Protecting Cryptography Against Compelled Self-Incrimination

Sarah Scheffler
Boston University

Mayank Varia
Boston University

Abstract

The information security community has devoted substantial effort to the design, development, and universal deployment of strong encryption schemes that withstand search and seizure by computationally-powerful nation-state adversaries. In response, governments are increasingly turning to a different tactic: issuing subpoenas that compel people to decrypt devices themselves, under the penalty of contempt of court if they do not comply. Compelled decryption subpoenas sidestep questions around government search powers that have dominated the Crypto Wars and instead touch upon a different (and still unsettled) area of the law: how encryption relates to a person's right to silence and against self-incrimination.

In this work, we provide a rigorous, composable definition of a critical piece of the law that determines whether cryptosystems are vulnerable to government compelled disclosure in the United States. We justify our definition by showing that it is consistent with prior court cases. We prove that decryption is often *not* compellable by the government under our definition. Conversely, we show that many techniques that bolster security overall can leave one more vulnerable to compelled disclosure.

As a result, we initiate the study of protecting cryptographic protocols against the threat of future compelled disclosure. We find that secure multi-party computation is particularly vulnerable to this threat, and we design and implement new schemes that are provably resilient in the face of government compelled disclosure. We believe this work should influence the design of future cryptographic primitives and contribute toward the legal debates over the constitutionality of compelled decryption.

1 Introduction

Two fundamental human rights in free and democratic societies are the right to remain silent and the right to avoid self-incrimination. More than 100 countries around the world have enshrined some version of these rights [67], which collectively protect people from being forced by their own governments to provide the evidence needed to convict themselves of a crime. In the United States, these rights stem from the Fifth Amendment to the U.S. Constitution, which states in part that "[n]o person. . . shall be compelled [by the government] in any criminal case to be a witness against himself" [75].

The rise of ubiquitous, strong cryptography has forced courts to consider how all aspects of the law apply to cryptography, including the right to silence. To date, the most prominent question surrounding cryptography and the right to silence is the following: if the government seeks as evidence a computer file that is encrypted using a key derived from a password, *can the government compel the device's owner to use her password in order to decrypt the file*?

We stress that this question about compelled assistance is different than the more prominent part of the Crypto Wars, in which governments wish to mandate use of cryptosystems that they can decrypt on their own. That question touches upon very different areas of the law, such as whether encryption provides a reasonable expectation of privacy and whether free speech extends to the right to develop encryption software [6]. In fact, we will show that it is possible to design encryption schemes that preclude governments from decrypting files on their own, but are nevertheless vulnerable to the government compelling you to decrypt files for them.

Taken at face value, it seems that the answer to the compelled decryption query should be "no": the device's owner can invoke her rights in order to refuse the government's request. However, the answer to this question is more subtle because the rights to silence and to avoid self-incrimination are not absolute: they only protect actions that depend non-trivially on the contents of one's mind. For instance, the U.S. Supreme Court has held that the government can compel people to state their own name [28], provide a handwriting exemplar [25], or provide a blood sample [53] despite the right to avoid self-incrimination.

The question then arises: how significantly does decryption depend on the contents of one's mind? Both the court system and scholars with expertise in law and technology have divided on this question, and they all provide different non-technical arguments about how to extend existing norms and principles surrounding the right to silence so that they apply to cryptography. In this work, we provide a technical framework for the relevant legal doctrine, which we then use to reason that the answer to the compelled decryption question should often be "no."

1.1 Our contributions

Rather than simply viewing cryptography as a technology that introduces new legal questions, in this work we leverage the ideas of cryptography to codify legal principles and then formally prove whether they apply to any given cryptosystem. Concretely, this work examines a small yet crucial part of the right to silence called the *foregone conclusion doctrine* that is the source of all government cases involving compelled decryption in the United States (we will describe it in detail in §2). We formalize this doctrine under a cryptographic lens, providing a rigorous definition and formally proving whether constructions are susceptible to it. Specifically, we make the following three types of contributions.

Rigorous definition. We form a simulation-based cryptographic definition that covers the foregone conclusion doctrine. At a high level, the goal of our definition is intuitive: the government can only compel a query if it can be answered without relying heavily on the contents of your mind, and that is the case if the government can simulate the response to the query based upon its prior evidence about the case and access to everything in the world *except* the contents of your mind. We also prove that the definition satisfies sequential composition, which means that compelling one action cannot change the status about whether any other action is compellable.

To justify our definition, we demonstrate that it correctly adjudicates all non-encryption-related cases argued in the U.S. Supreme Court since the modern interpretation of the Fifth Amendment arose in 1976 [22] and the five most important cases at the circuit court level (i.e., the next level of the court hierarchy) as identified by legal scholars [17, 39, 41, 85]. We purposely ignore cases involving encryption since the Supreme Court has never ruled on them and lower courts have split on them, leaving no reliable benchmark to use.

Determining if crypto can be compelled. We reason about the government's ability to compel disclosure of cryptographic secrets. To answer the question raised above: we prove that under our definition, decryption under a password-derived key is typically *not* compellable. However, if the encryption scheme is extended with certain features (including those that are often used to bolster security overall) then it may become compellable. Additionally, we show that compelled disclosure composes with the Crypto Wars in a debilitating way: if there exists a reliable method for the government to decrypt data without you, then the government can compel you to perform the decryption instead.

We also consider the government's ability to compel a person to reveal preimages to one-way functions, open messages protected within cryptographic commitments, and prove statements in zero knowledge. While we are unaware of any court challenges to date that compel use of these cryptographic primitives, they may come some day, and our definition enables us to be forward-looking to determine whether cryptosystems can withstand these threats.

Bolstering cryptosystems against compelled disclosure. We consider how voluntary use of cryptographic systems exposes parties to higher risk of compelled actions in the future. We find that secure multi-party computation (MPC) is vulnerable to this threat: engaging in MPC protocols may increase the compellability of a party's sensitive input data. Then, we design and implement countermeasures that provably render secure computation protocols resilient to compelled requests. Our countermeasures apply to 2-party computation via Yao's garbled circuits [88], with extensions to malicious security via cut-and-choose [44,48] or authenticated garbling [83]. We implement the latter and show that it adds a small additive factor to the runtime that is independent of the circuit size. We also show how to extend the construction to a multi-party protocol where several parties receive output while maintaining resilience against compelled requests, by incorporating techniques from differential privacy.

1.2 Remarks

We hope this work provides worthwhile designs of cryptosystems that withstand government compelled requests, inspires the community to include this threat when designing secure systems, and casts new light on the value of passwords as a useful protection against this threat. That having been said, we make several remarks to clarify the context of this work.

First, it is difficult to judge the accuracy of any legal definition in a common law system. We show the best possible evidence: that our definition is consistent with established precedents by the Supreme Court and the appellate courts. Nevertheless, subsequent decisions by the courts might strengthen or restrict government power in a way that renders our definition moot. Even if this should happen, we believe that our paper provides enduring value by showing a methodology to reverse-engineer a formal definition from common law.

Second, this work only captures a subset of legal cases, albeit a subset that we believe is useful. We presume that the government tells the truth when interacting with the court system, although we do not presume that the government tells the *whole* truth. This work only considers the right to silence as interpreted in the United States. Additionally, this work considers self-composition of compelled requests (§3.3) and reasons how compelled requests compose with the government's own decryption capabilities (§5.5), but we do not consider how the Fifth Amendment itself composes with other aspects of the law. We acknowledge that this gap can introduce two-sided error: actions that are permissible under the Fifth Amendment might be refuted on other grounds, and conversely, information protected under the right to silence might be accessible to the government via other means.

Third, we stress that this work only focuses on security against one specific threat: that of compelled action by the government. Therefore, the threat model in this work is necessarily incomplete and potentially counterproductive if protections against government compelled requests conflict with protections for other threats. For this reason, we prove the constructions in this work secure under their traditional definitions in addition to analyzing their resilience to foregone conclusion requests (§6). We hope that this work inspires the information security community to consider government compelled actions within scope in their threat models.

1.3 Related work

This work is the first to postulate a mathematically rigorous definition for the foregone conclusion doctrine, a crucial part of the right to silence in the United States. We are inspired by and build upon several prior efforts that (separately) formalize aspects of the law or reason about compelled decryption under the foregone conclusion doctrine.

Cryptographic modeling of the law. This work is inspired by recent endeavors to use cryptography to model and address other aspects of the law. Frankle et al. [23], building upon earlier work by Goldwasser and Park [27], propose the use of zero-knowledge proofs to provide public auditing of warrants issued secretly by intelligence courts. Nissim et al. [49] provide a formalization of privacy that they argue legally satisfies (part of) the privacy law in the United States governing education data. Cohen and Nissim [11] formalize one aspect of European privacy law about de-identifying personal data, and they prove that differential privacy achieves this notion but k-anonymity does not. Garg et al. [24] provide a simulation-based definition of the right to be forgotten.

Crypto Wars. Several prior works consider using cryptography to enable governments to execute search warrants where encryption is involved. Smith et al. [58] and Feigenbaum et al. [21] discuss broad principles for this topic. Specific encryption schemes with key escrow have been proposed since the 1990s [19], and more recent proposals combine cryptography with trusted hardware so that a device manufacturer can assist law enforcement in decryption [8, 52, 62]. There also exist MPC-based constructions that provide more fine-grained functionalities like auditable threshold decryption [10, 42] and private set intersection across private companies [55]. Bellovin et al. [4, 5] sidestep cryptography altogether and look to lawful hacking as a resolution to the Crypto Wars.

Conversely, several prior works use cryptography to limit government overreach technologically. Tyagi et al. [69] provide "self-revocable" encryption in which a user can temporarily revoke her own ability to access her secret data for the purpose of defending against temporary compelled decryption threats such as border crossings. Traffic unlinkability tools like Tor [61] protect against traffic analysis by governments, and encrypted search techniques can be used to limit collection of metadata stored at rest [37, 87]. There are works that protect against subversion by the government (or anyone else) for encryption schemes [29], digital signatures [1], and hash functions [2]. None of these works consider compelling the respondent to perform the decryption in a court setting, and as we show in §5.5 security against that threat is reliant upon the government's inability to get the data some other way.

Legal analyses of compelled decryption. To our knowledge, Cohen and Park [12] is the only legal analysis of the foregone conclusion doctrine by authors with cryptographic expertise; their expository work describes several legal concepts and how they fare against technological advances such as widespread use of deniable encryption or hardware kill switches. Additionally, there exist several normative works by law scholars whose reasonings (which often analogize encryption to other security mechanisms like safes or shredders) lead to very different conclusions. Winkler [85] argues that the right against self-incrimination prevents the government from compelling a respondent to use her passwords in any way. Kerr [39], McGregor [46], and Terzian [65] make distinct yet related arguments that the government should have the power to compel decryption in order to restore balance between government powers and civil liberties in light of modern encryption's strong confidentiality guarantees. Kiok [41] and Sacharoff [51] settle somewhere in the middle, only allowing the government to compel decryption if they already know certain aspects of the targeted files with "reasonable particularity." Unlike all of these works, our definition is rigorous, composable, applies directly to encryption rather than using an analogy, and is easier for the scientific community to analyze when evaluating the security of a new system.

Existing case law. Analyses of compelled decryption are timely because courts in the United States are currently divided on the issue. Some courts say people can be compelled to disclose passwords themselves (e.g. [59]), some say they can be compelled only to enter the password but not reveal it (e.g. [15, 16]), and others say that only specific files already known to the government can be compelled (e.g. [56]). See Appendix §4.1 for a summary of these rulings. Leading legal scholars believe that the U.S. Supreme Court is likely to take a compelled decryption case case soon and resolve this confusion [40], making it important for the computer science community to understand the legal nuance of this topic in order to contribute to this discussion.

1.4 Organization

In §2, we describe the body of law known as the foregone conclusion doctrine, which is the deciding factor in most decryption cases. In §3, we provide our formal definition of a foregone conclusion and analyze its properties. In §4, we compare our approach to legal scholarship and justify our definition by showing that it comes to the same outcome as U.S. Supreme Court and circuit court decisions. In §5, we analyze the compellability of common cryptographic primitives under the foregone conclusion doctrine.

While our paper is written for an audience of computer security researchers, readers with more legal background may be interested in §4.1, where we put our work into context within the legal literature. In

§6, we explore the extent to which voluntarily participating in a cryptographic protocol leaves one more vulnerable to future compelled requests; we call a protocol *resilient* if any compellable action after running the protocol was already compellable before running the protocol. We conclude in §7, and we defer some modeling and proof details to the appendices.

2 Overview of Foregone Conclusion Law

In this section, we provide a brief overview of the Fifth Amendment to the United States Constitution (abbreviated "5A"). We emphasize one aspect of 5A law called the foregone conclusion (FC) doctrine, which is the crux of all compelled decryption cases.

The right to silence as an interactive protocol. The right to silence in the United States involves three parties: a government actor G such as a prosecutor or law enforcement officer, an individual *respondent* R of the compelled request, and a neutral court. We use the term respondent rather than "suspect" or "defendant" because people can be compelled to perform government actions even without being accused of a crime, and we consider individuals because companies do not have Fifth Amendment rights. Also, we stress that this work focuses on G's compelled requests to R, not G's powers or restrictions to search for information on its own.

Compelled requests follow a 3-round interactive protocol: first G issues a subpoena asking R to respond to a query, then R responds by asserting her right to silence, and finally G requests that a court compels R to answer the query anyway. For the court to approve the government's request to "override" the respondent's right to silence, the burden of proof falls on the government to demonstrate that the compelled request is not covered under the respondent's rights [30].

The Fifth Amendment's protections are broad but not absolute: they only apply to government requests that are compelled, incriminating, and testimonial [22]. We describe the first two properties by way of contradiction: 5A cannot retroactively protect statements that R has previously provided voluntarily to the government, and it cannot be invoked if the government has granted R immunity from prosecution [70]. Because these two criteria are usually simple to verify, throughout this work we assume that all parties agree that G's request is compelled and potentially incriminating.

Testimony. We focus in this work on the final requirement: the government is only restricted from compelling people to perform acts that are *testimonial*, meaning that they "disclose the contents of [the respondent's] own mind" [18]. Based on this principle, speaking your password to the government is testimonial [73], but providing a blood sample [53] does not rely on any mental state of R, so it is non-testimonial and therefore compellable under 5A.

The law protects *direct testimony* in which the written or spoken output of a compelled request directly reveals information about R's mind, and *indirect testimony* in which the government can infer something within R's mind by "relying on the truthtelling" [22] of the respondent when performing an action C and producing the result. In implicit testimony, the *act of production* is the testimonial object in question, not the *contents* produced. For instance, G cannot compel R to provide written documents (whose contents are *not* 5A-protected) if G must rely upon "the respondent's truthful reply [to receive] the incriminating documents" [72]. If R provides the documents upon request, then R's act of producing them testifies to (at least) the existence of the documents, as well as R's possession of them and her belief that they are authentic [22]. Direct testimony is always forbidden within compelled requests, although implicit testimony need not be; for this reason, we focus on implicit testimony in this work.

