Crypto Crumple Zones: Enabling Limited Access Without Mass Surveillance

Charles V. Wright
Portland State University, cvwright@cs.pdx.edu

Mayank Varia
Boston University, varia@bu.edu

Abstract—**Governments around the world are demanding more access to encrypted data, but it has been difficult to build a system that allows the authorities *some* access without providing *unlimited* access in practice. In this paper, we present new techniques for maximizing user privacy in jurisdictions that require support for so-called "exceptional access" to encrypted data. In contrast to previous work on this topic (e.g., key escrow), our approach places most of the responsibility for achieving exceptional access on the government, rather than on the users or developers of cryptographic tools. As a result, our constructions are very simple and lightweight, and they can be easily retrofitted onto existing applications and protocols. Critically, we introduce no new third parties, and we add no new messages beyond a single new Diffie-Hellman key exchange in protocols that already use Diffie-Hellman.**

We present two constructions that make it possible—although arbitrarily expensive—for a government to recover the plaintext for targeted messages. First, our symmetric *crumpling* technique uses a hash-based proof of work to impose a linear cost on the adversary for each message she wishes to recover. Second, our public key *abrasion* method uses a novel application of Diffie-Hellman over modular arithmetic groups to create an extremely expensive puzzle that the adversary must solve before she can recover even a single message. Our initial analysis shows that we can impose an upfront cost in the range of $100M to several billion dollars and a linear cost between $1K-$1M per message. We show how our constructions can easily be adapted to common tools including PGP, Signal, SRTP, full-disk encryption, and file-based encryption.

1. Introduction

Since 2013, the security and privacy community has worked with renewed focus to protect Internet communications against warrantless government surveillance. In this paper, we focus on two resulting technologies that have been deployed to billions of smartphone and laptop users in a frictionless, user-friendly manner: end-to-end encrypted messaging and fine-grained encryption of data at rest. A new generation of user-friendly smartphone and web apps offer end-to-end encrypted messaging [101], including Apple iMessage [33], [48], Telegram, the Signal application along with other messaging systems like WhatsApp and Wire that implement the Signal protocol [27], [76], and a new browser-based standard for voice and video chat in WebRTC [81]. Additionally, encryption is now the norm for data at rest, in the form of full-disk encryption on all major PC operating systems and file-based encryption on the two major smartphone platforms.

The widespread proliferation of strong, seamless encryption in transit and at rest has provided billions of people with increased personal privacy and civil liberty. However, the very popularity of the encrypted systems has generated a political conflict with powerful national governments that demand some level of access to citizens' communications and files (which we refer to collectively as "messages" from now onward) [95]. These demands are usually made in the name of public safety, national security, and counter-terrorism.

Tension. In this work, we examine the tension between two essential components of modern societies: the human right to privacy and the rule of law to provide safety. With regard to encryption, we posit the following:

1) There exist a few messages whose disclosure would aid public safety and law enforcement.
2) The vast majority of messages are either irrelevant to public safety or offer insufficient value to outweigh the individual and public good from privacy.

In the physical world, justice systems around the world have spent centuries specifying and interpreting laws like the 4th Amendment to the US Constitution in an effort to find an appropriate balance between these competing objectives. In the digital world, the law enforcement and technology communities largely agree on the aforementioned two premises, but their different prioritization between the two has led to absolutist positions whereby law enforcement authorities' decryption abilities should be either technologically cheap (i.e., restricted only by judicial oversight) or effectively impossible. We seek a middle ground between these two extremes.

Governments around the world have bifurcated along these absolutist positions. According to a recent report [63], more than half of the world's population lives in countries where strong end-to-end encryption is already illegal without some sort of backdoor or assistance for the authorities. For example, Blackberry was almost pushed out of India in 2010 until they agreed to provide governmental access [9] to customers' communications, and WhatsApp has been repeatedly banned in Brazil after failing to deliver decrypted messages under a court order. Even in Western democracies with strong traditions of civil liberties, support for strong encryption is falling among citizens and their elected representatives. In France and the UK, recent laws passed in response to terror attacks have greatly expanded the surveillance powers of those governments [68]. The United States imposes fewer restrictions on encryption than most countries, but even in the US there is growing pressure—including from the leaders of both major political parties and the former and current directors of the FBI—to somehow expand access to encrypted data. In many cases, the law enforcement "side" of this debate enjoys broad public support near or above 50% [79], [87].

With existing systems, providing a government with the means to access *some* encrypted communications usually requires giving them the technological capability to access *all* encrypted communications. We seek to avoid this outcome.

1.1. Our Contributions

In this paper, we present formal definitions and proof-of-concept constructions of end-to-end cryptographic systems with a new wrinkle: they permit law enforcement to access a small number of targeted, warranted messages. We stress upfront that our constructions actually require the use of strong, modern cryptography as a precondition. Modern systems offer fine-grained key management (e.g., per-message encryption keys with forward and backward secrecy [27]), which we leverage in order to build a fine-tunable decryption capability that only lets law enforcement open a limited number of messages. We describe below the capabilities of our system, and then in §2 we discuss (some of) the ethical considerations that influence whether to adopt such a system.

Requirements. In this work we propose the design of (authenticated) encryption schemes that permit law enforcement to recover the plaintext for selected encrypted messages of high value. Notwithstanding this giant change from the ordinary design of cryptosystems, we do require the following six principles:

1) Technologically prevent large-scale surveillance. That is: bound the number of messages that can be decrypted, independent of any judicial oversight.
2) Minimize any increase in system complexity to keep the attack surface and performance overhead small.
3) Impose no new burdens on encryption providers. They need not interact with law enforcement or observe any secrets that give access to messages.
4) Maintain integrity and authenticity of messages. That is: allow only passive breaches of confidentiality, not active man-in-the-middle attacks.
5) Maintain cryptographic best practices including primitives like authenticated encryption, properties like forward secrecy, and devices like HSMs.
6) Avoid storing a dense collection of users' secrets in one place. Ensure composability with security mechanisms that reduce the impact of law enforcement breaches and permit post-compromise recovery.

Previous work on this topic, dating mostly from the 1990s, focused on *key escrow*, wherein an encrypted copy of each decryption key is provided to the authorities or to some trusted intermediary. Key escrow fails all of our requirements: it imposes unacceptable overheads on its users while also relying exclusively on human oversight mechanisms to protect against government abuse. For these reasons and more, the US government ultimately withdrew their standard pertaining to key escrow [75]. We summarize other key escrow efforts in §1.2 below and refer readers to Abelson et al. [1], [2] and Pfefferkorn [77] for more detailed discussion of key escrow's fatal flaws; their work motivates several of the requirements listed above.

In this paper, we present the first constructions that can satisfy all six of these seemingly-contradictory requirements at the same time. Rather than having users or developers transmit key material (or anything else), our approach requires law enforcement to expend its own effort.

Our Approach: Moderately hard puzzles. We propose two cryptographic puzzles with easily tunable and carefully chosen security parameters that together make it possible—yet very expensive—for *any* well-resourced attacker to recover an encrypted message. We embed these puzzles into the generation of each per-message ephemeral key.

First, the *crumpling puzzle* is chosen independently for each ephemeral key, so that the government must expend effort to solve each one. The idea is that, like a crumple zone in automotive engineering, in an emergency situation the construction should break a little bit in order to protect the integrity of the system as a whole and the safety of its human users. We design a portion of our puzzles to match Bitcoin's proof of work computation so that we can predict their real-world marginal cost with reasonable confidence.

Second, the *abrasion puzzle* requires substantially more effort to solve. It serves as a "gatekeeper" that distinguishes well-resourced law enforcement agencies from non-state actors like cyber criminals and data-monetizing companies: it enables the former to access targeted message contents while preventing the latter from realizing sufficient benefit to make targeted access worth the cost. The abrasion puzzle uses public key techniques so that even the software developers who design it need not know the answer, and thus they cannot be forced to reveal it. This puzzle changes infrequently (think: every 1-2 years), and it must be solved in order to expose all crumpling puzzles within this time epoch.

We stress that our approach does not boil down to using weak cryptosystems with small security parameters. On the contrary: our methodology actually requires strong cryptography precisely in order to build a system that is delicately tuned to be "on the edge of breakable" by well-funded attackers but that still appears to have a large security parameter for everyone else (cf. §2.2). To the extent that the abilities of non-state entities may be difficult to estimate [65] or may grow over time, we recommend using estimates that err on the side of requiring the government to expend more resources.

Technical contributions. In this paper, we introduce two puzzles that collectively create a dichotomy in the distribution of message keys: they have small entropy for well-funded attackers but high (pseudo-)entropy for everyone else. In §3 we define new notions of secrecy that capture this dichotomy. Subsequently, we provide proof-of-concept constructions. In §4 we show how to *crumple* ephemeral message keys by choosing them from a small (tunable) subset S of the overall key space. This transformation allows attackers to recover contents for a moderate (tunable) per-message cost by brute-force searching S. In §5 we design a second puzzle that hides S in a manner that can be *abraded*, or worn down, at high cost. Unless the attacker pays the initial cost to recover the location of the subset S of the key space, each ephemeral message key retains its full cryptographic strength.

In §6 we analyze the costs of these puzzles to show that law enforcement can be computationally prevented from decrypting too many messages (requirement 1). We stress though that human oversight is still necessary to adjudicate *which* messages should be decrypted (cf. §2.3).

In §7 we show that our puzzle constructions are simple to retrofit onto existing encrypted messaging and file system protocols (requirement 2). The constructions impose very little overhead: our symmetric construction adds just four new invocations of a cryptographic hash function for each message, and our asymmetric construction adds one new Diffie-Hellman key exchange to protocols that already rely on Diffie-Hellman to derive a secret key. Critically, we do not introduce any new third parties to the system, nor do we require the transmission of any new messages beyond the new Diffie-Hellman exchange (requirement 3).

Our puzzles only apply to the encryption key component of two-key authenticated encryption schemes, so that authorities can selectively read messages but not forge them (requirement 4). Our puzzles apply independently to each ephemeral message key; as such, they are compatible with and indeed leverage the forward and backward secrecy properties of existing cryptosystems (requirement 5). Finally, each developer can change its system's public parameters, and thus force the government to solve a new initial gatekeeper puzzle, if the result of the old one is ever leaked (requirement 6); this key agility-style response [74] incentivizes governments to strongly protect the results of initial puzzles.

Explicit non-goals. In the design of our key crumpling constructions, we explicitly reject certain other goals commonly sought in related approaches. We do not guarantee "timely access" to encrypted messages, but neither do we seek to prevent a government from achieving it. If a government needs quick access to the plaintext of encrypted messages, they must design a parallelized architecture that can solve the relevant puzzle(s) in the time available (cf. §6.2).

Also, as a corollary of requirement 3, we also make no attempt in this work to provide any government with so-called *NOBUS*, or "nobody but us," exclusive access to decryption capabilities. As a result, we sidestep the question of deciding *which* governments should be afforded NOBUS access. Instead, we make the plaintext messages equally available to any attacker who is able and willing to expend the resources necessary to recover them, without any need for prior coordination with the app developer. Using the concept from *rational cryptography* [47] that an attacker won't pursue an action if its cost exceeds its expected benefit, we argue that this economic approach discourages non-state actors from acquiring the ability to decrypt targeted messages.

For scenarios in which it is important to distinguish between nation-states that can recover messages, we note that our scheme can be composed with NOBUS systems. Because our puzzles distinguish between acceptable and unacceptable attackers in an independent way from NOBUS systems, this work provides an additional safety measure that further restricts the attacker's ability.

Non-technical contributions. We hope that this work makes a number of non-technical contributions to the problem of (countering) electronic surveillance.

First, our constructions aim to restore a level of proportionality. When surveillance is cheap, law enforcement agencies have little incentive to refrain from collecting information on as many targets as possible, and they risk extreme consequences for missing any major terrorist attack. When electronic surveillance is expensive, as in the physical world, law enforcement must choose their targets wisely to focus on only the most pressing threats.

