Secure Protocol for Accountable Warrant Execution. Supplementary.

November 3, 2016

Slides 2-4. Lets first consider a particular example where parties are making phone calls to each other, then we will see how this example can be generalized into a broader problem. Person A call person C and the calls goes through a telephone company, the telephone company keeps a record of this phonecall. Then perhaps person B calls person C and the telephone company keeps a record of that. Person D calls person A and so on. The telephone company stores records of these phone calls. Historically these records were used for billing purposes, now the data can be stored for various other reasons. For example telephone companies might want to draw statistics from these data for adjusting its marketing strategies. But we will be looking at another reason, when this data is stored to allow investigation by, for example, Federal government.

Slide 5. A real world example is when an investigator approaches the data carrier and asks for metadata or even data for a person of interest, the data carrier collects all the records it has about this person and hands it over to the investigator.

Slide 6-7. First let me share our view on the situation with warrant execution, so that you clearly understand our motivation in this project. The currently mandated data retention period is 18 months, 1.5 years [47C]. For telephone calls the carriers are required to store the names and telephone numbers of the two people, the date, time and length of each call for 18 months. This information is often called meta-data.
Surely, many civil rights organizations oppose to data collection (see e.g. “Petition to Repeal” 47 C.F.R. 42.6 (Retention of Telephone Toll Records) [Pet]).

The 4th Amendment protects “[t]he right of the people to be secure against unreasonable searches” [4th]. But metadata is not protected by this Amendment as it is not considered a “search” as was decided by a Supreme Court in Smith vs. Maryland, 1979. But in face the destinations of your phone calls may tell a lot about your health problems, let alone your private relationships, hobbies or even your location as was explored by Stanford’s researchers [Met]. Therefore is a lot of controversy in these requirements. A large number of data breaches made similar kind of information available to third parties. Several times, companies who stored this information and were breached have faced lawsuits. To name a couple:

- Target has agreed to pay $10 million to settle a lawsuit filed over the company’s 2013 data breach where the debit and credit card information of about 40 million customers was stolen [Tar].

- Home Depot has agreed to compensate its customers affected by a summer 2014 data breach whose debit and credit card information got stolen at the self-checkout lanes [Hom]. The settlement has been proposed in lawsuits.

By the law of California the companies are required to disclose the data breach to the Attorney General if it is believed that more than 500 customers have been affected [Cal].

To give you an idea, in 2015 the hackers have stolen personal data and background investigation records of 21 million former, current and prospective Federal employees [OPM]. This example shows how difficult it is to keep such data private. One may think that the Federal government who supports this data retention laws should be first to showcase how such data should be properly and securely stored, yet themselves they fail to keep it secure. But in fact its not their fault or the fault of the data storage entities in general, but rather there it is a fundamental problem that makes these breaches possible and the problem is that the data is stored in the clear or even if encrypted the secret keys are stored together with this data at the same place, thus creating a single point of failure.

Problem #1: The first problem that we are trying to solve in this project is that the data is stored in the clear of even if it is
encrypted the secret keys are often stored together with the data, i.e. at the same place.

**Slide 8-9.** A few more words about the state of the art in warrants executions. The federal government can issue a National Security Letter requesting access to the metadata information of an individual. NSLs permit the FBI to target credit agencies, telephone and Internet companies, financial institutions, and travel agencies. Often this letter does not require a prior approval from a judge, although the recipient of the NSL may still challenge the order in court. (In March 2006, the USA PATRIOT Improvement and Reauthorization Act allowed for judicial review of an NSL. Other amendments also allowed the recipient of an NSL to inform their attorney about the request and the government had to rely on the courts to enforce compliance with an NSL.) You may wonder, how many NSLs are being issued and how many of them are challenged. According to EFF for the period of 10 years, from 2003 to 2013 over 300,000 NSLs have been issued and only 7 were publicly challenged [EFF]. In 2013 President Obamas Intelligence Review Group reported; that the government continues to issue an average of nearly 60 NSLs every day [Pre]. So the rate is pretty high here.

