Beyond the Anonymous Inbox: Secure Whistleblowing for All

Gabriel Wechta
gabriel.wechta@nask.pl
NASK — National Research Institute
Warsaw, Poland

Mirosław Kutyłowski
miroslaw.kutylowski@nask.pl
NASK — National Research Institute
Warsaw, Poland

Tomasz Lizurej
tomasz.lizurej@nask.pl
NASK — National Research Institute
Warsaw, Poland

Ewa Syta
ewa.syta@trincoll.edu
Trinity College
Hartford, USA

Abstract

Directive (EU) 2019/1937 mandates reporting channels that do substantially more than accept a one-shot anonymous report submission. A compliant system must protect the confidentiality of the report and the reporting person, restrict submission to entitled persons, enforce prescribed workflows under statutory deadlines, and maintain auditable records of the reporting process. Existing whistleblowing tools typically address only fragments of these requirements, often reducing the problem to an anonymous inbox or web form. To our knowledge, no existing system addresses whistleblowing's full procedural and security requirements within a single integrated design.

We adopt a compliance-driven approach in which the Directive is treated as a normative source of system requirements rather than as background motivation. From Directive-mandated procedures, assets, and threats we derive, we propose a threat model, a set of security goals tailored to directive-compliant whistleblowing, and formalize the whistleblowing process.

We then propose a concrete Whistleblowing System architecture that separates enrollment from reporting and builds on an Auditable Log Service with an Identity Management Component. Our design supports report pseudonyms tied to entitled persons, controlled identity disclosure, and confidential yet accountable communication. It enforces an auditable workflow that separates procedural handling from merits-based processing and enables a practical deployment model deployable at both state and organization level that distributes trust across multiple entities, lowers adoption barriers through shared infrastructure, and supports two-level auditing of both procedural compliance and merits-based handling for dispute resolution.

Keywords

Whistleblowing, Identity Protection, Auditability

1 Introduction

Whistleblowing, defined as the lawful disclosure of information reasonably believed to evidence wrongdoing to an authorized entity [17, 37], is an important mechanism for identifying conduct that threatens the public interest, including fraud, corruption, violations of consumer and environmental protections, and large-scale security or compliance failures. In practice, however, reporting wrongdoing is inherently risky [39, 54]. Even when legal protections exist, prospective whistleblowers may still face retaliation, professional harm, social pressure, or simply distrust that the organization will handle the report properly. High-profile disclosures such as the Panama Papers [20, 22], Dieselgate [18], and LuxLeaks [16] demonstrate the societal value of whistleblowing, while also highlighting the personal risks faced by whistleblowers.

Prior work has argued that anonymous reporting alone is not enough: effective protection also depends on how the entire reporting process is governed, what guarantees it provides, and whether disputes about submission, timing, or handling can later be resolved [5, 36]. In practice, additional challenges arise, including support for confidential bidirectional communication and resistance to misuse, abuse, and selective handling of reports. These concerns often delay or prevent reporting altogether, undermining both accountability and public trust. Consequently, the security of whistleblowing should be understood as protecting the *entire* reporting workflow, not only the content of a report.

This observation is particularly relevant in the context of Directive (EU) 2019/1937, which requires EU Member States to establish secure internal and external reporting channels for persons reporting breaches of Union law [21]. The Directive does not model whistleblowing as a one-shot anonymous submission. Instead, it prescribes a robust reporting workflow with statutory deadlines: confidentiality must be preserved throughout, receipt must be acknowledged within seven days, handling must be performed by designated personnel, and feedback must be provided within a bounded period of up to three months. In effect, the Directive imposes concrete, time-bound operational requirements and requires organizations to both *ensure* and *demonstrate* compliance.

The gap between the requirements outlined in the Directive and real-world deployments is vast. Governance-oriented guidance and standards such as ISO 37002 [29] and Transparency International's guidelines [53] conceptualize whistleblowing as a full workflow process, but remain largely non-technical. By contrast, much of the technical literature models whistleblowing primarily as a one-shot anonymous submission channel [1, 45], while other works address only isolated aspects such as text sanitization, anonymization, or pseudonymization [5, 36, 49]. Even strong source-protection systems developed for journalists, such as SecureDrop [24] or CoverDrop [2], focus on confidential communication and do not provide support for the Directive-prescribed workflow and its security guarantees. See Section 7 for a detailed discussion.

As a result organizations often struggle with compliance in practice [56], as designing and deploying such systems raises a set of fundamental technical challenges that have received relatively little

attention in the specific setting imposed by the Directive, with a *specific set of legally imposed and real-world deployment constraints*.

Confidentiality and accountability. Simple anonymous submission mechanisms offer no way to prove that a report was received and processed, or ignored, leaving whistleblowers vulnerable and organizations unable to demonstrate compliance. Ensuring a level of anonymity is therefore necessary but not sufficient. Although protecting the whistleblower's identity is essential to realizing the Directive's protective intent, purely anonymous systems enable a range of abuses, including false or malicious reporting, framing attacks against targeted individuals or organizations, harassment and defamation, and automated spam or denial-of-service attacks that degrade or disable reporting channels. On the other hand, strong accountability mechanisms may introduce unconstrained risks to confidentiality of the report or whistleblower identity.

Directive-compliant systems must balance confidentiality and accountability across multiple stakeholders with conflicting incentives. From the whistleblower's perspective, the system must guarantee that an honestly submitted report cannot be altered, discarded, or ignored without detection, and that the whistleblower can later prove that a report was submitted at a particular time, as this establishes the start of their protected status. In addition, the system must protect the whistleblower's identity and provide assurances that the report is processed in accordance with the prescribed workflow. From the organization's perspective, the system must address issues of false claims of submission, fabricated timelines, or tampering with records, while preserving confidentiality and minimizing unnecessary disclosure. At the regulatory level, there must exist a mechanism to demonstrate compliance, such as adherence to deadlines and proper handling, without exposing sensitive report content or whistleblower identities. Achieving all of these properties simultaneously is non-trivial.

Entitlement and identity protection. Moreover, the Directive applies to specific but broad categories of reporting persons, such as workers, contractors, or associated individuals, creating a requirement to verify entitlement (eligibility to submit a report) without making the reporting event itself identifying. This exposes a fundamental tension. A compliant system must ensure that only authorized individuals can submit reports, while *simultaneously* preventing the organization or any single authority from learning who the whistleblower is, except in the limited circumstances where disclosure is legally permitted.

Time-bounded guarantees. Lastly, timeliness of processing is particularly important in this setting. Many systems state liveness as a secondary concern; here, delayed or denied processing is a core concern. Under the Directive, whistleblower protections apply only after a report has been submitted. An organization may claim that a report was never received, or delay acknowledging or processing it, in order to avoid obligations or discourage further reporting. At the same time, a dishonest individual may attempt to claim whistleblower protections without having actually filed a report.

In this paper, we treat the Directive not merely as background motivation but as a normative specification that should shape the system design from the beginning. Our starting point is that compliance should be built into the design, rather than argued only after implementation. This leads us to a design centered on two ideas. First, the reporting workflow should be represented as auditable and robust rather than as a one-shot report submission. Second, trust should be distributed so that no single entity controls critical aspects of report handling such as identity disclosure, while the system provides verifiable guarantees rather than relying solely on organizational trust or policy assurances.

At a high level, our design separates whistleblowing into *enrollment* and *reporting*. The Enrollment System (ES) is organization-specific and establishes entitlement in advance, before any report exists. Later, when a person decides to report, they use an enrollment token obtained from the organization together with an identity certificate obtained from a certification authority to derive a report pseudonym and submit a report without a new identifying interaction with the organization. The Reporting System (RS) then processes the report through an auditable workflow and communication and can be deployed as shared infrastructure across multiple organizations. Procedural tasks such as report validation, acknowledgment, assignment, archiving, and closure are performed by *managing operators* acting in a shared (t, n)-threshold model and restricted to well-defined tasks that do not involve human judgment, while handling of the report on the merits is performed by an assigned *operator* involving human judgment. To support oversight of this workflow, the system provides two levels of audit: process-level audit of procedural handling and privileged individual-report audit of merits-based handling.

The Whistleblowing System (WS), which consists of the the Enrollment System and Reporting System, relies on two technical building blocks: an Auditable Log Service (ALS), which supports auditable communication between all entities and logging of workflow actions, and an Identity Management Component (IMC), which supports report pseudonyms and controlled identity disclosure. In this way, we aim to move whistleblowing system design from trust-based compliance claims toward verifiable guarantees.

Contributions. We make the following contributions.

- We present a compliance-driven methodology that derives system procedures, protected assets, threats, and required protections directly from the Directive, providing explicit traceability from legal obligations to workflow design, threat modeling, and technical protections (Section 2).
- We formalize a model for Directive-compliant whistleblowing, including the entities, report structure, time bounds, threat model, and system and design goals (Section 4).
- We propose a whistleblowing system architecture (Section 3), specify two main technical building blocks (Section 5), and then provide a concrete system design for Directive-compliant whistleblowing (Section H).
- We analyze how the proposed design satisfies the security properties and compliance with the Directive (Section I).
- We discuss practical instantiation of ALS and IMC (Section F.3 and Section G.3, respectively), present a deployment model for Directive-compliant whistleblowing (Section 6), and provide a discussion of related work and future directions (Sections 7 and 8).

2 Directive-driven System Requirements

This section provides an overview of the key aspects of the Directive (EU) 2019/1937 [21], and it explains how the Directive shapes the

design of the WS we propose. This is important as we do not treat the Directive merely as policy background. Instead, we use it as a normative source of report workflow and timing obligations as well as security expectations and requirements that jointly determine what a compliant system must provide.

2.1 Overview of the Directive

The Directive's primary objective is to establish minimum obligations to create a whistleblowing system,[1] define formal requirements regarding their operation, and create a legal framework for protecting reporting persons (whistleblowers) from retaliation which each Member State must adopt and implement. More precisely, the Directive establishes a minimum framework for reporting breaches of Union law within the material scope of [Art.2], while allowing Member States to provide broader protection. The purpose of the reporting system is to enable the submission and processing of reports from persons who obtained information on breaches in *work-related context*. This extends beyond traditional employees to include several other categories of persons, such as workers, self-employed persons, shareholders, volunteers, trainees, contractors and former workers, as outlined in [Art.4].

In the technical sense used throughout this paper, a person's entitlement to use the system means that the reporting person falls within the protected personal scope of [Art.4(1a–1d)]. While [Art.4(1)] additionally requires that the reported information be acquired in a work-related context, this requirement is assessed at a later stage, once the reporting person who filed the report was claimed entitled and the report has handled to human processing. Entitlement therefore does not mean that the "allegation" has already been verified on the merits. It means that the reporting person is *entitled* to utilize the whistleblowing system.

A compliant whistleblowing system must provide secure channels that protect the confidentiality of the identity of the reporting person and of any third party mentioned in the report [Art.9(1a)], [Art.16(1)]. It must acknowledge receipt of an internal report within seven days [Art.9(1b)], diligently follow up on the report through designated personnel, provide feedback within three months [Art.9(1f)], and, where appropriate, refer the case to another competent authority. For external reporting, this authority must also acknowledge receipt within seven days, unless the reporting person explicitly requested otherwise or acknowledgment would jeopardize the protection of the reporting person's identity [Art.11(2b)], and must provide feedback within a reasonable timeframe not exceeding three months, or six months in justified cases [Art.11(2d)].[2] In addition, records of received reports must be kept in compliance with confidentiality requirements [Art.18(1)], while personal data that are not relevant for handling a specific report should not be collected or, if accidentally collected, should be deleted without undue delay [Art.17].

The reporting process must preserve the confidentiality of both the whistleblower and any third parties mentioned in the report. Only personnel explicitly authorized to handle such reports should have access to information that could reveal the reporting person's identity or the content of the report. Moreover, where identity disclosure is required or permitted under Union or national law, the Directive requires that the reporting person be informed before their identity is disclosed, unless it would jeopardize the related proceedings [Art.16(3)].

The Directive defines two channels[3] for processing reports: internal and external reporting. Internal reporting is managed directly by the organization where the breach occurred. It should be encouraged where the breach can be effectively addressed internally and there is no risk of retaliation [Art.7]. External reporting is overseen by competent authorities designated by Member States. A whistleblower may use this channel either after reporting internally or directly [Art.10]. At the same time, the Directive does not uniformly require Member States or entities to accept anonymous reports; rather, this is left to national law [Art.6(2)]. This distinction is relevant for system design: anonymity is not mandated in all deployments, while confidentiality of identity and related information is mandatory once a person is protected by the Directive.

Most organizations in both public and private sectors falling within the Directive's scope are required to implement internal reporting systems, subject to the threshold and exceptions in [Art.8]. In particular, entities in the private section with 50 to 249 workers may share resources for receiving and handling reports [Art.8(6)]. Additionally, member states are required to create external reporting channels [Art.11] and enforce protective measures to shield reporting persons from retaliation. For a detailed analysis of the Directive's articles and its implementation in Member States, we refer the reader to [36].

2.2 From the Directive to System and Security Requirements.

One of the key contributions of this paper is a systematic, compliance-driven analysis that use the Directive to guide the design process, directly deriving the system architecture from its requirements, rather than treating the Directive as informal motivation for *some, however reasonable,* design of a whistleblowing system and later arguing that it appears compliant, we faithfully treat it as a normative source of operational requirements and derive our system design bottom up from *procedures* which must be implemented by any system compliant with the Directive, *assets* that must protected by the system, and *threats* related to the assets that must be addressed.

Our process was structured as a "funnel" that progressively analyzed the Directive and mapped Directive-mandated obligations to derive the threat model, security goals, system procedures and protections, and eventually concrete system components and design. As a result, compliance with the Directive is enforced by construction rather than asserted post hoc, and the resulting design can be analyzed for later compliance and certification review.

We begin by identifying the *procedures* that a compliant system must support. These procedures collectively capture the operational workflow mandated by the Directive, including report submission, receipt and acknowledgment, validation, follow-up communication,

[1] Note that, in the overview of the Directive, we do not use the separation of Whistleblowing System into Enrollment System and Reporting System, as we do in our proposition.

[2] There are subtle issues associated with these periods; however, for brevity, we omit them here. For further details, see [Art.11(2d)].

