A Cryptographic Airbag for Metadata: Protecting Business Records Against Unlimited Search and Seizure

Charles V. Wright
Portland State University

Mayank Varia
Boston University

Abstract

Governments around the world require that electronic service providers, including telecoms, ISP's, and even online services like Twitter and Facebook, must provide law enforcement agencies (LEA's) with broad access to so-called "business records" including communications *metadata*. Metadata is data about data; it does not include the contents of the users' communications, but it does typically show *who* each user communicated with, and at what times, and for how long. Metadata is actually surprisingly powerful, especially in a time when more and more messages are being encrypted from "end-to-end."

In this paper, we present a new approach for protecting communications metadata and other business records against unwarranted, bulk seizure. Our approach is designed from the start to be robust against this new class of political and legal attack. To achieve this, we borrow the recent notion of cryptographic *crumple zones* [31], i.e. encryption that can be broken, but only at a substantial monetary cost. We propose that a service provider who wishes to protect their users' privacy should encrypt each business record with its own unique, crumpled, symmetric key. Then, a law enforcement agency who compels disclosure of the records learns only ciphertext until they expend the necessary resources to recover keys for the records of interest. We show how this approach can be easily applied to protect metadata in the form of network flow records. We describe how a service provider might select the work factor of the crumpling algorithm to allow legitimate investigations while preventing the use of metadata for mass surveillance.

1 Introduction

Governments around the world require that electronic service providers, including telecoms, ISP's, and even online services like Twitter and Facebook, must provide law enforcement agencies with broad access to so-called "business records" including communications *metadata*. Metadata is data about data; it does not include the contents of the users' communications, but it does typically show *who* each user communicated with, and at what times, and for how long. Metadata is actually surprisingly powerful, especially in a time when more and more messages are being encrypted from "end-to-end." General Michael Hayden, former director of both the CIA and the NSA, famously stated, "We kill people based on metadata" [9].

Perhaps because metadata is perceived as less sensitive than message contents, many jurisdictions allow law enforcement access to metadata with relatively little oversight or requirement for due process. For example, the USA FREEDOM Act [1] mandates that telecommunications providers produce "call detail records" including the calling and receiving numbers and the duration of each call. Such a system makes metadata vulnerable to abuse or misuse by corrupt or overzealous law enforcement agents.

At the same time, according to Granick and Pfefferkorn [15], US law is currently unclear (or perhaps even undecided) on the question of whether LEAs can compel disclosure of cryptographic keys or encrypted data. In the mean time, small providers (e.g. Lavabit [26]) are especially vulnerable, since they lack the resources of larger corporations like Apple and Facebook to fight a lengthy court battle.

In this paper, we present a new approach for protecting communications metadata and other business records against unwarranted, bulk seizure. Our approach is designed from the start to be robust against this new class of political and legal attack. To achieve this, we adopt the recent notion of cryptographic *crumpling* [31], i.e. encryption that can be broken, but only at a substantial monetary cost. Unlike the original application of crumpling, which gives the authorities new abilities to recover messages that they could not otherwise access, here we

propose to use the same primitive in order to limit government access to records that they can currently obtain with little or no restraint.

We propose that a service provider who wishes to protect their customers' privacy should encrypt each business record with a crumpled symmetric key, which can be recovered through a moderately expensive brute force search. Even if the LEA can demand that the provider turn over all of their records, the LEA obtains only ciphertext until they then expend some non-trivial monetary resources to recover each plaintext record of interest. The strategy here is to encrypt the records with parameters carefully chosen so that (1) bulk, warrantless decryption is so expensive as to be infeasible, and (2) decrypting only the records needed for a legitimate investigation is less expensive for the LEA than fighting a court battle against the service provider to compel disclosure of the master key.

