Categorizing, Analyzing, and Managing Third Party Trust

Andy Sayler

University of Colorado, Boulder

Abstract

The modern computing ecosystem requires users to trust a variety of third parties to complete even the most basic of digital tasks. From system administers to service providers to device manufactures, which third parties computer users must trust and the capabilities with which they must trust them is a critical component underpinning the privacy and security of our digital data. The rise of the "cloud" as the preferred platform for most modern computing applications makes questions of trust even more complicated and pressing. A lack of understanding or misplacement of such trust has the potential to lead to data leaks, questionable surveillance practices, and a wide range of related privacy-harming events.

It is thus desirable from a public policy perspective to help individuals understand and control third party trust and to minimize the likelihood of such trust being violated. Toward these ends, this paper presents a model for describing third party trust and the likelihood of trust violations. It applies this model to analyze the nature of third party trust across of a variety of popular cloud services and uses it to categorize the common manners in which third parties violate this trust. Finally, this paper presents a number of proposed techniques, both technological and policy-based, to minimize the degree of trust users must place in third parties as well as to decrease the likelihood of violation of this trust.

1 Introduction

Over the last decade, computing has undergone a monumental shift from storing and processing data on individually owned personal computers to storing and processing data on cloud services owned by a multitude of third parties. This shift has generated many benefits: sharing data with other users is trivial, multi-modal communication between users is easy, and computing devices are largely ephemeral and easily replaced without any significant loss of user data. This transition, however, has a significant side effect: user data is now stored in a manner where it is easily accessible to third parties beyond the user's immediate control. The shift from locally controlled data to third party controlled data raises a number of questions, especially with respect to whom users must trust in order to leverage modern computing services. Can users maintain the privacy of their digital data without having to trust anyone? One can imagine scenarios that maintain privacy without trust, but such scenarios generally involve only storing data on self-designed, built, and programmed devices that never leave one's possession. Such an arrangement is, at best, impractical for the vast majority of users, and at worst, simply not adequate to satisfy the demands of today. The range of manufacturers, developers, and service providers inherent in the modern computing landscape require that users make decisions regarding whom to trust at every step of any digital interaction in which they partake.

The popularity of the cloud model leads one to believe that most users are willing to trade the privacy and control afforded by traditional computing models for the convenience and features cloud-based services provide. Individuals regularly place their trust in third parties such as

Facebook, Dropbox, Google, and countless others to securely store their files, relay their communications, or process their data. But is this trust desirable and well placed? A 2014 Pew Research study found that over 90% of American adults feel that they have lost control over the data they store in the cloud; 80% are concerned about how cloud companies are using their data; and 70% are concerned about the manner is which the government might access their data [72]. Furthermore, the myriad of recently publicized data leaks at large companies (e.g. [6]) as well as ongoing government intrusions into third party user data stores (e.g. [47]) has propelled the debate over user privacy to new levels.

These facts raise a number of important questions. With which capabilities are users required to trust third parties? In what manners can this trust be violated? Is this trust an implicit necessity, or are there ways to reduce such trust? And finally, are there mechanisms that can reduce the likelihood of third parties violating a user's trust? This paper aims to address these questions in three parts. Section 2 presents a model for qualifying both *degree* of third party trust as well as *mechanisms* by which that trust can be violated. Section 3 provides an analysis of the capabilities users must entrust to third parties to use a variety of cloud services, as well examples of common classes of trust violations. Finally, Section 4 suggests mechanisms to allow users to reduce the degree by which they must trust individual third parties, as well as mechanisms for disincentivizing violations of this trust, all without limiting a user's ability to leverage modern computing services.

2 Modeling Trust

Researchers from a variety of disciplines have proposed a range of trust definitions and models [11, 31, 44, 76]. These models range from technical models for calculating reputations via machine-learning algorithms to sociological models for exploring legal and societal notions of trust. In this section, I propose a trust model for exploring the manner in which users interact with third parties across the modern computing landscape. In particular, this model aims to provide a basis for describing how users trust third parties with access to their digital data and the manners in which this trust might be violated.

Before defining a model for trust, it is useful to define some of the relevant terms used in this model. To start, I'll define *trust* as the expectation that a given entity will behave in a promised manner. *Violations* of trust thus occur whenever said entity deviates from this expectation. Trust is closely related to two other properties inherent in modern computing ecosystems: security and privacy. Like trust, these terms have wide-ranging meanings across a variety of disciplines. For the purposes of this discussion, I'll define *security* as the notion of user control over the behavior of a given system. A *secure system* is thus a system that behaves in the manner the user desires. Facets of this notion of security include *confidentiality*, the ability to control who can read user data, and *authenticity*, the ability to control who can modify user data. Finally, confidentiality and authenticity provide a definition of *privacy* as the ability to control both the access and modification of user data as well as the ability to control the meta-record of such access or modification.

When users leverage modern computing devices and services, they must trust third party manufacturers and service providers to be good stewards of digital data ranging from stored files to location information to communication messages. The nature of this trust has two main factors:

Degree: How much trust must a user place in a third party (e.g., what capabilities do they allow a third party to exercise with respect to user data)?

Violation: In what manners can the third party violate this trust (e.g., how can the third party abuse the capabilities they have been granted or why might they be inclined to do so)?

The security and privacy of a user's data is generally dependent on these two axes: the higher the degree of trust a user places in a third party, the more power that party has to subvert the privacy or security of a user's data. Similarly, the higher the risk of third party trust violations, the higher the risk of adverse effects to security or privacy. Intuitively, the best ways to enhance the security and privacy of user data is thus to minimize degree of third party trust, to minimize the likelihood of third party trust violations, or to minimize both.

2.1 Degree of Trust

Degrees of trust measure the capabilities a third party can exert over user data. I propose that third parties can be trusted with the following data-related capabilities:

Storage (S-Capability):
Can a third party faithfully store user data and make it available to the user upon request? Misuse of this capability may result in a loss of user data, but won't necessarily result in the exposure of user data.

Access (R-Capability):
Can a third party read and interpret the user data they store? Misuse of this capability may result in the unapproved exposure of user data.

Manipulation (W-Capability):
Can a third party modify the private user data to which they have access? Misuse of this capability may result in the ability to manipulate a user (e.g., changing appointments on a user's calendar, etc).

Meta-analysis (M-Capability):
Can a third party gather metadata related to any user data or a user's behavior interacting with this data? Misuse of this capability may result in the ability to infer information about a user (e.g., a user's friends).

While there are likely additional capabilities users can entrust to third parties, this collection represents the core set of data-related capabilities most commonly entrusted to cloud service providers.

2.2 Trust Violations

Trust violation occurs when a third party exercises any of the above capabilities without explicit user knowledge and consent. Put another way, a trust violation occurs whenever a third party leverages a capability with which they are entrusted in a manner in which the user does not expect the capability to be leveraged. I propose classifying such violations into four high-level categories. Each category is defined by the manner in which the violation occurs and the motivations behind it:

Implicit (P-Violation):
This class of trust violation occurs when a third party violates a user's trust in a manner approved by the third party. An example might be sharing user data with a business partner (e.g. an advertiser). Often these violations aren't really "violations" since a user may have clicked through a Terms of Service agreement that "granted" permission for such use, but if the third party is willfully engaging in behavior that the user would not generally expect, an implicit trust violation has occurred.

Compelled (C-Violation):

This class of trust violation occurs when a third party is compelled by another actor to violate a user's trust. The most common example would be a third party being forced to turn over user data or records in response to a request from the government with jurisdiction over the party.

Unintentional (U-Violation):

This form of violation occurs when a third party unintentionally discloses or manipulates user data. An example would be a coding error that allows unfettered access to user data. Traditional "hacking" attacks also fall into this class insofar that such attacks are often possible due to unintentional flaws in the design of a "secure" system.

Colluding (L-Violation):

This class of violation occurs when multiple third parties collude to gain capabilities over user data beyond what the user intended each to have individually. An example of such a violation might occur if a user has granted two separate parties access to different portions of user data (e.g., location data stored with their cellular service provider and credit card transaction data stored with their bank) that could be combined to reveal more about the user than the user intended either party to know.

While this list of violation categories is far from exhaustive, it does provide a good high-level framework for exploring the patterns underlying trust violations and potential methods of mitigation.

3 Analysis of Third Party Trust

The trust model proposed in §2 is primarily useful for describing the nature of user trust in the modern computing landscape. In this section, I apply the model to the analysis of both the capabilities users must entrust to a variety of popular third party services, as well as to examples of how that trust can, and has been, be violated.

3.1 Capability Examples

Third party based cloud computing services have become extremely popular over the previous ten years. The question of how *trustworthy* these services are is addressed later in §3.2. In this section, I explore how *trusted* such service are. That is, how much trust must users place in such services?