[3] The Directive recognizes *public disclosure* as an alternative reporting path for whistleblowers. However, this path is not formally structured and is addressed primarily to grant legal protection to whistleblowers under specific circumstances. Therefore, our analysis focuses solely on internal and external reporting channels.

feedback delivery, escalation or transfer to external authorities, identity disclosure under strict conditions, and audit and record keeping. This analysis reveals that whistleblowing under the Directive is inherently stateful and interactive. Reporting is not a one-time action but a sequence of interdependent steps which explicit timing constraints. This is a crucial difference form systems that reduce whistleblowing to a single anonymous inbox or web form submission. The full set of procedures together with their legal basis in the Directive, and involved actors is summarized in Table 1 in the Appendix.

These procedures also align with the report workflow we rely on. As we explain in Section 3.1, our design distinguishes two systems: the Enrollment System, in which the whistleblower obtains the artifacts needed to interact with the reporting infrastructure without exposing their real identity at reporting time, and the Whistleblowing System, in which the report is processed. Within the Whistleblowing System, report processing is organized into four main phases: *Create*, *Handle*, *Archive*, and *Close*. The Create Phase covers submission, receipt, acknowledgment, and assignment to an operator; the Handle Phase covers follow-up, feedback, and possible transfer or identity disclosure; and the later phases cover retention, archival, and closure. The high-level separation between these two systems is reflected in Figures 1 and 2.

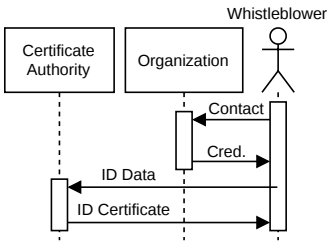

Figure 1: The Enrollment System **equips the whistleblower with an entitlement credential issued by an organization they have interacted with, enabling them to submit a report related with that organization.**

Again from the Directive, we derive the set of *assets* that must be protected for the system to operate securely. In line with standard risk analysis methodologies [32], we define assets as system entities, data objects, and states whose compromise would undermine the correctness and security of the whistleblowing process. These assets include, among others, whistleblower identity and entitlements, report content, report state, acknowledgment, follow up and transfer artifacts. A detailed list of assets is provided in Table 2 in the Appendix.

Each asset is associated with one or more security/correctness properties that must hold to assure compliance. For example, the Directive explicitly requires confidentiality of identity and report-related information, but it also implicitly requires correctness and timeliness of different stages of the process (such as issuing an acknowledgment). Importantly, different assets require different

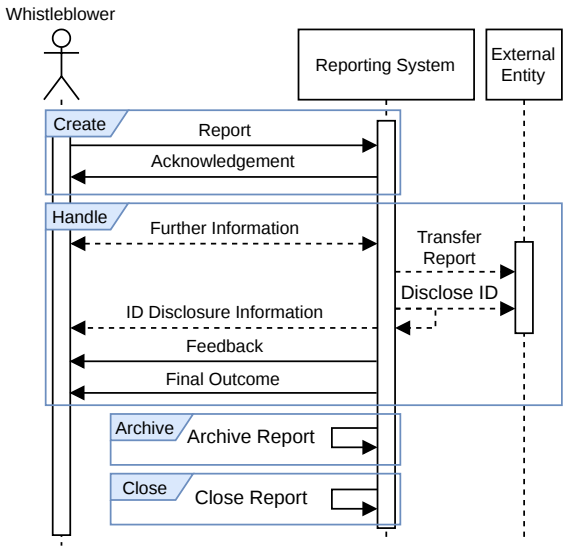

Figure 2: The Reporting System **processes reports submitted by whistleblowers through a workflow consisting of four main phases: Create, Handle, Archive, and Close.**

combinations of these properties, depending on their role in the workflow and the guarantees mandated by the Directive.

Building on the asset analysis, we identify threats as ways in which assets may be compromised, misused, made unavailable, either intentionally or accidentally, by various adversaries. This includes not only confidentiality and integrity failures, but also failures to acknowledge, follow up, document, or retain reports to the extent defined by the Directive. The full threat list and its mapping to affected assets are summarized in Table 3 in the Appendix.

Based on this analysis, we formally define procedures and requirements for a system that aims to implement operational procedures defined in the Directive in a way that properly protects the assets (see Section 4.4). In doing so, we intentionally distinguish between what is explicitly required by the Directive, what is a responsibility of a Member State, and what we introduce as additional technical safeguards. In Table 4 in the Appendix, we provide a mapping from procedures, assets and threats to specific system actions, while the overall system procedure flow is illustrated in the Figure 6 in the Appendix.

Finally, we map the identified threats to concrete *protection mechanisms*. At a high level, our design relies on two components. IMC is responsible for managing pseudonymous identities and identity disclose, while ALS provides auditable communication channel and auditable logging service. Together, they support a whistleblowing process in which confidentiality, accountability, and later compliance review can coexist. The mapping between threats and protection mechanisms is summarized in Table 4 in the Appendix.

2.3 Modifications to the Derived Requirements

The Directive leaves certain operational choices open or unspecified. In those cases, we make design choices that preserve the protective purpose of the Directive while remaining compatible with its text. Two such modifications are especially important for our design.

Acknowledgment Obligation. The Directive requires acknowledgment within seven days for internal reports [Art.9(1b)]. For external reports, the acknowledgment may be omitted if the reporting person explicitly requested otherwise or if acknowledging receipt would jeopardize the protection of the reporting person's identity [Art.11(2b)]. We interpret this not as a reason to remove acknowledgment from the system model, but as a reason to minimize what acknowledgment reveals and how it is received.

Concretely, our design separates *sending* acknowledgment from *collecting* it. The system always creates an acknowledgment tied to the submitted report and to the reporting person. Whether the reporting person later chooses to receive the acknowledgment is a separate question and, in our design, is an optional step. This design choice preserves the whistleblower's ability to demonstrate that the report was submitted *and* precisely when, but it does not force an active process of doing so that could potentially lead to disclosing their identity. At the same time, it prevents a particularly problematic Directive's security failure: withholding acknowledgment under the pretext of protecting the whistleblower, thereby depriving the whistleblower of evidence that the report was submitted at all. Accordingly, our design requires that:

- the existence of an acknowledgment be auditable;
- the acknowledgment reveals no unnecessary information beyond what is needed to confirm receipt of the report; and
- not receiving an acknowledgment not affect the fact that the report was submitted and acknowledged.

Minimizing Human Error Dependence. The Directive cannot be implemented by technical means alone. Some part of report handling necessarily requires human assessment of the reported matter and of the appropriate next steps. In our design, however, such judgment is intentionally minimized and confined to a specific role and stage of the workflow. The selected operator is the entity that handles the report on the merits: the operator evaluates the case, decides how it should progress, communicates with the whistleblower where needed, and prepares the outcome. By contrast, managing operators do not perform substantive case handling. Their role is limited to receiving the report, verifying its technical validity, sending acknowledgment, and assigning the report to an operator. The overall workflow is introduced later in Section 3.3, while the corresponding system actions are specified in Sections 4.4 and H. As a result, the threat model must account for operator error, bias, delay, and malicious behavior primarily at the operator level, rather than treating discretionary human judgment as pervasive throughout the whole workflow.

Our design therefore cannot and does not attempt to eliminate reliance on operator judgment. Instead, it ensures that the points at which such judgment is exercised are explicit and auditable. Concretely, the following functions rely on human judgment by the operator: conducting investigations; determining whether and how the report proceeds within the workflow (including priority); deciding on and formulating follow-up requests and feedback; initiating identity disclosure or report transfer; and preparing the final outcome prior to archiving and closure. ' What the system contributes is not replacement of these human functions, but enforcement of the surrounding evidentiary discipline: when the operator acted, which report state the action related to, what message or decision was issued, and whether timing and authorization constraints were respected. This is essential for later dispute resolution, auditing, and, where necessary, judicial review. It also clarifies the boundary of our claims: we secure and audit the process surrounding operator-level handling of the report, while treating the substantive merits of the underlying investigation itself as outside the scope of the protocol.

3 The Whistleblowing System

In this section we provide a high-level overview of the proposed Whistleblowing System, introducing its architecture, the division of responsibilities among participating entities, and the main phases of report processing.

3.1 High-level Overview

A person who may later act as whistleblower interacts with the Whistleblowing System, which consists of two separate components: the Enrollment System (see Figure 1) and the Reporting System (see Figure 2). At a high level, the ES enables future reporting, while the RS handles the reporting process itself. We separate these components because a directive-compliant solution must simultaneously establish who is entitled to whistleblow and protect the whistleblower's identity at the time of reporting. If reporting credentials needed for whistleblowing were issued only when a person decides to submit a report, then the act of obtaining them could be correlated with a recently submitted report.

Of course, the enrollment step itself is not anonymous: the organization learns which persons are entitled to report on matters concerning it. This is unavoidable if entitlement is to be verified against a real work-related relationship. The purpose of the separation between enrollment and reporting is therefore not to hide the set of potentially entitled whistleblowers, but to ensure that a later reporting event cannot be linked back to a particular enrollment interaction without undergoing through the controlled identity disclosure procedure (see security properties listed in Section 5.3).

In the system architecture considered in this section, the RS is shared across multiple organizations and provides a common infrastructure for the reporting process. It is operated in a consortium model by a set of entities acting as managing operators, drawn from different companies, public institutions, or non-profit organizations. Procedures executed by managing operators are implemented using (t, n)-threshold cryptographic primitives, requiring the cooperation of at least t out of n managing operators. Actions of all entities in the system are auditable, and managing operators, in particular, are restricted to well-defined procedural tasks that do not involve human judgment.

Other deployment variants are possible and are discussed in Section 6. However, the shared model is natural and appealing from security and deployability perspectives. It distributes trust across multiple entities rather than concentrating it within a single organization, reducing single point of failure, while still allowing

organization-specific handling of reports where human judgment is required. Moreover, since the Directive explicitly permits shared reporting infrastructures, this model lowers the barrier to adoption: even small organizations can enjoy compliance without bearing the cost and operational burden of deploying and maintaining it.

Enrollement System. The Enrollment System is organization-specific. It is used when a person begins a work-related relationship with an organization and prepares that person for potential future reporting. At a high level, this preparation is split into two independent steps. First, during work-enrollment with the organization, the person obtains a ticket that makes them entitled to whistleblow on that organization. Second, the person obtains a certificate for their real identity from a certification authority. Later, these two pieces of information are combined to derive a report pseudonym (using the IMC) for a specific report without requiring any new identifying interaction with the organization at the time of reporting. A person who has work-related relationships with multiple organizations therefore needs to enroll separately for each such organization using their dedicated ES.

Before any reporting credentials can be issued, each organization must set up its ES and establish the information that the RS later needs to process reports concerning that organization. This includes at least the identity of the organization, its corresponding public key, and the identities and public keys of the operators associated with the organization. As a result, in addition to provisioning future whistleblowers, the ES links each organization to the shared reporting infrastructure in a way that later enables organization-specific report submission and handling.

Reporting System. The Reporting System is only used once a person decides to submit a report concerning an organization for which they previously obtained reporting credentials for. Within the RS, we distinguish between managing operators and operators. The managing operators are responsible for receiving the submitted report, checking conditions for processing, verifying the entitlement-related conditions, issuing an acknowledgment, assigning an operator, and later archiving and closing the report. Their role is intentionally limited to actions with conditions that can be specified in advance and checked automatically, without human judgment. By contrast, an operator is responsible for handling the report on the merits. This includes deciding how the case should proceed once a valid report is assigned to them, conducting or initiating the investigation, communicating, if needed, with the whistleblower using a process that preserves confidentiality, deciding whether to initiate report transfer or identity disclosure, and preparing the final outcome. In this way, the system concentrates the human judgment in the hands of the operators and confines it to the Handle Phase, while keeping the rest of the workflow procedural. We describe the report workflow in detail in Section 3.3.

The same architecture can support both internal and external reporting paths. The difference lies primarily in the operator pool that handles the report on the merits. In the internal path, the assigned operator is tied to with which the report has a direct work-related relationship, whereas in the external path the assigned operator is drawn from an appropriate external pool, which can consist of operators from other organizations, regulatory authorities, or designated third parties.

Importantly, the shared deployment model can strengthen, rather than undermine or replace, internal reporting. The Directive explicitly encourages internal resolution where the reported matter can be effectively addressed within the organization. A shared and auditable infrastructure improves the credibility of internal channels to potential whistleblowers by reducing reliance on the unilateral control of a single organization over the procedural aspects of report handling. In particular, it makes the reporting process independently auditable, thereby increasing whistleblower confidence in the integrity of the process. At the same time, it provides organizations with stronger, verifiable evidence that prescribed procedures were followed, which is critical for compliance, oversight, and dispute resolution. As a result, the shared deployment model aligns the incentives of both whistleblowers and organizations.

3.2 Building Blocks

We now turn to the technical components and participating entities that implement this architecture. At the technical level, the system architecture is built around two main components: ALS and IMC.

The ALS supports auditable communication and logging throughout the reporting process. It enables the system to maintain a verifiable record of actions and message exchanges, and it supports on-demand auditing in the event of disputes, including non-attributable disputes such as claims of non-delivery. The ALS is used throughout the RS as the common communication and logging component. All communication between the whistleblower, the organization, managing operators, operators, and, where needed, external entities as well as all actions performed within the system are carried out and recorded through ALS (see Section 5.2).

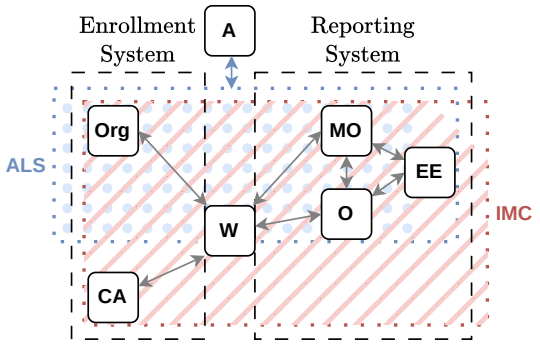

Figure 3: An overview of the main components and their relationships within the entities of Whistleblowing System**. Arrows indicate communication between entities; arrows enclosed within the blue dotted area represent communication that occurs via** ALS**. Entities located inside the red striped area make use of** IMC**. The arrow connecting the auditor (*A*) to** ALS **indicates that they have access to data recorded in** ALS**.**

The IMC supports confidential-yet-accountable reporting. It allows a whistleblower to report under a report pseudonym rather than under their real identity. At the same time, it ensures that each such pseudonym is bound to a real and entitled person, while

enabling controlled identity disclosure and, where permitted, controlled linking of related reports. The IMC is used in the ES and RS in the following ways. In the ES, it is used to issue the material later needed to derive report pseudonyms for a given organization. In the RS, it is used to derive a report pseudonym for each report, to authenticate report-related messages under that pseudonym, and, where justified, to support later identity disclosure or linking procedures, (see Section 5.3).

