

Pre-Constrained Encryption

Prabhanjan Ananth ✉

University of California Santa Barbara, CA, USA

Abhishek Jain ✉

Johns Hopkins University, Baltimore, MD, USA

Zhengzhong Jin ✉

Johns Hopkins University, Baltimore, MD, USA

Giulio Malavolta ✉

Max Planck Institute for Security and Privacy, Bochum, Germany

Abstract

In all existing encryption systems, the owner of the master secret key has the ability to decrypt *all* ciphertexts. In this work, we propose a new notion of *pre-constrained encryption* (PCE) where the owner of the master secret key does not have “full” decryption power. Instead, its decryption power is constrained in a pre-specified manner *during the system setup*.

We present formal definitions and constructions of PCE, and discuss societal applications and implications to some well-studied cryptographic primitives.

2012 ACM Subject Classification Security and privacy → Cryptography; Security and privacy → Public key encryption

Keywords and phrases Advanced encryption systems

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.4

Funding *Abhishek Jain*: Supported in part by NSF CNS-1814919, NSF CAREER 1942789, Johns Hopkins University Catalyst award and Office of Naval Research Grant N00014-19-1-2294

Zhengzhong Jin: Supported in part by NSF CNS-1814919, NSF CAREER 1942789, Johns Hopkins University Catalyst award and NSF CAREER 1845349.

1 Introduction

All encryption systems involve a setup procedure for generating an encryption key and a corresponding master secret key. A ubiquitous property of all such systems is that the setup authority – who holds the master secret key – can decrypt *all* ciphertexts computed using the encryption key. This is true both for classical encryption schemes as well as advanced encryption schemes such as identity-based encryption [43, 12], attribute-based encryption [41, 31] and functional encryption [41, 14, 37].

In advanced encryption systems, this property is referred to as the key-escrow problem [40]. This means that users who participate in the system must trust the setup authority to act in good faith and not do anything untoward such as spy on their communication or analyze their encrypted data. This strong trust requirement can be problematic in many scenarios, and potentially be a deal-breaker in the adoption of such systems.

In this work, we ask the following question:

Is it possible to achieve meaningful privacy guarantees even against the setup authority who holds the master secret key?

Reducing Trust in Setup Authority. The problem of key escrow in advanced encryption systems is well documented in the cryptography literature and several models have been studied over the years for reducing trust in the setup authority. Some involve distributing



© Prabhanjan Ananth, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta;
licensed under Creative Commons License CC-BY 4.0

13th Innovations in Theoretical Computer Science Conference (ITCS 2022).

Editor: Mark Braverman; Article No. 4; pp. 4:1–4:20



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

4:2 Pre-Constrained Encryption

the trust among multiple authorities [12, 18, 36], while others replace the setup authority with a (public) registration mechanism [23, 24, 28]. Yet another line of work focuses on detecting misbehavior by the authority [29, 30, 8].

These models offer different trade-offs between functionality and security. However, none of them are ideally suited to our envisioned applications discussed below (see Section 1.2 for a detailed discussion).

Our Work. In this work, we propose a new model for encryption schemes where the setup authority does not have full decryption power. Instead, its decryption power is restricted in a pre-specified manner. To motivate our work, we find it instructive to consider the following examples.

Scenario 1: Spam Detection. Suppose that a group of users want to allow their email service provider (say, Gmail) to perform spam detection on their emails. They are data-privacy conscious and want to make sure that Gmail cannot learn anything about the content of the emails, except whether or not it is spam. A potential approach based on functional encryption suggested in prior works is for each user to issue a key to Gmail that only allows for computing the spam detection algorithm, and nothing else. However, such an approach requires Gmail to share its (private) algorithm with all users, which is clearly undesirable.

Scenario 2: Law-Enforcement. Today, users regularly store their private data on online platforms such as chat messages on WhatsApp and Apple iMessage, and documents and media files on Dropbox, Apple iCloud and other similar providers. To increase transparency and user trust, these platforms use end-to-end encryption (E2E) systems to store user data. Consider the scenario where upon receiving some credible intelligence, a law-enforcement agency (say, FBI) wants to gain access to private data of some users. For example, it may want to perform a keyword search on the chat contents of some suspect individuals (such as those on the “no-fly list”) to check for any trace of communication with known criminals, or other red flags [3]. Alternatively, it may want to detect (and remove) abusive material that can cause social harm [2]. In either case, giving complete access (i.e., decryption keys of all users) to FBI is not an acceptable solution as we have seen in the past how authorities have abused their power and conducted mass surveillance [1]. This raises the important question whether it is possible to strike a balance – provide restricted access to FBI to exercise law and order but at the same time prevent potential abuse? Note that naive solutions based on FE do not work for similar reasons as in the first scenario discussed above.

In both of these scenarios, a convenient solution approach would be to maintain an encrypted copy of the user data (emails or chat contents, etc) under the public key of the respective authority – Gmail or FBI – such that the authority can use its secret key to run the necessary computation – spam detection or check for criminal activity – on the encrypted data. To prevent abuse, we crucially require that *the authority – who holds the master secret – can only perform such “authorized” computations, and nothing else.*

The main focus of this work is to define and construct such encryption systems.

Pre-Constrained Encryption. We propose a new model for reducing the trust in the setup authority. Our main idea is to constrain the decryption power of the setup authority during the system setup. We proceed to describe the key aspects of our model:

- *Syntax:* A *pre-constrained encryption* (PCE) scheme (Setup, KeyGen, Enc, Dec) is defined w.r.t. a constraint family \mathcal{C} and a function family \mathcal{F} . Roughly speaking, the constraint

family \mathcal{C} consists of boolean predicates $C \in \mathcal{C}$ that are used to model what kind of decryption capabilities the authority can possess after performing the system setup. The function family \mathcal{F} captures the universe of functions $f \in \mathcal{F}$ that can potentially be computed on encrypted messages.

During system setup, the authority receives a constraint $C \in \mathcal{C}$ as input and uses it to generate a public key PK and a constrained master secret key $\text{MSK}[C]$. The encryption algorithm takes as input the public key PK and a message x , and outputs a ciphertext CT . Finally, the decryption algorithm takes as input a ciphertext CT together with the constrained master secret key $\text{MSK}[C]$ and a function $f \in \mathcal{F}$ and outputs a value y .

- *Security against Authority:* Our goal is to allow the authority to compute “authorized” functions $f \in \mathcal{F}$ over encrypted messages. To model this, we require that given an encryption of a message x , even the authority can only compute $f(x)$ for all functions f s.t. $C(f) = 1$, and *nothing else*.

We consider security against *semi-honest* authorities who perform the setup honestly as well as *malicious* authorities who may use any adversarial strategy for computing the system parameters. While some societal applications may demand security against malicious authorities, we will later see that security against semi-honest authority is already sufficient for some cryptographic applications.

- *Constraint Hiding:* We also consider a *constraint hiding* property for PCE, which requires that the public key generated during the system setup does not leak any information about the constraint predicate to the users.