3.1.1 File Storage

Cloud file storage is a popular third party use case. Services such as Dropbox [19], Google Drive [38], and Microsoft OneDrive [63] all provide mechanisms for storing files in the cloud. These services allow users to sync files across multiple devices and provide the ability to share files with other users. Traditional cloud storage services such as Dropbox, Drive, and OneDrive are similar enough in their operation that I will use Dropbox as a stand-in for the analysis of all three.

What capabilities is a normal Dropbox user entrusting to Dropbox? Clearly, users must trust Dropbox to faithfully store their data since that is Dropbox's core purpose. Users therefore grant Dropbox the S capability. Furthermore, users must also grant Dropbox the ability to read and access their data (R capability) in order to support Dropbox's sharing and syncing features. While Dropbox doesn't generally utilize it, users are also effectively granting Dropbox the manipulation (W) capability as well since the user has no mechanism for ensuring that Dropbox can't manipulate their data. Finally, Dropbox has full access to user metadata related to their usage of the service, granting them the M capability. Therefore, Dropbox users must trust Dropbox with all possible data-related capabilities. Traditional cloud storage services are thus classified as "fully trusted" services: services that require the highest possible level of user trust. Such services are thus also in a position to do the greatest degree of damage to user privacy should trust in them as faithful stewards of private data turn out to be misplaced.

The level of trust required by traditional cloud storage services rightfully makes some users nervous or unwilling to use them. In response to such aversion, a number of systems have been developed with the aim of overcoming third party trust challenges in the storage space. Such systems include "end-to-end"[1] encrypted file storage services such as Tresorit [95], or SpiderOak [86]. These systems aim to limit third party use of the access (R) capability through the use of client-side encryption. Likewise, they aim to limit third party use of the manipulation (W) capability through the use of client-side cryptographic authentication.[2] In the trivial case where a user merely wishes to store data on a single device and not share it with others, these systems are fairly successful in achieving their desired trust mitigations. In order to sync data across multiple devices using such systems, a user must manually provide some secret (e.g. a password) on each device to secure its operation. While potentially burdensome and inconvenient, this practice aligns with these services' stated goals since it achieves multi-device syncing without requiring any additional third party trust.

The place where these systems falter is via their support for multi-user sharing and collaboration. Such services tend to accomplish multi-user sharing by acting as a trusted certificate authority (CA) in charge of issuing user certificates.[3] These certificates are then used with various asymmetric cryptographic primitives to exchange the necessary secrets for sharing files between users. Unfortunately, as a trusted CA, these services are capable of issuing fraudulent user certificates to themselves or other parties. This allows them to mount man-in-the-middle (MitM) attacks on any user trying to share data by impersonating the recipient of the shared data. This deficiency is discussed in depth in [110], and leads to a breakdown of such services' claim that their users need not trust them (at least when employing multi-user sharing). By mounting a MitM attack on a user trying to share data with another user, such service providers can regain the R and W capabilities they claim not to have. Furthermore, these services do little to mitigate their access to metadata (M capability). Nor do they provide ways for users to avoid data loss in the event that one of the services goes offline or shuts down (S capability).

"Secure" cloud file storage service such as Tresorit do more to minimize the required degree of third party trust then traditional services such as Dropbox. In single-user scenarios, such services generally succeed at reducing the degree of user trust from full (all four capabilities) to partial (only requiring the S and M capabilities). Yet when implementing multi-user use cases, such services fall back to requiring a more-or-less full degree of trust.

3.1.2 Social Media

Social media sites such as Facebook [27] or Google+ [43] are a popular class of cloud service. Such sites maintain a "social-graph" of connections between users, and facilitate communication and

[1] "End-to-end" cryptography refers to a style of cryptographic system where all sensitive cryptographic operations are performed by the end-user or client. Such systems aim to minimize the trust placed in third party services or systems.

[2] For example, asymmetric cryptographic signatures such as those provided by GnuPG [50] or symmetric cryptographic message authentication codes (MACs) available via a variety of algorithms [23, 24, 25].

[3] A certificate is a combination of a user's public key and identifying metadata signed by a trusted issuer.

sharing of pictures, events, and other data between users. Social media services are generally "free" to users – instead of charging users for the service, they monetize user data and interactions for the purpose of selling targeted advertising. Given their ubiquity in the modern Internet landscape, as well as their position as ad-supported services, it is useful to evaluate the trust profile of modern social media sites. Facebook is the largest social media site today, serving over 1.5 billion users as of 2015 [32]. As such, Facebook serves as a representative example of the variety of social media sites available today.

In terms of capabilities, Facebook, like Dropbox and other traditional cloud services, must be trusted with a full range of capabilities. Facebook is responsible for faithfully storing user data such as photos, videos, and messages (S capability). Facebook can read all data it stores (R capability), and indeed relies on the ability to read such data as the basis of its advertising-based business model. Facebook can manipulate the data it stores (W capability), and routinely does so for the purpose of curating user "news feeds" or even integrating user pictures into targeted ads [107]. Finally, Facebook is capable of applying a range of meta-analytic techniques to derive additional data about users (M capability) for the purpose of targeting ads and curating content.

Other social media sites such as Google+ require similar levels of trust. Since all mainstream social media services operate on ad-supported business models, there are business-related barriers to reducing this level of trust where doing so would also reduce third party access to user data. Thus, unlike in the storage space, there are not many options for alternate "secure" social media platforms that specifically aim to minimize third party trust.

3.1.3 Communications

Communication systems ranging from email and chat to voice and video calling are another popular set of third party services. The privacy and security of these systems are a matter of great public concern, and indeed many of the current privacy and security related legal battles revolve around the ability to communicate in a private and secure manner (e.g. [13, 47, 58]). Communication systems range from traditional services such as Gmail [40] to recent privacy-enhancing services such as TextSecure [61].

Email services such as Gmail [40] or chat services such as Hangouts [42] represent a fairly traditional approach to third party cloud services. As was the case with Dropbox and Facebook, users of such services must rely on the third party service provider (in this case, Google) to properly store (S capability) their messages, and the design of these systems does little to prevent the service provider from accessing (R capability) or manipulating (W capability) user messages.[4] Furthermore, since all communication flows through the service provider's servers, these providers have access to a range of potentially revealing meta-data about their users (M capability).

The need to place a high degree of trust in various third parties in order to leverage digital communication services has long been a concern amongst the users of such services. Indeed, many early privacy-enhancing software projects, including the venerable PGP [112, 113], were created in response to the lack of privacy inherent in most digital communication systems. Modern implementation of such systems, such as those conforming to the OpenPGP standard [10], aim to reduce the amount users must trust third party communication providers by adding end-to-end encryption and authentication support to traditional digital communication channels. The OpenPGP protocol can be applied atop mail traversing traditional email systems such as Gmail [39], as well as to messages traversing chat applications such as Hangouts. When used with such services, OpenPGP provides a level of trust mitigation above and beyond what is possible to achieve via the native

[4] Similar to Facebook, many communication services are ad-based, and thus the service provider often relies on their ability to access user data as the basis of their business model.

services themselves. In terms of trusted capabilities, a user employing PGP atop a traditional third party cloud service such as Gmail minimizes both the third party's access (via encryption) and manipulation (via authentication) capabilities. In such a scenario, only the end users involved in a given communication, and not any third party through which that communication might pass, have access to the necessary cryptographic keys required to read or alter the message. The third party, however, can still capture metadata (M capability) about the communication since metadata is outside the scope of the message content that PGP secures. The third party is also capable of dropping or deleting the communication all together, and thus still possesses the S capability.

Due to the numerous challenges and deficiencies associated with using OpenPGP-based systems [8, 45, 109], developers have created a number of alternate secure communication protocols. These protocols aim to provide forward-secrecy, metadata privacy, deniability, contact authentication, and message encryption and authentication for (primarily) real-time communication such as instant messaging and chat systems. Examples of such protocols include OTR [69] and OTR-derived protocols like TextSecure [61]. The TextSecure protocol (or a variant of it) is used by several apps such Open Whisper System's Signal [67] and Facebook's WhatsApp [108]. TextSecure uses various types of asymmetric cryptography to provide users with end-to-end encrypted and authenticated messaging capabilities. Use of TextSecure denies the access and manipulation capabilities to any third party through which TextSecure messages might pass (including the TextSecure server itself). Furthermore, TextSecure makes efforts to secure metadata from third party actors, including the TextSecure service provider itself. These efforts curtail a third party's ability to analyze message metadata.[5] TextSecure users are still dependent on a third party to operate a TextSecure server in order to communicate in the first place (it is not a decentralized protocol), but beyond this "storage"-like capability, TextSecure grants no other capabilities to any third party.