Besides whistleblowers, managing operators, and operators, the WS also involves several additional entities. In particular, it includes an *auditor* who may investigate the handling of reports; *external entities* which may receive transferred reports or disclosed identities; and a *certification authority* which is used prior to reporting to bind a real-world identity to the reporting pseudonym.

A central design principle of the WS is knowledge minimization. Each entity is given only the information and permissions needed for its part of the workflow. Managing operators are responsible for procedural handling while seeing only limited report information needed to validate and route a report. Operators gain access to report contents only because handling the report on the merits requires human judgment. Auditing is also designed so that routine review of procedural compliance can be separated from deeper, case-specific review of report data.

3.3 Report Workflow

The report is processed in the Reporting System in four main phases: *Create*, *Handle*, *Archive*, and *Close* (recall Figure 2), consistent with the procedures prescribed by the Directive (as outlined in Table 1) and other applicable regulations (e.g., GDPR [43]).

In the Create Phase, the whistleblower creates and submits a report to the Reporting System. To create a report, the whistleblower first chooses a report tag containing at least the category of the report and may include other non-confidential information needed for routing the report to a proper operator. This tag is then used together with the previously obtained ticket to derive a report pseudonym that is attached to the report throughout its processing. This design ensures that each distinct report is associated with its own pseudonym. As a result, reports submitted by the same whistleblower are not linkable by default. At the same time, the system supports controlled linking of related reports, as well as controlled identity disclosure, under specified policies and legal conditions. The managing operators then determine whether to (i) accept the report, issue an acknowledgment to the whistleblower, and assign an operator to the report for handling, or (ii) reject the report and communicate this outcome to the whistleblower. Their decision is based on only limited report fields sufficient for strictly procedural validation and is independent of the report's substantive merits. In particular, managing operators do not gain access to the report contents and do not learn the whistleblower's identity. At this stage, they also verify that the report and its associated pseudonym were generated correctly and that the report is valid.

In the Handle Phase[4] the assigned operator evaluates the report on its merits and provides feedback to the whistleblower on the outcome. If needed, the operator may request additional information or escalate the report by transferring it to another operator or entity, and when necessary, disclosing the whistleblower's identity.

During this phase, the assigned operator has full access to the report contents. However, access to the report contents does not by itself reveal the whistleblower's identity, which remains protected and bound to the report pseudonym unless disclosure is later triggered under the defined conditions. As in the case of managing operators, all operator actions within the Reporting System, particularly those involving follow up communication, report transfer or identity disclosure, are logged and auditable. At the same time, automated auditing is inherently limited to *process*, not *merits*. The system cannot evaluate the correctness of human judgment. For example, it cannot determine whether an operator should have escalated a report, whether a request for additional information was warranted, or whether identity disclosure was justified.

To address this concern, we define two complementary audit levels: *process-level audit* (level-1) and *individual-report audit* (level-2). Level-1 enables verification of procedural handling of the report without access to report contents, while level-2 permits a privileged auditor to access the report contents and assess the merit handling through independent review. Reliance on human judgment is unavoidable in any real-world system, with both its benefits and risks. Our design isolates judgment to a well-defined role and supports it with auditing capabilities to provide meaningful oversight and promote honest behavior, while preserving access and discretion necessary for report handling.

In the Archive Phase, the report is returned to the managing operators once processing has been completed or the report has been transferred to an external entity. At this point, the report becomes read-only: no further modifications are permitted, while the full record of actions performed on the report remains available for audit (at both level-1 and level-2).

In the Close Phase, the managing operators formally close the report. The report content is deleted, as required by GDPR [43, 56], while the log of actions performed on the report remains accessible. As a result, level-2 auditing is no longer possible, but level-1 auditing remains available.

An in-depth review of the report workflow is given in Figure 6. The detailed security goals and formal system actions corresponding to these phases are introduced in the sections that follow.

4 Model, System and Design Goals

In this section, we formalize the operational model of the Enrollment System and Reporting System, as well as the goals that the resulting Whistleblowing System must satisfy. We proceed in three steps. First, we define the participating entities, the structure of reports, and the time bounds used throughout the workflow. Second, we define functional goals that specify the required behavior of the system during report processing and the guarantees it must provide. Finally, we define design goals that constrain the system design.

4.1 Entities

Both systems interact with, and are operated by, a set of *entities*. Each entity can be viewed as a uniquely identifiable machine that maintains a private state. We distinguish the following entities.

[4]The Handle Phase corresponds to what the Directive broadly treats as *follow-up*, however we use the term *handle* to distinguish it from optional follow-up requests for additional information from the whistleblower.

- A time-varying set[5] of whistleblowers **W**. Each whistleblower $W \in \mathbf{W}$ interacts with the ES and can submit a report to RS.
- A fixed set of managing operators **MO**. Each managing operator $MO \in \mathbf{MO}$ receives and manages reports; however, managing operators do not assess the merits of the report and have access to limited parts of a report.
- A fixed set of operators **O**. Each operator $O \in \mathbf{O}$ manages the Handle Phase of the report processing in the RS.[6]
- A ixed set of external entities **EE**. Each external entity $EE \in \mathbf{EE}$ is a recognized entity entitled to communicate with with managing operators and operators (e.g., law enforcement agency in a given Member State or another higher-competence WS, etc.).
- An auditor A is an entity that can audit the whistleblowing process. Without loss of generality, we assume a single auditor; however, the functionality described in this work can be readily extended to accommodate multiple auditors.

4.2 Reports

Reports processed in the RS have the form of structured tuples; see Definition 1.

Definition 1 (Report). *A report R is a tuple* $(\mathsf{Org}, W, S, \mathsf{Cat}, \langle(E_i, \mathsf{Action}_i, \mathsf{Time}_i, \mathsf{Data}_i)_i\rangle)$, *where:*

- Org *is a string uniquely describing an organization.*
- W *is a string that allows to uniquely identify the whistleblower.*[7]
- S *is a string describing the identity of the subject of the report.*[8]
- Cat *is the category of the report, which is an element of the public predefined set* $\mathsf{Categories}$.[9]
- $\langle(E_i, \mathsf{Action}_i, \mathsf{Time}_i, \mathsf{Data}_i)_i\rangle$ *is a sequence of action tuples, where each tuple contains:*
  - E_i *is a link that allows to uniquely identify entity performing the action.*
  - Action_i *is the type of the performed action, drawn from the predefined set* $\boldsymbol{A} = \boldsymbol{A}_W \cup \boldsymbol{A}_O \cup \boldsymbol{A}_{MO} \cup \boldsymbol{A}_{EE}$, *which includes actions initiable by a whistleblower, operator, managing operator, or external entity, respectively.*[10]
  - Time_i *is the time at which the action occurred,*
  - Data_i *is the data associated with the action.*

Note that at this stage we do not specify privacy guarantees for the report's fields; these requirements are defined later in Section 4.4.

[5]For example, the set may increase due to newly employed persons and decrease over time due to various reasons.

[6]Operator is a *receiver of a report* in the sense of the Directive.

[7]This field contains cryptographic material that uniquely identifies a whistleblower while preserving confidentiality; the identity can be revealed only through the controlled identity disclosure procedure (see Section 4.4.3).

[8]It does not necessarily need to be any specific identity. It may be an identifier of an entity, "someone" or "a group of people". The Directive refers to the term "person concerned", a natural or legal person who is referred to in the report or public disclosure as a person to whom the breach is attributed or with whom that person is associated.

[9]Which includes categories like sanctions violations, financial fraud, etc. The set of allowed categories should comply with [Art.2].

[10]The action sets $\mathbf{A}_W, \mathbf{A}_{MO}, \mathbf{A}_O, \mathbf{A}_{EE}$ can be induced from the $\mathsf{add}(\mathsf{Entity}, \mathsf{Action})$ pairs in the Figure 6 or Table 4.

Definition 2 (Valid/Invalid Report). *A submitted report* $R = (\mathsf{Org}, W, S, \mathsf{Cat}, \langle(E_0, \mathsf{Action}_0, \mathsf{Time}_0, \mathsf{Data}_0)\rangle)$ *is considered* valid *if and only if the following conditions are satisfied:*

- W *is entitled to report on the organization* Org,
- *the action tuple is correctly formatted, that is:*
  - $E_0 = W$,
  - $\mathsf{Action}_0 = \mathsf{sendRep}$,
  - Time_0 *is the current time, and*
  - Data_0 *is non-empty.*

A submitted report R not satisfying any of the above conditions is considered invalid.

4.3 Time Bounds

We model the timing constraints that govern the report workflow using explicit time bounds. The Directive explicitly defines two such obligatory bounds: one for acknowledging receipt of a report, denoted by Δ^{ack} (ref. [Art.9(1b)], [Art.11(2b)]), and another for providing the outcome of the processing to the whistleblower, denoted by Δ^{info} (ref. [Art.9(1f)], [Art.11(2d)], [Art.11(5)]).

In addition, the system requires two further timing parameters to capture retention and workflow-management constraints. First, motivated by [Art.18(1)] together with [Art.5(1e)] of GDPR [43], which requires that personal data be kept no longer than necessary, we introduce a deadline Δ^{cls} after which report data are removed following a period of inactivity. Second, we introduce a deadline Δ^{feed} after which a request for further information is treated as unanswered; this parameter is needed to model report progression in the state machine of Figure 6. The concrete values of Δ^{cls} and Δ^{feed} are deployment parameters that should be determined by the applicable Member-State implementation.

Finally, we assume two timing bounds used by the protocol and the analysis: δ^{lag}, which bounds network communication delays, and δ^{sync}, which bounds the time required for all honest managing operators to synchronize.

4.4 System Goals

We now define the system goals that govern report processing in the Whistleblowing System. These goals specify the expected behavior of the workflow and the security guarantees that must hold during processing.

4.4.1 Report Processing. We start by formalizing the goals that define the report workflow, an ordered sequence of mandatory actions executed during report processing (goal G1), as well as goals that apply to each action within the WS (goals G2–G8).

G1: Process correctness. Any report processed in the RS is assigned a unique report identifier and is subject to a sequence of actions depicted in Figure 6. A report in a given state can only be subject to actions corresponding to outgoing arrows from that state. Each completed action modifies the state of the report, and the last action in the sequence uniquely determines the current state. Report processing starts with the sendRep action.

G2: Processing recordability. All actions performed on a report, including executing entity, action types, and execution time (i.e., the first three fields of the action tuples from Definition 1) are

recorded in an append-only data log and are available both during report processing and after the report is concluded (see Section 4.3).

G3: Report data time-bounded persistence. All data associated with the report (e.g., report content and investigation results; recall the fields of the action tuple from Definition 1) are recorded in an append-only log and are available during report processing.

G4: Report confidentiality. The report fields and the data associated with actions performed on the report as well as any information related to the identities of entities executing the actions are protected from access of entities that are not explicitly authorized to access that information. The authorization and access rules that determine under which conditions the information included in the report fields can be accessed are specified by the goals below.

G5: Process compliance auditability. The executing entity, action type and execution time of each action associated with a report can be accessed by an auditor. In particular, given a unique report identifier, the auditor can access the log of state transitions and their execution times for that report. The auditor must be able to validate that the sequence of actions applied to the report, as well as their execution times, is compliant with the requirements set out by the Directive (see Section 4.3).

G6: Individual report auditability. Data associated with a specific action can be accessed by an auditor given a unique report identifier and an access token for that action.

G7: Artifacts authenticated and bound to reports. All action artifacts (such as an acknowledgment, a conclusive statement of the report, etc.) produced during report processing are authenticated and bound to the report they concern. It is also infeasible to forge an artifact on behalf of any entity.

G8: Timely response. The system guarantees that the procedures can be carried out in a way to meet the time bounds listed in the Section 4.3.

4.4.2 Actions Involving Message Exchange. All actions involving communication between entities,[11] including action logging, are additionally subject to the following goals:

G9: Guaranteed delivery. All sent messages can be received by their designated receivers while preserving their integrity.[12] **G10: Designated receiver.** Only a designated receiver is able to receive and read the message.

4.4.3 Individual Action Requirements. Below we define additional security goals that are associated with individual actions.

Create, send & receive report

G11: Submittability. Every report sent to the RS is recorded and made available to managing operators.

G12: Unframeability. It is infeasible for any entity other than the legitimate identity owner to submit a report under that identity.

G13: Process valid reports. Any whistleblower who has obtained an entitlement credential issued by the ES for a given work-related organization is entitled to submit reports concerning that organization. Only valid reports (see Definition 2) are processed by the RS.

Send & Receive Acknowledgment

G14: Valid reports acknowledgment. For every valid report, the system sends an acknowledgment to the whistleblower. No acknowledgment is sent for invalid reports.

G15: Acknowledgment unforgeability. It is infeasible to forge an acknowledgment. In particular, the existence of an acknowledgment implies a submission of a valid report.

Reject Report

G16: Reject invalid reports. A report is rejected if and only if it is invalid. A reason for rejection must be recorded.

Send & Receive Final Outcome

G17: Contains valid information. The managing operators include in the final outcome message either the reason for report rejection (see G16) or a conclusive statement submitted by an operator (see G20), depending on the action that preceded the Send Final Outcome action.

Assign Operator

G18: Assign operator correctness. A single operator $O_j \in \mathbf{O}$ is assigned to a report in accordance with a deterministic operator-assignment function f^{assignOp}, which depends on the concerned organization, report category, the set of operators, their current availability, and the mapping of reports already assigned to operators.[13]

G19: Data access. The assigned operator can access details of the report, including the subject of the report, as well as the data associated with actions performed on the report.

Conclude Report

G20: Contains conclusive statement. The report conclusion submitted by the operator must include a final conclusive statement regarding the result of the report processing.

Decide & Choose Competence

G21: Decided action is logged. In the Handle Phase, the assigned operator is obliged to log each decision along with a justification for their choice.