One limitation of this approach is that, applied naively, it may still require the LEA to decrypt and examine the plaintext of many records that are not relevant to any investigation. For example, suppose the authorities know that their suspect made a telephone call from a certain city on a certain day. Then, given a collection of call detail records from that city and date, and no other way to tell which records correspond to their suspect, the LEA must decrypt every record in the data set. This is sub-optimal for both the LEA and for the public. It wastes government funds and exposes the metadata of potentially thousands of innocent people to unwarranted scrutiny. We can eliminate most of the problem with the use of encrypted indexes [12, 14]. As the service provider's systems are collecting and encrypting the metadata records, they should also create an encrypted inverted index, that is, an encrypted data structure that maps each keyword in the plaintext business records to the list of encrypted records that contain the given keyword. Then, to recover the plaintext records corresponding to some keyword, for example an IP address or telephone number, the LEA must first query the encrypted index to obtain the list of matching encrypted records. Next, the LEA must expend resources to decrypt the list of records, and finally, they must expend even more resources to decrypt each encrypted record in the list.

2 Metadata Uses and Privacy Concerns

Communications metadata is so readily available to authorities because it is commonly collected and used by providers for legitimate purposes. These purposes include: system management and capacity planning; identifying users who misuse or abuse the service; and detecting intrusions.

2.1 Uses of Metadata

In this section we give a brief overview of some common kinds of metadata for various communication systems.

Telephony In telephone systems, *call detail records* (CDRs) store metadata about telephone calls and text messages [17]. CDRs include the start and end time (or the duration) for each call and the telephone numbers of the caller and the callee. This data is useful not only for the telecom provider's own internal network management and planning, but also for research in other areas [5], including urban planning and development, personal mobility, and security and privacy.

IP Networks The most common format for network-level metadata is Cisco's Netflow [8]. Netflow records describe "flows" of network-layer packets sharing the same source and destination IP addresses, the same transport-layer protocol, and the same transport-layer source and destination ports. Intuitively, each flow corresponds to one "half" or one "side" of a TCP connection. For each flow, Netflow records give the total number of network-layer packets in the flow, the total number of bytes, and the arrival time of the first and last observed packets in the flow.

The most popular versions of Netflow are version 5 and version 9 [8]. Version 5 defines a fixed record format for every flow and supports only IP version 4 networks. Version 9 [8] defines a much more flexible, template-based record format and supports many other network-layer protocols, including IP version 6 and MPLS. Netflow v9's template approach was intended to make it easy to add new features without breaking backward compatibility with older devices. In Section 4, we describe how this extensibility can be used to add crumpled encryption to Netflow logs.

SMTP Email The simple mail transport protocol [19] is used to transmit email messages from client to server and from server to server. SMTP servers typically log the source address ("`MAIL FROM`"), destination address ("`RCPT TO`"), the arrival time, and the total size of each email that they receive.

Exceptions Some privacy-focused providers, such as the Signal encrypted messaging app, deliberately avoid logging metadata to non-volatile storage, in order to maximize users' privacy against LEA requests [27]. In doing so, they lose the ability to track how their system is used over time. Many providers who lack the engineering expertise and financial backing that Signal enjoys may not be able to afford such a trade-off.

2.2 (Meta)data Privacy

One of the first major legal attacks on privacy and confidentiality of communications was the Clinton administration's attempt in the 1990's to mandate key escrow [2, 10] for all encryption through use of the Clipper chip [4, 23]. More recently, the encrypted email provider Lavabit fought and lost against demands from the US government for encrypted data [26]. (Lavabit's vulnerability to this attack was amplified by a design flaw that gave them the ability to decrypt users' files [21].) The most famous legal attack against a provider of secure communications was the 2016 attempt by the FBI to compel Apple to assist in decrypting an iPhone once used by a terrorist [18]. Lewis et al describe the varying legal requirements for access to encrypted data around the world [20]. Granick and Pfefferkorn discuss the current legal ambiguity around LEA access to encrypted data in the US [15].

Protecting metadata is even more difficult from both a legal and technological standpoint. Legally, many countries make it easier for governments to acquire metadata than the contents of communications. For instance, in the United States, the Electronic Communications Privacy Act (ECPA) explicitly requires a higher bar for acquiring communication content (a warrant) than for non-contents like metadata (a subpoena, and sometimes not even this). Technologically, research into protecting metadata lagged decades behind cryptographic primitives like encryption for the protection of data contents. Approaches for protecting metadata typically focus on obscuring the communicating endpoints from observation by service providers in the first place, by channeling communications pseudorandomly through a network of intermediaries. Various approaches include: Chaum's mixes [6], Tor [11], I2P [32], Loopix [25], Vuvuzela [29], and Stadium [28].