Second, we aim to give the security and privacy community a new bargaining chip in the public policy debate around encryption. We *do not* advocate for deploying our constructions in jurisdictions that currently allow for strong end-to-end cryptography. But, if the proponents of exceptional access are willing to make substantial concessions that improve civil liberties in other areas, like ending warrantless metadata collection, then perhaps a very conservative configuration of our constructions might provide an acceptable trade-off.

Finally, we hope to raise the level of the public debate on this very important topic. Until now, experts in cryptography and security have been locked in a stalemate with law enforcement leaders in a debate where the two options are framed as "no access to any encrypted data" versus "full access to everything." We hope to reframe the public discussion of this issue around more productive questions, such as, *if truly limited access is possible, is it actually desirable?* and if so, then *how much access should law enforcement be afforded in a free society?*

1.2. Related Work

We summarize related work along two dimensions: those that enable exceptional access and those that (independently) construct moderately hard cryptography.

Targeted access. Most previous work on enabling exceptional access has focused on key escrow; Abelson et al. summarize and critique this approach [1]. (Many of the same authors outline the risks of exceptional access again in a more recent paper [2].) Of all the previous work on key escrow, the one most similar to ours is Shamir's *Partial Key Escrow* [89]. There, instead of escrowing the entire symmetric key, the user escrows only a small portion of the key, leaving the authorities to brute force the remaining bits for access to the message. Our symmetric crumpling approach achieves a similar outcome, but we remove the requirement to escrow any bits of the key. One problem with partial key escrow is that a malicious user could submit incorrect information in escrow; this motivated a line of work on *Verifiable Partial Key Escrow* (e.g., [11]). In our construction the user has no opportunity to submit bogus information in the first place.

More recent methods of providing targeted access have focused on the NOBUS scenario. Tait [94] proposes to apply key escrow to every ephemeral key within a secure messaging system; our system is similar to Tait's, except with "key wrapping under the FBI's key" replaced with an initial puzzle open to everyone. Additionally, many works have proposed to federate an escrow key over multiple legal jurisdictions using secret sharing (e.g., [25]) or multiple entities within a single nation-state (e.g., [54]) to further limit use and increase faith in the system; we refer readers to Denning and Branstad's taxonomy for more details [41].

Moderately-hard cryptography. Our symmetric crumpling approach resembles a Merkle puzzle [70] and later proof of work schemes such as HashCash [8] and a similar scheme from Dwork and Naor [42]. More recently, proof of work has been incorporated into cryptocurrency constructions such as Bitcoin [73]. We borrow certain parameters of Bitcoin's proof of work construction so that we can more easily and accurately estimate our adversaries' performance. Other work on *Time-Lock Puzzles* [66], [82] has aimed at bounding the time required to solve a puzzle. We focus instead on a simple monetary cost model based on the amount of electricity consumed by real-world hardware.

Our symmetric crumpling technique was also inspired by work from Lenstra and Verheul on estimating the security level afforded by various symmetric key lengths [60], [62].

1.3. Organization

The organization of this paper reflects the authors' wish to treat a topic that has political and societal underpinnings in a scientific manner. In §2, we discuss the ethical implications of and societal questions surrounding our work. Subsequent sections are purely scientific in nature.

We provide security definitions against concrete cost-based attackers in §3. We present our constructions for symmetric and asymmetric crumpling in §4 and §5, respectively. In §6 we give initial engineering estimates for the real-world costs that a law enforcement agency would face in recovering crumpled keys. In §7 we show how our constructions can be easily retrofitted onto popular protocols for encrypted communications (PGP, Signal, and SRTP) and data-at-rest protection (full-disk encryption and file-based encryption). We conclude in §8 with some directions for future work.

2. Ethical Discussion

Before delving into the scientific details of our proposed key crumpling and abrasion constructions, we first discuss some of the ethical considerations surrounding their design and use.

Debates about the societal balance between privacy and safety (and indeed, debates about the extent to which one even must choose between the two) have occurred for centuries in many contexts. In this section, we provide a snapshot of that debate in the context of encryption. It is not our intention to persuade the reader to embrace and adopt our ideas. Instead, we wish to inform the societal discussion around this issue with a variety of sensible considerations surrounding both the current lack of, and potential future presence of, targeted decryption.

We partition this section into several questions pertaining to targeted decryption of high-value messages. First, should it ever be permitted? Second, are its benefits greater than its costs? Third, who can be trusted sufficiently by society to possess such power? Finally, what alternatives exist for law enforcement (in a world without targeted decryption abilities) or technologists (in a world with targeted decryption), and do they lead to better or worse outcomes?

2.1. Should We Ever Permit Targeted Decryption?

This initial question touches upon the technological values underpinning the design of encryption and its value to society. Rogaway [85] argues that "cryptography rearranges power" and as such it "can be developed in directions that tend to benefit the weak or the powerful." He argues that technological innovations should be deployed to benefit the weak. Furthermore, he worries about the risk of letting powerful actors co-opt technologies to "provide risk-free political cover" simply by "mumbl[ing] words that sound privacy-friendly." As a result, Rogaway might argue that the risk is too high to deploy technologies like targeted decryption that benefit the powerful.

Conversely, US FBI Director Wray [105] argues that the cost of *not* developing and deploying such technology is too high. Observing that "there's a technology and digital component to almost every case we have now," he remarks that the FBI's inability to retrieve data from "more than half of all the devices we attempted to access" over the past year "is a major public safety issue."

In between these viewpoints, Smith and Green [91] observe that "it is highly unlikely that either extreme – total surveillance or total privacy – is good for our society." Instead, they seek to "[find] the right balance [via] an ethical debate centred around the potential for collateral damage," or in other words whether the technology can disrupt bad actors without causing harm to ordinary civilians. They observe that changes to default encryption can disrupt some criminals but are unlikely to stymie sufficiently motivated or tech-savvy criminals. The next two ethical questions examine whether these benefits are worthwhile and whether a governmental entity can be trusted not to harm ordinary citizens with a targeted decryption power.

2.2. Are the Benefits Greater Than the Costs?

We consider four types of cost-benefit analyses: whether our internet-connected society creates more problems for law enforcement ("going dark") than benefits ("going bright"), whether the benefits of targeted decryption outweigh the costs, whether the benefits are so strong as to entice non-state actors to use targeted decryption, and finally whether we can accurately predict the costs.

Going dark vs. going bright. The "Going Dark" problem, as defined by US Deputy Attorney General Rosenstein [96], is "the threat to public safety that occurs when service providers, device manufacturers, and application developers deprive law enforcement and national security investigators of crucial investigative tools" such as targeted decryption capabilities.

We stipulate that the latter part of Rosenstein's quote is commonly accepted: that encrypted digital communications have at the very least *changed* the process of law enforcement investigations, independent of one's view on whether this is a good or even intended consequence of encryption. Former United Kingdom prime minister David Cameron remarked that encryption can "allow a means of communication between people, which even in extremis, with a signed warrant from the home secretary personally, that [the UK government] cannot read" [68] even if the communication is still readable by its intended participants.

As for whether this change creates a "threat to public safety," Bellovin et al. [15] observe that the digital revolution goes both ways. While "accessing communications content through traditional means could be getting harder," law enforcement's increased access to metadata combined with CALEA's low-cost phone wiretapping capabilities yield a telephony system that is too easy for law enforcement to intercept and that is inherently too vulnerable for users. Ultimately, Bellovin et al. call for "targeted interception approaches" to restore the balance.

Cost-benefit analysis of targeted decryption. We next examine whether targeted decryption specifically offers a net benefit to society, and thus whether it should be added as another option for law enforcement. Sunstein [93] observes that "for many problems in the area of national security, it is difficult to specify either costs or benefits"; this is a common sentiment shared by Abelson et al. [2] and others.

Sunstein also makes a more subtle argument against cost-benefit analysis: "many of the variables at stake can [be] difficult or perhaps even impossible to monetize." For instance: how can one quantitatively balance the added probability of finding a criminal versus the cumulative societal impact from (accidental or malicious) invasions of privacy? Sunstein remarks that "in situations of uncertainty, when existing knowledge does not permit regulators to assign probabilities to outcomes, it is standard to follow the maximin principle: choose the policy with the best worst-case outcome," or in other words the outcome that reduces the harms of "the very worst of the worst-case scenarios."

Even this analysis is complicated by the fact that people may disagree on the worst-case scenarios. For instance, Sunstein notes that a laudable law enforcement agent might anyway take "aggressive steps [that are] objectionable from the standpoint of civil liberties" in order to prevent even further deterioriation of civil liberties after a future terrorist attack. On the other hand, Rogaway [85] offers a "radically different" worst-case outcome to be avoided; his argument begins with the observation that "surveillance is an instrument of power" and deduces that "cryptography offers at least some hope" against a dystopian future wherein "surveillance will tend to produce uniform, compliant, and shallow people."

Government surveillance has been shown to influence and censor social behavior. For example, China's Social Credit System combines a "fear of reprisals. . . for those who don't put their hand up" to influence people to start "reporting on friends and even family members, raising suspicion and lowering social trust in China" along with "rewards. . . for those citizens who prove themselves to be trustworthy" to create a reputation system that "can even affect [one's] odds of getting a date, or a marriage partner" [20]. This and other examples illustrate that the effects of a loss in privacy can be difficult to quantify or to quarantine to an individual; as a result, Tufecki argues that privacy should be viewed as a public good "like air quality or safe drinking water" [98].

Benefits and costs for non-state actors. Recall from §1 that our cryptographic puzzles must be tuned to be infeasible or insufficiently worthwhile for non-governmental actors to solve. Here, we conduct a cost-benefit analysis under the assumption of rationality, wherein attackers only expend a cost if the message recovered has sufficient expected value.

The crumpling puzzle is strong enough to withstand some, but probably not all, attackers in this scenario. Companies engaged in data mining for advertising purposes are likely to be deterred with very small costs: Google and Facebook's global advertising revenue averages to less than $100 per year per user [50], with about a 2-4× preference for the most lucrative North American market [78]. Ergo, we find that costs in the hundreds of dollars would be sufficient to withstand these attackers. Online ransoms and blackmail typically extract on the order of thousands to hundreds of thousands of dollars per victim [72], and corporate espionage can be even more lucrative. Imposing a marginal cost high enough to thwart these attacks is likely also to dissuade law enforcement from using crumpling in the first place. It is our hope that the "gatekeeper" abrasion puzzle provides a new degree of freedom that restores the appropriate balance.

The cost of erring on costs. Our puzzles tune cryptosystems "on the edge" of being breakable. This approach is counter to the trend of cryptography over the past several decades, in which security margins have been increasing so that the resulting algorithms can withstand moderate future advances in cryptanalysis. As Bernstein [16] put it: "2^{80} security is interesting. 2^{128} security is boring."

As such, our work requires more precise cryptanalysis than we currently have today; vetting assumptions about puzzles (like Assumption 5.1 in this paper) requires *lower* bounds on the hardness of problems in addition to upper bounds. The history of DES provides an example of the uncertainty of concrete costs for cryptanalytic attacks; in the late 1990s it was known that DES could be brute forced, but the EFF demonstrated that the cost of doing so was at least an order of magnitude less than the prevailing government view at the time [65]. Furthermore, the security margins must be set to handle advances in computing capability due to architectural improvements and Moore's law; we discuss this issue further in §6.4.

2.3. Who Can Be Trusted With Such Power?

We consider three questions pertaining to trust: whether to grant decryption ability to a nation-state, whether our primitive can be scoped further to restrict this power, and whether government can be incentivized to protect the precomputation secrets that underlie this capability.

Trust in local law enforcement. Determining the trustworthiness of a government is as complex within any country as it is varied between countries. The World Economic Forum [104] states that "less than half (49%) the world's population lives in a democracy of some sort."

We leave it to democratic societies to decide for themselves whether they wish to entrust targeted decryption technology to their government. Landau observes that privacy and democracy are linked: "Protecting the privacy of speech is crucial for preserving our democracy" [57]. We also emphasize that our technological rate-limit (requirement 1) does not curtail the risk of sustained, targeted abuse of power.