In Section 4, we provide a formal modeling of PCE. Similar to the literature on advanced encryption schemes, we explore PCE along two additional dimensions:

- First, we consider a *key delegation* feature wherein the master secret key can be used to derive decryption keys sk_f for authorized functions f . Specifically, we equip PCE with an additional key generation algorithm that takes as input $\text{MSK}[C]$ and a function $f \in \mathcal{F}$ such that $C(f) = 1$ and outputs a secret key sk_f .
- Second, we consider two special cases of PCE – *identity-based PCE* (IB-PCE) and *attribute-based PCE* (AB-PCE). Roughly speaking, our notion of IB-PCE allows for full decryption of ciphertexts computed w.r.t. authorized identities. For example, the authorized identities could capture people on a “no-fly list” in the aforementioned application. More generally, IB-PCE is useful for scenarios where we wish to allow an authority to be able to fully decrypt ciphertexts after an *exact match* of identities (which may capture real-world identities, keywords, etc). Our notion of AB-PCE extends this idea to allow for more complex decryption criterion (as opposed to exact match).

We believe that IB-PCE and AB-PCE are independently interesting and can serve as important milestones in a general study of PCE. As an example, we describe an application of IB-PCE to *coercion-resistant signatures* in Section 2.

Discussion. PCE is not meant to be used as a substitute for end-to-end encryption in the aforementioned application scenarios. Indeed, PCE – by design – requires that some ciphertexts cannot be decrypted even given the master secret key. Thus, if used in a stand-alone manner, it would prevent regular usage of the application (such as chat/email) by honest users. To achieve the best-of-both-worlds, PCE can be used *in conjunction* to E2E by maintaining two encrypted copies of user data – one under E2E and another under PCE key. Regular usage of the system only requires the first encrypted copy, while the concerned

authority (law-enforcement or Gmail in the above examples) only needs to use the second encrypted copy.¹

1.1 Our Results

In this work, we initiate the study of pre-constrained cryptosystems where the master key is not omnipotent. Specifically, we propose the notion of pre-constrained encryption, provide formal definitions and new constructions, and show implications to other powerful cryptographic primitives. Below, we elaborate upon our results.

I. Definitions and Compilers. Our first contribution is a formal treatment of PCE. We define two key properties for PCE – *security against authority (SAA)* and *constraint hiding*.

- Following the literature on secure computation, we define SAA against semi-honest (SH) adversaries and malicious (M) adversaries as well as the intermediate case of semi-malicious² (SM) adversaries.

Following the standard indistinguishability-based security formulation for encryption schemes, SH-SAA requires that for any two messages x_0 and x_1 s.t. $f(x_0) = f(x_1)$ for all functions f where $C(f) = 1$ and C is the input constraint, encryptions of x_0 and x_1 are indistinguishable to the authority who holds the master secret key and the setup randomness. Extending this notion to M-SAA requires care since a malicious adversary can ignore the input constraint and use some arbitrary mechanism to compute the public key. Our formulation requires the existence of an inefficient “constraint extractor” and guarantees that the adversary embeds a constraint C from the constraint family \mathcal{C} in any “well-formed” public key.

- We next present generic transformations from SM-SAA to M-SAA security. Typically, such transformations (e.g., in the secure computation literature) make use of zero-knowledge proofs [27] that either require interaction or an initial setup. Our transformations do not require any setup assumption and work in the *plain model*.

More specifically, we first present such transformations for IB-PCE and AB-PCE by relying on non-interactive witness-indistinguishable (NIWI) proofs. Such proof systems are known from standard assumptions on bilinear maps [33]. We next present a similar transformation for general PCE by additionally relying on 2-server homomorphic secret-sharing (HSS) schemes [15] with perfect correctness. Such HSS schemes can be built from sub-exponentially secure indistinguishability obfuscation (iO) [9, 21].

II. Constructions. We present new constructions for PCE that achieve security against semi-malicious authorities as well as the constraint-hiding property. To obtain some of these results, we leverage a connection between our notions and proofs of *selective security* in prior works on advanced encryption schemes.

- *Point Constraints:* We first construct an AB-PCE scheme for *point-constraints*, i.e., constraint circuits C_{att^*} that contain a point hardwired in their description and authorize a function f iff $f(\text{att}^*) = 0$. Our construction is based on the learning with errors (LWE) assumption and builds on the selectively-secure ABE scheme of Boneh et al. [13]. The

¹ Here, we assume that the parties use “honest” client software to ensure that both the encryptions contain the same message. To address malicious clients, one could augment the system with non-interactive zero-knowledge proofs [19, 11, 20] in the common random string model in a standard way. Random string setups can be heuristically instantiated in practice without trusted parties.

² Semi-malicious adversaries [7] behave honestly like semi-honest adversaries but may use arbitrarily chosen randomness.

key insight behind this construction is to turn their proof of security into an AB-PCE scheme.

- *General Constraints:* We next consider the case of *general constraints*. We first present an IB-PCE scheme for general circuit constraints from the learning with errors (LWE) assumption. We obtain this result by observing that the construction of “attribute-based secure functional evaluation with public reconstruction” – a new primitive recently defined and constructed in [6] in a different context – can be re-purposed as an IB-PCE scheme. To go beyond IB-PCE, we make use of stronger cryptographic assumptions. We construct an AB-PCE scheme for general circuit constraints from witness encryption (WE) [22] and non-interactive zero-knowledge (NIZK) proofs with perfect soundness. We also present a PCE scheme for general circuit constraints from iO and NIZKs with perfect soundness.

III. Implications to Other Primitives. We show that the use of WE and iO in our constructions of AB-PCE and PCE for general circuit constraints, respectively, is in fact inherent. Specifically, we show that AB-PCE for general circuit constraints implies WE for NP and similarly, PCE for general circuit constraints implies iO for P/Poly.

Two important remarks are in order: first, these transformations do *not* rely on the constraint hiding property. Second, these transformations do *not* incur sub-exponential security loss, unlike previously known transformations from FE to iO [5, 10].

1.2 Related Work

Prior “Pre-Constrained” Systems. PCE is related to the notion of non-interactive secure computation (NISC) [35]. For constraint families \mathcal{C} where a constraint $C \in \mathcal{C}$ authorizes a *single* function $f \in \mathcal{F}$, PCE with SH-SAA security can be obtained from NISC. Unlike NISC, however, PCE can also support constraint families where a constraint can authorize a large (potentially exponential-sized) class of functions. In this sense, PCE is stronger than NISC.

The idea of *pre-constraining* the system parameters can be seen as implicit in prior work on lossy trapdoor functions and its generalizations [38]. In these notions, the setup can be performed in a “special mode” to obtain keys such that encryption with these keys results in information loss. Furthermore, this mode is indistinguishable from the “normal mode” where injectivity is maintained. Over the years, such notions have found widespread applications in cryptography (see [39] and references therein). Our notion of PCE allows for pre-specifying the decryption ability within a single mode, and can be viewed as a generalization of this prior line of research.

PCE vs Functional Encryption. PCE is closely related to the notion of functional encryption (FE). PCE is meaningful and seemingly non-trivial to achieve even without key delegation feature. In contrast, FE without key delegation collapses to standard encryption. As such, PCE without key delegation does not imply FE. In Section 4, we define PCE with an additional *collusion-resistance* security property where an adversary who is provided keys for functions f_1, \dots, f_q cannot distinguish between encryptions of two messages x_0 and x_1 (provided that each f_i is functionally equivalent on x_0 and x_1). It is easy to see that PCE with collusion-resistance implies FE.