We emphasize that only the testimonial aspects of a compelled request C are covered under 5A. The output of C might reveal more or less information than the indirect testimony implied by it, but only the latter is protected. For example, suppose that G compels R to provide all documents sitting in plain sight within her locked office. Whether the documents themselves are incriminating is irrelevant; R only has the right to withhold from G the implicit testimony revealed by executing C, i.e., the knowledge in R's mind implied by her truthful response. In this example, the only implicit testimony from the compelled action is that R has the ability to access her own office. There is no ambiguity as to the choice of documents themselves, and thus no testimonial aspect – the government *could* have sent someone to break into her office and collect the documents themselves, without relying on R. So the only testimonial aspect of this compelled request is R's ability to access her office. If G already has evidence that R knows the location of her office key, then executing C would not reveal any *new* implicit testimony to G. This begs the question: does it violate R's rights for G to compel R to implicitly testify to a statement that G already knows to be true?

The foregone conclusion doctrine. The U.S. Supreme Court case *Fisher v. United States* answers the above question in the negative, thereby providing a power to the government that can counter R's invocation of the right to silence. The *Fisher* case says that the courts can compel R to execute an action C if its implicit testimony is a *foregone conclusion* to the government, in the sense that it "adds little or nothing to the sum total of the Government's information" [22]. Concretely, the law enumerates several blacklisted predicates: if G would learn about the existence, location, or authenticity of any new evidence from its interaction with the respondent, then the compelled action is *not* a foregone conclusion.

This work starts from the premise that simulation-based cryptographic definitions can dovetail with the concepts within the foregone conclusion doctrine for 3 reasons. First, simulatability formalizes the concept of "not learning new evidence" [26,43]. Second, simulation sidesteps entirely the task of enumerating sources of implicit testimony; instead, it holistically determines whether all implicit testimony present in a compelled action C is a foregone conclusion. Third, whereas predicate blacklist-based definitions often allow a series of individual requests that might be deemed to be invasive in totality, we will demonstrate security under composition.

3 Rigorously Defining Foregone Conclusions

The crux of the foregone conclusion question is how to know when the government is "relying" on the contents of the respondent's mind, when compelling her to perform an action? This work uses the cryptographic concept of *simulation* to codify the idea that running a foregone compelled action "reveals nothing" to the government about the respondent's mind, above and beyond what the government can learn from the rest of the world.

In this section, we provide both informal and then rigorous descriptions of our security game that encapsulates the foregone conclusion doctrine. Additionally, we prove that our definition remains secure under sequential composition.

3.1 Informal walkthrough

In this section, we provide an informal description of our game-based definition of the foregone conclusion doctrine. Our game proceeds interactively between the government and respondent to determine whether an action is (or is not) a foregone conclusion. We abstractly represent all of the information in the rest of the world (outside of the respondent's mind) as "Nature." We also assume as a pre-condition that the government and the respondent have already agreed on the evidence E of the case.

The government acts first in our game. It declares a compelled action C that it wants the respondent to perform; this action may make use of both the respondent's mind and Nature. (For interactive protocols, the government must also output a second machine G codifying the government's response to each message from C.) We stress that C represents the *act of production*, i.e., the process of obtaining the result rather than the result itself. The government has the burden of proof to demonstrate that its compelled request is a foregone conclusion, as required by the courts [39]. The government submits this proof in the form of a simulator S that tries to output the same result as C without access to the respondent's mind but with the significant power to view anything else in the world.

Second, the respondent has an opportunity to demonstrate that the compelled action depends non-trivially on her own mind. To do this, she must equivocate: specify a "world," comprising Nature N and the contents of her own mind R, where the simulation disagrees to match the compelled action with non-negligible probability. This world must be consistent with the evidence, or else the respondent loses our game.

Third, we run a thought experiment to test whether the government's uncertainty about the state of the world is too high for the compelled action to be deemed a foregone conclusion. Concretely, we run the compelled action and the simulation, and we ask a distinguisher to attempt to tell the two results apart. If the results are indistinguishable, then we declare that any implicit testimony in the compelled action is a foregone conclusion on top of the existing knowledge already available to the simulator (i.e., everything in the rest of the world). If the results are different, then we declare that the government is relying too much on the truthtelling of the respondent for the compelled action to be deemed a foregone conclusion. The government could try again with a different compelled action, an improved simulation strategy, or more evidence; any of these options would cause the game to begin anew.

3.2 Formal definition

In this section, we formally define a foregone conclusion. We emphasize that the evidence should be sufficient so that a *single* government simulator S can simulate the response of *any* respondent R that acts consistently with the evidence; that is, the choice of S cannot depend on the contents of any specific respondent's mind.

Participants. We describe below several components in our model: a string representing nature, and several probabilistic polynomial time (PPT) interactive Turing machines (ITMs) in the manner formalized by Canetti [9] (see Appendix A for a complete formalism). These machines are all poly-time in a security parameter λ that we define below.

- Nature N represents the entire world except the contents of the respondent's mind. It is a string that is exponentially long in λ.
- Respondent R represents the contents of the respondent's mind. It is called by the compelled action C via methods. (For example, the evidence might enforce the existence of method $R.pw$, and the output of this method will be used in the execution of the compelled action.) R also has a special method called $R.\texttt{Equivocate}$ that can make changes to Nature at the beginning of the security game.
- Evidence $E^N(R)$ verifies that the respondent R and the altered nature N are consistent with the government's knowledge about the world before the compelled action. It may query Nature and inspect the code of R (for example, if it is known that R can access her office, E might check that method $R.access$ correctly accesses R's office within N).
- Compelled request $C^{N,R}$ is the computation specified in the government's subpoena. C has oracle access to both nature and the respondent, and it might interact with the government G^N. We denote the resulting transcript as $\tau(G^N, C^{N,R})$.
- Simulator S^N attempts to reconstruct a transcript τ' that is indistinguishable from the real interaction. It can access all of nature N, but it cannot access the respondent R.
- A distinguisher $\mathcal{D}^N$ receives either the "real" execution $\tau(G^N, C^{N,R})$ or the "ideal" execution S^N, and attempts to distinguish between the two. If no $\mathcal{D}$ can distinguish between these, the result is a foregone conclusion.

In our game, the security parameter λ should be thought of as the number of queries the evidence E makes to N, with some constant lower bound to avoid pathologies (e.g. $\lambda = \max\{80, \#\text{queries}\}$); all other machines must operate in time polynomial in this. Bounding the runtime of these machines is consistent with the legal doctrine, which holds that "location/possession" is one of the three prongs of the foregone conclusion test. Without the location prong, the legal analysis would seem to allow compelling documents whose existence is known and that can be authenticated, but that could be literally anywhere in the world (i.e., S's runtime

```
$\mathsf{Game}_{E,C,G,S,R}(\lambda)$
$N \leftarrow_{\$} \Sigma^{2^\lambda}$   // initialize N randomly
$\Delta \leftarrow R.$Equivocate   // $\Delta$ is a set of index-value pairs
// $\Delta$ is a set of changes to $N$
for $(i, x)$ in $\Delta : N[i] = x$
// check evidence and return $\perp$ if false
if $E^N(R) =$ false : return $\perp$
// return either real or simulated transcript
return $N, \boxed{\tau(G^N, C^{N,R})}, \boxed{S^N}$
```

Figure 1: Real (solid) and ideal (dashed) foregone conclusion games. Steps without a box are common to both games.

would be unbounded). Bounding the execution of S restricts the government from searching the entire world and enables our definition to judge whether the government benefits non-trivially from R's knowledge of the location of information.

Allowed respondents. When checking for a foregone conclusion, the respondent is automatically "caught" if it does something that violates the evidence E. We say that R is *allowed* by the evidence if E (R) returns **true** with overwhelming probability over the initial random initialization of N.

Because all allowed Turing machines *could* represent the real state of the respondent's mind as far as the government is aware, our foregone conclusion definition will require that the government can simulate all allowed R. Conversely, the simulator is only required to succeed on allowed respondents. To avoid degeneracy, the definition will require the existence of at least one allowed respondent.

Security game. We specify the real and ideal versions of our security game in Figure 1. In the real game, the government interacts (possibly over multiple rounds) with the respondent who executes the compelled algorithm C. In the ideal game, the government's simulator S forges a transcript using only its access to Nature (which has previously been prepared by the respondent). The two games are identical except for the final step. In the last step, the $\boxed{\text{real game}}$ returns the transcript $\tau(G^N, C^{N,R})$ of all communications between the government and respondent, whereas the $\boxed{\text{ideal game}}$ returns the simulated transcript S^N. Both games also offer oracle access to N, and both return $\perp$ if the evidence was not satisfied.

Next, we provide our formal definition of the foregone conclusion principle in Def. 3.2.1. It requires that the government's simulator S faithfully emulates real-world transcripts. Moreover, it limits the respondent R's ability to equivocate and the evidence E's ability to censor R's use of nature.

Definition 3.2.1 (Foregone conclusion (FC_λ))**.** Let λ be a security parameter. The exchange between G and C is a *foregone conclusion* with respect to E and S if the following four conditions are met:

1. *Efficiency:* C, G, and S are PPT machines in 1^λ.
2. *Simulatability:* $\forall$ allowed R, $\forall$ ppt $\mathcal{D}$, $\exists$ negligible function negl such that

$$\big|\Pr[\mathcal{D}^N\big(\tau(G^N, C^{N,R})\big) = 1] - \Pr[\mathcal{D}^N\big(S^N\big) = 1]\big| < \mathsf{negl}(1^\lambda)$$

   where N, $\tau(G^N, C^{N,R})$, and S^N are the results of the real and ideal security games defined in Fig. 1.
3. *Satisfiability of evidence:* There exists at least one allowed R. Hence, simulatability cannot be vacuously true.

4. *Non-censorship of evidence:* For any allowed R where $R.\texttt{Equivocate} \rightarrow \Delta$, all R' where $R'.\texttt{Equivocate} \rightarrow \Delta'$ such that $\Delta \subseteq \Delta'$ are also allowed. That is, E does not prevent R from making additional changes to N beyond the locations it checks.

Notice that the probability in the satisfiability requirement is taken over the randomness of R and the random choice of N (modified by R), whereas the probability in the simulatability requirement is taken over R, N, $\mathcal{D}$, C, and S.

Remarks. We make several remarks about this definition. First, the definition puts the burden of proof on the government as is true in the legal regime [39] by requiring that it construct the simulator S rather than merely asserting that one exists, and by requiring that S is chosen before the respondent R chooses its equivocation strategy. Second, the code of R represents the respondent's current actions and limitations in the present (based upon the government's evidence) even if this doesn't correspond to the exact code that the respondent originally executed in the past. Third, because the simulator S can access nature, it doesn't need to forge the *contents* of any documents; rather, it must only forge the *process* of producing them. Fourth, we presume that the government tells the truth about its evidence. Fifth, due to the order of quantifiers and R's equivocation ability, if the compelled action C is deterministic then the simulator must match this action exactly. This is explained further in the proof of Lemma 5.1.1.

3.3 Sequential composition

In this section, we prove that Definition 3.2.1 remains secure under sequential composition. Essentially, our theorem states that the information disclosed by a government compelled action cannot immediately open up *new* actions that the government can subsequently compel. First, some notation: we denote a sequential composition of Turing Machines M_1 and M_2 as the machine $M_1 \| M_2$ that fully runs M_1 and then fully runs M_2; see Appendix A.1 for a formal description.

Theorem 3.3.1 (Sequential composition)**.** *Suppose C_1, G_1 is a foregone conclusion with respect to E and S_1. Then C_2, G_2 is a foregone conclusion with respect to E and S_2 if and only if there exists a simulator S_{12} such that $(C_1 \| C_2), (G_1 \| G_2)$ is a foregone conclusion with respect to E and S_{12}.*

Proof sketch. While composition generally follows naturally in simulation-based definitions, the proof in our setting is somewhat non-standard. For instance, proving composition for zero-knowledge proofs requires an auxiliary input so that later instances store the results of (simulated versions of) earlier instances, but our definition doesn't have a direct concept of auxiliary input. We proceed in the other direction: we proactively store simulated versions of later instances in Nature to test the limits of whether earlier instances are truly foregone conclusions. The full proof can be found in Appendix A.2. □

This theorem demonstrates that our foregone conclusion doctrine satisfies two intuitively-appealing goals. First, if a compelled action C would *not* be a foregone conclusion given the government's existing evidence, then it should not be possible to split C into smaller actions (compelled in sequence) that collectively perform C and that are each individually deemed foregone conclusions. Second, there should not be a way for the government to compel beforehand a different foregone conclusion C' in order to change the status of C into a foregone conclusion. We emphasize that the composition theorem only applies to two government requests made in sequence without changes to Nature or the Evidence in between; it *is* possible that the government could compel an action, use the response to guide its police investigation to gather more evidence, and then compel a second action based on this additional evidence.

4 Equivalence with Existing Legal Precedents

In this section, we justify Def. 3.2.1 by demonstrating its consistency with prior court cases that involve the foregone conclusion doctrine. We begin by comparing our definition with existing legal scholarship in

§4.1; this subsection is purposely written in detail for a law-savvy audience, and may safely be skipped by computer science readers. Then, we apply our definition to all relevant U.S. Supreme Court cases in §4.2 and all circuit court cases in §4.3 that were identified by the legal scholarship, and we demonstrate how our definition reaches the same conclusions as the courts. However, we deliberately defer discussion of encryption-related cases [13–16, 34, 54, 56, 59, 60, 73, 74, 77, 78, 80] to §4.4; we believe these cases are actually less illustrative than their non-encryption counterparts because these rulings are quite varied and subject to being overturned by higher courts.

4.1 Legal scholarship context

This section provides a thorough description about how our approach compares to prior legal scholarship on the foregone conclusion doctrine. It is written with a law audience in mind; computer scientists should feel free to skip ahead to §4.2.

Other legal analyses of compelled decryption [12, 39, 41, 46, 51, 65, 85] rely upon analogies between encryption and physical security mechanisms like safes or shredders. Kerr recently stated "whether [the Fifth Amendment] privilege bars compelled entry of the password. . . depends on a choice of analogy" [40]. These analogies are further muddled by ambiguous language in court cases: In a now-infamous dissent, Justice Stevens said that he "do[es] not believe [a defendant] can be compelled to reveal the combination to his wall safe – by word or deed" [20]. Does "reveal in deed" mean to be forced to enter the combination without the government seeing it? We assume so, but this is not the only interpretation.