Furthermore, we recognize that other societies have less direct influence over their governments, who might mimic any power that is asserted by other nations. For example, "the 1994 US solution of building wiretapping capabilities into switches was rapidly taken up in many other parts of the world under the generic name 'lawful intercept'" [15]. Additionally, participants in an Access Now panel about encryption noted that "officials in [a few countries] are, either purposefully or not, using factually inaccurate information about U.S. law to justify their own laws and policies" and that the citizens of those countries "would not be able to rely on protections in U.S. law or the Constitution" [4].

Geographic restrictions. As a result, it may be desirable to contextualize access. We remark that our abrasion puzzles can be composed in an 'onion' style where the solution to each puzzle exposes the next one. This composition of puzzles can realize a system with a hierarchy of costs.

- An initial upfront cost per time epoch (cf. §6.4).
- A fixed cost per geographic region.
- A cost per software product or protocol (cf. §7).
- A cost per user targeted by law enforcement.
- A cost per pair of users or communication session.
- Finally, a marginal cost per message.

Building on §2.2, this hierarchy provides additional degrees of freedom to (dis)incentivize attacker behavior as desired. For example, choosing different parameters for each geographic region reduces the incentive for sharing resources or results between countries. Additionally, it provides the opportunity to match the difficulty of the abrasion puzzle with the computing resources available to each nation. Finally, geographic partitioning introduces the ability to compose our work with NOBUS systems in order to reduce further the set of attackers who can possess targeted decryption capabilities.

Incentives to protect secrets. Our constructions place all of the technological onus for actually achieving exceptional access on the government or law enforcement agency who demands the access. Additionally, once a law enforcement agency completes the initial precomputation puzzle, it is crucial to citizens' privacy that the agency protect the resulting secret material from theft by others.

It remains an open question whether a governmental agency can adequately protect the result of the initial abrasion puzzle. Former NSA director Mike McConnell, former US Secretary of Homeland Security Michael Chertoff, and former Deputy Secretary of Defense William Lynn "recognize the importance our officials attach to being able to decrypt a coded communication under a warrant or similar legal authority" but on balance "believe that the greater public good is a secure communications infrastructure . . . without building in means for government monitoring." They reach this conclusion in part due to the risk of the government being hacked and losing its secrets, a risk that is "evidenced by major cyberintrusions into supposedly secure government databases and the successful compromise of security tokens held by a major information security firm" [69].

This work relies on the power of incentives, both positive (targeted decryption capability) and negative (key agility, cf. §6.4) to spur law enforcement to modernize its technological capabilities and design a digital protection mechanism that adequately protects its secrets at rest and in use. This is a modernization that has not yet happened. Landau [58] observes that, so far, "the [US] FBI has not adapted to the digital revolution," and even current FBI Director Wray admits that his organization "needs to focus more on innovation" in hiring, training, and the formation of partnerships to adapt to technological changes [105]. To reduce the risk of compromise of law enforcement secrets, it is likely that existing cybersecurity best practices must be augmented with advances in cryptographically secure computing platforms [18], [46] and trusted hardware execution enclaves [31], [32]; we remark that such an endeavor will take time to standardize and audit properly by the security community.

2.4. Alternate Recourses for Law Enforcement?

There exist a variety of (hypothesized or realized) alternative technologies that law enforcement can use in a world without targeted decryption. All of them have their own cost-benefit tradeoffs and ethical issues.

First, the government may exploit naturally-occurring vulnerabilities in endpoint software. A recent RAND study [3] estimated the cost to find and exploit a vulnerability at $30,000. One ethical concern here is the potential that such vulnerabilities may either be re-sold to or stolen by cybercriminals [6]; there exist widely-varying estimates of the rate at which this occurs [3], [53].

Second, the government may compel companies to insert deliberate systems-level weaknesses or “backdoors” into their products. This is how Apple CEO Tim Cook characterized the request that the FBI made of his company regarding the San Bernadino shooter’s phone [29]. Because the deliberate introduction of new vulnerabilities creates new risks for everyone, Bellovin et al. [15] argue that “exploitation of naturally occurring bugs in the platforms being used by targets may be a better alternative.”

Third, weaknesses may be introduced within cryptographic algorithms themselves. Several works demonstrate that, conceptually, many modern cryptosystems are vulnerable to being weakened undetectably [14], [88], [107]. As with the above two options, this approach is susceptible to re-purposing by cybercriminals and thus it increases security risks for everyone. Furthermore, Smith and Green note that, once such a weakness is discovered, then “the damage done to the reputation of the standardisation committee and the knock-on effect on business may be severe.”

Finally, a government can simply ban encryption outright or mandate encryption schemes with known deficiencies or short key lengths [39, §4]. By definition, this option leads to a strictly worse set of outcomes for the general public than targeted decryption [91].

2.5. Alternate Recourses for Technologists?

We conclude by examining whether governments’ requests for targeted decryption capabilities even suffice to solve their underlying wish for access to data. Put differently: we survey research areas with the potential to thwart access by law enforcement *even if* key crumpling is deployed ubiquitously.

For data in transit, network anonymity techniques reduce or eliminate law enforcement’s ability to identify which communications correspond to a person of interest. There exist information-theoretic mechanisms (i.e., without any crypto to crumple) that hide metadata to a provably-safe level using padding or differential privacy [59], [100], [103]. It is possible that even weaker solutions that simply introduce ‘chaff’ may suffice; while these types of systems are usually de-anonymizable [40], composing them with our crumpling technique might yield an intractable cost to execute a re-identification attack.

For data at rest, secret sharing data across multiple jurisdictions can complicate the effort of reconstruction [26]. Additionally, rate-limiting techniques based on online verification or hardware-enforced counters [7] may restrict the ability of law enforcement to expend its marginal resources in order to conduct the brute force search, or at the very least increase detection by requiring that they first obtain a device in the physical world. Finally, devices can leverage forward secrecy to provide simple, hardware-enforced deletion of all data at a moment’s notice [92].

We posit that all of these technologies are worth developing and deploying to the public whether or not governments possess the capabilities discussed in this paper.

3. Definitions

This section provides a cost-based definition of message privacy that can be fine-tuned expressly in order to permit attackers with sufficient resources to defeat its guarantees. We formalize new notions of secrecy (independent from any candidate construction) for encryption schemes that permit targeted decryption.

3.1. Symmetric Key Encryption

This work provides new security notions for encryption schemes that provide tunable confidentiality. We start with any symmetric key encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with the following properties:

- It satisfies indistinguishability under a chosen plaintext attack (IND-CPA), potentially against nonce-respecting adversaries.
- Its KeyGen routine samples uniformly from $\{0,1\}^\lambda$.
- It is *K-secure* [37, Def. 3.1]. In particular, its confidentiality degrades gracefully if a key is sampled from a distribution with smaller entropy. For example, encryption modes instantiated with the block cipher AES are believed to be K-secure [38].
- Either Dec or the structure of the plaintext message space enables an attacker to distinguish with high confidence between decrypting with the correct key and decrypting with any incorrect key.
- Optionally, $\mathcal{E}$ may include a key evolution routine $\mathsf{Evolve} : \{0,1\}^\lambda \to \{0,1\}^\lambda$. This evolution mechanism may provide forward (pre-compromise) or backward (post-compromise) secrecy.

We focus exclusively on encryption in this work because, as stated in requirement 4, we maintain integrity of all communications. Ergo, this work follows the authenticated-links model of Canetti-Krawczyk [24], which can be instantiated via (uncrumpled) key exchange and message authentication codes. We note one caveat regarding authenticated encryption: if our crumpling and abrasion puzzles are used within a one-key AEAD scheme, then both confidentiality and integrity will be impacted. Instead, it is preferable to use two-key authenticated encryption schemes such as SIV mode [86] or generic Encrypt-then-MAC [13] where we can crumple the encryption key but leave the authentication key untouched.

3.2. Crumpled and Abradable Secrecy

Suppose that an attacker $\mathcal{A}$ wishes to recover an encryption key, and thus reveal the content of all messages it protects (the set of which depends upon the forward/backward secrecy guarantees provided by Evolve). The listed properties of $\mathcal{E}$ imply that it is both necessary (by K-security) and sufficient (by the distinguishing property) for $\mathcal{A}$ to recover the key via an exhaustive search that costs $2^\lambda \cdot w$, where w equals the cost required to validate a key guess.

Based upon this observation, in this section we characterize an attacker $\mathcal{A}$'s ability to recover secret keys as a function of her available resources. These definitions are presented informally due to space constraints. A rigorous game-based definition bounds $\mathcal{A}$'s ability to recover multiple keys as a function of her available resources, even after adaptively requesting keys of her choice.

We begin by partitioning attackers into two categories based on the resources at their disposal.

Definition 3.1. Let $\mathcal{A}$ be an attacker against multiple instances of an encryption scheme with different keys. We say that $\mathcal{A}$ is (I, M)*-capable* if she has made an initial investment of I resources and subsequently chooses to spend M resources per instance attacked. Conversely, we say that $\mathcal{A}$ is (I, M)*-bounded* if she either has not yet spent I resources upfront or is unwilling to devote M resources per instance. If $I = 0$ then we simply say that $\mathcal{A}$ is M*-bounded.*

We next provide a definition of secrecy that applies in the $I = 0$ case. In this setting, we want to ensure that resource-constrained adversaries cannot recover messages.

Definition 3.2 (Crumpled IND-CPA secrecy)**.** Let $\mathcal{E}_c$ be a symmetric key encryption scheme. We say that $\mathcal{E}_c$ satisfies (M, ϵ)*-crumpled secrecy* if for all δ it is the case that any δM-bounded attacker $\mathcal{A}$ has at most $\max\{\delta, \epsilon\}$ advantage in winning the (optionally, nonce-respecting) IND-CPA game.

Using the argument above about the cost for an exhaustive key search, we can re-cast the standard IND-CPA guarantee into this definition. If a K-secure encryption scheme $\mathcal{E}$ is IND-CPA secure, then it also satisfies $(T_{\mathcal{E}}, \epsilon)$-crumpled secrecy for some $T_{\mathcal{E}} \triangleq 2^\lambda \cdot w$ and some negligible ϵ. In this work, we construct schemes satisfying crumpled secrecy with $M \ll T_{\mathcal{E}}$ so that it is feasible but costly to recover messages.

Next, we define the concrete security of an encryption scheme $\mathcal{E}_a$ whose privacy can be *abraded*, or worn away, by an attacker who has expended substantial resources to acquire a special *abrasion secret* (which need not be unique). We stress that the abrasion secret applies to the entire scheme, and thus it can be used to attack multiple instances of the scheme with different keys.

Definition 3.3 (Abradable IND-CPA secrecy)**.** A symmetric encryption scheme $\mathcal{E}_a$ satisfies (I, M, ϵ)*-abradable secrecy* if it satisfies (M, ϵ)-crumpled secrecy against any attacker who has completed the initial one-time effort and full IND-CPA security against all other attackers (i.e., those who are M-bounded or who lack an abrasion secret).

This definition captures the intuition that the attacker should expend $I + nM$ resources to recover n plaintext messages. It follows the lead of Rogaway's code-constructive mantra to consider the cost of *finding* a suitable strategy to conduct an attack, even one that clearly exists conceptually [84]. Furthermore, it requires that $\mathcal{A}$ account for the cost for this pre-computation; in other words, we only consider the uniform setting [55].

3.3. Quantifying Cost in Dollars

Left unspecified in the definitions above is a precise specification of resource accounting. Cryptographic definitions often measure $\mathcal{A}$'s running time or some combination of time and space [22], [52]. However, determining the feasibility of an attack with these metrics then depends upon $\mathcal{A}$'s physical instantiation [17, §Q.9]. Furthermore, prior attempts to quantify one-time initial costs in a complexity-theoretic manner typically involve measures like Kolmogorov complexity [56] that bear only passing relation to the actual initial costs faced in practice [17, §B.4].