FE and its weaker avatars such as ABE and IBE can be viewed as *post-constrained* encryption systems since constrained keys obtained via key delegation can be computed only *after* first computing a fully functional master secret key. Indeed, this is precisely why such systems do not provide any security whatsoever against the setup authority unless one considers unrealistic models where the authority simply “forgets” the master secret key and

the randomness used for setup.³ In contrast, PCE allows for *directly* computing a constrained master secret that can allow for a large set of computations. This is the key distinction between PCE and FE.

Exceptional Access to Law-Enforcement. Some recent works [42, 44, 32] have investigated the problem of providing exceptional access to private user data to law enforcement. The work of [44] takes a cost-based approach to this problem and proposes a solution assuming that the government has strictly more computing power than other individuals.

The approach taken in the work of Green et al. [32] is closer to ours: they build abuse-resistant access systems in different models that rely on different cryptographic tools ranging from NISC or lossy trapdoor functions to extractable witness encryption [22, 26]. Their solution based on lossy trapdoor functions (or NISC) can be viewed as implicitly constructing an IB-PCE scheme where the constraint authorizes an a priori (polynomially) bounded number of identities and the size of public parameter grows linearly with the number of authorized identities. Our work provides a general framework towards constructing such systems via the notion of PCE.

Prior Models for Reducing Trust. The problem of reducing trust in the setup authority in advanced encryption systems has been widely studied. Here, we provide a brief summary of some of the models.

- *Multiple authorities:* Boneh and Franklin [12] suggested the use of *multiple* authorities to distribute the trust assumption and prevent a single point of failure. Subsequently, this direction has been explored in many works, both in the context of IBE and ABE (see, e.g., [18, 36]). As discussed in [29], such a model might not always be convenient for users in advanced encryption systems since they must prove their credentials to multiple authorities to request keys. In terms of privacy, naturally, this notion does not provide any guarantees when only a *single* authority is available (or all authorities are corrupted).
- *Accountable authority:* Goyal [29] considered an alternative model where malicious behavior by the setup authority – such as distribution of “decryption boxes” to unauthorized users – can be detected. Further strengthenings of this model were subsequently studied; see e.g., [30]. We note that while this model provides for misbehavior *detection*, it does not *prevent* the authority from potentially decrypting all the ciphertexts generated by the system users. The work of Badrinarayanan et al. [8] considers a different notion of verifiable FE where correctness guarantees hold even when the setup authority is malicious. Their work, however, does not consider privacy guarantees against malicious authorities.
- *Registration-based systems:* Finally, we mention the recent line of work on registration-based encryption (RBE) [23, 24, 28]. In RBE, the authority is replaced by a key curator who does not issue any keys. Instead, the parties self-register, and the key curator aggregates the public-keys into a compact form. Consequently, the system requires periodic updates to ensure correct functioning. This notion has been proposed as an alternative to IBE, and does not naturally generalize to ABE or FE. While the original work of [23] assumed the key curator to be *trusted*, a recent work of [28] proposes some mitigations in the common reference string model.

None of these models seem ideally suited to some of our envisioned applications of PCE (discussed in Section 1). Multi-authority models do not provide any security when all the authorities are dishonest while PCE aims for meaningful security even in “full corruption”

³ Note that this is not even compliant with semi-honest security.

scenario. The registration-based model [23] goes to the other extreme in that the authority has no decryption power at all, while PCE requires partial decryption power in order to enable authorized computations.

2 Technical Overview

In what follows we give a cursory overview of the technical content of our work. We begin by presenting some pre-constrained encryption schemes with progressively more expressive functionality, then we discuss generic transformations to upgrade the security notion and relations to other well-known cryptographic primitives.

Throughout this section, we consider PCE with the key delegation feature by default.

2.1 Constructions

We discuss the intuition behind the constructions of pre-constrained encryption schemes. Throughout this section, we always consider security against a *semi-malicious* authority: The authority is assumed to run the setup honestly, but can choose the random coins of the algorithm arbitrarily. We will see later how to lift this notion to the more challenging setting where the authority can behave maliciously.

Prelude: Turning Proofs Into Constructions. Before delving into the description of specific schemes, we build up some informal intuition by presenting our general construction template, based on *selective security proofs* for advanced cryptographic systems. A commonly adopted proof technique to argue selective security is the following: the reduction cleverly embeds the instance of a hard problem into the cryptographic scheme in such a way that if an adversary distinguishes encryptions of two challenge messages, then it can be used to violate the hardness assumption. The reduction is responsible for generating the public key and the challenge ciphertext and answering any decryption key queries made by the adversary. To enable the aforementioned embedding process, the reduction typically first generates *constrained* public parameters⁴ that would *not* allow for issuing decryption keys of a specific kind – in particular, those that would allow for breaking the security of the challenge ciphertext. This way, if the adversary manages to break the security of the scheme, then it must be the case that it also solved the hard problem.

This observation paves the way for a natural construction of pre-constrained encryption:

- The setup of pre-constrained encryption runs the constraining process of the public key, performed by the reduction above.
- The key generation procedure is the procedure used by the reduction to answer the queries of the adversary.
- The encryption and the decryption procedures are the same as the underlying selectively secure scheme.

While this is our general template, achieving pre-constrained encryption for general classes of constraints will require more work. In particular, in our actual constructions we will use selectively secure encryption schemes in a non-blackbox way.

Example: The Boneh et al. ABE. To make our discussion more concrete, consider the ABE scheme of Boneh et al. [13]:

⁴ We remark that not all selective security proofs follow this approach. For example, the security proof in the IBE scheme of [12] does not constrain the public key.

- The public keys consist of the matrices $\{\mathbf{A}_i\}_{i \in [\ell]}$ along with another matrix \mathbf{A} , where ℓ is the length of the attribute. All the matrices are of the dimension $n \times m$ (n is set to be security parameter and $m = O(n \log(q))$) and over a large field \mathbb{Z}_q . The matrix \mathbf{A} is generated along with its trapdoor T_A such that $\mathbf{A}T_A = 0$.
 - The ciphertext consists of LWE samples encoding the attribute att and the secret message μ .
 - To generate a key with respect to f , do the following: Using the trapdoor of \mathbf{A} , generate a matrix \mathbf{X}_f such that \mathbf{X}_f is a trapdoor for the matrix $[\mathbf{A}|\mathbf{A}_f + \mathbf{G}]$, where \mathbf{G} is the gadget matrix and \mathbf{A}_f is generated using the following algorithm: $\mathbf{A}_f = \text{EvalPK}(f, \{\mathbf{A}_i\}_{i \in [\ell]})$
- In the proof of security, the matrices $\{\mathbf{A}_i\}_{i \in [\ell]}$ are punctured with respect to the challenge attribute att^* . That is, set $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i - \text{att}_i^* \cdot \mathbf{G}$. Then, the attribute keys are generated using $\{\mathbf{R}_i\}$ and the trapdoor of the gadget matrix \mathbf{G} . At this point, we can invoke LWE to switch the LWE samples in the ciphertext to be uniformly random.

Attribute-Based PCE for Point Constraints. We now turn our attention to constructing attribute-based PCE (AB-PCE) schemes. An AB-PCE scheme follows a similar syntax as PCE except that the encryption algorithm now additionally takes as input an attribute att (from an attribute universe). We require that the authority can decrypt a ciphertext associated with an attribute att only if there exists an authorized predicate f (i.e., $C(f) = 1$) such that $f(\text{att}) = 1$.