When making this decision, the operator should consider at least the following factors:

Report not repeated. On the first read of any new information added by the whistleblower to the report, the operator must check that the report is not a duplicate of another report describing the same case and submitted by the same whistleblower. The requirement may be restricted to information contained in reports associated with the operator assigned to the selected report.

Valid category. On the first read of any new information added by the whistleblower to the report, the operator must check that the category assigned to the report is relevant to the reported information on breaches.

Operator has competence. The operator must decide whether they have the competence to continue processing the report or whether it should be transferred to a higher-competence entity.

Further Information Request, Send & Receive Further Information

G22: Further information request contains justification. The further information request contains a justification statement for the request.

[11] For each message, Figure 6 specifies its delivery deadline and receiver.

[12] In the Directive, the "integrity" of the messages is described in terms of "integrity" and "completeness". We use the single term to describe both.

[13] For instance, workload optimization could be improved by assigning reports concerning the same category to a single operator. The details of the f^{assignOp} logic are left to the deploying entity, as they primarily depend on the deployment environment.

G23: Further information is unframeable. Only the author of the report may respond to a further information request associated with that report.

Send & Receive Feedback

G24: Contains feedback statement. The feedback message contains a feedback statement.

Request Identity Disclosure, Validate Identity Disclosure Request, Send & Receive Identity Disclosure

G25: Identity protection. All submitted reports must be attributable to a uniquely identifiable whistleblower. Disclosure of a whistleblower's identity may occur only with quorum-based approval from the managing operators. Such disclosure can either reveal the whistleblower's full identity or provide *linkability information*, enabling all reports associated with the same entitlement credential to be linked.

G26: Identity disclosure request. Prior to identity disclosure, an operator must submit an identity disclosure request that uniquely encodes an explicit set of intended receivers and a justification asserting that the identity or linkability information associated with a given report may be disclosed to all enumerated receivers.

G27: Validation of identity disclosure requests. Only identity disclosure requests that that satisfy a well-specified function $f^{\mathsf{idDiscReq}}$ authorize the managing operators of the report to continue processing the request. The function $f^{\mathsf{idDiscReq}}$ takes as input a list of receivers and a justification proof, and outputs 1 if and only if the justification proof is sufficient to reveal the identity or linkability information to all listed receivers.

Request (Partially) Transfer Report, Transfer & Receive Transferred Report

G28: Report (partial) transfer request. Prior to report (partial) transfer, an operator must submit a report transfer request that uniquely encodes an explicit set of intended receivers and a justification asserting that the (partial) report may be disclosed to all enumerated receivers.

G29: Validation of report-(partial)-transfer requests. Only report (partial) transfer requests which satisfy a well specified function f^{transReq} authorize the operator of the report to further transfer the report. The function f^{transReq} takes as input a list of receivers of the request and a justification proof and outputs 1 if and only if the justification is sufficient to reveal the declared data to all of the listed receivers.

Write to Report

G30: Authorized access to write. Upon assignment of an operator, the designated operator is granted write permissions on that report.

4.5 Design Goals

Finally, we define a small set of design goals that constrain the system design beyond the action-specific goals above and support Directive-compliant deployment. In particular, separation of duties aligns with the Directive's requirement that only specifically designated or otherwise authorized personnel may access and handle reports ([Art.9(1a)], [Art.11(1)], [Art.16(1)]), while denial-of-service resilience and evolutionary design are derived engineering goals motivated by the need to keep reporting channels reliably available and able to process large inflow of reports ([Art.11(5)]) as well as by ensuring that the design is suitable for long-term compliance-oriented deployment ([Art.14())].

DG1: Denial of service protection. The system preserves report-submission availability under high load and under denial-of-system attempts.

DG2: Evolutionary design. The system is built, whenever possible, from standardized cryptographic primitives and modular components, thereby supporting real-world deployment, later compliance assessment or certification.

DG3: Separation of duties. The system enforces role separation such that report processing is distributed across independent entities, each receiving minimal information and capabilities required for its assigned action.

5 Building Blocks

This section introduces the technical building blocks of the proposed WS. We describe the two main components that realize the goals defined in Section 4: the Auditable Log Service, which provides auditable communication and logging, and the Identity Management Component, which supports entitlement verification and report pseudonyms. We define an interface and security properties, and provide a concrete instantiation and a security analysis of each component in Section F and Section G, respectively.

5.1 Assumptions

We make the following assumptions about the entities and the environment of the entities.

PKI for All Entities. All entities have access to a public key infrastructure (PKI). In particular, their identities are verifiable by a trusted certificate authority, which issues certificates under a certificate scheme Crt. Such certificates bind an identity string τ to arbitrary data supplied by an entity if and only if the entity is duly authenticated as representing the identity associated with τ.

The Clock. All entities interacting with the system have access to local synchronized clocks. At any point, the entities can call their local clock for a current time.

Communication Between Entities. We assume access to the following idealized communication channels: an *anonymous communication channel* that allows an entity to send a message to another entity without revealing sender-identifying communication metadata to external observers; and a *broadcast channel* that allows an entity to send a message to a specified subset of entities. In practice, both channels can be instantiated using standard techniques (e.g., Tor [15] and reliable broadcast protocols [7, 8]).

Helper Entity. In addition to the default set of entities that operate and interact with WS, we introduce a class of helper entities, referred to as ALS provider, responsible for operating the ALS. We assume that ALS provider operates in an honest-but-curious manner (see Section F.3 for more information).

The Adversary. We model the adversary as a probabilistic polynomial-time (PPT) algorithm that may corrupt a subset of system entities and resources. Upon corrupting an entity, the adversary obtains its full view, including: (i) the messages sent and received by that entity, (ii) its internal state, and (iii) the public information

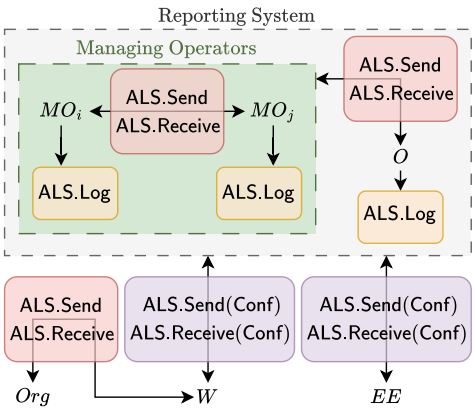

Figure 4: Representative interaction patterns supported by ALS**. Managing operators and operators exchange information using** ALS.Send **and** ALS.Receive**, and record actions via** ALS.Log**. The whistleblower follows the same interaction pattern when communicating with the organization representative. Communication between the** RS **and the whistleblower or other external entities also adheres to this pattern; however, stronger guarantees of verifiable receipt can be achieved by using** ALS.SendConf **and** ALS.ReceiveConf**; see Section H.1.3.**

accessible to all entities, and may impersonate that entity thereafter. By contrast, the helper entity ALS provider is not modeled as adversarially controlled, but as honest-but-curious, as stated above.

Bound on malicious and faulty managing operators. Let n denote the total number of managing operators, and let f be an upper bound on the number of managing operators that may be malicious or faulty. Malicious operators may arbitrarily deviate from the protocol, while faulty operators may be offline or unresponsive. We assume that $n \geq 2f + 1$.

Synchronization for honest managing operators. We assume that all honest managing operators can communicate with one another and that any synchronization step among them completes within a bounded time interval δ^{sync}.

5.2 Auditable Log Service

Auditable Log Service (ALS) provides two core functionalities: auditable communication between entities and auditable logging of actions that do not require communication. Operationally, ALS supports three modes of use, the main interaction patterns used in the RS are illustrated in Figure 4.

(1) Logging performed by a single entity within the RS, such as a managing operator or an operator, does not require communication with another entity. This mode is handled by the Log function.[14]
(2) Internal communication between the managing operators and the operator requires sending a message to a designated entity and later receiving it. This mode is handled by Send and Receive functions.
(3) Communication involving the whistleblower and external entities is handled either by Send and Receive or, when verifiable receipt is additionally required, by SendConf and ReceiveConf.

To support auditability, ALS equips the auditor with a set of audit procedures. These procedures allow the auditor to resolve potential disputes that may arise between the entities, for example regarding the time or content of messages sent into or out of the WS. In particular, both the whistleblower and the system entities can rely on ALS to provide an auditor with evidence about message transmission, receival, and content.

More concretely, ALS provides the following capabilities:

- Every message sent through ALS is timestamped, and messages received through ReceiveConf also obtain a timestamp for receival.
- The sender of a message sent via SendConf can verify whether the receiver has received the message.
- The auditor can determine whether a message was sent, and, for messages exchanged via (SendConf, ReceiveConf), whether it was received.
- The auditor can inspect the first part of a message (called the *level-1 message*), without interacting with either sender or receiver.
- The auditor can inspect the second part of a message (called the *level-2 message*), but only in cooperation with the sender or the receiver.

Conceptually splitting a message into two parts allows the system to place basic procedural information, such as as the type of the action applied to the report, in the level-1 message, while keeping more detailed report-related data in the level-2 message (see Section 3.3 and definition 1).

A major advantage of this design is that level-1 auditing can operate as a form of process-level auditing that does not require human inspection. The auditor can passively monitor level-1 messages to detect missed deadlines or non-responsiveness of the system. When a deeper investigation is needed, the auditor can, upon request and with cooperation from the relevant entity, inspect level-2 messages and use them for dispute resolution or further escalation.

ALS can be deployed either as a single instance shared across organizations or as an organization-specific service, which makes it suitable for both internal and external reporting channels. One natural way to instantiate ALS is as a group mailbox built on top of a bulletin board accessed through an anonymous communication channel. Such an instantiation maintains the following principles:

- Anyone can *anonymously* post a message consisting of a level-1 and level-2 part addressed to a chosen receiver.
- Anyone can *anonymously* inspect the bulletin board, enabling intended receivers to monitor it for incoming messages, while only the intended receiver and the auditor can determine who the message is addressed to.[15]

[14]For convenience, we occasionally use communication-oriented terminology even for Log, since logging can be viewed as sending a message to the auditor.

[15]This property is particularly important for preserving anonymity in feedback-style communication channels.

- The auditor can access the level-1 message passively and, upon request from the sender or receiver, can also access the level-2 message.
- In the case of ReceiveConf, the receiver must authenticate to ALS provider before receiving the message, which allows the receival event to be recorded.

The applicability of ALS extends beyond whistleblowing, so the abstraction is formulated independently of the specific whistleblowing report workflow. The SendConf functionality of ALS is also conceptually related to a Registered Delivery Service[16] as defined in Regulation (EU) 2024/1183, known as the eIDAS 2.0 Regulation [55].

5.3 Identity Management Component

The Identity Management Component supports privacy-preserving report submission by enabling the whistleblower to derive a report pseudonym repnym that can be attached to report-related messages. Its purpose is to ensure that report messages can be authenticated and tied to an entitled reporting person without making the reporting event itself identifying. More precisely, the report pseudonym repnym is designed to satisfy the following properties:

(1) The whistleblower is able to create only a single report pseudonym for each distinct report.
(2) Each pseudonym is bound to a *real* identity.
(3) The pseudonym does not allow the entities operating the Whistleblowing System to link a given pseudonym to a particular time, place or the identity of the whistleblower that obtained the pseudonym, with an exception that the *real* identity of the whistleblower can be conditionally extracted from the pseudonym.
(4) All report pseudonyms generated by a single whistleblower are not linkable, with an exception that the reports from a single whistleblower can be conditionally linked to each other by an operator of the report.

The uniqueness property (Item 1) is achieved by introducing unique report identifiers, called *report tags*, and ensuring that the derived pseudonym depends on the chosen tag. The identity-binding (Item 2) and unlinkability (Item 3, Item 4) properties are achieved by combining organization-issued entitlement material with an independently issued identity certificate. The conditional linkability (Item 3, Item 4) properties are ensured by dividing into a set of shares the knowledge about the identity of the whistleblower or the knowledge that allows to link pseudonyms created by a single whistleblower.

Operationally, obtaining a report pseudonym is divided into three stages. First, during enrollment with the organization, the whistleblower obtains a ticket that later enables pseudonym generation for that organization. Second, before reporting, the whistleblower obtains an identity certificate from an independent certificate authority. Third, immediately before report submission, the whistleblower combines the ticket and identity certificate to derive a report pseudonym together with a certificate for that pseudonym. This separation allows the organization to establish entitlement in advance while reducing the identifying linkage between enrollment and the later reporting event. In addition, the report pseudonym certificate can be used, under well-specified conditions, to support controlled identity disclosure or controlled linkability of related reports. Figure 8 summarizes how the IMC supports the process described above.

[16] A key difference is that ALS additionally aims to preserve sender anonymity.

6 Deployment Considerations

The previous sections describe the Whistleblowing System at three levels. Section 3 gives the high-level architecture, separating the Enrollment System from the Reporting System. Section 4 describes details of requirements for the Whistleblowing System. Section 5 then introduces the two technical building blocks on which this architecture relies, ALS and IMC. Section H puts these elements together and gives the detailed system design specifications.

We now turn from system specification to deployment and discuss the main deployment choices and the practical issues they raise. The goal is not to prescribe a single institutional model, but to make explicit which choices affect deployability, trust, availability, and compliance with the Directive. In practice, successful deployment depends on how the proposed system can be embedded into an existing institutional and technical infrastructure.

6.1 Deployment Models

We distinguish three default deployment models, corresponding to different configurations of internal and external reporting channels. Each model assumes the existence of managing operators, which run procedural parts of the WS, and operators, which are trained personnel responsible for handling reports on the merits during the Handle Phase. Irrespective of the deployment model or chosen reporting path, communication and logging are conducted through the ALS, which may be state-hosted or otherwise operated under a governance arrangement that permits independent audit (see Figure 5).

State deployment. The *state deployment* model is maintained by a Member State and supports external reporting. In this configuration, the state-managed Reporting System is overseen by managing operators selected by the state, who, in turn, assign state-authorized operators, regulators, or other designated external entities to individual reports. This model is natural for external reporting under [Art.11] and for cases in which an internal channel lacks competence, independence, or authority to continue processing a report.

Standalone deployment. In a standalone deployment, an organization operates its own ES and RS, including its own managing operators and operators. This mirrors the state model inside a single organization. It may be appropriate for large organizations with sufficient technical, legal, and compliance capacity. Its main drawback is institutional concentration of control: the same organization that may be the subject of a report also controls much of the reporting infrastructure. While the system provides auditing and role separation, the perceived trustworthiness might be weaker unless the organization engages independent auditors, external managing operators, or comparable oversight mechanisms.