Rather than utilizing any of the above resource metrics, we opt to tabulate resource consumption concretely as the monetary costs spent by an attacker (say, in US Dollars) to construct the appropriate hardware and conduct the attack. We adopt this approach for two principal reasons.

First, it is more intuitively understood by society at large. While mathematicians and cryptographers might understand that the number field sieve breaks Diffie-Hellman over a 1024-bit finite field in $L_p(1/3)$ marginal asymptotic running time [61], most people better comprehend the statement that $\mathcal{A}$ can recover their data using a machine that costs \$100M to build and \$10 per use [5].

Second, it is forced to scale with improvements in attack capability. In particular: even if the security analysis for a cryptosystem remains unchanged, Moore's Law (for general- and special-purpose hardware) causes the costs I and M to change over time. Many cryptographic definitions ignore these details, and implementations respond by purposely choosing security parameters that are deemed to be strong enough to stand for decades. By contrast, this work tunes cryptosystems "on the edge" of being breakable by some attackers and not others; therefore, we must be more precise when accounting for hardware runtime energy costs in §6. As a corollary: our methodology both necessitates and incentivizes the design of cryptosystems with greater agility (cf. §6.4).

4. Symmetric Key Crumpling

Our first set of constructions rely exclusively on symmetric key primitives, and they impose a linear marginal cost M on an adversary without any upfront cost (that is, $I = 0$). The construction transforms $\mathcal{E}$ into a new encryption scheme $\mathcal{E}_c$ with a smaller effective key length. Our construction only requires appending a new *crumpling* algorithm to the end of KeyGen and Evolve. Looking ahead to §7, note that we intend to deploy this crumpling technique in systems that use each (crumpled) key to encrypt only a single message.

4.1. The Concept of Crumpling

Given a symmetric key encryption scheme $\mathcal{E}$, we produce a new crumpled encryption scheme $\mathcal{E}_c$ that works exactly as $\mathcal{E}$ does except with a new crumpling routine added to the end of KeyGen and Evolve. This new routine transforms a key k_0 produced by the original encryption scheme into a new key k_1 that is also λ bits long but that has an *effective* key length of only $\ell < \lambda$ bits. Concretely, k_1 shall be chosen from a subset $S \subset \{0,1\}^\lambda$ where $|S| = 2^\ell$ and S is known to the attacker. Because the encryption scheme is K-secure, the cost of a brute-force search of k_1 is proportional to 2^ℓ.

Keeping k_1 the same bit length as the original key allows us to leave the rest of the host application unchanged. On the flip side, this key crumpling algorithm gives us freedom to choose any effective key length $1 \le \ell \le \lambda$, and the wide spectrum of options for S consistent with the size requirements allows for several constructions with different features and security guarantees.

The constructions in this section use two cryptographic hash functions: a "big" hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$ and a "little" hash function $h : \{0,1\}^* \to \{0,1\}^\ell$. The concrete instantiation in §4.4 uses $\lambda = 256$ and $H(x) = \texttt{SHA256}(\texttt{SHA256}(x))$.

4.2. A Naïve Key Crumpling Algorithm

In the most naïve formulation of our approach, we could simply define our key crumpling function as $k_1 = H(h(k_0))$. The little hash function h projects k_0 down into a subspace of just ℓ bits. The big hash function H then maps it from the small space back to the original key space, so that the crumpled key can be used in the host program with its original choice of ciphers without further modification.

Cost to recover k_1. The adversary can brute-force search the entire key space as follows. For each candidate digest $d_i \in \{0,1\}^\ell$, $\mathcal{A}$ first computes a candidate key $k_i = H(d_i)$ and then attempts to decrypt the message using k_i. If successful, she stops the search and outputs the key. The cost of this procedure is $M = 2^\ell u$, where u is the cost incurred by performing one H and one Dec operation.

Bounded cost security. Because $\mathcal{E}_c$ is K-secure, we claim that M-bounded attackers cannot break the scheme using key k_1 as long as H is collision-resistant even on the restricted subspace $S = \{0,1\}^\ell$. Because the output space of H is larger than S, this property may trivially be satisfied if no collisions exist in $H|_S : \{0,1\}^\ell \to \{0,1\}^\lambda$. Generically, a sufficient criterion to satisfy this property for all subsets of size 2^ℓ is for H to be a computational extractor [12], [36]. Looking ahead, our formal proof in Theorem 4.1 uses the stronger random oracle model to keep track of costs.

Rational security. The resource bounds above restrict which adversaries are capable of recovering a crumpled key. From the perspective of rational cryptography [47], it also follows from the resource bounds that, even if a rational adversary can recover a message, she will not actually choose to do so unless the expected benefit she gains from learning the plaintext exceeds her cost to obtain it. In this paper, §6 gives estimates of the adversary's real-world costs. In future work, we plan to explore a more thorough game-theoretic economic analysis of the problem.

Limitations. This naïve approach has some important time-memory tradeoffs when multiple instances are composed. For example, if ℓ is sufficiently small, then an attacker can simply store all 2^ℓ candidate keys when she performs the first brute-force search. Then for all subsequent attacks, she does not need to re-run H again; instead she only needs to perform the trial decryptions. Put another way, some of the work thought to be marginal might in fact be a one-time initial cost.

4.3. Practical Symmetric Key Crumpling

We can improve on the naïve version of the key crumpling algorithm by drawing each key from a distinct 2^ℓ-sized subset of $\{0,1\}^\lambda$. First, we tie the search space to each pair of communicating parties by including labels for the sender and the receiver in the final hash computation. For a message sent from Alice to Bob, we set $k_1 = H(h(k_0) \mid\mid A \mid\mid B)$, where A is an address or identifier for Alice and B is the same for Bob. This forces the adversary to perform the full brute-force search for each pair of users.

The identifiers A and B should be public, in the sense that anyone who can observe the encrypted message can either extract them from the message itself or can infer them from its context. For example, the identifiers might be IP addresses, usernames, email addresses, public keys, or cryptographic hashes of any of these basic identifiers. For messages with multiple recipients, the receiver identifier B could simply be a list of the recipients' individual IDs.

We note that this requirement for identifiers does *not* preclude the protection of anonymity or pseudonymity in the host protocol. For example, in the Signal Double Ratchet protocol, each participant has a long-term Diffie-Hellman key pair that is used to identify her to other users and to sign ephemeral keys. These public keys make good identifiers in our protocol because they are public in the host protocol, they are stable over a relatively long period of time, and they are not bound to any real identity without some other source of auxiliary information.

Nonces. If the adversary is interested in recovering a large number of messages for a given (sender, receiver) pair, she still may be able to save on computation by precomputing the hashes if ℓ is sufficiently small. We remedy this by also including a nonce n in the big hash. Like the identifiers A and B above, the nonce must also be publicly known to the legitimate recipient(s) and to any third party observers.

One easy source for our nonce is a timestamp. For example, in PGP encrypted email, the header fields for each mail, including `TO`, `FROM`, `SUBJECT`, and `DATE`, are sent in the clear. If we create the message headers before we encrypt the body, we can use the exact timestamp from the `DATE` field to generate the crumpled key. In the Signal protocol, messages do not have a cleartext timestamp, and transmission delays make it impossible for the receiver of a message to determine exactly when it was sent. Fortunately Signal does include in each message its message number in the sending chain. Together with a coarse timestamp, e.g., one truncated to the nearest hour, the message number can provide a suitable nonce.

Multi-Channel Support. Some applications may want to provide secure communications over multiple channels at the same time, for example providing both encrypted voice over IP and text chat in the same application. In this case we can also include a channel identifier c in our big hash, so that the adversary cannot re-use work done to break one channel to more easily break another channel in the same application.

Putting it all together. Combining the above techniques, we have $k_1 = H(h(k_0) \mid\mid n \mid\mid c \mid\mid A \mid\mid B)$. We design a concrete header format that follows this approach in §4.4. First though, we formally prove the crumpled secrecy of this $\mathcal{E}_c$ construction in the random oracle model.

Theorem 4.1. *Let H be a random oracle whose cost is v per query, and let $M = 2^\ell v$. Additionally, let $\mathcal{E}$ be a K-secure symmetric key encryption scheme that satisfies IND-CPA, or in other words $(T_\mathcal{E}, \epsilon)$-crumpled secrecy where $T_\mathcal{E}$ equals 2^λ times the cost to verify a key guess. Then, $\mathcal{E}_c$ satisfies $(M, 3\epsilon)$-crumpled secrecy.*

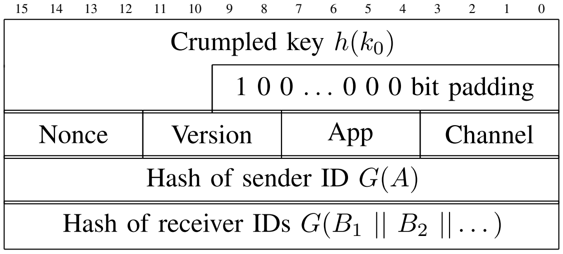
At a high level, the random oracle provides two benefits here: correlation intractability and a concrete method to measure costs. We use both properties in the following proof.

Proof. The nonce components of the header ensure that different instances of the symmetric key encryption scheme (i.e., different outputs of KeyGen or Evolve) feed different input to H, and the random oracle property in turn ensures that the resulting crumpled keys are independent. Hence, it suffices to consider the success probability against a single crumpled key by an attacker who expends $M' = \delta M$ resources.

Set γ such that γM denotes the resources that $\mathcal{A}$ expends calling its random oracle. With this expenditure, $\mathcal{A}$ can make $c = \gamma M / v$ calls to the oracle and thus recover k_1 with probability $2^{-\ell} c = \gamma$.

If $\mathcal{A}$ never recovers k_1 through oracle queries, then she learns nothing about k_1 since the random oracle is an all-or-nothing transform [21]. In this case, k_1 appears to come from a distribution with entropy λ, in which case $\mathcal{A}$'s advantage at breaking $\mathcal{E}_c$ and $\mathcal{E}$ are identical. Furthermore, oracle queries don't help $\mathcal{A}$ to understand the Enc or Dec routines since they are independent of the random oracle.

The attacker may then use its remaining $(\delta - \gamma)M$ resources to attack $\mathcal{E}_c$, which by K-security is no better than using these resources to attack an ideal encryption scheme. Hence, $\mathcal{A}$ has $\max\{\delta - \gamma, \epsilon\}$ advantage since $\mathcal{E}_c$ satisfies bounded security. Finally, a union bound limits $\mathcal{A}$'s overall distinguishing advantage per instance to 3ϵ, as desired. □



$$G(x) = \texttt{SHA256}(x) >> 128$$
$$h(x) = (\texttt{SHA256}(x) >> (256 - \ell)) << (256 - \ell)$$
$$k_1 = \texttt{SHA256}(\texttt{SHA256}(\textsf{header}))$$

Figure 1. Symmetric crumple zone header format and specification of the hash functions utilized.

4.4. A Concrete Instantiation

Here we give a concrete realization of our symmetric key crumpling scheme that can be easily retrofitted onto applications that use symmetric encryption to protect the confidentiality of data in transit or at rest. In §7 we show how this scheme can be applied to the PGP, ZRTP, and Signal message transit protocols, as well as to the Linux file-based encryption method used in Android.

Fig. 1 shows our prototype construction. It purposely follows the same 80-byte block structure and "double SHA-256" hash function design used within the Bitcoin header specification [73]. Concretely, we set the big hash function H as "double `SHA256`", and we construct the smaller hash function h as a length-truncated `SHA256`. The output of h is provided as one part of the header that is fed into H. Because our header is built to support any choice of ℓ, we allocate 32 bytes of space in the header for the output of h and use 10^* bit padding to fill the remaining space.

5. Public Key Abrasion

The crumpling construction in the previous section results in a linear expected cost M to attackers. Anyone who is willing and able to bear this cost may decrypt as many messages as she can afford to crack.