Following the trend set by other cloud services, communication systems such as Gmail (and email in general) or Hangouts (and related unencrypted chat systems) require users to place a high degree of trust in the corresponding service providers. Overlay privacy-enhancing systems such as those leveraging the OpenPGP protocol allow users to reduce this level of trust by employing client-side cryptography to limited third party use of the access and manipulation capabilities. Modern full-stack, privacy-focused communication protocols such as those employing flavors of the OTR protocol take these privacy-preserving cryptographic techniques a step further by limiting third party metadata access in addition to limiting third party data access and manipulation.

3.1.4 Password Managers

Password management systems are commonly used by those wishing to both manage and increase the security of the credentials they use to access various digital services. Such programs are useful for helping end users remember passwords, and by extension, for encouraging users to use stronger (i.e., longer and/or more random) passwords [9, 52, 80]. Since cloud-based password managers potentially allow third parties access to sensitive user credentials, it is worth evaluating the trust users much place in such services.

LastPass [55] is one of the most popular cloud-based password managers. Using LastPass, passwords are encrypted by the client and then stored on LastPass servers. Each password is encrypted using a key derived from a user-supplied "master" password. LastPass never stores this master password directly, making it difficult for them to derive the key necessary to decrypt

[5] It is still possible for a network-level adversary or the TextSecure server provider to discover the raw network metadata (e.g., IP addresses, etc) involved in a TextSecure exchange, but higher level details are not available. It is possible to couple TextSecure with existing network anonymity systems such as Tor [17] to mitigate such network-level meta-analysis [56].

the encrypted data they store. Thus, LastPass intentionally limits its access (R capability) to user passwords. LastPass does not, however, appear to perform any kind of cryptographic authentication on the data it stores, meaning it still has the ability to manipulate (W capability) user data.[6] Similarly, LastPass is responsible for faithfully storing user data and has full access to all user metadata associated with any stored password. Therefore LastPass requires users to trust it with three of the possible four capabilities – less trust than cloud services such as Facebook or Dropbox, but more than is strictly necessary to perform its password storage duties.

Other open-source password managers such as KeePass [74], Password Safe [81], or Pass [18] aim to reduce the need to trust third parties. Such systems accomplish this by either requiring no third party support at all (e.g. a purely local password manager)[7] or by allowing the user to decouple client-side encryption and authentication operations from optional third party backend data storage providers such as Dropbox. In addition to limiting third party access capability via encryption, such services often aim to limit both manipulation and metadata capabilities via the use of client-side cryptography.

3.2 Violation Examples

Capabilities measure the degree of trust users must place in third party services. But how likely is that trust to be violated? In this section, I present an analysis of the motivations behind certain classes of trust violation as outlined in §2. I also provide examples of specific trust violation events that have occurred over the past ten years.

3.2.1 Implicit Violations

Implicit trust violations represent the most direct form of trust violation. Implicit violations occur when a trusted third party intentionally misuses a capability in a manner the user did not intend. As the most direct form of trust violation, implicit violations also present the simplest analysis of motivations regarding such violations.

One of the clearest potential incentives for companies to commit implicit trust violations comes via the advertising-supported business models employed by many cloud services [26]. In these models, the user is provided with access to a cloud service for “free”. The service provider monetizes their service either by selling advertising space on the service directly, or by collecting and selling user data to third party advertising firms. In contrast to more traditional mass media advertising schemes, cloud services are often designed as platforms for highly targeted advertising. That is, cloud services can leverage the vast amount of data to which they have access as a mechanism for building detailed dossiers on each user, and then use these dossiers as the basis for serving personally tailored ads. Advertisers are generally willing to pay higher prices for more carefully targeted ads, incentivizing cloud providers to harvest user data in pursuit of such targeting.

While such advertising practices do not inherently represent an implicit trust violation, they do set up a series of perverse incentives where companies can benefit by leveraging the access (R) or metadata (M) capability to harvest user data. Most ad-supported cloud services require the user to agree to a terms of service that grants the service provider the right to collect user data for advertising purposes. But it’s well known that few, if any, users actually read such terms, leading to situations where users are surprised by the way in which their data is used [48, 62]. Thus, while

[6] Such a lack of client-side cryptographic protections against modification leaves the door open to a range of potential attacks on LastPass’s client side encryption as per the “cryptographic doom” principle [60].

[7] These purely client-side solutions limit third party trust, but do so at the expense of usability – e.g., such solutions rarely provide users with the ability to easily access their passwords from multiple devices or to share passwords with trusted colleagues.

the user may have technically "agreed" to certain advertising practices, it is still reasonable to fault a service provider for having committed an implicit trust violation in situations where their use of user data deviates from what would be generally expected.

The Target Corporation provides an example of an implicit trust violation triggered by ad-motivated misuses of access to user data. In 2012, it became public that Target had developed a statistical system for predicting if its shoppers were pregnant based the items they bought. Target leveraged this data to send customers coupons tailored toward pregnant individuals. In one case, this practice lead to the outing of a pregnant teenager to her previously unaware father [49]. Clearly such outcomes are not within the realm of what most shoppers expect when purchasing items at Target. Facebook committed a similar ad-motivated implicit trust violations when it began to incorporate user-provided images into the ads it served to other users [107]. These actions caught many users by surprise, as one does not normally expect one's personal photos to be re-purposed to endorse third party products.

Not all implicit violations are tied to the kinds of perverse incentives data-driven advertising often elicits. Sometimes third parties simply make poor decisions about the manner in which they use the capabilities a user has granted them. One of the more infamous examples of such misuse comes from Facebook's ability to manipulate (W capability) user newsfeeds. In 2014, it came to light that Facebook had engaged in research that involved manipulating what users saw in their news feeds in order to study the effects of one user's emotions on others [37]. The "emotional contagion" study was performed on $\approx$ 700 users without their knowledge or consent. Facebook misused the trust placed in it by its users to faithfully curate their newsfeeds to instead manipulate these feeds in unforeseen and potentially behavior-altering ways.

Some cloud companies rely on charging their users for access to a given service, and are thus particularly disincentivized from committing implicit violations, the revelations of which might harm their reputation and business prospects. But such companies are not immune to committing implicit violations. For example, in 2014 ride-share app Uber [99] made headlines when it used the travel history of a number of its more prominent users to display a live user-location map at a launch party [83]. This map allowed party guests to track these users in real time – an outcome the average Uber user certainly does not expect when trusting Uber with access to their location data. Similarly, Uber also used user travel history data to compose a blog post detailing its ability to detect a given user's proclivity for "one-night stands" [70]. In both cases, Uber committed implicit trust violations by leveraging data it had about users in manners users did not approve of or intend.

3.2.2 Compelled Violations

While implicit trust violations are perhaps the most egregious form of violations, they are probably not the most pervasive. Instead, that honor likely falls to compelled violations. As discussed in §2, compelled violations occur when an entity other than the third party the user is trusting forces the third party to manipulate or provide user data in a manner not approved by the user. The most common form of compelled violation comes via government search and seizure powers. In the United States, such powers are often exercised via a variety of methods including subpoenas issued under the Third Party Doctrine [94], probable cause search warrants [102], National Security Letters [30], and Foreign Intelligence Surveillance Court (FISC) orders [101].

The scope of compelled violations can be partially evaluated by studying the transparency reports published by many cloud companies. Companies such as Dropbox [20], Amazon [2], Facebook [28], Google [41], and Twitter [98] all publish bi-annual transparency reports detailing the number and type of data requests they receive as well as the frequency at which they turn over user data in response to these requests. While the requests outlined in these reports are generally lawful,

Company	2011	2012	2013	2014	2015
Facebook	Unknown	Unknown	19292	23666	29694
Google	11413	14612	17749	18300	19180
Twitter	Unknown	1072	1179	2203	4060
Amazon	Unknown	Unknown	Unknown	Unknown	1733
Dropbox	Unknown	71	198	404	869

Table 1: U.S. Government Data Requests Resulting in User Data Being Provided By Year

and in some cases are likely important for protecting the safety of the public, turning over data in response to such requests without user permissions represents a compelled violation since users do not generally intend for third parties to provide their data to government actors – especially in cases where the user is given no notice or ability to contest the provision of such data. Table 1 shows the number of instances in which major third party service providers were compelled to turn over user data or metadata over the previous five years. As shown, the largest providers turned over user data on the order of tens of thousands of times per year. Furthermore, the number of compelled violations committed each year has steadily risen from year to year.