Shared deployment. In a shared deployment, multiple organizations rely on a common RS, potentially operated by a consortium, public body, non-profit organization, or regulated third-party provider. The ES remains organization-specific, because entitlement

must be tied to the reporting person's work-related relationship with a particular organization. The Reporting System infrastructure, however, is shared. Operators assigned to reports may come from the reporting organization, from a shared pool, or from an external authority depending on the reporting path, the report category, and the applicable national implementation of the Directive.

The shared model is especially attractive from both trust, security and deployability perspectives. It distributes procedural control across multiple entities, reduces the burden on smaller organizations, and aligns with the Directive's recognition that private-sector entities with 50 to 249 workers may share resources for receiving and investigating reports [Art.8(6)]. It also improves the credibility of internal reporting: an organization can still handle reports internally where appropriate, while the report workflow is no longer under the unilateral control of that organization.

In all three models, reports must be transferable when the assigned operator lacks competence or when a higher-authority for handling is legally required. In our architecture, this corresponds to transferring the report to a higher-authority RS, typically the state-level instance. This process is governed by the report-transfer procedure described in Section H.1.4.

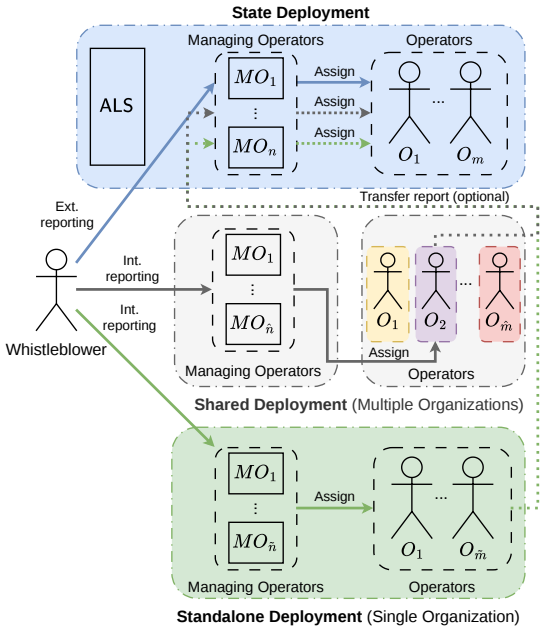

Figure 5: Proposed deployment architecture. Blue, gray, and green colors denote state-operated, shared, and standalone deployments, respectively. Solid arrows represent reporting flows, while dotted arrows indicate optional interactions. The figure depicts a state-operated shared ALS **instance.**

6.2 Organizational Onboarding and Enrollment

Each organization that participates in the WS must first establish its ES and register the information needed by the selected RS. This includes the organization's identity, public keys, authorized organizational representatives, applicable report categories, operator pool, assignment policy, and transfer rules. These choices instantiate deployment-specific functions introduced earlier, such as the operator-assignment function f^{assignOp}, the identity disclosure function $f^{\text{idDiscReq}}$, and the report transfer function f^{transReq}.

In our implementation, we assume, for simplicity, that enrollment tickets in the ES are issued by a single *organization representative* holding a protected key $\text{sk}^{\text{sys}}_{Org}$. In practice, this assumption is too narrow for real organizations and we expect $\text{sk}^{\text{sys}}_{Org}$ to be managed as any other secret key, for example, maintained in a hardware module and protected by threshold authorization. Each ticket issuance event should itself be logged, policy-bound, and auditable within the organization.

Onboarding and offboarding are also deployment-sensitive. The deployed system should support organization-specific entitlement policies, allow tickets to be refreshed or to expire, enable revocation of compromised enrollment keys, and retain sufficient evidence that a ticket was valid at the time it was issued.

6.3 Identity and Certificate Infrastructure

The Identity Management Component relies on an independent certification authority that binds a real-world identity to a public key. In practice, this assumption can be fulfilled using existing public infrastructures, including qualified certificate ecosystems in the the European Union. Such certificates bind a public key to an authenticated identity and protect the corresponding private key on a trusted device, such as a smart card or secure wallet.

6.4 Practical Realization of the ALS Provider

The Auditable Log Service is the main mechanism that turns the reporting workflow into an auditable process. In the current implementation of the ALS (see Section F.3), the ALS provider is an honest-but-curious entity. In practice, ALS provider can be instantiated in several ways, and the right choice depends on the deployment model.

One option is a state-hosted or consortium-hosted append-only bulletin board with authenticated timestamps, redundancy, and independent audit access. Another option is a permissioned ledger maintained by a set of public authorities, regulated providers, or consortium members. A public ledger that supports non-interactive zero-knowledge NIZK proofs can provide integrity, transparency, and public verifiability for append-only records, but it is not by itself sufficient for the full ALS interface. In particular, SendConf requires ALS provider to maintain private state for receival confirmation, and both level-1 and level-2 auditing require control over what is revealed to whom. These requirements point toward a permissioned or hybrid design. The deployment requirements also connect naturally to the eIDAS 2.0 ecosystem [55] as qualified trust services such as electronic ledgers and timestamps are plausible instantiation blocks for ALS.

6.5 Performance, Practical Considerations, and Justification of System Assumptions

The objective of this work is not to optimize the efficiency of the proposed system; we leave performance-oriented refinements to future work. Nevertheless, the implementation presented here is compatible with realistic deployments and helps justify the assumptions in Section 5.1.

The system is intended for designated reporting channels operated by organizations, Member States, or regulated service providers, not for an open adversarial environment with no institutional trust anchors. It is therefore reasonable to rely on synchronized clocks, bounded communication delays, high-availability infrastructure, and a threshold assumption in which only a limited fraction of managing operators are malicious or unavailable. The set of managing operators can be fairly small, as long as the threshold parameters tolerate the expected number of faulty or unavailable members. Similarly, the ALS provider may be instantiated using a permissioned ledger run by a limited number of entities.

Lastly, our design relies on standard cryptographic primitives with readily implementable in efficient constructions, including signatures, commitments, encryption, secret sharing, and, in particular, NIZK proofs [28, 48].

7 Related Work

Whistleblowing has been studied extensively from legal, organizational, and sociological perspectives, highlighting its impact on individuals and organizations alike. Prior work examined whistleblower well-being [40], career risks and post-disclosure consequences [19], and the persistent distrust that often discourages reporting even where legal protections exist [12, 38, 57]. Other works highlight cultural, organizational, legal and misuse-related barriers (such as malicious reporting or strategic disclosure of sensitive information) that further complicate effective reporting mechanisms [35, 51]. In parallel, legal scholarship has examined the structure and implications of the Directive [21], its interaction with national legal frameworks [34], and the challenges of achieving uniform implementation across Member States [47, 50]. Collectively, this line of work shows that whistleblowing cannot be reduced to a confidential or anonymous submission channel. Rather, it constitutes a socio-technical process shaped by regulatory deadlines and procedural obligations, organizational incentives, and perceived risks of retaliation. These factors influence whether the reporting process is perceived as trustworthy by potential reporters, and consequently, whether it can be meaningfully adopted and used in practice.

Bridging the gap between legal requirements and technical system design, a separate line of work provides implementation and governance guidelines for whistleblowing systems. Standards such as ISO 37002 [29] and best-practice frameworks such as Transparency International's guidelines [53] conceptualize whistleblowing as a full workflow process, similarly to the Directive. While highly relevant, this body of work remains largely non-technical and does not specify concrete mechanisms that support the process of whistleblowing.

In contrast, work on technical foundations of whistleblowing systems remains limited and fragmented. Much of the existing works adopts a narrow abstraction, modeling whistleblowing primarily as an anonymous report submission channel [1, 45]. This approach reduces whistleblowing to a one time act of anonymous disclosure, not addressing the entire report workflow mandated by the Directive, including acknowledgment, follow up, or identity disclosure.

Other works address isolated technical components. For example, [49] focuses on report text sanitization to mitigate re-identification risks, an important but orthogonal aspect that complements our approach. Similarly, [36] presents a security-by-design analysis of the Directive and surveys relevant privacy-enhancing technologies, while [5] examines anonymization and pseudonymization techniques and their inherent limitations in the context of whistleblowing. Taken together, these works underscore a key gap: the absence of mature, integrated, and deployable systems that address the Directive's requirements, both from a security and compliance perspective.

Related efforts in secure communication systems include systems developed for source protection in journalism (e.g., secure dropboxes such as SecureDrop [24] and anonymous contact channels, as well as more advanced designs like CoverDrop [2, 27], enable secure communication under strong adversarial models, but typically do not incorporate procedural guarantees or mandated timelines.

A natural direction is to consider existing anonymous communication and authentication systems; however, these do not fully satisfy the requirements as imposed by the Directive. For example, systems such as Privacy Pass [14] are susceptible to abuse (e.g., Sybil attacks) and do not support stateful interactions. Authentication frameworks like FIDO2 [23] do not provide mechanisms for controlled identity disclosure. Similarly, anonymous credential systems, including U-Prove [41, 42] and Idemix [9, 10, 44, 46], remain insufficiently mature for deployment in this context.

Whistleblowing systems are also related to areas such as electronic voting and auditable communication protocols (see e.g., [30]), where privacy, integrity, and accountability must be carefully balanced. However, whistleblowing introduces additional requirements not typically directly addressed in these systems but prescribed by the Directive, most importantly controlled identity disclosure and management of stateful operations over the lifecycle of the report.

Finally, the Directive itself references an existing and deployed reporting platform, the EU Sanctions whistleblowing service [52], which supports both identified and anonymous submissions.[17] However, this service provides little transparency regarding its underlying architecture and security, particularly with respect to confidentiality guarantees and the extent to which its functionality aligns with the procedural requirements mandated by the Directive.[18]

Many of these limitations are representative of a broader class of commercial whistleblowing platforms marketed as "guaranteed-compliance" solutions. Such systems, offered by private vendors,

[17] The website in fact encourages identified reporting. See https://eusanctions.integrityline.com/. Accessed: 2026-04-27.

[18] This service does not appear to be compliant even with the basic timeline and procedural obligations.

typically operate as closed, black-box services.[19] While they assert properties such as confidentiality and regulatory compliance, these claims are rarely substantiated by publicly available technical documentation or independently verifiable security analyses. Consequently, users (potential reports and companies alike) are required to rely largely on trust in the providers.

Taken together, there is a clear gap. While legal, organizational, and governance frameworks define what compliant whistleblowing systems should achieve, and existing technical work provides partial building blocks, there is a lack of integrated and deployable system architectures that simultaneously satisfy the Directive's security and procedural requirements.

8 Conclusions and Future Work

This paper argues that Directive-compliant whistleblowing should not be modeled as a one-time anonymous submission mechanism, but as an interactive, stateful, and auditable process. Treating Directive (EU) 2019/1937 as a normative source of technical requirements rather than merely as policy background leads to a different design methodology. Accordingly, we derived procedures, assets, threats, and security goals from the Directive and used them through the design process of the Whistleblowing System. Concretely, the paper contributes at three levels. First, it derives a realistic threat model and a set of security goals tailored to directive-compliant whistleblowing. Second, it proposes a concrete Whistleblowing System design that separates the Enrollment System from the Reporting System while relying on an Auditable Log Service and an Identity Management Component as common technical building blocks. Third, it discusses implementation and deployment considerations that connect the design to realistic organizational and regulatory settings.

Several directions remain for future work. First, an attempt should be made to instantiate the design with possibly efficient cryptographic mechanisms. Second, the system should be validated against diverse national transpositions of the Directive and sector-specific reporting requirements, since key procedural and compliance choices are delegated to Member States and regulators. Third, further work is needed to develop practical integration pathways with existing digital identity and trust-service ecosystems. In particular, the eIDAS Regulation [55] ecosystem provides a natural deployment setting. Its trust services, such as electronic ledgers and qualified timestamps, align with the requirements of the auditable logging component, while the European Digital Identity Wallet offers a plausible option for user credential management during enrollment. Finally, an important direction for future work is to determine whether it is both necessary and feasible to incorporate additional security guarantees into the system, such as safeguards against whistleblowing coercion, in line with the concept of coercion-freeness in voting systems [31].

Generative AI Usage. Some text was edited for grammar, flow, and conciseness using Grammarly and ChatGPT tools. All AI-assisted content has been carefully checked and verified.

[19] See e.g., https://www.speakup.com/blog/top-whistleblowing-software-tools. Accessed: 2026-04-27.

References

[1] Thomas Agrikola, Geoffroy Couteau, and Sven Maier. 2021. Anonymous Whistleblowing over Authenticated Channels. Cryptology ePrint Archive, Paper 2021/1341. https://eprint.iacr.org/2021/1341

[2] Sebastian Angel, Hao Chen, Yan Ji, Andrei Lapets, Srinath Setty, and Michael Walfish. 2022. CoverDrop: Blowing the Whistle Through A News App. *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2022, 2 (2022), 458–477. doi:10.56553/popets-2022-0035

[3] Foteini Baldimtsi, Jiaqi Cheng, Rishab Goyal, and Aayush Yadav. 2024. Non-interactive Blind Signatures: Post-quantum and Stronger Security. Cryptology ePrint Archive, Paper 2024/614. https://eprint.iacr.org/2024/614

[4] Foteini Baldimtsi, Rishab Goyal, and Aayush Yadav. 2025. Batched & Non-interactive Blind Signatures from Lattices. Cryptology ePrint Archive, Paper 2025/1771. https://eprint.iacr.org/2025/1771

[5] Bettina Berendt and Stefan Schiffner. 2021. Whistleblower protection in the digital age – why 'anonymous' is not enough. From technology to a wider view of governance. doi:10.48550/ARXIV.2111.02825

[6] Manuel Blum, Paul Feldman, and Silvio Micali. 1988. Non-Interactive Zero-Knowledge and Its Applications. 103–112 pages. doi:10.1145/62212.62222

[7] Gabriel Bracha. 1987. Asynchronous Byzantine Agreement Protocols. *Information and Computation* 75, 2 (1987), 130–143.

[8] Christian Cachin, Rachid Guerraoui, and Luís Rodrigues. 2011. *Introduction to Reliable and Secure Distributed Programming* (2nd ed.). Springer, Berlin, Heidelberg.