3 Addressing Government Overreach

In this work, we consider a different point of view in which the service provider is *not* the adversary. Instead, this work focuses on the design of low-cost tools that a service provider can deploy proactively to protect against overzealous government requests for information. We believe that this threat has ample precedent, including the legal disputes mentioned in the previous section. We comment that our work might also influence the discovery process in civil lawsuits, although this is not a focus of our work.

We rely upon a communication service provider's interest in protecting their customers' civil liberties against unreasonable search and seizure. (We acknowledge that the validity of this assumption may vary widely between different types of service providers; e.g., federated chat platforms versus established large telecommunication providers). Our technological objective is to give the LEA a workable approach for obtaining (only) the data that it needs while reducing the risk that the government can obtain large amounts of metadata.

3.1 Technology in the Context of Society

While this work focuses on the design of a technological measure, we stress upfront that it is intended to *complement* social and legal structures rather than to supplant them. Our goal is merely to equip small communications providers and privacy advocates with low-cost tools that they need to fight and win in courts of law, as well as in the court of public opinion. As a result, we assume in this work that the courts place some value on privacy and on limiting the powers of law enforcement; an authoritarian regime with weak courts could always demand the master decryption key.

Furthermore, we continue to rely upon established legal and societal safeguards against against targeted misuse or abuse of the system by LEAs who are willing to expend public resources for their own corrupt or illicit goals, against which our approach provides no technological protection. Existing legal incentives against targeted misuse include, for instance, the criminal penalties that employees of the United States federal government face for misusing government property (e.g., 18 U.S.C. § 741) or for using government property for personal benefit (e.g., 5 C.F.R. § 2635.704). Our work relies upon a technological tool called *crumpling* that requires the expenditure of government resources, thereby creating the opportunity to leverage existing legal incentives against misuse.

3.2 Technological Tool: Crumpling

The main tool we use to protect metadata is *cryptographic crumpling*, a notion recently proposed by Wright and Varia [31] as a more secure alternative to key escrow or other encryption "back doors." They propose two classes of cryptographic puzzles for constructing encryption schemes that are breakable only through immense expenditure of resources. First, an *abrasion* puzzle is one that costs many millions, perhaps billions, of dollars to solve, and thus serves as a gatekeeper to discourage any adversary except for a nation state from investing in a key recovery effort. Second, a *crumpling* puzzle is one that can be applied to derive each symmetric key, and that can be solved for a moderate amount of money, e.g. hundreds or thousands of dollars. Because crumpling puzzles only use existing symmetric cryptography standards,

crumpled encryption is incredibly fast and also benefits from existing hardware crypto acceleration (cf. §4.4).

We stress this work and [31] use the same fundamental crypto tool for very different ends. Wright and Varia [31] weakens the security of existing encryption systems in the hope of avoiding an even greater loss to political and legal attacks. By contrast, this work applies the crumpling to metadata, which is currently exposed in the clear in most cases, rather than to the contents of end-to-end encrypted messages. As a result, this work gives a monotone increase in privacy.

3.3 Two Motivating Scenarios

Here we describe one hypothetical example and one historical example to illustrate how our techniques could be used to protect users' privacy against unreasonable or overly broad demands for bulk metadata.

We envision our approach as being especially appealing for smaller communications providers who value their customers' privacy but also appreciate the rule of law and the need for law enforcement access in legitimate investigations. Eventually, we hope that it might be adopted by larger providers who share similar values.

Scenario 1: Counter-Terrorism Investigation Suppose a national law enforcement agency receives a tip that a terrorist cell might be starting to plot an attack in a certain city. The LEA can use its emergency powers to obtain the telephone records of everyone who lives in the target city.

If the records are turned over in plaintext, then the LEA can immediately perform some network analysis to help identify the conspirators based on their communications patterns. However, obtaining such a large number of call records also gives the LEA everything they need to perform a similar analysis targeting peaceful protest groups, civil rights activists, political organizers, and similar groups.

