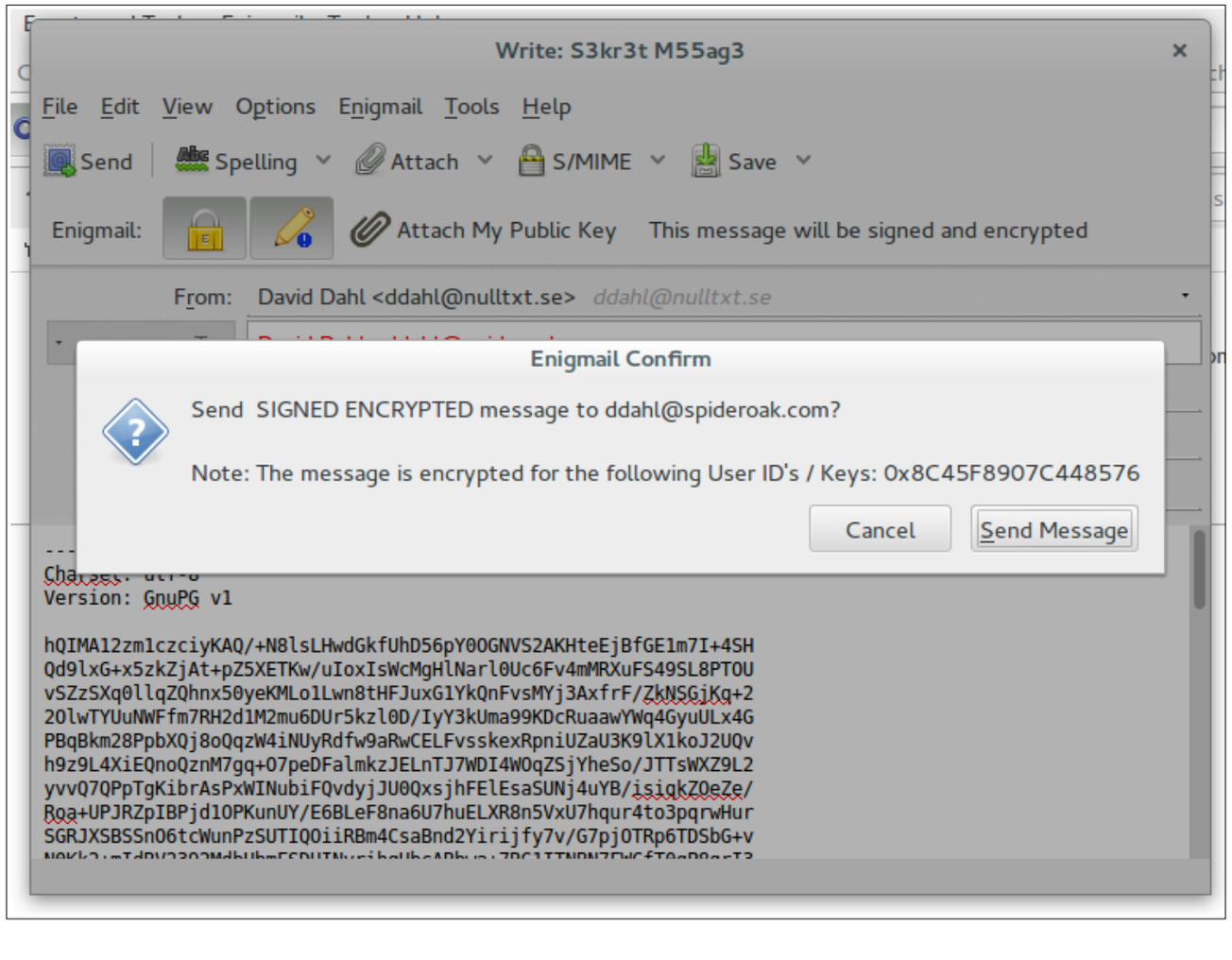
My default settings set the messages to not be signed when encrypting. This is bad default behavior.



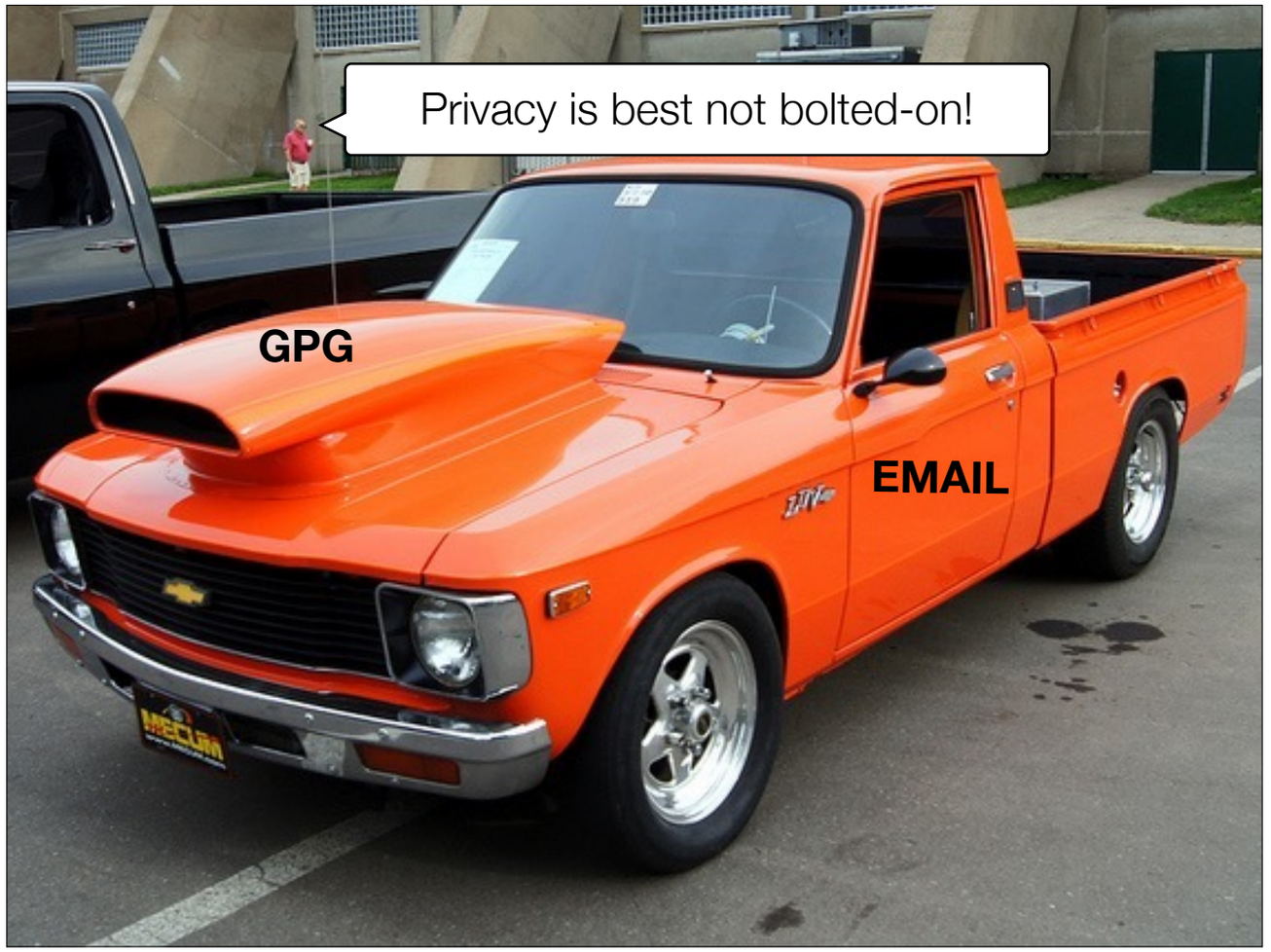
“Pre-send” warning!

My privacy is guarded by a **Modal Dialog** and a **Toolbar**

I’m not even kidding. This is cutting edge stuff. The app now interrupts your message sending to make sure its actually encrypted and a toolbar makes it easy to make sure all messages are signed as well! Thank you UX gods!