We show how to turn the above ABE scheme into a AB-PCE scheme (with key delegation) for the family of *point constraints*, namely, constraints restricted to be of the form $C_{\text{att}^*}(f) = 1 \iff f(\text{att}^*) = 0$ where att^* is some *fixed* attribute.

The public parameters PK_{att^*} of AB-PCE consist of a matrix \mathbf{A} and a series of “GSW encryptions” defined as $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i - \text{att}_i^* \cdot \mathbf{G}$ where \mathbf{R}_i is a binary matrix and \mathbf{G} is the gadget matrix. This is exactly the form that the public parameters have in the *security proof* of Boneh et al. [13], where the attribute $\text{att}^* = (\text{att}_1^*, \dots, \text{att}_\ell^*)$ is punctured out: The reduction (playing the role of the key authority) can efficiently compute decryption keys for any circuit f such that $f(\text{att}^*) = 0$. To compute a decryption key for the function f , one can consider the following homomorphic evaluation algorithm defined in [13]: $\mathbf{R}_f = \text{EvalRand}(f, \{\mathbf{A}_i\}_{i \in [\ell]}, \{\mathbf{R}_i\}_{i \in [\ell]}, \text{att}^*)$, where the knowledge of \mathbf{R}_f allows one to compute a uniformly sampled trapdoor \mathbf{X}_f for the matrix $[\mathbf{A} \ \mathbf{A}_f + \mathbf{G}]$ (provided that $f(\text{att}^*) = 0$), where $\mathbf{A}_f = \mathbf{A}\mathbf{R}_f$.

Issue: Semi-Malicious Security. One subtle aspect, that prevents us from using this approach off-the-shelf, is that the security proof of [13] assumes that matrix \mathbf{A} is honestly sampled. However, the argument completely breaks down if \mathbf{A} is sampled with a trapdoor (in fact, this change is not even detectable, since the two distributions are statistically close). This means that the construction is not secure against a semi-malicious authority. To overcome this obstacle, we resort to the techniques of [17, 6, 25]: Instead of sampling \mathbf{A} uniformly, we choose it from a structured distribution (which is guaranteed to have no trapdoor) and we adapt the encryption algorithm to hide the message, unless one has a trapdoor for $[\mathbf{A} \ \mathbf{A}_f]$.

Identity-Based PCE for General Constraints. With the above observation in mind, we show that a similar approach gives rise to an identity-based PCE (IB-PCE) scheme for *general constraints*. An IB-PCE scheme follows a similar syntax as PCE except that the encryption algorithm now additionally takes as input an identity id (from an identity universe) and

secret keys are also associated with identities. We require that the authority can decrypt a ciphertext associated with an identity id iff $C(\text{id}) = 1$.⁵

In our scheme we set the public parameters PK_C to the same GSW encryptions $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i - C_i \cdot \mathbf{G}$, where (C_1, \dots, C_ℓ) is the binary representation of the constraint C . To generate a key for an identity id , we define $\mathbf{A}_{\text{id}} = \text{EvalPK}(U_{\text{id}}, \{\mathbf{A}_i\}_{i \in [\ell]})$ where U_{id} is the universal circuit that evaluates $C(\text{id})$. Using the knowledge of the random coins $(\mathbf{R}_1, \dots, \mathbf{R}_\ell)$ we can generate a trapdoor for $[\mathbf{A} \ \mathbf{A}_{\text{id}}]$ (i.e., a key for id) only if $U_{\text{id}} = C(\text{id}) = 1$. Note the ability to “compute over the attributes” here is used to enforce the constraint C , rather than enabling the ABE functionality. As we will see later, achieving AB-PCE for general constraints will (inherently) need more powerful tools.

Application: Coercion-Resistant Signatures. Consider the following scenario: Alice wants to generate a digital signature key pair (sk, vk) that allows her to sign any message m , except if m contains some flagged words (e.g., connected to hate speech or racial discrimination). Alice wants to ensure that this continues to hold even in the event where the secret key sk gets stolen, so that even the impostor cannot sign such messages on behalf of Alice. Put differently, she wants to ensure that *not even Alice herself* can sign flagged messages under external pressure (e.g., in case she is physically forced to). One plausible approach to realize such a cryptographic primitive would be to use functional/delegatable signatures [16], encoding the flagged words in the delegation policy. However, here all security is lost if an attacker gets access to the master secret key (which allows one to sign any message). This could happen if the master key is inadvertently leaked or not properly erased by Alice.

To overcome this limitation, we propose to construct such “coercion-resistant” signatures using our newly developed IB-PCE: The signing key consists of a pre-constrained key PK_C for the appropriate constraint C , and a message m can be signed by simply generating a key for the identity m , i.e., a trapdoor \mathbf{X}_m for the matrix $[\mathbf{A} \ \mathbf{A}_m]$. Given the public parameters, anyone can recompute \mathbf{A}_m and check that the signature \mathbf{X}_m is indeed a valid trapdoor for $[\mathbf{A} \ \mathbf{A}_m]$. The security against authority guarantees that, at any point in time, not even the owner of the master secret key, can sign messages m for which $C(m) \neq 1$.

Attribute-Based PCE for General Constraints. Next we consider the problem of constructing AB-PCE for general constraints where we can specify an arbitrary circuit C at setup time we constrain the key generation algorithm to only be able to compute keys for functions f such that $C(f) = 1$. We show that AB-PCE for general constraints is a significantly powerful object, and it is in fact *equivalent* to the notion of *witness encryption* [22].

Given a witness encryption scheme, we can define the setup of AB-PCE to compute a commitment $\mathbf{c} = \text{Comm}(C^*; r)$ and the common reference string (CRS) for a non-interactive zero-knowledge (NIZK) scheme. The secret key for a function f simply consists of a NIZK proof π for the statement

$$\exists (r, C^*) \text{ such that: } \mathbf{c} = \text{Comm}(C^*; r) \wedge C^*(f) = 1.$$

To encrypt a message μ for an attribute att , one computes a witness encryption for the language that takes as input (f, π) , verifies π and returns μ if $f(\text{att}) = 1$. It is not hard to see that no (semi-malicious) authority can issue valid keys for constrained functions, as

⁵ For non-trivial IB-PCE, it is important that the public parameters do not reveal the constraint. See Remark 11 in Section 4.

4:10 Pre-Constrained Encryption

it would contradict the soundness of the proof.⁶ Furthermore, the constraint C^* is hidden by the hiding of the commitment and the zero-knowledge property of NIZK. We defer the discussion on the reverse direction (AB-PCE \implies witness encryption) to a later point in this overview.

PCE for General Constraints. Naturally, one may wonder whether the above ideas can be lifted to the general setting of PCE, where instead of releasing the message μ if $f(\text{att}) = 1$, we want to reveal $f(\mu)$ to the owner of a key for f . Drawing from our previous scheme, we achieve this by relying on indistinguishability obfuscation: We only need to change the encryption algorithm to obfuscate a circuit that computes

If $V(\text{crs}, (\mathbf{c}, f), \pi) = 1$ then return $f(\mu)$.

We will discuss the reverse implication (PCE \implies obfuscation) later in more details.

2.2 Malicious Security

Thus far we have considered an authority that can choose the random coins arbitrarily, but otherwise it is trusted to sample the public parameters of the schemes correctly. While this notion might not be sufficient for some practical application, it functions as a gateway for the stronger notion of security in the presence of a fully corrupted authority. In fact we show that, using techniques from the literature on combiners [4], we can generically turn a semi-malicious pre-constrained encryption scheme into a fully malicious one.