[9] Jan Camenisch and Els Van Herreweghen. 2002. Design and Implementation of the Idemix Anonymous Credential System. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM, Washington, DC, USA, 21–30. doi:10.1145/586110.586114

[10] Jan Camenisch and Anna Lysyanskaya. 2001. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. Cryptology ePrint Archive, Paper 2001/019. https://eprint.iacr.org/2001/019

[11] Ran Canetti. 2000. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Paper 2000/067. https://eprint.iacr.org/2000/067

[12] Anastasia Cheliatsidou, Nikolaos Sariannidis, Alexandros Garefalakis, Ioannis Passas, and Konstantinos Spinthiropoulos. 2023. Exploring Attitudes towards Whistleblowing in Relation to Sustainable Municipalities. doi:10.3390/admsci13090199

[13] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. 1985. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *26th Annual Symposium on Foundations of Computer Science (SFCS 1985)*. IEEE, Portland, OR, USA, 383–395. doi:10.1109/SFCS.1985.64

[14] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. 2018. Privacy Pass: Bypassing Internet Challenges Anonymously. *Proceedings on Privacy Enhancing Technologies* 2018, 3 (April 2018), 164–180. doi:10.1515/popets-2018-0026

[15] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. 303–320 pages.

[16] Ben Dooley. 2024. *Ten years on, "Lux Leaks" remains a byword for corporate tax chicanery.* International Consortium of Investigative Journalists. https://www.icij.org/investigations/luxembourg-leaks/ten-years-on-lux-leaks-remains-a-byword-for-corporate-tax-chicanery/ Accessed: 2026-04-22.

[17] Janelle Brinker Dozier and Marcia P. Miceli. 1985. Potential Predictors of Whistle-Blowing: A Prosocial Behavior Perspective. *The Academy of Management Review* 10, 4 (Oct. 1985), 823. doi:10.2307/258050

[18] EBSCO Information Services. n.d.. Volkswagen emissions scandal ("dieselgate"). https://www.ebsco.com/research-starters/business-and-management/volkswagen-emissions-scandal-dieselgate Accessed: 2026-04-22.

[19] Leora F. Eisenstadt and Jennifer M. Pacella. 2018. Whistleblowers Need Not Apply. *American Business Law Journal* 55, 4 (Nov. 2018), 665–719. doi:10.1111/ablj.12131

[20] European Parliament. 2017. *The Impact of Schemes revealed by the Panama Papers on the Economy and Finances of a Sample of Member States.* Study PE 572.717. European Parliament, Policy Department for Citizens' Rights and Constitutional Affairs. https://www.europarl.europa.eu/RegData/etudes/STUD/2017/572717/IPOL_STU(2017)572717_EN.pdf Accessed: 2026-04-22.

[21] European Parliament and Council of the European Union. 2024. Directive (EU) 2019/1937 of the European Parliament and of the Council of 23 October 2019 on the protection of persons who report breaches of Union law. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019L1937 Accessed: 2024-12-02.

[22] European Parliamentary Research Service. 2016. *"Panama papers" in a nutshell.* At a Glance PE 580.903. European Parliament. https://www.europarl.europa.eu/RegData/etudes/ATAG/2016/580903/EPRS_ATA(2016)580903_EN.pdf Accessed: 2026-04-22.

[23] FIDO Alliance. 2023. Overview of the FIDO Alliance. https://fidoalliance.org/overview/ Accessed: 2024-11-13.

[24] Freedom of the Press Foundation. 2026. SecureDrop. https://securedrop.org/.

[25] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. 2014. Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials. Cryptology ePrint Archive, Paper 2014/944. doi:10.1007/s00145-018-9281-4
[26] Lucjan Hanzlik. 2023. Non-Interactive Blind Signatures for Random Messages. Cryptology ePrint Archive, Paper 2023/388. https://eprint.iacr.org/2023/388
[27] Daniel Hugenroth et al. 2025. *CoverDrop: Guardian Implementation White Paper*. Technical Report. The Guardian / University of Cambridge. https://www.coverdrop.org/coverdrop_guardian_implementation_june_2025.pdf Accessed: 2026-04-29.
[28] iden3. 2018. circomlib: Library of basic circuits for circom. https://github.com/iden3/circomlib.
[29] International Organization for Standardization. 2021. ISO 37002:2021 Whistleblowing management systems — Guidelines. https://www.iso.org/standard/65035.html.
[30] Ari Juels, Dario Catalano, and Markus Jakobsson. 2002. Coercion-Resistant Electronic Elections. Cryptology ePrint Archive, Paper 2002/165. https://eprint.iacr.org/2002/165
[31] Ari Juels, Dario Catalano, and Markus Jakobsson. 2004. Coercion-Resistance and Receipt-Freeness in Electronic Voting. *IEEE Security & Privacy* 2, 3 (2004), 41–49. doi:10.1109/MSP.2004.17
[32] Tino Jungebloud, Nhung H. Nguyen, Dongseong Kim, and Armin Zimmermann. 2025. Model-based structural and behavioral cybersecurity risk assessment in system designs. *Computers & Security* 157 (2025), —. doi:10.1016/j.cose.2025.XXXXXX
[33] Jonathan Katz and Yehuda Lindell. 2020. *Introduction to Modern Cryptography* (3rd ed.). Chapman and Hall/CRC, Boca Raton, FL.
[34] Julia Kaufmann, Katja Häferer, and Kerstin Grimhardt. 2019. The new EU Whistleblowing Directive. *Computer Law Review International* 21 (2019), 14–17. https://api.semanticscholar.org/CorpusID:211724190
[35] Magdalena Kun-Buczko. 2022. Poles' Attitudes to the Concept of Whistleblowing. Historical and Present Background. *Białostockie Studia Prawnicze* 27 (12 2022), 121–134. doi:10.15290/bsp.2022.27.04.08
[36] Mirosław Kutyłowski and Gabriel Wechta. 2025. Pseudonymization and reporters' protection by design in the EU whistleblower directive. *Journal of Cybersecurity* 11, 1 (10 2025), tyaf028. arXiv:https://academic.oup.com/cybersecurity/article-pdf/11/1/tyaf028/64829139/tyaf028.pdf doi:10.1093/cybsec/tyaf028
[37] Lydia Mechtenberg, Gerd Muehlheusser, and Andreas Roider. 2020. Whistleblower protection: Theory and experimental evidence. *European Economic Review* 126 (2020), 103447. doi:10.1016/j.euroecorev.2020.103447
[38] NAVEX Global. 2021. Whistleblowing in Italy: The EU Whistleblower Protection Directive. https://cdn.navex.com/image/upload/v1635985206/media/EU_Directive_Article_Italy_Viewpoint_ENG.pdf Accessed: 2024-10-28.
[39] Notes from Poland. 2024. Polish higher education minister resigns following criticism over revealing name of whistleblower. https://notesfrompoland.com/2024/12/19/polish-higher-education-minister-resigns-following-criticism-over-revealing-name-of-whistleblower/
[40] Jennifer M. Pacella. 2022. Making Whistleblowers Whole. https://escholarship.org/uc/item/3dk4h7xr
[41] Christian Paquin and Greg Zaverucha. 2014. *U-Prove Cryptographic Specification*. Technical Report. Microsoft Corporation. https://github.com/microsoft/uprove-node-reference/blob/main/doc/U-Prove%20Cryptographic%20Specification%20V1.1%20Revision%205.pdf Accessed: 2024-11-18.
[42] Christian Paquin and Greg Zaverucha. 2024. *U-Prove Technology Overview*. Technical Report. Microsoft Corporation. https://github.com/microsoft/uprove-node-reference/blob/main/doc/U-Prove%20Technology%20Overview%20V1.1%20Revision%203.pdf Accessed: 2024-11-18.
[43] European Parliament. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). 88 pages.
[44] Hyperledger Fabric Project. 2024. MSP Implementation with Identity Mixer. https://hyperledger-fabric.readthedocs.io/en/latest/idemix.html (accessed November 27, 2024).
[45] Willy Quach, LaKyah Tyner, and Daniel Wichs. 2023. Lower Bounds on Anonymous Whistleblowing. Cryptology ePrint Archive, Paper 2023/1483. https://eprint.iacr.org/2023/1483
[46] IBM Research. 2009. Specification of the Identity Mixer Cryptographic Library Version 2.3.0. https://dominoweb.draco.res.ibm.com/reports/rz3730_revised.pdf Accessed December 16, 2024.
[47] Harald Scheu. 2024. Pitfalls in Implementing the EU Whistleblower Directive. *Wroclaw Review of Law, Administration & Economics* 14 (2024), 1–12. doi:10.2478/wrlae-2024-0002
[48] SCIPR Lab and contributors. 2012–2017. libsnark: a C++ library for zkSNARK proofs. https://github.com/scipr-lab/libsnark.
[49] Dimitri Staufer, Frank Pallas, and Bettina Berendt. 2024. Silencing the Risk, Not the Whistle: A Semi-automated Text Sanitization Tool for Mitigating the Risk of Whistleblower Re-Identification. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (Rio de Janeiro, Brazil) *(FAccT '24)*. Association for Computing Machinery, New York, NY, USA, 733–745. doi:10.1145/3630106.3658936
[50] F.M. Teichmann and C. Wittmann. 2022. Whistleblowing: procedural and dogmatic problems in the implementation of directive (EU) 2019/1937. *Journal of Financial Regulation and Compliance* 30, 5 (2022), 553–566. doi:10.1108/JFRC-12-2021-0118
[51] Fabian Maximilian Johannes Teichmann and Chiara Wittmann. 2022. Cultural aspects of the EU-Whistleblowing Directive. *Journal of Financial Crime* 30, 4 (June 2022), 999–1005. doi:10.1108/jfc-04-2022-0091
[52] The European Commission. 2024. EU Sanctions Compliance Reporting Platform. https://finance.ec.europa.eu/eu-and-world/sanctions-restrictive-measures/overview-sanctions-and-related-resources/eu-sanctions-whistleblower-tool_en Accessed: 2024-11-14.
[53] Transparency International. 2022. Internal Whistleblowing Systems: Best Practice Principles for Public and Private Organisations. https://www.transparency.org/en/publications/internal-whistleblowing-systems.
[54] Transparency International. 2024. *EU whistleblower protection: Falling short when it matters most.* Transparency International. https://www.transparency.org/en/news/eu-whistleblower-protection-falling-short-when-it-matters-most
[55] European Union. 2024. Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 on European Digital Identity (eIDAS 2.0). https://eur-lex.europa.eu/eli/reg/2024/1183/oj/eng
[56] Wim Vandekerckhove and Andrew Samuels. 2026. Internal reporting channels and the EU Whistleblowing Directive. A white paper on big problems and easy fixes. doi:10.13140/RG.2.2.20084.21126
[57] Krystyna Ziółkowska. 2023. The (Legal) Status of a Whistleblower in Poland. Selected Issues. *Studies in Logic, Grammar and Rhetoric* 68 (12 2023), 573–588. doi:10.2478/slgr-2023-0033

A Open Science

All information used in this work is contained within the article and its appendices. There are no additional research artifacts needed to evaluate the research contributions.

B Ethical Considerations

This paper does not involve any ethical issues.

C Organization of the Appendices

In Section K, we present details of the cryptographic primitives used to construct ALS and IMC and components as well as the implementation of the Whistleblowing System. In section E, we present details of the sequence of actions which must be run on the report in the Reporting System. In section J, we present tables describing details of the "funnel" analysis that pilots the reader from the requirements included in the Directive to the system requirements given in this paper. In Sections F.3 and G.3 we present details of implementations of the ALS and IMC components. In Section I, we present a security analysis of our implementation of Whistleblowing System.

D Preliminaries

We first summarize the cryptographic notions used to construct the building blocks for our system and the system design itself. Our construction relies on:

- a digital signature scheme $\mathsf{Sig} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$;
- a CCA-secure symmetric key encryption scheme $\mathsf{Sym} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$;
- a CCA-secure symmetric key authenticated encryption scheme $\mathsf{SymAuth} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$;

- a CCA-secure asymmetric key encryption scheme Asm = (KeyGen, Enc, Dec);
- a hiding randomness commitment scheme RandCom = (Gen, Verify);
- a certificate scheme Crt = (KeyGen, Certify, Verify, Decode);
- a pseudorandom function PRF_K, a verifiable secret sharing scheme VSS = (Share, Verify, Recon); and
- a non-interactive zero-knowledge proof of knowledge $\mathsf{NIZKPoK}_{\mathcal{R}}$ = (Setup, Prove, Verify).

Detailed definitions are provided in Section K. In the definitions below, λ indicates a security parameter.

E State Diagram

State diagram is presented in Figure 6.

F ALS Technical Details

This section contains technical details of Auditable Log Service (ALS).

F.1 ALS Interface

We now define the interface of ALS. For convenience, we omit writing the system state state explicitly as an input to every procedure. Unless stated otherwise, the procedures operate on the current state, and the procedures Log, Send, SendConf, and ReceiveConf additionally update it.

