Breaking Omertà: On Threshold Cryptography, Smart Collusion, and Whistleblowing

Mahimna Kelkar[*]
Cornell University

Aadityan Ganesh[*]
Princeton University

Aditi Partap
Stanford University

Joseph Bonneau
NYU & a16z crypto

S. Matthew Weinberg
Princeton University

Abstract

Cryptographic protocols often make honesty assumptions—e.g., fewer than t out of n participants are adversarial. In practice, these assumptions can be hard to ensure, particularly given monetary incentives for participants to collude and deviate from the protocol.

In this work, we explore combining techniques from cryptography and mechanism design to discourage collusion. We formalize protocols in which colluders submit a cryptographic proof to *whistleblow* against their co-conspirators, revealing the dishonest behavior publicly. We provide general results on the cryptographic feasibility, and show how whistleblowing fits a number of applications including secret sharing, randomness beacons, and anonymous credentials.

We also introduce *smart collusion* — a new model for players to collude. Analogous to blockchain smart contracts, smart collusion allows colluding parties to arbitrarily coordinate and impose penalties on defectors (e.g., those that blow the whistle). We show that unconditional security is impossible against smart colluders even when whistleblowing is anonymous and can identify all colluding players. On the positive side, we construct a whistleblowing protocol that requires only a small deposit and can protect against smart collusion even with roughly t times larger deposit.

A preliminary version of this paper appears in the proceedings of ACM CCS 2025. This is the full version.

[*]The first two authors contributed equally to this work.

Contents

1 Introduction

Cryptographic protocols often make *honesty* assumptions on how participants act. A common assumption is that fewer than t (out of a total of n) players are corrupt, while all other players execute the protocol honestly. This model is seen in many applications, including secret sharing [71], threshold cryptography [35], multi-party computation [9, 51], and private information retrieval [26].

But in practice, it is often more realistic to consider nodes as *rational* with potential *monetary incentives* to collude. As an example, in the context of secret sharing (where each player holds a share of a secret s), a coalition of players might profit by colluding to recover s earlier than expected. It is therefore important to design protocols to discourage rational players from colluding as they seek to maximize their utility.

In this work, we explore a broad framework for integrating techniques from both mechanism design and cryptography to establish feasibility and impossibility results for dealing with collusion.

Deterring collusion. Since we cannot make it impossible for players to collude, we will instead focus on *deterring* collusion by e.g., making it unstable or unprofitable for rational players. One promising approach is to provide a way for colluding players to *whistleblow* (or *snitch*) against their co-conspirators for a reward. This was explored very recently for secret-sharing protocols starting with Dziembowski et al. [40] and later in [17, 53] (works concurrent to ours). In this context, [17, 40] consider only the cryptographic angle for whistleblowing while [53] also formalizes how rewards should be set. A similar question on deterring collusion was also previously considered in the context of PIR [52]. Separately, other works build *cryptographic* tools for accountability (i.e., detecting malicious parties) for various primitives (see Section 1.2 for related works).

In this work, we design a broad framework for whistleblowing protocols that captures several applications. Abstractly, a whistleblowing protocol will consist of two components: (1) a *cryptographic* tool that enables players to verifiably prove the existence of collusion and/or its members; and (2) a *game-theoretic* mechanism to incentivize players to actually blow the whistle and impose penalties on colluders. The cryptographic proof fits within the broad umbrella of proving failures in the protocol's original security model. But without understanding the incentives, it *says nothing* about whether players will want to use it as opposed to staying silent.

We demonstrate how a strong yet natural form of collusion—which to the best of our knowledge has not been considered by prior work—can be used to *circumvent the positive results* from [40, 53]. Our work considers whistleblowing protocols in the context of this new form of collusion, which we will call *smart collusion*[1].

To avoid additional trust assumptions on who verifies the correctness of the collusion proof, we will require that our whistleblowing protocols not require trusted executors or private state. In other words, the verification can itself be instantiated through a blockchain smart contract.

A new notion — *smart* collusion. The basic idea behind smart collusion is that colluders have knowledge of the mechanism chosen by the protocol designer and can act accordingly in an attempt to circumvent the mechanism.

As an illustration, consider the following scenario. Suppose that we have a cryptographic protocol (e.g., secret sharing) that is susceptible to collusion. In an attempt to deter collusion,

[1]The term *smart* collusion comes from its similarity to the usage of blockchain smart contracts by colluding players to coordinate with each other in arbitrary ways.

the protocol contains a cryptographic way for players to prove the existence of a collusion, and provides a reward R for submitting such a proof. But now, knowing the existence and details of this whistleblowing mechanism, colluders can adapt. In particular, all colluders can first agree on a contract C that holds large deposits from each player in the collusion and *retaliates* if anyone attempts to defect (from the collusion) by whistleblowing. As a simple example, retaliation might mean burning the deposit of the whistleblower if the mechanism is not anonymized or burning *all* deposits if it is. One potential way to instantiate C is as a blockchain smart contract.

In general, we will allow players to collude in arbitrary ways—both cryptographically to prevent the *ability* to whistleblow, and through designing threats of retaliation to deter players from whistleblowing. In essence, we note that for smart collusion, the same tools available to the protocol designer to protect against collusion will also be available to the colluders in their attempt to circumvent the protocol. Colluders in fact may have a *last-mover advantage* as the protocol designer must generally commit to their mechanism first. To the best of our knowledge, this perspective is missing from prior works on collusion in cryptographic protocols. Most recently, concurrent work by Bormet et al. [17] allows colluders to use a trusted party (such a smart contract) in the secret sharing context—this is however restricted only to one particular "retaliation strategy" as opposed to the general form that our model permits.

Smart collusion can be thought of analogously to real-world retaliation and intimidation tactics used to discourage defection from criminal enterprises. For example, mafia bosses might threaten to harm or kill members (or their families) for violating the code of *omertà* (which mandates, among other principles, a refusal to cooperate with law enforcement). In game-theoretic language, smart collusion enables creating a "credible commitment" [64] to enforcing such penalties on defectors.

1.1 Our Contributions and Results

We make contributions on whistleblowing protocols from both the cryptographic and the mechanism design perspectives. We summarize our main results in this section.

Whistleblowing formalism (Section 3). We start by proposing a general design framework for whistleblowing protocols in Section 3. Abstractly, whistleblowing protocols comprise of a cryptographic tool to prove collusion, and a game-theoretic mechanism to incentivize players to submit such proofs. On the cryptographic side, we define security through typical notions of completeness (some player within the collusion can compute a collusion proof) and soundness (if there is no collusion, no player can compute a collusion proof). On the game-theoretic side, we define security in terms of incentivizing rational players to not collude. This requires ensuring that if players were to collude, they would be incentivized to defect by whistleblowing (i.e., negating the benefits of colluding). To capture this, we use the standard game-theoretic notion of a subgame-perfect pure Nash equilibrium (SPPNE).

Whistleblowing settings (Sections 4 and 5). We formalize two settings that capture several natural cryptographic applications and explore whistleblowing protocols for them.

1. *Blockchain-assisted transparent services (Section 4).* We model a generic distributed cryptographic service provided by a set $\mathcal{S}$ of servers. A key k is secret-shared among the servers. Any client can provide an input x and learn the result of some (potentially randomized) function $f(k, x)$

computed by the servers in a threshold manner. We illustrate how this models natural blockchain-based use cases such as threshold signing-as-a service, randomness beacons [74], and anonymous credentials [38] (see Section 4.4).

For many applications, it can be important to provide *transparency* (i.e., keep a public record of all answered queries) to prevent equivocation or enable auditability. As an example, for anonymous credentials that need to be rate limited, we want to ensure that *even a malicious server quorum* cannot provide extra credentials to a client. A natural first step is to use e.g., a blockchain, to log all queries (potentially in a privacy-preserving way). As an orthogonal point, using the blockchain for communication also provides natural *censorship-resistance* guarantees, similar to its use in [23, 27, 67].

However, even when queries are logged, notice that servers can still collude (individually or together) with a client to provide their service *out-of-band*, i.e., without communicating through the blockchain. A client may want to pay a premium for such out-of-band service that is not recorded (e.g., to receive extra credentials).

Our goal is to disincentivize this out-of-band collusion through whistleblowing. We accomplish this by ensuring that the client will be able to prove that it has received service out-of-band—i.e., the client obtains a whistleblowing proof. We prove general feasibility results on when such whistleblowing proofs can be (cryptographically) constructed, and how they can be incentivized.

2. *General setting with collusion proofs (Section 5).* Second, we model a general setting that captures any protocol where proofs of collusion arise. That is, when players collude, some member necessarily obtains information that allows it to prove the existence of the collusion. Further, we even permit the collusion to decide the exact subset of colluding players who will receive such a proof.

While we don't have generic results for when cryptographic tools for proving collusion exist, we show how this fits several existing protocols including secret sharing with snitching [40, 53], threshold decryption with self-incriminating proofs [22], and BFT forensics [72].

The core reason for modeling this general setting is to show when such proofs of collusion are actually *effective*. We apply our game-theoretic framework and results to highlight the challenge of deterring collusion even given proofs of collusion (especially in light of our smart collusion notion).

Game-theoretic results (Sections 6, 7, 8, and 9). We formalize a general game-theoretic framework (capturing all our settings) that models the incentives of whistleblowing protocols (Section 6). More concretely, to deter collusion, we allow for designing a mechanism $\mathcal{M}^{\mathbb{W}}$ that takes deposit D from each player, and executes appropriate penalties / rewards when whistleblowing proofs are submitted. In turn, following our smart collusion notion, the colluders may themselves effectively write a "retaliation" mechanism $\mathcal{M}^{\mathbb{R}}$ that takes in deposit D' from each colluding player, and executes appropriate "counter-penalties" based on whether whistleblowing proofs were submitted by defectors. For our game-theoretic analysis, we will abstract out the exact details of how whistleblowing proofs are constructed cryptographically. Our highlight results are as follows:

- *Impossibility of unconditional security*. We prove the impossibility of *unconditional* protection against smart collusion. In other words, for *any* whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$, there exists a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ that takes in sufficiently high deposit from colluders and succeeds in deterring them from blowing whistles. Perhaps surprisingly, we show this impossibility holds even when (1) the whistleblowing is *anonymous*, i.e., prevents identification of who is blowing the whistle; and (2) the whistleblowing is *fully accountable*, i.e., provably identifies *all* players in the collusion.

This can be seen in contrast with the positive results in e.g., [53] where standard collusion (i.e., without a retaliation contract) can be deterred when all players are rational. We show a similar feasibility for whistleblowing without retaliation in our broad framework.

Similarly, this impossibility highlights the challenge of *accountable* cryptography (e.g., [11, 12, 14, 15, 25, 54]) which often assumes trusted executors to identify malicious players. Our result may be interpreted as an impossibility for accountability in settings where this executor is rational and could collude with other players.

- *Small whistleblowing deposits can prevent collusion based on much larger retaliation deposits.* On the positive side, we show a concrete whistleblowing mechanism (with anonymous proofs) which requires a deposit D from each player and protects against any smart collusion with much larger deposits—up to roughly $(t-1)D$ from each colluder who has the ability to whistleblow. This result has practical significance especially in settings where n (and therefore, t) are large.

- *Lower bound on the whistleblowing deposit.* Further, as a lower bound, we show that any whistleblowing mechanism that collects a whistleblowing deposit D can be rendered ineffective by some retaliation mechanism collecting a deposit $(t-1)\,D$. This demonstrates that our mechanism is near-optimal.

- *Positive result in the transparent service setting.* We find an interesting nuance that leads to a positive result for natural applications of the transparent service setting. Specifically, our result applies if the servers provide their service to clients on an ongoing basis.

Servers can be expected to let their whistleblowing deposit be held *indefinitely* (until a whistle is blown), effectively as a *cost of doing business.* But now, we show that the only way to break this whistleblowing mechanism is for the servers to force the client's deposit in the retaliation contract to also be held indefinitely. This would be a permanent cost to the client, which would intuitively negate any benefit from colluding to obtain off-band service. In turn, this disincentivizes collusion.

Paper organization. The rest of the paper is organized as follows: Section 1.2 presents related works. Section 2 gives relevant cryptography and mechanism design preliminaries. Section 3 introduces the broad framework for whistleblowing protocols; the cryptographic side is explored in Sections 4 and 5, and the mechanism design side is explored in Sections 6, 7, 8, and 9.

1.2 Additional Related Works

Rational cryptography. A long line of work [5, 47, 56, 57] models the security of various cryptographic protocols under rational adversaries. Here, rationality is modeled in one of two ways: (i) either as a monolithic external attacker but with a budget (which is *weaker* than the typical arbitrary malicious model), or (ii) as all players being individually rational. Collusion amongst players is typically not considered. In our setting, we model individually rational players who can also *collude*, and moreover do so in a new powerful way with smart collusion.

Accountable protocols. Accountability—i.e., identification of malicious players—is a recurring theme in the cryptography literature. Accountable protocols have been put forth in a number of different contexts—traitor tracing [25, 77], threshold signatures (see e.g., [11, 13] and references therein), tracing for threshold decryption [12], detection of failures in trusted applications [3, 61], secret sharing [14, 54], threshold VRFs [15], and BFT forensics [18, 72].

But, typically, these protocols require that a trusted party executes some "accountability procedure" to identify the culprits (except for accountable threshold signatures). Even in cases where this executor does not require private state (such as a tracing key), if the executor is rational, it would still need to be *incentivized* to run this accountability procedure. In other cases (such as accountable signatures), similar to our whistleblowing design, individual parties or clients could still be required to submit their proofs to a public verifier. Consequently, smart collusion (with e.g., the executor) could prevent accountable protocols from working as desired.

Collusion in the mechanism design literature. A sequence of works [1, 7, 10, 42, 63] study collusion-resistance in mechanism design. The literature broadly follows one of the following two approaches. In the first, colluding players merge into a single entity and maximize the coalition's joint utility [21, 36, 50, 55, 68]. In the other, collusion only modifies the payoffs in the game, but players continue to act individually to maximize their own utility [43, 46, 49, 66]. Our work aligns more closely with the latter.

Geffner and Tennenholtz [49] propose to let players place "bets" on information learned about others through collusion as a way to make collusion unstable. Schummer [70] considers bribes for taking specific actions within a collusion, but makes the trust assumption that the briber does not strategically misreport its preferences. In general, smart collusion is not captured since these works do not allow colluders to *extend their strategy space* and *respond* strategically to such anti-collusion mechanisms.

Another relevant thread is the literature on mechanisms with credible commitments [23, 37, 41, 44, 59]. These typically focus on commitments made by an auctioneer, but smart collusion instead uses them to strengthen the collusion.

A recent line of work [28, 46, 68] explores blockchain transaction fee mechanisms (TFMs) which aim to prevent collusion between the miner (auctioneer) and users. But results are largely negative even with standard collusion, and feasibility results using cryptography [73] still require honest majority. Credible commitments between colluders (similar to smart collusion) are not considered.

2 Preliminaries

2.1 Cryptographic Preliminaries

Definition 1 (NIZK-PoK)**.** A non-interactive proof system $(\mathcal{P}, \mathcal{V})$ for a relation R is said to be honest-verifier zero-knowledge proof-of-knowledge if it satisfies the following properties:

- **Completeness.** $\forall\ (X, w) \in R$,
$$\Pr[\mathcal{P}(X, w) \leftrightarrow \mathcal{V}(X) \implies 1] \geq 1 - \mathrm{negl}(\lambda)$$
- **κ-Knowledge soundness.** There exists an extractor $\mathcal{E}$ that runs in expected polynomial time, such that, for every PPT prover $\mathcal{P}^*$ and every statement X,
$$\Pr[(X, w) \in R : w \leftarrow\$\ \mathcal{E}^{\mathcal{P}^*}(X)] \geq \Pr[\mathcal{P}^*(X) \leftrightarrow \mathcal{V}(X) \implies 1] - \kappa$$

- **Honest-verifier zero-knowledge,** There exists an efficient simulator Sim such that

$$\forall\ (X, w) \in R, \{\mathsf{Sim}(X)\} \approx_c \{\text{View}_{\mathcal{V}}(\mathcal{P}(X, w) \leftrightarrow \mathcal{V}(X))\}$$

where $\text{View}_{\mathcal{V}}$ is the verifier's view in the protocol execution.

2.2 Mechanism Design Preliminaries

We will be modeling the games induced by whistleblowing mechanisms as *extended form games.* Intuitively, an extended form game captures a multi-stage game, where each stage requires players to play an action from some action space. The game proceeds differently to the next stages based on different actions taken by the players in each stage of the game.

Definition 2 (Extended form game)**.** An extended form game consists of a directed tree (H, E), where H denotes the set of histories. For players $1, \dots, n$, the game starts at the root with history $h = \emptyset$. For any given history h, the actions available to player i is given by the action space $A_i(h)$. For any actions $(a_1, \dots a_n) \in \Pi_i A_i(h)$ taken by the n players at history h, the game transitions from history h to $h' = \mathcal{T}(h, a_1, \dots, a_n)$ for some $(h, h') \in E$. The game terminates when the history h reaches a terminal vertex (i.e, there exists no $(h, h') \in E$). Player i then receives a utility $u_i(h)$ at termination.

Players are modeled as rational, i.e., each player selfishly tries to optimize its utility received at the end of the game. It is also *common knowledge* that all players are utility-maximizing.

Intuitively, in the simplest case where the game consists of only a single round (i.e, the root $h =$ is also a terminal node), an action profile $(a_1, \dots, a_n) \in \Pi_i A_i(\emptyset)$ is a *Nash equilibrium* if no player i can increase their utility by deviating from playing a_i.

Definition 3 (Pure Nash equilibrium)**.** In a single-round game with action space $A_1, \dots A_n$, an action profile $(a_1, \dots, a_n) \in \Pi_i A_i$ is a pure Nash equilibrium if for all players i and actions $a'_i \neq a_i$ in A_i, i's utility $u_i(a_i, a_{-i})$ is at least $u_i(a'_i, a_{-i})$, where a_{-i} is the action profile $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$ of the players other than i.

In a multi-stage extended form game, player i's strategy $s_i : H \to \bigcup_{h \in H} A_i(h)$ denotes the action it would take at any given point in history. A profile of strategies $(s_1, \dots, s_n)$ is a subgame perfect Nash equilibrium if no player i can increase its utility by switching to a different strategy s'_i.

Definition 4 (Subgame perfect pure Nash equilibrium, SPPNE)**.** For a strategy profile $\vec{s} = (s_1, \dots, s_n)$, let $\vec{s}|_h$ denote the strategy profile $\vec{s}$ restricted to the extended form game with h as the root. Then, $\vec{s}$ is a subgame perfect pure Nash equilibrium if, for all histories h reached by playing the strategy $\vec{s}$, $\vec{s}|_h$ is a pure Nash equilibrium.

We adopt SPPNE as the equilibrium concept for the games induced by the whistleblowing and retaliation mechanisms — we will analyze the induced games for the existence of SPPNEs.

3 Whistleblowing Framework

In this section, we introduce the basic design of whistleblowing protocols. We aim only to provide a general overview here; some details may be left under-specified and will be elaborated on in later sections for our specific settings.

Whistleblowing protocols. As discussed earlier, abstractly, a whistleblowing protocol will consist of two parts: (i) a cryptographic tool that allows a colluding party to prove existence of the collusion and/or its participants to a verifier $\mathbb{V}$; and (ii) a game-theoretic mechanism that is allowed to take deposits from players, will essentially run the verifier $\mathbb{V}$ (to check valid proofs of collusion), and can execute rewards / penalties based on submitted whistleblowing proofs (or just *whistles*).

Definition 5 below provides a general, (informal) definition for whistleblowing protocols, which includes security goals for both the cryptographic and game-theoretic components.

We expand on the cryptographic part in Section 4 and the game-theoretic part in Section 6.

Definition 5 (Whistleblowing Protocol (Informal))**.** Consider a (ideal) functionality $\mathcal{F}$ among the set $\mathcal{P}$ of n players and consider $\mathbb{A} \subseteq 2^{\mathcal{P}}$ (intuitively denoting "allowed" adversarial corruption sets). An α-whistleblowing protocol for $(\mathcal{F}, \mathbb{A})$ is a tuple $(\Pi, \mathbb{P}, \mathbb{V}, \mathcal{M}^{\mathbb{W}})$ where Π is a protocol, $\mathbb{P} \leftrightarrow \mathbb{V}$ is a proof-system, and $\mathcal{M}^{\mathbb{W}}$ is a mechanism as follows.

(I. Cryptography) First, for the cryptographic properties, as is typical, we can model a *monolithic* adversary $\mathcal{A}$ to define security.

- (*Protocol validity or correctness*). Π emulates $\mathcal{F}$ in the presence of an adversary $\mathcal{A}$ that controls a set of players $\mathcal{P}_{\mathcal{A}} \in \mathbb{A}$.
- (*Whistleblowing completeness*) If $\mathcal{A}$ controls players $\mathcal{P}_{\mathcal{A}} \notin \mathbb{A}$ (e.g., $\mathcal{A}$ corrupted more players than expected), then there are at least α distinct players ($\in \mathcal{P}_{\mathcal{A}}$) that can convince $\mathbb{V}$.
- (*Whistleblowing soundness / unframability*) If $\mathcal{A}$ controls players $\mathcal{P}_{\mathcal{A}} \in \mathbb{A}$, then no player $P \in \mathcal{P}$ can convince $\mathbb{V}$.

Here, by "P convinces $\mathbb{V}$", we mean that the verification program $\mathbb{V}$ outputs 1 when interacting with P. Note that $\mathbb{V}$ will be provided with the public setup parameters from Π.

We say that the whistleblowing protocol is anonymous if $\mathbb{V}$ cannot distinguish which player submitted a valid proof. Further, it is *β-identifiable* if, as part of the proof to $\mathbb{V}$, it can correctly identify β players in $\mathcal{P}_{\mathcal{A}}$ (without framing honest ones in $\mathcal{P} \setminus \mathcal{P}_{\mathcal{A}}$).

Further details are given in Sections 4 and 5.

(II. Mechanism Design) Here, we will model each player as independent and rational. Each player can can choose one of two actions, "collude" and "honest" (intuitively, if P plays "collude", from the cryptographic perspective, P will be considered part of the adversary $\mathcal{A}$).

We allow the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ to take deposits from each player, view whistles that were submitted, and execute rewards / penalties. Following our smart-collusion notion, colluders themselves will be allowed to collude using a mechanism $\mathcal{M}^{\mathbb{R}}$ of their choice. Now, security is defined in the following sense:

- (*Not colluding is rational*) For all $\mathcal{M}^{\mathbb{R}}$ according to which collusion can take place, the subgame-perfect pure Nash equilibrium for the game induced by $(\mathcal{M}^{\mathbb{W}}, \mathcal{M}^{\mathbb{R}})$ is where everyone plays "honest."

Section 6 provides a more rigorous description.

Remark 1 (Processing rewards). Rewarding whistleblowing proofs, even when they are anonymous, is straightforward with cryptocurrency. For example, submission of a whistle can be accompanied with a fresh cryptocurrency wallet address (owned by the submitter) to which any reward from $\mathcal{M}^{\mathbb{W}}$ will be sent.