The high number of compelled violations likely represent a significant increase in the amount of user data being provided to government actors relative to pre-cloud computing times. While it is possible that governments could serve the same number of data requests on individual users were we to live in a world where most user data was stored locally instead of held by third parties, it seems unlikely that this would be the case. The concentration of user data in a handful of third parties greatly reduces the effort required by government actors to request access to it. Furthermore, while certain types of compelled legal orders (e.g., probable cause warrants) could be served on individuals instead of third parties, other legal orders (e.g., subpoenas served under the third party doctrine) would not be legally valid if served on an individual due to the higher protection locally stored data enjoys relative to third party stored data.

Beyond the kinds of direct requests for user data documented in published transparency reports, there are also several notable examples of governments seeking to compel individual third parties to modify their services in order to enable compelled access to user data. Lavabit provides an example of one such case from 2013. Lavabit was a private email service with 400,000 users premised on the idea that popular free email services such as Gmail lacked adequate security and privacy guarantees. In August 2013 Lavabit shuttered its service in response to a U.S. government subpoena requiring it to turn over all of its encrypted user traffic as well as the associated SSL encryption keys necessary to decrypt it [57, 58]. After a legal fight, Lavabit founder Ladar Levison was forced to disclose the encryption keys protecting his service.

Recently, a similar case arose between Apple and the FBI. In response to the 2015 San Bernardino shootings, the FBI attempted to compel Apple to help it decrypt one of the shooters' iPhone [3]. The form of encryption Apple uses to protect the iPhone involves a hardware-linked encryption key that can not be easily extracted from the phone, limiting out-of-band cracking opportunities. Furthermore, this key can not be used on the phone without a user-provided passcode. By default, Apple limits the number of guesses a user may make at this passcode and throttles the speed at which a user may guess passcodes. The FBI wished to compel Apple to update the software on the iPhone so that they could try to guess an unlimited number of passcodes at a high rate of speed [7]. Apple was disinclined to acquiesce to this request [13]. The case was dropped by the FBI after they were able to leverage an undisclosed security vulnerability to bypass Apple's passcode guessing limits directly [29, 51]. As in the Lavabit case, this case demonstrates the government's interest in compelling companies to assist them in accessing private user data, even going so far as to require companies to avoid the use of certain forms of encryption or security-enhancing features that would make such assistance difficult or impossible to provide.

Not all compelled trust violations inherently involve government requests for user data. Sometimes third parties may be compelled to turn over user data due to civil or business circumstances. In particular, it is not unusual for user data to be bought or sold in the event that a third party goes bankrupt [65, 84, 85]. Since the sale of such data in a bankruptcy or acquisition is often beyond the direct control of the third party holding the data, such data transfers represent a compelled trust violation.[8] While not as common as government-compelled data requests, such financially or civilly-compelled data access similarly represents a privacy and security risk to users of third party services.

3.2.3 Unintentional Violations

As mentioned in §2, unintentional violations occur when a third party violates a user's trust in a manner that they neither intended nor were forced to undertake. Unintentional violations can be broadly sorted into two subcategories: external and internal violations. External violations are caused by an external actor (e.g. an adversarial attacker) leveraging a third party's capabilities to cause a trust violation. Internal violations are caused by mistakes within the third party (e.g. a coding error) leading to a trust violation. Often, internal violations beget external violations – for example, a security bug caused by a programming mistake could open the door to external attacks that leverage the bug to expose user data.

There have been a number of notable internal unintentional trust violations committed by third parties over the past ten years. For example, in 2011 Dropbox introduced a bug into their authentication system that allowed anyone to log into the service using any password for a five hour period [22]. While Dropbox certainly did not intend to effectively share their users files with the entire world, they unintentionally did so via a coding error. In some cases, internal violations occur due to factors beyond the third parties direct control. For example, third parties are susceptible to a range of software flaws in the externally maintained libraries they rely on. Prominent examples of such flaws include Heartbleed [12], a flaw in OpenSSL [68] that allowed attackers to steal private data from secure web servers, and Shellshock [92], a GNU bash [73] flaw that allowed attackers to execute arbitrary code on certain web servers. Both flaws were widespread and effected large swaths of third party sites and services, potentially exposing the users of these services to data exfiltration or manipulation (i.e. violations of R or W capabilities).

While bugs like Heartbleed or Shellshock demonstrate that having publicly reviewable code is not sufficient for eliminating bugs, these bugs also demonstrate effective disclosure and patching processes practiced by open source communities. Had similar bugs been discovered internally in closed source code, it is possible they would have gone unreported and potentially exploitable for long periods of time. The difficulty of hiding bugs or ignoring publicly disclosed bugs in open source code has led a number of security and privacy enhancing software projects to specifically leverage open source models for trust related purposes. Projects such as GnuPG [50], Signal [67], and End-to-End [39, 111] all tout their open source nature as a mechanism for reducing the likelihood of both unintentional bugs and intentional backdoors. Such practices can be viewed as a form of trust mitigation since they allow the user to reduce the trust they must place in third party in favor of distributing trust across a wider community of reviewers. While open source implementations

[8] As opposed to the implicit trust violation that would occur if a third party willfully sold or shared user data in a manner the user did not expect their data to be sold or shared.

alone do not guarantee the lack of such bugs [33] or backdoors [93], they do help maximize the number of eyes on the code, making violations harder to hide.

Internal unintentional violations often pave the way for external unintentional violations. Take for example the recent OPM data breach. In 2015, the U.S. Office of Personnel Management (OPM) announced that their systems had been breached, exposing the personal data of essentially anyone who has held or currently holds a U.S. Government security clearance [34, 105]. This breach, in addition to having high strategic value to foreign attackers, reveled sensitive personal data of a huge number of U.S. government employees and contractors. This leak was largely due to the use of old and outdated storage and security systems employed by the OPM. While such usage did not necessarily directly result in the exposure of sensitive user data, it certainly made it far easier for an attacker to break in and steal such data. It is likely that a similar situation occurred in early 2015 when several major U.S. health insurance companies were subject to attacks that breached their user records, allowing the release of personal, financial, and medical information on millions of users [53, 54]. While the details of these breaches have not been made public, it is reasonable to assume that mistakes on the part of the third party storing the data paved the way for external actors to steal user data. Indeed, recent investigations of common data breach factors list errors on the part of third party data stewards as the leading cause of breaches [35, 106].

Some external unintentional violations occur not due to the fault of third parties, but due to a fault of users themselves. The most common example of such failures involve the use of weak passwords by users to protect their accounts on third-party services. Dropbox has been the target of various external trust violations mounted by adversaries who obtain and exploit common user passwords [21]. While it is tempting to not attribute these faults to third parties directly, third party service providers must shoulder at least some of the blame for allowing users to utilize weak credentials or similar error-prone authentication mechanisms. For example in 2014, a number of celebrity users of Apple's iCloud data storage service [5] were subject to a public release of personal photos they had stored with the service. This leak was the result of a targeted attack on the corresponding users' passwords and iCloud accounts [6]. These attacks appear to have been propagated over several months prior to the public release. While this leak was not a result of an overt flaw in Apple's iCloud system, the weak default password requirements for iCloud accounts made it relatively simple for attackers to compromise such accounts and steal user data.

Finally, some external unintentional violations occur due to an adversary's use of techniques that many third parities could not be reasonably expected to anticipate and defend against. Nation-state-level attacks generally fall into this category. While governments are often able to access user data by compelling third parties to turn it over, in some cases they prefer to attack the third party directly, triggering an external unintentional violation. For example, MUSCULAR was/is a joint National Security Agency (NSA) and U.K. Government Communication Headquarters (GCHQ) effort to intercept and monitor traffic traversing Google's and Yahoo's intra-datacenter networks [36]. Prior to MUSCULAR's disclosure, this intra-datacenter traffic was not generally encrypted or thought to be vulnerable, and thus was an ideal point for the government to intercept and monitor user data. The government, however, was able to utilize such far-reaching technique as tapping undersea communication cables or obtaining physical access to "secure" Internet exchange facilities in order to collect such data. These types of violations are often bootstrapped either via internal unintentional violations (e.g., exploitable bugs in a crypto algorithm) or via compelled orders (e.g., granting government access to the facilities from which to mount such attacks).

3.2.4 Collusion Violations

Collusion violations occur when multiple third parties work in concert to leverage or misuse capabilities in manners that would not be possible for each individually to do. Inherent to the notion of collusion violations is the notion of trust separation – i.e. the ability to spread trusted capabilities across multiple third parties, reducing the amount any individual third party must be trusted in the process. This fact makes examples of real world collusion violations harder to come by due to the fact that very few deployed systems require or even allow users to distribute trust in these manners.