In order to keep M not too large for legitimate use by law enforcement agencies, while still discouraging non-state actors, in this section we present a proof-of-concept public key *abrasion* algorithm that disincentivizes all but the largest attackers from investing in a key cracking effort. The idea is to impose a large upfront cost I, which even the developers of an encrypted messaging system cannot sidestep, on any attacker who wishes to recover messages encrypted with the symmetric crumpling scheme. Under our new encryption scheme, denoted $\mathcal{E}_a$, it is only rational for the attacker to attempt to recover a set of messages $m_1, ..., m_n$ if the expected value of the messages is greater than the upfront cost plus the sum of the per-message costs.

5.1. Background: Diffie-Hellman and Logjam

Our approach uses a novel application of Diffie-Hellman key exchange over modular arithmetic groups (as opposed to elliptic curve groups). Adrian et al.'s "Logjam attack" [5] demonstrates that the number field sieve algorithm to solve the (non-elliptic curve) Discrete Logarithm problem can be split into a one-time precomputation phase that requires a large-but-feasible initial effort I followed by near-zero marginal cost per instance attacked. RSA is similarly vulnerable to number field sieving by attackers willing to invest in dedicated hardware [97].

In more detail: Adrian et al. compute discrete logarithms modulo a large prime p. To do so, the attacker must first perform an expensive precomputation that depends *only* on p. Then, given any two elements g and $y \bmod p$, the attacker can perform a relatively simple descent step to recover the discrete logarithm $\log_g(y)$. The cost I of the initial sieving step depends upon the Diffie-Hellman security parameter; for some of the values observed in practice today, I is tunable in the range of millions to billions of dollars; see Figure 4 and [5, §4]) for more details.

To mitigate this concern, Adrian et al. recommend that users of 1024-bit Diffie-Hellman upgrade to larger primes (e.g., 1536 or 2048 bits) or switch to the elliptic curve version of the algorithm, which is not vulnerable to the attack.

5.2. Practical Public Key Abrasion

We propose to bring back Diffie-Hellman (D-H) key exchange over modular arithmetic groups in a controlled way, where the resulting key is only used to protect the crumpling puzzle. Any attacker who wishes to recover a message's symmetric encryption key must have first performed a successful precomputation attack. An encrypted messaging provider can select the public parameters without gaining any special ability to learn the D-H shared key or to perform the precomputation.

In our approach, the author of each encrypted messaging protocol samples a safe prime p of the appropriate size (e.g., about 1024 bits). Then, whenever the host protocol specifies a Diffie-Hellman key exchange (perhaps even using an elliptic curve group) to derive a shared symmetric key k_0, we add to this a second *abradable* D-H exchange using arithmetic mod p in order to derive another secret S_a. To avoid introducing a new round-trip latency to the host protocol, the new abradable D-H round can be performed in parallel with the native (strong) D-H. Finally, we add the shared secret S_a from the abradable D-H to the list of inputs needed to derive the crumpled symmetric key:

$$k_1 = H(h(k_0) \mid\mid t \mid\mid c \mid\mid A \mid\mid B \mid\mid S_a) \tag{1}$$

If an attacker $\mathcal{A}$ has completed the log precomputation attack for the prime p, then with only a small amount of additional work, she can then compute discrete logs mod p to recover the shared secret S_a from the abradable D-H key exchange. She can then proceed with the brute force attack on $h(k_0)$ to recover the crumpled key k_1.

However, to any attacker who does not have the results of the log precomputation work, the 256 bit long string S_a is (1) protected from discovery thanks to the hardness of discrete logarithms and (2) contributing toward the infeasibility of a brute force attack on H. As a result, the security of the crumpled key k_1 is equivalent to that of the original, uncrumpled key k_0.

We can make the upfront work arbitrarily expensive by varying the length of the prime p. Adrian et al. speculate that in 2015 some governments may already have had the capability to attack 1024-bit abradable D-H [5]. And in 2016, academic teams succeeded in computing a discrete log modulo 768-bit and 1024-bit safe primes [45]. We present and discuss more detailed estimates for the expected costs incurred by the attacker in §6.3.

We restrict our attention in this section to a single abrasion puzzle. We remark that Eq. (1) can be extended to incorporate the secrets from several abradable D-H executions in order to realize the hierarchy of costs delineated in §2.3.

5.3. Security Analysis

To analyze this protocol formally, we introduce a new assumption about Diffie-Hellman key exchange over modular arithmetic groups. In essence, this assertion claims that the number field sieves at the heart of the Logjam attack offer the best opportunity to amortize costs between instances.

Assumption 5.1. *An attacker $\mathcal{A}$ can only find the shared secret key produced by Diffie-Hellman key exchange mod p if she has first performed the precomputation attack for the prime p. We denote any result of a precomputation attack as an* abrasion secret *for this system and we let I denote the resources required to perform this attack.*

We stress here that we have little evidence that this assumption is true. While the assumption does encode the best publicly-known attack against Diffie-Hellman today, it is unclear whether better attacks exist. Put differently: one consequence of designing cryptosystems "on the edge" of being breakable is the need for more fine-grained cryptanalysis. Nevertheless, our construction in Eq. (1) is quite modular, so we believe that if someone finds a different technique that requires a large one-time precomputation then it should be possible to substitute that in place of this assumption.

For now, we proceed with our proof-of-concept assumption and show that it suffices to construct an encryption scheme with abradable secrecy.

Theorem 5.2. *Let H be a random oracle whose cost is v per query, let $M = 2^\ell v$, and let I denote the cost required to find an abrasion secret as per Assumption 5.1. Additionally, let $\mathcal{E}$ be a K-secure symmetric key encryption scheme that satisfies IND-CPA, or in other words $(T_\mathcal{E}, \epsilon)$-crumpled secrecy where $T_\mathcal{E} \gg M$. Then, $\mathcal{E}_a$ satisfies $(I, M, 3\epsilon)$-abradable secrecy.*

Proof. Suppose that an attacker $\mathcal{A}$ can break the $(I, M, 3\epsilon)$-abradable secrecy of $\mathcal{E}_a$. Then we claim that $\mathcal{A}$ must be

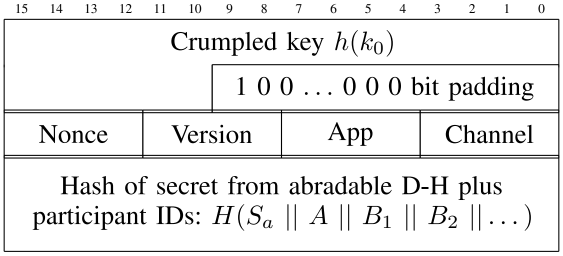

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Crumpled key $h(k_0)$
1 0 0 . . . 0 0 0 bit padding
Nonce
Version
App
Channel
Hash of secret from abradable D-H plus participant IDs: $H(S_a \mid\mid A \mid\mid B_1 \mid\mid B_2 \mid\mid \ldots)$

Figure 2. Asymmetric crumple zone header. Note that h and k_1 are defined identically as in Fig. 1.

(I, M)-capable: that is, she is willing to spend M resources per instance attacked and she must possess an abrasion secret.

To the first point: the crumpled secrecy of $\mathcal{E}_a$ is equivalent to the crumpled secrecy of $\mathcal{E}_c$, so by Theorem 4.1 the attacker must be M-capable.

To the second point: we partition into two sub-cases depending on whether $\mathcal{A}$ has queried H on a preimage of k_1 that follows the format of equation (1). If so, then we can extract a secret from $\mathcal{A}$, which means that $\mathcal{A}$ must possess an abrasion secret by Assumption 5.1. If not, then the oracle queries are independent of the crumpling mechanism used to produce k_1 and thus $\mathcal{A}$'s advantage in winning the IND-CPA game on $\mathcal{E}_a$ and $\mathcal{E}$ are identical. □

5.4. A Concrete Instantiation

Figure 2 shows a concrete instantiation of our scheme that can easily be retrofitted onto many existing protocols. To help us predict our attackers' real-world performance, we keep the two main parameters that we borrowed from Bitcoin: the 80-byte header and the double-SHA256 hash function. To fit all parameters within 80 bytes, we deviate slightly from Eq. (1) by having the final 256 bits of state correspond to the hash of S_a concatenated with participant IDs. In the random oracle model, this alteration has negligible effect on Theorem 5.2 since H is collision-resistant and its output has as much entropy as its input.

6. Concretely Estimating Law Enforcement Costs

Here we use publicly available information to derive engineering estimates for the baseline costs for a real key cracking operation that would be needed in national-level law enforcement agencies like the US Federal Bureau of Investigation or the UK's Security Service MI5. We stress that our preliminary estimates in this work are intended only to demonstrate the basic feasibility of the concept. Further analysis would be required in order to choose parameters for a real system that could be deployed at (inter)national scale.

6.1. Cost to Recover Symmetric Crumpled Keys

In this initial feasibility analysis, we focus solely on estimating the electricity cost required for the hash function $H()$ in the brute-force recovery of a symmetric key. Of course, a real adversary would incur many other costs as well, just like the operator of any other data center(s). These include: the initial purchase of processing hardware for the key cracking effort; hardware replacement at regular intervals to keep up with Moore's Law; hardware and software for networking, power distribution, and system management; cooling; facilities; and people to run it all.

Type	Vendor	Product Name	Efficiency (MH/J)
ASIC	BitMain	AntMiner S9	10,204.1
FPGA	FPGA Mining	X6500	24.3
GPU	nVidia	GTX 1080	5.8
CPU	nVidia	Tegra2	1.1
CPU	AMD	Ontario C-50	0.7
CPU	Intel	Core i7 6800K	0.5

TABLE 1. Bitcoin mining performance of CPU, GPU, FPGA, and ASIC systems [19], [35]

We designed the parameters of our symmetric key crumpling algorithm to match important parameters of the hash-based proof of work required for mining the crypto currency Bitcoin [73]. We did this in order to use the performance of various Bitcoin miners to predict our adversary's brute force search rate with high confidence. Because Bitcoin has been very popular for many years, with substantial financial incentives for fast and efficient mining, it is reasonable to assume that the performance of today's specialized mining hardware is very close to what a well-resourced adversary could achieve.

Table 1 shows information from the Bitcoin project [19] and others [35] on the mining performance of various CPUs versus the best GPU, FGPA, and ASIC for which we could find data. One important observation from this data is the large difference in efficiency between ASIC and CPU-based miners—the ASICs are more than 20 thousand times more efficient than a recent high end Intel CPU. These results are possible because the current ASIC miners are the result of many years' research and development in response to Bitcoin's substantial financial incentives. While it is possible that a well-funded adversary could design and build a more efficient miner, it is unlikely that such an effort could achieve the orders of magnitude improvement necessary to enable mass decryption of crumpled messages. For example, the recent AsicBoost technique [51] gives only a 20% increase, and a group of previous optimizations in 2014 yielded only 37% [34].

To estimate the adversary's monetary cost, we use pricing data from the US Energy Information Administation [102]. For the US as a whole in 2016, electricity cost about 10 cents per kilowatt hour. In the cheapest area of the country (West South Central), the average price for industrial customers was just over half the national average, at 5.23 cents per kWh in November 2016, so our adversary has a large incentive to locate his data center where power is cheap. Using this lowest average cost of 5.23 cents per kWh, we computed the adversary's expected cost to recover keys with an effective length ranging between 30 and 90 bits. Figure 3 shows

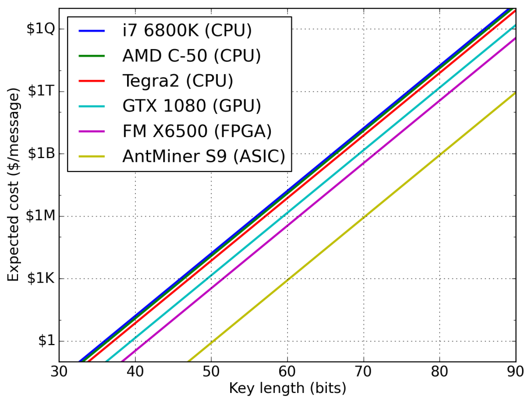


Figure 3. Adversary's expected electricity costs for symmetric keys.