4 Transparent Service Model

We now describe a formal model for whistleblowing in our transparent service setting.

Basic model. We model three entities: a set of n service providers (or servers) $\mathcal{S}_1, \ldots, \mathcal{S}_n$, a user (or client) $\mathcal{U}$, and the logger $\mathcal{L}$ (e.g., a blockchain). The servers $\{\mathcal{S}_i\}$ collectively provide users with a computational functionality for a (potentially randomized) function f such that the user can learn $f(x)$ on input x by interacting with any t servers. For our formalism, since each user query is done separately, it is sufficient to only model one user. The goal for $\mathcal{L}$ is to (publicly) log *all* user queries (potentially in a privacy-preserving way)—intuitively, this will happen by parties communicating through $\mathcal{L}$. However, since the servers $\{\mathcal{S}_i\}$ can always communicate with users off-band (i.e., circumventing $\mathcal{L}$), the goal is to make it incentive-compatible to use $\mathcal{L}$. The cryptographic part of this is to ensure that the user can *whistleblow* by proving that it received off-band service.

We start by formally defining a generic computational service provided by the servers $\{\mathcal{S}_i\}$. This captures a broad range of applications, which we discuss in more detail in Section 4.4. In Section 4.2, we present a formal definition of a whistleblowing protocol for off-band collusion. Later, we characterize what functionalities admit whistleblowing, and present concrete protocols.

4.1 A Generic Computation Functionality

Definition 6 (Computation Functionality Syntax)**.** Consider a (randomized) function $f : \mathcal{K} \times \mathcal{X} \times \mathcal{R}_S \to \mathcal{Y}$ over key space $\mathcal{K}$, input space $\mathcal{X}$, randomness space $\mathcal{R}_S$, and output space $\mathcal{Y}$. We write $(x, y) \in f_k$ if there exists r_s such that $f(k, x, r_s) = y$.

We focus on a threshold functionality, wherein the evaluation key is secret-shared among the n servers such that the client can communicate with any t servers to compute the function.

Formally, a computational functionality $\mathcal{F}$ between the servers $\{\mathcal{S}_i\}$ and a user $\mathcal{U}$ for computing f is a tuple $(\mathsf{Gen}, \mathsf{Query}, \mathsf{Respond}, \mathsf{Output}, \mathsf{PartialVer}, \mathsf{Ver})$ of algorithms as follows:

- $\mathcal{F}.\mathsf{Gen}(1^\lambda, t, n) \to (k_1, \ldots, k_n, \mathsf{vk})$ is a (randomized) key generation algorithm run either by a trusted party or by MPC between the servers $\{\mathcal{S}_i\}$. It generates a key pair $(k, \mathsf{vk}) \in \mathcal{K} \times \mathcal{K}_V$ along with the secret shares $\{k_i\}$ of the key k that will be stored by the servers $\{\mathcal{S}_i\}$.

- $\mathcal{F}.\mathsf{Query}(x; r^c) \to x^*$ is a (randomized) algorithm run by the client that takes as input $x \in \mathcal{X}$ and randomness r_c and outputs a leakage x^* of the input. We will formalize the notion of leakage later in this section.

- $\mathcal{F}.\mathsf{Respond}(k_i, x^*; r_i^s) \to (y_i^*, \pi_i)$ is a (randomized) algorithm run by the ith server $\mathcal{S}_i$ that takes as input the key share k_i, the leakage x^*, and randomness r_s and outputs an intermediate result y_i^* and a proof π_i.

- $\mathcal{F}.\mathsf{Output}(\{(y_{i_1}^*, \pi_{i_1}), \ldots, (y_{i_t}^*, \pi_{i_t})\}; r^c) \to (y, \pi)$ is a deterministic algorithm run by the client that takes as input the intermediate results $\{y_{i_1}^*, \ldots, y_{i_t}^*\}$ and randomness r_c, and computes the final output $y \in \mathcal{Y}$ along with a proof π. If the result is *self-verifiable*, the proof might be empty: $\pi = \emptyset$.

- $\mathcal{F}.\mathsf{PartialVer}(\mathsf{vk}, x^*, y_i^*, \pi_i) \to \{0, 1\}$ is a deterministic verification algorithm which takes as input vk, x^*, y_i^*, and a proof π_i and outputs a single bit denoting whether the response of the ith server was valid.

- $\mathcal{F}.\mathsf{Ver}(\mathsf{vk}, x, y, \pi) \rightarrow \{0, 1\}$ is a deterministic verification algorithm which takes as input vk, x, y, and a proof π and outputs a single bit denoting whether y is indeed the correct function evaluation at x, i.e. $(x, y) \in f_k$.

Leakage. Intuitively, x^*, y_i^* capture the "leakage" of the function – i.e., the values recorded by $\mathcal{L}$. For input x and randomness r^c, we will use $\mathsf{leak}_{\mathcal{F}}(x; r^c)$ to denote the output x^* of $\mathcal{F}.\mathsf{Query}(x; r^c)$. We further use $\mathsf{leak}_{\mathcal{F}}(x)$ to denote the distribution $\{\mathsf{leak}_{\mathcal{F}}(x; r_c)\}_{r_c \in \mathcal{R}_C}$ and $\mathsf{leak}_{\mathcal{F}}(X)$ to denote the joint distribution of $\mathsf{leak}_{\mathcal{F}}(x)$ for all $x \in X$. We often drop the subscript when $\mathcal{F}$ is clear from context.

Of particular interest to us will be the three classes of leakage functions, which we now describe.

- *Public.* A public leakage function reveals x to $\mathcal{S}$ and $\mathcal{L}$. Formally, $x^* = x$ and $\mathcal{R}_C = \perp$. This captures functionalities such as signatures, VRFs, decryption, and Identity-based encryption (IBE) enrollment (see Section 4.4 for details).

- *Oblivious Leakage.* Intuitively, oblivious means that the leakage function reveals nothing about x. This holds even if the adversary $\mathcal{A}$ is allowed to see evaluations of the function at arbitrary inputs of its choice. We formalize this as a standard indistinguishability game where $\mathcal{A}$ chooses two inputs x_0, x_1, is given back $\mathsf{leak}(x_b)$, and wins if it correctly guesses b (see Definition 7).

Definition 7. A functionality $\mathcal{F}$ has $\epsilon(\lambda)$-bounded oblivious leakage if for all $\lambda \in \mathbb{N}$ and all adversaries $\mathcal{A}$ (making $\mathsf{poly}(\lambda)$ queries to $\mathcal{F}$), the following expression is bounded by $\epsilon(\lambda)$:

$$\mathsf{Adv}_{\mathcal{F}}^{\mathrm{ob-leak}}(\mathcal{A}) = \left| \Pr\left[(\mathcal{A}(l_b) \Rightarrow b') = b \;\middle|\; \substack{b \leftarrow\$ \{0,1\} \\ x_0, x_1 \leftarrow \mathcal{A}(1^\lambda) \\ r \leftarrow\$ \mathcal{R}_C \\ l_b = \mathsf{leak}(x_b; r)} \right] - \frac{1}{2} \right|.$$

Applications such as anonymous credentials and blind signatures have oblivious leakage. Later in this section, we will prove that unfortunately, it is impossible to whistleblow for such applications.

- *One-way function leakage.* This captures a weaker but still useful notion of privacy. Intuitively, the leakage function reveals some information about x, but not enough to recover x. Concretely, for a random input x, given $\mathsf{leak}(x)$, it is difficult for any PPT adversary to compute x. This is formalized in Definition 8.

Definition 8. A functionality $\mathcal{F}$ has one-way function leakage if, for all $\lambda \in \mathbb{N}$ and all PPT adversaries $\mathcal{A}$, the following advantage is negligible in λ:

$$\mathsf{Adv}_{\mathcal{F}}^{\mathrm{ow-leak}}(\lambda)(\mathcal{A}) = \Pr\left[\mathcal{A}(l) \Rightarrow x \;\middle|\; \substack{x, r \leftarrow\$ \mathcal{X} \times \mathcal{R}_C \\ l = \mathsf{leak}(x; r)} \right].$$

While we do not find applications that inherently have such leakage, in Section 4.3.3, we show how we can support whistleblowing for applications (such as anonymous credentials) with oblivious leakage by weakening the privacy guarantee to one-way leakage.

Security properties. Security of the functionality $\mathcal{F}$ can be formulated as natural correctness and soundness properties.

Informally, correctness means that if all servers are honest, the client with input x will receive the correct output $f(x)$ and can verify this. Further, soundness requires that it should be hard for a client to compute the function at a new input $\hat{x}$, even if it can corrupt up to $t-1$ servers, and has oracle access to query other arbitrary inputs. This can be seen as a generalization of the typical *unforgeability* definition for e.g., signatures.

Since the ways to formalize these properties use relatively standard techniques, and our focus is instead on *whistleblowing* protocols for $\mathcal{F}$, we defer the full formal definitions to Appendix A.

4.2 Whistleblowing Formalism

Recall that our goal is to prevent off-band collusion between any t servers $\{\mathcal{S}_i\}$ and a client. To facilitate this, a first step is to allow the client to report off-band service provided by the servers $\{\mathcal{S}_i\}$. In this section, we detail formalism for how such *whistleblowing* can take place. Intuitively, a whistleblowing protocol will allow a user to convince the logger $\mathcal{L}$ exactly when it possesses a valid function evaluation (x, y) that was not recorded on $\mathcal{L}$. This can be used as an indication that at least t of the n servers are involved in providing off-band services. For simplicity, we focus on the functionalities where there is no way to detect which t out of n servers were responsible for providing off-band service. This fits several applications like (non-accountable) threshold signatures, VRFs, and IBE enrollment. Extending the formalism to identifiable whistleblowing (which applies to e.g., accountable threshold signatures) is straightforward (see Appendix A.)

Definition 9 (Whistleblowing for off-band service)**.** Functionality $\mathcal{F} = (\mathsf{Gen}, \mathsf{Query}, \mathsf{Respond}, \mathsf{Output}, \mathsf{Ver})$ cryptographically supports a whistleblowing protocol $\mathcal{W}_{\mathcal{F}}$ if its proof system $(\mathbb{P}, \mathbb{V})$ satisfies the following:

- *Completeness.* Intuitively, completeness means that a client with access to a valid function evaluation (x, y) such that a previous query for any x^* in $\mathsf{leak}_{\mathcal{F}}(x)$ was not logged by $\mathcal{L}$ (i.e., this was the result of off-band communication with $\mathcal{S}$), should be able to convince the verifier of this. More formally, for all $(k_1, \ldots, k_n, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda, t, n)$, $L \subseteq \mathsf{leak}_{\mathcal{F}}(\mathcal{X})$, $x \in \mathcal{X}$ such that $\forall r^c \in \mathcal{R}_C, \mathsf{leak}_{\mathcal{F}}(x; r^c) \notin L$ and y such that $(x, y) \in f_k$, the following holds:

$$\Pr\left[\mathbb{P}((L, \mathsf{vk}), (x, y)) \leftrightarrow \mathbb{V}(L, \mathsf{vk}) \Rightarrow 1\right] = 1$$

where the probability is taken over the randomness of $\mathbb{P}$.

- *Unframability.* Intuitively, unframeability guarantees that a client cannot falsely frame the servers, even if it colludes with upto $t-1$ servers. Note that this is reasonable since it is hard for the client to compute the function on any input if less than t servers offer off-band service.

Definition 10. A whistleblowing protocol $\mathcal{W}_{\mathcal{F}}$ for a functionality $\mathcal{F}$ is said to be unframeable if the following is negligible in λ for all PPT adversaries $\mathcal{A}$:

$$\mathsf{Adv}^{\text{unfram}}_{\mathcal{W}_{\mathcal{F}}}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{unfram}}_{\mathcal{W}_{\mathcal{F}}}(\mathcal{A}) = 1]$$

where $\mathbf{G}^{\text{unfram}}_{\mathcal{W}_{\mathcal{F}}}$ is as defined in Figure 1.

We further say that $\mathcal{W}_{\mathcal{F}}$ is *anonymous* if the following holds:

```
Game $\mathbf{G}^{\text{unfram}}_{\mathcal{W}_\mathcal{F}}$
1 :  $(\mathsf{st}, n, t, \mathcal{C}) \leftarrow \mathcal{A}(1^\lambda)$
2 :  $(k_1, \ldots, k_n, \mathsf{vk}) \leftarrow\!\!\$\ \mathcal{F}.\mathsf{Gen}(1^\lambda, t, n)$
3 :  $\Pi^* \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{EvalO}(\cdot)}(\mathsf{st}, \{k_i\}_{i \in \mathcal{C}}, \mathsf{vk})$
4 :  return $\mathbb{V}(\mathcal{Q}, \mathsf{vk}, \Pi^*) = 1 \land |\mathcal{C}| < t$

    Oracle $\mathsf{EvalO}(i, x^*)$
    1 :  $r_i^s \leftarrow\!\!\$\ \mathcal{R}_S$
    2 :  $y_i^*, \pi_i \leftarrow \mathcal{F}.\mathsf{Respond}(k_i, x^*; r_i^s)$
    3 :  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{x^*\}$
    4 :  return $\sigma$
```

Figure 1: The unframability game for a whistleblowing protocol $\mathcal{W}_\mathcal{F}$ and a functionality $\mathcal{F}$.

- (Anonymity) For all $(k, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$, $L \subset \mathsf{leak}_\mathcal{F}(\mathcal{X})$, $x_1 \neq x_2$ such that $\forall r^c \in \mathcal{R}_C, \mathsf{leak}(x_1; r^c) \notin L \land \mathsf{leak}(x_2; r^c) \notin L$, and y_1, y_2 such that $(x_1, y_1) \in f_k$ and $(x_2, y_2) \in f_k$, the two ensembles are computationally indistinguishable:

$$\{\mathrm{View}_\mathbb{V}(\mathbb{P}((L, \mathsf{vk}), (x_1, y_1)) \leftrightarrow \mathbb{V}(L, \mathsf{vk}))\}$$
$$\approx_c \{\mathrm{View}_\mathbb{V}(\mathbb{P}((L, \mathsf{vk}), (x_2, y_2)) \leftrightarrow \mathbb{V}(L, \mathsf{vk}))\}$$

where the distributions are over the randomness of the prover. Intuitively, anonymity implies that servers $\{\mathcal{S}_i\}$ learn nothing about which input was used for the whistleblowing.

4.3 Results on Whistleblowing Feasibility

We first look at whether whistleblowing is possible based on what is leaked (about the client query) to the servers $\{\mathcal{S}_i\}$ and the logger $\mathcal{L}$ by the functionality $\mathcal{F}$. Recall that $\mathsf{leak}_\mathcal{F}(x)$ denotes the distribution of the output $\mathcal{F}.\mathsf{Query}(x; r^c)$ over client randomness r^c.

4.3.1 Oblivious Leakage does not permit Whistleblowing

We show a simple result that whistleblowing (whether anonymous or not) is not possible for functionalities where the leakage is oblivious. Intuitively, this holds because oblivious leakage implies that no adversary $\mathcal{A}$ can distinguish between $\mathsf{leak}_\mathcal{F}(x_0)$ and $\mathsf{leak}_\mathcal{F}(x_1)$. This implies that a user should be able to whistleblow with both x_0 and x_1 regardless of whether $\mathsf{leak}(x_0)$ or $\mathsf{leak}(x_1)$ was recorded on $\mathcal{L}$. But this contradicts the soundness property.

Lemma 4.1. *Consider any computational functionality $\mathcal{F}$ with leakage $\mathsf{leak}(\cdot)$ that is ϵ-oblivious (where $\epsilon = \epsilon(\lambda)$ is a negligible function). Then, no whistleblowing protocol with $(\mathbb{P}, \mathbb{V})$ can be ϵ-sound.*

The above lemma implies that it is impossible to support whistleblowing for functionalities like blind signatures and anonymous credentials (and their applications), which have oblivious leakage. A full proof is given in Appendix B. Nevertheless, in Section 4.3.3 we show how whistleblowing can be supported for some applications by weakening their privacy guarantee to one-way function leakage.

4.3.2 Whistleblowing with other leakages

We now present a protocol which works for *self-verifiable* functionalities with public leakage, as well as those with one-way leakage and small randomness space, i.e., $|\mathcal{R}_C|$ is polynomial in λ.

At a high level, if the client has a valid evaluation (x, y) for an input x not logged on $\mathcal{L}$, then it can use a NIZK-PoK to prove this to the logger. More formally, let $0 < t < n \in \mathbb{N}$ and $(k_1, \ldots, k_n, \mathsf{vk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda, t, n)$. Let L be the set of input leakages that have been recorded on the logger $\mathcal{L}$ so far. Suppose that a party is in possession of (x, y) such that $\mathsf{Ver}(\mathsf{vk}, x, y) = 1$ and $\forall r^c \in \mathcal{R}_C, \mathsf{leak}(x; r^c) \notin L$. Note that the Ver algorithm does not take a proof π as input since we are only considering self-verifiable functionalities wherein $\pi = \perp$ (we explain why we need this restriction in Remark 2). Now, to perform whistleblowing, the party computes a NIZK-PoK proof for the following relation and sends it to the verifier:

$$\mathcal{R} = \{(\mathsf{vk}, (x, y)) : \mathsf{Ver}(x, y, \mathsf{vk}) = 1 \land \forall r \in \mathcal{R}_C, \mathsf{leak}(x; r) \notin L\}$$

For the above relation to be polynomial-sized, we require the size of the randomness space $\mathcal{R}_C$ to be polynomial in the security parameter λ. This is trivially true for any functionality with public leakage, but this does restrict the functionalities with one-way leakage that we can support with the above protocol. We defer the full proof that this protocol is secure and anonymous to Appendix B.

Remark 2 (Generalizing to functionalities with verification proofs)*.* The above whistleblowing protocol works only for *self-verifiable* functionalities, i.e., where the proof π is empty, and it is possible to directly verify the evaluation pair (x, y) by just running the Ver algorithm. The reason is that there is a simple impossibility result against whistleblowing for functionalities where a special non-empty π is required. Specifically, the servers can choose to only provide the intermediate results y_i^* off-band, with either no proof, or just a designated verifier proof. This allows the client to compute the evaluation y, but not the evaluation proof. As a consequence, there is no efficient way for the client to prove to the logger $\mathcal{L}$ that y is indeed the correct evaluation at x.

4.3.3 Supporting whistleblowing for anonymous credentials by weakening privacy

In Section 4.3.1, we showed that it is impossible for functionalities with oblivious leakage to support whistleblowing. Nevertheless, we observe that for certain functionalities, it is possible to weaken the privacy guarantee to one-way leakage, without breaking the application. Specifically, our technique works for self-verifiable functionalities like blind signatures, if the input comes from a high-entropy space. This weakening, though not ideal, allows us to circumvent the impossibility result for blind signatures, which are used in anonymous credentials as well as privacy pass.

Specifically, we modify the functionality as follows. If the input has high entropy, we require the client to send a one-way function $z = \mathsf{owf}(x)$ of its input x in addition to the query $x^* \in \mathsf{leak}(x)$, along with a zero-knowledge proof of knowledge that it knows x, r^c used to generate x^* and z. More formally, it sends a NIZK-PoK of the following relation:

$$R_q = \left\{ ((z, x^*), (x, r^c)) : \begin{array}{c} x \in \mathcal{X}, r^c \in \mathcal{R}^C \land \\ x^* = \mathsf{leak}(x; r^c) \land z = \mathsf{owf}(x) \end{array} \right\}$$

.

Note that the servers only learn a one-way function of the client's input (and a zero-knowledge proof), and since we are considering an input space with high entropy, this functionality now has one-way leakage. Next, whenever the client wants to use the functionality output (x, y) (which is a blind signature in our example), it cannot simply reveal the tuple (x, y), because the servers can then trivially link this x back to one of the past queries by comparing $\mathsf{owf}(x)$ with the z values of past queries. We get around this by modifying how the clients use their functionality output (x, y). Specifically, instead of directly revealing (x, y), the client can provide a non-interactive zero-knowledge proof of knowledge (NIZK-PoK) of such a tuple. This can also be extended to scenarios where the tuple (x, y) must be used only once (e.g. single-use tokens in privacy pass) by requiring the client to provide a different one-way function $\mathsf{owf}_2(x)$ of x when using the functionality output (x, y).

Lastly, the logger $\mathcal{L}$ will store the list L_z of $\{z = \mathsf{owf}(x)\}$ for all the inputs queried by the client, and the whistleblowing protocol can be run with respect to L_z, as described in Section 4.3.2.

If the input comes from a lower entropy space, then we can replace $\mathsf{owf}(x)$ (and $\mathsf{owf}_2(x)$) in the above technique to $\mathsf{owf}(x, r)$ (and $\mathsf{owf}_2(x, r)$), where r comes from a poly-sized space $\mathcal{R}$ (this is important for the efficiency of the whistleblowing protocol given in Section 4.3.2). This can be used to provide *concrete* security guarantees, in cases where it is possible to brute force the space of inputs $\mathcal{X}$ and randomness $\mathcal{R}$ separately, *but not together* (i.e., $\mathcal{X} \times \mathcal{R}$ is infeasible to brute force).

Note that this would not work for some applications of anonymous credentials such as PAKE and PSI, because leaking any one-way function of the inputs would trivially break the applications. We expand more on this in Appendix A.

4.4 Applications

We will now outline a number of natural applications that can benefit from whistleblowing to detect off-band collusion. While we describe them as distributed applications, note that clients will obtain similar collusion-proofs even with a single service provider.

We consider whistleblowing feasibility only from a cryptographic lens here. For deployment, additional constraints (e.g., estimating collusion utility) may come up. We defer discussion to Section 9.

Threshold signatures and variants. A threshold signature scheme allows n servers to sign a message only if at least t of them participate in the signing process. Signature validity can be directly verified by the client, with efficient schemes featuring constant signature size even as n increases.

Our framework can model a use-case where servers are supposed to provide a public signing service (e.g., a notary) via a blockchain. Here, off-band communication could allow the notary to equivocate on e.g., conflicting documents. Through whistleblowing, a client can easily submit an anonymous whistleblowing proof by showing that it knows a signature on a message that was not logged.

This setting can be applied to all signature varieties. For instance, a private threshold signature reveals nothing about the threshold t or the quorum of t parties that generated the signature. These have applicability when there is a need to hide the inner-workings of an organization (e.g., to avoid leaking exactly what an attacker needs to compromise). On the other hand, accountable signatures [13, 54] identify the exact quorum of servers that generated the signature. These have wide applicability in finance and blockchain (e.g.,the Bitcoin multisig [4]) where accountability is essential. It can also apply to weighted threshold signatures [48], in which a non-uniform number of servers can sign so long as their cumulative weight is above a threshold.

For the above, our whistleblowing protocol from Section 4.3.2 (with public leakage) will directly apply.