Now we are sending the message correctly after clicking on both the Lock and the Pencil. Normally, this works correctly, but this is perilous as email is designed to be clear text and we are bolting on some privacy measures. *Don't be in a hurry*. Again, this is unusable by humans.



We need applications that are built from the ground up with privacy at the core of the application. Bolt-ons do work, but are dangerous to rely on.

You can also use GPG as a general encryption tool for files and messages outside of an email program. This is how Snowden passed along his documents to Laura & Glenn.

We have ‘**Mobile** first’ development

And we have ‘**User** First’ development

We also need ‘**Privacy** First’ development

This is the future. Privacy First development

Greenwald nearly missed the story of a lifetime!

GPG fail to install even! THANK YOU **LAURA POITRAS**

Here we have a screen capture from “Citizen4”, the Snowden Documentary.
This is an extreme example, but Glenn Greenwald was **unable to even install GPG** in order to communicate with Snowden.
I highly suggest you see the movie.

Of course this talk is not about the UX to help **avoid NSA targeting**. You might as well tape your mobile phone to a Greyhound bus bumper, smash your computer and travel overland with **cash** and **false identity papers** to South America and become a **farmer** in the **Pampas**

(I did some research for a friend)

Avoiding some blanket wiretapping and dragnet surveillance is possible. Avoiding surveillance when personally targeted is probably impossible - and I am **not** advocating that is a possible:)

Side Note:

What is a likely threat model?

Advertisers are a threat

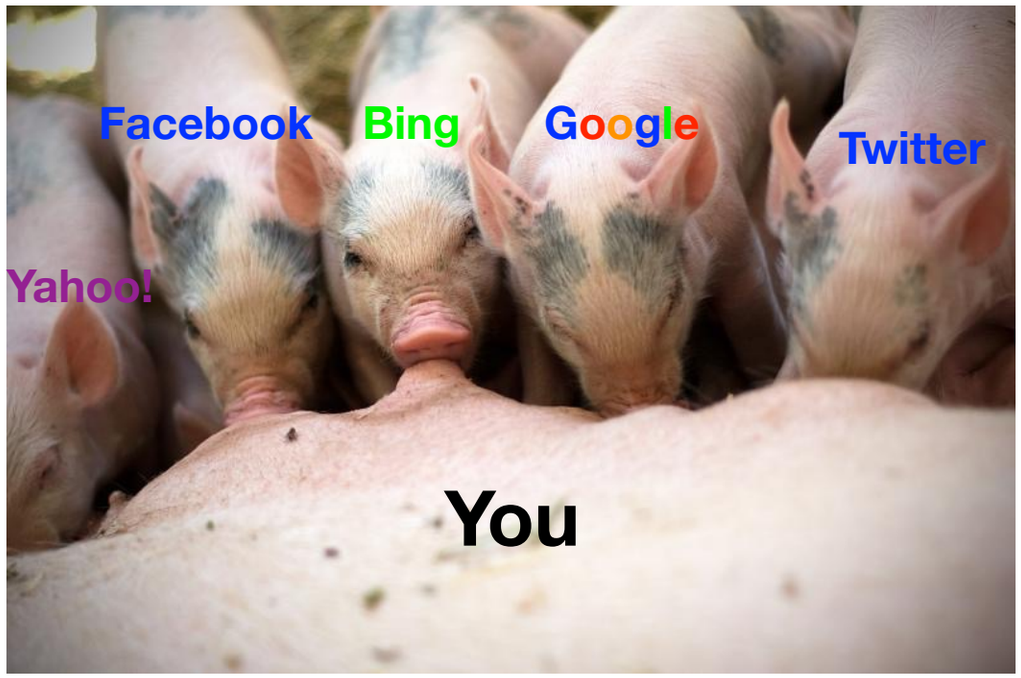
Ok, not advertisers, it is companies that

sell ~~**your data**~~ **you** to advertisers.

Like, *nearly* **everyone** in Silicon Valley

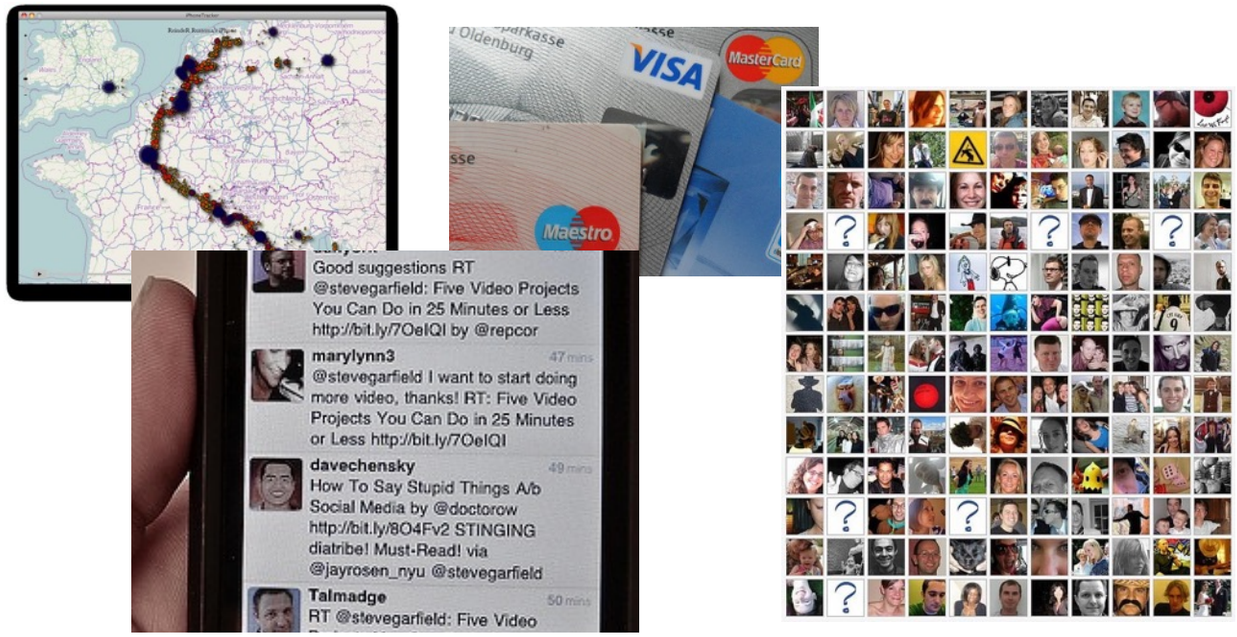
We really have to stop and think about what is happening here. What is the logical conclusion of all of the data being gathered and stored about you?

You are *not* the Customer
You are the **product**



We all may think these search engines and social networks are free software. They are not. The real reason they exist is to milk you of your private life and sell it to the highest bidder. This will cost you in the long run.

[Insurance company rates example: read about beers, go to many bars, car insurance rates through the roof. Buy too much junk food and watch A LOT of Amazon Prime or Netflix? Your Health insurance rates will climb.]



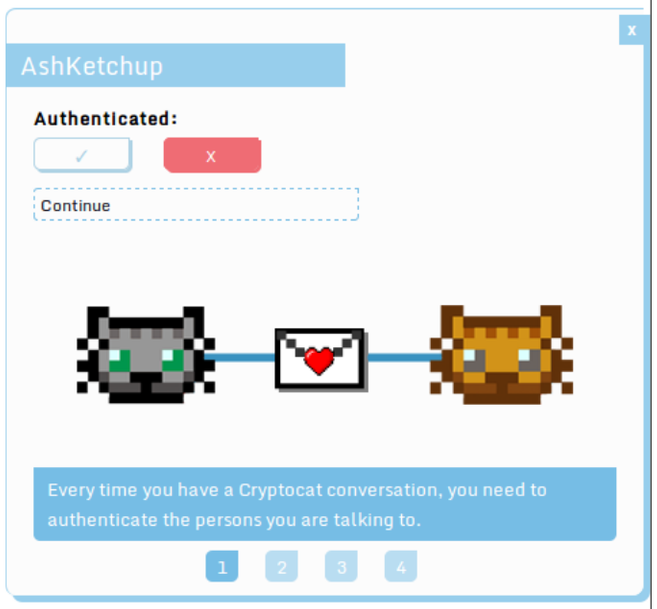
The real danger is the aggregation of all of our social networking posts, social graph, search history and location history. We are being sliced, diced and categorized into buckets of user types. Our most private conversations, opinions and movements are known to marketers and possibly worse. Even things we don’t search for are known to them. It is probably true that our search engines know more about us than the NSA and even our closest friends.