We describe here the transformation for the general case of PCE – the case of IB-PCE/AB-PCE follows a similar approach (but is slightly simpler). We adopt a dual-system approach, where the authority samples two independent public parameters $(\text{PK}_0, \text{PK}_1)$ of a semi-malicious scheme and computes a non-interactive witness indistinguishable (NIWI) proof that either of them is well-formed. At encryption time, the user secret shares its message $(\mu_0, \mu_1) \leftarrow \text{Share}(\mu)$ and encrypts μ_b with respect to PK_b . Intuitively, this guarantees that, if the function f falls into the constrained set, then the authority cannot issue a valid key for f and thus at least one of the two shares (and consequently the message) is hidden. To ensure that the functionality of the scheme is preserved, we need the additional property that the function f can be locally evaluated on each share individually, since the two independent PCE schemes cannot “communicate” with each other. More formally, we need the existence of an evaluation algorithm Eval such that $\text{Eval}(f, \mu_0, 0) \oplus \text{Eval}(f, \mu_1, 1) = f(\mu)$, which is exactly what is guaranteed by homomorphic secret sharing schemes [15]. One can then obtain the original functionality by issuing the functional keys for the functions $\text{Eval}(f, \cdot, 0)$ and $\text{Eval}(f, \cdot, 1)$. Our scheme crucially relies on the fact that the secret sharing scheme to program the correct output of the computation in the security proof. For further details, we refer the reader to the technical sections.

2.3 Cryptographic Implications

We now argue that the strong cryptographic tools that we use to construct AB-PCE/PCE for general constraints are in some sense necessary, by presenting the reverse implications. Note that the only security property that we need from our pre-constrained schemes is that

⁶ For this argument to go through, we need to assume that the NIZK is perfectly sound if the CRS is correctly generated, which can be instantiated from [34].

of security against authority; in particular, the notion of constraint hiding (although useful for other applications) is not required for the following implications.

AB-PCE to WE. An AB-PCE scheme can be transformed into a witness encryption scheme (for a language \mathcal{L} with relation \mathcal{R}) in the following manner: to encrypt a message μ w.r.t. a statement $x \in \mathcal{L}$, publish the tuple $(x, \text{Enc}(\text{PK}, x, \mu), \text{MSK}[C])$, where the constraint C restricts the functions f for the AB-PCE scheme to be of the form $f_w(x) = \mathcal{R}(w, x)$. Anyone who has a valid witness w for x can first compute a secret key sk_{f_w} using $\text{MSK}[C]$ and then use it to decrypt the ciphertext. However, if $x \notin \mathcal{L}$, then no admissible function can satisfy the policy associated with the ciphertext and thus the message μ is hidden.

PCE to iO. For the case of PCE, one may think that the implication is obvious, since it is well-known that FE implies obfuscation [5, 10]. However, we stress that we are only relying on the notion of security against authority for PCE, which is incomparable to the standard notion of security for FE. In fact, our transformation is much simpler and more direct than those of [5, 10].

Given a PCE scheme, we can obfuscate a circuit Γ by computing $(\text{Enc}(\text{PK}, \Gamma), \text{MSK}[C])$, where the constraint C restricts the functions to be universal circuits of the form $U_x(\Gamma) = \Gamma(x)$. To evaluate the obfuscated circuit Γ on an input x , the user can first use $\text{MSK}[C]$ to derive a secret key sk_{U_x} for circuit U_x and then use it to decrypt $\text{Enc}(\text{PK}, \Gamma)$, which returns $\Gamma(x)$.

Interestingly, this transformation does *not* incur in an exponential loss of security (which is the case for all known FE to obfuscation compilers), which suggests that PCE – in its full generality – is even more intimately related to obfuscation.

2.4 Organization

We first present the preliminaries in Section 3. We divide the technical sections into three parts:

- On Page 12, we present the first part. This part consists of the definitions and also semi-malicious to malicious security transformations.
- On Page 17, we present the second part. This part consists of different constructions of PCE. We defer this part to the full version.
- On Page 17, we present the third part. This part consists of the implications of PCE.

3 Preliminaries

Let \mathbb{Z} be the set of all integers. For any integer q , denote $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. For any integer n , let $[n] = \{1, 2, \dots, n\}$. For any discrete distributions P and Q , we let $\text{SD}(P, Q)$ denote the statistical distance between P and Q , i.e. $\text{SD}(P, Q) = \sum_i |\Pr[P = i] - \Pr[Q = i]|/2$. For any random variables X and Y , we let $H_\infty(X) = -\log_2(\min_i \Pr[P = i])$ denote the min-entropy of X , and let $\tilde{H}_\infty(X|Y) = -\log(\mathbb{E}_y[\max_x \Pr[X = x|Y = y]])$ denote the average conditional min-entropy.

We defer the rest of the preliminary to the full version.

Part I: Definitions and Compilers

4 Pre-Constrained Encryption Systems

We start by presenting our definition of pre-constrained encryption (PCE). Later, we will consider the weaker variants of PCE, namely attribute-based pre-constrained encryption and identity-based pre-constrained encryption.

► **Remark 1 (On Key Delegation).** Our definitions presented below already incorporate *key delegation* feature, namely, the ability to derive special-purpose decryption keys for “authorized” functions from the master secret key. We remark that it is easy to adapt our definitions to exclude the key delegation feature in the manner as discussed in Section 1.

4.1 Definition of PCE

Syntax. A pre-constrained encryption (PCE) scheme for a constraint family \mathcal{C} and function family \mathcal{F} consists of four algorithms (Setup, KeyGen, Enc, Dec) described below.

- **Setup**($1^\lambda, C$): The setup algorithm takes as input a constraint predicate $C \in \mathcal{C}$ and produces a master secret key $\text{MSK}[C]$ along with a public key PK.
- **KeyGen**($\text{MSK}[C], f$): The key generation algorithm takes as input $\text{MSK}[C]$, function $f \in \mathcal{F}$ and produces either a functional key sk_f (when f satisfies the predicate, i.e., $C(f) = 1$) or \perp .
- **Enc**(PK, x): On input PK and a message x , the encryption algorithm outputs either a ciphertext CT (when the public key is “well formed”) or \perp .
- **Dec**(sk_f, CT): The decryption algorithm takes as input a functional key sk_f along with a ciphertext CT and outputs a value y .

We require a PCE scheme to satisfy many properties. We start with the correctness property.

Correctness. We require that the decryption of an encryption of x using a functional key associated with a function f yields $f(x)$. That is, for every $C \in \mathcal{C}$ and every $f \in \mathcal{F}$ such that $C(f) = 1$, for every $m \in \{0, 1\}^\ell$ where ℓ is the input length of f ,

$$\Pr \left[(\text{PK}, \text{MSK}[C]) \leftarrow \text{Setup}(1^\lambda, C), \text{sk}_f \leftarrow \text{KeyGen}(\text{MSK}[C], f), \text{CT} \leftarrow \text{Enc}(\text{PK}, m), y \leftarrow \text{Dec}(\text{sk}_f, \text{CT}), : y = f(x) \right] \geq 1 - \text{negl}(\lambda)$$

4.1.1 Security Against Authority

We consider two security properties that we refer to as *security against authority* (SAA) and *constraint hiding* (CH).