Verifiable random functions (VRFs) and distributed randomness beacons (DRBs). A VRF [62] is a pseudorandom function[2] whose output can be publicly verified. In a threshold VRF, the function evaluation key is secret shared among n parties, and a quorum of t parties is needed to evaluate the VRF. VRFs can also be viewed as private threshold signatures with a unique signature for every message. VRFs are used broadly in transparency systems to support privacy by obscuring e.g., user names or domain names while ensuring they are uniquely mapped to a pseudorandom identifier.

For our purpose, we can consider a *threshold* VRF used to construct a distributed randomness beacon [24], as deployed in practice by Drand [74]. Here, a committee evaluates the VRF to generate verifiable randomness for downstream applications to use. For applications to function correctly, it is crucial that this randomness is not leaked through e.g., off-band collusion. Our whistleblowing protocol (for public leakage) can be directly used here. Note however, that some VRFs are not self-verifiable (for example [15, 33]). Hence, our techniques only apply to DRBs based on certain VRFs [58, 74]. Further details on this application are given in Section 9.2.2.

Threshold Blind signatures and anonymous credentials. A threshold blind signature scheme is an interactive protocol involving a client and n servers with shares of a signing key, wherein the client can get a message signed by any t servers, but without revealing the contents of the message or signature to the servers. Blind signatures were originally proposed for anonymous digital cash [20] and have since been proposed for many other applications (see [45]), but we focus on the use case of servers issuing *anonymous credentials* [38]. Such systems allow users to obtain credentials privately, and later prove possession of credentials without revealing any other information about themselves.

Often, applications can necessitate that this issuance is somehow rate-limited (e.g., for Privacy Pass [34]). For this, it is important to ensure that servers do not collude with the client off-band to issue it extra credentials. Our whistleblowing framework would apply here. However, as the name might suggest, blind signatures (and in turn, anonymous credentials) have oblivious leakage—i.e., leak no information to the servers about the message being signed. Fortunately, we can use the technique described in Section 4.3.3 to support whistleblowing by weakening the privacy guarantee to one-way leakage instead of obliviousness.

Oblivious PRFs. Oblivious PRFs (OPRFs) [19] are similar to blind signatures, except the output is the PRF evaluation of the input instead of a signature. Unlike signatures, PRFs do not necessarily support efficient proofs of correct evaluation. OPRFs are useful anywhere a hash function is used that should be rate-limited, for example in password-based encryption or for contact discovery in messaging applications via private set intersection (PSI). While OPRFs have oblivious leakage, we can support whistleblowing for self-verifiable OPRF as a service by relaxing the privacy

[2]There is also the closely related notion of a verifiable unpredictable function (VUF) whose output is not necessarily pseudorandom but cannot be predicted completely.

to one-way leakage as before. However, for many applications that use OPRFs—such as private set intersection or password authenticated key exchange—this weakening of privacy could result in trivial breaks. This makes it challenging for practical applications based on OPRFs to support whistleblowing. We highlight an example in Appendix A.

5 General n-party Model

We now describe our second model, which applies to general n-party protocols with whistleblowing support. Specifically, we model a set of n parties $\mathcal{T}_1, \ldots, \mathcal{T}_n$ running a protocol Π (e.g., secret sharing or threshold decryption), wherein (i) security of the protocol holds as long as less than t parties collude; and (ii) if $\geq t$ parties collude, some of them get access to a publicly verifiable proof of the existence of collusion. Abstractly, we can divide Π into three sub-protocols:

- $\Pi.\mathsf{Gen}(1^\lambda, t, n) \to (k_1, \ldots, k_n, \mathsf{vk}, \mathbf{F})$ is a randomized protocol run between a trusted dealer and all the parties. A distributed procedure could also be used. The output of the protocol includes secret shares $\{k_i\}$ of a secret value k to be stored by the parties $\{\mathcal{T}_i\}$, along with a public verification key vk which will be used to verify collusion proofs and $\mathbf{F}$, which characterizes the set of functions with the key k as input, that the parties are allowed to compute.

- $\Pi.\mathsf{Output}(f, k_{i_1}, \ldots, k_{i_t}) \to y$ is a (potentially randomized) protocol executed between any t parties, that takes as input a function description $f \in \mathbf{F}$, and outputs $f(k)$ where k is the value secret-shared among the parties.

- $\Pi.\mathsf{Ver}(\mathsf{vk}, \pi) \to \{0,1\}$ is a deterministic algorithm that takes as input a proof of collusion π, and outputs a bit denoting its validity. Note that this collusion proof is publicly verifiable, and hence can be checked by a public smart contract.

Security properties (informal). In this model, a protocol with whistleblowing support, analogous to our description in Definition 5, will satisfy three properties: correctness, whistleblowing completeness, and unframability. We provide only informal descriptions below. These are relatively straightforward generalizations of properties from [40] to arbitrary protocols. Our focus in this paper is instead to highlight how other applications also fit this whistleblowing framework.

Correctness states that if all the parties are honest, then the Output protocol outputs the correct value, i.e. $f(k)$.

Whistleblowing completeness guarantees that if a set of atleast t parties collude to learn some information about the secret k, then, at least one of the colluding parties can generate a publicly verifiable proof to either prove the existence of a collusion, or implicate another colluding party. This generalizes the punishability notion in [40].

Unframability can be formalized in two ways. The first applies to protocols wherein there is no way to detect which parties took part in the collusion. For such protocols, unframability requires that even if upto $t-1$ parties collude, they should not be able to produce a valid proof of collusion. The second notion applies to protocols where the collusion proof identifies atleast one colluding party. In this scenario, unframability states that even if upto $n-1$ parties collude with the trusted dealer (if any), they cannot blame any honest party for collusion. The formal definition for both notions are direct extensions of the unframability notion defined in Section 4.

5.1 Applications

We highlight several protocols that adhere to the above model. Our goal is to unify these different applications under our umbrella of whistleblowing, enabling the application of our game-theoretic insights to all of them (see Sections 6 and 9 for further details).

Secret sharing with snitching. A secret sharing scheme allows storing sensitive information among n parties such that any t of them can reconstruct the secret. Secret sharing is widely used for storage, MPC, as well as threshold cryptography. In all of these applications, any t parties can collude to learn something about the secret. Recently, there has been work on developing secret sharing schemes with whistleblowing capabilities [40, 53], such that at least one of the colluders learns a publicly verifiable proof implicating a colluding party. This is directly captured by our model, with the function space $\mathbf{F}$ containing just the identity function $f(x) = x$.

Threshold decryption. A threshold decryption scheme shares a decryption key among n parties such that any t of them can decrypt a ciphertext. It has applications in voting [30], auctions [69] and more recently, encrypted mempools in several blockchain projects [8, 65, 75] to protect against frontrunning MEV attacks [32]. But these protections are lost if t or more decrypting parties collude. While standard threshold decryption has no way to identify or prove that a collusion occurred, there has been recent work [22] on building threshold decryption schemes wherein if any t parties collude to illegally decrypt a ciphertext, at least one of them learns a proof of collusion. Collusion here is different from secret sharing, since the colluders only learn the decryption of a ciphertext, and not the full secret key. Our framework can be used to model this, with the function space $\mathbf{F}$ being the decryption for any ciphertext.

BFT forensics. Byzantine fault tolerant (BFT) consensus is a core primitive in the distributed systems literature. Intuitively, consensus protocols ensure that a distributed set of n servers agree on the same execution state. Typically, in asynchronous networks, these protocols require that the adversary corrupts upto $t' < n/3$ servers. Sheng et al. [72] show how some consensus protocols [2, 76] can support strong *forensics*—even if the adversary corrupts more than t' (but upto $2t'$) servers, the transcript of a single server can be used to detect adversarial servers. A similar model is also explored in [18] for permissionless protocols. In our context, such a transcript can function as a whistleblowing proof of collusion. Note that the adversarial coalition may employ smart collusion to prevent servers from revealing their transcript to the forensics procedure.

6 Game-Theoretic Model

We formally model the incentives associated with collusion and whistleblowing in this section. We consider a set of n players $1, \dots, n$ participating in a generic functionality where the players can receive an additional utility of V per player through collusion. Our core model in this section formalizes (n, n)-collusion — all n out of the n players need to collude in order to earn the reward nV. Section 8 extends the model to (t, n)-collusion (where it suffices that t players collude) and the transparent service setting.

Our goal is to design whistleblowing mechanisms that deter players from colluding and forming coalitions. Formally, we want to ensure that *all* subgame perfect pure Nash equilibria in the whistleblowing model involves players not colluding.

Model and game progression. We describe three (increasingly layered) models to arrive at the final model induced by the whistleblowing mechanism and smartly-colluding players.

1. The *base game* $\mathbb{B}$ considers only the incentives from colluding in the absence of whistleblowing — either all players collude and receive a utility V per player or some player remains honest, in which case, each player receives a zero utility.

2. The *whistleblowing game* $\mathbb{W}$ models collusion in two stages when the protocol designer implements a whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ to counteract collusion — players decide whether to collude or not in the first stage, and whether to snitch on each other and submit whistleblowing proofs (aka blow whistles) to $\mathcal{M}^{\mathbb{W}}$ in the second stage. This would result in slashing the players for colluding and rewarding the whistleblowers.

3. The *retaliation game* $\mathbb{R}$ extends $\mathbb{W}$ by letting the players agree upon a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ that could penalize players for ratting each other out to $\mathcal{M}^{\mathbb{W}}$ and could also redistribute the whistleblowing rewards amongst them in case some player submits a whistleblowing proof. This is our final model for smart collusion.

Remark 3 (Modeling choices). For concreteness of our game-theoretic results, when building our model, we made several design choices which we thought best represent a practical setting. Interestingly, we found that the alternate path in many of our choices to identical qualitative results. We encourage the reader to also look at our exploration into alternate collusion models in Section 7.3 and Appendix D.

Results. Before delving into the details of the three layers, we provide a brief overview of our results. We argue that *no* whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ can provide unconditional economic security — there always exists some retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ such that players can reach a subgame perfect pure Nash equilibrium by colluding in the first stage. Intuitively, $\mathcal{M}^{\mathbb{R}}$ could collect a deposit $D^{\mathbb{R}}$ orders of magnitude larger than the whistleblowing rewards and the gains V from colluding and threaten to burn all of them even if a single whistle is blown in $\mathcal{M}^{\mathbb{W}}$. This would nudge players to collude, but not blow any whistles since they would end up with an extremely negative utility otherwise. However, such a retaliation mechanism would require an extremely large deposit to be collected from the players upfront. We investigate the natural goal of designing whistleblowing mechanisms that are resilient to retaliation mechanisms that collect a deposit of up to $D^{\mathbb{R}}$ per player.

We formally describe the three layers next.

6.1 Base game $\mathbb{B}$

First, we consider the simple single-shot game $\mathbb{B}$ with no whistleblowing protocol in place, where players choose either to collude or to remain honest. If all players choose to collude, then each of them receives a utility V. Otherwise, everybody receives zero utility.

Formally, each player i plays an action $b_i \in \{\mathsf{collude}, \mathsf{honest}\}$ for $1 \leq i \leq n$. Player i's utility function is then given by

$$U_i^{\mathbb{B}}(b_1, \ldots, b_n) = \begin{cases} V & b_1 = \cdots = b_n = \mathsf{collude}, \\ 0 & \text{otherwise.} \end{cases}$$

Note that all players colluding is a pure Nash equilibrium in the base game $\mathbb{B}$ — changing their action from $\mathsf{collude}$ to honest will only decrease their utility from V to 0. Further, it is also the utility-maximizing outcome for the n players.

6.2 Whistleblowing game $\mathbb{W}$

Next, we consider the game $\mathbb{W}$ induced by the protocol designer implementing a whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ to deter collusion. Players cannot deploy smart collusion contracts here. We discuss $\mathbb{W}$ as a warm up for the general game $\mathbb{R}$ that models smart collusion.

When the n players collude in the base game, suppose that some k of them learn a whistleblowing proof π. These k players can submit the whistleblowing proofs to $\mathcal{M}^{\mathbb{W}}$ (i.e., blow whistles) for a reward. To disincentivize players from colluding, the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ should slash the n players upon receiving whistles. However, the rewards and slashes should be designed so that players are not discouraged to submit whistles in the first place, which could happen if the disutility from the slash far outweighs the utility from the reward. Note that retaliatory measures taken by the players to deter whistleblowing is not considered in the whistleblowing game $\mathbb{W}$.

The whistleblowing mechanism. We begin by discussing the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ before describing the game it induces. Formally, $\mathcal{M}^{\mathbb{W}} = (\mathbf{P}^{\mathbb{W}}, \mathbf{S}^{\mathbb{W}}) : \mathbb{N} \cup \{0\} \to \mathbb{R}^2_{\geq 0}$ is a map from the number w of whistles blown to the reward (or the payout) $\mathbf{P}^{\mathbb{W}}(w)$ disbursed per whistle and the slash $\mathbf{S}^{\mathbb{W}}(w)$ per player.

We model $\mathcal{M}^{\mathbb{W}}$ to have anonymous and simultaneous whistleblowing. Anonymity implies that the whistleblowing rewards and slashes cannot depend on the identities of the whistleblowers.[3] Simultaneity ensures that players decide on their actions at the same time—e.g., they decide the number of whistles to submit without knowing the decisions of the rest (see Remark 5). Note that anonymous whistleblowing implies that multiple whistles per player must be supported since the mechanism cannot distinguish between e.g., a single player submitting two whistles, and two players each submitting one whistle.

The reward $\mathbf{P}^{\mathbb{W}}(w)$ and slash $\mathbf{S}^{\mathbb{W}}(w)$ must satisfy the following natural conditions. To begin, whistleblowers cannot be penalized for submitting whistles — since whistles can be submitted from pseudonymous identities (e.g., fresh blockchain addresses), whistleblowers cannot be forced to pay any penalties enforced by $\mathcal{M}^{\mathbb{W}}$.

We also require $\mathcal{M}^{\mathbb{W}}$ to satisfy *budget balance* — the total slash collected should cover the rewards disbursed to whistleblowers. In other words, $w\mathbf{P}^{\mathbb{W}}(w) \leq n\mathbf{S}^{\mathbb{W}}(w)$ for all $w \in \mathbb{N} \cup \{0\}$.

[3] Anonymity does not rule out the rewards and slashes depending on the *order* in which the whistleblowing mechanism receives the whistles. However, for simplicity, we assume that all n players have similar computational power and network connectivity, and thus, all possible orders of the whistles reaching $\mathcal{M}^{\mathbb{W}}$ are equally likely. Then, $\mathbf{P}^{\mathbb{W}}(w)$ can be thought of the expected reward per whistle (expectation taken over the randomness in network conditions) upon w whistles getting blown.

Note that this implies $\mathbf{S}^{\mathbb{W}}(w) \geq 0$. Indeed, $\mathcal{M}^{\mathbb{W}}$ should neither reward nor slash anyone when no whistles are blown, and thus $\mathbf{P}^{\mathbb{W}}(0) = \mathbf{S}^{\mathbb{W}}(0) = 0$.

To ensure that the players pay the slash charged to them, the whistleblowing mechanism collects a deposit $D^{\mathbb{W}}$ upfront from all the players. The deposit must be large enough to cover the slash irrespective of the number of whistles blown, and thus, $D^{\mathbb{W}} = \sup_w \mathbf{S}^{\mathbb{W}}(w)$. Our goal is to design "effective" whistleblowing mechanisms that collect as little deposit as possible.

Definition 11 (Whistleblowing Mechanism $\mathcal{M}^{\mathbb{W}}$)**.** A whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ is given by the pair $(\mathbf{P}^{\mathbb{W}}, \mathbf{S}^{\mathbb{W}}) : \mathbb{N} \cup \{0\} \rightarrow \mathbb{R}^2_{\geq 0}$ such that upon receiving w whistles, the perpetuator of each whistle is awarded $\mathbf{P}^{\mathbb{W}}(w)$ and all players are slashed $\mathbf{S}^{\mathbb{W}}(w)$, satisfying $w\mathbf{P}^{\mathbb{W}}(w) \leq n\mathbf{S}^{\mathbb{W}}(w)$ and $\mathbf{P}^{\mathbb{W}}(0) = \mathbf{S}^{\mathbb{W}}(0) = 0$. $D^{\mathbb{W}} = \sup_w \mathbf{S}^{\mathbb{W}}(w)$ is said to be the deposit of the mechanism $\mathcal{M}^{\mathbb{W}}$.

Remark 4 ($\mathcal{M}^{\mathbb{W}}$ is public). We model $\mathcal{M}^{\mathbb{W}}$ to be *public*, i.e., it reveals the *total number* of whistles w that were blown (although whistleblower identities can still be hidden). This is reflective of $\mathcal{M}^{\mathbb{W}}$ being instantiated as a (public) smart contract, and will only make it easier for colluders, later on, in the retaliation context.

Remark 5 (Simultaneity). While we model players decisions on whistleblowing as simultaneous, if $\mathcal{M}^{\mathbb{W}}$ is deployed as a smart contract, there may be concern of strategy leakage through the public mempool. Observe first simultaneity comes for free with encrypted mempools where transaction contents are hidden. A similar guarantee can be achieved by using an off-chain TEE that accepts whistles and posts them to the on-chain whistleblowing contract after the conclusion of the whistleblowing period.

Even without this, time-lock encryption or commit-reveal schemes can be used instead. A similar modeling choice is made in [53]. There are several ways to achieve this but we provide one approach that also permits anonymous whistleblowing. As an illustration, $\mathcal{M}^{\mathbb{W}}$ will only accept time-lock encrypted messages, with decryption possible only after the whistleblowing period ends. The encryption can either be of a valid whistleblowing proof π, or of a garbage message (e.g., 0). Further, if no encrypted messages are submitted, $\mathcal{M}^{\mathbb{W}}$ will slash all players. Now, notice that no player can distinguish if a valid π has been submitted when it chooses its action. In other words, it is as if players need to move simultaneously.

Number of whistles. For a given number k of players that learn the whistleblowing proof, there is some natural upper bound on the number of whistles that they can blow within the time frame of the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$.[4] Reflecting this, we consider r-whistle mechanisms, that rewards nothing and slashes the entire deposit $D^{\mathbb{W}}$ upon receiving $r+1$ whistles or more. Essentially, the players would receive a net utility $-D^{\mathbb{W}}$ from blowing $r+1$ or more whistles, which is at most the utility that they would receive from blowing r whistles or less. Thus, the players would choose to blow at most r whistles even if they could submit more.

Definition 12 (r-whistle mechanism)**.** An r-whistle mechanism $\mathcal{M}^{\mathbb{W}} = (\mathbf{P}^{\mathbb{W}}, \mathbf{S}^{\mathbb{W}})$ is a whistleblowing mechanism where $\mathbf{P}^{\mathbb{W}}(w) = 0$ and $\mathbf{S}^{\mathbb{W}}(w) = D^{\mathbb{W}}$ for all $w > r$.

[4]Even if the players create many pseudonymous identities to blow whistles, this is not truly costless and would consume non-zero time (and in practice, non-zero transaction fee). This upper bounds the number of whistles that each player can submit within the time frame in which $\mathcal{M}^{\mathbb{W}}$ is eliciting whistles.

The extended form game. We now describe the extended-form two-round whistleblowing game $\mathbb{W}$ that models a whistleblowing protocol as seen by the n players. Prior to the start of the two rounds, players learn the utility V from colluding. The protocol designer then announces the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ and collects the whistleblowing deposit $D^{\mathbb{W}}$ from all players.

1. The first round of $\mathbb{W}$ is identical to the base game $\mathbb{B}$. Player i plays an action $b_i \in \{\mathsf{collude}, \mathsf{honest}\}$, to receive a utility

$$U_i^{(\mathbb{W},1)}(b_1, \ldots, b_n) = U_i^{\mathbb{B}}(b_1, \ldots, b_n).$$

The game proceeds to the second round if all n players $\mathsf{collude}$ and terminates immediately after the first round otherwise.

2. In the second stage of $\mathbb{W}$, the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ elicits whistles from the k players that know the whistleblowing proof (for convenience, re-index players so that these k players are labelled $1, \ldots, k$). For $1 \leq i \leq k$, player i's action space in the second round consists of blowing $w_i \in \mathbb{N} \cup \{0\}$ whistles. Players $k+1, \ldots, n$ each blow $w_i = 0$ whistles. Then, when $w = \sum_{i=1}^{n} w_i$ whistles are blown, player i receives a whistleblowing reward equal to $w_i \mathbf{P}^{\mathbb{W}}(w)$ and is slashed $\mathbf{S}^{\mathbb{W}}(w)$ for a total second stage utility

$$U_i^{(\mathbb{W},2)}(w_1, \ldots, w_n) = w_i \mathbf{P}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w).$$

Across the two stages, player i earns a net utility $U^{\mathbb{W}}(b_1, \ldots, b_n, w_1, \ldots, w_n)$ given by the sum of the utilities in the two stages.

For notational convenience, we denote the SPPNE $(\mathsf{collude}, \ldots, \mathsf{collude}), \vec{w}$ by $\vec{w}^2$.

Subgame perfect pure Nash equilibrium in $\mathbb{W}$. The protocol designer would want to ensure that all SPPNEs of the whistleblowing game $\mathbb{W}$ consist of some player i playing $b_i = \mathsf{honest}$ in the first stage, while the coalition would want to play an equilibrium where all players $\mathsf{collude}$ (if one exists). We will begin our analysis of the players' incentives in $\mathbb{W}$ by characterizing all SPPNEs.

A typical approach to computing all SPPNEs is through a bottom-up dynamic program. The players can be expected to play a pure Nash equilibrium $\vec{w} = (w_1, \ldots, w_n)$ in the second round of $\mathbb{W}$, since $\mathbb{W}$ terminates at the end of the round irrespective of their actions. Then, for a given second-stage equilibrium $\vec{w}$, the utilities from playing $\vec{w}^2 = (\mathsf{collude}, \ldots, \mathsf{collude}), \vec{w}$ and from playing honest can be compared. $\vec{w}^2$ is a SPPNE if and only if all players i receive a utility $U_i^{\mathbb{W}}(\vec{w}^2)$ at least zero, the utility from deviating and playing honest in the first stage (which causes the game to terminate immediately in the first round).

For $\vec{w}$ to be a second-stage equilibrium, no player i should be able to achieve a larger utility $U_i^{(\mathbb{W},2)}$ from deviating and blowing w_i' whistles instead of w_i for all $1 \leq i \leq k$. Formally, $\vec{w}$ is a pure Nash equilibrium in stage two if and only if

$$\begin{aligned} & w_i \mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) \\ & \qquad \geq w_i' \mathbf{P}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) \end{aligned} \tag{1}$$

for all players $1 \leq i \leq k$ and $w_i' \in \mathbb{N} \cup \{0\}$.