New-ish **Privacy Applications**

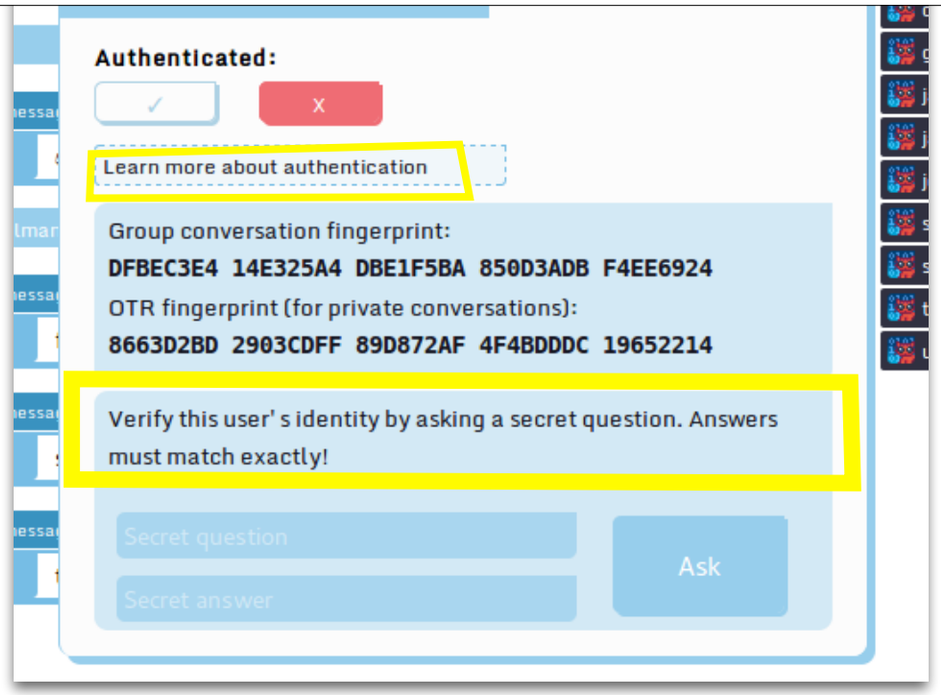
Ok, that was a bit of a side note…
Lets talk about some newer privacy apps and the potential issues we will run into

Crypto.cat

- Relatively easy to use
- Installation: Browser extension /Chrome App
- Obtuse verification process
- Great for real-time chat for groups



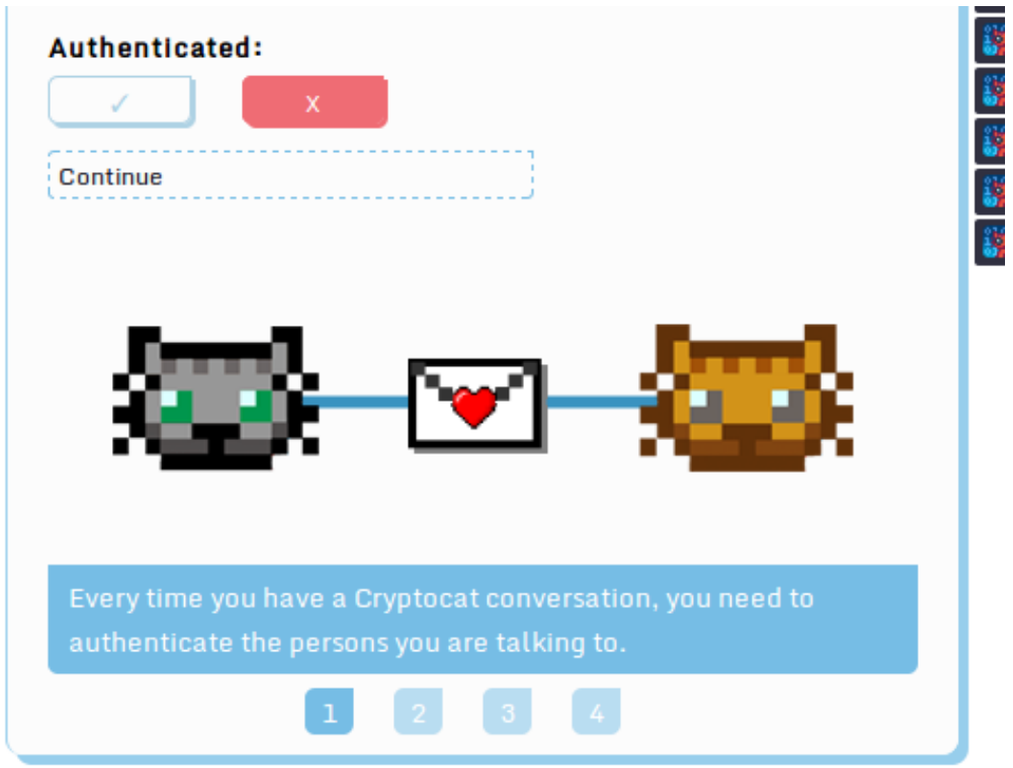
Overview: easy to use compared to GPG, provides instant messaging for groups. Still difficult to verify and authenticate others



We need to set up a **secret question** & **answer** for ***each person*** we talk to?

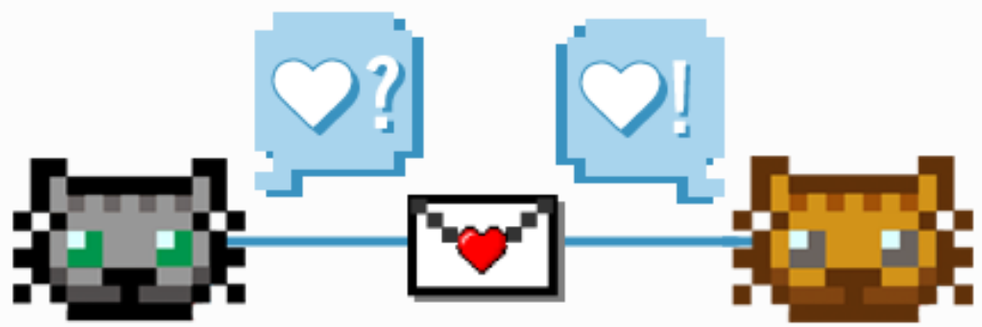
That's a lot of work

Since Crypto.cat allows you to use it anyway without doing this, it provides for bad habits and is just too much work for non-nerds to use



Crypto.cat docs are built in and very good. Regardless, it is a lot of work to use properly.

Continue



One way you can authenticate is by using Cryptocat to ask your friend a secret question that only they would know the answer to.