We wrote this paper to move the compelled decryption debate beyond the choice of analogy. We recognize the prevalence and value of analogies in the development of common law, but because their use leads to such differing results in this case, we believe this situation warrants rejecting analogies. Under our model, we can reason directly about the principle that for a compelled action to be a foregone conclusion, it should not "rely on the contents of the mind." This also suggests a change to the three-prong test of existence, location/possession, and authenticity for determining whether an action is a foregone conclusion. Rather than reasoning only about these (which happened to be the implicit testimony in *Fisher*, as we will show in §4.2) we can reason in a thought experiment about the government's ability to recreate the act of production without using the contents of the respondent's mind.

Because we can avoid the use of analogies, our reasoning is different than all prior work. The closest legal landmark to our model is Sacharoff's authentication-based interpretation of "reasonable particularity" [51], but there are some important differences between the two approaches. Sacharoff's envisioned test, like our method, is based on the idea that information entered into evidence from non-respondent sources can be used to demonstrate a non-reliance on the contents of the respondent's mind. Indeed, one could argue that the simulator in our scheme must produce "reasonably similar" output to that of the true compelled action. However, the methods are not the same. First, addressing an issue brought up by Kerr [39], our method applies to any compelled action even if there are no produced documents at the end that could be described with "reasonable particularity." Second, and more importantly, our method highlights the fact that the action taken, not the objects produced, contains the implicit testimony. For better or worse, the reasonable particularity method makes it harder to distinguish between the "door-opening" and the "treasure," as Kerr would put it [39]. Our model makes it clear that the government must not learn the new implicit testimony involved in the process of complying with the request (as opposed to the results).

Our interpretation is very different from other prior work. As mentioned, Kerr [39] distinguishes between "door-opening" and "treasure." This analogy, reasonably, tries to separate the act of production from the contents produced. In the same paper, Kerr proceeds to claim that "'I know the password' is the only assertion implicit in unlocking the device" [39, p. 779] We disagree; we described the "reliance" on the respondent's mind in §5.5. Our objection is solely in the compelled action, not the contents revealed.

Kiok [41] bemoans the fact that the cryptography analogies have, thus far, "missed the metaphor." McGregor [46] also notes that the choice of analogy greatly impacts the outcome, and proposes the analogy of piecing together shredded papers without knowing which order they go in. This analogy is an improvement over the safe/combination dichotomy, but we believe our approach avoids the issue entirely.

The papers...	$\exists k, \mathsf{p}$ such that:
are in the possession of the taxpayer [22, line 409]	$k \in \mathsf{locations}$ (where $\mathsf{locations}$ is a small set of indices in Δ)
	also implies $\exists R.\mathtt{M}$ that returns p
were prepared by the accountant [22, line 411]	implies that $\exists (k, \mathsf{p}) \in N$
are the kind usually prepared [in this situation] [22, line 411]	Δ contains code within a small, known set of indices acc that creates p
can be authenticated by the accountant [22, note 13]	$\exists \mathsf{Auth} : \mathsf{Auth}^{\mathsf{acc}}(x) = 1$ iff $x = \mathsf{p}$

Figure 2: The evidence check E in *Fisher v. U.S.*

In his discussion on foregone-conclusion-based compelled decryption, Terzian [66] describes a split between courts that compel decryption of an entire device and decryption of specific files, and places the burden of proof on those who argue for specific files. Our analysis does not fit neatly into either of these categories, but it is closer to the files interpretation. We do not require the government to specify "every scrap of paper" that must be produced, but we do require the government to avoid compelling files for which the contents of the mind are demonstrably necessary to access (since they did not demonstrate an alternative method of production).

Finally, our conclusion does not go as far as Winkler [85], who claims that the foregone conclusion doctrine does not apply to non-physical evidence and thus compelled decryption is never a foregone conclusion.

4.2 U.S. Supreme Court cases

This section contains our analysis of all federal Supreme Court cases involving the foregone conclusion doctrine. In §4.3, we also analyze several key foregone conclusion Circuit Court cases.

The foregone conclusion doctrine dates back to *Fisher v. United States* [22]. We checked all citations of *Fisher* in Google Scholar's database of case law and found only two subsequent Supreme Court cases that deal with the foregone conclusion doctrine: *United States v. Doe (1984)* [71] and *United States v. Hubbell* [72]. In this section, we show that our definition agrees with the result of all three cases.

Fisher v. U.S. [22]. The *Fisher* case examined a hypothetical[1] in which a taxpayer R was compelled to produce an accountant's papers in R's possession (similar to the motivating example in §2). The act of producing the papers communicates potentially testimonial and incriminating evidence to the government; "[c]ompliance with the subpoena tacitly concedes the existence of the papers demanded and their possession or control by the taxpayer. It would also indicate the taxpayer's belief that the papers are those described in the subpoena."

Fig. 2 translates the circumstances of the hypothetical into an evidence test within our framework. The evidence includes the facts that the government knows that the papers p exist, they reside in one of a small set of possible $\mathsf{locations}$, the papers can be authenticated using only the accountant's testimony (without the taxpayer's help), and finally the taxpayer R can produce them. Hence, the compelled action is simulatable using only information within nature: S can search through $\mathsf{locations}$ and use the accountant to test which papers are the desired ones. This simulation is perfect no matter how the taxpayer R equivocates, as long as R puts the papers $\mathsf{p} \in \mathsf{locations}$ as required by the evidence check E. Moreover, E does not censor R, it allows the true taxpayer code, and all procedures are efficient. Thus, the taxpayer R must produce the legitimate papers p. This analysis matches the Supreme Court ruling that compelling the papers is a foregone conclusion.

We emphasize that all facts contained within the evidence E in Fig. 2 are necessary for the simulator to succeed. The remaining two cases show how the foregone conclusion decision changes when the government cannot pin down the location of, or independently authenticate, the papers.

U.S. v. Doe [71] (1984). The *Doe* case also required the respondent R to produce documents, but unlike in *Fisher*, in this case the government did not have much prior information about the documents. As a

[1] Although hypothetical scenarios described in a court opinion generally do not contribute to the ruling, in the case of *Fisher* the entire foregone conclusion doctrine has arisen from the basis of this hypothetical.

consequence, the non-censorship requirement states that E cannot restrict where the respondent R places the documents in nature, or indeed whether she writes the documents anywhere at all. The wide variety of possible respondent equivocations defeats the simulator S from above (and indeed any other simulator), so Definition 3.2.1 is not satisfied. Our definition again agrees with the result of the case, in which the Court found that "nothing in the record that would indicate that the United States knows . . . that each of the myriad documents demanded by the five subpoenas in fact is in the appellee's possession or subject to his control" [71, note 12] and thus the act of production is not a foregone conclusion.

U.S. v. Hubbell [72]. The *Hubbell* case is complicated by a grant of immunity that is outside of our model; we describe here a subset of the facts that remain relevant in our setting. The government compelled Hubbell to provide "documents fitting within . . . 11 broadly worded subpoena categories." In this case, the government not only sought the documents themselves, but also the "respondent's assistance . . . to identify potential sources of information" and to "testif[y] that those were all of the responsive documents in his control." Our definition is unsatisfiable for compelled actions that are subject to either one of these considerations: given any simulator S, we can construct an equivocating respondent R that decides differently from S which documents are relevant. Once again, our definition aligns with the Supreme Court's decision that it was "unquestionably necessary for respondent to make extensive use of 'the contents of his own mind' in identifying the hundreds of documents responsive to the requests of the subpoena" [72, line 43].

4.3 Circuit Court cases

In addition to Supreme Court foregone conclusion cases, there are also several important foregone conclusion cases in the circuit courts. Rather than check all approximately 1200 circuit court cases citing Fisher, we rely on prior law review articles [17,35,39,41,50,64,85,86] to identify the most relevant circuit court cases. Five circuit court cases are mentioned or cited that involve the foregone conclusion doctrine but do not involve encryption. For a brief summary of the encryption cases, see §4.4.

U.S. v. Greenfield (2nd circuit) [79]. *Greenfield*, a 2016 case from the 2nd Circuit Court of Appeals, is a good example of a case that came close to being a foregone conclusion, but needed slightly stronger evidence to prevent the recipient Greenfield from equivocating. Greenfield had been accused of tax evasion and, after some back-and-forth, had been compelled to produce three categories of bank records (for accounts already known to the government), some non-bank documents ("ownership records", "professional services documents", and "communication documents") plus his expired passport and other documentation for trips in his passport. [79, line 114] The court found that, had the government requested the documents in 2001, production of the passport and travel documents would have been a foregone conclusion (authenticated by the Department of State or airlines), and while the government showed knowledge of the existence and control of the other documents, it would have relied on Greenfield's testimony to authenticate them. However, in 2013, when the summons occurred, even the passport and travel documents were no longer a foregone conclusion, because the government could not show that Greenfield had retained control over them through the 12-year gap. For our purposes, the 2001 analysis is more interesting than the 2013 analysis, since the categories of documents compelled were very similar, but came to be different nonetheless. Let br or, psd, cd and td be unknown strings, and exists Auth such that $\mathsf{Auth}_{\mathsf{td}}(x) = 1$ if $x = \mathsf{td}$. Let $\mathsf{possess}$ be a small set of indices. The evidence established the information shown in Figure 3.

We first examine what would have occurred in *Greenfield* if the production had been compelled in 2001. The 2nd Circuit determined that compelling the travel documents td was a foregone conclusion, since the existence, control, and authenticity of the documents were established. We can see that a S that reads each location in $\mathsf{possess}$ and checks its authenticity with $\mathsf{Auth}_{\mathsf{td}}(\cdot)$ will always return the same td as is returned by R.`M'`.

The existence of the communication documents cd was not proven, thus, R can choose not to place them in N at all, making the chance of their recovery by S^N exponentially small. Similarly, the ownership records

Knowledge of government	**Formalization in E**
In 2001, the travel documents...	$\exists k, \mathsf{td}$ such that:
existed [79, line 123]	$(k, \mathsf{td}) \in \Delta$
were in Greenfield's possession [79, line 123]	$k \in \mathsf{possess}$ ($\mathsf{possess}$ is a small set of indices)
were under Greenfield's control [79, line 123]	$\exists \mathtt{M} : R.\mathtt{M} = \mathsf{td}$
could be authenticated [79, line 123]	$\exists \mathsf{Auth} : \mathsf{Auth}^N(x) = 1$ iff $x = \mathsf{td}$
In 2001, the communication documents...	
would be in Greenfield's possession [79, line 123]	$(\exists i : N[i] = \mathsf{cd}) \to i \in \mathsf{possess}$
In 2001, the ownership records...	
existed [79, line 122]	$(j, \mathsf{or}) \in \Delta$
In 2001, the bank records...	
existed [79, line 119]	$(i, \mathsf{br}) \in \Delta$
were in Greenfield's possession [79, line 119]	$i \in \mathsf{possess}$
were under Greenfield's control [79, line 120]	$\exists \mathtt{M'} : R.\mathtt{M'} = \mathsf{br}$

Figure 3: Evidence shown in *Greenfield*

existed somewhere in N, but since no knowledge is known about their whereabouts or control, S^N's chance of recovering them is still exponentially small.

However, the bank records br could not be authenticated. Supposing the government has some idea of what the documents should look like (i.e. their distribution, which we assume is not overwhelmingly in favor of one outcome; else authentication would be trivial), S has the power to search $\mathsf{possess}$ (a small subset of N) and return the document there most likely to be the correct record. However, R has the ability to forge an alternate $\mathsf{br'}$ (e.g. by resampling) and *also* put that in $\mathsf{possess}$ in addition to the true $\mathsf{br'}$. In this case, C (calling $R.\mathtt{M}$ and returning the result, without interacting with G) would return the true br consistently, but S would mistake the false $\mathsf{br'}$ for the real br a non-negligible amount of the time. This allows $\mathcal{D}$ which has the correct br hardcoded into it to distinguish between the two possible distributions.

As an added note, the non-foregone decision for the bank records seems to rest on the inability of the government to authenticate them, and in fact some other court cases seem to be more lenient in their standards for authentication. If such an authentication mechanism were known, the bank records would have been a foregone conclusion by the same rationale as the travel documents.

In re Grand Jury Proceedings (8th circuit) [30]. In this 8th circuit case, the Bayirds were served with a subpoena for several categories of documents, primarily business documents related to their income. The circuit court found that the subpoena was insufficiently well-defined – the Bayirds' choice of which documents were covered and which were not could be testimonial [30, line 381]. In short, R could return different distributions of documents while still complying with the evidence. As long as it returns a different distribution than S (which must work for all choices of R), the results will be distinguishable and therefore not a foregone conclusion.

In re Grand Jury Subpoena (9th circuit) [31]. In this case, respondent John Doe was an employee of a corporation accused of price fixing DRAM chips. Two subpoenas were issued: the first to John Doe, compelling the production of all documents he had related to DRAM sales, including calendars, diaries, and notes; the second to the corporation [31, line 911]. The government had reasonably extensive knowledge of Doe's actions through interviews and other documents. Nonetheless, the subpoena served to Doe was overbroad and asked for categories of documents that the government did not know existed until the results of the *second* subpoena (to the corporation) was responded to. Furthermore, the government never provided a means to authenticate the personal documents (e.g. diaries) without Doe's testimony. S would have no means of simulating these categories of documents, and as such, the result is not foregone.

U.S. v. Ponds (D.C. circuit) [81]. In *Ponds*, a defense attorney Ponds was accused of tax evasion and fraud with regard to a previous case, and was asked to produce documents in several categories. For our purposes, we will consider only two of these categories: documents referencing a white Mercedes Benz (thought by the government to have been illegally held by Ponds) and copies of correspondence between the Law Offices of Navron Ponds and courts and prosecutors having to do with the prior case. For the first category, while the government suspected Ponds had the car, the only information in E was that the car was "normally parked at Ponds' apartment and was registered to his sister." [81, line 325] For the second category, since the government was party to the correspondance [81, line 325] and Ponds was expected to have retained a copy, E may require the value in N representing the documents to be the exact string that the government saw before, allowing S to duplicate it exactly. Compelling the first category was not a foregone conclusion; compelling the second category was.

In re grand jury subpoena (2nd circuit) [32]. In this 2nd circuit case, the government subpoenaed John Doe to produce "[t]he original version of any diary or calendar for the year 1988, a copy of which has been produced to the SEC" [32, line 88], and this act of production was determined to be a foregone conclusion. Ultimately, this case is fairly straightforward in our model; the evidence demands that a copy of the documents be in Nature in a set of indices corresponding to the SEC, and the simulator could produce the documents by obtaining them from there.

4.4 Compelled decryption cases

As stated at the beginning of this section, we do not fully analyze prior encryption cases under our model. This is because, to our knowledge, only two encryption cases concerning the foregone conclusion doctrine have risen to the level of the circuit courts, and they were decided quite differently. The 11th Circuit found in *In re Grand Jury Subpoena Duces Tecum* [33] that compelled decryption of a hard drive with unknown contents was *not* a foregone conclusion, since the government had not shown that the drives contained any files. That is, they impose a requirement that the government must know what they will find on the encrypted drive with "reasonable particularity" [17, 31, 33, 81]. However, the 3rd Circuit has rejected this requirement [76]. They found in *U.S. v. Apple MacPro Comput.* [76] that decryption of a particular hard drive *was* a foregone conclusion, in part because they had verbal testimony from the defendant's sister as to the contents of the drives.