Security against Authority. We first define security against authority for PCFE. Following the literature on secure computation, we consider three formulations: security against *semi-honest* authority (SH-SAA), security against *semi-malicious* authority (SM-SAA), and security against *malicious* authority (M-SAA).

We start by presenting the definition for SH-SAA. Roughly speaking, this notion requires that for any constraint C , even an authority who honestly runs the setup procedure on input C (and therefore knows the randomness used for the setup) should not be able to distinguish between encryptions of any two equal length messages x_0 and x_1 such that for all functions f satisfying $C(f) = 1$, we have that $f(x_0) = f(x_1)$.

► **Definition 2** (PCE: Security against Semi-Honest Authority). A PCE scheme $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ for constraint family \mathcal{C} and function family \mathcal{F} satisfies security against semi-honest authority if for any non-uniform PPT adversary \mathcal{A} , any $C \in \mathcal{C}$, and every pair of equal-length messages (x_0, x_1) s.t. for all $f \in \mathcal{F}$ satisfying $C(f) = 1$, $f(x_0) = f(x_1)$,

$$\left| \Pr \left[r \xleftarrow{\$} \{0, 1\}^{\text{poly}(\lambda)}; (\text{PK}, \text{MSK}[C]) \leftarrow \text{Setup}(1^\lambda, C; r); b \xleftarrow{\$} \{0, 1\} : \mathcal{A}(1^\lambda, r, \text{Enc}(\text{PK}, x_b)) = b \right] \right. \\ \left. < \frac{1}{2} + \text{negl}(\lambda). \right.$$

We next present the definition for SM-SAA. The main difference from above is that the adversary can run the setup process using arbitrary randomness. That is, in the generation of setup, it uses a constraint circuit C from the same constraint family \mathcal{C} (as done by the semi-honest adversary) but the randomness used in the setup can be adversarially chosen.

► **Definition 3** (PCE: Security against Semi-Malicious Authority). A PCE scheme $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ for constraint family \mathcal{C} and function family \mathcal{F} satisfies security against semi-malicious authority if for any non-uniform (stateful) PPT adversary \mathcal{A} , $|\Pr[\text{Expr}_0^{\text{SM-SAA}} = 1] - \Pr[\text{Expr}_1^{\text{SM-SAA}} = 1]| \leq \text{negl}(\lambda)$, where $\text{Expr}_b^{\text{SM-SAA}}$, for every $b \in \{0, 1\}$, is defined as the following experiment:

Experiment $\text{Expr}_b^{\text{SM-SAA}}$:

- $(\text{PK}, C, r, x_0, x_1) \leftarrow \mathcal{A}(1^\lambda)$.
 - $\text{CT} \leftarrow \text{Enc}(\text{PK}, x_b)$.
 - $b' \leftarrow \mathcal{A}(\text{CT})$.
 - Output b' if:
 - $C \in \mathcal{C}$
 - $\text{PK}' = \text{PK}$, where $(\text{PK}', \text{MSK}'[C]) \leftarrow \text{Setup}(1^\lambda, C; r)$
 - $|x_0| = |x_1|$, and for all $f \in \mathcal{F}$ satisfying $C(f) = 1$, it holds that $f(x_0) = f(x_1)$.
- Else output 1.

Finally, we present the definition for M-SAA. In this case, the adversary may run the setup process maliciously. We therefore require the existence of a (possibly inefficient) extractor algorithm that takes as input the public key PK output by the adversary and extracts a constraint C from PK. We require that it is efficient to check membership in \mathcal{C} .

► **Definition 4** (PCE: Security against Malicious Authority). A PCE scheme $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ for constraint family \mathcal{C} and function family \mathcal{F} satisfies security against malicious authority if there exists a (possibly inefficient) extractor algorithm Ext such that for any non-uniform (stateful) PPT adversary \mathcal{A} , $|\Pr[\text{Expr}_0^{\text{M-SAA}} = 1] - \Pr[\text{Expr}_1^{\text{M-SAA}} = 1]| \leq \text{negl}(\lambda)$, where $\text{Expr}_b^{\text{M-SAA}}$, for every $b \in \{0, 1\}$, is defined as the following experiment:

Experiment $\text{Expr}_b^{\text{M-SAA}}$:

- $(\text{PK}, x_0, x_1) \leftarrow \mathcal{A}(1^\lambda)$.
 - $\text{CT} \leftarrow \text{Enc}(\text{PK}, x_b)$.
 - $b' \leftarrow \mathcal{A}(\text{CT})$.
 - Output b' if:
 - $C \in \mathcal{C}$, where $C \leftarrow \text{Ext}(1^\lambda, \text{PK})$.
 - $|x_0| = |x_1|$, and for all $f \in \mathcal{F}$ satisfying $C(f) = 1$, it holds that $f(x_0) = f(x_1)$.
- Else, output 1.

4.1.2 Constraint Hiding

We next define the constraint hiding property for PCE. Roughly speaking, we require that the public key PK computed w.r.t. a constraint C should not leak any information about P beyond what is leaked by the functional keys issued by the authority. Suppose $\text{sk}_{f_1}, \dots, \text{sk}_{f_q}$ are the functional keys issued by the authority. We want the guarantee that the adversary cannot distinguish whether the predicate used in the setup is either C_0 or C_1 , where C_0 and C_1 are predicates whose descriptions are of the same length such that $C_0(f_i) = C_1(f_i)$, for all $i \in [q]$.

► **Definition 5** (PCE: Constraint Hiding). *A PCE scheme $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ for constraint family \mathcal{C} and function family \mathcal{F} satisfies constraint hiding (CH) if for any non-uniform admissible (stateful) PPT adversary \mathcal{A} ,*

$$\Pr \left[\begin{array}{l} \mathcal{A}^{\text{KeyGen}(\text{MSK}[C_b]; \cdot)}(\text{PK}) = b : \\ b \xleftarrow{\$} \{0,1\}, \\ (C_0, C_1) \leftarrow \mathcal{A}(1^\lambda) \\ (\text{PK}, \text{MSK}[C_b]) \leftarrow \text{Setup}(1^\lambda, C_b) \end{array} \right] < \frac{1}{2} + \text{negl}(\lambda),$$

where \mathcal{A} is said to be admissible if $|C_0| = |C_1|$, $C_0, C_1 \in \mathcal{C}$, and every function key query f that it issues to the key generation oracle is such that $C_0(f) = C_1(f)$.

4.1.3 Collusion Resistance

So far, we have considered two different adversarial scenarios. In the security against authority property, we considered the scenario where the authority is adversarial and the goal was to still guarantee the fact that the ciphertexts hide the underlying messages. In the constraint hiding property, we wanted to protect the privacy of the constraint circuit used in the setup algorithm.

We consider a third scenario that is not captured by the two properties defined before. In this scenario, the entities who receive the keys for functions, say f_1, \dots, f_q , could be corrupted. In this case, we would like to guarantee that the ciphertexts don't leak any more information than revealed by the output of the functions f_1, \dots, f_q . In more detail, we would like the computational indistinguishability of the encryptions of x_0 and x_1 to hold even given keys for functions f_1, \dots, f_q . Note that this is the same security requirement in advanced encryption systems, such as functional encryption.

We define this formally below.