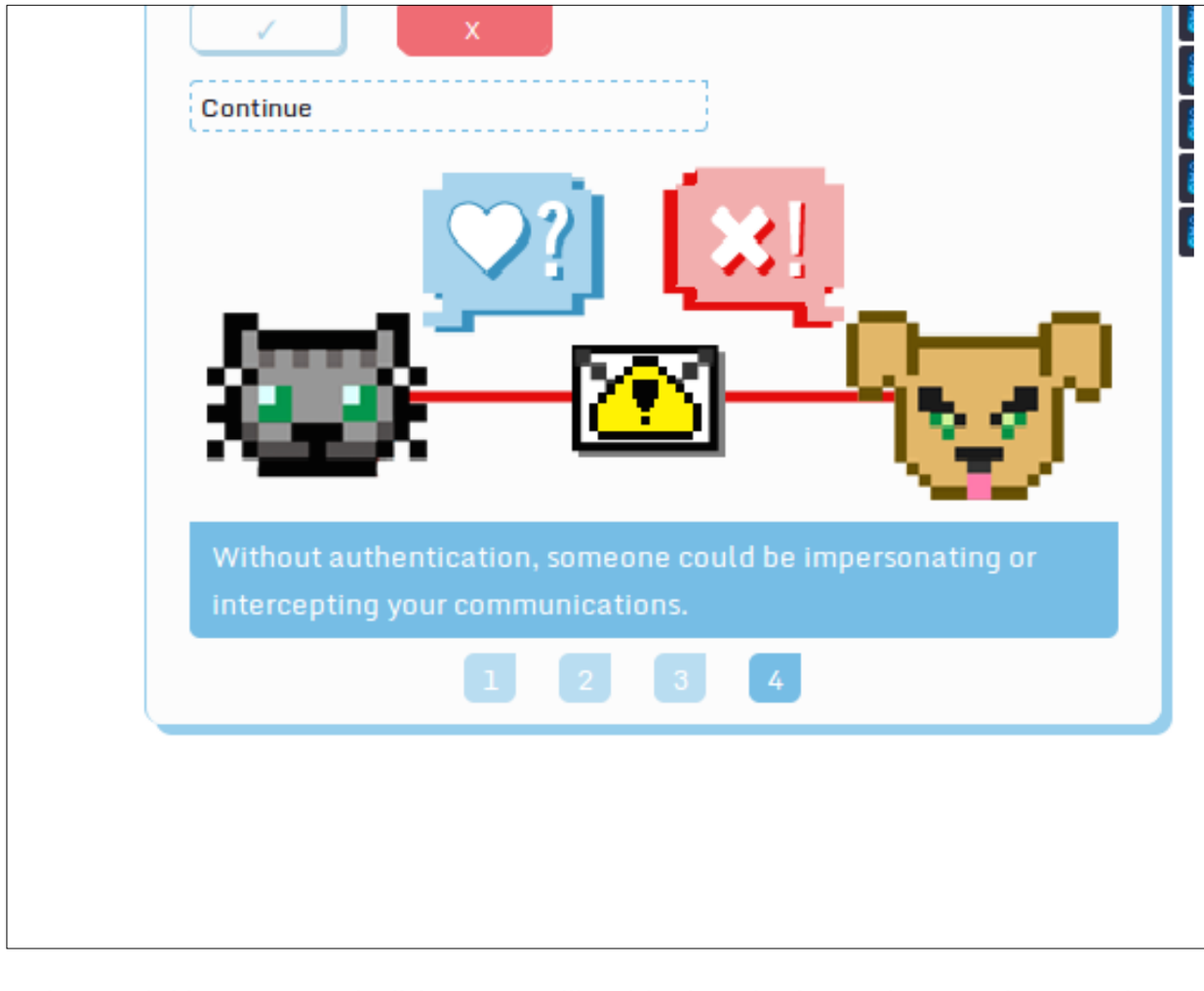
1 2 3 4

✓ x

Continue

You can also contact them via a trusted channel, such as by phone, and ask them to read their fingerprints.

1 2 3 4



While not ideal, this built-in documentation is good. How can we build systems like this that don’t need or require even less built-in documentation?

“UX is like a joke if you have to explain it you have done it wrong”

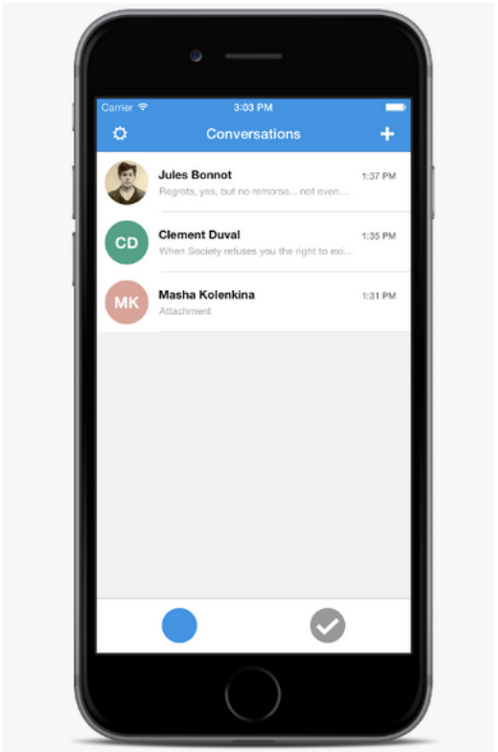
I don’t blame crypto.cat here, there are just doing what everyone else has done.

They are trying to make it fun and more humane and I applaud them for that.

Signal

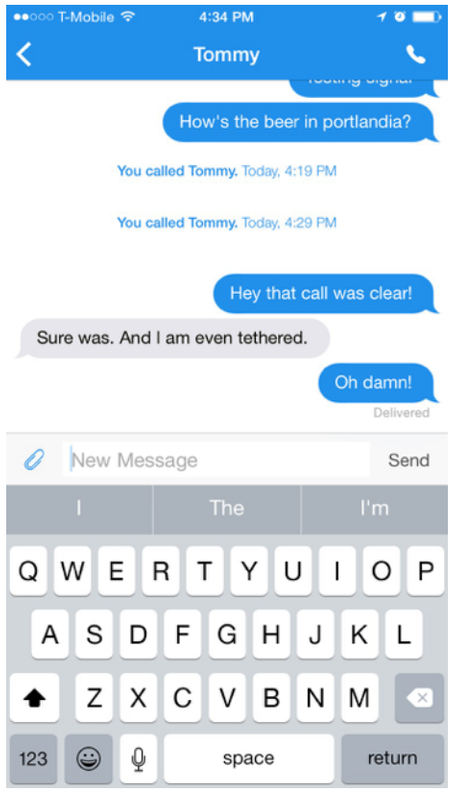
Probably the best experience in privacy: encrypted phone calls and texting

https://whispersystems.org/



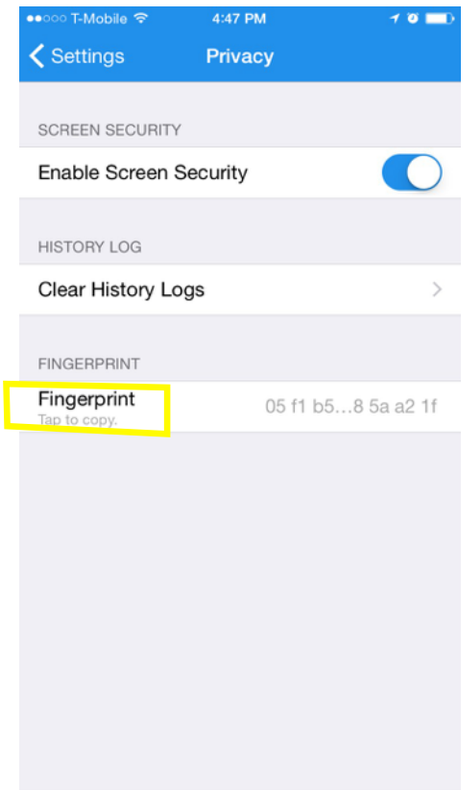
Started off as “text secure” on android, now on iOS and Android, its called “Signal”

Signal looks and feels like any other texting application but all texting and calling is end to end encrypted



Key generation is done behind the scenes. The contacts are verified over an initial SMS message that hands the other your public key and fingerprint

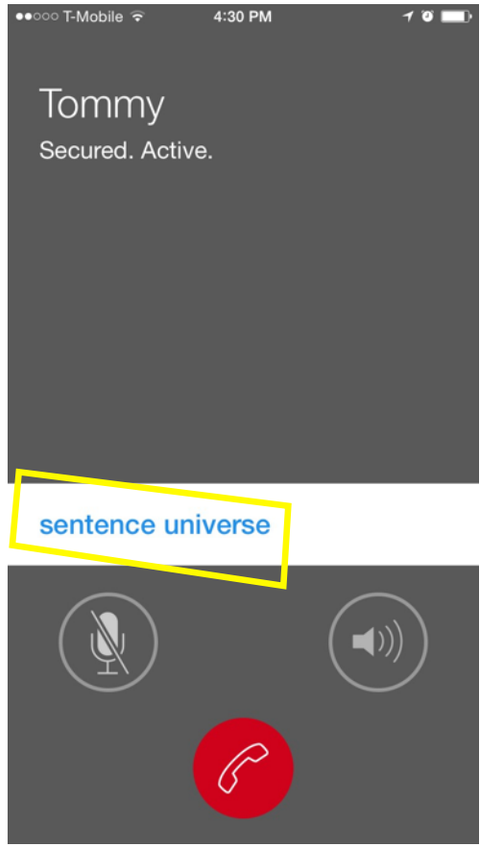
You still need to verify fingerprints.
“**tap to copy**” makes
it pretty easy
to get your own fingerprint
for others to verify
over a 2nd channel



you get a new phone and your key will change, so others will be warned when they are sent a new public key behind the scenes.