Still, one can imagine how certain collusion violations might occur. For example, a password manager provider such as LastPass could collude with a mobile keyboard provider such as Swype [66] for the purpose of capturing a user's "master" encryption password and using it to decrypt the users stored passwords. Normally, LastPass lacks access to the plain-text variants of the user passwords it stores since these are encrypted on the client's device prior to being sent to the LastPass servers. Similarly, mobile keyboard software such as Swype does not normally possess access to a LastPass user's passwords since these are encrypted directly via the LastPass app. Therefore, neither Swype nor LastPass can individually access a user's password data. Swype does, however, have the ability to modify their keyboard software to record a user's typed input and report it back to a central server, and could collude with LastPass to capture and provide a user's master password. LastPass could then use this information to read the user's stored passwords even though the user attempted to limit such access via client-side encryption.[9]

While real world collusion violations appear to be relatively rare today as a side effect of the singular nature of most third party trust arrangements, this may change in the future. §4 discusses techniques for reducing trust in single third parties, the side effect of which may be a reduction in the kinds of violations that are common today, but an increase in the potential for compelled violations. It is thus worth keeping an eye on the nature and frequency of such violations in the future.

4 Managing Trust

The current trust situation inherent in using most cloud services – i.e. trusting third parties with a wide range of capabilities and only moderate disincentivizes to violating user trust – is far from ideal. This state places private user data and metadata at a high degree of risk for unapproved exposure or manipulation. It is natural to ask what solutions might aid in better controlling third party trust arrangements, reducing the degree of risk involved when leveraging third party services. There are a myriad of potential solutions in this space, ranging from technical to policy-based. In this section I suggest a few high-level approaches to managing third party trust and minimizing third party trust violations.

The trust model presented in §2 discusses two axes of third party trust: the capabilities we entrust to third parties and the manners in which this trust might be violated. Both axes can be targeted when seeking to increase the security and privacy of user data. By reducing the degree or trust – i.e. limiting the number of capabilities third parties are granted – users can limit the amount of harm a third party can inflict should this trust be violated. By disincentivizing the

[9] This example is a bit contrived for the purposes of demonstration. In reality, if LastPass wished to capture a user's master password, or otherwise decrypt user passwords, they could simply modify the closed source LastPass client to either record all user input or to backdoor the encryption mechanism. They would not inherently need to involve an additional third party such as Swype in order to mount such an attack.

various types of trust violations, a user can decrease the likelihood that a third party violates their trust at all.

4.1 Limiting Capabilities

Limiting the number of capabilities granted to third parties is an obvious way to reduce the risk of privacy harms due to trust violations. Furthermore, controlling which capabilities to entrust to a third party is largely within the control of individual end users, making this a relatively direct manner in which to reduce the risk of harm. In the most extreme case, users can simply elect to avoid using third party services, effectively granting third parties no data-related capabilities at all. For most users, however, such an approach is at best impractical and in some cases simply not possible. Therein lies the crux of third party capability reduction – simply reducing capabilities is not enough. Instead, users must reduce capabilities while also maintaining the ability to benefit from third party services in the manners to which they are accustomed. Thus the true aim of third party trust reduction is to identify "trust surpluses" – situations where third parties are being trusted with more capabilities than are strictly necessary to provide the benefits the user derives from the service. Finding and eliminating such surpluses allows users to reduce the degree by which they must trust third parties while also continuing to leverage third party services for desirable benefits.

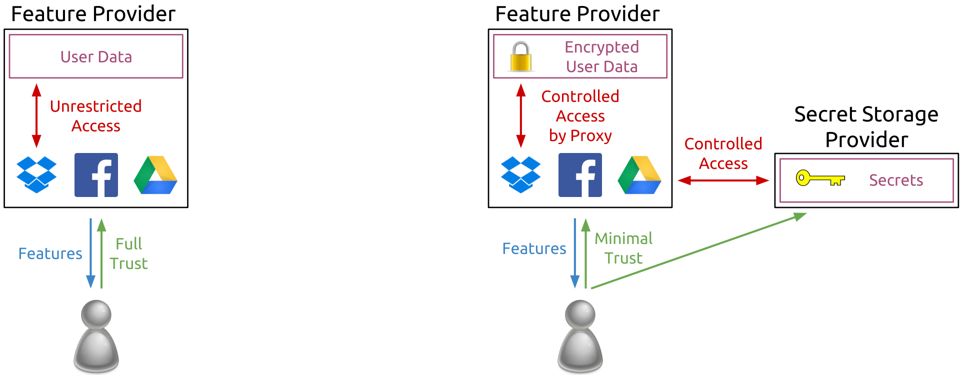
Fortunately,[10] trust surpluses appear to be relatively common in modern third party services. Take for example the Dropbox file syncing service. As discussed in §3.1, users must currently entrust Dropbox with all available capabilities: storage (S), access (R), modification (W), and metadata (M). In order to provide Dropbox's core service, however, Dropbox only requires a single capability: storage. Thus, granting Dropbox the access, modification, and metadata capabilities represents a trust surplus that can conceivably be eliminated without reducing Dropbox's ability to provide the syncing and sharing benefits users expect.[11]

But how does one best limit Dropbox's access to these surplus capabilities? As mentioned, client-side cryptographic techniques provide tools for limiting the access capability (via encryption) as well as the modification capability (via authentication). In the case of Dropbox, a client could encrypt and authenticate their data prior to uploading it to Dropbox and then decrypt and verify the data after retrieving it from Dropbox. Dropbox is unable to read or modify such encrypted and authenticated data when stored on their servers. Such techniques, however, have a downside. They require the user to mange and maintain certain secrets to which Dropbox is not privy – namely, the private keys necessary to perform data encryption or authentication. Furthermore, the user must find a way to manually distribute these keys across any device from which they wish to use Dropbox or to manually share them with any user with whom they wish to collaborate. These requirements impose an additional burden on the user, violating the original premise that users should be able to reduce third party trust without also reducing their ability to derive benefits from third party services. Such burdens significantly reduce the ease of use that draws most users to solutions such as Dropbox in the first place.

In order to overcome this deficiency, users require mechanisms that allow them to both leverage cryptographic techniques to limit Dropbox's capabilities while also avoiding additional usability burdens. For example, the user could turn to an additional third party service provider capable of

[10] Or unfortunately, depending on your perspective.

[11] The techniques discussed here focus primarily on limiting surplus access and manipulation capabilities. Unfortunately, limiting the metadata capability is historically much more difficult than limiting capabilities such as access or manipulation. Thus, until better solutions present themselves, it may be necessary to continue granting third party service providers the metadata capability – even in surplus.



(a) Traditional Trust Relationship

(b) Distributed Trust Relationship

Figure 1: SSaaS Model Relationships

automatically storing, syncing, and sharing secrets such as cryptographic keys on the users behalf. Such Secret Storage as a Service (SSaaS) models have been proposed in [77], [78], and [79]. When used in conjunction with a traditional cloud storage provider and existing cryptographic techniques, a secret storage service can be employed to transparently limit third party trust without imposing any significant additional burden on the end user. In such an arrangement, the end user stores only encrypted and authenticated file data with Dropbox, limiting Dropbox's access to the R and W capabilities. The user then stores the associated cryptographic secrets with a "secret storage provider" (SSP) capable of controlling access to the secrets in a user-defined manner and syncing or sharing them as requested. Neither Dropbox (called a "feature provider" (FP) in the SSaaS model) nor the SSP have the ability to access or manipulate user data since Dropbox lacks the keys necessary to perform such operations and the SSP lacks the data on which these operations are to be performed. Figure 1 illustrates such an arrangement. Using these techniques, the user can successfully eliminate two of the surplus capabilities traditionally granted to Dropbox in a manner that allows them to continue using Dropbox to sync and share files as they are accustom.

Techniques such as SSaaS are a form of "trust distribution" – a technique for reducing trust in individual third parties by instead spreading it across multiple parties. Similar techniques have been used within cryptographic protocols for the purpose of eliminating single-points-of-trust in the past [82] and have also been proposed as the basis for new cryptographic standards today [4, 64].[12] Trust distribution techniques are capable of allowing users to reduce or eliminate trust surpluses across a range of use cases without introducing significant additional usage burdens. While there are approaches to limiting third party trust that aim to avoid trusting any third party (e.g. the OTR chat protocol [69]), such techniques are often difficult to apply generally or to use without creating additional usability challenges. Trust distribution strategies, however, provide a relatively generic framework for eliminating trust in any single third party.[13]

To summarize, the proposed recipe for reducing the number of trusted capabilities afforded to third parties is as follows:

1. Identity any surplus capabilities

[12] Such techniques also bear some resemble to previously proposed "key escrow systems", albeit with a somewhat opposite end-goal [16]: escrow systems aim to allow additional third parties access to user data whereas trust distribution systems aim to reduce the access to user data any single party can achieve.