There are also many cases in lower courts involving encryption and passwords. Most of these courts agree that compelling the disclosure of the password (instead of compelling the defendant to enter the password into the device to unlock it) is not permissible even under the foregone conclusion exception to the fifth amendment [14,54,80]. Only one state supreme court found that disclosure of the password itself is allowable, stating that passwords are "of minimal testimonial value" [59]. Several states found that compelling entry of passwords (rather than disclosure) is allowed, but cite different reasons. In Massachusetts, the standard is either that the government must show that the defendant knows the password [16] or that she knows the password/key, knows that the device is encrypted, and has been shown to be the owner of the device and its contents [15]. In North Carolina, a recent case allowed compelling entry of the password, but the defendant had already admitted to using the device to store illegal material [77], leaving the alternative undecided. The U.S. district court for the northern district of California decided that since biometrics are compellable, so too passwords must be compellable [74]. On the other hand, when denying an application for a search warrant, the same court decided a year later that since biometrics often serve the same purpose as passwords, perhaps both biometrics and passwords are not compellable [45]! Finally, Indiana's state Supreme Court ruled that even entry of the password is not compellable unless the government can show that it knows the existence of specific files, and that they belong to the defendant [56]. We refer readers to [47] for additional details on several of these cases.

5 Compellability of Cryptographic Systems

In this section, we analyze whether it is a foregone conclusion for the government to compel the respondent to use some common cryptographic constructs: one way functions, commitment schemes, encryption schemes, and non-interactive zero-knowledge proofs. We show that compelling the use of these cryptographic primitives is typically *not* a foregone conclusion under our definition, although there exist fact patterns for which it is foregone.

For consistency, throughout this section we presume that the respondent contains a method $R.s$ that, if called, deterministically reveals a secret within the respondent's mind like a password, encryption key, or value inside a commitment. Our theorems often explicitly encode the government's awareness that the respondent knows this secret (even if the government does not know the value of $R.s$).

5.1 One Way Functions

Let $f : \mathcal{X} \to \mathcal{Y}$ be a one-way function. In this section, we show that compelling a preimage of $y \in \mathcal{Y}$ is typically not a foregone conclusion. Specifically, this compelled action is only foregone if the government can demonstrate that R knows exactly one preimage and the government knows an alternative method to produce the same preimage.

Lemma 5.1.1. *Let $E^N(R) := \exists R.s \in \mathcal{X} \wedge f(R.s) = y \wedge E'$ be the evidence that the method $R.s$ exists, it produces an element in $\mathcal{X}$ that is a preimage to y, and any additional evidence E' that the government knows. Then, the compelled action $C^{N,R} := R.s$ is a foregone conclusion with respect to evidence E if and only if this evidence suffices for the government to provide a simulator S that reliably produces $R.s$.*

Proof. This compelled action C is deterministic, so the government must simulate it perfectly to evade detection by the distinguisher that has the real $R.s$ hardcoded into it. □

Whether the government can build S depends on the additional evidence E' at its disposal. If $E' = \emptyset$ and $y \leftarrow \mathcal{Y}$ is sampled uniformly, then simulation is impossible by the one-wayness of f. However, there exists evidence that permits government simulation, such as if E shows that the respondent wrote down $R.s$ somewhere in Nature.

This question has immediate relevance to existing court cases – in the most famous example, the 3rd Circuit ruled that a device owner can be compelled to decrypt the contents if the Government can show its knowledge (via hash values) of files on the device, and that the owner is capable of accessing them [76, line 248]. Our definition would arrive at a similar conclusion but via different means. By Lemma 5.1.1, compelling the preimage of a hash is not a foregone conclusion on its own. Nevertheless, in the facts of the case [76], digital forensic examiners were able to identify encrypted files with specific hash values that were known to contain child pornography. We believe the Government could have shown that it was able to produce testimony or evidence that would describe the files (preimages) – the forensic examiners could likely fill such a role. This would allow the creation of a simulator that would make requesting the files (preimages) a foregone conclusion. While the court in [76] forced the decryption of the entire device (actually multiple devices), we believe that only the specific files with known preimages should have been compelled. The remaining files could not have been returned without the use of the respondent's mind.

5.2 Commitment schemes

Compelling a randomized functionality introduces a new wrinkle beyond the cases discussed in §4 and §5.1: now the simulator merely needs to be computationally indistinguishable from the real transcript, rather than being identical.

Concretely, we consider below a randomized commitment scheme $(\mathsf{Com}, \mathsf{Decom})$ that is computationally binding and hiding. The algorithm $\mathsf{Com}(s) = (c, r)$ produces a commitment c that is sent to the (government) receiver and a random state r that is maintained by the (respondent) committer, and $\mathsf{Decom}(c, r) = s$ uses both of these values to recover the original secret s. we show below that it is a foregone conclusion for the government to compel the respondent to commmit (but not decommit!) to the secret in her mind.

Lemma 5.2.1. *A compelled action $C^{N,R}_{comm}$ to sample a commitment $c \leftarrow \mathsf{Com}(R.s)$ is a foregone conclusion, as long as the government has evidence E that the method $R.s$ exists.*

Proof. The government can provide the trivial simulator S_{comm} that chooses a random value x and returns a commitment to it. We claim that this simulator can even fool a distinguisher that has $R.s$ hardcoded into it, because R *cannot* communicate the randomness used within the real commitment since it is only chosen later within C_{comm}. If there exists a distinguisher $\mathcal{D}$ that can distinguish a commitment to $R.s$ from a commitment to a random x without knowing the randomness used (i.e., without opening), then $\mathcal{D}$ breaks the hiding property of Com. □

Similarly, it is also foregone to compel a commitment to a value s that is not within R. This includes the settings in which C samples a secret s at random, hardcodes s, or obtains s from a known location in nature.

On the other hand, compelling the opening of a commitment to a secret value is *not* foregone unless the government already had the ability to compel the secret via other means. This lemma leverages the power of our composition theorem.

Lemma 5.2.2. *Let $C^{N,R}_{decom}$ be a machine that decommits to a value c provided by the government G^N. Also, let $E := \exists R.s \wedge E'$ be any evidence that includes the fact that $R.s$ exists, and let S be any simulator.*

Then, $(C^{N,R}_{decom}, G^N)$ is a foregone conclusion with respect to E and S if and only if there exists a simulator S' such that compelling the secret $R.s$ is a foregone conclusion with respect to E and S' independently of the commitment scheme.

Proof. We apply the Theorem 3.3.1 with the machines C_{comm} and C_{decom}. Combining the theorem with Lemma 5.2.1, there exists some simulator S' such that the composed machine $C := C_{\text{comm}} \| C_{\text{decom}}$ is a foregone conclusion with respect to E and S' if and only if C_{decom} is foregone with respect to E and S. Note that C simply commits to this value, provides the commitment to the government and then receives the same commitment back, and opens the commitment in a binding manner. Hence, C is equivalent to the machine C' that outputs the secret $R.s$, without any commitment scheme involved. Therefore, C_{decom} is foregone with respect to E and S if and only if C' is foregone with respect to E and S', as desired. □

5.3 Zero knowledge proofs

Next, we consider an interactive proof protocol Π, where R's secret equals a witness to an NP language. It turns out that compelling a ZK proof is possible but uninteresting. While most of the claims in this section require the government's evidence to contain the fact that R knows a secret with a particular structure, in this case that is already equivalent to the knowledge gained from the ZK proof itself. Sadly, this evidence is required, even for languages in P! The lemma below also applies to ZK arguments and to proofs of knowledge since it is agnostic to the knowledge soundness property.

Lemma 5.3.1. *Let (C, G) execute an interactive ZK proof where C acts as the Prover with witness R.w, and G acts as the Verifier. Given any evidence E, there exists a simulator S such that (C, G) is a foregone conclusion with respect to E and S if and only if the government's evidence suffices to show that R.s is a witness to the NP statement.*

Proof. If the evidence E allows R to equivocate between a valid and invalid witness for $R.s$, then no simulator can consistently emulate both options. On the other hand, if E guarantees that $R.s$ is a witness, then the compelled action is simulatable by the algorithm S that hardcodes the circuit G and runs an execution between the ZK simulator S_{ZK} and verifier G, potentially rewinding G as usual. The only remaining equivocation available to the respondent is her choice of $R.s$ among satisfying witnesses, but this change is inconsequential by witness indistinguishability. □

Next, we consider non-interactive ZK proofs of knowledge using a common reference string (CRS) as the trusted setup, which is sampled honestly by the respondent and checked by the evidence. In this scenario, the government is in a weaker position than before: in order to compel a NIZK, the government must know a witness themselves.

Lemma 5.3.2. *Let C denote a non-interactive ZK proof using the witness $R.s$. It accesses a CRS stored in Nature, where the CRS is placed by R and verified by E. If there exists E and S such that C is a foregone conclusion with respect to E and S, then there exists an extractor X that returns a witness.*

Proof. Because the S has no control over the CRS, its proofs are real. If S produces a proof with noticeable probability (over the random sampling of the CRS, among other things), then the knowledge soundness property guarantees the existence of an extractor X' that can extract a witness when executing S multiple times with on different choices of CRS. While the foregone conclusion game in Def. 3.2.1 only runs S once (without rewinding), we can construct the desired extractor X by running the entire game many times, since R will honestly sample the CRS independently each time. □

5.4 Pseudorandom functions

Next, we examine the circumstances under which the government may compel the use of a pseudorandom function family $\{F_k : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$. This question turns crucially on whether the key is sampled freshly and ephemerally as part of the compelled action, or if the action requires the use of a long-running key that can be used elsewhere in Nature.

Lemma 5.4.1. *Let $C_{prf}^{N,R}$ be the circuit that samples a random key $k \in \mathcal{K}$ and outputs $F_k(R.s)$. This compelled action is a foregone conclusion with respect to any evidence E that includes the fact that the method $R.s$ exists.*

Proof. Just as with Lemma 5.2.1, the government can provide the trivial simulator S that chooses a random output $y \in \mathcal{Y}$. Any algorithm $\mathcal{D}$ that can distinguish C_{prf}^{R} from S also serves to break the pseudorandomness of F_k. □

Lemma 5.4.2. *Let $\tilde{C}_{prf}^{N,R}$ be the circuit that computes $F_k(x)$, where the key equals the respondent's secret $k = R.s$ and the constant $x \in \mathcal{X}$ is publicly known. Given the minimal evidence $E := \exists R.s$ that R knows the key, there is no simulator S under which $\tilde{C}_{prf}$ is foregone with respect to E and S.*

Proof. This evidence permits R to equivocate between two secrets k and k' that produce different outputs $F_k(x) \neq F_{k'}(x)$, and it must be possible to efficiently sample such keys or else making a query to x would distinguish the PRF from a random function. Any simulator S must fail to output at least one of these strings with noticeable probability, and R can choose this one to evade simulation. □

Lemma 5.4.3. *Let $\tilde{C}_{prf}$ be defined as in the previous lemma. Suppose the government knows the value of k as evidence $E^N(R) := (R.s = k)$. Now, there exists a simulator such that $\tilde{C}_{prf}$ is a foregone conclusion.*

Proof. Simulator S^N computes $F_k(m)$ from the known values. This perfectly emulates the real transcript. □

5.5 Symmetric encryption

In this section, we consider the compellability of symmetric (authenticated) encryption, which is of particular importance due to its ubiquitous use within full-disk encryption systems. We show that if the respondent keeps the secret key (or a high-entropy password used to derive it) only in her mind, and there are no side channels in Nature capturing the intermediate state during encryption and decryption, then both compelled encryption and decryption are not foregone conclusions.

We focus on the Counter Mode construction of symmetric encryption from a pseudorandom function where KeyGen samples a PRF key, $\mathsf{Enc}(k, m) = (r, F_k(r) \oplus m)$ and $\mathsf{Dec}(k, (r, c)) = F_k(r) \oplus c$. We remark though that the following theorem would also hold for many other modes of operation, including ones that provide authenticity.

Theorem 5.5.1. *Suppose the respondent stores two secrets: a secret key k and a message m; that is, $R.s = (k, m)$. With respect to the evidence $E := \exists R.s$ that R knows the secrets,*

- *Compelled encryption of message m under an ephemeral key $k^* \leftarrow \mathcal{K}$ is a foregone conclusion using the simulator that outputs a random element of the ciphertext space.*
- *Compelled encryption of message m or decryption of a ciphertext c using the respondent's secret key k are not a foregone conclusion with any simulator.*

Proof. For the first claim, S can simply sample a random string c' in the ciphertext space. Any algorithm $\mathcal{D}$ that can distinguish $(r, F_{k^*}(r) \oplus m)$ from (r, c) also serves to break the pseudorandomness of F_k.

For the second claim, we assume without loss of generality that the distinguisher has m or c hardcoded, and thus the question reduces to simulating $F_k(r)$. For most alternative keys k' it must be the case that $F_k(r) \neq F_{k'}(r)$ by pseudorandomness, and the evidence permits R to equivocate between secrets k and k'. Any simulator S must fail to output at least one of these strings with noticeable probability, and R can choose this one to evade simulation. □

The above theorem leverages the strength of the respondent's key management within her own mind and the weakness of the government's evidence in preventing R from equivocating. If either of these two properties changes, then decryption might be compellable. Essentially: if there exists any method for the government to decrypt data without your help, then they can instead compel you to do so.

Theorem 5.5.2. *If the government knows evidence E and a PPT algorithm K such that $s \leftarrow K^N$ recovers R's secret key s, then there is a simulator S such that compelled decryption of a known ciphertext c is a foregone conclusion under E and S.*

Proof. Construct the simulator S^N that runs K^N, fetches the ciphertext from the known location, and uses the key to decrypt the ciphertext. This simulator is efficient and it perfectly emulates the real transcript. □

This theorem applies broadly to several categories of encryption schemes: enterprise or cloud backup systems that use an external key (e.g., one stored in a Hardware Security Module), threshold encryption with a threshold smaller than the full number of parties since the Fifth Amendment only protects against *self*-incrimination, and exceptional access systems that permit law enforcement access to encrypted devices via a key known to the vendor [8,62], one or more courts [10,42], law enforcement [8], or the device itself [52]. In all such cases, the existence of an alternative key bypasses the testimonial aspects of the respondent's assistance.

In the next section, we show specific constructions of secure multi-party systems that remain resilient to compelled actions; these can be used to build threshold and backup systems with stronger Fifth Amendment protections.

6 Resilience Against Compelled Requests

So far, we have only considered how *past* actions impact whether or not a *current* compelled request is foregone. In this section, we ask whether a *current* protocol execution may open parties up to *future* compelled requests. If running a protocol does not open a party up to additional compelled requests, we call it *FC-resilient.* In this section, we formally define FC-resilience, design and implement a 2-party secure computation protocol that is both malicious secure and FC-resilient for one party, and leverage differential privacy to design a multi-party computation protocol that is FC-resilient for many parties.