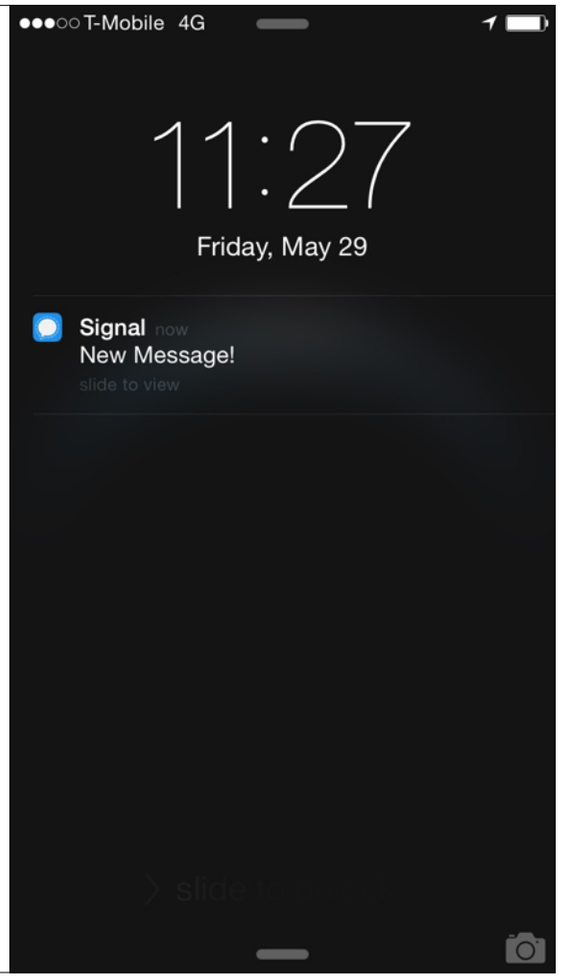
Encrypted calling is elegant

A phrase is used to verify
your caller is
who you think it is.
It should be
the same on each device



Encrypted calling in signal started off life as “RedPhone” for android, which was mentioned in that Snowden slide.
There is a lot to say about Signal. It is the next generation app for ease of use and slick UX for privacy. The crypto is pretty cutting edge as well, which uses “key ratcheting” for text messages. Each text is encrypted with a new key. Very slick and hard to attack.

A downside: notifications are generic



This is true of most privacy-leaning applications. Since notifications might pop up at any time - and when your phone is visible to others, a generic notification is all we can really do. Perhaps using sensors we can determine if the user is holding the device we can decrypt the actual notification. (On iOS, this may be problematic as background operations can be limited, apple’s concern about using too much CPU in the background)

Anonymity Online

- Tor Browser
- Ricochet (IM)

Anonymity online is possible. The best tool for this is the Tor Browser for web browsing, and tools built on top of the Tor network protocol. Ricochet is one of these applications and is used for IM.

My focus is on applications that are open source. Also, these are apps you can help improve the UX for. Apps that are closed source I have to wonder about the "privacy" being trustable.

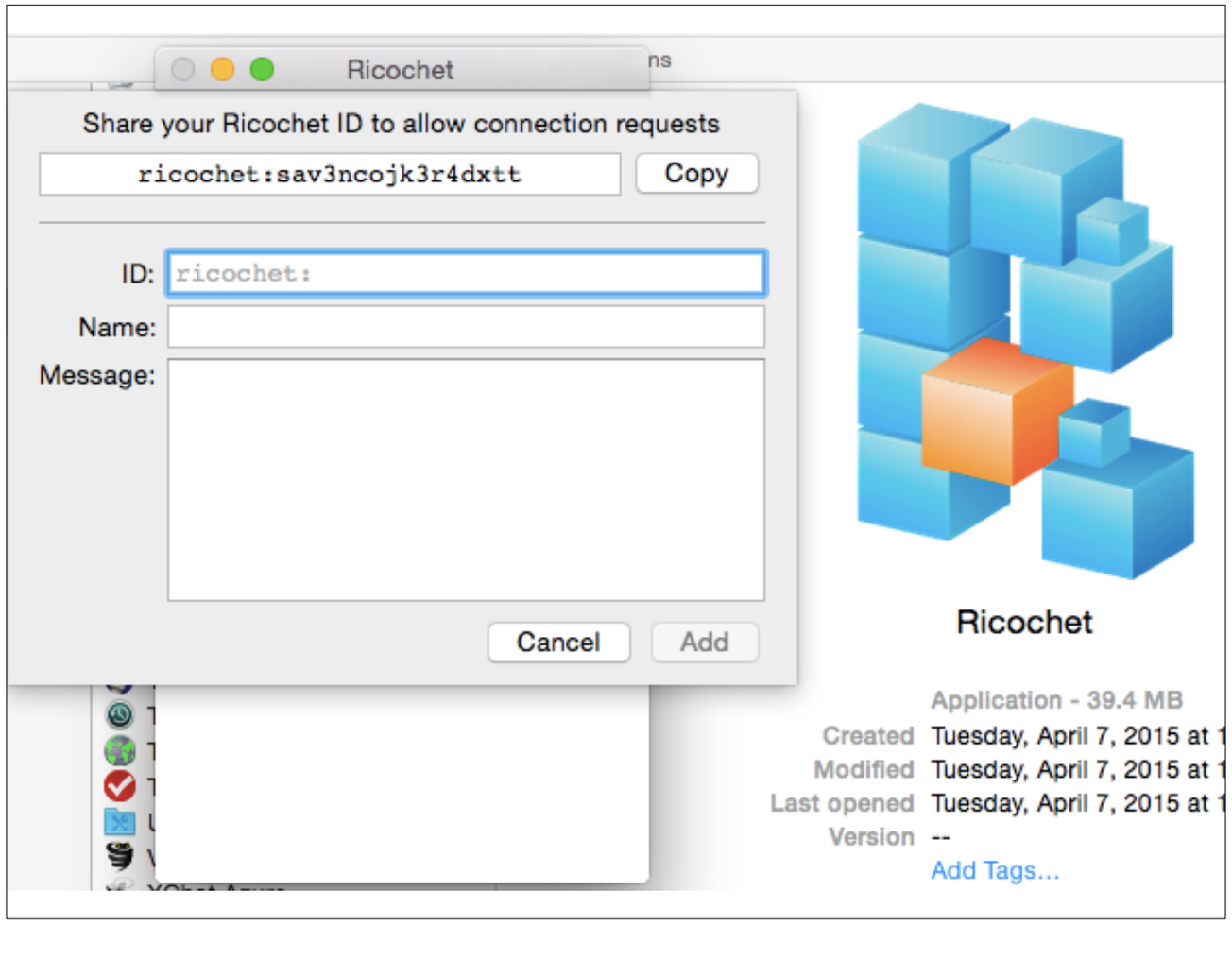
While Tor Browser is a bit slow, it does allow for real anonymity online. It is built on top of Firefox and through the Snowden leaks we have learned that **Tor does indeed work**! It is used heavily by dissidents and activists living under repressive regimes as well as many journalists that don't want any of their research queries tied back to them, etc.