**Problem #2:** The second problem that we are trying to solve here therefore is that cour’s approval is not required for execution of a warrant.

**Slide 10-11.** Often the NSL contains a nondisclosure agreement and even criminal penalties are provided for disclosing the existence of the warrant to any third party, including the service provider’s users.

In general there is currently no way to learn if you were investigated even after the fact, may be you were cleared and you already faced prosecution, you have no rights to find out what portion of your data was revealed. And that is because nobody is required to keep records for executions of warrants. Some companies used a method called canary watch [https://canarywatch.org/](https://canarywatch.org/), where the company informs users that they were not investigated; however if the NSL letter is received for a particular user the user will no longer see the same message. And so if he is really paying attention hell know he has been under investigation. Many companies still publish transparency reports on how many requests for information do they receive each year and number of requests satisfied, without giving out concrete names. Here are a few
examples:

- Tumblr https://www.tumblr.com/transparency

But still there is a problem here in that if your data was disclosed nobody is required to keep a record of that.

**Problem #3:** So the third and the last problem that we are trying to solve is that there are no audit records of data accessed. We still want to keep the nondisclosure requirement as it may be essential to the investigation process, but we want to keep all the audit records for all data accesses and make it possible to obtain them after the fact.

**Slide 12.** The goal of our project is to solve these three problems. Namely we wanted to design a system that would enforce a clear, just and correct workflow in warrant execution.

At a high level we wanted the data to be stored in a secure way, so that no secret keys or data in the clear is held by data carriers. We want the system to have authorized access, so that every access requests needs to be authorized by the court. And finally we want transparency, we want to enforce log records of all data accesses to allow after the fact verification.

**Slide 13.** Here is what design we propose at a high level. Lets first focus on three participants: the data carrier, the investigator and the court.

The data carrier will store the meta-data information in the following format. It will store in the clear who made the call and when at a coarse level (for example the month but not the time) and the rest of the metadata will be stored encrypted. With this approach if the breach happens it will be leaked how many phone calls did the person make, but to prevent that we can always introduce padding and append some dummy empty messages to our database, so thats not really a problem. So essentially the attacker who steals the database will learn nothing.

So how exactly will the records be encrypted. The court will be responsible for creating the proper keys, the court will create a public key and a secret key. So the public key is used for encrypting the information and the
secret key is used for decrypting it, this is an asymmetric procedure, you have different keys for encryption and decryption. And given the public key, that is the encryption key, it is hard to learn what the secret key is. So the data carrier will use the court's public key for encryption.

So the investigator approaches the data carrier asking for metadata for a person of interest, the data carrier sends back the encrypted data. Now, the investigator has no idea what data is there, because it is encrypted and only the court holds the key to this encryption. So the investigator has to talk to the court. The investigator forwards the data to the court together with a justification on why does it need this data and the court if approves decrypts the data and hands it back to the investigator.

I want to stress that in our system we are not making it illegal for the investigator to ask for data without the approval from the court, but it makes it impossible to obtain such data without courts approval.

With this simple design we already solved the two out of three problems: we have no mandatory data or secrets at carriers, the court’s approval is required for access. Now, let’s see how to add audit.

**Slide 14.** A fourth participant in our protocol would be an auditor. The auditor will use an append-only database to keep the log.

The court after receiving the decryption request from the investigator will forward the request to the log and the log will acknowledge the recipient of this request. How exactly? It will sign what it has received and send the signature back to the court. In this case, the court will be able to verify that the signature came from the auditor and the court will be able to use this signature as a proof that the auditor has received the message to log.

We need to add a few encryption layers to the messages to ensure that the auditor does not see the requests and the court does not see the actual data, so the design is actually slightly more complicated and for details please look at the paper.