For some second-round equilibrium $\vec{w}$, to verify whether $(\mathsf{collude}, \ldots, \mathsf{collude}), \vec{w}$ is a SPPNE, we will first calculate the cumulative utility $U_i^{\mathbb{W}}(\mathsf{collude}, \ldots, \mathsf{collude}, \vec{w})$ received by each player i.

$$\begin{aligned} U_i^{\mathbb{W}}(\vec{w}^2) &= U_i^{(\mathbb{W},1)}(\mathsf{collude}, \ldots, \mathsf{collude}) + U_i^{(\mathbb{W},2)}(\vec{w}) \\ &= V + w_i \mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j). \end{aligned}$$

For $\vec{w}^2$ to be a SPPNE, each player i should receive a non-negative utility. Otherwise, player i will guarantee itself a zero utility by being honest in the first stage. In other words, for all $1 \leq i \leq n$,

$$V + w_i \mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) \geq 0. \tag{2}$$

In summary, to ensure that all SPPNE of $\mathbb{W}$ involve some player being honest, the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ must ensure that either Equation (1) is violated for some $1 \leq i \leq k$ or Equation (2) is violated for some $1 \leq i \leq n$ for all possible strategy profiles $\vec{w}$ of the players in the second round.

Example. We will construct the "1-whistle skeleton" mechanism $\mathcal{M}^{\mathbb{W}}$ and argue that all SPPNE involve some player playing honest.

At a high level, for a collusion reward of V per player, the 1-whistle skeleton collects a deposit $V + \frac{\varepsilon}{n-1}$ from each player and slashes all of it even if a single whistle is blown.[5] The whistleblower is rewarded $V + \varepsilon > V + \frac{\varepsilon}{n-1}$ if exactly a single whistle is blown, but disburses no reward if two or more whistles are submitted to $\mathcal{M}^{\mathbb{W}}$.

- $\mathbf{P}^{\mathbb{W}}(w) = \begin{cases} V + \varepsilon & \text{if } w = 1, \\ 0 & \text{otherwise.} \end{cases}$
- $\mathbf{S}^{\mathbb{W}}(w) = \begin{cases} 0 & \text{if } w = 0, \\ V + \frac{\varepsilon}{n-1} & \text{otherwise.} \end{cases}$

Theorem 6.1. *For any $1 \leq k \leq n$, the 1-whistle skeleton mechanism $\mathcal{M}^{\mathbb{W}}$ described above deters collusion, i.e, all subgame perfect pure Nash equilibria in the whistleblowing game $\mathbb{W}$ involves some player playing honest in round one.*

Proof. We will calculate the bottom-up dynamic program by first spotting all second-stage pure Nash equilibria $\vec{w}$ (so that Equation (1) holds for all $1 \leq i \leq k$) and arguing that some player ends up with a negative utility for all such $\vec{w}$ (i.e, Equation (2) is violated for some player i).

We will argue that $\vec{w} = (0, \ldots, 0)$ does not satisfy Equation (1) for any of the k players that can blow a whistle. Indeed, by blowing no whistles, $U_i^{(\mathbb{W},2)}(0, \ldots, 0) = 0$, but by deviating and blowing a whistle, player i increases its utility to

$$1 \times \mathbf{P}^{\mathbb{W}}(1) - \mathbf{S}^{\mathbb{W}}(1) = (V + \varepsilon) - (V + \frac{\varepsilon}{n-1}) > 0.$$

Next, we will argue that for any $\vec{w} \neq (0, \ldots, 0)$, Equation (2) cannot be simultaneously satisfied for all n players, and thus, someone would want to deviate and play honest in the first stage.

[5]Think of ε as any small positive constant—orders of magnitude smaller than other discussed quantities. We only use it to clean up our analysis by breaking ties between the utilities of different actions taken by the players.

Consider some player i that receives no whistleblowing reward in the second stage. Such a player always exists — if exactly one whistle is blown, consider some player other than the whistleblower and if more than one whistle is blown, no player receives a whistleblowing reward. $U_i^{(\mathbb{W},2)}(\vec{w}) = -(V + \frac{\varepsilon}{n-1})$ and thus, $V - (V + \frac{\varepsilon}{n-1}) < 0$, violating Equation (2).

Thus, all SPPNE in the whistleblowing game $\mathbb{W}$ induced by the 1-skeleton mechanism involve some player playing honest in the first round. □

Note that if the deposit $D^{\mathbb{W}}$ collected by $\mathcal{M}^{\mathbb{W}}$ is less than V, the n players may choose to collude without concern of the second-stage outcome of $\mathbb{W}$, since they are slashed at most V in $(\mathbb{W}, 2)$. Hence, collecting a whistleblowing deposit V from each player can be a natural baseline to have in mind for all the discussions to follow.[6]

6.3 Retaliation Game $\mathbb{R}$

In the two-round whistleblowing game $\mathbb{W}$, the protocol designer had to announce the whistleblowing mechanism before the start of the first round. This gives the players an opportunity to smartly collude by launching their own *retaliation mechanism* $\mathcal{M}^{\mathbb{R}}$ to *break* the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ — the retaliation mechanism looks to modify the payoffs of the second stage so that blowing $(w_1, \ldots, w_n)$ whistles is both, a second stage equilibrium and leaves every player with a positive net utility across the two rounds.

In particular, note that the second-stage equilibrium need not prevent every playing from blowing any whistles. The retaliation mechanism can choose to induce a different equilibrium as long as each player profits from colluding. For example, $\mathcal{M}^{\mathbb{R}}$ could actually ask colluders to whistleblow and then redistribute rewards so that the utility from colluding is larger than not-colluding for each player $1 \leq i \leq n$.

The retaliation mechanism. We describe the retaliation mechanism before formally setting up the retaliation game. The retaliation mechanism $\mathcal{M}^{\mathbb{R}} = (\mathcal{M}_1^{\mathbb{R}}, \ldots, \mathcal{M}_n^{\mathbb{R}})$, where $\mathcal{M}_i^{\mathbb{R}} : \mathbb{N} \cup \{0\} \to \mathbb{R}$ maps the number w of whistles submitted to the whistleblowing mechanism to the penalty charged to player i by the retaliation mechanism. We use the term "penalty" quite loosely here — apart from penalizing players for not playing according to the coalition's strategy, the retaliation mechanism also serves the dual purpose of redistributing the whistleblowing rewards received by players $1, \ldots, k$ to $k+1, \ldots, n$, who cannot blow a whistle. Thus, without loss of generality, we can assume that the players $k+1, \ldots, n$ are charged a negative penalty $\mathcal{M}_i^{\mathbb{R}}(w) \leq 0$ (i.e, rewarded by $\mathcal{M}^{\mathbb{R}}$), irrespective of the number w of whistles blown. Otherwise, such a player is strictly better off by being honest than colluding.

Similar to the whistleblowing mechanism, we require $\mathcal{M}^{\mathbb{R}}$ to satisfy budget balance. The retaliation mechanism is only a means to penalize players and to redistribute rewards amongst themselves. Thus, we assert $\sum_i \mathcal{M}^{\mathbb{R}}(w) \geq 0$ for all $w \in \mathbb{N} \cup \{0\}$. Additionally, to ensure player i pays the penalty charged to it, it will have to submit a *retaliation deposit* equal to the largest penalty charged to i by $\mathcal{M}^{\mathbb{R}}$. Indeed, if a player is never penalized but is only rewarded by the retaliation mechanism, they do not have to submit a deposit. Formally, the retaliation deposit $D_i^{\mathbb{R}} = \max\{\sup_w \mathcal{M}_i^{\mathbb{R}}(w), 0\}$.

[6]Our lower bounds do not rule out a whistleblowing mechanism with a deposit smaller than V that deters collusion. Even though all players might be better off colluding in the first stage irrespective of the outcome in the second, there might not exist a pure Nash equilibrium in $(\mathbb{W}, 2)$. We defer a more detailed discussion to Appendix C.

Definition 13 (Retaliation mechanism)**.** The retaliation mechanism $\mathcal{M}^{\mathbb{R}} : \mathbb{N} \cup \{0\} \to \mathbb{R}^n$ takes as input the total number of whistles w submitted by the players $1, \ldots, k$ to the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ and outputs the penalty imposed on each of the n players satisfying $\sum_{i=1}^{n} \mathcal{M}_i^{\mathbb{R}}(w) \geq 0$ and $\mathcal{M}_i^{\mathbb{R}}(w) \leq 0$ for all $i > k$. The deposit of the retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ is given by $D_i^{\mathbb{R}} = \max\{\sup_w \mathcal{M}_i^{\mathbb{R}}(w), 0\}$.

Unlike the whistleblowing mechanism, the retaliation mechanism need not be anonymous. $\mathcal{M}^{\mathbb{R}}$ can suggest different players to submit a different number of whistles to $\mathcal{M}^{\mathbb{W}}$, and can penalize them differently if the number of whistles that are blown are not as recommended by $\mathcal{M}^{\mathbb{R}}$.

Remark 6 (Executing $\mathcal{M}^{\mathbb{R}}$). We model the retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ to be "self-executing" in the sense that it can observe the public output of $\mathcal{M}^{\mathbb{W}}$ and execute appropriate penalties without requiring external intervention. The reader might observe here that a smart-contract instantiation of $\mathcal{M}^{\mathbb{R}}$ would require a transaction to trigger it—this "relays" the public information about $\mathcal{M}^{\mathbb{W}}$ to $\mathcal{M}^{\mathbb{R}}$.

Even in such a scenario, the retaliation mechanism can be made to be effectively self-executing. In particular, by offering a small reward to anyone in the world (not restricted to the n players) who relays this public information from $\mathcal{M}^{\mathbb{W}}$ to $\mathcal{M}^{\mathbb{R}}$, it can be guaranteed that the penalties in $\mathcal{M}^{\mathbb{R}}$ will execute.

We further show that even without self-executing retaliation, our results stay the same. As one example, a different $\mathcal{M}^{\mathbb{R}}$ can be used to break any whistleblowing mechanism. We provide details in Section 7.3.

The extended-form game. The retaliation game $\mathbb{R}$ proceeds similar to $\mathbb{W}$, except that players receive an additional penalty $\mathcal{M}_i^{\mathbb{R}}(w)$ in the second stage from blowing w whistles.

Apart from learning the collusion utility V, hearing about the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ and locking in the whistleblowing deposit $D^{\mathbb{W}}$ before the first round commences, the players also agree upon a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ and put down the necessary retaliation deposits $D_1^{\mathbb{R}}, \ldots, D_n^{\mathbb{R}}$.

Similar to $\mathbb{W}$, the retaliation game $\mathbb{R}$ also occurs over two stages.

1. The first round consists of the base game $\mathbb{B}$, where players can play actions $b_1, \ldots, b_n \in \{\mathsf{collude}, \mathsf{honest}\}$ to receive a utility

$$U_i^{(\mathbb{R},1)}(b_1, \ldots, b_n) = U_i^{\mathbb{B}}(b_1, \ldots, b_n).$$

The retaliation game $\mathbb{R}$ proceeds to the second stage if and only if all n players $\mathsf{collude}$ in the first stage.

2. As with the whistleblowing game, the second-round action space for players $1 \leq i \leq k$ consists of submitting w_i whistles to $\mathcal{M}^{\mathbb{W}}$, while the remaining players blow $w_i = 0$ whistles. The whistleblowing mechanism then awards each player i with a utility $w_i \mathbf{P}^{\mathbb{W}}(\sum_i w_i) - \mathbf{S}^{\mathbb{W}}(\sum_i w_i)$. Additionally, they are also charged a penalty $-\mathcal{M}_i^{\mathbb{R}}(\sum_i w_i)$ by the retaliation mechanism, to receive a total utility equal to

$$U_i^{(\mathbb{R},2)}(w_1, \ldots, w_n) = w_i \mathbf{P}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w) - \mathcal{M}_i^{\mathbb{R}}(w)$$

in the second stage, where $w = \sum_{i=1}^{n} w_i$.

The total utility $U_i^{\mathbb{R}}(b_1, \ldots, b_n, w_1, \ldots, w_n)$ received by player i equals the sum of utilities from the two stages.

Similar to the whistleblowing game $\mathbb{W}$, we will denote the SPPNE $(\mathsf{collude}, \ldots, \mathsf{collude}), (w_1, \ldots, w_n)$ by $(w_1, \ldots, w_n)^2$.

Subgame perfect pure Nash equilibria in $\mathbb{R}$. As with the whistleblowing game $\mathbb{W}$, we locate all SPPNEs of $\mathbb{R}$ by a bottom-up dynamic program. When $\vec{w}^2$ is a SPPNE, $\vec{w}$ played by $1, \ldots, k$ must be a second-stage equilibrium, which implies

$$\begin{aligned} &w_i \mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) - \mathcal{M}_i^{\mathbb{R}}(\sum_j w_j) \\ &\quad \geq w_i' \mathbf{P}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \mathcal{M}_i^{\mathbb{R}}(w_i' + \sum_{j \neq i} w_j) \end{aligned} \tag{3}$$

for all $w_i' \in \mathbb{N} \cup \{0\}$ and $1 \leq i \leq k$. Further, $\vec{w}^2$ is a subgame perfect pure Nash equilibrium if and only if each player i is better off than deviating and playing honest in the first stage.

$$\begin{aligned} U_i^{\mathbb{R}}(\vec{w}^2) &= U_i^{(\mathbb{R},1)}(\mathsf{collude}, \ldots, \mathsf{collude}) + U_i^{(\mathbb{R},2)}(\vec{w}) \\ &= V + w_i \mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) - \mathcal{M}_i^{\mathbb{R}}(\sum_j w_j) \\ &\geq 0 \end{aligned} \tag{4}$$

for all $1 \leq i \leq n$.

The coalition aims to induce some SPPNE $\vec{w}^2$ in $\mathbb{R}$ where all players $\mathsf{collude}$ in the first stage. In such a case, we say that the retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ and the SPPNE $\vec{w}^2$ *breaks* the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$.

Breaking *any* whistleblowing mechanism through retaliation. For a given whistleblowing deposit $D^{\mathbb{W}}$, we will argue that any whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ can be broken by a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ with a sufficiently large deposit $D^{\mathbb{R}}$. Intuitively, if the retaliation mechanism collects a deposit much larger than the utility from collusion and the whistleblowing rewards, $\mathcal{M}^{\mathbb{R}}$ can threaten to burn the entire $D_i^{\mathbb{R}}$ for every player i even if a single whistle is blown. Players are better off receiving zero utility in the second stage from blowing no whistles than incurring the massive retaliation penalty from blowing whistles.

Theorem 6.2. *For any whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$, there exists a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ with a sufficiently high deposit $D^{\mathbb{R}} \geq (n-1) \cdot D^{\mathbb{W}}$ such that $\mathcal{M}^{\mathbb{R}}$ and $(0, \ldots, 0)^2$ breaks $\mathcal{M}^{\mathbb{W}}$.*

Proof. Consider the retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ that collects a deposit $D_i^{\mathbb{R}} = (n-1) \times D^{\mathbb{W}}$ from each player i given by

$$\mathcal{M}_i^{\mathbb{R}}(w) = \begin{cases} 0 & \text{if } w = 0 \\ D^{\mathbb{R}} & \text{otherwise.} \end{cases}$$

We will argue that $(0,\dots,0)^2$ is a SPPNE of $\mathbb{R}$. To see Equation (3) is satisfied for all $1 \le i \le k$, observe that for $w \ge 1$

$$\begin{aligned}
& w_i \mathbf{P}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w) - \mathcal{M}_i^{\mathbb{R}}(w) \\
& \le w\mathbf{P}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w) - \mathcal{M}_i^{\mathbb{R}}(w) \\
& \le n\mathbf{S}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w) - \mathcal{M}_i^{\mathbb{R}}(w) \\
& \le (n-1) \times D^{\mathbb{W}} - (n-1) \times D^{\mathbb{W}} \le 0.
\end{aligned}$$

The third line follows from budget balance of $\mathcal{M}^{\mathbb{W}}$. Thus, the players are all better off not blowing any whistles in the second stage. Indeed, Equation (4) is trivially satisfied. All players receive a net utility equal to V across the two stages playing $(0,\dots,0)^2$, which is strictly better than the utility from being honest. Thus, $(0,\dots,0)^2$ is a SPPNE of $\mathbb{R}$ and $\mathcal{M}^{\mathbb{R}}$ and $(0,\dots,0)^2$ breaks $\mathcal{M}^{\mathbb{W}}$. $\square$

This suggests that whistleblowing mechanisms cannot provide *unconditional economic security* against retaliation-enabled smart-collusion. For a fixed whistleblowing deposit $D^{\mathbb{W}}$, a natural objective for the protocol designer would then be to maximize the retaliation deposit required to break $\mathcal{M}^{\mathbb{W}}$.

We propose that the economic security of a whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ is characterized by the maximum retaliation deposit submitted by any player in the optimal retaliation mechanism that breaks $\mathcal{M}^{\mathbb{W}}$. This best captures the largest "financial stress" placed by $\mathcal{M}^{\mathbb{R}}$ on the n players in their attempt to break $\mathcal{M}^{\mathbb{W}}$. In a sense, the collusion is only as strong as its weakest member, who if this financial stress is too large would not take part, resulting in the collusion itself not forming. Accordingly, we define the economic security of $\mathcal{M}^{\mathbb{W}}$ as $\min_{\mathcal{M}^{\mathbb{R}}:\mathcal{M}^{\mathbb{R}} \text{ breaks } \mathcal{M}^{\mathbb{W}}} \left(\max_i D_i^{\mathbb{R}}\right)$.

Remark 7 (Borrowing money). One might ask if players could *borrow* money (with an under-collateralized loan) from an external source to put as deposit in $\mathcal{M}^{\mathbb{W}}$ (or $\mathcal{M}^{\mathbb{R}}$). Note that in practice, this doesn't change our results. While it may seem like such a player would care less about losing their deposit, note that an under-collateralized loan puts the risk squarely on the lender, who in an efficient market would be unwilling to lend in the first place. In general, any lender-player coalition would not profit from such a strategy.

7 Effective Whistleblowing

7.1 The 1-Whistle Mechanism $\mathbb{M}_{1*}^{\mathbb{W}}$

We extend our 1-whistle skeleton to be resilient against retaliation mechanisms with deposit up to $D^{\mathbb{R}} = (n-1)\,\gamma$ for $\gamma \ge 0$.

We propose the 1-whistle $(n-1)\,\gamma$-secure mechanism $\mathbb{M}_{1*}^{\mathbb{W}}$, which collects a whistleblowing deposit $D^{\mathbb{W}} = V + \gamma + \frac{\varepsilon}{n-1}$ from each player and transfers the entire collected deposits to the whistleblower if a single whistle is blown. If two or more whistles are blown, then the entire deposit is slashed and none of the whistleblowers receive any reward. Formally, our mechanism is given by

1. $\mathbf{P}^{\mathbb{W}}(w) = \begin{cases} nV + n\,\gamma + \varepsilon & \text{if } w = 1, \\ 0 & \text{otherwise,} \end{cases}$

2. $\mathbf{S}^{\mathbb{W}}(w) = \begin{cases} 0 & \text{if } w = 0, \\ V + \gamma + \frac{\varepsilon}{n-1} & \text{otherwise.} \end{cases}$

We argue that $\mathbb{M}_{1*}^{\mathbb{W}}$ cannot be broken by any retaliation mechanism that collects a retaliation deposit of up to $(n-1)\,\gamma$. Intuitively, when a single whistle is blown, $\mathbb{M}_{1*}^{\mathbb{W}}$ transfers greater than $V+\gamma$ units of money from all the players to the whistleblower. The $(n-1)$ players other than the whistleblower need to be paid strictly greater than γ per player for their net utility across the two rounds to be non-negative (they receive V from collusion in the first round and are slashed greater than $V+\gamma$ in the second). However, the retaliation deposit is at most $(n-1)\,\gamma$, and thus, the whistleblower cannot be forced to redistribute more than $(n-1)\,\gamma$ units of cash. As a consequence, there exists some player that receives a negative utility from colluding, and therefore, $\mathbb{M}_{1*}^{\mathbb{W}}$ cannot be broken with a retaliation deposit at most $(n-1)\,\gamma$.

Theorem 7.1. *$\mathbb{M}_{1*}^{\mathbb{W}}$ cannot be broken by any retaliation mechanism that collects a deposit of at most $(n-1)\,\gamma$ from each of its players, irrespective of the number of players k that learn the whistleblowing proof.*

We defer a formal proof to Appendix E.1.

Observe that the retaliation deposit needed to break $\mathbb{M}_{1*}^{\mathbb{W}}$ scales linearly with the number of players. To paint a picture, for $n = 1000$, an additional \$1000 deposit from each player on top of the baseline deposit V will result in $\mathbb{M}_{1*}^{\mathbb{W}}$ being resilient to retaliation mechanisms with deposits of up to roughly a million dollars!

In Theorem 6.2, we argued that any whistleblowing mechanism that collects a deposit $V+\gamma$ can be broken by a retaliation mechanism with a deposit $(n-1)(V+\gamma)$. We explore tighter lower bounds in the next section.

7.2 Lower Bounds

We first argue that the $\mathbb{M}_{1*}^{\mathbb{W}}$ is optimal when $k = 1$, i.e, only a single player knows the whistleblowing secret (proof in Appendix E.2).

Theorem 7.2. *Any whistleblowing mechanism that collects a whistleblowing deposit of at most $V+\gamma$ can be broken by a retaliation mechanism with a retaliation deposit of at most $(n-1)\,\gamma$ and some strategy profile $\vec{w}^2$ when $k = 1$.*

We also argue that $\mathbb{M}_{1*}^{\mathbb{W}}$ is optimal for $k > 1$ amongst all mechanisms that (a) have a pure Nash equilibrium in the second stage and (b) the pure Nash equilibrium is such that the joint utility of the coalition across the two rounds is non-negative. A formal analysis is given in Appendix E.3.

We motivate assuming the existence of a second-stage pure Nash equilibrium satisfying the condition on net positive utility for the coalition in Appendix C.

7.3 Alternative Collusion Models

In Appendix D, we consider other interesting models of collusion. We summarize our exploration below.

Jointly controlled accounts (Appendix D.1). We consider models where colluders can set up and blow whistles from *jointly controlled* accounts. Jointly controlled accounts can simplify collusion by making sure no player can collect whistleblowing rewards in their private accounts and "abscond" without sharing with the other players. We find that our results do not change even when colluders can instantiate joint accounts.

As an extension, we also allow we joint accounts that can automatically blow whistles—an extremely strong power given to colluders. Here, the players cannot deviate to stop the automated whistles that are blown, but can only blow additional whistles from their private accounts. We find that our results continue to hold in the automated whistleblowing model with little modifications.

Interestingly, if desired, the whistleblowing mechanism could prevent joint accounts altogether using recent cryptographic techniques [39, 60] that allow proving *individual* ownership of keys.

Non-anonymous whistleblowing (Appendix D.2). Here, we restrict the mechanism designer such that $\mathcal{M}^{\mathbb{W}}$ no longer supports anonymous whistleblowing. Here, we show that our results continue to hold except in the case that the whistleblowing deposit can be held indefinitely. One example of this is the client-setting with transient clients (see Section 8.2) which requires anonymity for our positive result.

Non-automatic retaliation (Appendix D.3) Here, we restrict the power of colluders by not allowing them to deploy retaliation contracts that automatically trigger penalties after observing the output of $\mathcal{M}^{\mathbb{W}}$. We find that this does not change our results.