6.1 Defining FC-resilience

In this section, we ask whether running protocols that are unrelated to any current legal issues will open the parties up to *future* compelled requests that would not have been possible before running the protocol. To see why this is an issue, consider the following scenario: Alice participates in a multi-party computation with several other parties, including Bob, in which she and Bob receive the same output. Later, Alice is the target of a compelled request in which the government seeks the result of the computation. Since the government could access the information without involving Alice (by compelling testimony from Bob instead), the output of the protocol is a foregone conclusion and Alice must provide it. Depending on the function computed, this may reveal information about Alice's secret inputs that was not previously compellable because it had only been stored in Alice's mind.

We provide a proactive cryptographic countermeasure against the above scenario, which we dub *FC-resilience*. Informally, we say that a protocol is FC-resilient if all compelled actions that are foregone after running the protocol were already foregone before running the protocol.

Model. Concretely, we consider an interactive protocol Π between $n+1$ parties $P_*, P_1, \ldots, P_n$ that is secure for computing function f with the $n+1$ parties' inputs up to abort and with erasures. If P_* has just as much ability to equivocate on any compelled action before running the protocol as after, then we say that Π is *FC-resilient for* P_*.

We use the nomenclature that the government's evidence E *checks* a string X if it verifies that X exists in Nature at a public canonical location, returning **false** otherwise. (The evidence may still return false even if X does exist, unless some other conditions are met as well.)

In our setting, we presume the government knows that the parties have executed t timesteps of the protocol and that its evidence will check for this fact. Given a protocol Π and a timestamp t, we say that Π's modifications to nature in the $\mathcal{F}$-hybrid model with secure erasures, denoted M_t^{Π,P_*}, include the messages and local state of all protocol parties after running t steps of Π, except for P_*'s tapes for its communication with sub-module $\mathcal{F}$. (Formal modeling of the Turing machines of the parties can be found in Appendix A.1.)

We are now ready to define FC-resilience. Our definition requires that the execution of Π cannot subject P_* to any new compelled actions, no matter what time the government pauses Π to issue its request.

Definition 6.1.1 (FC-resilience for P_*)**.** Let protocol Π be a protocol among parties $P_*, P_1, \ldots, P_n$. Let E be an evidence machine. We say that Π is *FC-resilient for party* P_* if the following holds true:

Suppose (C, G) is a foregone conclusion in the $\mathcal{F}$-hybrid model when addressing party P_* with respect to E_t^Π, S, for some $t \geq 0$, where E_t^Π runs machine E and also checks M_t^{Π,P_*}. Then there exist machines C_0, G_0, and S_0 such that: (1) (C_0, G_0) is a foregone conclusion with E_0 and S_0; and (2) The two compelled disclosures have indistinguishable transcripts: $\forall\ R, \forall N,\ \tau({C_0}^{N,R}, {G_0}^{N}) \approx_c \tau(C^{N,R}, G^N)$

$\mathcal{F}$ **separates P_*'s mind from local state.** It would be convenient if we could keep all of P_*'s state as part of the "contents of her mind" rather than Nature. However, P_* is not likely to be storing her state or performing computations in her head. More likely, P_* will be doing these on a local computer, and she can only hold a small amount of state (e.g., a password) in her head.

To model this, we permit P_* to access an ideal sub-module which encapsulates both the small, long-term "state of the respondent's mind" as well as the limited operations that the respondent carefully performs only when she is not at risk of being compelled. Qualitatively, it is preferable to minimize the number of times $\mathcal{F}$ is invoked and the state that it stores.

The formal design of this sub-module is inspired by the treatment of tamper-proof hardware tokens in UC [38]. However, it represents something very different in this model: the occasions when the party is "currently using" the limited long-term state of the mind. The model prevents this state from entering Nature during the computation. However, any function of the output of this sub-module does become part of Nature, and it is incumbent upon P_* to choose a functionality $\mathcal{F}$ whose outputs don't trivially cause new compelled action to become foregone conclusions.

Different possibilities for the actual functionality of $\mathcal{F}$ are possible depending on how assured P_* is of a lack of sudden compelled requests. In this work, we consider the functionality $\mathcal{F}_{\text{pbkdf}}$ that computes a PBKDF of the party's password in a safe space. This functionality is described in Fig 4. $\mathcal{F}_{\text{pbkdf}}$ samples a long-term password from a distribution with sufficient min-entropy λ and then runs a password-based key derivation function on demand.

One might worry that using a password-derived key would subject our construction to password brute-force attacks that would not occur with a non-password-derived key K. Fortunately, as long as we store

```
Functionality $\mathcal{F}_{\text{pbkdf}}$
Public parameters: $\lambda$, PBKDF $f : \{0,1\}^* \to \{0,1\}^\lambda$
Setup: Upon receiving setup from $P$, do the following:
  If there is already a stored pw, halt.
  Generate a random pw from a distribution with good min-entropy
  Generate a random salt uniformly at random with good entropy
  Store pw
  Output salt to $P$
Refresh: Upon receiving refresh from $P$, do the following:
  Generate a random salt uniformly at random with good entropy
  Output salt to $P$
Query: Upon receiving $(\mathsf{query}, \mathsf{salt}, m)$ from $P$:
  Check whether there is a stored pw. If there is not, halt.
  Output $f(\mathsf{pw}, \mathsf{salt}, m)$ to $P$
```

Figure 4: Ideal functionality for $\mathcal{F}_{\text{pbkdf}}$, a possible version of $\mathcal{F}$, which assumes P will not be compelled while computing a PBKDF of her password

the PBKDF salt in the same manner that K would have been stored (e.g., in a trusted enclave), then our password-derived key resists brute-force attacks and retains the same cybersecurity protections as K.

Government may compel at any time. Just as our foregone conclusion definition gave the government the strong power to view anything in the rest of the world, our FC-resilience definition allows the government full freedom to determine when to make its compelled request. An FC-resilient protocol must maintain protection against compelled requests made against P_* whether the protocol has completed or has been interrupted partway through (e.g., with intermediate state that has not yet been deleted). We presume that compelled requests only occur at one instant of the protocol execution; because the government is non-censoring, we presume that parties can alert each other to abort the protocol if they have been compelled to disclose information. Due to our composition theorem, it suffices to consider a single compelled request made by the government to P_*. Finally, we presume that the government is aware of the protocol execution.

6.2 FC-resilient two-party computation

In this section, we design and implement secure 2-party computation protocols based on Yao's garbled circuits [88] that are FC-resilient for one party in the $\mathcal{F}_{\text{pbkdf}}$-hybrid setting.

This is a non-trivial objective: While executing most MPC protocols, the parties' inputs and intermediate state are typically all foregone conclusions for the simple reason that all the (large) state is distributed throughout Nature rather than being stored within anyone's mind. This compelling adversary violates the non-collusion assumption required for secure MPC (even if the original protocol was malicious secure, or handled adaptive or mobile adversaries).

Using fully homomorphic encryption (FHE) can protect against compelled disclosure because compelled decryption is not a foregone conclusion (§5.5). For faster performance, we construct and implement a new secure computation protocol that is resilient to government compelled disclosure without the need for FHE. Our protocol involves careful modifications to Yao's garbled circuits at the input and output stages. It assumes secure deletion and a reliable communication channel whereby the parties can halt the secure computation if any or all of them are compelled to provide their state.

Tag	**Self-garbled masked input** ($\hat{x}_w$)
$PBKDF(w, x_w = 0)_{1 \cdots n-1}$	$PBKDF(w, x_w = 0)_n \oplus \lambda_w$
$PBKDF(w, x_w = 1)_{1 \cdots n-1}$	$PBKDF(w, x_w = 1)_n \oplus (\lambda_w \oplus 1)$

(a) Self-garbled input tables for wire w (permutation not shown)

Info from E	**Self-garbled masked output** (x_w)
$\hat{x}_w = 0, s_w = 0$	$PBKDF(w, \hat{x}_w = 0, s_w = 0) \oplus r_w$
$\hat{x}_w = 0, s_w = 1$	$PBKDF(w, \hat{x}_w = 0, s_w = 1) \oplus (r_w \oplus 1)$
$\hat{x}_w = 1, s_w = 0$	$PBKDF(w, \hat{x}_w = 1, s_w = 0) \oplus (r_w \oplus 1)$
$\hat{x}_w = 1, s_w = 1$	$PBKDF(w, \hat{x}_w = 1, s_w = 1) \oplus r_w$

(b) Self-garbled output tables for wire w

Table 1: Self-garbled tables for the garbler in the authenticated-garbling-based 2PC protocol FC-resilient for the garbler. w is the wire index, x is the true wire value, $\hat{x}$ is the masked wire value, and $\lambda = r \oplus s$ is the mask on the wire. r was held by the garbler during pre-processing but was securely deleted; s is held by the evaluator.

6.2.1 Construction of FC-resilient 2PC

We consider Yao's garbled circuits where the garbler additionally has access to the ideal module $\mathcal{F}_{\text{pbkdf}}$. For now assume that only the garbler receives output from the 2PC; we will relax this assumption later. Our method maintains malicious security against the evaluator, and it is compatible with two methods for ensuring malicious security against the garbler: cut-and-choose [44, §3.3] and authenticated garbling [83].

The main idea is that the garbler will "self-garble" tables for her input and output wires, so that even she does not know how to interpret her input or output without re-entering her password. The garbler inputs her password into the $\mathcal{F}_{\text{pbkdf}}$ module during three phases of the protocol. First, during pre-computation when preparing the garbled circuits, the garbler generates labels for the input wires uniformly at random (as normal) and augments these labels with a pseudorandom tag that is based on the PBKDF. For the outputs to the circuit, garbler appends no-op gates to the circuit where the output wire labels are again chosen pseudorandomly using the PBKDF. She then securely deletes the mapping of wire labels for her input and output bits, so that it can only be reconstructed with her own password. Second, upon receiving her own input, the garbler uses the PBKDF again and matches the resulting values with the pre-computed tags; this informs the garbler which wire labels to send to the evaluator while safeguarding the input itself. Third, at the end of the protocol, the garbler uses her PBKDF to find the outputs by using the output tables of the no-op gates. The concrete self-garbled tables for authenticated garbling are shown in Table 1.

Figure 5 depicts the full protocol Π, including a detailed description of all changes to garbled circuits compatible with the cut-and-choose approach. In total, our construction imposes an additive overhead to Yao's garbled circuits equal to a constant number of PBKDF calls per input and output wire. We emphasize that neither the password sub-module nor the garbler's password are required during circuit evaluation; they are only used at the beginning and end to provide input and read output.

Theorem 6.2.1 (simplified)**.** *Under the same cryptographic assumptions as malicious-secure Yao's garbled circuits, protocol* Π *in Figure 5 is secure against malicious adversaries and is FC-resilient for the garbler.*

We rigorously specify this theorem and its proof in Appendix B. Here, we provide a high-level overview.

Proof sketch. If the protocol is not a secure computation of f, then either we can break the pseudorandom function called by $\mathcal{F}_{\text{pbkdf}}$ or we break the security of the existing 2-party secure computation of f [44]. This follows by a hybrid argument in which the functionality outputs are replaced by random values that are independent of the garbled circuit.

Additionally, the protocol is FC-resilient for the garbler no matter when the government interrupts the protocol execution. Since the protocol contains only one secure deletion step, without loss of generality the

Figure 5: 2PC protocol Π that securely computes $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ among garbler G and evaluator V with malicious security for both parties, and with FC-resilience for the garbler.

Input: G has input $x \in \{0,1\}^n$, V has input $y \in \{0,1\}^n$.

Auxiliary input and setup: G has access to $\mathcal{F}_{\text{pbkdf}}$ and has already run $\mathcal{F}_{\text{pbkdf}}(\mathsf{setup}, 2\lambda)$. Both parties have λ, s, and a description of circuit C_1 such that $C_1(x,y) = f(x,y)$. The parties have a reliable channel through which they can indicate that they were the target of a compelled request.

Output: G receives $f(x,y)$,V receives no output.

Throughout the protocol, if either party receives a compelled request, communicate this on the reliable channel and halt.

Preprocessing:

0. G generates a fresh salt $\mathsf{salt} \leftarrow \mathcal{F}_{\text{pbkdf}}(\mathsf{refresh})$.

1. Circuit construction and modification: The parties append a single no-op gate to the output of C_1; call the resulting circuit C_2. The parties then modify C_2 as in [44].

2. Commitment Construction:

a. Output Wire Label Generation: Let the output wire of C_2 (the output of the no-op gate) be indexed by out. For $c \in [s]$ and for $b \in \{0,1\}$, G creates $w_{c,\mathsf{out},b} = \mathcal{F}_{\text{pbkdf}}(\mathsf{query}, \mathsf{salt}, (c, \mathsf{out}, b))_{1\cdots\lambda}$. No table from this wire to the output is created.

b. Remaining Circuit and Commitment Generation: G generates the rest of the wire labels and garbled tables as normal. She also computes commitments to V's input wires, and commitment-sets for her own input wires as normal (see [44]). Let $w_{c,i,b}$ be the wire label for circuit c, input wire i, and wire value b. Let $r_{c,i,b}$ be the commitment randomness used to commit to $w_{c,i,b}$.

Evaluation Part 1 (V's input known):

3. Execute Oblivious Transfers as usual

4. Send Circuits and Commitments. (Note that V does not have the ability to map the final gate output to 0 or 1, it only gets $w_{c,\mathsf{out},b}$ which it must send back to G.)

5. Prepare challenge strings

6. Decommitment phase for check-circuits. Let the remaining set of evaluation circuits be $C \subset [n]$.

Preparation for G input:

7. Generation of Permuted Tables

a. Tag and mask sampling: For $c \in C, i \in [n], b \in \{0,1\}$, G queries $z_{c,i,b} = \mathcal{F}_{\text{pbkdf}}(\mathsf{query}, \mathsf{salt}, (c,i,b))$, and parses $z_{c,i,b} = (t_{c,i,b}, m_{c,i,b})$ as a "tag" $t_{c,i,b}$ and a "mask" $m_{c,i,b}$, each of length λ.

b. Permutation bit sampling: G picks $p_{c,i} \leftarrow \{0,1\}$ uniformly at random for $c \in C, i \in [n]$.

c. Build permuted masked tables: For each circuit $c \in C$, for each input wire $i \in [n]$, mask the wire value and commitment randomness with $m_{c,i,b}$, label it with tag $t_{c,i,b}$ and permute the rows for $b = 0$ and $b = 1$ depending on $p_{c,i}$. That is, record rows $(c, i, t_{c,i,p_{c,i}}, m_{c,i,p_{c,i}} \oplus (w_{c,i,p_{c,i}} \| r_{c,i,p_{c,i}}))$ and $(c, i, t_{c,i,1-p_{c,i}}, m_{c,i,1-p_{c,i}} \oplus (w_{c,i,1-p_{c,i}} \| r_{c,i,1-p_{c,i}}))$ ordered by $p_{c,i}$.