But this design ensures that every decryption request is logged, that the log learns nothing and that the log allow after the fact audit. And we’ve now have all three problems solved.

**Slide 15.** Full picture. At this point you may notice that the court holds all the secrets and therefore constitutes a single point of failure, to avoid that we could either use a secure hardware module to hide the court's secret key.
in it, or alternatively we can distribute trust, that it is distribute the secret key to multiple entities. One of these entities could be an EFF, others could be other governmental or civil rights organizations. The secret key will never be present at one place and we will need all the trustees in order to decrypt. Actually the system can be made flexible regarding how many trustees are required to decrypt and we tune it to set specific thresholds. The challenging task we faced here is how exactly to make sure that the trustees decrypt the request without knowing the request itself and how do they make sure that the request is logged before they decrypt. To solve this problem we design a new cryptographic protocol which we term auditable oblivious tranfer (aOT) and you can find more details in the paper.

**Slide 16.** We have to make some assumptions here, because clearly if all the parties behave maliciously we have no security. So we will assume that all the parties are honest, but that they can be curious. Meaning that they will follow the protocol correctly, but can later put some more effort into analysing the messages that they received during the execution. And we assume that at most one party is malicious. By malicious I mean that the party does not follow the protocol and can deviate from the protocol in an arbitrary way.

If these assumptions hold, then the crypto that we use guarantees that the records are secret from everyone, except the data-carrier initially (before it encrypts them), the investigator (after it got a proper authorization from the court); neither the court nor the trustees will see the records in the clear during execution! Next, we guarantee that the investigator can justify any records that it holds by pointing to the corresponding entries in the log. And finally if there is malicious party (as I told you one party can be malicious), the logs when decrypted will reveal who this party is.

**Slide 17-18.** Should be self-explanatory.

**Appendix.** The actual data-retention time is much longer than required by the law. According to US news [Act] (as of May 2015):

- Verizon Wireless: 1 year retention
- AT&T: 5 years (according to spokesman Michael Balmoris)
• T-Mobile: 10 years (spokesman Viet Nguyen says)

• MetroPCS: 2 years

• Sprint: 1.5 years (spokeswoman Stephanie Vinge Walsh says)

• U.S. Cellular 1 year (spokeswoman Katie Frey)

With the growing demand for privacy it is anticipated that the company claiming to guarantee more privacy and shorter data retention periods may win large portion of the market.

A few highlights from the workshop on “Policy, Law and Technology in the Current Crypto Wars” at the Law School at Stanford.  

Jennifer Granick answering the question of Dan Boneh on how can help in educating law professional told a personal story from one of her first cases. She was presented with an evidence that one person was accessing another person’s computer in the form of logs that were printed out on white paper, there were apparently cut and pasted from the full list of logs. She approached the judge and said that she cannot trust this paper, because it is just cut and pasted, the judge gave her a weird look and said: “What do you mean?! With scissors?”.

Very interesting facts from the discussion with the Honorable Stephen W. Smith, Magistrate Judge of the Southern District of Texas:

• He said a lot of warrants that he is presented with do not have an expiration period, if the warrant is granted a person can be spied on forever. Not always there is an expiration date for the non-disclosure requirement of the warrant too. He personally try to enforce agents to add concrete expiration dates.

• He says that a federal agent brings a warrant in person on paper and that the warrants aren’t scanned or digitized they are stored on paper. He has a large red stamp that says “DENIED”, he says using this stamp is one of the most pleasant highlights of this job.

• Always he says he has hard time understanding exactly what the warrant entitles, as sometimes it requires computer search or installing malware, or monitoring network traffic. He says he is very reluctant
to give a warrant for breaking into a computer as there might be a lot of other information unrelated to the investigation. Often he invites the federal agent in order to ask questions about the exact nature of the technology that is being used, and often the agent himself can't answer these question, because he is not a technician and there are other people or departments who were developing the technology.

References