8 Beyond (n, n)-Collusion

In this section, we discuss designing whistleblowing mechanisms beyond (n, n)-collusion. Specifically, we discuss details for (t, n)-collusion and the transparent service setting.

8.1 (t, n)-Collusion

We find that from the game-theoretic perspective, (t, n)-collusion is surprisingly similar to (n, n)-collusion.

(t, n)-collusion generalizes the (n, n)-collusion environment as follows — it suffices for a coalition to contain any t out of the n players to successfully collude and derive a utility of V per player in the coalition. For a given coalition of players $1, \ldots, t$, we discuss designing whistleblowing mechanisms to deter smart-collusion.

Observe that the whistleblowing and retaliation games of (t, n)-collusion are identical to that of (t, t)-collusion, except for the budget balance constraint. Rather than using the whistleblowing slashes from just the t colluders to disburse whistleblowing rewards, the whistleblowing mechanism can also rely on the slashes collected from the $n - t$ players that are not part of the coalition. Thus, budget balance in the (t, n)-collusion environment requires that $w\mathbf{P}^{\mathbb{W}}(w) \leq n\mathbf{S}^{\mathbb{W}}(w)$ for all $w \in \mathbb{N} \cup \{0\}$.

It is fairly straightforward to see that $\mathbb{M}_{1*}^{\mathbb{W}}$ (from Section 7.1) continues to remain resilient against retaliation mechanisms that collect a deposit of up to $(t-1)\gamma$.

Proposition 8.1. *The $\mathbb{M}_{1*}^{\mathbb{W}}$ mechanism cannot be broken by any retaliation mechanism with a retaliation deposit of at most $(t-1)\gamma$ in the (t,n)-environment, irrespective of the number of players $1 \leq k \leq t$ that learn the whistleblowing proof.*

Lower bounds in the (t,n)-collusion setting can also be derived analogously. We defer the details to Appendix E.4.

Identifying colluders. While some cryptographic whistleblowing proofs can identify a subset of the t players that choose to collude, we note that our impossibility result on unconditional economic security continues to hold. In fact, it holds even if the whistleblowing mechanism exactly knows which t out of the n players collude.

To see why, observe that even with (n,n)-collusion where the protocol designer knows that all n players have to collude in order to receive the collusion reward V (i.e., the whistleblowing proof "identifies" all colluders), no mechanism could be resilient against retaliation mechanisms with a sufficiently large retaliation deposit.

Remark 8 (Adaptive (t,n)-collusion). An interesting extension is to model *adaptive corruption*, where parties may be corrupted at any time during execution instead of just on initialization.

First, note that the underlying cryptography primitive for generating whistleblowing proof must be adaptively secure. Without this, whistleblowing would not work as intended even without considering incentives. Building adaptively secure cryptographic primitives is an orthogonal question.

From a definitional standpoint, observe that our definitions of unframability can be readily extended to support adaptive corruptions. Moreover, our whistleblowing constructions should directly inherit adaptive security from the underlying threshold primitive. For example, if the threshold VRF underlying a distributed randomness beacon service is secure against adaptive corruptions, then our whistleblowing protocol from Section 4.3.2 also achieves unframability against adaptive corruptions. We do note that achieving adaptive security for threshold cryptographic primitives can be challenging for some primitives (e.g., see [31] for adaptive threshold signatures).

Now, given an adaptively secure underlying cryptographic primitive, note that from the game-theoretic side, our analysis begins given a t-sized collusion and does not care about the details on how it formed. In turn, our game-theoretic results will not change even if the collusion resulted from adaptive corruption. Still, it is interesting to study this "collusion formation phase" that precedes the game we model. Intuitively, the goal here will be to understand how players propose coalitions, join or leave them, or even form sub-coalitions. We leave the details to future work.

8.2 Transparent Service Setting

In this section, we design whistleblowing mechanisms for the transparent service provider model (with n servers and a client) from Section 4. The induced whistleblowing and retaliation games will mirror our analysis from earlier. We discuss some differences below.

We focus on collusion forming between t servers and the client. The client receives a whistleblowing proof upon receiving off-band service. A subset of the servers might or might not receive the whistleblowing proof too. The transient service model differs from (t,n)-collusion in that the whistleblowing mechanism can only collect deposits from servers and not from client.

A very similar proof to Theorem 7.1 will show that an appropriate adaptation of $\mathbb{M}_{1*}^{\mathbb{W}}$ continues to remain resilient against any retaliation mechanism that collects a retaliation deposit of up to $n\gamma$

from the client and the servers (for $n = 1000$ and $\gamma = 1000$, imagine a commercial service asking its client to lock in roughly a million dollars to use its service). Similarly, it is also not hard to see that our result on the impossibility of unconditional economic security carries over.

Nevertheless, we find an interesting positive result for unconditional security by leveraging a characteristic specific to this setting.

Unconditional economic guarantees for long term services with *transient* clients. A natural application of the transparent service setting is one where servers provide their functionality as a *long term* service, while clients are transient and only appear when they wish to submit a query. This provides an interesting tool to the whistleblowing protocol designer—deposits from servers can be held *indefinitely* (essentially as a "cost of doing business"), or until there is a whistleblowing proof submitted. In turn, to break this mechanism and successfully provide off-band service to a client, the servers would also have to hold a retaliation deposit from the client indefinitely. Otherwise, the client could simply wait until its deposit is returned before whistleblowing.

However, it is likely that no client will want to have its retaliation deposit be held indefinitely in order to collude off-band. As a consequence, in our formalism, this translates precisely to the setting with $D^{\mathbb{R}} = 0$ (i.e., whistleblowing without a retaliation contract). This allows us to provide unconditional economic security.

Proposition 8.2. *There exist whistleblowing mechanisms that collect a deposit $D^{\mathbb{R}} = V + \frac{\varepsilon}{n-1}$ that cannot be broken by any retaliation mechanism in the transparent service model with long term service and transient clients.*

9 Estimating Collusion Utility

So far, we have assumed that the utility V that each player gains through collusion is known to the whistleblowing mechanism designer in advance. This allows setting the deposit D required from each player. In this section, we use illustrative examples to explore the practicality and limitations of estimating collusion utility.

Unconditional security result is unchanged. First, observe that our headline negative result—the impossibility of unconditional security in the presence of smart-collusion—continues to hold regardless of the application or whether V is known. In particular, the impossibility result holds *even when* the protocol can exactly learn the collusion utility, and application specific constraints will only make the impossibility stronger. Our discussion in this section therefore focuses on our positive results.

Section structure. First, in Section 9.1, we discuss several natural applications where V is easy to estimate (or upper bound). As one illustrative example, we detail an application where a custodial wallet service is used to store cryptocurrency funds.

Next, Section 9.2 considers settings where it is difficult for the mechanism designer to know V. Here, we offer a new perspective on V based on *insurance*. Abstractly, the client herself will be asked to provide the value W for which she would like to be protected. For any loss suffered due to the collusion, the whistleblowing mechanism will ensure that the client is made whole up to W.

We show how this perspective unlocks new settings for our framework by detailing an illustrative application with randomness beacons.

Finally, in Section 9.3, we discuss limitations for applying our framework in the context of estimating collusion utility, and discuss potential directions for further analysis.

9.1 Settings that Permit Estimation of Utility

We begin with a representative example on custodial wallets in Section 9.1.1 to show estimation of collusion utility. Section 9.1.2 describes additional applications that are similar in nature.

9.1.1 Custodial Wallets

Imagine a custodial wallet service for users to hold their long-term funds. Such a service is useful for users not wanting to worry about storing wallet keys themselves. Note that existing exchanges like Coinbase and Binance already provide such a service (of course, in addition to other functionalities). Further, it is increasingly common to expect that the key is stored in a distributed fashion using secret sharing (see e.g., [29]).

Now, in this setting, it is natural for a user Alice to still want protection against a malicious committee that could collude to steal her funds. Existing deployments do not account for this; we note that our whistleblowing framework could be applied here.

Concrete specification. We begin by describing the cryptographic components to implement custodial wallets with whistleblowing. In more detail, committee $\{P_1, \ldots, P_n\}$ will, on Alice's behalf, jointly store a key sk for a wallet with value W. The funds are fully custodied by the committee (i.e., Alice does not know the underlying sk). To enable whistleblowing, the key will be split amongst the committee using *secret-sharing with snitching* (SSS) [40]. SSS extends regular secret-sharing with the additional guarantee that if the committee colludes to learn sk, then some member will be able to individually learn sk. Importantly, with SSS, unlike regular secret sharing, the committee cannot use MPC to sign transactions with sk without first leaking sk to some member. Now, knowledge of the secret sk effectively serves as a whistleblowing proof of collusion.

Observe here that the utility from collusion is known apriori. In particular, for a wallet worth W and (t, n)-collusion, V can be set to be W/t to ensure that Alice's assets are protected. Now, if a whistle is blown, instead of burning all committee deposits, the mechanism will first compensate W to Alice. For this, intuitively, we will need to hold an additional deposit (roughly $\frac{2t-n}{n}V$) from each committee member to obtain the same quantitative security guaranty.

More specifically, we begin by recalling the $(t-1)\gamma$-secure mechanism discussed in Section 7.1 and 8.1. If exactly one whistle is received, the mechanism effectively fully slashes the deposits D of all committee members and rewards the whistleblower $tD - \epsilon'$ (for some $\epsilon' > 0$). If an additional tV is required to compensate Alice, then we would need the total payment $2tD - \epsilon$ to be smaller than the total deposit nD. Concretely, a deposit larger than $\frac{2t}{n}V + \gamma + \frac{\epsilon}{t-1}$ per player will be sufficient to protect against smart-collusion with deposit $(t-1)\gamma$. Note that when $t < n/2$, no additional deposit is necessary; additionally, a maximum of V extra deposit is sufficient which happens in the (n, n) collusion setting.

Time-varying utility. In case additional funds are added to the wallet, $\mathcal{M}^{\mathbb{W}}$ can ask the committee to add a higher deposit to protect the wallet's increased value. Alternatively, the same deposit can be maintained, but the user's guarantee will be degraded to protection only against the original value.

Wallet custody vs backup. One subtlety to note here is the difference between wallet *custody* and wallet *backup* and the challenge in the latter setting since Alice knows the key herself and could falsely whistleblow.

In particular, in the backup setting, the whistleblowing proof can no longer just be knowledge of sk—the protocol must prevent Alice from blowing the whistle herself and claiming committee collusion. One approach is to ensure that only committee members can whistleblow by using e.g., identities or credentials. However, we observe that this does not suffice either—Alice could collude with one member P, give P the secret key sk so that it can whistleblow, and split the profits.

We leave an exploration of whistleblowing for this key backup setting to future work.

9.1.2 Similar Settings

We briefly mention additional applications where collusion utility can similarly be estimated.

For BFT blockchain protocols that support whistleblowing through forensics, the maximum possible gain W from colluding can be thought of as the total value locked TVL on-chain. The mechanism can now use the bound $V < W/t$ on the per player utility.

As another example, consider accountable threshold decryption used specifically in the context of encrypted mempools. Here, the maximum gain W from colluding is essentially the maximum extractable value (MEV) [6, 32]. In turn, historic MEV data (e.g., payments made by MEV searchers and builders to validators) could be used to find a reasonable upper tail bound on V.

9.2 Settings with Difficult Ex-Ante Estimation

We now describe applications where it is difficult for the mechanism to know the collusion utility in advance. But, by asking the client herself to supply the maximum loss $W = tV$ that she can suffer, the mechanism can essentially insure her against that amount. We describe this new *insurance*-based perspective in Section 9.2.1 using our custodial wallet example. Then, in Section 9.2.2, we detail a new setting for a randomness beacons service that uses this perspective.

9.2.1 The Insurance Perspective

As alluded to earlier, the core idea is to view V in terms of how much *loss protection* the client desires.

Basic example. A natural starting point is to revisit the custodial wallet application, but in a setting where the cryptocurrency is privacy-preserving. Here, the mechanism will not know the value of the wallet to appropriately set deposits. However, imagine that the client Alice herself provides the value W she wishes to be protected or *insured* against. The mechanism can now proceed as if each player's utility $V = W/t$ from collusion is known.

As before, if a whistle is blown, the mechanism will first give W to Alice (to make her whole). Effectively, Alice will never lose her money, even if the committee colludes to recover her private key.

Client-committee collusion. There is an important new subtlety to account for here. We need to ensure that the client Alice has no incentive to "cheat" the mechanism by colluding with the committee. Broadly, we now need to capture two forms of collusion.

First, when Alice is not part of the colluding coalition, the coalition is exogenously specified the value V by Alice, making the resulting game identical to the standard retaliation game $\mathbb{R}$.

Now, for client-committee collusion, the concern is that Alice attempts "insurance fraud"—that is, she exaggerates the value of her wallet so as to receive a higher payout, which she then redistributes amongst the coalition. A crucial observation we make here is that due to budget balance of $\mathcal{M}^{\mathbb{W}}$, the joint utility of Alice and the coalition is non-positive. This is because any payment from $\mathcal{M}^{\mathbb{W}}$ to Alice comes from slashing the committee, while the utility gained by colluding committee members through learning the secret key comes directly at the expense of Alice. As a result, the *total utility* of any coalition containing Alice cannot be positive and therefore, Alice cannot profit by exaggerating the value of her assets.

Insurance pricing. To prevent abuse in practical deployment, we must also ensure that the client Alice does not request unnecessary insurance. The concern is that Alice does this, not to profit but rather to force the committee members to lock up larger deposits. Therefore, it may make sense for Alice to be required to pay an *insurance premium*, calculated as a function of the value W she requests insurance for. The pricing of such insurance is an interesting market design question in itself but out of scope for this paper. However, note that pricing insurance can be done independent of the whistleblowing mechanism, as once the value V from colluding is exogenously set, the game proceeds exactly the same way independent of the mechanism used to arrive at V.

9.2.2 Randomness Beacon Service

Consider a distributed randomness beacon (DRB) service where a committee generates verifiable randomness for applications to use. Recall that this setting permits whistleblowing proofs of collusion when a self-verifiable threshold VRF is used to build the DRB (see Section 4.4).

Suppose now that a third-party application like a casino subscribes to the DRB service to run games for its own users. Note here that the casino is "a client" for the DRB service. Now, as a natural goal, the casino would like to ensure that it does not lose money because of collusion to learn the randomness before hand.

However, the DRB service should not ex-ante be expected to know the value attackers might be able to gain from each individual application that subscribes to use its randomness. Instead, following our insurance modeling, the casino application itself will self-report the value W it would like to be insured against based on what it stands to lose. The casino will pay the DRB committee an insurance premium to have each member lock up an additional W/t deposit. In case of whistleblowing, the casino will be reimbursed up to W.

Further, at any point, the casino may purchase additional insurance if e.g., the amount it stands to lose from collusion increases.

Multiple clients. Observe that the insurance perspective can also model e.g., multiple different applications using the same DRB service and still be protected against loss from a collusion attack. Here, the committee members will need to lock up deposits corresponding to the sum of insurance requested by the applications.

9.3 Limitations for Estimating Utility

Infeasible restrictions. In some settings, obtaining bounds on collusion utility may be infeasible without heavy restrictions. As an example, consider our public signing service (i.e., notary) application. In the generic case, collusion to obtain signatures on conflicting documents could provide arbitrarily large utility. Moreover, the client also might not know this utility making it hard to deploy an insurance-style mechanism. It is unclear what document restrictions would need to be placed in order to bound V.

Similarly, for accountable threshold decryption, in the generic case, the value of a collusion to recover the plaintext could be arbitrarily large. Only in specific deployments (such as encrypted mempools described earlier) can a suitable bound be obtained.

Dynamic utility changes. An orthogonal concern, especially in applications being run for extended periods of time, is whether the utility of collusion can change. Sometimes, as illustrated through our example applications, changes in utility are easy to observe.

In other cases, a reasonable estimate of how V changes is feasible. For instance, V may be affected by currency exchange rates if e.g., V was realized in Bitcoin but interoperability reasons necessitated deposits in USDC. Here, an assumption that the price of Bitcoin does not rise by e.g., 20% in the whistleblowing period, along with an analogously larger USDC deposit can be used.

On the other hand, sharp or sudden changes in V could substantially change the incentives of collusion. We posit that in such cases, it may be impossible to build good whistleblowing mechanisms. We leave further explorations to future work.

Conclusion

We have introduced a broad framework for *whistleblowing* with the goal of deterring collusion in cryptographic protocols. Whistleblowing protocols combine cryptographic tools for colluders to defect by ratting on each other, with mechanism design techniques to incentivize such defection.

We presented the new notion of smart collusion (i.e., collusion using blockchain smart contracts) and showed how it significantly boosts the colluders power. We showed generic cryptographic and game-theoretic results on the feasibility of whistleblowing.

Finally, we showed several existing and new cryptographic applications that can be captured in our whistleblowing framework, and would benefit from our analysis.

Acknowledgments

We thank Valeria Nikolaenko and the anonymous CCS reviewers for helpful suggestions. Joseph Bonneau was supported by a16z crypto, NSF grant CNS-2239975, and gifts from Google, Meta, and Handshake Labs. S. Matthew Weinberg was supported by the NSF CAREER Award CCF-1942497 and an IC3 grant.

References

[1] Ittai Abraham, Lorenzo Alvisi, and Joseph Y Halpern. “Distributed computing meets game theory: combining insights from two fields”. In: *ACM Sigact News* 42.2 (2011), pp. 69–76.

[2] Ittai Abraham, Dahlia Malkhi, and Alexander Spiegelman. "Asymptotically optimal validated asynchronous Byzantine agreement". In: *PODC*. 2019, pp. 337–346.

[3] Prabhanjan Ananth, Gilad Asharov, Hila Dahari, and Vipul Goyal. "Towards Accountability in CRS Generation". In: *Eurocrypt*. 2021, pp. 278–308.

[4] Gavin Andresen. *Bitcoin M-of-N Standard Transactions*. BIP-0011. 2011.

[5] Gilad Asharov, Ran Canetti, and Carmit Hazay. "Towards a Game Theoretic View of Secure Computation". In: *Eurocrypt*. 2011, pp. 426–445.

[6] Kushal Babel, Philip Daian, Mahimna Kelkar, and Ari Juels. "Clockwork Finance: Automated Analysis of Economic Security in Smart Contracts". In: *IEEE S&P*. 2023, pp. 2499–2516.

[7] Salvador Barbera and Matthew O Jackson. "Strategy-proof exchange". In: *Econometrica* 63.1 (1995), pp. 51–87.

[8] Joseph Bebel and Dev Ojha. *Ferveo: Threshold Decryption for Mempool Privacy in BFT networks*. Cryptology ePrint Archive, Paper 2022/898. 2022. URL: `https://eprint.iacr.org/2022/898`.

[9] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation". In: *STOC*. 1988, pp. 1–10.

[10] B Douglas Bernheim, Bezalel Peleg, and Michael D Whinston. "Coalition-proof nash equilibria i. concepts". In: *Journal of economic theory* 42.1 (1987), pp. 1–12.

[11] Dan Boneh and Chelsea Komlo. "Threshold Signatures with Private Accountability". In: *CRYPTO*. 2022, pp. 551–581.

[12] Dan Boneh, Aditi Partap, and Lior Rotem. "Accountability for Misbehavior in Threshold Decryption via Threshold Traitor Tracing". In: *CRYPTO*. 2024, pp. 317–351.

[13] Dan Boneh, Aditi Partap, and Lior Rotem. "Proactive Refresh for Accountable Threshold Signatures". In: *FC*. 2025, pp. 140–159.

[14] Dan Boneh, Aditi Partap, and Lior Rotem. "Traceable Secret Sharing: Strong Security and Efficient Constructions". In: *CRYPTO*. 2024, pp. 221–256.

[15] Dan Boneh, Aditi Partap, and Lior Rotem. *Traceable Verifiable Random Functions*. Cryptology ePrint Archive, Paper 2025/312. 2025. URL: `https://eprint.iacr.org/2025/312`.

[16] Dan Boneh, Aditi Partap, and Brent Waters. *Accountable Multi-Signatures with Constant Size Public Keys*. Cryptology ePrint Archive, Paper 2023/1793. 2023. URL: `https://eprint.iacr.org/2023/1793`.

[17] Jan Bormet et al. "Strong Secret Sharing with Snitching". In: *CRYPTO 2025*. 2025, pp. 295–327.

[18] Eric Budish, Andrew Lewis-Pye, and Tim Roughgarden. "The Economic Limits of Permissionless Consensus". In: *EC*. 2024, pp. 704–731.

[19] Sílvia Casacuberta, Julia Hesse, and Anja Lehmann. "SoK: Oblivious Pseudorandom Functions". In: *EuroS&P*. 2022, pp. 625–646.

[20] David Chaum. "Blind signatures for untraceable payments". In: *CRYPTO*. 1982, pp. 199–203.

[21] Jing Chen and Silvio Micali. "Collusive dominant-strategy truthfulness". In: *Journal of Economic Theory* 147.3 (2012), pp. 1300–1312.

[22] James Hsin-yu Chiang et al. *Detecting Rogue Decryption in (Threshold) Encryption via Self-Incriminating Proofs*. Cryptology ePrint Archive, Paper 2024/794. 2024. URL: `https://eprint.iacr.org/2024/794`.

[23] Tarun Chitra, Matheus V. X. Ferreira, and Kshitij Kulkarni. "Credible, Optimal Auctions via Public Broadcast". In: *AFT*. Vol. 316. LIPIcs. 2024, 19:1–19:16.

[24] Kevin Choi, Aathira Manoj, and Joseph Bonneau. "SoK: Distributed Randomness Beacons". In: *IEEE S&P*. 2023, pp. 75–92.

[25] Benny Chor, Amos Fiat, and Moni Naor. "Tracing Traitors". In: *CRYPTO*. 1994, pp. 257–270.

[26] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. "Private information retrieval". In: *FOCS*. 1995, pp. 41–50.

[27] Arka Rai Choudhuri et al. "Fluid MPC: Secure Multiparty Computation with Dynamic Participants". In: *CRYPTO*. 2021, pp. 94–123.

[28] Hao Chung and Elaine Shi. "Foundations of transaction fee mechanism design". In: *SODA*. 2023, pp. 3856–3899.

[29] Coinbase. *"What does Coinbase do with my digital assets?"*. link. 2023.

[30] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. "A Secure and Optimally Efficient Multi-Authority Election Scheme". In: *Eurocrypt*. 1997, pp. 103–118.

[31] Elizabeth C. Crites, Chelsea Komlo, and Mary Maller. "Fully Adaptive Schnorr Threshold Signatures". In: *CRYPTO*. 2023, pp. 678–709.

[32] Philip Daian et al. "Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability". In: *IEEE S&P*. 2020, pp. 585–602.

[33] Sourav Das, Benny Pinkas, Alin Tomescu, and Zhuolun Xiang. *Distributed Randomness using Weighted VUFs*. Cryptology ePrint Archive, Paper 2024/198. 2024. URL: `https://eprint.iacr.org/2024/198`.