Category	**2016 ($M)**	**2017 ($M)**
Cyber	541.4	626.5
Going Dark	31.0	69.3
Data Center Transformation	0.0	85.0
Total FBI Budget	8,798.8	9,502.4

TABLE 2. US FEDERAL BUREAU OF INVESTIGATION (FBI) REQUESTED 2017 BUDGET [28]

the adversary's expected electricity costs for each of the processing hardware choices from Table 1. (Note that the Y axis is logarithmic.)

Limiting Law Enforcement Access. To put the costs above into context, Table 2 shows selected numbers from the US Federal Bureau of Investigation's proposed budget for 2017. Of special interest for this work, the FBI is requesting to double the budget allocated to the problem of *going dark*—that is, "the inability to access data because of challenges related to encryption, mobility, anonymization, and more."

Comparing the FBI's budget to the costs in Figure 3 shows that symmetric key crumpling gives the public a practical mechanism for limiting the government's surveillance capabilities. For example, suppose the FBI applied their entire $9.5B budget for 2017 to buy electricity for a cluster of AntMiner S9's. Then, based on the data in Figure 3 and Table 2, they could recover on average fewer than ten 80-bit keys per year. More realistically, if we set the effective key length to just 70 bits, then still the entire $69.3M "going dark" budget could recover fewer than 70 keys per year with the same hardware.

For another perspective on this problem, we can compare the above results with data from large online services' transparency reports for 2016. Table 3 shows the number of users/accounts on whom the US government requested information from Facebook, Google, Microsoft, Twitter, and Yahoo in 2016. If the FBI applied their entire 2017 "going dark" budget to investigating just these accounts, they would have less than $400 for every user/account request. Therefore an effective key length as small as 60 bits ($1K/key on the AntMiner ASIC) appears sufficient to motivate even a large, well-resourced agency like the FBI to choose their targets carefully.

Online Service		1H 2016	2H 2016	Total 2016
Facebook	[44]	38,951	41,492	80,443
Google	[49]	30,123	27,272	57,395
Microsoft	[71]	13,755	10,533	24,288
Twitter	[99]	8,009	5,618	13,627
Yahoo	[106]	9,408	7,745	17,153
			Total	192,906

TABLE 3. US GOVERNMENT REQUESTS FOR USER/ACCOUNT DATA

User's	Text messages per month		
age bracket	Sent	Received	Total
18–24	2022	1831	3853
25–34	1110	1130	2240
35–44	831	726	1557
45–54	525	473	998
55+	247	244	491

TABLE 4. SMS MESSAGING USE BY AGE GROUP [43]

Discouraging Misuse. Another common concern is that a system for exceptional access might be abused by corrupt or overzealous law enforcement officials to surveil inappropriate targets, such as politicians, celebrities, or officers' personal love interests. Even a moderate per-message cost appears sufficient to limit the potential for such misuse to go undetected. In 2011, a study from Pew Research Center found that mobile phone users in the US sent or received an average of 41.5 messages per day, while the median user sent or received 10 messages per day [90]. Another study from credit reporting agency Experian in 2012 showed that text message volume was highly dependent on user's age [43]. In that study (Table 4) even the oldest, lowest-volume cohort of users generated nearly 500 messages per month. At $1M or even $1000 per message, the cost to monitor a user for any length of time rapidly grows too large to conceal.

Discussion. Although the final choice of an effective key length is beyond the scope of this paper, the results in this section indicate that reasonable values for the US can probably be found somewhere in the range between 60 and 80 bits.

6.2. Hardware Costs for Timely Access

In certain kinds of cases, including most notably terrorism investigations, law enforcement agencies sometimes need to recover message contents within a very short window of time. We reiterate that our design provides the LEAs with no special mechanism for timely access, but neither do we seek to prevent it. Any agency which requires quick access to encrypted messages must simply obtain sufficient hardware to perform the brute-force computation within the time available.

In general, to recover a message encrypted with an effective key length of ℓ bits, the LEA must compute up to 2^{ℓ} hashes in the worst case. The key search problem is an "embarrassingly parallel" workload, so the LEA can reduce the time required to find the solution by simply adding more processors. If the LEA assembles a cluster of AntMiner

S9's with an average power draw of 1.25 kW, then searching a key space of 60 bits requires a total power budget of

$$\frac{2^{60}\text{ Hashes}}{1.02\times 10^{9}\text{ Hashes/Joule}}\times\frac{1\text{ Wh}}{3600\text{ Joules}}\approx 314\text{ kWh.}$$

or about 251 device-hours. Therefore a cluster of 251 devices would be sufficient to recover the key in about an hour. Assuming an average price of US $2000 per device, a cluster of 250 units would cost about half a million dollars.

With a 70 bit key space, the LEA would need $2^{10} = 1024$ times as many units to complete the brute force search within the same amount of time. The 257209 devices required for such a cluster would cost about $514 million. A much smaller cluster of just 10717 devices could recover the key within one day, for an upfront cost of about $21.4 million.

For a key space of 80 bits, the capability to recover a key within one hour becomes prohibitively expensive. Such a cluster would require over 263 million devices and would cost almost $527 billion. On the other hand, if the LEA can afford to wait a week to recover any message, then they could get by with a cluster of about 1.57 million devices for an upfront cost of just over $3.1 billion.

6.3. Cost to Recover Abradable D-H Secrets

The Logjam attack on Diffie-Hellman over modular arithmetic groups requires a large up-front precomputation based on the prime modulus p. Adrian et al. [5] estimate that an adversary could design a special-purpose ASIC for accelerating the matrix math used in the precomputation step, and could fabricate sufficiently many chips for a large cluster, for a total of about $8 million. They estimate the cost for the initial computation for a 1024-bit safe prime to be about $11 billion if the adversary uses general-purpose CPUs, or about 80 times cheaper (ie, about $137.5 million) if using ASICs. All together, the adversary's startup costs come to about $145.5 million.

We can also increase the upfront cost to the adversary by increasing the size of the Diffie-Hellman prime modulus. According to [5], precisely estimating the computation required for the number field sieve with prime groups of around 1024 bits is difficult due to the complexity of the algorithm and the trade-offs that can be made to optimize various subroutines versus one another. Like Adrian et al. [5], we fall back to a coarse approximation from asymptotic complexity. For a prime modulus p, the complexity of the precomputation step is $\exp((k + o(1))(\log p)^{1/3}(\log\log p)^{2/3})$. Fitting that curve to match the estimated cost of $137.5M for a 1024-bit key, we arrive at the estimates shown in Figure 4. This (admittedly rough) analysis predicts that we could require about $1 billion in upfront computation by choosing a 1075-bit prime p, and that we could double that cost again by increasing the length of the prime to 1093 bits.

With primes in this range, an adversary should be able to compute discrete logs and recover Diffie-Hellman secrets with only a relatively modest power budget compared to what is required for brute-forcing a symmetric crumpled key. For example, the authors of [5] estimate that a 1024-bit prime would require about 30 core days on a recent CPU.

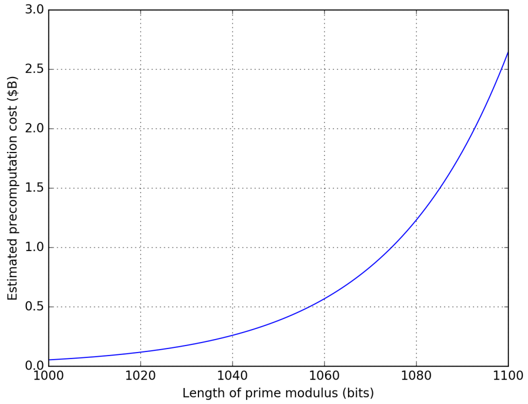


Figure 4. Costs for attacking asymmetric crumpling

6.4. Key Agility and Moore's Law

According to Moore's Law, the density of transistors in an integrated circuit tends to double about every 18 months. As a result, the amount of computation that can be performed for a given energy input tends to increase at roughly the same rate. To keep the adversary's costs constant as computation grows cheaper, we must increase the effective security parameter of our crumpled keys at regular intervals. In practice, the developers of an encrypted messenger app might choose an initial value for the effective key length ℓ based on the costs in §6.1 above. Then the app itself could automatically increase ℓ to keep up with Moore's Law, say by increasing ℓ by one bit every 18 months or two years after the inital release date.

Another consequence of Moore's Law in this model is that old messages are easier to recover than newer messages. If the value of the message degrades over time at least as quickly as the cost to recover its contents, then this is not a problem. No rational adversary would ever choose to waste his resources on such a message. Intuitively, we believe this case encompasses the great majority of messages in the real world.

To protect information that retains its value over time, we must choose a greater initial effective key length in order to maintain an acceptable level of security in the future. For example, suppose the adversary's computational power doubles every 18 months and we want to support messages that will still be worth up to $1000 in 15 years. Then we should choose an initial key length so that the adversary's costs is at least $\$1000 \cdot 2^{15/1.5} = \$1\text{M}$ today.

Another concern is that a law enforcement organization with secrets that enable targeted decryption will inevitably become a target for malicious actors. In our approach, the only long-term secret is the result of the abrasion puzzle. To incentivize proper storage, we advocate for advances in cryptographic engineering agility [74] so users may rapidly shift cryptographic algorithms in case a nation-state loses control of a precomputed secret. Furthermore, shifting to a new Diffie-Hellman prime p at a regular interval to deal with Moore's Law also reduces the impact of the theft of a nation-state's secrets or a cryptanalytic breakthrough in any of the cryptographic primitives used in the abrasion or crumpling puzzles.

7. Applications

In this section, we show how our key crumpling and abrasion techniques can easily be retrofitted onto several popular applications for securing data in transit and at rest.

7.1. PGP and S/MIME Email

PGP [108] encrypts each part of an email message with a random symmetric key. It then encrypts the symmetric key with a public-key encryption, and sends the encrypted key and the ciphertext over email. The competing S/MIME protocol [80] is similar.

The application of symmetric crumpling to these protocols is straightforward. To crumple the original (random) key k_0 for a message from Alice to Bob sent at time t, we compute the new key k_1 as

$$k_1 = H(h(k_0) \mid\mid t \mid\mid c \mid\mid A \mid\mid B)$$

where A is Alice's email address, B is Bob's, and for multi-part MIME messages, the channel identifier c is the MIME part to be encrypted.

7.2. Signal

The Signal protocol encrypts each message with a new, unique symmetric key. It derives these keys using a novel "double-ratchet" construction that uses public-key operations and symmetric operations to provide both forward secrecy and backward secrecy along with message deniability. In this construction, each party holds a set of cryptographic keys for each conversation:

- A master (symmetric) key-derivation key for the conversation, called the *root key*, which we denote as k_R.
- Two symmetric *chain keys* that are used to derive a new encryption or decryption key for each message. One chain key is used for outgoing messages, and the other is used for incoming messages. We denote the chain key for messages from Alice to Bob as k_C^{AB} and the chain key for messages in the opposite direction as k_C^{BA}.
- An ephemeral elliptic curve Diffie-Hellman key pair, called the *ratchet key pair*.
- A long-term asymmetric key pair, called the *identity key pair*, which is used to authenticate its owner. We denote Alice's long-term public key as I_A and Bob's as I_B.

To these we add an ephemeral Diffie-Hellman key pair, called the *abradable key pair*, for use with our public key abrasion algorithm.

Abrading The Diffie-Hellman Ratchet. During the course of a Signal conversation, either party can periodically generate a new ratchet key pair. They then attach their new public ratchet key to the next outgoing message. This invokes the Diffie-Hellman ratchet protocol. When Alice receives a message from Bob that contains a new public key for him, the Diffie-Hellman ratchet step proceeds as follows. We describe our modifications to the protocol in *italics*.