8. Secure deletion: G securely deletes all information except the permuted table from step 7c and the salt.

Evaluation Part 2 (G's input known):

9. Decommitment phase for G's input in evaluation-circuits:

a. Now that G has her input $x = x_1 \cdots x_n$, she recomputes $(t_{c,i,x_i}, m_{c,i,x_i}) = \mathcal{F}_{\text{pbkdf}}(\mathsf{query}, \mathsf{salt}, (c,i,x_i))$ for $c \in C, i \in [n]$.

b. G finds the table row indexed by c, i, t_{c,i,x_i} and uses m_{c,i,x_i} to unmask w_{c,i,x_i} and r_{c,i,x_i}.

c. G decommits to w_{c,i,x_i} and sends that to V (as usual).

10. Correctness and consistency checks

11. Circuit evaluation

a. For circuits $c \in C$, V evaluates the circuit up until the final wire label, $w_{c,\mathsf{out},b}$. It sends this value back to G.

b. G queries $w'_{c,\mathsf{out},b'} = \mathcal{F}_{\text{pbkdf}}(\mathsf{query}, \mathsf{salt}, (c, \mathsf{out}, b'))$ for circuits $c \in C$ and $b' \in \{0,1\}$. If neither of these match the value sent by V, G aborts and outputs $\perp$. If $w'_{c,\mathsf{out},b'}$ matches one of the values sent by V for all $c \in C$, then G takes the b' that appears the most and outputs it.

N		Total time	**FC-resil. parts**	Unmod. parts
1	G	1551 (15.48)	1161 (14.01)	389.9 (7.760)
	E	1406 (17.02)	1003 (15.37)	402.5 (7.270)
10	G	4758 (37.90)	1673 (15.74)	3084 (34.50)
	E	4612 (42.04)	1417 (17.98)	3195 (33.63)
100	G	35320 (1229)	2279 (175.9)	33040 (1089)
	E	35180 (1216)	1073 (134.1)	34106 (1127)

Table 2: Performance times (ms) for our test implementation of FC-resilient authenticated garbling, computing N iterations of SHA-256. The average time over 10 runs is shown with the standard deviation in parentheses. Pre-processing and online times are combined.

government should interrupt the protocol execution either before the secure deletion or at the end of the protocol. In the first case, the garbler hasn't yet used her own input so it cannot be revealed, and in the second case the garbler herself cannot identify her own inputs or outputs without using $\mathcal{F}_{\text{pbkdf}}$. □

6.2.2 Implementation of FC-resilient 2PC

We implemented an FC-resilient two-party computation based on the authenticated garbling work of [83]. Our implementation was forked from `emp-toolkit` [84], and our source code can be found at this GitHub repository.[2]

The main part of our implementation was about 250 additional lines of code. The code contained two main changes: giving the output to the garbler rather than the evaluator, and implementing the the self-garbled tables shown in Table 1. The tables were created during function-dependent pre-processing, and accessed at the beginning and end of the online phase. The PBKDF used was Argon2i [7].

We emphasize that the added runtime is linear in the input/output wires, but is independent of the size of the circuit itself. To demonstrate this, we tested our implementation by running repeated iterations of SHA-256 while XORing the result with a "chaining" value as is done in computing PBKDF2 [68]. All experiments were performed on a Dell XPS laptop with an Intel i7-8650U processor and 16GB of RAM. The results are in Table 2; they show that the FC-resilience cost of running thousands of executions of a PBKDF (two per input wire, four per output wire) is costly for small circuits but quickly becomes negligible.

6.2.3 Constructing FC-resilient zero-knowledge proofs

ZKGC [36] is a zero knowledge proof of knowledge in which the verifier garbles a circuit and the prover evaluates the circuit using its witness as input. It follows from the Theorem 6.2.1 that ZKGC with self-garbled tables is FC-resilient for the verifier.

Corollary 6.2.2. *The ZKGC protocol combined with our self-garbled table construction is FC-resilient for the verifier.*

What about FC-resilience for the prover? Suppose Alice engages in an interactive zero-knowledge proof with Bob. Interactive zero-knowledge proofs are generally not transferable, from a cryptographic point of view. However, from a legal viewpoint, if the government wishes to investigate Alice, it can instruct Bob to disclose his interaction with her. Since Bob is not part of Alice's mind, he is part of Nature, and we presume that his testimony is truthful and can be added to the evidence E. Hence, the government can learn one bit about Alice based on testimony from Bob, so we believe that zero knowledge proofs cannot be FC-resilient for the prover.

[2] https://github.com/sarahscheffler/password-ag2pc

6.3 FC-resilient multi-party computation

Whereas the constructions in the last section only provided results to one party, in this section we describe a technique that permits everyone to receive the output of a large n-party secure computation, using ideas from differential privacy. This construction uses the BMR multi-party garbled circuit protocol [3], and it only achieves semi-honest security.

From an FC-resilience perspective, there are two challenges that occur when multiple parties receive output. First, we require a more complicated output opening protocol that requires all n parties to use their passwords in order to read the final result. In the semi-honest setting, the self-garbled no-op gates from the previous section solve this problem: each party masks the output table with a PBKDF of their password during garbling, and then each party in sequence can de-garble the final output wire at the end of the protocol.

Second, any party must operate under the assumption that the result of the computation can be compelled by the other participants, so she must ensure that the result reveals very little about any party's input. We propose to address this issue by considering MPC applied to differentially private functions. This comes at the expense of requiring a looser distinguishing bound when defining foregone conclusions, since differential privacy cannot achieve negligible statistical distance between neighboring distributions.

More formally, we present the following lemma to show that differentially-private functionalities are foregone if we assume this looser distinguishing bound. Recall that a function f is differentially private if for all "neighboring" inputs X_1 and X_2, for all possible subsets $S \subseteq im(f)$ we have $\Pr[f(X_1) \in S] \leq e^\epsilon \Pr[f(X_2) \in S]$, where ϵ is a parameter. In the context of MPC, inputs X_1 and X_2 are neighboring if they differ by the inclusion/exclusion of one party P_*'s input.

Lemma 6.3.1. *Let f be a differentially-private mechanism, X be a dataset in Nature, and $X' = X \cup \{R.s\}$. Consider the compelled action $C \coloneqq f(X')$ with the evidence E that checks for the existence of a secret input $R.s$. Then there exists a simulator S such that C is foregone with respect to E and S.*

Proof. Construct the simulator S^N that queries Nature to recover X and then returns $y \leftarrow f(X)$. Differential privacy guarantees that the two distributions $C^{N,R}$ and S^N are e^ϵ-statistically close (over the coins of f), as desired. □

From a scientific perspective, widening the distinguishing bound makes actions easier to compel, which has a two-sided impact on FC-resilience: the evidence gathered from voluntarily executing a cryptographic protocol might be used to compel more functionalities, but these functionalities may also have been compellable beforehand. The question then arises: what distinguishing bound is desired by the courts? This appears to be an open question: although the burden of proof to show that an action is a foregone conclusion is on the government, "how high that burden is remains surprisingly unclear" [39] from existing case law. Possible evidentiary standards to apply in foregone conclusion cases include the "reasonable suspicion" standard [63] that almost certainly does allow for noticeable chance of error, and the "beyond a reasonable doubt" standard that may not [82]. While we have taken the approach in this work that a negligible error rate is desirable since it suffices "to protect the innocent who otherwise might be ensnared by ambiguous circumstances" [57] and it yields an appealing composition property, our definition is easily extensible to any choice that courts decide as a policy decision.

7 Conclusion

This work initiates a scientific study of disclosures compelled by the U.S. government under the foregone conclusion doctrine. We provide a cryptographic security definition that is grounded in the law but that can be used by security researchers without the need to understand the law. We show that existing cryptosystems can be vulnerable to this threat, yet it is possible to design countermeasures at reasonable cost.

Beyond this paper's scientific contributions, this work also has significant bearing on a potential upcoming Supreme Court case. As we discuss in §4.1, state Supreme Courts and lower federal courts are divided on the issue of compelled decryption under the foregone conclusion doctrine. Legal scholars believe that the U.S.

Supreme Court will take one of these cases soon to resolve the issue [40]. For this case to come to a sound conclusion, the courts must analyze the foregone conclusion doctrine from many perspectives. We hope that the technical lens provided by this paper will shine new light on the doctrine that was not provided by prior legal analysis. The Supreme Court's decision will impact compelled decryption for the foreseeable future; we can only hope that the result is not already a foregone conclusion.

Acknowledgments

The authors thank Aloni Cohen, Sunoo Park, Andy Sellars, the Berkman Klein Center's Bridging Privacy group, and anonymous reviewers for their insightful comments, helpful conversations, and valuable feedback on this work. This material is supported by a Clare Boothe Luce Graduate Research Fellowship, a Google PhD Fellowship, the DARPA SIEVE program under Agreement No. HR00112020021, and the National Science Foundation under Grants No. 1414119, 1718135, 1739000, 1801564, 1915763, and 1931714.

References

[1] Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. Subversion-resilient signatures: Definitions, constructions and applications. *CCS*, 2015.

[2] Balthazar Bauer, Pooya Farshim, and Sogol Mazaheri. Combiners for backdoored random oracles. In *Annual International Cryptology Conference*, pages 272–302. Springer, 2018.

[3] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *22nd Annual ACM Symposium on Theory of Computing*, pages 503–513, 1990.

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

[5] Steven M. Bellovin, Matt Blaze, Sandy Clark, and Susan Landau. Lawful hacking: Using existing vulnerabilities for wiretapping on the internet. In *Privacy Legal Scholars Conference*, 2013.

[6] Bernstein v. US Dept. of State, 922 F. Supp. 1426 - Northern District of California 1996.

[7] Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. Argon2: new generation of memory-hard functions for password hashing and other applications. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 292–302. IEEE, 2016.

[8] Ernie Brickell. A proposal for balancing access and protection requirements from law enforcement, corporations, and individuals, August 2018.

[9] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001. 25 August 2019 version found at https://eprint.iacr.org/2000/067.

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

[11] Aloni Cohen and Kobbi Nissim. Towards formalizing the gdpr's notion of singling out. *CoRR*, abs/1904.06009, 2019.

[12] Aloni Cohen and Sunoo Park. Compelled decryption and the Fifth Amendment: Exploring the technical boundaries. *Harvard Journal of Law & Technology*, 32:169–234, 2018.

[13] Commonwealth v. Baust, 89 Va. Cir. 267 (2014).

[14] Commonwealth v. Davis, Pa: Supreme Court, Middle Dist. 2019.

[15] Commonwealth v. Gelfgatt, 468 Mass. 512, 11 N.E.3d 605, 11 N.E. (2014).

[16] Commonwealth v. Jones, 811 A. 2d 994 - Pa: Supreme Court 2002.

[17] Mark A Cowen. The act-of-production privilege post-Hubbell: United States v. Ponds and the relevance of the reasonable particularity and foregone conclusion doctrines. *Geo. Mason L. Rev.*, 17:863, 2009.

[18] Curcio v. United States, 354 U.S. 118 - Supreme Court 1957.

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

[20] Doe v. United States, 487 US 201 - Supreme Court 1988.

[21] Joan Feigenbaum and Daniel J. Weitzner. On the incommensurability of laws and technical mechanisms: Or, what cryptography can't do. In *26th International Security Protocols Workshop*, pages 266–279. Springer, 2018.

[22] Fisher v. United States, 425 US 391 - Supreme Court 1976.

[23] Jonathan Frankle, Sunoo Park, Daniel Shaar, Shafi Goldwasser, and Daniel J. Weitzner. Practical accountability of secret processes. In *27th USENIX Security Symposium*, pages 657–674. USENIX Association, 2018.

[24] Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten. In *EUROCRYPT (2)*, volume 12106 of *Lecture Notes in Computer Science*, pages 373–402. Springer, 2020.

[25] Gilbert v. California, 388 US 263 - Supreme Court 1967.

[26] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[27] Shafi Goldwasser and Sunoo Park. Public accountability vs. secret laws: Can they coexist?: A cryptographic proposal. In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, pages 99–110. ACM, 2017.

[28] Hiibel v. Sixth Judicial Dist. Court of Nevada, Humboldt County, 542 U.S. 177 - Supreme Court 2004.

[29] Thibaut Horel, Sunoo Park, Silas Richelson, and Vinod Vaikuntanathan. How to subvert backdoored encryption: security against adversaries that decrypt all ciphertexts. *arXiv preprint arXiv:1802.07381*, 2018.

[30] In re Grand Jury Proceedings, 41 F. 3d 377 - Court of Appeals, 8th Circuit 1994.

[31] In re Grand Jury Subpoena, 383 F. 3d 905 - Court of Appeals, 9th Circuit 2004.

[32] In re Grand Jury Subpoena Duces Tecum, 1 F. 3d 87 - Court of Appeals, 2nd Circuit 1993.

[33] In re Grand Jury Subpoena Duces Tecum Dated March 25, 2011 (United States v. Doe), 670 F.3d 1335 - 11th Circuit 2012.

[34] In re Grand Jury Subpoena to Sebasetien Boucher, No. 2:06-mJ-91, 2009 WL 424718.

[35] Joseph Jarone. An act of decryption doctrine: Clarifying the act of production doctrine's application to compelled decryption. *FIU L. Rev.*, 10:767, 2014.

[36] Marek Jawurek, Florian Kerschbaum, and Claudio Orlandi. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 955–966, 2013.

[37] Seny Kamara. Restructuring the NSA metadata program. In *Financial Cryptography and Data Security*, pages 235–247. Springer, 2014.

[38] Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 115–128. Springer, 2007.

[39] Orin S Kerr. Compelled decryption and the privilege against self-incrimination. *Tex. L. Rev.*, 97:767, 2018.

[40] Orin S Kerr. Decryption originalism: The lessons of Burr. *Available at SSRN*, 2020.

[41] Jeffrey Kiok. Missing the metaphor: Compulsory decryption and the fifth amendment. *Boston University Public Interest Law Journal*, 24:53–80, 2015.

[42] Joshua A. Kroll, Edward W. Felten, and Dan Boneh. Secure protocols for accountable warrant execution, 2014. https://www.cs.princeton.edu/~felten/warrant-paper.pdf.

[43] Yehuda Lindell. How to simulate it - A tutorial on the simulation proof technique. In *Tutorials on the Foundations of Cryptography.*, pages 277–346. Springer International Publishing, 2017.

[44] Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. *J. Cryptology*, 28(2):312–350, 2015.

[45] Matter of Residence in Oakland, California, 354 F. Supp. 3d 1010 - Dist. Court, ND California 2019.

[46] Nathan K. McGregor. The weak protection of strong encryption: Passwords, privacy, and Fifth Amendment privilege. *Vanderbilt Journal of Entertainment & Technology Law*, 12:581–609, 2010.