If the telephone records are protected using our approach, then the LEA must be more careful in their analysis. If the LEA knows one of the conspirators' telephone numbers, or the number of someone with whom they have recently corresponded, then they can still obtain the information that they need. First, they use the encrypted index (Section 4.1) to find the list of encrypted metadata records related to the first known suspect in the terror network. Then they decrypt the matching records to learn which other telephone numbers have communicated with the first suspect. The process continues until the LEA has identified enough suspects to proceed with the next phase of their investigation, for example with digital or physical surveillance, search warrants, arrests, etc.

In order to also map out the networks of protestors or activists, the LEA must expend substantial additional resources beyond what was required for the legitimate investigation. If we assume that terror cells are much less common than protestors and activists, this creates the possibility that the communications providers could carefully tune the decryption costs so that the LEA's budget allows it to perform all legitimate investigations and not *too much* else. The question of how exactly to set the costs to achieve such a delicate balance remains an open question and is outside the scope of this paper.

Scenario 2: Unusually Broad Warrants In 2017, a judge in North Carolina granted local police a very broad search warrant for information on all Google accounts for users within several acres of the site of a murder [13]. It is unclear how many people were affected, but depending on the population density in the given area, the number could be in the hundreds or thousands. Although such broad warrants are believed to be quite rare in practice, the North Carolina case shows that these things do happen in the real world.

If Google had used our techniques to protect the metadata records (if any) that they turned over in response to the warrant, then the incidental exposure of innocent people could have been reduced. Suppose there were 300 people with Google accounts in the 17 acres covered by the warrant. Then our approach makes it 100 times more expensive to examine all records versus examining only those records for the top 3 suspects. Although the local police could likely afford a very broad search in one or two cases, a department that routinely multiplies its data recovery costs by a factor of 100 would quickly expend its budget.

4 Proposed Constructions

In this section, we present a novel application of "crumpled" encryption to protect communications metadata and other business records against bulk search and seizure. We use network flow records as our running example of a metadata source that a small service provider might want to protect. We chose this data type because it is simple and well known to many readers with interests in free and open communication. Of course, the same techniques could also be used to protect packet header traces or communications logs produced by email servers or by other messaging services, e.g. internet relay chat (IRC) or the Matrix [30] federated chat platform.

Let K_M be the master secret key for a set of records. To protect against seizure of the master key, it might be held within a secure enclave or derived from a passphrase known only to a few trusted personnel. We do not use K_M

to encrypt any records—only to generate other keys.

In recent versions of Netflow, flow records are transmitted and stored in groups called *FlowSets* [8]. Each Netflow export packet consists of some header information including a timestamp, the source ID of the device that collected the flow records, and a sequence number unique to the Netflow export packet. The packet then contains *template flowsets* describing the format of the following flow records and *data flowsets* that contain the flow records describing each observed network flow.

For the i^{th} netflow record r of the j^{th} data flowset in a Netflow packet having sequence number *Seq* and source ID *sID*, we generate a unique, crumpled encryption key as follows. First, we generate a random salt s and a random nonce n. Unlike earlier applications of cryptographic crumpling, we do not have an "original" full-strength encryption key to crumple. So we simply generate one, using a pseudorandom function F with the master key and the salt s:

$$k_0 \leftarrow F_{K_M}(s)$$

Then we derive the crumpled key k_1 from our made-up "original" key k_0 and the Netflow packet's metadata about the flow record r. The crumpling algorithm applies two hash functions to the key and its associated data. The "little" hash function h projects the "original" key k_0 down into a much smaller space of size 2^ℓ, which can be brute-force searched via a moderately expensive computation on advanced hardware. (See Section 5.) The outer hash function H maps the reduced-strength key back into the proper number of bits for use as an encryption key, e.g. 128 or 256 bits. Although the resulting key is the same length as a normal key, we call ℓ its *effective* length, because it provides only ℓ bits of security.

$$k_1 \leftarrow H(h(k_0) \,||\, sID \,||\, Seq \,||\, i \,||\, j \,||\, n \,||\, \text{padding})$$

The padding is required so that the outer hash H will be computed over a block of data the of same size as the header that is hashed in the Bitcoin mining function [22]. This allows us to accurately predict the cost to recover the key through a brute-force search. (See Section 5.) For a more detailed security analysis of this construction, we refer the reader to the original work on crumpling [31]. Finally, we encrypt the original contents of the flow record, using k_1 as the encryption key.