[34] Alex Davidson et al. "Privacy Pass: Bypassing Internet Challenges Anonymously". In: *Proceedings on Privacy Enhancing Technologies* 2018 (June 2018), pp. 164–180.

[35] Alfredo De Santis, Yvo Desmedt, Yair Frankel, and Moti Yung. "How to share a function securely". In: *STOC*. 1994, pp. 522–533.

[36] Alan Deckelbaum and Silvio Micali. "Collusion, efficiency, and dominant strategies". In: *Games and Economic Behavior* 103 (2017), pp. 83–93.

[37] Nikhil R Devanur, Yuval Peres, and Balasubramanian Sivan. "Perfect bayesian equilibria in repeated sales". In: *SODA*. 2014, pp. 983–1002.

[38] Jack Doerner et al. "Threshold BBS+ Signatures for Distributed Anonymous Credential Issuance". In: *IEEE S&P*. 2023, pp. 773–789.

[39] Stefan Dziembowski, Sebastian Faust, and Tomasz Lizurej. "Individual Cryptography". In: *CRYPTO*. 2023, pp. 547–579.

[40] Stefan Dziembowski, Sebastian Faust, Tomasz Lizurej, and Marcin Mielniczuk. “Secret Sharing with Snitching”. In: *CCS*. 2024, 840–853.

[41] Meryem Essaidi, Matheus VX Ferreira, and S Matthew Weinberg. “Credible, strategyproof, optimal, and bounded expected-round single-item auctions for all distributions”. In: *arXiv preprint arXiv:2205.14758* (2022).

[42] Joan Feigenbaum, Christos Papadimitriou, and Scott Shenker. “Sharing the cost of muliticast transmissions (preliminary version)”. In: *STOC*. 2000, pp. 218–227.

[43] Matheus VX Ferreira, Yotam Gafni, and Max Resnick. “Incentive-Compatible Collusion-Resistance via Posted Prices”. In: *arXiv preprint arXiv:2412.20853* (2024).

[44] Matheus VX Ferreira and S Matthew Weinberg. “Credible, truthful, and two-round (optimal) auctions via cryptographic commitments”. In: *EC*. 2020, pp. 683–712.

[45] Georg Fuchsbauer and Mathias Wolf. “Concurrently Secure Blind Schnorr Signatures”. In: *Eurocrypt*. 2024, pp. 124–160.

[46] Aadityan Ganesh, Clayton Thomas, and S Matthew Weinberg. “Revisiting the primitives of transaction fee mechanism design”. In: *arXiv preprint arXiv:2410.07566* (2024).

[47] Juan Garay et al. “Rational Protocol Design: Cryptography against Incentive-Driven Adversaries”. In: *FOCS*. 2013, pp. 648–657.

[48] Sanjam Garg et al. “Cryptography with weights: MPC, encryption and signatures”. In: *CRYPTO*. 2023, pp. 295–327.

[49] Ivan Geffner and Moshe Tennenholtz. “Strengthening Nash Equilibria”. In: *arXiv preprint arXiv:2312.14745* (2023).

[50] Andrew V Goldberg and Jason D Hartline. “Collusion-resistant mechanisms for single-parameter agents”. In: *SODA*. 2005, pp. 620–629.

[51] Oded Goldreich, Silvio Micali, and Avi Wigderson. “How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority”. In: *STOC*. 1987, pp. 218–229.

[52] Tiantian Gong, Ryan Henry, Alexandros Psomas, and Aniket Kate. “More is Merrier: Relax the Non-Collusion Assumption in Multi-Server PIR”. In: *IEEE S&P*. 2024, pp. 4348–4366.

[53] Tiantian Gong, Aniket Kate, Hemanta K. Maji, and Hai H. Nguyen. *Disincentivize Collusion in Verifiable Secret Sharing*. Cryptology ePrint Archive, Paper 2025/446. 2025. URL: `https://eprint.iacr.org/2025/446`.

[54] Vipul Goyal, Yifan Song, and Akshayaram Srinivasan. “Traceable secret sharing and applications”. In: *CRYPTO*. 2021, pp. 718–747.

[55] Jerry Green and Jean-Jacques Laffont. “On coalition incentive compatibility”. In: *The Review of Economic Studies* 46.2 (1979), pp. 243–254.

[56] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. “Byzantine Agreement with a Rational Adversary”. In: *ICALP*. 2012, pp. 561–572.

[57] Joseph Halpern and Vanessa Teague. “Rational secret sharing and multiparty computation: extended abstract”. In: *STOC*. 2004, 623–632.

[58] Timo Hanke, Mahnush Movahedi, and Dominic Williams. "DFINITY Technology Overview Series, Consensus System". In: *CoRR* abs/1805.04548 (2018). arXiv: 1805.04548. URL: http://arxiv.org/abs/1805.04548.

[59] Nicole Immorlica, Brendan Lucier, Emmanouil Pountourakis, and Samuel Taggart. "Repeated sales with multiple strategic buyers". In: *EC*. 2017, pp. 167–168.

[60] Mahimna Kelkar et al. "Complete Knowledge: Preventing Encumbrance of Cryptographic Secrets". In: *CCS*. 2024, pp. 2415–2429.

[61] Mahimna Kelkar et al. *The Sting Framework: Proving the Existence of Superclass Adversaries*. Cryptology ePrint Archive, Paper 2024/1676. 2024. URL: https://eprint.iacr.org/2024/1676.

[62] Silvio Micali, Michael Rabin, and Salil Vadhan. "Verifiable random functions". In: *FOCS*. 1999, pp. 120–130.

[63] Hervé Moulin. "Incremental cost sharing: Characterization by coalition strategy-proofness". In: *Social Choice and Welfare* 16.2 (1999), pp. 279–320.

[64] Douglass C. North. "Institutions and Credible Commitment". In: *Journal of Institutional and Theoretical Economics (JITE) / Zeitschrift für die gesamte Staatswissenschaft* 149.1 (1993), pp. 11–23.

[65] *Osmosis Docs*. https://docs.osmosis.zone/overview.

[66] Michal Penn, Maria Polukarov, and Moshe Tennenholtz. "Congestion games with failures". In: *EC*. 2005, pp. 259–268.

[67] Ruben Recabarren and Bogdan Carbunar. "Tithonus: A Bitcoin Based Censorship Resilient System". In: *Proc. Priv. Enhancing Technol.* 2019.1 (2019), pp. 68–86.

[68] Tim Roughgarden. "Transaction Fee Mechanism Design". In: *Journal of the ACM* 71.4 (2024), pp. 1–25.

[69] Kazue Sako. "An Auction Protocol Which Hides Bids of Losers". In: vol. 1751. Jan. 2000, pp. 422–432. ISBN: 978-3-540-66967-8. DOI: 10.1007/978-3-540-46588-1_28.

[70] James Schummer. "Manipulation through bribes". In: *Journal of Economic Theory* 91.2 (2000), pp. 180–198.

[71] Adi Shamir. "How to share a secret". In: *Commun. ACM* 22.11 (1979), 612–613.

[72] Peiyao Sheng et al. "BFT protocol forensics". In: *CCS*. 2021, pp. 1722–1743.

[73] Elaine Shi, Hao Chung, and Ke Wu. "What can cryptography do for decentralized mechanism design". In: *arXiv preprint arXiv:2209.14462* (2022).

[74] drand Team. *drand: Cryptography*. https://drand.love/docs/cryptography/. 2025.

[75] Henry de Valence. *The Penumbra Protocol*. https://protocol.penumbra.zone/main/crypto/flow-encryption/threshold-encryption.html.

[76] Maofan Yin et al. "HotStuff: BFT Consensus with Linearity and Responsiveness". In: *PODC*. 2019, pp. 347–356.

[77] Mark Zhandry. *Optimal Traitor Tracing from Pairings*. Cryptology ePrint Archive, Paper 2024/867. 2024. URL: https://eprint.iacr.org/2024/867.

```
Game $\mathbf{G}^{\text{sound}}_{\mathcal{F}}$
1 : $(\mathsf{st}, n, t, \mathcal{C}) \leftarrow \mathcal{A}(1^{\lambda})$
2 : $(k_1, \ldots, k_n, \mathsf{vk}) \leftarrow\!\!\$\ \mathcal{F}.\mathsf{Gen}(1^{\lambda}, t, n)$
3 : $(\hat{x}, \hat{y}, \hat{\pi}) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{EvalO}(\cdot)}(\mathsf{st}, \{k_i\}_{i \in \mathcal{C}}, \mathsf{vk})$
4 : return $\mathcal{F}.\mathsf{Ver}(\mathsf{vk}, \hat{x}, \hat{y}, \hat{\pi}) = 1 \wedge |\mathcal{C}| < t \wedge \forall r^c \in \mathcal{R}_C, \mathsf{leak}(\hat{x}; r^c) \notin \mathcal{Q}$

    Oracle $\mathsf{EvalO}(i, x^*)$
    1 : $r_i^s \leftarrow\!\!\$\ \mathcal{R}_S$
    2 : $y_i^*, \pi_i \leftarrow \mathcal{F}.\mathsf{Respond}(k_i, x^*; r_i^s)$
    3 : $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{x^*\}$
    4 : return $\sigma$
```

Figure 2: The soundness game $\mathbf{G}^{\text{sound}}_{\mathcal{F}}$ for a functionality $\mathcal{F}$.

A Deferred Cryptography Details

Security properties for the computation functionality $\mathcal{F}$. Correctness requires that if all the servers are honest, then the output passes verification. More formally, for all $0 < t < n \in \mathbb{N}$, $(k_1, \ldots, k_n, \mathsf{vk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^{\lambda}, t, n)$, all inputs $x \in \mathcal{X}$, all randomness $r^c \in \mathcal{R}_C$, all subsets $\mathcal{J} \subseteq [n]$ of size t, and all randomness $r_i^s \in \mathcal{R}_S$ for all $i \in \mathcal{J}$, the following holds:

$$\Pr\left[\begin{array}{c} \mathcal{F}.\mathsf{Ver}(\mathsf{vk}, x, y, \pi) = 1 \wedge \forall i \in \mathcal{J} : \\ \mathcal{F}.\mathsf{PartialVer}(\mathsf{vk}, x^*, y_i^*, \pi_i) = 0 \end{array} \middle| \begin{array}{c} x^* \leftarrow \mathcal{F}.\mathsf{Query}(x; r^c) \\ \forall i \in \mathcal{J} : \\ (y_i^*, \pi_i) \leftarrow \mathcal{F}.\mathsf{Respond}(k_i, x^*; r_i^s) \\ (y, \pi) \leftarrow \mathcal{F}.\mathsf{Output}(\{y_i^*\}_{i \in \mathcal{J}}, r_c) \end{array}\right]$$

Further, soundness requires that it should be hard for a client to compute the function at a new input $\hat{x}$. This should hold even if it can see function evaluations at arbitrary inputs of its choice, and upto $t-1$ servers are corrupt. We formalize the notion of a 'new' input based on the leakage—the adversary is not allowed to query partial function evaluations on any input whose leakage belongs in $\mathsf{leak}_{\mathcal{F}}(\hat{x})$[7]. This definition can be seen as a generalization of the unforgeability definition for signatures to arbitrary functionalities. We formalize this requirement using a game-based definition, which is presented in Figure 2 and Definition 14.

Definition 14. A functionality $\mathcal{F}$ is sound if, for all $\lambda \in \mathbb{N}$ and all PPT adversaries $\mathcal{A}$, the following advantage is negligible in λ:

$$\mathsf{Adv}^{\text{sound}}_{\mathcal{F}}(\lambda)(\mathcal{A}) = \Pr[\mathbf{G}^{\text{sound}}_{\mathcal{F}}(\mathcal{A}) = 1]$$

Formalism for identifiable whistleblowing. We now describe how our model from Section 4 can be extended to capture applications such as accountable threshold signatures (e.g. multi-

[7]Note that this definition is meaningful only in the context of public or one-way function leakages. For oblivious leakage, one could define a variant of soundness similar to the one-more unforgeability property of blind signatures. However, we do not explore this definition in this paper, as our focus is on non-oblivious leakages.

signatures) where it is indeed possible to identify at least one server in the quorum offering off-band service:

- The $\mathcal{F}.\mathsf{Gen}$ algorithm outputs another verification key vk' which can be used to verify proofs that blame certain servers.
- The $\mathcal{F}.\mathsf{Output}$ algorithm now also outputs a set $\mathcal{J}$ of parties being blamed, along with a proof π'.
- We define a new algorithm $\mathcal{F}.\mathsf{VerBlame}$ that takes as input vk', the input-output pair (x, y), the set of servers being blamed $\mathcal{J}$ and the proof π', and outputs a single bit denoting whether all servers in $\mathcal{J}$ were indeed responsible for generating this output.
- We can define a strong unframability property for both the functionality $\mathcal{F}$ and the whistle-blowing protocol $\mathcal{W}_{\mathcal{F}}$, which says that even if $n-1$ parties collude, they should not be able to produce a valid proof π' that blames a set $\mathcal{J}^*$ containing an honest party. This can be seen as a generalization of the unforgeability property for multisignatures (see [16] and the references therein).

Example attack if OPRF privacy is relaxed to one-way leakage. OPRFs are widely used in applications such as keyword search, private set intersection, private information retrieval, and password-authenticated key-exchange. As the name suggests, these applications have oblivious leakage. Unfortunately, weakening this to one-way leakage would trivially break these applications. For example, consider Private set intersection (PSI), wherein two parties have sets S_1 and S_2, and the second party P_2 want to compute the intersection $S_1 \cap S_2$ without revealing anything about its set to the first party P_1. Let us assume that the first party has a OPRF key k. It starts with sending the OPRF evaluations $E = \{u_i\}_{i \in S_1}$ at all elements in its set S_1 to the second party P_2. P_2 and P_1 then run the OPRF protocol with P_2 querying P_1 for the evaluation v_j at all points $j \in S_2$. Then, for any $j \in S_2$, if v_j is in E, then P_2 adds j to its output. It is easy to see that P_2 output is equal to $S_1 \cap S_2$. Unfortunately, if P_2 is required to reveal $owf(j)$ for its OPRF queries to P_1, then P_1 can learn a lot about S_2. Specifically, it can learn $S_1 \cap S_2$ by comparing $owf(j)$ to the values $owf(i)$ for all elements i in its set S_1. A similar attack can be mounted on other applications of OPRFs as well. To conclude, whistleblowing is impossible for such applications, based on Section 4.3.1.

B Deferred Cryptography Proofs

B.1 Impossibility of Whistleblowing for Oblivious Leakage (Lemma 4.1)

Proof. Suppose that there is a functionality $\mathcal{F}$ (for a function f) with a whistleblowing protocol $(\mathbb{P}, \mathbb{V})$ that is ϵ-sound.

We will now show that the leakage cannot be ϵ-oblivious by constructing a distinguisher $\mathcal{D}$. Let $\mathbf{G}_{ob}$ denote the oblivious leakage game. $\mathcal{D}$ starts with sampling $0 < t < n \in \mathbb{N}$ and $(k_1, \ldots, k_n, \mathsf{vk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, t, n)$. It then samples $x_0, x_1 \leftarrow\$ \mathcal{X}$, sends them to its challenger, and gets back x^*. Additionally, $\mathcal{D}$ runs the $\mathsf{Query}, \mathsf{Respond}$ and Output functions to get a function evaluation at x_0, which we denote by y_0. In other words, $(x, y) \in f_k$.

To distinguish whether x^* came from $\mathsf{leak}(x_0)$ or $\mathsf{leak}(x_1)$, $\mathcal{D}$ simply runs the whistleblowing prover for $(\mathsf{vk}, L = \{x^*\})$ with the witness (x_0, y_0). More formally, it runs $\Pi \leftarrow\!\!\$\ \mathbb{P}((\{x^*\}, \mathsf{vk}), (x_0, y_0))$. If $\mathbb{V}(\{x^*\}, \mathsf{vk}, \Pi)$ outputs 1, then $\mathcal{D}$ guesses 1 (i.e., that x^* came from $\mathsf{leak}(x_1)$); otherwise it guesses 0 (i.e., that x^* came from $\mathsf{leak}(x_0)$).

To compute the advantage of $\mathcal{D}$, let E_0 and E_1 denote the cases $b = 0$ and $b = 1$ respectively, where b is the bit sampled by the challenger in the oblivious leakage game. First, by completeness of the whistleblowing protocol, we have that for all $L_1 \subseteq \mathsf{leak}(x_1)$:

$$\Pr\left[\mathbb{P}((L_1, \mathsf{vk}), (x_0, y_0)) \leftrightarrow \mathbb{V}(L_1, \mathsf{vk}) \Rightarrow 1\right] = 1.$$

This implies that $\mathcal{D}$ will always output one if $b = 1$, because $\{x^*\}$ is a subset of $\mathsf{leak}(x_1)$ in that case. More formally, $\Pr[\mathbf{G}_{ob}(\mathcal{D}) = 1|\mathsf{E}_1] = 1$.

Next, to analyse the $b = 0$ case, we claim that we can use $\mathcal{D}$ to construct an adversary $\mathcal{A}$ that breaks unframability with advantage equal to $\Pr[\mathbf{G}_{ob}(\mathcal{D}) = 1|\mathsf{E}_0]$. Specifically, $\mathcal{A}$ samples $0 < t < n \in \mathbb{N}$, sends $t, n, \mathcal{C} = \perp$ to its challenger, and receives vk. It then samples x_0, x_1, and $x^* \leftarrow\!\!\$\ \mathsf{leak}(x_0)$. It queries its unframability challenger for t partial evaluations on x^*, for an arbitrary subset of t servers, and combines the outputs to compute y_0. Note that at this point, the leakage set stored by the unframability challenger only includes x^*. Then, similar to $\mathcal{D}$, $\mathcal{A}$ runs the honest prover of the whistleblowing protocol, to get $\Pi^* \leftarrow\!\!\$\ \mathbb{P}((\{x^*\}, \mathsf{vk}), (x_0, y_0))$, and sends Π^* to its challenger.

Now, observe that the distribution of $\mathsf{vk}, x_0, x_1, x^*, y_0$ and Π^* in the above unframability game is identical to that in the oblivious leakage game when $b = 0$. Hence, the probability that $\mathcal{A}$ wins the unframability game is equal to the probability that the verifier $\mathbb{V}$ accepts Π^*, which is $\Pr[\mathbf{G}_{ob}(\mathcal{D}) = 1|\mathsf{E}_0]$. Since the whistleblowing protocol is ϵ-unframeable, this means that $\Pr[\mathbf{G}_{ob}(\mathcal{D}) = 1|\mathsf{E}_0]$ is less than ϵ.

Finally, we compute the advantage of $\mathcal{D}$:

$$\begin{aligned}\mathsf{Adv}^{\text{ob}-\text{leak}}_{\mathcal{F}}(\mathcal{D}) &= \left|\frac{1}{2} \cdot (\Pr[\mathbf{G}_{ob}(\mathcal{D}) = 1|\mathsf{E}_1] + \Pr[\mathbf{G}_{ob}(\mathcal{D}) = 0|\mathsf{E}_0]) - \frac{1}{2}\right| \\ &\geq \left|\frac{1}{2} \cdot (1 + 1 - \epsilon) - \frac{1}{2}\right| \\ &\geq \frac{(1 - \epsilon)}{2} \qquad (5)\end{aligned}$$

This proves that if the whistleblowing protocol is ϵ-unframable, then there exists a distinguisher that breaks oblivious leakage with non-negligible advantage. □

B.2 Whistleblowing Protocol for other Leakages

We complete the proof deferred from Section 4.3.2.

Completeness. This follows directly from the completeness of the NIZK-PoK system.

Anonymity. This also follows trivially from the honest-verifier zero-knowledge property of the NIZK-PoK system.

Unframability. Theorem B.1 below proves unframability of the above whistleblowing protocol by relying on the knowledge-soundness of the NIZK-PoK proof and the soundness of the functionality $\mathcal{F}$.

Theorem B.1. *For every* PPT *adversary $\mathcal{A}$ that breaks unframability of the whistleblowing protocol, there exist another* PPT *adversary $\mathcal{B}$ such that,*

$$\mathsf{Adv}^{\text{unfram}}_{\mathcal{W}_{\mathcal{F}}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{sound}}_{\mathcal{F}}(\lambda) + \kappa$$

where κ is the knowledge soundness error of the whistleblowing protocol.

Proof. Given an adversary $\mathcal{A}$ for the unframability game, we will use it to construct an adversary $\mathcal{B}$ for the soundness of the functionality $\mathcal{F}$. Let $\mathsf{E_c}$ denote the event that $\mathcal{A}$ corrupts less than t servers, i.e. $|\mathcal{C}| < t$. Note that $\mathcal{A}$ can win its game only if this event occurs. This implies,

$$\mathsf{Adv}^{\text{unfram}}_{\mathcal{W}_{\mathcal{F}}}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{unfram}}_{\mathcal{W}_{\mathcal{F}}}(\mathcal{A}) = 1 | \mathsf{E}_c]$$

In other words, we only need to focus on adversaries that corrupt less than t servers.

We now describe how $\mathcal{B}$ works:

- Receive $(n, t, \mathcal{C})$ from $\mathcal{A}$. Send these to the soundness challenger.
- Receive $\mathsf{vk}, \{k_i\}_{i \in \mathcal{C}}$ from the challenger, and forward to $\mathcal{A}$.
- For every EvalO oracle query, simply forward it to its challenger, and return its response to $\mathcal{A}$.

Eventually, $\mathcal{A}$ returns a proof Π. Note that $\mathcal{A}$ can be seen as a NIZK-PoK prover for the statement $(\mathsf{vk}, \mathcal{Q})$, which returns a valid proof with probability $\Pr[\mathbf{G}^{unfram}_{\mathcal{W}_{\mathcal{F}}}(\mathcal{A}) = 1 | \mathsf{E}_c]$. Hence, $\mathcal{B}$ runs the extractor $\mathcal{E}$ on $\mathcal{A}$ to extract the witness $(\hat{x}, \hat{y})$. $\mathcal{B}$ simply outputs $(\hat{x}, \hat{y})$. (Note that there is no $\hat{\pi}$ since we are only considering self-verifiable functionalities.)

Let E_e be the event that the extractor succeeds. Observe that, if this event occurs, then, $(\hat{x}, \hat{y})$ must be a valid witness, which means, $\mathcal{F}.\mathsf{Ver}(\mathsf{vk}, \hat{x}, \hat{y}) = 1$, and the list $\mathcal{Q}$ does not contain any leakage for $\hat{x}$, i.e. $\forall r^c \in \mathcal{R}_C$, $\mathsf{leak}(\hat{x}; r^c) \notin Q$. This is exactly the winning condition of $\mathcal{B}$. Formally, this means that,

$$\mathsf{Adv}^{\text{sound}}_{\mathcal{F}}(\mathcal{B}) \geq \Pr[\mathsf{E}_e].$$

By knowledge soundness of the whistleblowing protocol, we know that

$$\Pr[\mathsf{E}_e] \geq \Pr[\mathbf{G}^{\text{unfram}}_{\mathcal{W}_{\mathcal{F}}}(\mathcal{A}) = 1 | \mathsf{E}_c] - \kappa.$$

This proves the theorem. □

C Existence of a "Reasonable" Second-Stage Pure Nash Equilibrium

In this section, we motivate the assumption in Theorem E.1 on the existence of a second-stage pure Nash equilibria that leaves the coalition with a joint positive utility across the two stages.