Ricochet:

Anonymous
metadata-resistant
instant messaging
that just works

https://github.com/ricochet-im/ricochet/releases

Ricochet is a somewhat new anonymous chat tool that tunnels all of its communications through the Tor network. It is not the easiest thing to use but is anonymous, hiding all of its metadata well.



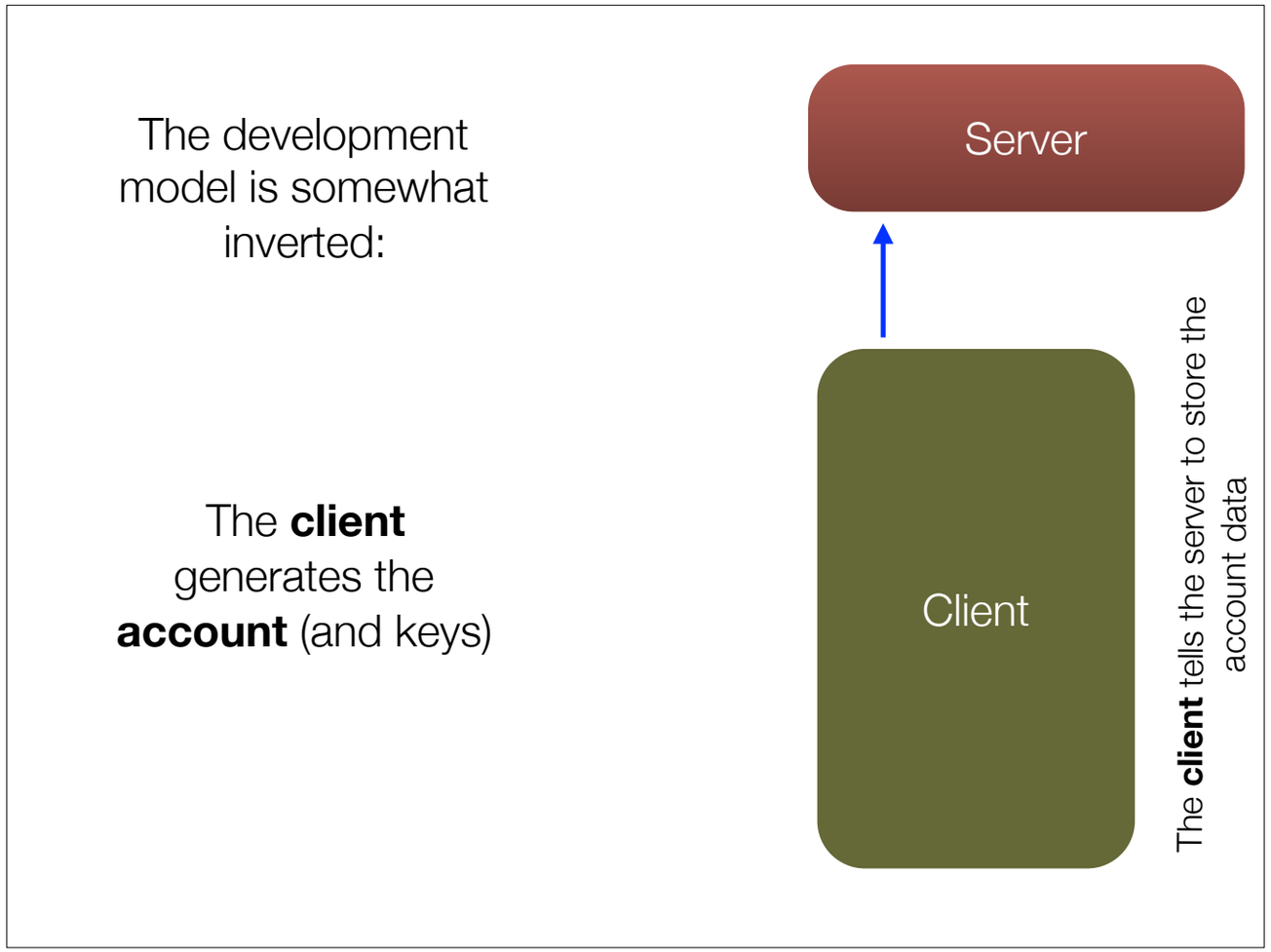
You exchange identifiers through another channel, perhaps even in person (which is best). Once you exchange IDs, you can chat anonymously with end to end encryption

Design for Privacy Applications

How do we do better?
Lets talk about the operating and design model for Privacy Applications
The client is the powerful actor. This turns things a a bit upside down. The source of truth is not 100% encapsulated in the server. Working with this model can be challenging and some use cases are difficult to implement.

The development model is somewhat inverted:

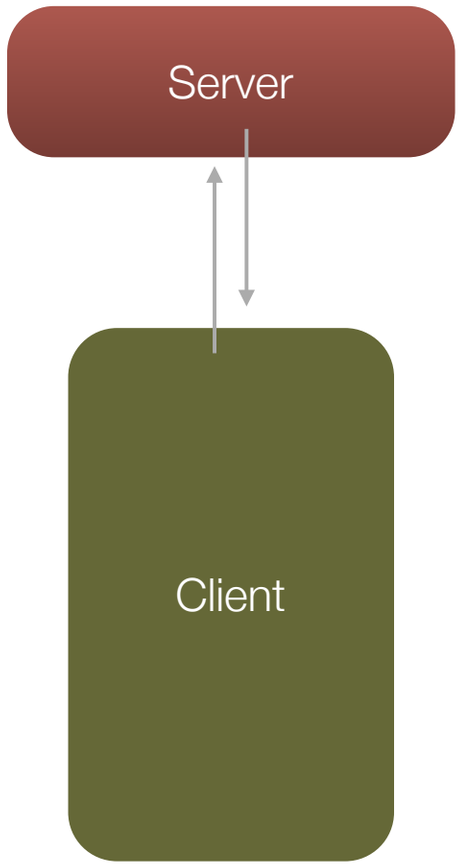
The **client** generates the **account** (and keys)