► **Definition 6** (PCE: Collusion Resistance). *A PCE scheme $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ for constraint family \mathcal{C} and function family \mathcal{F} satisfies collusion-resistance (CR) if for any non-uniform admissible (stateful) PPT adversary \mathcal{A} ,*

$$\Pr \left[\begin{array}{l} \mathcal{A}(\text{PK}, \text{sk}_{f_1}, \dots, \text{sk}_{f_q}, \text{CT}_b) = b : \\ b \xleftarrow{\$} \{0,1\}, \\ (C, f_1, \dots, f_q, x_0, x_1) \leftarrow \mathcal{A}(1^\lambda) \\ (\text{PK}, \text{MSK}[C]) \leftarrow \text{Setup}(1^\lambda, C), \\ \text{CT}_b \leftarrow \text{Enc}(\text{PK}, x_b), \\ \forall i \in [q], \text{sk}_{f_i} \leftarrow \text{KeyGen}(\text{MSK}[C], f_i) \end{array} \right] < \frac{1}{2} + \text{negl}(\lambda),$$

where \mathcal{A} is said to be admissible if: (a) $C(f_i) = 1$, for all $i \in [q]$, (b) $f_i(x_0) = f_i(x_1)$ for all $i \in [q]$ and, (c) $|x_0| = |x_1|$.

We say that a PCE scheme satisfies bounded collusion resistance if q , the number of functions declared by the adversary, and s , the maximum size of the circuits input to the key generation algorithm, are specified as an input parameter in the setup algorithm. Specifically, the size of the public parameters could grow polynomially in q and s .

► **Remark 7.** In the above security definition, we require the adversary to declare all the functions at the time of setup. In line with the different security definitions of functional encryption, we can similarly relax this requirement and allow the adversary to adaptively make function queries. We leave the exploration of the stronger definitions for future work.

Generic Transformation. We show that the two properties we defined in Sections 4.1.1 and 4.1.2 have implications to collusion resistance.

► **Lemma 8.** *If a PCE scheme satisfies the (semi-honest) security against authority property and constrained hiding property then it also satisfies bounded collusion resistance.*

We defer the proof to the full version.

► **Remark 9.** Suppose a PCE scheme satisfies a stronger constraint hiding property, where the adversary can declare *any* two equivalent and efficient circuits C_0 and C_1 (refer to Definition 5 for notation). Specifically, there is no requirement that the descriptions of C_0 and C_1 need to have the same length. This property alongside semi-honest security against authority property implies collusion resistance (rather than *bounded* collusion resistance). Since the proof is identical to the proof of the above lemma, we omit the details.

4.2 Special Cases: Identity-Based PCE and Attribute-Based PCE

Attribute-Based Pre-Constrained Encryption. We consider a notion of attribute-based pre-constrained encryption (AB-PCE) that is defined w.r.t. a constraint family \mathcal{C} , an attribute universe \mathcal{U} and a predicate family \mathcal{P} where a predicate $P \in \mathcal{P}$ operates over vectors of attributes in \mathcal{U} . In an AB-PCE scheme, messages are encrypted w.r.t. attribute vectors in a manner similar to (key-policy) attribute-based encryption [41, 31]. The constraint \mathcal{C} captures the set of *authorized* predicates: specifically, the authority can only decrypt a ciphertext computed w.r.t. an attribute vector $\text{att} \in \mathcal{U}$ iff there exists an authorized predicate $P \in \mathcal{P}$ s.t. $C(P) = 1$ and $P(\text{att}) = 1$.

More specifically, an AB-PCE scheme consists of four algorithms (Setup, KeyGen, Enc, Dec) that are defined in a similar manner as a PCE scheme except for the following differences:

- The key generation algorithm KeyGen computes keys associated with (boolean) predicates $P \in \mathcal{P}$.
- The encryption algorithm Enc takes an attribute vector att as an additional input.
- The decryption algorithm Dec on input a key sk_P and a ciphertext CT – computed using an attribute vector att and message x – outputs x iff $P(\text{att}) = 1$.

As in the case of PCE, we consider two main security properties – *security against authority* and *constraint hiding* – for AB-PCE. The constraint hiding definition is almost identical to the one for PCE and therefore we skip it here. The definitions for security against authority are also very similar to the PCE case.

We present the formal definition for the case of semi-malicious authority here for completeness. The definitions for the other two cases – semi-honest or malicious – can be easily derived with appropriate modifications.

► **Definition 10** (AB-PCE: Security against Semi-Malicious Authority). *An AB-PCE scheme (Setup, KeyGen, Enc, Dec) for constraint family \mathcal{C} , attribute universe \mathcal{U} and predicate family \mathcal{P} satisfies security against semi-malicious authority if for any non-uniform (stateful) PPT adversary \mathcal{A} , $|\Pr[\text{Exp}_0^{\text{SM-SAA}} = 1] - \Pr[\text{Exp}_1^{\text{SM-SAA}} = 1]| \leq \text{negl}(\lambda)$, where $\text{Exp}_b^{\text{SM-SAA}}$, for every $b \in \{0, 1\}$, is defined as the following experiment:*

4:16 Pre-Constrained Encryption

Experiment $\text{Exp}_b^{\text{SM-SAA}}$:

- $(\text{PK}, C, r, \text{att}, x_0, x_1) \leftarrow \mathcal{A}(1^\lambda)$.
- $\text{CT} \leftarrow \text{Enc}(\text{PK}, \text{att}, x_b)$.
- $b' \leftarrow \mathcal{A}(\text{CT})$.
- Output b' if:
 - $C \in \mathcal{C}$
 - $\text{PK}' = \text{PK}$, where $(\text{PK}', \text{MSK}'[C]) \leftarrow \text{Setup}(1^\lambda, C; r)$
 - $\text{att} \in \mathcal{U}$, $|x_0| = |x_1|$, and for all $P \in \mathcal{P}$ satisfying $C(P) = 1$, it holds that $P(\text{att}) = 0$.
- Else output 1.

Identity-Based Pre-Constrained Encryption. We also consider a notion of identity-based pre-constrained encryption (IB-PCE) that is defined w.r.t. a constraint family \mathcal{C} and an identity universe \mathcal{I} . In an IB-PCE scheme, messages are encrypted together with identities (in a manner similar to identity-based encryption [12]). Crucially, the constraint C captures the set of *authorized* identities, i.e., the authority can only decrypt ciphertexts computed using identities $\text{id} \in \mathcal{I}$ s.t. $C(\text{id}) = 1$.

Intuitively, IB-PCE allows for *full decryption* of ciphertexts computed w.r.t. authorized identities. For example, the authorized identities could capture people on a “no-fly list”. More generally, IB-PCE is useful for scenarios where we wish to allow an authority to be able to fully decrypt ciphertexts after an *exact match* of identities (which may capture real-world identities, keywords, etc).

More specifically, an IB-PCE scheme consists of four algorithms (Setup , KeyGen , Enc , Dec) that are defined in a similar manner as a PCE scheme except for the following differences:

- The key generation algorithm KeyGen takes as input an identity $\text{id} \in \mathcal{I}$ together with the (pre-constrained) master secret key and outputs a key sk_{id} .
- The encryption algorithm Enc takes an identity $\text{id} \in \mathcal{I}$ as an additional input.
- The decryption algorithm Dec on input a key sk_{id} and a ciphertext CT – computed using an identity id' and message x – outputs the plaintext x iff $\text{id} = \text{id}'$.