All the whistle-blowing mechanisms discussed thus far have a second-stage pure Nash equilibrium. In other words, there exists some second-stage pure Nash equilibrium that the players can coordinate on if they choose to collude, but rather *choose* to remain honest since some player receives a negative utility from colluding. The lower bound in Theorem E.1 considers only such mechanisms with a second-stage pure Nash equilibrium.

On the other hand, we can also construct whistleblowing mechanisms that do not have a second-stage pure Nash equilibrium (Example 1). In such mechanisms, all SPPNE will involve players playing honest in the first stage purely because no second-stage pure Nash equilibrium exists when all players play collude. In other words, the onus of "creating a second-stage pure Nash equilibrium" lies on the retaliation mechanism.

Even though whistleblowing mechanisms with no second-stage pure Nash equilibria deter collusion, from a modelling perspective, such mechanisms should be considered "different" to whistleblowing mechanisms that have a second-stage pure Nash equilibrium.

Example 1. Consider the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ given by

1. $\mathbf{P}^{\mathbb{W}}(w) = \begin{cases} 81/90 & \text{if } w = 90, \\ 1 & \text{if } w = 100, \\ 0 & \text{if } w = 110, \\ 0 & \text{otherwise} \end{cases}$

   and,

2. $\mathbf{S}^{\mathbb{W}}(w) = \begin{cases} 0 & \text{if } w = 0, \\ 1 & \text{if } w = 90, \\ 15.2 & \text{if } w = 100, \\ 5 & \text{if } w = 110, \\ 15.2 + 1/2^w & \text{otherwise.} \end{cases}$

Proposition C.1. *The whistle-blowing game induced by $\mathcal{M}^{\mathbb{W}}$ does not have a pure Nash equilibrium when the number of players k that know the whistleblowing proof equals* 2.

Proof. It is not hard to see that the two players jointly blowing w whistles for $w \neq 90, 100, 110$ whistles is not an equilibrium. If $w = 0$, then, one of the players is better off by blowing a 100 whistles. If $w \notin \{0, 90, 100, 110\}$, then, each player is better off blowing 111 more whistles, which reduces the whistleblowing slash without changing the whistleblowing reward.

We now have to rule out equilibria where the players jointly blow 90, 100 or 110 whistles. First, assume player 1 blows all w whistles. If $w = 90$, then player 1 is better off blowing 100 whistles instead. If $w = 100$, player 2 is better off blowing 10 additional whistles to get slashed 5 instead of getting slashed 15.2 from blowing no whistles. If $w = 110$, player 1 is better off blowing 10 less whistles.

Now, consider players 1 and 2 blowing w_1 and w_2 whistles respectively such that $w = w_1 + w_2$. Without loss of generality, assume $w_1 \geq w_2$.

If $w = 90$, consider player 1 blowing 10 additional whistles. By blowing w_1 whistles, player 1 receives a utility $\frac{81}{90}w_1 - 1$, but by blowing $w_1 + 10$ whistles, it receives a utility $1 \times (w_1 + 10) - 15.2 = w_1 - 4.8$. Note that $w_1 - 4.8 \geq \frac{81}{90}w_1 - 1$ whenever $w_1 \geq 38$. Since, $w_1 + w_2 = 90$ and $w_1 \geq w_2$, w_1 is at least 45, and therefore, player 1 benefits from deviating and blowing 10 additional whistles.

IF $w = 100$, consider player 2 blowing 10 less whistles (we will consider the case where $w_2 < 10$ later). By blowing w_2 whistles, player 2 gets a utility $1 \times w_2 - 15.2$, while by blowing $w_2 - 10$ whistles, it receives a utility $\frac{81}{90}(w_2 - 10) - 1 = \frac{81}{90}w_2 - 10$. The latter is greater than the former whenever $w_2 \leq 52$. Indeed, $w_1 + w_2 = 100$ and $w_2 \leq w_1$, and thus, $w_2 < 52$. Therefore, player 2 is better off deviating and blowing 10 less whistles.

If $w = 100$ and $w_2 < 10$, then player 2 is better off blowing 10 more whistles. By blowing w_2 whistles, it receives a utility $w_2 - 15.2 < 10 - 15.2 = -5.2$. By blowing 10 more whistles, it receives a utility $0 - 5 = -5$, which is strictly better.

Finally, if $w = 110$, player 1 is better of blowing 10 less whistles to receive a reward $(w_1 - 10) - 15.2 > -5$, the reward from blowing 110 whistles in total.

Thus, $\mathbb{W}$ does not have a pure Nash equilibrium. □

It might also be the case that all pure Nash equilibria induced by $\mathcal{M}^{\mathbb{W}}$ can be extremely "unrealistic." For example, in r-whistle mechanisms, every player blowing $n \times r$ whistles is a pure Nash equilibria for any $k > 2$. Even if some player deviates and blows a different number of whistles, the total number of whistles blown is strictly greater than r and thus, all players receive a zero reward from the whistleblowing mechanism and are slashed their entire deposit. However, one can claim that whistleblowing mechanisms with such "unrealistic" equilibria should "spiritually" be considered similarly as whistleblowing mechanisms with no second-stage pure Nash equilibria. Assuming that the coalition gets a positive joint utility in the pure Nash equilibrium across the two stages necessarily ensures that the equilibrium is "interesting."

Theorem E.1 argues that the 1-whistle $(n-1)\,\gamma$-secure mechanism is optimal amongst all whistleblowing mechanisms that support a second stage pure Nash equilibrium where the coalition earns a net positive utility across the two stages of $\mathbb{W}$.

However, there can also exist whistleblowing mechanisms with "interesting" second stage equilibria that results in the joint net utility of the coalition in $\mathbb{W}$ being negative.

Example 2. Consider the 2-whistle mechanism $\mathcal{M}^{\mathbb{W}}$ given by

1. $$\mathbf{P}^{\mathbb{W}}(w) = \begin{cases} V + n\gamma + \varepsilon & \text{if } w = 1, \\ \varepsilon & \text{if } w = 2, \\ 0 & \text{otherwise} \end{cases}$$

   and,

2. $$\mathbf{S}^{\mathbb{W}}(w) = \begin{cases} 0 & \text{if } w = 0, \\ V + \gamma + \frac{\varepsilon}{2} & \text{otherwise.} \end{cases}$$

Note that blowing 0 or 1 whistle is not an equilibrium when $k \geq 2$. Any of the k players blowing zero whistles is incentivized to blow a whistle in both cases. Further, note that 2 players blowing a whistle each is an equilibrium. Neither of the whistleblowers will want to blow a different number of whistles (the whistleblowing slash remains the same, but the reward decreases from ε to zero if they blow a different number of whistles), while the other players are indifferent between blowing no whistles and blowing some number of whistles.

The joint utility of the coalition across the two stages, given by $nV - n \times (V + \gamma + \frac{\varepsilon}{2}) + 2\varepsilon$ is strictly negative. However, the Nash equilibrium where two players blow a whistle each is certainly "interesting," unlike the equilibrium where every player blows much more than r whistles in a r-whistle mechanism.

We leave the problem of classifying all whistleblowing mechanisms with "interesting" pure Nash equilibria and finding the optimal whistleblowing mechanism (i.e, the mechanism $\mathcal{M}^{\mathbb{W}}$ with a given whistleblowing deposit $D^{\mathbb{W}}$ that requires the largest retaliation deposit to be broken) open.

D Alternate Models for the Whistleblowing and Retaliation Games

In this section, we explore various other models for the whistleblowing and retaliation games that could arise naturally in applications.

D.1 Whistleblowing through Jointly-Controlled Accounts

In the discussions thus far, we have implicitly assumed that the players blow whistles from privately controlled accounts to which the whistleblowing rewards are transferred (the *private accounts model*). Thus, the deposit collected by the retaliation mechanism must also be large enough so that the whistleblowers do not "abscond" after getting rewarded by the whistleblowing mechanism without sharing it with the remainder of the players. We can also consider alternate collusion models, where the players collude by blowing whistles through jointly controlled wallets. Since the whistleblowing rewards are transferred to the joint wallets, the retaliation mechanism is not responsible to prevent the whistleblowers from absconding. We present two possible models for whistleblowing and retaliation from joint accounts. In the first model, some player will still have to submit a whistleblowing proof on behalf of the joint account, while in the second, blowing whistles from joint accounts is fully automated.

Retaliation with jointly controlled wallets but non-automated whistleblowing. We start by describing the model with jointly controlled accounts where the players still have to blow whistles on behalf of the account — we call this the *non-automated joint accounts model.* While the first round is identical to the private accounts model, the action-space for the players $1, \dots, k$ in the second round consists of $(u_i, v_i) \in (\mathbb{N} \cup \{0\})^2$. u_i denotes the number of whistles blown by player i from its private accounts while v_i denotes the number of whistles it blows on behalf of joint accounts.

Each player then derives a utility $u_i \mathbf{P}^{\mathbb{W}}(\sum_i u_i + v_i) - \mathbf{S}^{\mathbb{W}}(\sum_i u_i + v_i)$ from the whistleblowing mechanism and an additional utility equal to $\mathcal{R}_i(\sum_i u_i + v_i)$ from the retaliation mechanism.[8] Since the retaliation mechanism collects some of the whistleblowing rewards and distributes it back to the players, budget balance for $\mathcal{R}$ would now require $\sum_i \mathcal{R}(v) \leq v\mathbf{P}^{\mathbb{W}}(v)$ for all $v \in \mathbb{N} \cup \{0\}$. In other words, the net reward distributed by $\mathcal{R}$ cannot be larger than the total money it receives from the whistleblowing mechanism. The non-automated joint accounts model is otherwise identical to the private accounts model.

It should generally be possible to stop players from blowing whistles from jointly controlled accounts. Using such joint accounts effectively amounts to whistleblowing through smart contract addresses instead of fresh privately-owned addresses (e.g., EOAs on Ethereum). However, since it is typically visible if the calling entity is an EOA or a smart contract, a whistleblowing mechanism may choose to only allow EOAs to blow whistles.

Further, even if joint accounts arise through private means (e.g., a trusted hardware that controls the key), we note that if desired, the whistleblowing protocol designer could prevent joint accounts through recent cryptographic techniques [39, 60] that allow proving *individual* ownership of keys. This also prevents automated whistleblowing from joint wallets.

[8] To be precise, the retaliation mechanism $\mathcal{R}_i$ should take as input the total number of whistles $\sum_i u_i + v_i$ blown by the players (which can be learnt by $\mathcal{R}$ since $\mathcal{M}^{\mathbb{W}}$ publishes this information anyways) and the joint accounts which received a reward from $\mathcal{M}^{\mathbb{W}}$. However, for notational cleanliness, we do not explicitly mention the joint accounts as input to $\mathcal{R}$, but they can be deduced from the pair (u_i, v_i) played by player i.

Even if the protocol designer cannot contain coalitions from using joint accounts, we will argue that breaking a whistleblowing mechanism is equally difficult with purely private accounts and with jointly controlled accounts as long as whistleblowing cannot be automated.

Theorem D.1. *A whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ can be broken by a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ with retaliation deposit $D^{\mathbb{R}}$ and some SPPNE $\vec{w}^2$ in the private accounts model if and only if there exists a retaliation mechanism $\mathcal{R}$ in the non-automated joint accounts model with the same retaliation deposit $D^{\mathbb{R}}$ such that $\mathcal{R}$ and some SPPNE $((u_1, v_1), \ldots, (u_n, v_n))^2$ breaks $\mathcal{M}^{\mathbb{W}}$.*

Proof. Proving the "only if" direction is easy — if there exists a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ and $\vec{w}^2$ that breaks $\mathcal{M}^{\mathbb{W}}$ purely with private accounts, then $\mathcal{M}^{\mathbb{R}}$ and $\vec{w}^2$ also breaks $\mathcal{M}^{\mathbb{W}}$ in the non-automated joint accounts model. Formally, $\mathcal{R}_i(w) = -\mathcal{M}^{\mathbb{R}}_i(w)$.

We use the following intuition to prove the "if" direction. While the retaliation mechanism gets direct control of the funds flowing into the jointly controlled accounts, instead of blowing whistles from the joint accounts of the coalition, players can potentially blow whistles from their private accounts and then abscond with all the whistleblowing rewards without sharing them. For instance, if the coalition strategically decides that player i must blow u_i whistles from private accounts and v_i whistles from joint accounts, player i might instead blow $w_i = u_i + v_i$ whistles from its private accounts and pocket all of the whistleblowing rewards that would have otherwise flown into the joint accounts. We will argue that stopping the above "redirection" attack requires as much collateral as stopping the player from absconding in the private accounts model.

Consider a retaliation mechanism $\mathcal{R}$ and SPPNE $((u_1, v_1), \ldots, (u_n, v_n))^2$ that breaks $\mathcal{M}^{\mathbb{W}}$ in the joint accounts model. We will show a reduction to construct a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ in the private accounts model that also breaks $\mathcal{M}^{\mathbb{W}}$. Instead of first handing over the whistleblowing rewards to the coalition and then receiving some share of the rewards via $\mathcal{R}$, we will construct $\mathcal{M}^{\mathbb{R}}$ such that each player directly receives the rewards from $\mathcal{M}^{\mathbb{W}}$ and then forwards the money to be redistributed. Define $\mathcal{M}^{\mathbb{R}}$ to be

$$\mathcal{M}^{\mathbb{R}}_i(\sum_j w_j) = \begin{cases} v_i \mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathcal{R}_i(\sum_j w_j) & \text{if } \sum_j w_j = \sum_j u_j + v_j, \\ D^{\mathbb{R}} & \text{otherwise.} \end{cases}$$

We argue that $(w_1, \ldots, w_n)^2$ is a SPPNE in the retaliation game $\mathbb{R}$ where $w_i = u_i + v_i$ for all $1 \le i \le n$.

To this extent, we will begin by showing that $(w_1, \ldots, w_n)$ is a second-stage Nash equilibrium in the private accounts model. Further, when all players $j \neq i$ blow w_j whistles, we argue that player i's utility from blowing w_i whistles under the retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ equals the utility from blowing u_i private whistles and v_i whistles from joint accounts under $\mathcal{R}$. Player i's utility from blowing w_i whistles equals

$$\begin{aligned}
&w_i\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) - \mathcal{M}_i^{\mathbb{R}}(\sum_j w_j) \\
&= (u_i + v_i)\mathbf{P}^{\mathbb{W}}(\sum_j u_j + v_j) - \mathbf{S}^{\mathbb{W}}(\sum_j u_j + v_j) - \mathcal{M}_i^{\mathbb{R}}(\sum_j u_j + v_j) \\
&= u_i\mathbf{P}^{\mathbb{W}}(\sum_j u_j + v_j) - \mathbf{S}^{\mathbb{W}}(\sum_j u_j + v_j) \\
&\qquad\qquad + (v_i\mathbf{P}^{\mathbb{W}}(\sum_j u_j + v_j) - \mathcal{M}_i^{\mathbb{R}}(\sum_j u_j + v_j)) \\
&= u_i\mathbf{P}^{\mathbb{W}}(\sum_j u_j + v_j) - \mathbf{S}^{\mathbb{W}}(\sum_j u_j + v_j) + \mathcal{R}_i(\sum_i u_i + v_i),
\end{aligned}$$

which equals the utility from blowing u_i private whistles and v_i whistles on behalf of joint accounts. On the other hand, by deviating and blowing $w'_i \neq w_i$ whistles in the private accounts model, player i receives a second-stage utility

$$\begin{aligned}
&w'_i\mathbf{P}^{\mathbb{W}}(w'_i + \sum_{j\neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w'_i + \sum_{j\neq i} w_j) - D^{\mathbb{R}} \\
&\leq w'_i\mathbf{P}^{\mathbb{W}}(w'_i + \sum_{j\neq i} u_j + v_j) - \mathbf{S}^{\mathbb{W}}(w'_i + \sum_{j\neq i} u_j + v_j) \\
&\qquad\qquad + \mathcal{R}_i(w'_i + \sum_{j\neq i} u_j + v_j) \\
&\leq u_i\mathbf{P}^{\mathbb{W}}(u_i + v_i + \sum_{j\neq i} u_j + v_j) - \mathbf{S}^{\mathbb{W}}(u_i + v_i + \sum_{j\neq i} u_j + v_j) \\
&\qquad\qquad + \mathcal{R}_i(u_i + v_i + \sum_{j\neq i} u_j + v_j) \\
&= w_i\mathbf{P}^{\mathbb{W}}(w_i + \sum_{j\neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i + \sum_{j\neq i} w_j) - \mathcal{M}_i^{\mathbb{R}}(w_i + \sum_{j\neq i} w_j)
\end{aligned}$$

The second line follows since $\mathcal{R}$ cannot penalize player i more than $D^{\mathbb{R}}$. The third line follows since blowing u_i private whistles and v_i whistles from joint accounts is a Nash equilibrium in the non-automated joint accounts model — the left hand side is the utility from blowing w'_i private whistles while the right hand side is the equilibrium utility. The final equality follows from the definition of $\mathcal{M}^{\mathbb{R}}$.

To conclude the proof, we need to argue that $(w_1, \ldots, w_n)^2$ satisfies Equation (4) for all players i (i.e, each player earns a net positive utility across the two rounds) and is therefore, a SPPNE. However, it immediately follows since the players receive a utility identical to $((u_1, v_1), \ldots, (u_n, v_n))^2$, which satisfies Equation (4). Thus, $(w_1, \ldots, w_n)^2$ is indeed a SPPNE and hence, $\mathcal{M}^{\mathbb{R}}$ breaks $\mathcal{M}^{\mathbb{W}}$. □

Retaliation with jointly controlled wallets and automated whistleblowing. The reduction in Theorem D.1 crucially uses the fact that players can deviate from the coalition's strategy to not blow whistles on behalf of the joint accounts, but rather blow whistles from accounts that are their own. To circumvent such deviations by the players, the coalition can look to automate whistle-blowing from joint accounts (call this the *automated joint accounts model*). In the automated joint accounts model, players are still free to blow whistles from their private accounts. However, they will not be able to stop whistles from getting submitted on behalf of the accounts jointly controlled by the coalition (if the coalition decides to submit one).

Formally, automated whistleblowing modifies the retaliation game as follows. The first round is identical to the two models discussed thus far — players either collude to receive a utility V each or remain honest. In the second round however, the coalition submits v whistles from jointly controlled accounts for some v determined before the start of the first round (thus, the players cannot tamper with these v whistles). Player i's action in the second round, as in the private accounts model, consists of blowing $w_i \in \mathbb{N} \cup \{0\}$ whistles from its privately owned accounts. For a retaliation mechanism $\mathcal{R}$, player i receives a utility $w_i \mathbf{P}^{\mathbb{W}}(\sum_j w_j + v) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j + v)$ from the whistleblowing mechanism and $\mathcal{R}_i(\sum_j w_j + v)$ from the retaliation mechanism. The model is otherwise identical to the non-automated joint accounts model.

We will denote the second-stage actions taken by the players and the coalition in the retaliation game $\mathbb{R}$ by $((w_1, \dots, w_n), v)$. Similar to the private accounts model, the strategy profile (collude, $\dots$, collude), $((w_1, \dots, w_n), v)$ will then be denoted by $((w_1, \dots, w_n), v)^{\mathbb{2}}$.

We argue that it is strictly harder to prevent players from colluding in the automated joint accounts model than in the private accounts model (and therefore, the non-automated joint accounts model). In particular, we will sketch a retaliation mechanism that does not collect any deposit from the players, but still breaks the 1-whistle skeleton mechanism which was sufficient to prevent collusion in the private accounts model.

Proposition D.2. *There exists a retaliation mechanism $\mathcal{R}$ that collects no deposits such that $\mathcal{R}$ and $((0, \dots, 0), 1)^{\mathbb{2}}$ breaks the 1-whistle skeleton mechanism.*

Proof. Recall that the 1-whistle skeleton disburses a reward $V + \varepsilon$ in the case of a sole whistleblower, but never rewards whistleblowers otherwise. It penalizes all players $V + \frac{\varepsilon}{n-1}$ even if a single whistle is blown.

Consider the retaliation mechanism that sets up exactly 1 joint account and blows a whistle on behalf of this account. Further, it distributes $\frac{V+\varepsilon}{n}$ to each of the n players if and only if no other whistle is blown.

In the second round, observe that none of the players are incentivized to blow any whistles — on top of the $V + \frac{\varepsilon}{n-1}$ slashed by the whistleblowing mechanism, the players receive $\frac{V+\varepsilon}{n}$ from blowing no whistles, but a zero utility from blowing a non-zero number of whistles. Thus, $((0, \dots, 0), 1)$ is a second-stage equilibrium. We can quickly verify that each player receives a utility $\frac{V}{n} - \varepsilon(\frac{1}{n-1} - \frac{1}{n}) > 0$ across the two stages of $\mathbb{R}$ by playing $((0, \dots, 0), 1)^{\mathbb{2}}$, which is better than the utility from staying honest. Thus, $((0, \dots, 0), 1)^{\mathbb{2}}$ is a SPPNE and the retaliation mechanism discussed above and $((0, \dots, 0), 1)^{\mathbb{2}}$ breaks the 1-whistle skeleton mechanism. □

However, the 1-whistle $(n-1)\,\gamma$-secure mechanism $\mathbb{M}_{1*}^{\mathbb{W}}$ can be modified to collect a slightly larger deposit to be secure against retaliation mechanisms with $D^{\mathbb{R}} \leq (n-1)\,\gamma$ that use automated whistles. Consider the following "aggressive" 1-whistle $(n-1)\,\gamma$-secure mechanism:

1. $\mathbf{P}^{\mathbb{W}}(w) = \begin{cases} \frac{n}{n-1}V + n\,\gamma + \varepsilon & \text{if } w = 1, \\ 0 & \text{otherwise} \end{cases}$
   for some small $\varepsilon > 0$, and,

2. $\mathbf{S}^{\mathbb{W}}(w) = \begin{cases} 0 & \text{if } w = 0, \\ \frac{n}{n-1}V + \gamma + \frac{\varepsilon}{n-1} & \text{otherwise.} \end{cases}$

Note that $\mathcal{M}^{\mathbb{W}}$ collects a deposit $D^{\mathbb{W}} = \frac{n}{n-1}V + \gamma + \frac{\varepsilon}{n-1}$.

Proposition D.3. *The aggressive* 1*-whistle* $(n-1)\,\gamma$*-secure mechanism cannot be broken by any retaliation mechanism that collects a deposit of at most* $(n-1)\,\gamma$ *from each of its players, irrespective of the number of players* k *that learn the whistleblowing proof.*

Unlike the proof approach sketched in Section 6.3, we run a top-down dynamic program to argue Proposition D.3. We start by arguing that no SPPNE consists of one or more whistles getting blown. We then rule out a second-stage Nash equilibrium with no whistles.