$$r' \leftarrow Enc(k_1, r)$$

Finally, we save the encrypted flow record r', along with its plaintext nonce and salt. The most recent versions of the Netflow specification [8] support flexible template-based record formats. We propose to make use of this functionality to support breakable encryption of arbitrary records. To do so, we simply add to the existing record template new fields for our nonce and salt, and for any other cryptographic data required for the encryption itself, e.g. an initialization vector (IV) or a message authentication code (MAC).

Given these encrypted records, the data owner can use the master key K_M together with the salt to efficiently re-derive the key to decrypt any record. An LEA who obtains the encrypted record does not immediately learn anything about the encrypted flows except for their timestamps. But by brute-force searching all the possibilities for the secret $h(k_0)$, the LEA can recover any desired decryption key k_1 by expending the necessary resources.

4.1 Reducing Incidental Exposure

To reduce the number of irrelevant records exposed to law enforcement, and to reduce the unnecessary expenditure of resources to recover those records, we propose to use techniques developed for symmetric searchable encryption [12]. The idea is that, as the system receives, encrypts, and stores the individual records, it should also create (in memory) a search index over the records that it has seen. Then, at some regular interval, e.g. every hour or every day, the system should encrypt the index and save it to non-volatile storage.

The index is a data structure that maps from keywords or header values, e.g. phone numbers or IP addresses, to the lists of encrypted records that contain those values. In practice, the index can be implemented in a very simple structure, especially if it is saved to disk frequently enough that the number of records for the most common keyword is not too much longer than the list for less-frequent keywords. In its simplest form, the index can be saved in a key-value store such as Berkeley DB [24] or any number of NoSQL databases.

The index for each keyword w during indexing interval i is encoded as a key-value pair (K,V), where

$$K = H(w||i)$$
$$V = Enc(k_{w,i}, < r_1, r_2, \ldots, r_n >)$$

and $< r_1, r_2, \ldots, r_n >$ is the list of encrypted records created in interval i that contain keyword w. To further limit the reach of the LEA, the key $k_{w,i}$ should also be a crumpled key. For example, let

$$k_{w,i} \leftarrow H(h(F_{K_M}(w||i)) \,||\, w \,||\, i \,||\, \text{padding})$$

Then the LEA must also break that encryption in order to recover the list of matching encrypted records for w.

4.2 Indexing Structured Data

The term "keyword" implies free-form, unstructured data such as natural language text. In fact much of the search-

able encryption literature was written with unstructured data in mind.

However, business records tend to consist of structured data, where each record follows the same schema, and each attribute in the record has a known, fixed data type. For structured data, it makes sense to enable searches on particular attribures. For example, in a packet capture data set, the LEA might want to find all records where $SrcIP = 6.6.6.6$. We can support queries over structured data using the construction sketched above with the following simple extension.

For attribute a taking on value v, we store the list of encrypted records from interval i having $a = v$ as the pair (K,V) in the key-value store, where

$$K = H(a||v||i)$$
$$V = Enc(k_{a,v,i}, < r_1, r_2, \ldots, r_n >)$$

and, as above, the key $k_{a,v,i}$ is also a crumpled key.

4.3 Conjunctive and Disjunctive Queries

The encrypted index construction described above is admittedly quite crude compared to recent work in searchable encryption [12], but it suffices for our needs. It can also support (again in a crude way) more sophisticated kinds of queries. For example, to find all records containing both keywords w_1 and w_2, the LEA can query for each keyword individually, obtain the lists of matching records, and take the set intersection of the lists. Similarly, to find the records containing w_1 or w_2, they can take the set union of the two lists.

4.4 Overhead Incurred by the Provider

Our approach requires that the provider must expend additional resources to protect their users' privacy. In practice, they are unlikely to do so if the cost is too high. Fortunately, our constructions require relatively few cryptographic operations, all of which are symmetric rather than public key and can be efficiently accelerated in hardware.