[47] National Association of Criminal Defense Lawyers. Compelled decryption primer, 2019. https://www.nacdl.org/Content/Compelled-Decryption-Primer.

[48] Jesper Buus Nielsen and Claudio Orlandi. LEGO for two-party secure computation. In *6th Theory of Cryptography Conference*, pages 368–386. Springer, 2009.

[49] Kobbi Nissim, Aaron Bembenek, Alexandra Wood, Mark Bun, Marco Gaboardi, Urs Gasser, David R. O'Brien, and Salil Vadhan. Bridging the gap between computer science and legal approaches to privacy. In *Harvard Journal of Law & Technology*, volume 31, pages 687–780, 2016 2018.

[50] Minerva Pinto. The future of the foregone conclusion doctrine and compelled decryption in the age of cloud computing. *Temp. Pol. & Civ. Rts. L. Rev.*, 25:223, 2016.

[51] Laurent Sacharoff. Unlocking the Fifth Amendment: Passwords and encrypted devices. *Fordham Law Review*, 87:203–251, 2018.

[52] Stefan Savage. Lawful device access without mass surveillance risk: A technical design discussion. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1761–1774. ACM, 2018.

[53] Schmerber v. California, 384 U.S. 757 - Supreme Court 1966.

[54] SEC v. Huang, 186 F. Supp. 3d 380 - Dist. Court, ED Pennsylvania 2016.

[55] Aaron Segal, Bryan Ford, and Joan Feigenbaum. Catching bandits and only bandits: Privacy-preserving intersection warrants for lawful surveillance. In *4th USENIX Workshop on Free and Open Communications on the Internet*. USENIX Association, 2014.

[56] Seo v. State, 109 N.E.3d 418 (Ind. Ct. App. 2018).

[57] Slochower v. Board of Higher Ed. of New York City, 350 US 551 - Supreme Court 1956.

[58] Matthew Smith and Matthew Green. A discussion of surveillance backdoors: Effectiveness, collateral damage and ethics, February 2016.

[59] State v. Andrews, 197 A. 3d 200 - NJ: Appellate Div. 2018.

[60] State v. Stahl, 206 So. 3d 124 (Fla. Dist. Ct. App. 2016).

[61] Paul Syverson, Roger Dingledine, and Nick Mathewson. Tor: The second-generation onion router. In *Usenix Security*, pages 303–320, 2004.

[62] Matt Tait. Going dark, crypto wars, and cryptographic safety valves, August 2018.

[63] Terry v. Ohio, 392 US 1 - Supreme Court 1968.

[64] Dan Terzian. The fifth amendment, encryption, and the forgotten state interest. *UCLA L. Rev. Discourse*, 61:298, 2013.

[65] Dan Terzian. The fifth amendment, encryption, and the forgotten state interest. *UCLA Law Review*, 61:298–312, 2014.

[66] Dan Terzian. Forced decryption as a foregone conclusion. *6 California Law Review Circuit 27*, 2015.

[67] The Law Library of Congress. Miranda warning equivalents abroad, 2016. https://www.loc.gov/law/help/miranda-warning-equivalents-abroad/miranda-warning-equivalents-abroad.pdf.

[68] Meltem Sönmez Turan, Elaine B Barker, William E Burr, and Lidong Chen. Sp 800-132. recommendation for password-based key derivation: Part 1: Storage applications, 2010.

[69] Nirvan Tyagi, Muhammad Haris Mughees, Thomas Ristenpart, and Ian Miers. Burnbox: Self-revocable encryption in a world of compelled access. In *USENIX Security Symposium*, pages 445–461. USENIX Association, 2018.

[70] Ullmann v. United States, 350 U.S. 422 - Supreme Court 1956.

[71] United States v. Doe, 465 US 605 - Supreme Court 1984.

[72] United States v. Hubbell, 530 US 27 - Supreme Court 2000.

[73] United States v. Kirschner, 823 F. Supp. 2d 665 - Eastern District of Michigan 2010.

[74] United States v. Spencer, No. 17-cr-00259-CRB-1 (N.D. Cal. Apr. 26, 2018).

[75] U.S. Constitution. Amend. V.

[76] US v. Apple MacPro Computer, 851 F.3d 238 - Court of Appeals, 3rd Circuit 2017.

[77] US v. Burns, Dist. Court, MD North Carolina 2019.

[78] US v. Fricosu, 841 F. Supp. 2d 1232 - Dist. Court, D. Colorado 2012.

[79] US v. Greenfield, 831 F. 3d 106 - Court of Appeals, 2nd Circuit 2016.

[80] US v. Maffei, Dist. Court, ND California 2019.

[81] US v. Ponds, 454 F. 3d 313 - Court of Appeals, Dist. of Columbia Circuit 2006.

[82] Victor v. Nebraska, 511 US 1 - Supreme Court 1994.

[83] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Authenticated garbling and efficient maliciously secure two-party computation. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 21–37, 2017.

[84] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Authenticated garbling and efficient maliciously secure two party computation, 2017. https://github.com/emp-toolkit/emp-ag2pc, last updated.

[85] Andrew T. Winkler. Password protection and self-incrimination: Applying the Fifth Amendment privilege in the technological era. *Rutgers Computer & Technology Law Journal*, 39:194–215, 2013.

[86] Timothy A Wiseman. Encryption, forced decryption, and the constitution. *ISJLP*, 11:525, 2015.

[87] Charles V. Wright and Mayank Varia. A cryptographic airbag for metadata: Protecting business records against unlimited search and seizure. In *8th USENIX Workshop on Free and Open Communications on the Internet*. USENIX Association, 2018.

[88] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science*, pages 162–167. IEEE, 1986.

A Formalizing our Model

In this appendix, we design a Turing machine-based model for the interactions between our parties. Then, we rigorously prove the sequential composition theorem.

A.1 Detailed ITM Model

In this section, we rigorously define each of the Turing machines present in the foregone conclusion game as well as the special tapes that they use to interface and interact with each other. This model follows the interactive Turing machine (ITM) model by Canetti [9], augmented with a few special symbols and random tapes that are specific to this work.

At a high level, we treat C, G, S, and R as Turing machines with a read-only randomness tape that is set to a uniformly random string at each invocation of the machine (and where C and G are interactive). E is a deterministic polynomial-time Turing machine that returns either **true** or **false**. We create a random string N that represents the state of the world besides R's mind; this string is exponentially long so that nobody (not even the government) can read all of it. Then, we allow R to read and modify N at a polynomial number of points. The evidence machine $E^N(R)$ restricts R's allowed changes to nature: it may require that R place certain values within N (e.g., if the government knows that a document already exists), and it can inspect the code of R to ensure that it contains certain methods and that those methods are performed correctly. While we lack a general way to check that the government's claimed evidence is valid, we require at least that the evidence be satisfiable: that is, at least one R will satisfy the evidence with high probability over the randomly chosen N and over R's random tape. Finally, we require that S^N be indistinguishable from a transcript of the interaction between G^N and $C^{N,R}$ for all R that satisfy the evidence with high probability.

In the remainder of this section, we describe the general properties of all participating Turing machines and then any extra properties of each individual machine.

General Turing machine model. For our purposes, all interactive Turing machines contain the following:

- Oracle access to one or two oracles (C, G, S, and R during the reading phase all have access to N, which is queried on indices, and C also has oracle access to R).
- A read-only random tape.
- An internal read/write working tape.

Respondent. R is a PPT Turing machine that represents the subpoena respondent's actions. R has an internal working tape, a write-only output tape, and a read-only randomness tape. Because R often acts as an oracle, we presume that it has methods whose existence and interface may be specified in the evidence. It receives these method calls on a read-only oracle-input tape and writes the results to a write-only oracle-output tape. R always has one special method called `Equivocate`, which is used near the beginning of the security game. When R.`Equivocate` is called, R outputs a set of locations and values Δ; N will be modified at these locations to be these values; i.e. setting $N[i] = x$ for each pair in Δ. E may mandate the existence of additional methods, or verify that the code of specific methods acts in a certain way (e.g. that an internal variable was generated from the correct distribution using the randomness tape).

Evidence. E is a Turing machine that restricts the ways in which R can modify N. It has oracle access to the modified N and takes the source code of R as input. E is deterministic and bounded to run in polynomial time. E can also force a concrete time bound T on R: It can automatically return **false** if R has not completed within time T. We will require two properties of the evidence, as described further in Def. 3.2.1: First, the evidence must be *satisfiable* (contrary to the usual computer science meaning of this term, we simply mean that there must exist at least one respondent for which E returns true with sufficient probability). Second, E must be *non-censoring* – E can only check polynomially many locations in N, and it must not prohibit R from modifying locations it has not checked.

Compelled request and government responder. C is the interactive Turing machine representing the action compelled by the subpoena, in the form of an interaction with a government agent G. The compelled request C has oracle access to N and R, whereas G only has oracle access to N. While a simple request could be done in a single round in which C outputs a message to G, this formalism also permits more complex requests which take multiple rounds.

Both C and G have the following tapes:

- A read-only identity tape
- A one-bit activation tape (determining whether it is currently in the process of "executing"
- A read-only (but externally-writable) input communication tape on which it receives messages
- A write-only output communication tape on which it sends messages

We denote the transcript between G^N and $C^{N,R}$ as $\tau(G^N, C^{N,R})$.

At all times, the input tape of C should match the output tape of G, and vice versa, and their switch bits should always be opposed. When finishing sending a message, the machine always sends a special eol character. We refer to $\tau(G^N, C^{N,R})$ as the *transcript* of G^N and $C^{N,R}$, consisting of a "log" of the messages sent between the two machines. This is formatted as an alternating string of messages between each machine's output tape, alternating to the other machine at each // character. This transcript is formatted as a concatenation of "id:message//" for each alternating message where id is either C or G, message is the message written on the output tape by id (received on the input tape of the opposite machine), and // is a reserved end-of-line character. At the end of the transcript (sent by whichever machine sends the last message) is a special □ symbol, reserved at the beginning of the computation. After receiving or sending this symbol, both machines halt. (A computation that consists of two separate computations concatenated together may use a different ■ symbol to denote the end of the separate computations.) As an example, if G sends a and then C sends b, completing the computation, the transcript τ is "G:a// C:b// □".

Finally, one should consider consider individual instances of C and G to be completely unrelated; in other words, they fully securely erase all of their state after the computation.

Government simulator. S is the government simulator. Its goal is to generate a result that is indistinguishable from the exchange between C and G. It is a PPT machine with oracle access to N (and no ability to query R). S has a single output tape in which it will attempt to simulate the transcript of a paired set of machines, but is not itself interactive. S is prohibited from returning $\perp$ to this output tape.

Composing Turing machines. Let M_1 and M_2 be Turing machines that, upon completion, use the computation-end symbol ■. Then let $M_1 \| M_2$ denote a machine with computation-end symbol □ that begins by running M_1, exchanging messages as normal and ending with ■. Then it starts computing M_2, exchanging messages as normal and also ending with ■. Then, since its computation has ended, it outputs □. In short, $M_1 \| M_2$ denotes the sequential composition of M_1 and M_2.

Pausing an interactive protocol. Given an interactive protocol Π, we wish to specify formally the state of nature N after t timesteps. This models a scenario where the protocol is interrupted by a request to compel information in the middle of the protocol. (After this interruption, we assume all parties abort and do not complete the protocol.)

Following the ITM model of Canetti [9], only one machine is deemed to be active in a given timestep. The active machine can write to or read from a tape, or move a head. This action could represent one step of local computation (writing/reading a local work tape) or communication to another party (by writing on that machine's communication input tape).

At any point in time, we consider the *configuration* of an interactive Turing machine to comprise the contents of all tapes (including input and output tapes), the current state, and the location of the head in each tape. We refer readers to [9, Def. 4] for more details.

Aside from the contents of P_*'s sub-module $\mathcal{F}$, the definition below puts the entire state of Π at the time of the government compelled actions exists in nature.

Definition A.1.1 (Protocol modifications to Nature)**.** Let Π be a protocol among parties $P_*, P_1, \ldots, P_n$, where P_* has a sub-module $\mathcal{F}$. Given $t \in \mathbb{N}$, let Π's *modifications to nature* M_t^{Π,P_*} be the set of all messages sent to and from all parties up through t total timesteps of computation (by all parties), the current configuration of all parties except P_* after t timesteps, and the current configuration of P_* except its input tape, its output tape, and its tapes communicating with sub-module $\mathcal{F}$ (and the tapes of $\mathcal{F}$ itself).

A.2 Proof of Sequential Composition

In this section, we prove the sequential composition theorem (Thm. 3.3.1). While composition generally follows naturally in simulation-based definitions, the proof in our setting is somewhat non-standard. For instance, proving composition for zero-knowledge proofs requires an auxiliary input so that later instances store the results of (simulated versions of) earlier instances, but our definition doesn't have a direct concept of auxiliary input. We proceed in the other direction: we can proactively store (simulated versions of) later instances in nature in order to test the limits of whether earlier instances are truly foregone conclusions.

We prove the two directions of Theorem 3.3.1 separately. Beforehand though, we find it useful to make a simple observation: the transcript of a composed machine is equivalent to the composition of the transcripts of the individual machines.

Lemma A.2.1 (Concatenation of composed transcripts)**.** *If M_{1a} and M_{1b} are paired interactive Turing machines with identities a and b respectively, and $M_{2a'}$ and $M_{2b'}$ are paired interactive Turing machines with identities a' and b', then*

$$\tau(M_{1a}, M_{1b}) \| \tau(M_{2a'}, M_{2b'}) \square = \tau(M_{1a} \| M_{2a'}, M_{1b} \| M_{2b'}).$$

Proof. This is a straightforward consequence of the sequential (i.e., non-parallelized) nature of a Turing machine and the way we defined Turing machine composition in §A.1. □

The first direction of the theorem states that compelling two foregone conclusions in a row is still a foregone conclusion.

Lemma A.2.2. *If C_1, G_1 is a foregone conclusion with respect to E, S_1, and C_2, G_2 is a foregone conclusion with respect to E, S_2, then $C_1 \| C_2$ is a foregone conclusion with respect to $E, S_1 \| S_2$.*

Proof. Since C_1, G_1 is a foregone conclusion with respect to S_1, E, we know that $\tau(G_1^N, C_1^{N,R}) \approx_c S_1 \; \forall R, \forall \mathcal{D}$ even with oracle access to N. This must include R that sent $\mathsf{NEnc}(\tau(G_2^N, C_2^{N,R}))$. ($R$ can generate this value by running the code of G_2 and C_2, reading from N and responding from its own code when appropriate.) Thus, $\tau(G_1^N, C_1^{N,R}) \| \tau(G_2^N, C_2^{N,R}) \approx_c S_1 \| \tau(G_2^N, C_2^{N,R}) \forall R, \forall \mathcal{D}^N$. By Lemma A.2.1 we therefore also have

$$\tau((G_1 \| G_2)^N, (C_1 \| C_2)^{N,R}) \approx_c S_1 \| \tau(G_2^N, C_2^{N,R}) \tag{1}$$

$\forall R, \forall \mathcal{D}^N$.