[13] When coupled with techniques such as [82], trust distribution techniques can eliminate trust in even larger subsets of third parties, e.g. not having to trust up to three of five total parties.

2. Leverage cryptographic techniques to limit third party access to these capabilities
3. Leverage trust distribution techniques to store and control access to the secrets required by the aforementioned cryptographic techniques in a manner that avoids burdening the end user with the need to manage such secrets manually.

This process eliminates trust surpluses by distributing user trust across multiple third parties so that individual third parties can not subvert this trust. As mentioned in §3.2, such arrangements do run the risk of encouraging collusion-type trust violations where multiple third parties work together to regain capabilities that have been denied to them individually. Nonetheless, such collusion violations are strictly less likely to occur than single-party violations since they require multiple parties to all be willing to commit an equivalent single-party violation in concert. The techniques discussed in the next section for disincentivizing single-party violations thus also help to disincentive collusion violations.

4.2 Disincentivizing Violations

Beyond limiting the number of capabilities users must entrust to third parties, it is also desirable to disincentivize the mechanisms by which third parties might violate such trust. While technological solutions provide options for reducing the degree of trust, it is largely policy solutions that will drive the disincentivization of common classes of trust violations. By disincentivizing certain classes of trust violations, we can reduce the likelihood that third parties will commit such violations, leading to more "trustworthy" third parties and fewer instances of trust violations. There are a variety of mechanisms that one might employ with an aim toward disincentivizing trust violations.

4.2.1 Distributed Trust Markets

In today's traditional third party trust relationships, users primarily select third party services on the basis of their features. When users pay for these services, they're primarily paying to support the core features such services provide. Privacy and security, while potential end user concerns, are at best secondary goals. Furthermore, on many free cloud services, the ability to harvest user data is the basis of the service provider's business model. As discussed in Section 4.1, these situations create a number of perverse incentives in terms of a third party's respect for user security and privacy. In the first case, the third party simply does not prioritize user security since that is not the basis on which users are choosing to use a service. In the second case, a third party actively works to subvert user security in order to further leverage user data for the generation of profit.

Distributed trust relationships (Figure 1), such as those employed by the aforementioned SSaaS model, aim to rectify these issues by introducing additional third party actors whose primary goal is the protection of user secrets and, by proxy, the data such secrets can be used to cryptographically protect. The ability of distributed trust architectures to separate privacy-oriented secret storage duties from feature-oriented service provider duties allows users to select each service on the basis of its associated merits. This quality avoids the issues associated with putting desirable features in direct competition with security and privacy – a competition that security and privacy have historically lost. Distributed trust relationships not only allow users to eliminate trust surpluses as discussed in §4.1, they also allow users to escape from traditional, but largely artificial, trade-offs between desirable third party features and the control of their data.

By severing desirable features from desirable security-enhancing properties, independent markets can form around feature provision and secret protection, optimized for the respective priorities of each field. A distributed trust ecosystem is thus able to make security and privacy tradable commodities, and to leverage market powers to price and improve both. A competitive distributed trust market has a number of security and privacy enhancing benefits:

Reputation: A user's ability to easily switch between secret storage providers forces providers to compete on the basis of their security and privacy preserving reputations.[14] Providers who do a superior job avoiding the trust violations discussed in §3.2 can attract more users and/or command a higher price for their services. Since a secret storage provider's reputation is tied solely to their ability to faithfully protect user secrets, they will not be able to "iron over" any privacy-related reputation failings with superior end user feature sets – a practice employed by many traditional feature providers.[15]

Multiple Providers: A healthy ecosystem of competing secret storage providers allows users to select from multiple independent providers over which they may further distribute their trust beyond a binary "feature provider + secret storage provider" relationship. Such secret "sharding" provides a number of benefits over relying on a single SSP, from additional trust reduction to data redundancy.

Cost: As in other competitive markets, having a number of competing providers will allow the user to select a provider that offers the best combination of cost and service.

Distributed trust markets are potentially useful for disincentivizing a range of trust violations, from implicit violations to unintentional violations. Such markets help align economic incentives with practices that do not favor such violations.

4.2.2 Digital Due Process

While mechanisms such as distributed trust markets are useful for disincentivizing the implicit, unintentional, and colluding classes of trust violations, other mechanisms are needed to disincentivize compelled violations. While, trust markets potentially encourage third parties to push back against compelled trust violations to the maximum extent permitted under the law, they do little to protect users in cases where the law requires such violations. While there are some cases where such violations are in the public interest, it appears that in many (if not most) compelled violation cases, the public interest is not well served [46, 47]. To reduce unnecessary compelled violations, it is important to ensure "digital due process" rights.

The first step toward protecting such rights is to ensure that user data stored or processed by third party services receives the same level of protection as data stored or processed locally. This concept runs counter to the Third Party Doctrine established by current U.S. case law [94]. This doctrine holds that individuals who voluntarily store their data with third parties have no "reasonable expectation of privacy" for such data [88]. While this viewpoint may have made sense in the mid-20th century when it was established by a series of Supreme Court rulings [89, 90], it does not translate well to a world where third party access to user data is the norm. As shown in §3.2, compelled violations are a growing trend, and in many cases such violations are served via third party doctrine mechanisms. Such trends suggest a likely overreach of government data collection, leading to a range of adverse "chilling" effects (e.g. [71]). One possible way of halting or reversing this trend would be to eliminate the third party doctrine and begin requiring probable cause warrants in order to compel third parties to provide or modify user data.

[14] In order to achieve such mobility, it is likely necessary to standardize a common distributed trust protocol [77].

[15] As an example, consider Facebook's numerous trust violations [37, 59, 97] and the fact that such violations have had no noticeable impact on the number of people using Facebook [32]. A secret storage provider would enjoy no such network benefit from providing additional services beyond secret storage were they to violate user trust; instead, users would simply switch to a new provider.

Recently, changes to the third party doctrine have begun to progress on multiple fronts. Recent Supreme Court decisions have suggested a willingness to expand user privacy rights in the digital realm, and may eventually lead the Supreme Court to revisit the third party doctrine directly [91]. Congress has also long debated updates to third party doctrine-derived laws such the Electronic Communications Privacy Act (ECPA) [103] to include a warrant requirement for digitally stored emails [14]. Recently, the U.S. House of Representatives unanimously passed a bill amending the ECPA to require warrants in most cases [96]. These movements suggest growing recognition of the due process rights of digital data, regardless of whether it is stored locally or by third parties. Such trends likely represent the best hope for reducing unnecessary compelled violations, ensuring such violations only occur in cases where the public interest is significantly favored by the compelled violation of user trust.

4.2.3 Third Party Liability

Another mechanism for disincentivizing trust violations, especially of the unintentional variety, would be to establish standards of liability for trust violations that result in harms to user privacy. If a third party violates a user's trust and harms the user in the process, it is reasonable to expect that users should be able to seek some measure of relief from such violations. Trust violation liability would follow the growing trend toward holding companies liable for digital data breaches resulting from poor security practices (e.g. [104]).

The nature of this liability could take several forms. The most obvious form would be to impose civil liability commensurate with the harm caused by a trust violation on the party committing the violation. This opens up the thorny issue of how to value such harms. Anecdotally, how much a user is harmed by a trust violation varies widely from case to case. For example, the harm from a trust violation resulting in the public exposure of a set of not particularly sensitive scenic landscape photos is likely to be far less than that caused by a violation that leaks trade secrets, medical data, or other sensitive material [1, 75].

One way to overcome this challenge would be to have users declare the value of the harm that would result from the misuse of trust capabilities when entrusting third parties with such capabilities. This approach is similar to the manner in which one might declare the value of a parcel when shipping it for the purpose of securing insurance. Third parties could even use such declarations to charge a user varying amounts for the services they provide – entrusting a third party with access to more "valuable" data would increase the cost of the provided service to the end user, while entrusting a third party with access to less "valuable" data would reduce the cost. The damages owed to the user in the event that the third party violates their trust could then be calculated relative to this value. In cases where a trust violation occurs due to an unforeseeable event or otherwise through no negligence on the part of the third party (e.g. a flaw in a respected external code library or similar unintentional trust violation), the user would be reimbursed at or below the declared value of the harm. In the case where trust is violated due to third party negligence, malpractice, or malfeasance, (e.g. an implicit or particularly egregious unintentional trust violation) the user would be reimbursed several multiples of the declared harm (e.g. similar to the statutory damage multiplier leveraged in some intellectual property cases where the infringement is found to be "willful").