Definition 3 (Auditable Log Service). Auditable Log Service *is a system that implements the following set of PPT procedures (some of which are run as two-party protocols):*

— $\mathsf{AuditorSetup}(1^\lambda) \to (\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}})$*: Takes as input a security parameter* 1^λ *and returns a pair* $(\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}})$*, where* $\mathsf{sk}^{\mathsf{audit}}$ *is the private audit key and* $\mathsf{pk}^{\mathsf{audit}}$ *is the corresponding public audit key. The* $\mathsf{pk}^{\mathsf{audit}}$ *is made publicly available.*

— $\mathsf{Setup}(1^\lambda, \mathsf{pk}^{\mathsf{audit}}) \to (\mathsf{sparams}, \mathsf{pparams}, \mathsf{state})$*: Takes as input a security parameter* 1^λ *and auditor's public key* $\mathsf{pk}^{\mathsf{audit}}$ *and outputs secret parameters* sparams*, public parameters* pparams *and an initial (empty) system state* state*. This procedure initializes the* ALS *environment prior to any communication. The* state *includes* ALS.PublicView*, the publicly available view of all message-records maintained by* ALS*. The public parameters* pparams *are made publicly available.*

— $\mathsf{Log}(\mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2) \to (\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2})$*: Takes as input the auditor's public key* $\mathsf{pk}^{\mathsf{audit}}$*, a level-1 message* msg_1*, and a level-2 message* msg_2*, and outputs a unique message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, an opening token* $\mathsf{t}^{\mathsf{msg}_2}$ *for* msg_2*. As a result, a new message-record with* sent *status is appended to the* state*. The public key* $\mathsf{pk}^{\mathsf{audit}}$ *enables auditing; and* msg_1 *together with* msg_2 *form the message content. The opening token* $\mathsf{t}^{\mathsf{msg}_2}$ *is output to the sender to enable cooperation with the auditor during level-2 audit.*

— $\mathsf{AuditOpenMsg1}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}) \to \{\mathsf{msg}_1, \perp\}$*: Takes as input the auditor's secret key* $\mathsf{sk}^{\mathsf{audit}}$ *and a message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, and outputs the* msg_1 *field of the* $\mathsf{id}^{\mathsf{mr}}$ *message-record if* $\mathsf{id}^{\mathsf{mr}}$ *exists and has* sent *status, otherwise* $\perp$*.*

— $\mathsf{AuditOpenMsg2}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2}) \to \{\mathsf{msg}_2, \perp\}$*: Takes as input the auditor's secret key* $\mathsf{sk}^{\mathsf{audit}}$*, a message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, an opening token* $\mathsf{t}^{\mathsf{msg}_2}$*, and outputs* msg_2 *if* $\mathsf{id}^{\mathsf{mr}}$ *exists, has* sent *status, and* $\mathsf{t}^{\mathsf{msg}_2}$ *corresponds* $\mathsf{id}^{\mathsf{mr}}$*, otherwise* $\perp$*.*

— $\mathsf{GenReceiverParams}(1^\lambda) \to (\mathsf{sk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{recv}})$*: Takes as input a security parameter* 1^λ *and outputs a receiver key pair* $(\mathsf{sk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{recv}})$ *used by the receiver for addressing and receiving messages.*[20]

— $\mathsf{Send}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2) \to (\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2})$*: Takes as input the receiver's public key* $\mathsf{pk}^{\mathsf{recv}}$*, the auditor's public key* $\mathsf{pk}^{\mathsf{audit}}$*, a level-1 message* msg_1*, and a level-2 message* msg_2*, and outputs a unique message-record identifier* $\mathsf{id}^{\mathsf{mr}}$ *and an opening token* $\mathsf{t}^{\mathsf{msg}_2}$ *for* msg_2*. This procedure is invoked by the sender and as a result, a new message-record with* sent *status is appended to the* state *of* ALS*. The public key* $\mathsf{pk}^{\mathsf{recv}}$ *is used to identify and later authenticate the receiver; the public key* $\mathsf{pk}^{\mathsf{audit}}$ *identifies the auditor; and* msg_1 *together with* msg_2 *form the message content. The opening token* $\mathsf{t}^{\mathsf{msg}_2}$ *is output to the sender to enable cooperation with the auditor in a level-2 audit.*

— $\mathsf{Check}(\mathsf{sk}^{\mathsf{recv}}) \to \langle(\mathsf{id}^{\mathsf{mr}})_i\rangle$*: Takes as input the receiver's secret key* $\mathsf{sk}^{\mathsf{recv}}$ *and outputs a (possibly empty) list* $\langle(\mathsf{id}^{\mathsf{mr}})_i\rangle$ *of message-record identifiers intended for that receiver. This procedure is invoked by the receiver to discover and list all available messages.*

— $\mathsf{Receive}(\mathsf{sk}^{\mathsf{recv}}, \mathsf{id}^{\mathsf{mr}}) \to \{(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2}), \perp\}$*: Takes as input the receiver's secret key* $\mathsf{sk}^{\mathsf{recv}}$ *and a message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, and outputs* $(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2})$ *if* $\mathsf{sk}^{\mathsf{recv}}$ *corresponds to the public key associated with* $\mathsf{id}^{\mathsf{mr}}$ *message-record, otherwise* $\perp$*. This function can be invoked only if a message was sent using the* Send*, otherwise it returns* $\perp$*.*

— $\mathsf{SendConf}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2) \to (\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2}, \mathsf{t}^{\mathsf{verec}})$*: Takes as input the receiver's public key* $\mathsf{pk}^{\mathsf{recv}}$*, the auditor's public key* $\mathsf{pk}^{\mathsf{audit}}$*, a level-1 message* msg_1*, and a level-2 message* msg_2*, and outputs a unique message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, an opening token* $\mathsf{t}^{\mathsf{msg}_2}$ *for* msg_2*, and a receipt verification token* $\mathsf{t}^{\mathsf{verec}}$*. This procedure is invoked by the sender and constitutes an interaction with* ALS *provider. As a result, a new message-record with* sent *status is appended to the* state*.* $\mathsf{pk}^{\mathsf{recv}}$ *is used to identify and later authenticate the receiver;* $\mathsf{pk}^{\mathsf{audit}}$ *enables auditing; and* msg_1 *together with* msg_2 *form the message content. The opening token* $\mathsf{t}^{\mathsf{msg}_2}$ *is output to the sender to enable cooperation with the auditor in a level-2 audit.* $\mathsf{t}^{\mathsf{verec}}$ *allows the sender to verify that the message was received.*

— $\mathsf{ReceiveConf}(\mathsf{sk}^{\mathsf{recv}}, \mathsf{id}^{\mathsf{mr}}) \to \{(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2}), \perp\}$*: Takes as input the receiver's secret key* $\mathsf{sk}^{\mathsf{recv}}$ *and a message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, and outputs* $(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2})$ *if* $\mathsf{sk}^{\mathsf{recv}}$ *corresponds to the public key associated with* $\mathsf{id}^{\mathsf{mr}}$ *message-record, otherwise outputs* $\perp$*. This procedure is invoked by the receiver and requires interaction with* ALS *provider. This function can be invoked only if a message was sent using* SendConf *function, otherwise it returns* $\perp$*.*

— $\mathsf{VerifyMsgRecv}(\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{verec}}) \to \{b, \perp\}$*: Takes as input the sender's receipt verification token* $\mathsf{t}^{\mathsf{verec}}$ *and a message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, and outputs* $b \in \{0, 1\}$ *if* $\mathsf{id}^{\mathsf{mr}}$ *exists, has* sent *status and* $\mathsf{t}^{\mathsf{verec}}$ *corresponds to* $\mathsf{id}^{\mathsf{mr}}$*,* $\perp$ *otherwise. The output* $b = 1$ *indicates that the message-record has been successfully received; otherwise,* $b = 0$*. This function is invoked only by the sender.*[21]

[20] We do not define an analogous key-generation procedure for senders, since sending a message in ALS does not require a sender-specific key pair.

[21] Defining VerifyMsgSent for communicating entities is unnecessary. For the sender, it is trivial, as they already know they sent the message. For the receiver, message existence becomes known only once obtained from Check. Likewise, VerifyMsgRecv is redundant for the receiver, who already knows they have received the message.

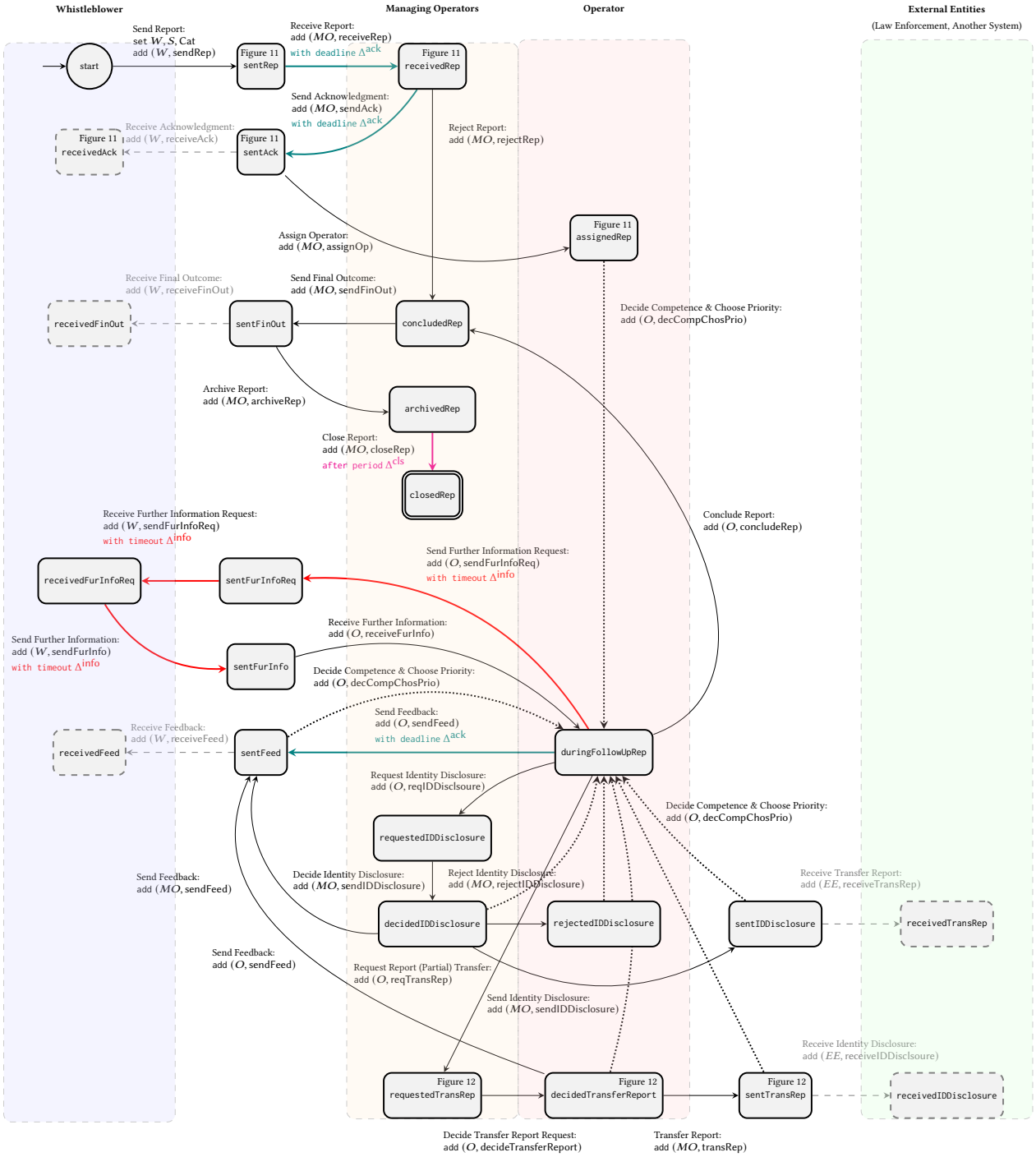

Figure 6: Complete overview of the report workflow. Each node (rectangle) represents a report state, and each arrow denotes an action. Every action is associated with a responsible entity, a unified action type, and, where applicable, a time bound. When an arrow enters a labeled column (whistleblower, managing operators, operator, external entity), the column label identifies the entity responsible for that action. When an arrow leaves a labeled column and enters an unlabeled column (e.g., the sendRep **transition), the label of the subsequent action specifies the responsible entity. Time bounds are indicated by colors and timers, as defined in Section 4.3. Gray denotes actions executed in parallel, meaning the system does not block in the corresponding state. The action** decCompChosPrio**, being a "coupling" action, is represented with dotted lines to improve clarity.**

— $\mathsf{AuditVerifyMsgSent}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}) \to \{b, \bot\}$*: Takes as input the auditor's secret key* $\mathsf{sk}^{\mathsf{audit}}$ *and a message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, and outputs* $b \in \{0,1\}$ *if* $\mathsf{id}^{\mathsf{mr}}$ *exists and has* sent *status,* $\bot$ *otherwise. The output* $b = 1$ *indicates that the message with message-record identifier* $\mathsf{id}^{\mathsf{mr}}$ *was sent, otherwise* $b = 0$*. This function is invoked only by the auditor.*

— $\mathsf{AuditVerifyMsgRecv}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}) \to \{b, \bot\}$*: Takes as input the auditor's secret key* $\mathsf{sk}^{\mathsf{audit}}$ *and a message-record identifier* $\mathsf{id}^{\mathsf{mr}}$*, and outputs* $b \in \{0,1\}$ *if* $\mathsf{id}^{\mathsf{mr}}$ *exists and has* sent *status,* $\bot$ *otherwise. The output* $b = 1$ *indicates that the message with message-record identifier* $\mathsf{id}^{\mathsf{mr}}$ *was received by the receiver, otherwise* $b = 0$*. This function is invoked only by the auditor.*

In ALS, the public keys of senders and receivers effectively serve as public addresses. The ALS abstraction does not prescribe how such addresses are managed: they may be single-use, persistent, or any mixture thereof. Depending on the deployment, these addresses may or may not be linkable to a real-world identity. In the following discussion, we use the term *address* to capture this notion more generally.

F.2 ALS Security Properties

ALS satisfies the following security properties:

P1: Correctness. For ALS initialized with

$$(\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}}) \leftarrow \mathsf{AuditorSetup}(1^\lambda),$$

$$(\mathsf{sparams}, \mathsf{pparams}, \mathsf{state}) \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pk}^{\mathsf{audit}}),$$

and a receiver key pair generated with $(\mathsf{sk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{recv}}) \leftarrow \mathsf{GenReceiverParams}(1^\lambda)$, it holds that for any two-part message $\mathsf{msg} = (\mathsf{msg}_1, \mathsf{msg}_2)$ output by a polynomial-time entity, msg which is logged, sent or sent with receival confirmation to the holder of $\mathsf{pk}^{\mathsf{recv}}$ via

$$(\mathsf{id}^{\mathsf{mr}}, _) \leftarrow \mathsf{Log}(\mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2),$$

$$(\mathsf{id}^{\mathsf{mr}}, _) \leftarrow \mathsf{Send}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2), \text{ and}$$

$$(\mathsf{id}^{\mathsf{mr}}, _, _) \leftarrow \mathsf{SendConf}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2),$$

respectively. A new message with record identifier $\mathsf{id}^{\mathsf{mr}}$ is created. In the case of Send and SendConf the holder of $\mathsf{sk}^{\mathsf{recv}}$ can receive the list of all message records addressed to them by invoking $\langle(\mathsf{id}^{\mathsf{mr}})_i\rangle \leftarrow \mathsf{Check}(\mathsf{sk}^{\mathsf{recv}})$, after which they may receive the message via

$$(\mathsf{msg}_1, \mathsf{msg}_2, _) \leftarrow \mathsf{Receive}(\mathsf{sk}^{\mathsf{recv}}, \mathsf{id}^{\mathsf{mr}}),$$

$$(\mathsf{msg}_1, \mathsf{msg}_2, _) \leftarrow \mathsf{ReceiveConf}(\mathsf{sk}^{\mathsf{recv}}, \mathsf{id}^{\mathsf{mr}}),$$

respectively. The latter appends a new record $(\mathsf{id}^{\mathsf{mr}}, \mathsf{recv})$ to the state.