Runtime Performance To encrypt a metadata record, we must first derive the key, then perform the actual encryption. To derive a crumpled encryption key, we require just four invocations of the SHA256 hash function. Because metadata records tend to be small — on the order of several dozen bytes — the encryption itself should be similarly inexpensive on modern commodity hardware. Since 2010, Intel processors have supported specialized instructions for AES encryption and decryption, offering very fast performance of between 1–4 clock cycles per byte [3]. Since 2013, they also support instructions for accelerating the SHA256 hash used to derive the key [16].

Storage Overhead The provider must also expend extra resources to store the additional data necessary for key derivation and decryption of each encrypted record. These data include the initialization vector (IV) and message authentication code (if any) required by the encryption scheme and the nonce and the salt that we use to derive the crumpled key.

For example, suppose a collection of Netflow records uses a template that includes all the same information as the old v5 Netflow. (See Table B-4 in [7].) Then each plaintext flow record is 48 bytes long. With AES in CBC mode, the initialization vector is 16 bytes. If our nonce and salt are each also 16 bytes, then the crumpled encrypted Netflow record consumes twice as much space as the original plaintext record.

Legal Risks A provider that uses our techniques also risks the possibility of a court battle with a law enforcement agency who is unhappy with limited access to crumpled metadata records and demands the master key. We believe this risk can be mitigated by choosing reasonable decryption costs; we give an example in the following section. However, the risk can never be eliminated entirely.

5 Estimating Real-World Costs for the LEA

In the previous work on crumpled encryption [31], we used the energy efficiency of commercial Bitcoin mining devices and the price of electricity in the United States to predict the real-world expected cost (in dollars) of recovering a crumpled key. We estimated that, using the most efficient commercially available mining hardware, an LEA would spend on average about one dollar to recover a crumpled key with an effective length of 50 bits. Then, for each 10 additional bits of effective key length, the LEA's expected cost increases by a factor of $2^{10} \approx 1000$. The projected costs are summarized in Table 1.

Effective Key Length (bits)	Expected Cost ($)
50	1
60	$2^{10} \approx \$1K$
70	$2^{20} \approx \$1M$
80	$2^{30} \approx \$1B$
90	$2^{40} \approx \$1T$

Table 1: Projected costs to recover crumpled keys [31]

Example: Call Records Suppose we impose a small but unavoidable new cost of $8 per "call detail record" from a telephone provider. Based on the estimates in Table 1, we can do this by encrypting each record with a key having an effective length of just 53 bits. Then, targeted monitoring of a criminal suspect who makes 100 calls per month would cost less than $10K per year—a small price relative to the salaries of the technicians, police officers, detectives, and prosecuting attorneys working the case. Assuming the same average rate of 100 calls per user per month, monitoring all 300 million Americans would cost more than $2.88 trillion per year—more than half the budget of the US federal government in 2016.

6 Conclusion

In this work, we described a new technique for protecting communications metadata against bulk surveillance. Our approach, based on the recent notion of breakable "crumpled" encryption, is designed to be robust not only against technical (e.g. cryptanalytic) attacks, but also against a new class of political and legal attacks that circumvent technical protections to obtain secrets via court order or legislative mandate. Our approach is not guaranteed to succeed in all cases. Nevertheless, we hope that it may enable privacy-oriented communications providers and privacy advocates to fight and win in courts of law, as well as in the court of public opinion. Finally, we stress that our technological approach is designed to leverage rather than sidestep existing legal and societal systems; nevertheless, improving accountability further remains an important topic for future work.

Acknowledgments

We gratefully acknowledge the anonymous reviewers and our shepherds, Bill Marczak and Riana Pfefferkorn, for their valuable advice. This material is based upon work supported by the National Science Foundation under Grant No. 1414119.

References

[1] 113th Congress (2013-2014). H.R.3361 - USA FREEDOM Act, 2014.

[2] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, P. G. Neumann, R. L. Rivest, J. I. Schiller, et al. The risks of key recovery, key escrow, and trusted third-party encryption. *World Wide Web Journal*, 2(3):241–257, 1997.

[3] K. Akdemir, M. Dixon, W. Feghali, P. Fay, V. Gopal, J. Guilford, E. Ozturk, G. Wolrich, and R. Zohar. Breakthrough aes performance with intel aes new instructions. *White paper, June*, page 11, 2010.