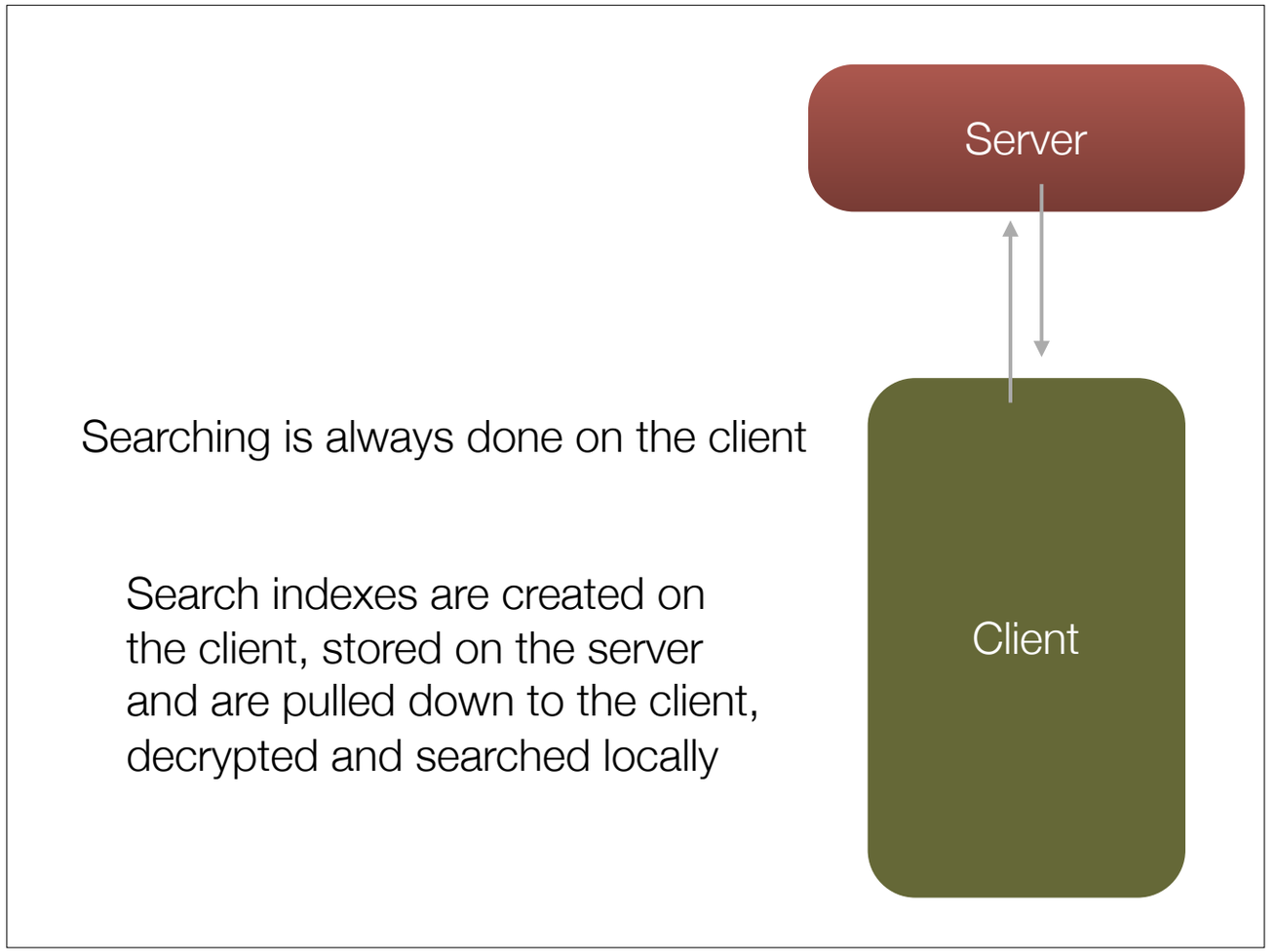


An inverted model makes for some very tricky operations. The client is the source of truth (it generates keys and tells the server what to store)

All communications data is encrypted on the client

And passed to the server, where the server is unaware of what the data is





Search is difficult as we must search the plain text on the client. Normally this data is indexed on the server and easily searchable there. This breaks what is known as “*Zero Knowledge*”. Indexed data must be searched on the client only. There are some advances in encryption technology where we can search encrypted data directly on the server, but this is not mainstream or easy to implement, it is still very much “research"

The UX of Privacy starts at the API

We need APIs that encapsulate the complex crypto operations that underlying libraries require. Developers need better “SDK UX” to be able to safely design and develop applications in a “privacy first” effort.

Crypton is an attempt at better UX for privacy-centered apps

Well, mainly better UX for APIs so developers can use crypto properly by NOT actually using Crypto!

It is not easy to use crypto APIs - and of course that much more difficult to IMPLEMENT crypto APIs. Crypton is a stab at an SDK any developer who knows JavaScript can implement privacy-centered applications with little knowledge of crypto. **Crypton** provides APIs that resemble normal app-building APIs but everything is encrypted end to end.

Sadly, there are few projects like this.

Crypton at a glance

```
crypton.generateAccount('username', 'password',
  function (error, account){});

crypton.authorize('username', 'password', function (error, session){});

session.getOrCreateItem('wineList', function (error, item){});

session.items.wineList.value.reds = [{cabernet: 'sonoma'}];

session.items.wineList.save(function (err) {});

session.getPeer('alice', function (err, alice) {});

session.items.wineList.share(alice, function (err) {});
```

Not to get too far into the APIs crypton provides, but Crypton's APIs read like any other web API. All of the crypto is encapsulated inside functions anyone can use, and all of the normal parameters that developers need to choose are set for you to "smart defaults" - that have been audited by professionals. This is UX *too*, making the framework APIs simple. The developer never deals directly with the encryption. When you call "getFoo()", decryption happens even before you are handed the plain text object.

The SAFE (Secure Access for Everyone) network can be best described as a fully distributed data management service. This network manages static and dynamic data as well as communications. Importantly the data held is either:

Encrypted by **clients**
Cryptographically **signed** by clients

MaidSafe is another project that is attempting to build a Peer to Peer network that by default does nothing but end to end encryption. The project is working on a beta version of its network now and has a lot of potential. MaidSafe will allow you to build most applications we use today where the network (everyone else's computer on the network) is the datacenter.

Top Privacy UX issues

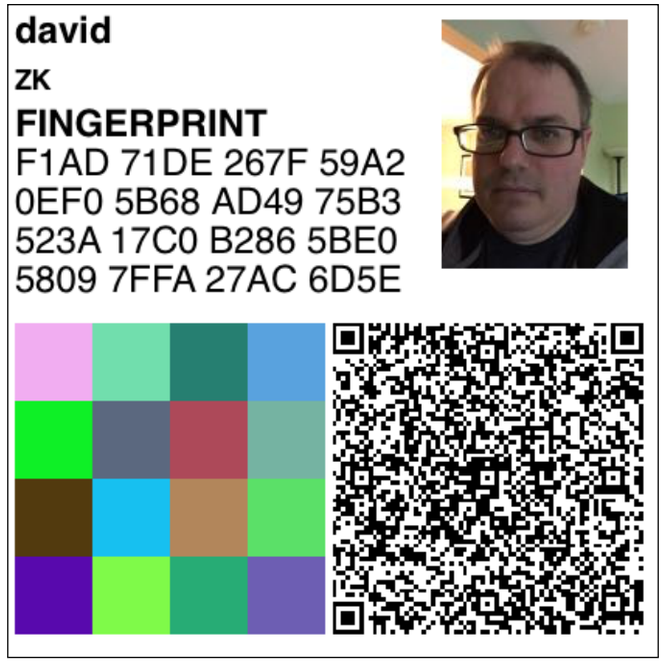
- Key Exchange
- Verifying others

The management of keys has always been a complex and confusing affair. Keys need to be exchanged to begin communication and again when keys change.
Verification of contacts is usually awkward, with one or more contacts wondering why it is necessary and be OK with using the application in an unsafe, unverified mode.

New Metaphors

New Metaphors need to be created to allow application users to better understand (or to completely hide the complexity of key exchange)

Crypton's Contact Card Concept

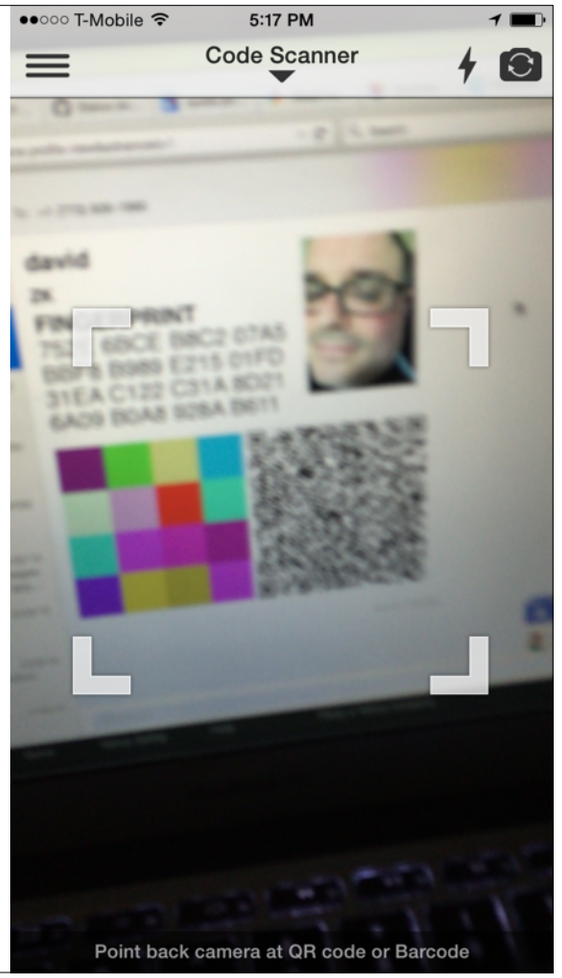


The Contact Card metaphor.
Everyone has an ID or Business Card. You show it to people so they can verify who you are, or learn a bit about you.
Eliminates specialized jargon from privacy: no "keys" are referred to.
An exchange of Contact Cards establishes a secure communication channel