Proof. Assume for contradiction that $\vec{w}^2$ is a SPPNE in the retaliation game $\mathbb{R}$ induced by some retaliation mechanism $\mathcal{R}$, where $\vec{w}$ is a pure Nash equilibrium consisting of one or more whistles blown. We will argue that the coalition jointly receives a negative utility, and thus, some player receives a negative utility from colluding in the first stage and will deviate to play honest.

If one or more whistles are blown in the second phase, each player is slashed $\frac{n}{n-1}V + \gamma + \frac{\varepsilon}{n-1}$ for a total slash of $\frac{n^2}{n-1}V + n\gamma + \frac{n}{n-1}\varepsilon$. The whistleblowers, on the other hand, are rewarded at most $\frac{n}{n-1}V + n\,\gamma + \varepsilon$ in total. Thus, the net utility of the coalition across the two rounds can be upper bounded by

$$nV + \left(\frac{n}{n-1}V + n\,\gamma + \varepsilon\right) - \left(\frac{n^2}{n-1}V + n\gamma + \frac{n}{n-1}\varepsilon\right) < 0.$$

The first term (nV) is the collusion utility from the first round. Thus, the coalition achieves a negative utility by colluding and blowing one or more whistles.

Finally, we will argue that blowing no whistles cannot be a second-stage equilibrium for any retaliation mechanism $\mathcal{R}$ with $D^{\mathbb{R}} \leq (n-1)\,\gamma$. If no whistles are blown, some player i strictly profits by deviating and blowing a whistle.

$$\begin{aligned} &1 \times \mathbf{P}^{\mathbb{W}}(1) - \mathbf{S}^{\mathbb{W}}(1) + \mathcal{R}_i(1) \\ &\quad \geq \left(\frac{n}{n-1}V + n\,\gamma + \varepsilon\right) - \left(\frac{n}{n-1}V + \gamma + \frac{\varepsilon}{n-1}\right) - (n-1)\,\gamma \\ &\quad > 0. \end{aligned}$$

Thus, blowing no whistles cannot be a second-stage equilibrium.

Hence, no retaliation mechanism with a deposit $(n-1)\,\gamma$ can break the aggressive 1-whistle $(n-1)\,\gamma$-secure mechanism in the joint accounts model with automated whistleblowing. ☐

D.2 Anonymous vs Non-Anonymous Whistleblowing Models

We have assumed anonymous whistleblowing in all the models considered so far — i.e, the identities of the whistleblowers remain unknown to both the whistleblowing mechanism and the retaliation mechanism. In this section, we consider a relaxed non-anonymous model of whistleblowing where the coalition learns exactly how many whistles were blown by each player. In other words, the retaliation mechanism can depend on the number of whistles blown by each of the n players.

We will argue that non-anonymity will not help coalitions in designing better retaliation mechanisms — for a given whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$, if there exists a retaliation mechanism and a SPPNE that breaks $\mathcal{M}^{\mathbb{W}}$ in the non-anonymous model, then, there exists a retaliation mechanism with the same deposit that breaks $\mathcal{M}^{\mathbb{W}}$ in the anonymous model.

Theorem D.4. *Suppose $\mathcal{M}^{\mathbb{W}}$ is broken by a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ that learns the number of whistles blown by each player and a SPPNE $(w_1, \dots, w_n)^2$. Then, there exists an anonymous retaliation mechanism $\hat{\mathcal{M}}^{\mathbb{R}}$ that collects the same retaliation deposit as $\mathcal{M}^{\mathbb{R}}$ such that $\hat{\mathcal{M}}^{\mathbb{R}}$ and $(w_1, \dots, w_n)^2$ breaks $\mathcal{M}^{\mathbb{W}}$.*

Proof. At a high level, the retaliation mechanism expects the players to coordinate on the second-stage equilibrium $(w_1, \dots, w_n)$ and thus, expects $\sum_i w_i$ whistles to be blown in total. If a different number of whistles are submitted to $\mathcal{M}^{\mathbb{W}}$, then the retaliation mechanism does not have to investigate which player i deviated from blowing w_i whistles — it can penalize everybody.

Construct $\hat{\mathcal{M}}^{\mathbb{R}}$ as follows.

$$\hat{\mathcal{M}}_i^{\mathbb{R}}(w) = \begin{cases} \mathcal{M}_i^{\mathbb{R}}(w_1, \dots, w_n) & \text{if } w = \sum_i w_i, \\ D^{\mathbb{R}} & \text{otherwise.} \end{cases}$$

It is not hard to see that $\hat{\mathcal{M}}^{\mathbb{R}}$ does not need access to the exact number of whistles blown by each of the n players. It only needs the aggregate number of whistles w blown to execute, which is published by $\mathcal{M}^{\mathbb{W}}$. Further, it also requires only the same retaliation deposit as $\mathcal{M}^{\mathbb{R}}$.

We begin by arguing that $(w_1, \dots, w_n)$ is a second-stage equilibrium (Equation (3)) in the retaliation game induced by $\hat{\mathcal{M}}_i^{\mathbb{R}}$. Observe that

$$\begin{aligned}
& w_i \mathbf{P}^{\mathbb{W}}(w_i + \sum_{j \neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i + \sum_{j \neq i} w_j) - \hat{\mathcal{M}}^{\mathbb{R}}(w_i + \sum_{j \neq i} w_j) \\
&= w_i \mathbf{P}^{\mathbb{W}}(w_i + \sum_{j \neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i + \sum_{j \neq i} w_j) - \mathcal{M}^{\mathbb{R}}(w_1, \dots, w_n) \\
&\geq w_i' \mathbf{P}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \mathcal{M}^{\mathbb{R}}(w_i', w_{-i}) \\
&\geq w_i' \mathbf{P}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - D^{\mathbb{R}} \\
&= w_i' \mathbf{P}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \mathbf{S}^{\mathbb{W}}(w_i' + \sum_{j \neq i} w_j) - \hat{\mathcal{M}}^{\mathbb{R}}(w_i' + \sum_{j \neq i} w_j).
\end{aligned}$$

Here (w_i, w_{-i}) is the vector obtained by replacing w_i by w_i' in $\vec{w}$. The inequality in the third line follows since $\vec{w}$ is a second-stage equilibrium in $\mathbb{R}$ induced by $\mathcal{M}^{\mathbb{R}}$. The fourth line follows since

$\mathcal{M}^{\mathbb{R}}(\cdot) \leq D^{\mathbb{R}}$. The final equality follows from the construction of $\hat{\mathcal{M}}^{\mathbb{R}}$. Thus, $(w_1, \ldots, w_n)$ is a second-stage equilibrium in the retaliation game induced by $\hat{\mathcal{M}}^{\mathbb{R}}$.

We can immediately conclude that $\vec{w}^2$ is a SPPNE is the retaliation game induced by $\hat{\mathcal{M}}^{\mathbb{R}}$. When each player i blows w_i whistles, their net utility across the two rounds is identical between the two retaliation mechanisms $\mathcal{M}^{\mathbb{R}}$ and $\hat{\mathcal{M}}^{\mathbb{R}}$. Thus, if all players receive a positive utility under $\mathcal{M}^{\mathbb{R}}$ (since $\vec{w}$ is a SPPNE under $\mathcal{M}^{\mathbb{R}}$), then, all players also receive a net positive utility under $\hat{\mathcal{M}}^{\mathbb{R}}$. Thus, $\vec{w}^2$ is a SPPNE.

We have therefore proved that $\hat{\mathcal{M}}^{\mathbb{R}}$ and $\vec{w}^2$ breaks $\mathcal{M}^{\mathbb{W}}$. □

D.3 Non-automatic Retaliation

Our core model considers "self-executing" retaliation mechanisms. That is, $\mathcal{M}^{\mathbb{R}}$ can observe the public output of $\mathcal{M}^{\mathbb{W}}$ (e.g., the number of whistles blown) and directly trigger any retaliation penalties. While Remark 6 shows how $\mathcal{M}^{\mathbb{R}}$ can easily be made self-executing, here we also show that our results do not change even otherwise.

Specifically, we now consider a model where a player has to manually trigger the retaliation mechanism. The intuition is that this weakens the collusion's $\mathcal{M}^{\mathbb{R}}$ since even if a whistle is blown, no player would want to trigger a "burn everything" retaliation mechanism since it would also cause them to incur penalties.

We observe however that a different retaliation mechanism will achieve the collusion's goal of preventing players from whistleblowing. As long as some player receives a positive utility from the retaliation mechanism irrespective of the number of whistles blown, then the two models — with automatic and manually-triggered retaliation mechanisms — become equivalent. Such a player is then incentivized to trigger the retaliation mechanism.

More specifically, when $k < n$, any retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ can be modified to always yield some player a positive utility. Suppose $\mathcal{M}^{\mathbb{R}}_i(w) < 0$ for all k players that have access to the whistleblowing proof. Then, modify $\mathcal{M}^{\mathbb{R}}$ to redirect the retaliation penalties as rewards to the $n - k$ players that do not know the whistleblowing proof. The intuition is that this incentivizes colluding players who cannot whistleblow themselves to retaliate if they see another party blowing the whistle. A SPPNE in the retaliation game $\mathbb{R}$ induced by $\mathcal{M}^{\mathbb{R}}$ will remain an equilibrium even for the modified protocol $\widetilde{\mathcal{M}}^{\mathbb{R}}$. The payoffs for the $n-k$ players that don't know the whistleblowing proof only increases, and thus, they will want to continue colluding in the first stage of $\mathbb{R}$ (and they don't have any actions in the second). On the other hand, the payoffs for the k potential whistleblowers remain identical under $\mathcal{M}^{\mathbb{R}}$ and $\widetilde{\mathcal{M}}^{\mathbb{R}}$, and they would not prefer to change their actions. Thus, when $k < n$, the automatic and manually-triggered retaliation models are equivalent and all our results carry over to the manual retaliation model *mutatis mutandis*.

When $k = n$, to incentivize colluders to relay information about whistles, instead of slashing the retaliation deposits of all colluders, $\widetilde{\mathcal{M}}^{\mathbb{R}}$ will function as follows: If only one player submits a retaliation, $\widetilde{\mathcal{M}}^{\mathbb{R}}$ will provide that player a reward (and slash others). But if more than one retaliation is submitted, $\widetilde{\mathcal{M}}^{\mathbb{R}}$ will slash all retaliation deposits. Observe that is similar in spirit to how we designed our whistleblowing mechanism $\mathbb{M}^{\mathbb{W}}_{1*}$—except now, the colluders use the same strategy for their retaliation contract.

Intuitively, this creates a classic "prisoner's dilemma" situation. All colluding players will be incentivized to submit a retaliation if they observe any whistleblowing on $\mathcal{M}^{\mathbb{W}}$. But if they do so, they will all be slashed. In turn, such a retaliation contract $\widetilde{\mathcal{M}}^{\mathbb{R}}$ (with suitably large $D^{\mathbb{R}}$) will incentivize colluders to not whistleblow to $\mathcal{M}^{\mathbb{W}}$, showing the impossibility of unconditional security. We note that our standard 1-whistle $(n-1)\,\gamma$-secure mechanism $\mathbb{M}^{\mathbb{W}}_{1*}$ continues to deter collusion up to a maximum retaliation deposit of $(n-1)\,\gamma$.

E Deferred Game-Theoretic Proofs

E.1 Proof of Theorem 7.1

Consider any retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ with a retaliation deposit $D^{\mathbb{R}} \leq (n-1)\,\gamma$. We will calculate the bottom-up dynamic program described in Section 6.3 to calculate all SPPNE of the induced retaliation game $\mathbb{R}$.

First, observe that any pure Nash equilibria in the second round consists of some player blowing at least one whistle. Indeed, if nobody blows any whistles, player i receives a zero second-round utility. On the other hand, if i deviates and blows a whistle, it is rewarded $nV + n\gamma + \varepsilon$, slashed $V + \gamma + \frac{\varepsilon}{n-1}$ by $\mathcal{M}^{\mathbb{W}}$ and is penalized at most $(n-1)\,\gamma$ by $\mathcal{M}^{\mathbb{R}}$, which yields a utility at least $(n-1)V + (1 - \frac{1}{n-1})\,\varepsilon > 0$. Thus, blowing no whistles is not a second-stage equilibrium.

Next, we observe that if some whistle is blown, then, there exists some player who is better off playing `honest` in the first round than colluding. If two or more whistles are blown, then, every player is slashed at least $V + \gamma$ and is not rewarded at all by $\mathcal{M}^{\mathbb{W}}$. Even with the collusion reward V, all of them end up with a negative net utility. On the other hand, suppose that exactly one whistle is blown. We show that the $(n-1)$ players that did not blow the whistle will end up with a negative joint utility, and thus, at least one of them must receive a negative utility across the two rounds. The $(n-1)$ players together receive a collusion reward equal to $(n-1)V$ in the first stage and are slashed $(n-1)\,(V + \gamma + \frac{\varepsilon}{n-1})$ in the second by the whistleblowing mechanism. They are awarded at most $(n-1)\,\gamma$ by the retaliation mechanism, leaving them with a joint utility of at most $-\varepsilon < 0$, as required.

Thus, no retaliation mechanism with a deposit at most $(n-1)\,\gamma$ can break the 1-whistle $(n-1)\,\gamma$-secure mechanism.

E.2 Proof of Theorem 7.2

We will argue that the retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ described below and some $\vec{w}^2$ will break any whistleblowing mechanism with deposit up to $V + \gamma$. $\mathcal{M}^{\mathbb{R}}$ penalizes player 1 nothing if no whistles are blown, and collects $(n-1)\,\gamma$ from player 1 and distributes γ each to the remaining $(n-1)$ players even if a single whistle is blown.

1. $$\mathcal{M}^{\mathbb{R}}_1(w) = \begin{cases} 0 & \text{if } w = 0, \\ (n-1)\,\gamma & \text{otherwise.} \end{cases}$$

2. $$\mathcal{M}^{\mathbb{R}}_i(w) = \begin{cases} 0 & \text{if } w = 0, \\ -\gamma & \text{otherwise,} \end{cases}$$

   for $i \neq 1$.

We will analyze two separate cases — the largest utility $\max_w w\mathbf{P}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w)$ that player 1 can derive from the whistleblowing mechanism is (a) smaller than $(n-1)\,\gamma$ and (b) larger than $(n-1)\,\gamma$.

In the former case, $\mathcal{M}^{\mathbb{R}}$ will disincentivize player 1 from blowing any whistles. The whistleblowing reward it receives is smaller than the retaliation penalty. Formally, suppose that $w\mathbf{S}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w) \leq 0 = 0 \times \mathbf{P}^{\mathbb{W}}(0) - \mathbf{S}^{\mathbb{W}}(0)$ for all $w \in \mathbb{N}$. Then, blowing zero whistles is a second-stage Nash equilibrium. Indeed, all of the n players will receive a net utility V across the two stages. Thus, both Equation (3) and Equation (4) are satisfied for all $1 \leq i \leq n$, and $(0, \ldots, 0)^2$ is a subgame perfect Nash equilibrium.

In the latter case, player 1 will blow whistles to optimize its second-stage utility and share $(n-1)\,\gamma$ of the whistleblowing rewards with the other $(n-1)$ players. Suppose that $w = \arg\max_{\hat{w}} \hat{w}\, \mathbf{P}^{\mathbb{W}}(\hat{w}) - \mathbf{S}^{\mathbb{W}}(\hat{w})$ and $w\mathbf{P}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w) \geq (n-1)\,\gamma$. Then, blowing w whistles is a second-stage pure Nash equilibrium (Equation (3) is satisfied for player 1). To verify that Equation (4) is satisfied, note that all players $i \neq 1$ are slashed $\mathbf{S}^{\mathbb{W}}(w) \leq V + \gamma$ by the whistleblowing mechanism and are allocated γ by the retaliation mechanism. Thus, their net utility across both the rounds

$$\begin{aligned} U_i^{\mathbb{R}}((w, 0 \ldots, 0)^2) &= V - \mathbf{S}^{\mathbb{W}}(w) + \mathcal{M}_i^{\mathbb{R}}(w) \\ &\leq V - (V + \gamma) + \gamma \\ &= 0. \end{aligned}$$

Player 1 on the other hand receives a utility $w\mathbf{P}^{\mathbb{W}}(w) - \mathbf{S}^{\mathbb{W}}(w) \geq (n-1)\,\gamma$ from $\mathcal{M}^{\mathbb{W}}$ and is charged $(n-1)\,\gamma$ by the retaliation mechanism. Thus, it receives a positive second-stage utility and therefore a positive utility across the two stages.

Thus, $\mathcal{M}^{\mathbb{R}}$ breaks any whistleblowing mechanism that collects a deposit at most $V + \gamma$ for some strategy profile $\vec{w}^2$.

E.3 Optimality of $\mathbb{M}_{1*}^{\mathbb{W}}$

Theorem E.1. *Let $\mathcal{M}^{\mathbb{W}}$ be a whistleblowing mechanism with a second-stage pure Nash equilibrium $\vec{w} = (w_1, \ldots, w_n)$ such that the joint utility $\sum_i U_i^{\mathbb{R}}((w_1, \ldots, w_n)^2) = \sum_i \big(V + w_i\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j)\big)$ of the n players across the two rounds is strictly positive. If $\mathcal{M}^{\mathbb{W}}$ collects a whistleblowing deposit $V + \gamma$, then there exists $\mathcal{M}^{\mathbb{R}}$ that collects a deposit $(n-1)\,\gamma$ such that $\mathcal{M}^{\mathbb{R}}$ and $(w_1, \ldots, w_n)^2$ breaks $\mathcal{M}^{\mathbb{W}}$.*

Proof. At a high level, we will consider a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ that reallocates utility from each player that receives a positive utility in the equilibrium $\vec{w} = (w_1, \ldots, w_n)$ (call them *receivers*) to the remaining players (call them *defaulters*) irrespective of the number of whistles blown. Since the utilities of the players are shifted by the same constant irrespective of their second-round actions, $\vec{w}$ remains an equilibrium. We have to ensure that the redistribution occurs so that each player ends up with a positive net utility $U_i^{\mathbb{R}}(\vec{w}^2)$ across the two stages.

Consider the retaliation mechanism that collects $\min\{(n-1)\,\gamma, w_i\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) + V\}$ from each receiver i and distributes $\max\{\mathbf{S}^{\mathbb{W}}(\sum_j w_j) - w_{i'}\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - V, 0\}$ to each defaulter i', irrespective of the number of whistles blown.

It is fairly straightforward to argue that $\vec{w}^2$ satisfies Equation (3) and Equation (4) for all players i. Each receiver is penalized at most an additional V on top of their whistleblowing reward $w_i\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j)$, and thus combined with the collusion reward V in the first round, their net utility across the rounds is non-negative. Similarly, each defaulter is compensated at least V less than what they lose in the whistleblowing mechanism, and thus, they receive a net non-negative utility combined with the collusion reward V. Hence, $\vec{w}^2$ is a SPPNE in $\mathbb{R}$.

It is easy to verify that $\mathcal{M}^{\mathbb{R}}$ collects a deposit at most $(n-1)\,\gamma$. Indeed, only the receivers are charged a positive penalty by $\mathcal{M}^{\mathbb{R}}$ and are charged at most $(n-1)\,\gamma$. Thus, $D^{\mathbb{R}} \leq (n-1)\,\gamma$.

It only remains to verify budget balance of $\mathcal{M}^{\mathbb{R}}$, i.e, that the net penalty charged by $\mathcal{M}^{\mathbb{R}}$ is non-negative. We consider two cases for the same — (a) all receivers are charged $w_i\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) + V$ and (b) some receiver is charged a penalty $(n-1)\,\gamma$. In the first case, the total penalty charged by $\mathcal{M}^{\mathbb{R}}$ from both receivers and defaulters is bounded below by

$$\sum_i \Big(w_i\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - \mathbf{S}^{\mathbb{W}}(\sum_j w_j) + V\Big) \geq 0.$$

The inequality follows since the left hand side is the joint net utility of the coalition across the two stages in the whistleblowing game $\mathbb{W}$ for the SPPNE $\vec{w}^2$, which is larger than zero, as assumed in the proposition statement. In the second case when some receiver i is penalized $(n-1)\,\gamma$, note that no defaulter is rewarded more than γ by $\mathcal{M}^{\mathbb{R}}$.

$$\begin{aligned}\mathcal{M}^{\mathbb{R}}_{i'}(w) &\leq \mathbf{S}^{\mathbb{W}}(\sum_j w_j) - w_{i'}\mathbf{P}^{\mathbb{W}}(\sum_j w_j) - V \\ &\leq \mathbf{S}^{\mathbb{W}}(\sum_j w_j) - V \\ &\leq (V+\gamma) - V \leq \gamma.\end{aligned}$$

The last line follows since the whistleblowing deposit $D^{\mathbb{W}} \leq V+\gamma$. It immediately follows that the $(n-1)\,\gamma$ penalty collected from receiver i is sufficient to cover the compensations rolled out by $\mathcal{M}^{\mathbb{R}}$ to all the defaulters. Thus, $\mathcal{M}^{\mathbb{R}}$ satisfies budget balance as required.

To summarize, $\mathcal{M}^{\mathbb{R}}$ with $D^{\mathbb{R}} \leq (n-1)\,\gamma$ and $\vec{w}^2$ breaks the whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ with $D^{\mathbb{W}} \leq V+\gamma$. $\square$

E.4 Deferred details for (t,n)-collusion

Below, we state the lower bound result for whistleblowing in the (t,n)-setting. The proof proceeds analogously to the lower bound proofs in the (n,n) setting.

Proposition E.2. *1. For any whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ in the (t,n)-collusion environment which collects a deposit $D^{\mathbb{W}}$, there exists a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ with a retaliation deposit $D^{\mathbb{R}} = (n-1)\times D^{\mathbb{W}}$ such that $\mathcal{M}^{\mathbb{R}}$ and some SPPNE $\vec{w}^2$ breaks $\mathcal{M}^{\mathbb{W}}$.*

2. Further, if the number of players k that know the whistleblowing proof equals 1, then any whistleblowing mechanism $\mathcal{M}^{\mathbb{W}}$ with a whistleblowing deposit $D^{\mathbb{W}} = V+\gamma$ can be broken by some retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ and a SPPNE $\vec{w}^2$, where $D^{\mathbb{R}} = (t-1)\,\gamma$.

3. Finally, if there exists a second-stage Nash equilibrium in the whistleblowing game $\mathbb{W}$ induced by $\mathcal{M}^{\mathbb{W}}$ with $D^{\mathbb{W}} = V+\gamma$ such that the joint utility of the coalition of t players across the two stages is positive, then, there exists a retaliation mechanism $\mathcal{M}^{\mathbb{R}}$ with deposit $D^{\mathbb{R}} = (t-1)\,\gamma$ and some SPPNE that breaks $\mathcal{M}^{\mathbb{W}}$.