[4] M. Blaze. Protocol failure in the escrowed encryption standard. In *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pages 59–67. ACM, 1994.

[5] V. D. Blondel, A. Decuyper, and G. Krings. A survey of results on mobile phone datasets analysis. *EPJ Data Science*, 4(1):10, Aug 2015.

[6] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[7] Cisco. Netflow export datagram format, September 2007.

[8] B. Claise. Cisco Systems NetFlow Services Export Version 9. RFC 3954, RFC Editor, October 2004.

[9] D. Cole. We kill people based on metadata. *The New York Review of Books NYR Daily*, May 2014. `http://www.nybooks.com/daily/2014/05/10/we-kill-people-based-metadata/`.

[10] D. E. Denning and D. K. Branstad. A taxonomy for key escrow encryption systems. *Commun. ACM*, 39(3):34–40, 1996.

[11] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.

[12] B. Fuller, M. Varia, A. Yerukhimovich, E. Shen, A. Hamlin, V. Gadepally, R. Shay, J. D. Mitchell, and R. K. Cunningham. SoK: Cryptographically protected database search. In *IEEE S&P*, pages 172–191. IEEE Computer Society, 2017.

[13] S. Fussell. North Carolina Police Issued Sweeping Warrants to Search Data On All Google Devices Near Murder Scene. *Gizmodo*, March 2018.

[14] E.-J. Goh et al. Secure indexes. *IACR Cryptology ePrint Archive*, 2003:216, 2003.

[15] J. Granick and R. Pfefferkorn. When the cops come a-knocking: Handling technical assistance demands from law enforcement, 2016. `https://www.youtube.com/watch?v=PX2RjJAfTYg`.

[16] S. Gulley, V. Gopal, K. Yap, W. Feghali, J. Guilford, and G. Wolrich. Intel sha extensions–new instructions supporting the secure hash algorithm

on intel architecture processor. *Intel White Paper*, 2013.

[17] R. Horak. *Telecommunications and data communications handbook*. John Wiley & Sons, 2007.

[18] M. Isaac. Explaining Apple's Fight With the F.B.I. *The New York Times*, February 2016.

[19] J. Klensin. Simple Mail Transfer Protocol. RFC 5321, RFC Editor, October 2008.

[20] J. A. Lewis, D. E. Zheng, and W. A. Carter. The effect of encryption on lawful access to communications and data. Technical report, Center for Strategic and International Studies, February 2017.

[21] M. Marlinspike. Op-ed: Lavabit's primary security claim wasn't actually true, November 2013. `https://arstechnica.com/information-technology/2013/11/op-ed-a-critique-of-lavabit/`.

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

[23] National Institute of Standards and Technology. Federal Information Processing Standards Publication 185: Escrowed Encryption Standard. February 1994.

[24] M. A. Olson, K. Bostic, and M. I. Seltzer. Berkeley db. In *USENIX Annual Technical Conference, FREENIX Track*, pages 183–191, 1999.

[25] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis. The loopix anonymity system. In *26th USENIX Security Symposium, USENIX Security*, pages 16–18, 2017.

[26] K. Poulsen. Edward Snowden's E-Mail Provider Defied FBI Demands to Turn Over Crypto Keys, Documents Show, October 2013. `https://www.wired.com/2013/10/lavabit_unsealed/`.

[27] Signal Developers. Grand jury subpoena for Signal user data, Eastern District of Virginia, October 2016.

[28] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich. Stadium: A distributed metadata-private messaging system. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 423–440. ACM, 2017.

[29] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich. Vuvuzela: scalable private messaging resistant to traffic analysis. In *SOSP*, pages 137–152, 2015.

[30] N. Willis. Matrix: a new specification for federated realtime chat, February 2011.

[31] C. V. Wright and M. Varia. Crypto crumple zones: Enabling limited access without mass surveillance. In *Proceedings of the 3rd IEEE European Symposium on Security and Privacy*, April 2018.

[32] B. Zantout and R. Haraty. I2P data communication system. In *Proceedings of ICN*, pages 401–409. Citeseer, 2011.