Separately, we know that since C_2, G_2 is a foregone conclusion with respect to S_2, E we know that $\tau(G_2^N, C_2^{N,R}) \approx_c S_2 \; \forall R, \forall \mathcal{D}$ even with oracle access to N. Since $\mathcal{D}$ can run the code of S_1, it must also be true that

$$S_1 \| \tau(G_2^N, C_2^{N,R}) \approx_c S_1 \| S_2 \tag{2}$$

$\forall R, \forall \mathcal{D}$.

Thus, combining equations 1 and 2, we have

$$\tau((G_1 \| G_2)^N, (C_1 \| C_2)^{N,R}) \approx_c S_1 \| \tau(G_2^N, C_2^{N,R}) \approx_c S_1 \| S_2$$

Thus, $\tau((G_1 \| G_2)^N, (C_1 \| C_2)^{N,R}) \approx_c S_1 \| S_2$, fulfilling the simulatability property of a foregone conclusion. In order to show that the composed version is a foregone conclusion, we must also show efficiency and satisfiability of evidence.

Clearly, if all of the component machines are efficient (guaranteed by the efficiency of machines in foregone conclusions), their concatenation must also be efficient. The evidence is the same for the composed and non-composed versions, so it maintains the satisfiability and non-censorship properties. This completes the argument. □

The second direction of the theorem states that if the composed request is a foregone conclusion, then each individual request must also be a foregone conclusion.

Lemma A.2.3. *Suppose C_1, G_1 is a foregone conclusion with respect to E, S_1. Suppose also $C_1 \| C_2$, $G_1 \| G_2$ is a foregone conclusion with regard to E, S_{12}, where S_{12} is S_1 concatenated with some S_2. Then C_2, G_2 is a foregone conclusion relative to E, S_2.*

Proof. Suppose by way of contradiction $\exists R, \mathcal{D}$ such that $\mathcal{D}$ is capable of distinguishing $\tau(G_2^N, C_2^{N,R})$ from S_2^N.

Then we can easily construct $\mathcal{D}'$ that can distinguish $\tau((G_1 \| G_2)^N, (C_1 \| C_2)^{N,R})$ from S_{12}^N for the same R. $\mathcal{D}'$ simply removes the first set of messages (before the ■ symbol of the interaction between G_1 and C_1) as well as the final □ symbol ending the composed interaction. The result is simply either $\tau(G_2^N, C_2^{N,R})$ or S_2^N. $\mathcal{D}'$ then calls $\mathcal{D}$ on this input and returns the same result. It is easy to see that $\mathcal{D}'$ can distinguish the composed computation with the same advantage as $\mathcal{D}$ can distinguish only the second interaction. (Notice also that by renaming the arguments, this proof also applies to the first compelled argument.) Thus, we have a contradiction; such a distinguisher cannot exist.

The other properties of a foregone conclusion also continue to hold: If the composed machines are efficient, then the decomposed machines must also be efficient. Furthermore, the same evidence is used in all the iterations, so the satisfiability and non-censorship properties of the evidence remain unaltered.

This completes the proof. If $C_1 \| C_2$, $G_1 \| G_2$ is a foregone conclusion with regard to E, S_{12}, then it must also be true that C_2, G_2 is a foregone conclusion relative to E, S_2. □

Finally, combining Lemmas A.2.2 and A.2.3 proves Theorem 3.3.1 and demonstrates that the government gains no advantage by waiting for the result of one foregone conclusion request before beginning the next.

B Security proof: FC-Resilient 2PC

In this section, we prove the two portions of Theorem 6.2.1 that claim the malicious security and FC-resilience of our secure two-party computation protocol Π in Figure 5.

B.1 Preliminaries

In this appendix we list some definitions and lemmas used in the proofs within this section.

Definition B.1.1 (Straightforwardly compellable)**.** We say a value x is *straightforwardly compellable* (or just *compellable*) under evidence E if there exists efficient C that makes no calls to R, and exists G such that $x \in \tau(C, G)$.

Lemma B.1.2. *Let X be a public distribution. Then $x \sim X$ is straightforwardly compellable under any E.*

If X is a distribution described at a public location in N, then $x \sim X$ is straightforwardly compellable under E that checks that X is described at the proper location in N.

Proof. For the first statement, C can simply sample its own fresh $x \sim X$. For the second statement, C can query N to get X, then sample $x \sim X$. (Oftentimes, X is a point mass.) □

Lemma B.1.3. *Let x be straightforwardly compellable under evidence E. Let C, G be such that $\tau(C^{N,R}, G^N) = x$ for all N and allowed R. Then there exist efficient C_0, G_0, and S, where C_0 does not query R, such that (C_0, G_0) is a foregone conclusion with respect to E and S.*

Proof. Notice that the first three properties of a foregone conclusion are trivial to achieve (aside from S efficiency, which we will come back to). There exist efficient C_0 and G_0 by Lemma B.1.2. The evidence is the same, so the existence of an α-allowed R, and the non-censorship property of E, are unaltered.

S emulates the execution of C_0 and G_0 and outputs the result.

Thus, (C_0, G_0) is a foregone conclusion with respect to E and S. □

Lemma B.1.4. *Let X be a distribution that is computationally indistinguishable from a public distribution Y. Then there exists C, G, S such that $\tau(C^{N,R}, G^N)$ contains a fresh sample from distribution X and S returns a fresh sample from Y. Then (C, G) is a foregone conclusion with respect to to S and any evidence E.*

Proof. We know that sampling a fresh $y \sim Y$ is straightforwardly compellable under any evidence E. Let (C, G) be a foregone conclusion with respect to E and a simulator S, where S samples $y \sim Y$ and outputs the result.

Consider probabilistic poly-time distinguisher D attempting to distinguish X from Y. Observe that if $x \sim X$ is *not* a foregone conclusion with respect to E and S, then D can distinguish X from Y. Thus, compelling a fresh sample x from C, G is a foregone conclusion with respect to E and S. □

B.2 Malicious security for both parties

Our first goal is to ensure that this protocol retains malicious security against (separately) corrupt G and V. Informally, we will show that if this protocol does not have this property, then we can break either the PRF called by $\mathcal{F}_{\text{pbkdf}}$, or the security of the 2PC protocol described in [44].

Theorem B.2.1. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Let Π be an instantiation of our protocol (Figure 5) for f. Assume that the oblivious transfer protocol is secure, that we have a perfectly-binding commitment scheme (used by the garbler) and a perfectly-hiding commitment scheme (used by the evaluator for coin-tossing), and that $\mathcal{F}_{pbkdf}$ generates a password with good min-entropy and contains a good pseudorandom function. Then, Π securely computes f.*

Proof of Theorem B.2.1. We prove this via a series of hybrids that show that an execution of this protocol (using circuit C_2) is indistinguishable from an execution of the protocol of [44], which we know securely computes f under the same assumptions:

H_0 The real protocol described above

H_1 No permuted/masked tables – follow the protocol through step 6 (decommitment for check-circuits), but then skip steps 7, 8, 9a, and 9b. Instead, keep the original garbled tables and go straight to step 9c.

H_2 Same as H_1, but instead of sampling the outputs of the no-op gate $w_{c,i,b}$ from $\mathcal{F}_{\text{pbkdf}}$, G instead chooses them as normal wire labels (random values).

H_3 The ideal functionality for a 2PC computing $f(x, y)$.

It is easy to see that $\mathsf{H}_0 \equiv \mathsf{H}_1$: Observe that the messages sent in each protocol are exactly the same; the only changes are to G's local state. G clearly cannot learn anything new about V's input in H_0 that it couldn't learn in H_1, since all the extra computation happens on information G already had. Since the distributions of H_0 and H_1 are equal for all x and y, no distinguisher can tell which game we are playing.

We next show that $\mathsf{H}_1 \approx_c \mathsf{H}_2$. Let $\mathcal{A}$ be an adversary attempting to distinguish $\mathcal{F}_{\text{pbkdf}}$ from random. Suppose it has access to $\mathcal{A}'$ which can distinguish H_1 from H_2. $\mathcal{A}$ first calls $\mathcal{F}_{\text{pbkdf}}(\mathsf{setup}, 2\lambda)$. It then runs an entire execution of the protocol in Figure 5 and inputs the transcript into $\mathcal{A}'$. When it computes the wire labels for the output wire, it sets $w_{c,\mathsf{out},b} = \mathcal{F}_{\text{pbkdf}}(\mathsf{query}, \mathsf{salt}, (c, i, b))$ for $c \in [s]$, $i \in [n]$, and $b \in \{0, 1\}$. When $\mathcal{A}'$ outputs its guess as to which hybrid it is in, $\mathcal{A}$ guesses that it is interacting with a random function if $\mathcal{A}'$ guessed H_2, and a pseudorandom function if it guessed H_1. It is not hard to see that $\mathcal{A}$ will have the same advantage as $\mathcal{A}'$. Thus, the security of $\mathcal{F}_{\text{pbkdf}}$ ensures that H_1 is indistinguishable from H_2.

Finally, to show that $\mathsf{H}_2 \approx_c \mathsf{H}_3$, we rely on the original security proof of [44]. Observe that H_2 is the protocol of [44]. We can use C_2 as the auxiliary input circuit (called C_0 in that paper) to that protocol, since it computes $f(x, y)$, The proof of security follows directly, for both a malicious garbler and a malicious evaluator. □

B.3 FC-resilience for the garbler

Finally, we wish to show that the protocol in Figure 5 is FC-resilient for the garbler.

Theorem B.3.1. *Assuming secure deletion for the garbler G, and assuming the existence of a channel through which the parties can halt the computation if any of them is compelled, then the protocol in Figure 5 is FC-resilient for G.*

Proof of Theorem B.3.1. Let Π be an instantiation of the protocol in Figure 5, and suppose it runs on inputs x and y for parties garbler G and evaluator V respectively. Let C, G be a compelled request, let E be an evidence machine that checks the auxiliary inputs of Π and V's input, and let S be a simulator. Suppose the compelled request is made after $t \in \mathbb{N}$ timesteps of computation of Π, and let $M_t^{\Pi,G}$ be the additions to Nature by timestep t, as in Definition A.1.1. Let E_t check $M_t^{\Pi,G}$ and then run E. Say (C, G) is a (λ, α)-foregone conclusion with evidence E_t and simulator S.

We will prove the claim by showing that all elements of $M_t^{\Pi,G}$ are straightforwardly compellable under E.

First, notice that the satisfiability and non-censoring properties are met: E checks a subset of the locations in N compared to E_t, so all α-allowed R from the post-protocol foregone conclusion are still α-allowed under E. And, by construction, E_t is non-censoring if and only if E is.

The remaining goal is to show that all elements of $M_t^{\Pi,G}$ are straightforwardly compellable. This will complete the proof, by Lemma B.1.3 and by our composition theorem (Theorem 3.3.1).

We can lean on Lemma B.1.2 to show this, and we can instead show that all values in $M_t^{\Pi,G}$ were generated from known distributions or could be generated from Nature.

Essentially, this means we must "simulate" all values added to $M_t^{\Pi,G}$ starting from only the auxiliary inputs and the other parties' inputs, and show that they could have been simulated *before* running the protocol. This will prove the claim, since this is the only additional information that the evidence E and simulator S can work with. Consider two cases, depending on t.

Case 1: The compelled request occurs before step 9. In this case, it is Pareto-optimal for the request to occur immediately before step 8. It is always better for the government for information to be added to N, and until step 8, $M_t^{\Pi,G}$ only grows as t increases. Only during step 8 is G's local state erased, so $M_t^{\Pi,G}$ is smaller for t in step 8 than it is for t immediately before step 8.

We proceed to demonstrate that we can simulate these values even before running the protocol. At t immediately before step 8, the information in $M_t^{\Pi,G}$ is:

- The auxiliary inputs for both parties and V's input. These exist before the protocol and were checked by E, and are compellable by Lemma B.1.2.
- Preprocessing phase (steps 0, 1, 2a): All values generated by G during these steps are ephemeral random values with known distributions and are compellable by Lemma B.1.2).
- Preprocessing phase (step 2b): The output wire labels are the output of a PRF with an ephemeral "key" (actually salt). By Lemma 5.4.1, compelling the wire labels is foregone.
- Evaluation Part 1 phase (steps 3-6): All new additions to $M_t^{\Pi,G}$ in this phase are either new ephemeral random values, or a function of straightforwardly compellable information.
- Preparation phase (step 7a): As discussed in §5, the results of the calling the PRF in step 7a straightforwardly compellable.
- Preparation phase (steps 7b, 7c): Step 7b involves only fresh random samples, and step 7c is a function of existing values.
- All information held by V. This is all a function of V's input and the messages sent to V (which already showed was compellable).

As mentioned, running step 8 will only limit the government's ability to create a foregone conclusion. Thus, for t before step 9 of Π, all values in $M_t^{\Pi,G}$ are straightforwardly compellable.

Case 2: The compelled request occurs during or after step 9.

If the compelled request occurs during or after step 9, then it is Pareto-optimal for the request to occur at the end of the protocol, for the same reasoning as in the previous case. We proceed to demonstrate how to simulate Part 2 of the evaluation. In this phase, we must also deal with G's input x, which is *not* efficiently compellable (though it also is not directly part of $M_t^{\Pi,G}$).

- Step 9: Now that the mappings of wire labels to values have been deleted, G has in effect "garbled" her own state so that simply writing m_{c,i,x_i} or w_{c,i,x_i} (or t_{c,i,x_i}) does not reveal x_i itself. These values cannot be compelled by indexing x_i (since the government does not know G's secret x_i), so the best the government can do in the existing C, G is compel one of these values indexed by an arbitrarily chosen bit b (unless E itself contains x_i). By Lemma B.1.4, these values are compellable. Furthermore, all values computed in the decommitments are functions of values that were straightforwardly compellable.
- Step 10, 11a: All this is done by V, and thus is in N either way.
- Step 11b: First, note that the $w_{c,\mathsf{out},b}$ value was already compellable. Second, recall that the output tapes of $\mathcal{F}_{\text{pbkdf}}$ (and G's final output tape) are not included in $M_t^{\Pi,G}$. Thus, all the information in this step was already compellable.

So, for t after step 8 of Π, all values in $M_t^{\Pi,G}$ are compellable.

Thus, by Lemma B.1.3 and the Sequential Composition Theorem (3.3.1), we know that if (C, G) was a foregone conclusion with respect to E_t and S and the request occurred after step 8 of the protocol, we can create machines C_0, G_0, S_0 such that $\tau({C_0}^{N,R}, G^N) = \tau(C^{N,R}, G^N)$ for all N, R and (C_0, G_0) is a foregone conclusion with respect to E and S_0. This shows that the protocol Π in Figure 5 is FC-resilient for the garbler and completes the proof. □