1) Bob generates a new ratchet key pair. He attaches the new public ratchet key on a message to Alice.
   Here we have Bob also generate a new abradable key pair and attach the new public abradable key on the outgoing message, along with his new public ratchet key.
2) Alice computes a new Diffie-Hellman shared secret, SS_{BA}, from her current private ratchet key and Bob's new public ratchet key.
   Here Alice also computes a new abradable D-H shared secret, ss_{ba}, from her current private abradable key and Bob's new public abradable key.
3) The new DH shared secret SS_{BA} is used as input to a PRF keyed with the root key. The output of the PRF gives a new root key and a new chain key for Alice's receiving chain.

$$\begin{aligned} k_C^{BA} &= F(k_R, SS_{BA}||0) \\ k_R &= F(k_R, SS_{BA}||1) \end{aligned}$$

4) Alice then generates a new ratchet key pair of her own. She attaches her new public ratchet key on the next message to Bob.
   Alice also generates a new abradable key pair and attaches the public abradable key on the message to Bob.
5) She uses her new private ratchet key together with Bob's new public ratchet key to derive another new DH shared secret, SS_{AB}.
   She uses her new private abradable key together with Bob's new public abradable key to derive the abradable shared secret ss_{ab}.
6) Alice applies the PRF again, with SS_{AB} as input, to derive yet a newer version of the root key k_R and a new chain key for her sending chain.

$$\begin{aligned} k_C^{AB} &= F(k_R, SS_{AB}||0) \\ k_R &= F(k_R, SS_{AB}||1) \end{aligned}$$

Note that the abradable shared secrets ss_{ab} and ss_{ba} are never used in the derivation of the new root keys or the new chain keys. Instead, they are only used later in the symmetric ratchet to derive the individual message keys.

Crumpling the Symmetric Ratchet. The application of symmetric crumpling to Signal's double ratchet is straightforward. In the symmetric-key ratchet step for a message from Alice to Bob, Signal derives a new chain key k_C^{AB}

and a new message-encryption key k_M as outputs of a PRF keyed with the previous chain key.

$$k_M = F(k_C^{AB}, 0)$$
$$k_C^{AB} = F(k_C^{AB}, 1)$$

The symmetric ratchet is performed on the sending chain key for every outgoing message and on the receiving chain key for every incoming message.

We simply crumple the message key k_M for each message, leaving the chain keys, the root key, and the ratchet key pairs at their full original strength. Then an adversary who recovers k_M for one message learns nothing new about any other message. We can apply the symmetric crumpling either by itself or in conjunction with the public key abrasion described above.

For a message from Alice to Bob, sent at time t with message ID id, we set the nonce $n = \texttt{Truncate}(t) \mid\mid id$. Without public key abrasion, we compute the crumpled message key k_M in the ratchet as:

$$k_M = H(h(F(k_C^{AB}, 0)) \mid\mid n \mid\mid I_A \mid\mid I_B)$$

When using public key abrasion in the Diffie-Hellman ratchet, we also include the abradable D-H shared secret in the computation of the crumpled message key:

$$k_M = H(h(F(k_C^{AB}, 0)) \mid\mid ss_{ab} \mid\mid n \mid\mid I_A \mid\mid I_B)$$

7.3. SRTP

The secure real-time transport protocol [10] is commonly used for encryption of multimedia data, e.g. for voice over IP. Recently popular examples include voice calling and video chat features in WebRTC [81] and the Signal private messenger app. SRTP uses a relatively simple key derivation scheme. It represents a sort of middle ground between PGP, which encrypts an entire email with a single key, and Signal, which generates a new key for every short message in a conversation.

SRTP relies on an external key agreement protocol, such as ZRTP [23] or Signal [67], to first derive a master secret key. Then, for every r^{th} packet, SRTP uses the master key to derive new session keys for encryption and authentication and a new pseudorandom salt. These are used to encrypt the following r packets with a stream cipher like AES-CTR. The key derivation rate $1/r$ is a tunable parameter.

The naive approach would be to crumple the master secret key, but this has some significant drawbacks. Because the authentication keys are derived from the master secret, allowing the adversary to break the master key enables her also to forge messages. Additionally, this approach is too coarse, and it violates our rule that the cost to recover a set of messages should be proportional to the number of messages recovered. Intuitively, it should not cost more to decipher three 1-minute conversations than to recover one 60-minute conversation.

In order to make the adversary's cost proportional to the amount of media he wishes to recover, we can apply our symmetric crumpling function to the session keys. Then, the adversary must do work to recover each period of r packets of media from each party on the call on top of the initial abrasion effort.

7.4. Full Disk Encryption (FDE)

The most ubiquitous disk encryption tools for PCs encrypt data at the block device layer, so the entire storage device is encrypted with a single key. The actual disk encryption key is typically taken from a random number generator. For ease of use it is stored in a header on the disk, encrypted (wrapped) with another key derived from a password or some other more usable key generation mechanism. Systems that employ this approach include Linux `dm-crypt`, Apple FileVault, and Microsoft BitLocker.

The adaptation of our symmetric key crumpling approach to these systems is relatively straightforward. We leave the disk encryption key at its full, original strength, but we crumple the key wrap key that protects it. As in the previous sections, we use cleartext identifiers to tie the search space for a device's key wrap key to that particular volume. For a disk with hardware model number M and serial number S, when we encrypt partition number P with application A, we compute the crumpled key wrap key k_1 as:

$$k_1 = H(h(k_0) \mid\mid M \mid\mid S \mid\mid P \mid\mid A)$$

where k_0 is the original key wrap key. Here k_0 may be derived from a passphrase or a hardware token, or it may be pseudorandom. Public key abrasion can be added to this construction in a straightforward manner as shown in §5.

7.5. File-Based Encryption (FBE)

Recent versions of both Apple iOS and Google Android use file-based encryption rather than full disk encryption. With FBE, each file is encrypted with its own unique symmetric key, which in turn is derived from a master key and a nonce for the given file.

Here we describe the application of our symmetric key crumpling technique to the Linux kernel's file-based encryption implementation used by Android in the Ext4 filesystem [30]. Ext4 FBE encrypts the contents of each file with AES-256 in XTS mode. Due to a quirk in the specification of XTS mode, this requires two AES-256 keys per file. Following the recommendation of Liskov and Minematsu [64], we derive a single 256-bit key and use it for both "halves" of the XTS key, as in the original XEX construction [83] on which XTS is based. According to [64], this approach does not actually reduce the security of the scheme, and it allows us to keep our Bitcoin-based cost estimates.

Rather than simply crumpling the master key, we leave the master key at its full strength and instead crumple the individual keys for each file. Then the mobile device must do a small amount of additional work. Each time a file is opened, the device must derive the file's original full-strength key from the master key and the nonce, then it must derive the crumpled key from the original key. Once a crumpled key has been computed, it can be cached in memory in the kernel for future use.

As in previous sections, we use publicly available information about the file to tie the search space for its crumpled key to that particular file. Given a file having nonce N, inode number I, creation time t, original un-crumpled key k_0, residing on a filesystem with UUID U, we compute the crumpled key as

$$k_1 = H(h(k_0) \mid\mid N \mid\mid I \mid\mid U \mid\mid t)$$

Once again, adding public key abrasion to this construction is straightforward.

7.6. Transport Layer Security (TLS)

Many application protocols, e.g. HTTP, IMAP, and SMTP, rely on TLS to protect long-term secrets like passwords and session cookies. An adversary who can recover a TLS encryption key could potentially use these secrets to gain long-term, read-write access to the targeted user's account. This violates our philosophy that the level of access achieved should be proportional to the resources expended by the adversary. Therefore we do not recommend the use of crumpling with TLS.

8. Conclusion & Future Work

This work shows the technological feasibility of constructing a system that balances individual privacy with limited, targeted access by law enforcement. The system imposes minimal computing and networking burden on users and developers, and it does not require either of them to communicate with law enforcement. We view this work as a catalyst that can inspire both the research community and the public at large to explore this space further.

Whether such a system will ever be (or should ever be) adopted depends less on technology and more on questions for society to answer collectively: whether to entrust the government with the power of targeted access and whether to accept the limitations on law enforcement possible with only targeted access. Furthermore, even if society deems this technology worthy of adoption, we stress that this paper should only be viewed as a starting point in a long research path that must be conducted before these ideas are adopted. Future research paths include: refining cryptanalysis techniques so as to make more precise statements about (in)feasibility akin to Assumption 5.1, finding alternate constructions that rely upon different assumptions for robustness, and adding new features for both law enforcement and the public at large such as the ability to verify whether a ciphertext has been crumpled and to audit transparently whether law enforcement's reconstruction powers are used appropriately.

Acknowledgments

We gratefully acknowledge Ran Canetti, Scott Coull, Mike Rosulek, and the anonymous reviewers and shepherds for their valuable advice; any concerns about this work should be directed at us and not at them. This material is based upon work supported by the National Science Foundation under Grant No. 1414119. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] 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.

[2] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, W. W. Diffie, J. Gilmore, M. Green, S. Landau, P. G. Neumann, et al. Keys under doormats. *Communications of the ACM*, 58(10):24–26, 2015.

[3] L. Ablon and T. Bogart. Zero days, thousands of nights: The life and times of zero-day vulnerabilities and their exploits, 2017.

[4] Access Now. Encryption in the U.S.: Crypto colloquium outcomes report, January 2018. https://www.accessnow.org/cms/assets/uploads/2018/02/Encryption-in-the-United-States-Crypto-Colloquium-Outcomes-Report.pdf.

[5] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelin, and P. Zimmermann. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *CCS*, Oct. 2015.

[6] A. Ahmed and N. Perlroth. Using texts as lures, government spyware targets mexican journalists and their families. In *The New York Times*, June 2017. https://www.nytimes.com/2017/06/19/world/americas/mexico-spyware-anticrime.html.

[7] Apple. iOS security–white paper, March 2017. Available at https://www.apple.com/business/docs/iOS_Security_Guide.pdf.

[8] A. Back. Hashcash—A Denial of Service Counter-Measure. Technical report, August 2002.

[9] V. Bajaj. India may soon resolve Blackberry dispute. The New York Times, August 2010. http://www.nytimes.com/2010/08/18/business/global/18rim.html.

[10] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The secure real-time transport protocol (SRTP). Technical Report RFC 3711, IETF, 2004.

[11] M. Bellare and S. Goldwasser. Verifiable partial key escrow. In *CCS*, pages 78–91. ACM, 1997.

[12] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. In *CRYPTO*, pages 398–415, 2013.

[13] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *ASIACRYPT*, pages 531–545, 2000.

[14] M. Bellare, K. G. Paterson, and P. Rogaway. Security of symmetric encryption against mass surveillance. In *CRYPTO*, volume 8616 of *LNCS*, pages 1–19. Springer, 2014.

[15] S. M. Bellovin, M. Blaze, S. Clark, and S. Landau. Going bright: Wiretapping without weakening communications infrastructure. *IEEE Security & Privacy*, 11(1):62–72, 2013.

[16] D. J. Bernstein. Boring crypto, October 2015. Invited talk at SPACE 2015. https://cr.yp.to/talks/2015.10.05/slides-djb-20151005-a4.pdf.

[17] D. J. Bernstein and T. Lange. Non-uniform cracks in the concrete: The power of free precomputation. In *ASIACRYPT*, pages 321–340, 2013.

[18] A. Bestavros, A. Lapets, and M. Varia. User-centric distributed solutions for privacy-preserving analytics. *Commun. ACM*, 60(2):37–39, 2017.

[19] Bitcoin. Mining hardware comparison. https://en.bitcoin.it/wiki/Mining_hardware_comparison. Accessed 19 May 2017.

[20] R. Botsman. Big data meets big brother as china moves to rate its citizens, October 2017. http://www.wired.co.uk/article/chinese-government-social-credit-score-privacy-invasion.

[21] V. Boyko. On the security properties of OAEP as an all-or-nothing transform. In *CRYPTO*, pages 503–518. Springer, 1999.

[22] R. P. Brent and H. T. Kung. The area-time complexity of binary multiplication. *J. ACM*, 28(3):521–534, 1981.