As earlier, we consider two key security properties – security against authority and constraint hiding – for IB-PCE. Both these properties are defined in a similar manner as for PCE and AB-PCE.

► **Remark 11 (On Non-Trivial IB-PCE).** For non-trivial IB-PCE, it is important that the public parameters do not reveal the constraint. Indeed, in the absence of such a requirement, there is a trivial construction of IB-PCE where the encryptor first simply checks – given the public constraint C – whether or not the input identity satisfies C . If the check fails, then it simply outputs \perp .

We note, however, that the above strategy is not always possible and therefore constraint hiding is not always necessary for PCE. Indeed, we later discuss applications of PCE (such as to witness encryption and program obfuscation) where constraint hiding is not required.

5 Semi-Malicious to Malicious Security Transformations

We demonstrate generic transformations that convert any pre-constrained \mathcal{X} scheme, where $\mathcal{X} \in \{\text{IB-PCE}, \text{AB-PCE}, \text{PCE}\}$ satisfying security against semi-honest authority property into another scheme that guarantees security against a malicious authority. We defer the construction and its security proofs to the full version.

Part II: Constructions

In this part, we construct PCE for point constraints. Then we construct IB-PCE, AB-PCE, and PCE for general constrains. We defer the constructions to the full version.

Part III: Implications

We present implications of pre-constrained encryption systems to primitives such as witness encryption and indistinguishability obfuscation (iO). We show the following:

- Pre-constrained ABE implies witness encryption for NP.
- Pre-constrained encryption implies iO for P/poly.

For both the above implications it suffices for the pre-constrained encryption system to satisfy the weaker semi-honest security property.

We defer the proofs to the full version.

References

- 1 A declassified court ruling shows how the fbi abused nsa mass surveillance data, 2019. <https://theintercept.com/2019/10/10/fbi-nsa-mass-surveillance-abuse/>.
- 2 Graham, Blumenthal, Hawley, Feinstein Introduce EARN IT Act to Encourage Tech Industry to Take Online Child Sexual Exploitation Seriously, 2020. URL: <https://www.judiciary.senate.gov/press/rep/releases/graham-blumenthal-hawley-feinstein-introduce-earn-it-act-to-encourage-tech-industry-to-take-online-child-sexual-exploitation-seriously>.
- 3 Mubaraz Ahmed and Fred Lloyd George. A war of keywords, how extremists are exploiting the internet and what to do about it. https://institute.global/sites/default/files/inline-files/IGC_War%20of%20Keywords_23.08.17_0.pdf.
- 4 Prabhanjan Ananth, Aayush Jain, Moni Naor, Amit Sahai, and Eylon Yogev. Universal constructions and robust combiners for indistinguishability obfuscation and witness encryption. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 491–520. Springer, Heidelberg, August 2016. doi:10.1007/978-3-662-53008-5_17.
- 5 Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Heidelberg, August 2015. doi:10.1007/978-3-662-47989-6_15.
- 6 Prabhanjan Ananth, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Unbounded multi-party computation from learning with errors. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 754–781. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77886-6_26.
- 7 Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 483–501. Springer, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_29.
- 8 Saikrishna Badrinarayanan, Vipul Goyal, Aayush Jain, and Amit Sahai. Verifiable functional encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 557–587. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_19.

- 9 Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_1.
- 10 Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 171–190. IEEE Computer Society Press, October 2015. doi:10.1109/FOCS.2015.20.
- 11 Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988. doi:10.1145/62212.62222.
- 12 Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_13.
- 13 Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_30.
- 14 Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011. doi:10.1007/978-3-642-19571-6_16.
- 15 Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 509–539. Springer, Heidelberg, August 2016. doi:10.1007/978-3-662-53018-4_19.
- 16 Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, Heidelberg, March 2014. doi:10.1007/978-3-642-54631-0_29.
- 17 Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 370–390. Springer, Heidelberg, November 2018. doi:10.1007/978-3-030-03810-6_14.
- 18 Melissa Chase. Multi-authority attribute based encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 515–534. Springer, Heidelberg, February 2007. doi:10.1007/978-3-540-70936-7_28.
- 19 Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 52–72. Springer, Heidelberg, August 1988. doi:10.1007/3-540-48184-2_5.
- 20 Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990. doi:10.1109/FSCS.1990.89549.
- 21 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013. doi:10.1109/FOCS.2013.13.
- 22 Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013. doi:10.1145/2488608.2488667.
- 23 Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. Registration-based encryption: Removing private-key generator from IBE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 689–718. Springer, Heidelberg, November 2018. doi:10.1007/978-3-030-03807-6_25.

- 24 Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, Ahmadreza Rahimi, and Sruthi Sekar. Registration-based encryption from standard assumptions. In Dongdai Lin and Kazuo Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 63–93. Springer, Heidelberg, April 2019. doi:10.1007/978-3-030-17259-6_3.
- 25 Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 97–126. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77883-5_4.
- 26 Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nikolai Zeldovich. How to run turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_30.
- 27 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985. doi:10.1145/22145.22178.
- 28 Rishab Goyal and Satyanarayana Vusirikala. Verifiable registration-based encryption. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 621–651. Springer, Heidelberg, August 2020. doi:10.1007/978-3-030-56784-2_21.
- 29 Vipul Goyal. Reducing trust in the PKG in identity based cryptosystems. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 430–447. Springer, Heidelberg, August 2007. doi:10.1007/978-3-540-74143-5_24.
- 30 Vipul Goyal, Steve Lu, Amit Sahai, and Brent Waters. Black-box accountable authority identity-based encryption. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008*, pages 427–436. ACM Press, October 2008. doi:10.1145/1455770.1455824.
- 31 Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309. doi:10.1145/1180405.1180418.
- 32 Matthew Green, Gabriel Kaptchuk, and Gijs Van Laer. Abuse resistant law enforcement access systems. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 553–583. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77883-5_19.
- 33 Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Heidelberg, August 2006. doi:10.1007/11818175_6.
- 34 Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Heidelberg, May / June 2006. doi:10.1007/11761679_21.
- 35 Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 406–425. Springer, Heidelberg, May 2011. doi:10.1007/978-3-642-20465-4_23.
- 36 Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Heidelberg, May 2011. doi:10.1007/978-3-642-20465-4_30.
- 37 Adam O’Neill. Definitional issues in functional encryption. *IACR Cryptol. ePrint Arch.*, 2010:556, 2010. URL: <http://eprint.iacr.org/2010/556>.

- 38 Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008. doi:10.1145/1374376.1374406.
- 39 Willy Quach, Brent Waters, and Daniel Wichs. Targeted lossy functions and applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 424–453, Virtual Event, August 2021. Springer, Heidelberg. doi:10.1007/978-3-030-84259-8_15.
- 40 Phillip Rogaway. The moral character of cryptographic work. *IACR Cryptol. ePrint Arch.*, 2015:1162, 2015. URL: <http://eprint.iacr.org/2015/1162>.
- 41 Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005. doi:10.1007/11426639_27.
- 42 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, 2018.
- 43 Adi Shamir. On the security of DES. In Hugh C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, pages 280–281. Springer, Heidelberg, August 1986. doi:10.1007/3-540-39799-X_22.
- 44 Charles Wright and Mayank Varia. Crypto crumple zones: Enabling limited access without mass surveillance. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 288–306. IEEE, 2018.