In addition to allowing users to seek compensation for harm suffered due to third party trust violations, this approach also further incentivizes the use of distributed trust architectures. Since such architectures reduce the number of capabilities with which any single third party must be trusted, they also reduce the declared value of any associated harms. For example, the loss of user data (violation of the S capability) is in most cases a lessor harm than the public exposure of user data (violation of the R capability). By distributing trust across multiple parties, the user devalues the harm each party can inflict, allowing the user to declare lower harm costs and pay less for the third party services they use. A mechanism of trust-violation liability both incentivizes users to spread their trust across multiple parties and encourages third parties to avoid any trust violation that would require them to pay out the associated damages.

To manage this liability, third parties would likely be required to secure insurance to cover the cost of damages in the event that a trust violation occurs [15, 87].[16] These insurers would be in a position to provide additional economic disincentives to third party trust violations. For example, insurers could charge each third party on the basis of how "secure" (or the inverse, how "risky") a third party's infrastructure is. Third parties who employ additional security protections or who otherwise adhere to security best practices would end up paying lower insurance premiums to indemnify them against claims for trust violation damages.

Regardless of mechanism, establishing a standard system for trust violation liability will help disincentivize trust violations via a variety of mechanisms. Tying financial penalties to such undesirable behaviors encourages third parties to avoid trust violations, even when such parties act only in their own self interest. Using a declaratory harm valuation model avoids the challenges associated with properly accessing the harm caused by breaches of user privacy, and provides a straightforward mechanism for compensating users for breaches of third party trust.

5 Conclusion

The pervasiveness of third parties across the modern cloud computing landscape is undeniable. What this pervasiveness means for the privacy and security of users and their data is an area of active research. In this paper, I presented a bi-axial model for evaluating third party trust by both degree of trust and manner of violation. I then applied this model to a variety of popular third party services as well as examples of historic trust violations. This analysis is useful in helping to understand the manners in which user privacy relies on trusted third parties as well as the motivations that might undercut this trust. From this analysis, it is clear that user security and privacy has been, and continues to be, at risk from a wide range of third part trust violations. Addressing and minimizing this risk is essential in order to preserve security and privacy of user data in the digital age.

Toward this end, I provided a number of suggestions for reducing both the degree of third party trust (e.g. via the use of distributed trust architectures) as well as for disincentivizing common classes of trust violations (e.g. by holding third parties liable for such violations). By taking the multi-pronged approach of both reducing the degree of power granted to third parties while also creating disincentives to abusing this power, it is possible to significantly decrease the degree of risk users expose themselves to while utilizing third party services. While these techniques are unlikely to fully eliminate the privacy and security risks inherent to the use of trusted third parties, they do provide a basis on which such risks can begin to be measured and mitigated.

[16] It is even possible that the government itself might act as such an insurer (or insurance underwriter), as they currently do with banks via the Federal Deposit Insurance Corporation (FDIC) [100].

References

[1] Acquisti, A., John, L. K., and Loewenstein, G. What is privacy worth? *The Journal of Legal Studies 42*, 2 (June 2013), 249–274.

[2] Amazon.com, Inc. Amazon information request report. `https://aws.amazon.com/compliance/amazon-information-requests/`.

[3] Anthony, S. Tim Cook says Apple will fight US gov't over court-ordered iPhone backdoor. *Ars Technica* (February 2016).

[4] Apache Software Foundation. Milagro: Distributed Cryptosystem for Cloud Computing. `http://milagro.apache.org/`.

[5] Apple, Inc. iCloud. `https://www.apple.com/icloud`.

[6] Apple, Inc. Update to celebrity photo investigation. `https://www.marketwatch.com/story/apple-media-advisory-2014-09-02`, September 2014.

[7] Bonneau, J. A technical perspective on the Apple iPhone case. `https://www.eff.org/deeplinks/2016/02/technical-perspective-apple-iphone-case`, February 2016.

[8] Borisov, N., Goldberg, I., and Brewer, E. Off-the-Record communication, or, why not to use PGP. In *Workshop on Privacy in the Electronic Society* (2004).

[9] Brodkin, J. The secret to online safety: Lies, random characters, and a password manager. *Ars Technica* (June 2013).

[10] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and Thayer, R. RFC 1880: OpenPGP message format. Tech. rep., Internet Engineering Task Force, 2007.

[11] Camp, L. J. Designing for trust. In *Trust, Reputation, and Security*. Rino Falcone, Springer-Verlang, Berlin, 2003.

[12] Codenomicon. The Heartbleed bug. `http://heartbleed.com`.

[13] Cook, T., and Apple, Inc. A message to our customers. `https://www.apple.com/customer-letter/`, February 2016.

[14] Cope, S. Senate Judiciary Committee finally focuses on ECPA reform. `https://www.eff.org/deeplinks/2015/09/senate-judiciary-committee-finally-focuses-ecpa-reform`, September 2015.

[15] Council of Insurance Agents and Brokers. Cyber Market Watch Survey. Tech. rep., April 2015.

[16] Denning, D. E., and Branstad, D. K. A taxonomy for key escrow encryption systems. *Communications of the ACM 39*, 3 (March 1996), 34–40.

[17] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium* (2004).

[18] Donenfeld, J. A., and Others. pass: the standard unix password manager. `https://www.passwordstore.org/`.

[19] Dropbox, Inc. Dropbox. http://www.dropbox.com.

[20] Dropbox Inc. Dropbox transparency report. http://www.dropbox.com/transparency.

[21] Dropbox, Inc. Dropbox wasn't hacked. http://blog.dropbox.com/2014/10/dropbox-wasnt-hacked/.

[22] Dropbox, Inc. Yesterday's authentication bug. http://blog.dropbox.com/2011/06/yesterdays-authentication-bug/.

[23] Dworkin, M. Recommendation for block cipher modes of operation: The CMAC mode of authentication. Tech. Rep. 800-38B, National Institute of Standards & Technology, 2005.

[24] Dworkin, M. Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC. Tech. Rep. 800-38D, National Institute of Standards & Technology, 2007.

[25] Dworkin, M. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. Tech. Rep. 800-38C, National Institute of Standards & Technology, 2007.

[26] Evans, D. S. The online advertising industry: Economics, evolution, and privacy. *Journal of Economic Perspectives* (April 2009).

[27] Facebook, Inc. Facebook. http://www.facebook.com.

[28] Facebook, Inc. United States law enforcement requests for data. https://govtrequests.facebook.com/country/United%20States/#.

[29] Farivar, C. Feds break through seized iPhone, stnad down in legal battle with Apple. *Ars Technica* (March 2016).

[30] Federal Bureau of Investigation. FOIA: National security letters. http://vault.fbi.gov/National%20Security%20Letters%20(NSL), 2007.

[31] Flowerday, S., and Solms, R. V. Trust: An element of information security. In *Security and Privacy in Dynamic Environments*, S. Fischer-Hübner, K. Rannenberg, L. Yngström, and S. Lindskog, Eds., vol. 201 of *IFIP International Federation for Information Processing.* Kluwer Academic Publishers, Boston, 2006, pp. 87–98.

[32] Foster, B. How many users on Facebook. http://www.benphoster.com/facebook-user-growth-chart-2004-2010/, 2015.

[33] Frosch, T., Mainka, C., Bader, C., Bergsma, F., Schwenk, J., and Holz, T. How secure is TextSecure? *Cryptology ePrint Archive 2014/904*, February (2014), 17.

[34] Gallagher, S. EPIC fail: How OPM hackers tapped the mother lode of espionage data. *Ars Technica* (July 2015).

[35] Gallagher, S. Blame the victim: Report shows fifth of breaches caused by miscellaneous errors. *Ars Technica* (April 2016).

[36] Gellman, B., and Soltani, A. NSA infiltrates links to Yahoo, Google data centers worldwide, Snowden documents say. *The Washington Post* (October 2013).

[37] Goel, V. Facebook tinkers with users' emotions in news feed experiment, stirring outcry. `http://www.nytimes.com/2014/06/30/technology/facebook-tinkers-with-users-emotions-in-news-feed-experiment-stirring-outcry.html`, June 2014. New York Times.

[38] Google, Inc. Drive. `http://drive.google.com`.

[39] Google, Inc. End-to-end. `http://github.com/google/end-to-end`.

[40] Google, Inc. Gmail. `https://mail.google.com`.

[41] Google Inc. Google transparency report. `https://www.google.com/transparencyreport/userdatarequests/`.

[42] Google, Inc. Hangouts. `https://hangouts.google.com`.

[43] Google, Inc. Plus. `https://plus.google.com/`.

[44] Grandison, T., and Sloman, M. A Survey of Trust in Internet Applications. *IEEE Communications Surveys & Tutorials 3*, 4 (2000), 2–16.