[23] R. Bresciani and A. Butterfield. A formal security proof for the ZRTP protocol. In *2009 International Conference for Internet Technology and Secured Transactions, (ICITST)*, pages 1–6, Nov 2009.

[24] R. Canetti and H. Krawczyk. Universally composable notions of key exchange and secure channels. In *EUROCRYPT*, pages 337–351, 2002.

[25] D. Chaum. Privategrity: online communication with strong privacy. In *Real World Cryptography*, 2016.

[26] D. Chaum, D. Das, F. Javani, A. Kate, A. Krasnova, J. de Ruiter, and A. T. Sherman. cMix: Mixing with minimal real-time asymmetric cryptographic operations. In *ACNS*, pages 557–578, 2017.

[27] K. Cohn-Gordon, C. J. F. Cremers, B. Dowling, L. Garratt, and D. Stebila. A formal security analysis of the signal messaging protocol. In *Euro S&P*, pages 451–466, 2017.

[28] J. B. Comey. FBI budget request for fiscal year 2017, February 2016. Statement Before the House Appropriations Committee, Subcommittee on Commerce, Justice, Science, and Related Agencies. https://www.justice.gov/jmd/file/822286/download.

[29] T. Cook. A message to our customers, February 2016. https://www.apple.com/customer-letter/.

[30] J. Corbet. Ext4 encryption, April 2015. https://lwn.net/Articles/639427/.

[31] V. Costan, I. A. Lebedev, and S. Devadas. Secure processors part I: background, taxonomy for secure enclaves and Intel SGX architecture. *Foundations and Trends in Electronic Design Automation*, 11(1-2):1–248, 2017.

[32] V. Costan, I. A. Lebedev, and S. Devadas. Secure processors part II: Intel SGX security analysis and MIT Sanctum architecture. *Foundations and Trends in Electronic Design Automation*, 11(3):249–361, 2017.

[33] S. E. Coull and K. P. Dyer. Traffic analysis of encrypted messaging services: Apple iMessage and beyond. *ACM SIGCOMM Computer Communication Review*, 44(5):5–11, 2014.

[34] N. T. Courtois, M. Grajek, and R. Naik. Optimizing SHA256 in Bitcoin mining. In *International Conference on Cryptography and Security Systems*, pages 131–144. Springer, 2014.

[35] cpuboss.com. Intel core i7 3770k vs 6800k. http://cpuboss.com/cpus/Intel-Core-i7-3770K-vs-Intel-6800K Accessed 19 May 2017.

[36] D. Dachman-Soled, R. Gennaro, H. Krawczyk, and T. Malkin. Computational extractors and pseudorandomness. In *TCC*, pages 383–403, 2012.

[37] J. Daemen. *Cipher and hash function design strategies based on linear and differential cryptanalysis*. PhD thesis, Doctoral Dissertation, March 1995, KU Leuven, 1995.

[38] J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[39] K. W. Dam and H. S. Lin. *Cryptography's Role in Securing the Information Society*. National Academy Press, 1996.

[40] G. Danezis and S. Gurses. A critical review of 10 years of privacy technology. In *Proceedings of Surveillance Cultures: A Global Surveillance Society?*, 2010.

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

[42] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference*, pages 139–147. Springer, 1992.

[43] Experian. Young adults: Texting is just as meaningful as a phone call. Technical report, December 2012.

[44] Facebook. Government requests report, 2016. https://govtrequests.facebook.com/country/United%20States/2016-H1/.

[45] J. Fried, P. Gaudry, N. Heninger, and E. Thomé. A kilobit hidden SNFS discrete logarithm computation. In *EUROCRYPT*, pages 202–231, 2017.

[46] 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.

[47] J. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *FOCS*, pages 648–657, Oct 2013.

[48] C. Garman, M. Green, G. Kaptchuk, I. Miers, and M. Rushanan. Dancing on the lip of the volcano: Chosen ciphertext attacks on apple iMessage. In *USENIX Security Symposium*, 2016.

[49] Google. Google transparency report, 2016. https://www.google.com/transparencyreport/userdatarequests/countries/?p=2016-06.

[50] Google. Form 10-K, for the fiscal year ended December 31, 2016, 2017. https://abc.xyz/investor/pdf/2016_google_annual_report.pdf.

[51] T. Hanke. AsicBoost - A speedup for bitcoin mining. *CoRR*, abs/1604.00575, 2016.

[52] M. E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Trans. Information Theory*, 26(4):401–406, 1980.

[53] T. Herr, B. Schneier, and C. Morris. Taking stock: Estimating vulnerability rediscovery. In *Belfer Cyber Security Project White Paper Series*, March 2017.

[54] M. P. Hoyle and C. J. Mitchell. On solutions to the key escrow problem. In *State of the Art in Applied Cryptography, Course on Computer Security and Industrial Cryptography*, volume 1528 of *LNCS*, pages 277–306. Springer, 1997.

[55] N. Koblitz and A. Menezes. Another look at non-uniformity. *Groups Complexity Cryptology*, 5(2):117–139, 2013.

[56] A. N. Kolmogorov. On tables of random numbers (reprinted from "Sankhya: The indian journal of statistics", series a, vol. 25 part 4, 1963). *Theor. Comput. Sci.*, 207(2):387–395, 1998.

[57] S. Landau. Protecting the republic: Securing communications is more important than ever, 2016. https://www.lawfareblog.com/protecting-republic-securing-communications-more-important-ever.

[58] S. Landau. Punching the wrong bag: The deputy AG enters the crypto wars, October 2017. https://www.lawfareblog.com/punching-wrong-bag-deputy-ag-enters-crypto-wars.

[59] D. Lazar and N. Zeldovich. Alpenhorn: Bootstrapping secure communication without leaking metadata. In *OSDI*, pages 571–586, 2016.

[60] A. K. Lenstra. Key lengths. In *The Handbook of Information Security*, 2004.

[61] A. K. Lenstra, H. W. L. Jr., M. S. Manasse, and J. M. Pollard. The number field sieve. In *STOC*, pages 564–572, 1990.

[62] A. K. Lenstra and E. R. Verheul. Selecting cryptographic key sizes. In *Public Key Cryptography*, volume 1751, pages 446–465. Springer, 2000.

[63] 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.

[64] M. Liskov and K. Minematsu. Comments on XTS-AES, 2008.

[65] M. Loukides and J. Gilmore. Cracking DES: secrets of encryption research, wiretap politics and chip design, 1998.

[66] M. Mahmoody, T. Moran, and S. P. Vadhan. Time-lock puzzles in the random oracle model. In *CRYPTO*, pages 39–50, 2011.

[67] M. Marlinspike. Video calls for Signal now in public beta, February 2017. https://whispersystems.org/blog/signal-video-calls-beta/.

[68] R. Mason. UK spy agencies need more powers, says Cameron, January 2015.

[69] M. McConnell, M. Chertoff, and W. Lynn. Why the fear over ubiquitous data encryption is overblown, July 2015. http://wapo.st/1LQMTtk.

[70] R. C. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, Apr. 1978.

[71] Microsoft. Law enforcement requests report, 2016. https://www.microsoft.com/en-us/about/corporate-responsibility/lerr/.

[72] K. Murphy. In online dating, sextortion and scams, January 2016. https://www.nytimes.com/2016/01/17/sunday-review/in-online-dating-sextortion-and-scams.html.

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

[74] National Academies of Sciences, Engineering, and Medicine. *Cryptographic Agility and Interoperability: Proceedings of a Workshop*. The National Academies Press, Washington, DC, 2017.

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

[76] T. Perrin and M. Marlinspike. The double ratchet algorithm. Technical report, Open Whisper Systems, November 2016.

[77] R. Pfefferkorn. The risks of “responsible encryption”, 2018. https://cyberlaw.stanford.edu/files/publication/files/2018-02-05%20Technical%20Response%20to%20Rosenstein-Wray%20FINAL.pdf.

[78] Quartz. How much money did you make for Facebook last year?, 2016. https://qz.com/605343/how-much-money-did-you-make-for-facebook-last-year/.

[79] M. M. L. Rainie. Americans attitudes about privacy, security and surveillance, May 2015. http://www.pewinternet.org/2015/05/20/americans-attitudes-about-privacy-security-and-surveillance/.

[80] B. Ramsdell and S. Turner. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. Technical Report RFC 5751, IETF, January 2010.

[81] E. Rescorla. WebRTC security architecture. Technical report, IETF, June 2016. https://www.ietf.org/archive/id/draft-ietf-rtcweb-security-arch-12.txt.

[82] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Massachusetts Institute of Technology, 1996.

[83] P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In *ASIACRYPT*, pages 16–31. Springer, 2004.

[84] P. Rogaway. Formalizing human ignorance. In *VIETCRYPT*, pages 211–228, 2006.

[85] P. Rogaway. The moral character of cryptographic work. *IACR Cryptology ePrint Archive*, 2015:1162, 2015.

[86] P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In *EUROCRYPT*, pages 373–390, 2006.

[87] J. F. Rogers. Security trumps privacy in British attitudes to cyber-surveillance, 2017. https://rusi.org/commentary/security-trumps-privacy-british-attitudes-cyber-surveillance.

[88] B. Schneier, M. Fredrikson, T. Kohno, and T. Ristenpart. Surreptitiously weakening cryptographic systems. *IACR Cryptology ePrint Archive*, 2015:97, 2015.

[89] A. Shamir. Partial Key Escrow: A New Approach to Software Key Escrow. In *Key Escrow Conference*, September 1995.

[90] A. Smith. How Americans use text messaging. Technical report, September 2011.

[91] M. Smith and M. Green. A discussion of surveillance backdoors: Effectiveness, collateral damage and ethics, February 2016. http://mattsmith.de/pdfs/SurveillanceAndCollateralDamage.pdf.

[92] M. Specter. On deniability and duress, 2017. http://www.mit.edu/~specter/articles/17/deniability1.html, Accessed 25 January 2018.

[93] C. R. Sunstein. Beyond cheneyism and snowdenism. *The University of Chicago Law Review*, pages 271–293, 2016.

[94] M. Tait. An approach to James Comey’s technical challenge, 2016. https://lawfareblog.com/approach-james-comeys-technical-challenge.

[95] M. Tait. Decrypting the going dark debate, 2017. https://lawfareblog.com/decrypting-going-dark-debate.

[96] The United States Department of Justice. Deputy Attorney General Rod J. Rosenstein Delivers Remarks on Encryption at the United States Naval Academy, October 2017. https://www.justice.gov/opa/speech/deputy-attorney-general-rod-j-rosenstein-delivers-remarks-encryption-united-states-naval.

[97] E. Tromer. *Hardware-Based Cryptanalysis*. PhD thesis, Weizmann Institute of Science, May 2007.

[98] Z. Tufekci. The latest data privacy debacle, January 2018. https://www.nytimes.com/2018/01/30/opinion/strava-privacy.html.

[99] Twitter. Transparency report: Information requests, 2016. https://transparency.twitter.com/en/information-requests.html.

[100] 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.

[101] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith. SoK: Secure messaging. In *IEEE S&P*, pages 232–249, May 2015.

[102] US Energy Information Administration. Electric power monthly, November 2016. https://www.eia.gov/electricity/monthly/.

[103] 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.

[104] World Economic Forum. Which are the world’s strongest democracies?, February 2017. https://www.weforum.org/agenda/2017/02/which-are-the-worlds-strongest-democracies.

[105] C. Wray. Raising our game: Cyber security in an age of digital transformation, 2018. https://www.fbi.gov/news/speeches/raising-our-game-cyber-security-in-an-age-of-digital-transformation.

[106] Yahoo. Transparency report: Government data requests, 2016. https://transparency.yahoo.com/government-data-requests?tid=34.

[107] A. L. Young and M. Yung. The dark side of ”black-box” cryptography, or: Should we trust capstone? In *CRYPTO*, volume 1109 of *LNCS*, pages 89–103. Springer, 1996.

[108] P. R. Zimmermann. *The official PGP user’s guide*. MIT press, 1995.