Scanning or loading
the Contact ID compares
the fingerprint against one
queried from the server

A *match* adds the contact to
your contacts DB

This whitelists the scanned ID
and allows them to **“follow you”**



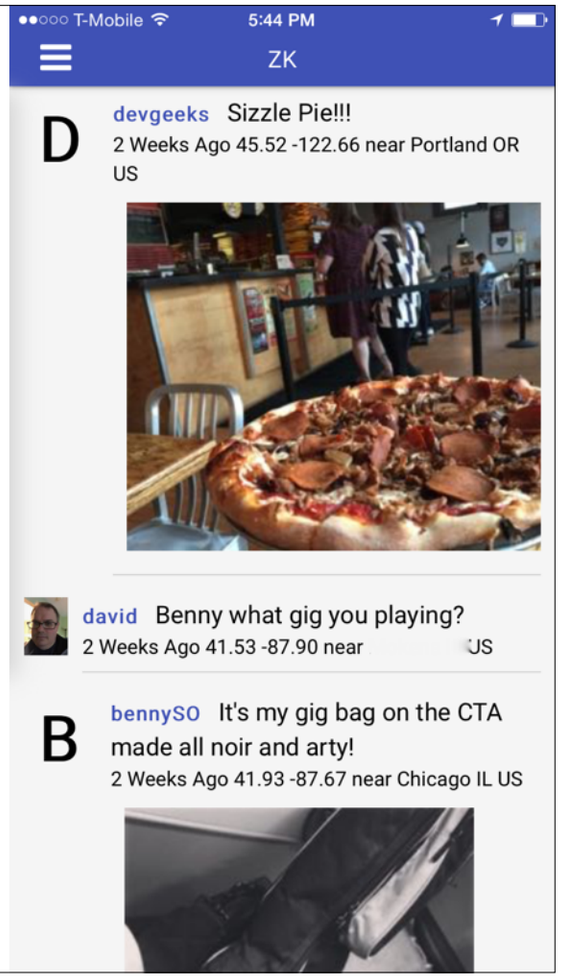
Also, we are in the process of simplifying the Card design as well as the workflow, it needs to be as seamless as possible.

My current project:

ZK: A “Zero-Knowledge Twitter”

Fun Fact!

ZK does “GPS to place name”
via a database lookup
instead of a “maps GPS API”



“ZK” is the code name for a Zero Knowledge twitter-like application I am building as a proof of concept of non-data-minable social networking. Users create accounts, exchange Contact cards (out of band via SMS, email) and are then following each other. A major difference here is that following is whitelisted. Bob must scan Alice’s card in order for Alice to follow Bob

CMU
Privacy Engineering Program

SpiderOak is currently working with a group of grad students in this program on a privacy UX project

The good news is that programs like Carnegie Mellon's Privacy Engineering Program exist, and is turning out a great group of multi-disciplinary engineers who really understand technology UX, crypto and humans

The students are helping to improve the Contact Card concept by doing UX research via focus groups and Mechanical Turk surveys.

Adopt a project!

I would like to see UX designers adopting crypto projects.
Its a huge challenge! *& you can learn while doing*
Crypto nerds need to pay attention to UX, to design, designers will blow their minds

Why should I (UX Designer Extraordinaire) do this?

WHY DO THIS?
[why should I as a UX designer pay attention to crypto communications systems?]

Future Demand!

Current & Future Demand!

Government and corporate spying is at an all time high. We are living in a “golden age of surveillance”. The web, our apps, our networks are all surveilling us.
All email probably routes through NSA, Chinese, Russian, corporate collection points.
The facilities NSA is building to house data can house a copy of everything digital we produce that interests them. They can store all of it forever, decrypting it once they get the keys or key factoring is possible.

End 2 End Encryption solves for (some of) the **Sony Hack**

(also, email should **die** in a **fire**)

All cloud systems that consume plain text are one tiny exploit from a 'Sony Email situation' or ‘The Fappening'.
End-2-end encryption systems make cloud storage and communications data complete garbage to an attacker. Imagine a messaging system a corporation can put in place for internal and external collaboration.
Naturally, with external collaborators this breaks down. It makes good business sense to build a communications tool that outside users can use, however, this is not going to be commonplace anytime soon.

With E2E encryption tools, attackers must attack the device

THIS IS EXPENSIVE

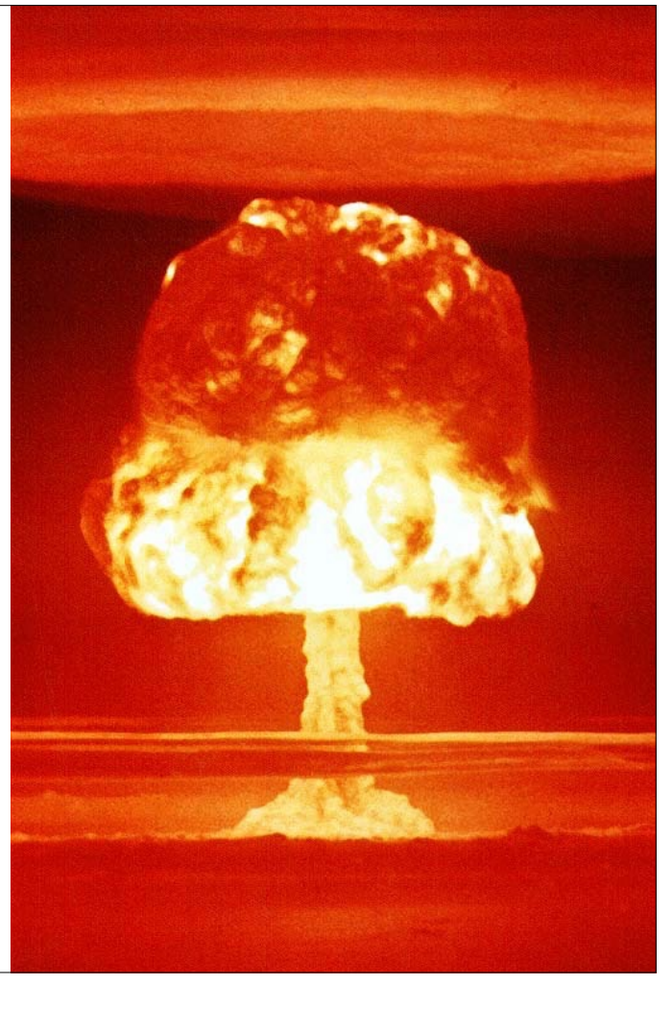
The target goes from 1 semi-well-protected network to hundreds maybe thousands of devices.
The economics of surveillance make is easy and cheap to do dragnet spying.
For instance:
@dymaxian, on Twitter: “We need to go from 10 Cents per user per day to $10000 per user per day to get state surveillance to the right scale”

The ‘Alpha Incident’

A privacy bomb has already gone off, but, like the nuke tests in the South Pacific, many are not at all aware of the damage done.

Just wait.

I’m talking about a wide-scale “The Fappening" with *your* junk!



There has yet to be what I call the "Alpha Incident"
This is when - overnight - a cloud communications company is hacked and 10, 20, 50 million people's message history is stolen and hosted in keyword-searchable format on a server in Eastern Europe or similar jurisdiction. People will begin to question why they use these systems.

Silicon Valley's Business Model

Real capitalism is when you offer a service or good to a customer and they pay you for it because it benefits THEM. Silicon Valley's model makes you the product and *you* are sold to companies. Your devices are logical extensions of you.

"If you have something that you don't want anyone to know, maybe you shouldn't be doing it in the first place."

- Eric Schmidt

'Some might say "I don't care if they violate my privacy; **I've got nothing to hide**." … Arguing that you don't care about the right to privacy because you have nothing to hide is no different than saying you don't care about free speech because **you have nothing to say**.'

- Edward Snowden

2 quotes here, I'll leave this as my last slide. If you are in the Eric Schmidt camp, you are sadly mistaken.