[45] Green, M. What's the matter with PGP? *A Few Thoughts on Cryptographic Engineering* (2014).

[46] Greenwald, G. The crux of the NSA story in one phrase: 'collect it all'. *The Guardian* (July 2013).

[47] Greenwald, G., and MacAskill, E. NSA Prism program taps in to user data of Apple, Google, and others. *The Guardian* (June 2013).

[48] Hern, A. I read all the small print on the internet and it made me want to die. *The Guardian* (June 2015).

[49] Hill, K. How Target figured out a teen girl was pregnant before her father did. *Forbes* (February 2012).

[50] Koch, W. GnuPG. `http://www.gnupg.org`.

[51] Kravets, D. FBI paid "grey hats" for zero-day exploit that unlocked seized iPhone. *Ars Technica* (April 2016).

[52] Krebs, B. Safeguarding your passwords. *Krebs on Security* (2008).

[53] Krebs, B. Data breach at health insurer Anthem could impact millions. *Krebs on Security* (February 2015).

[54] Krebs, B. Premera Blue Cross breach exposes financial, medical records. *Krebs on Security* (Mar 2015).

[55] LastPass. LastPass password manager. `https://lastpass.com`.

[56] Lee, M. Chatting in secret while we're all being watched. *The Intercept* (July 2015).

[57] Levison, L. Lavabit. `http://lavabit.com`.

[58] Levison, L. Secrets, lies and Snowden's email: Why I was forced to shut down Lavabit. *The Guardian* (May 2014).

[59] Lomas, N. Facebook data privacy class action joined by 11,000 and counting. *TechCrunch* (August 2014).

[60] Marlinspike, M. The Cryptographic Doom Principle. *Thought Crime* (December 2011).

[61] Marlinspike, M. Advanced cryptographic ratcheting. Tech. rep., November 2013.

[62] McDonald, A. M., and Cranor, L. F. Americans' attitudes about internet behavioral advertising practices. *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society* (2010), 63–72.

[63] Microsoft. OneDrive. `http://onedrive.live.com`.

[64] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. Tech. rep., 2008.

[65] Nguyen, X.-T. Collateralizing privacy. Tech. rep., Indianapolis, 2004.

[66] Nuance. Swype: The keyboard that changed everything. `http://www.swype.com/`.

[67] Open Whisper Systems. Open Whisper: Privacy that fits in your pocket. `https://whispersystems.org`.

[68] OpenSSL Dev Team, and Others. OpenSSL. `http://www.openssl.org`.

[69] OTR Development Team. Off-the-record messaging protocol version 3. Tech. rep.

[70] Pagliery, J. Uber removes racy blog posts on prostitution, one-night stands. *CNN Money* (November 2014).

[71] Penney, J. Chilling effects: Online surveillance and wikipedia use. *Berkeley Technology Law Journal* (2016).

[72] Pew Research Center. Public perceptions of privacy and security in a post-Snowden era. `http://www.pewinternet.org/2014/11/12/public-privacy-perceptions`, November 2014.

[73] Ramey, C., and Others. GNU Bash. `http://www.gnu.org/software/bash/`.

[74] Reichl, D., and Others. Keepass password safe. `http://http://keepass.info/`.

[75] Romanosky, S., and Acquisti, A. Privacy costs and personal data protection: Economic and legal perspectives. *Berkeley Technology Law Journal 24*, 3 (December 2009).

[76] Sabater, J., and Sierra, C. Review on computational trust and reputation models. *Artificial Intelligence Review 24*, 1 (September 2005), 33–60.

[77] Sayler, A. *Securing Secrets and Managing Trust in Modern Computing Applications*. PhD thesis, University of Colorado Boulder, April 2016.

[78] Sayler, A., Andrews, T., Monaco, M., and Grunwald, D. Tutamen: A next-generation secret-storage platform. In *2016 ACM Symposium on Cloud Computing (SoCC 16)* (Santa Clara, CA, October 2016), Association of Computing Machinery.

[79] Sayler, A., and Grunwald, D. Custos: Increasing security with secret storage as a service. In *2014 Conference on Timely Results in Operating Systems (TRIOS 14)* (Broomfield, CO, October 2014), USENIX Association.

[80] Schneier, B. Password advice. *Schneier on Security* (2009).

[81] Schneier, B., and Others. Password safe: Simple & secure password management. `https://pwsafe.org/`.

[82] Shamir, A. How to share a secret. *Communications of the ACM 22*, 11 (November 1979), 612–613.

[83] Sims, P. Can we trust Uber? `http://medium.com/@petersimsie/can-we-trust-uber-c0e793deda36`, September 2014. Medium.

[84] Singer, N., and Merrill, J. B. When a company is put up for sale, in many cases, your personal data is, too. *New York Times* (June 2015).

[85] Solove, D. Going bankrupt with your personal data. *Teach Privacy: Privacy and Security Blog* (July 2015).

[86] SpiderOak. SpiderOak: Store, sync, share, privately. `http://spideroak.com`.

[87] Starks, T. Cyber insurance gets Hill attention. *Politico* (March 2016).

[88] Supreme Court of the United States. Katz v. United States. `https://www.law.cornell.edu/supremecourt/text/389/347`, 1967.

[89] Supreme Court of the United States. United States v. Mitchell Miller. `https://www.law.cornell.edu/supremecourt/text/425/435/`, 1976.

[90] Supreme Court of the United States. Smith v. Maryland. `https://www.law.cornell.edu/supremecourt/text/442/735`, June 1979.

[91] Supreme Court of the United States. United States v. Jones. `https://www.law.cornell.edu/supremecourt/text/10-1259`, 2012.

[92] Symantec Corporation. ShellShock: All you need to know about the Bash Bug vulnerability. `http://www.symantec.com/connect/blogs/shellshock-all-you-need-know-about-bash-bug-vulnerability`.

[93] Thompson, K. Reflections on trusting trust. *Communications of the ACM 27*, 8 (June 1984), 761–763.

[94] Thompson, II, R. M. The Fourth Amendment Third-Party Doctrine. Tech. Rep. R43586, Congressional Research Service, 2014.

[95] Tresorit. Tresorit: The ultimate was to stay safe in the cloud. `http://tresorit.com`.

[96] Trujillo, M. House unanimously passes email privacy bill. *The Hill* (April 2016).

[97] Tsukayama, H. Facebook draws fire from privacy advocates over ad changes. `http://www.washingtonpost.com/blogs/the-switch/wp/2014/06/12/privacy-experts-say-facebook-changes-open-up-unprecedented-data-collection/`, June 2014. Washington Post.

[98] Twitter, Inc. Transparency report / information requests. https://transparency.twitter.com/information-requests/.

[99] Uber. Uber: Your ride, on demand. http://www.uber.com.

[100] United States. Federal deposit insurance corporation. https://www.fdic.gov.

[101] United States. Foreign intelligence surveillance court. www.fisc.uscourts.gov/.

[102] United States. Fourth Amendment to the U.S. Constitution. http://www.gpo.gov/fdsys/pkg/CDOC-110hdoc50/pdf/CDOC-110hdoc50.pdf, September 1789.

[103] U.S. Department of Justice, Office of Justice Programs, Bureau of Justice Assistance. Electronic Communications Privacy Act of 1986 (ECPA), 18 U.S.C. Section 2510-22. https://it.ojp.gov/privacyliberty/authorities/statutes/1285.

[104] U.S. Federal Trade Commission. ASUS settles FTC charges that insecure home routers and cloud services put consumers' privacy at risk. https://www.ftc.gov/news-events/press-releases/2016/02/asus-settles-ftc-charges-insecure-home-routers-cloud-services-put, February 2016.

[105] U.S. Office of Personnel Management. Cybersecurity resource center: Cybersecurity incidents. https://www.opm.gov/cybersecurity/cybersecurity-incidents/.

[106] Verizon. 2016 data breach investigations report. Tech. rep., 2016.

[107] Wagner, K. How Facebook is using your photos in ads. *Mashable* (September 2013).

[108] WhatsApp, Inc. WhatsApp: Simple, personal, real time messaging. https://www.whatsapp.com/.

[109] Whitten, A., and Tygar, J. D. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium* (1999), pp. 679–702.

[110] Wilson, D., and Ateniese, G. To share or not to share in client-side encrypted clouds. *arXiv:1404.2697* (November 2014).

[111] Yahoo. End-to-end. https://github.com/yahoo/end-to-end.

[112] Zimmermann, P. *PGP Source Code and Internals.* MIT Press, 1995.

[113] Zimmermann, P. PGP marks 10th anniversary. https://www.philzimmermann.com/text/PGP_10thAnniversary.txt, June 2001.