P2: Sender anonymity. Neither the act of sending a message nor the resulting message record reveals any information about the sender to any entity.

P3: Receiver confidentiality. Neither the act of sending a message to a specified receiver nor any interaction of a sender and receiver with the system reveals information about the receiver to any entity. With an exception that the holder of $\mathsf{sk}^{\mathsf{audit}}$ may learn the receiver's public key via a level-1 audit. In particular, the ALS provider remains oblivious to the address of the receiver.

P4: Message confidentiality. Only the holder of the receiver's key $\mathsf{sk}^{\mathsf{recv}}$ can access both parts of the message, msg_1 and msg_2.[22] In addition, only the holder of $\mathsf{sk}^{\mathsf{audit}}$ can access these parts via level-1 and level-2 audits, respectively (see P9 and P10). No other entity, including the ALS provider, can infer any information about the message content from observing message records. Moreover, inspection of the state reveals nothing beyond the mere fact that a communication took place.

P5: Message status confidentiality. In the case of (SendConf, ReceiveConf), no entity other than ALS provider, the auditor, the sender, and the intended receiver can distinguish whether a message identified by $\mathsf{id}^{\mathsf{mr}}$ was received (i.e., whether state contains a record $(\mathsf{id}^{\mathsf{mr}}, \mathsf{recv})$ for this message). In the case of (Send, Receive), only the receiver learns the reception status of the message.

P6: Assured availability. All system operations including message sending, receiving, checking, and auditing are accessible to all authorized entities whenever invoked.

P7: Integrity protection. Messages sent and received via ALS remain unmodified, preventing unauthorized alteration during transmission or storage.

P8: Receival control. Entity holding the verification token $\mathsf{t}^{\mathsf{verec}}$, obtained by running $(\mathsf{id}^{\mathsf{mr}}, _, \mathsf{t}^{\mathsf{verec}}) \leftarrow \mathsf{SendConf}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2)$, can verify whether the message identified by $\mathsf{id}^{\mathsf{mr}}$ has been received by executing $b \leftarrow \mathsf{VerifyMsgRecv}(\mathsf{t}^{\mathsf{verec}}, \mathsf{id}^{\mathsf{mr}})$. The output bit b equals 1 if and only if the message was received, in which case the state contains the record $(\mathsf{id}^{\mathsf{mr}}, \mathsf{recv})$. Otherwise, the state does not contain this record and $b = 0$.

P9: Level-1 auditability. Entity holding the secret key $\mathsf{sk}^{\mathsf{audit}}$, can verify the status of a sent message via $\mathsf{AuditVerifyMsgSent}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}})$ and $\mathsf{AuditVerifyMsgRecv}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}})$, as well as access the level-1 message by running $\mathsf{msg}_1 \leftarrow \mathsf{AuditOpenMsg1}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}})$, for any message record identified by $\mathsf{id}^{\mathsf{mr}}$ without requiring interaction with the ALS provider. Accessing the message status enables time binding of the message — for sent, this binds the message to the time it was sent, whereas for recv, it confirms that a message addressed to a specific address was received at a particular time.

P10: Level-2 auditability. Entity holding the secret key $\mathsf{sk}^{\mathsf{audit}}$, can, upon request from either the sender or the receiver of the message with message record $\mathsf{id}^{\mathsf{mr}}$ and upon receiving the corresponding opening token $\mathsf{t}^{\mathsf{msg}_2}$ from one of the entities, access the level-2 message by running $\mathsf{msg}_2 \leftarrow \mathsf{AuditOpenMsg2}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2})$, without requiring interaction with the ALS provider.

F.3 ALS Technical Overview

We proceed by considering a concrete implementation of ALS. We assume that the ALS is operated by an honest-but-curious entity, referred to as ALS provider, who process requests faithfully. In other words, they correctly execute requested function on provided inputs and do not compromise the integrity of the data, this includes providing correct teimestamps. However, we do not assume that the ALS provider preserve data confidentiality; still being interested in inspecting the data they handle. Note that these assumptions

[22] Both messages are output by a polynomial-time entity.

capture the scenario in which a service is operated by a trusted entity (such as a government body), which is susceptible to malicious takeover or data leaks.

The main role of the ALS provider is to initialize and keep the database of records that can be efficiently queried using a unique identifier through publicly available ALS.PublicView. Additionally, we assume that all communication with ALS provider occurs over an anonymous channel; consequently, ALS provider is oblivious to any metadata associated with the communicating entity (in particular, its location), beyond the time of access.

Assuming that ALS provider executes requests honestly it is enough to ensure that messages transmitted through ALS are delivered without compromising their confidentiality, that is, while sending a level-1 message msg_1, the sender must guarantee confidentiality and receiver privacy, even in the presence of curious ALS provider or external observers, and ensure that the receiver as well as the auditor obtain the identical message. To achieve this, sender generates a fresh symmetric encryption key. This key is then encrypted under both the receiver's and the auditor's public keys. The message msg_1 is encrypted using the symmetric key, and sender creates a NIZK proof demonstrating that the same key is used consistently across these encryptions. Additionally, the sender encrypts the receiver's public key $\mathsf{pk}^{\mathsf{recv}}$ under both the receiver's and auditor's public keys, and extends the NIZK proof that ensures consistency across all ciphertexts. Once the sender submits the generated ciphertexts and the NIZK to the ALS provider, the ALS provider assigns a unique identifier to the record and returns it to the sender. The receiver can later query stored records and identify the ones addressed to them by checking which entry properly decrypts to $\mathsf{pk}^{\mathsf{recv}}$.

Logging of level-1 messages follows a similar process: a symmetric key is generated to encrypt the data and then is encrypted under the auditor's public key. The encrypted message, along with a consistency proof, is stored in a database record that can be queried by a unique identifier.

To provide receive confirmation, the sender commits to a random value, appends it to the encryption of receiver's public key $\mathsf{pk}^{\mathsf{recv}}$. These elements are sent along with the artifacts described in the previous paragraphs together with an updated NIZK proof. When the receiver requests the message, they must additionally send the random value (i.e., commitment opening); only then the ALS provider releases the key allowing receiver to access the ciphertext containing the actual message.

Level-2 messages are processed similarly as the level-1 messages, except that the symmetric key used to encrypt these messages is encrypted only under the receiver's public key (and not under the auditor's public key). And thus to receive the message auditor first must acquire that symmetric key from either sender or receiver.

F.4 ALS Implementation

In this section, we present details of the ALS construction.

Let $\mathcal{R}^{\mathsf{ALS}}_{\mathsf{Send}}$ denote an NP-hard relation describing all pairs (x, w), where for each statement

$x = (\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, c^{\mathsf{msg}_2}_{\mathsf{msg}}, c^{\mathsf{msg}_2}_{\mathsf{key}})$,

there exists a witness $w = (\mathsf{msg}_1, \mathsf{msg}_2, K^{\mathsf{msg}_1}, K^{\mathsf{msg}_2})$ such that:

(1) $c^{\mathsf{tar}}_{\mathsf{R}}$ ($c^{\mathsf{tar}}_{\mathsf{A}}$) is a ciphertext of $\mathsf{pk}^{\mathsf{recv}}$ under key $\mathsf{pk}^{\mathsf{recv}}$ ($\mathsf{pk}^{\mathsf{audit}}$),
(2) $c^{\mathsf{msg}_1}_{\mathsf{msg}}$ is a ciphertext of msg_1 under key K^{msg_1},
(3) $c^{\mathsf{msg}_1}_{\mathsf{R,key}}$ ($c^{\mathsf{msg}_1}_{\mathsf{A,key}}$) is a ciphertext of K^{msg_1} under key $\mathsf{pk}^{\mathsf{recv}}$ ($\mathsf{pk}^{\mathsf{audit}}$),
(4) $c^{\mathsf{msg}_2}_{\mathsf{msg}}$ is a ciphertext of msg_2 under key K^{msg_2},
(5) $c^{\mathsf{msg}_2}_{\mathsf{key}}$ is a ciphertext of K^{msg_2} under key $\mathsf{pk}^{\mathsf{recv}}$.

Let $\mathcal{R}^{\mathsf{ALS}}_{\mathsf{SendConf}}$ denote an NP-hard relation describing all pairs (x, w), where for each statement

$x = (\mathsf{com}, \mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, c^{\mathsf{msg}_2}_{\mathsf{msg}}, c^{\mathsf{msg}_2}_{\mathsf{key}})$,

there exists a witness $w = (\mathsf{val}, \mathsf{msg}_1, \mathsf{msg}_2, K^{\mathsf{msg}_1}, K^{\mathsf{msg}_2})$ such that:

(1) val is the opening of com,
(2) $c^{\mathsf{tar}}_{\mathsf{R}}$ ($c^{\mathsf{tar}}_{\mathsf{A}}$) is a ciphertext of $\mathsf{pk}^{\mathsf{recv}}$ under key $\mathsf{pk}^{\mathsf{recv}}$ ($\mathsf{pk}^{\mathsf{audit}}$),
(3) $c^{\mathsf{msg}_1}_{\mathsf{msg}}$ is a ciphertext of msg_1 under key K^{msg_1},
(4) $c^{\mathsf{msg}_1}_{\mathsf{R,key}}$ ($c^{\mathsf{msg}_1}_{\mathsf{A,key}}$) is a ciphertext of K^{msg_1} under key $\mathsf{pk}^{\mathsf{recv}}$ ($\mathsf{pk}^{\mathsf{audit}}$),
(5) $c^{\mathsf{msg}_2}_{\mathsf{msg}}$ is a ciphertext of msg_2 under key K^{msg_2},
(6) $c^{\mathsf{msg}_2}_{\mathsf{key}}$ is a ciphertext of K^{msg_2} under key $\mathsf{pk}^{\mathsf{recv}}$.

Additionally, let $\mathsf{GenUUID}() \to \mathsf{id}$ be a procedure that outputs a universally unique identifier for a message record, and let $\mathsf{GetTime}() \to \mathsf{time}$ be a procedure that outputs the current time. ALS is implemented as follows:

— $\mathsf{AuditorSetup}(1^\lambda) \to (\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}})$: Auditor generates $(\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}}) \leftarrow \mathsf{Asm.KeyGen}(1^\lambda)$ and outputs $(\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}})$.

— $\mathsf{Setup}(1^\lambda, \mathsf{pk}^{\mathsf{audit}}) \to (\mathsf{sparams}, \mathsf{pparams}, \mathsf{state})$: ALS provider samples secret parameters sparams, and publishes public parameters pparams, which include: the address of ALS.PublicView, and common reference strings for NIZK proofs, namely $\mathsf{CRS}^{\mathsf{ALS}}_{\mathsf{Send}} \leftarrow \mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{ALS}}_{\mathsf{Send}}}.\mathsf{Setup}(1^\lambda)$ and $\mathsf{CRS}^{\mathsf{ALS}}_{\mathsf{SendConf}} \leftarrow \mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{ALS}}_{\mathsf{SendConf}}}.\mathsf{Setup}(1^\lambda)$. It further initializes state as the current state of ALS.PublicView and outputs $(\mathsf{sparams}, \mathsf{pparams}, \mathsf{state})$.

— $\mathsf{GenReceiverParams}(1^\lambda) \to (\mathsf{sk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{recv}})$: Generate $(\mathsf{sk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{recv}}) \leftarrow \mathsf{Asm.KeyGen}(1^\lambda)$ and output $(\mathsf{sk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{recv}})$.

— $\mathsf{Log}(\mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2) \to (\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2})$: The sender (logger) encrypts the level-1 message for the auditor $c^{\mathsf{msg}_1}_{\mathsf{A}} \leftarrow \mathsf{Asm.Enc}_{\mathsf{pk}^{\mathsf{audit}}}(\mathsf{msg}_1)$. Generates a symmetric authentication key $K^{\mathsf{msg}_2} \leftarrow \mathsf{SymAuth.KeyGen}(1^\lambda)$ and encrypts the level-2 message under this key $c^{\mathsf{msg}_2}_{\mathsf{msg}} \leftarrow \mathsf{SymAuth.Enc}_{K^{\mathsf{msg}_2}}(\mathsf{msg}_2)$. The sender sends $(c^{\mathsf{msg}_1}_{\mathsf{A}}, c^{\mathsf{msg}_2}_{\mathsf{msg}})$ to ALS provider. ALS provider samples a unique message record identifier $\mathsf{id}^{\mathsf{mr}} \leftarrow \mathsf{ALS.GenUUID}()$. Next, sends $\mathsf{id}^{\mathsf{mr}}$ to the sender. Then encrypts the type of the message for the auditor $c^{\mathsf{typ}}_{\mathsf{A}} \leftarrow \mathsf{Asm.Enc}_{\mathsf{pk}^{\mathsf{audit}}}(\text{“log”})$. Finally, ALS provider records $\mathsf{id}^{\mathsf{mr}}$: (“log”) and publishes $(\mathsf{id}^{\mathsf{mr}}, c^{\mathsf{msg}_1}_{\mathsf{A}}, c^{\mathsf{msg}_2}, c^{\mathsf{typ}}_{\mathsf{A}})$ to ALS.PublicView. Sender outputs $(\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2} \coloneqq K^{\mathsf{msg}_2})$.

— $\mathsf{Send}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2) \to (\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2})$: The sender encrypts the receiver's public key $\mathsf{pk}^{\mathsf{recv}}$ under the receiver and auditor public keys, i.e., $(c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}) \leftarrow \mathsf{Asm.Enc}_{(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}})}(\mathsf{pk}^{\mathsf{recv}})$. Next generates symmetric encryption key $K^{\mathsf{msg}_1} \leftarrow \mathsf{Sym.KeyGen}(1^\lambda)$, encrypts the level-1 message $c^{\mathsf{msg}_1}_{\mathsf{msg}} \leftarrow \mathsf{Sym.Enc}_{K^{\mathsf{msg}_1}}(\mathsf{msg}_1)$, then encrypts K^{msg_1} under

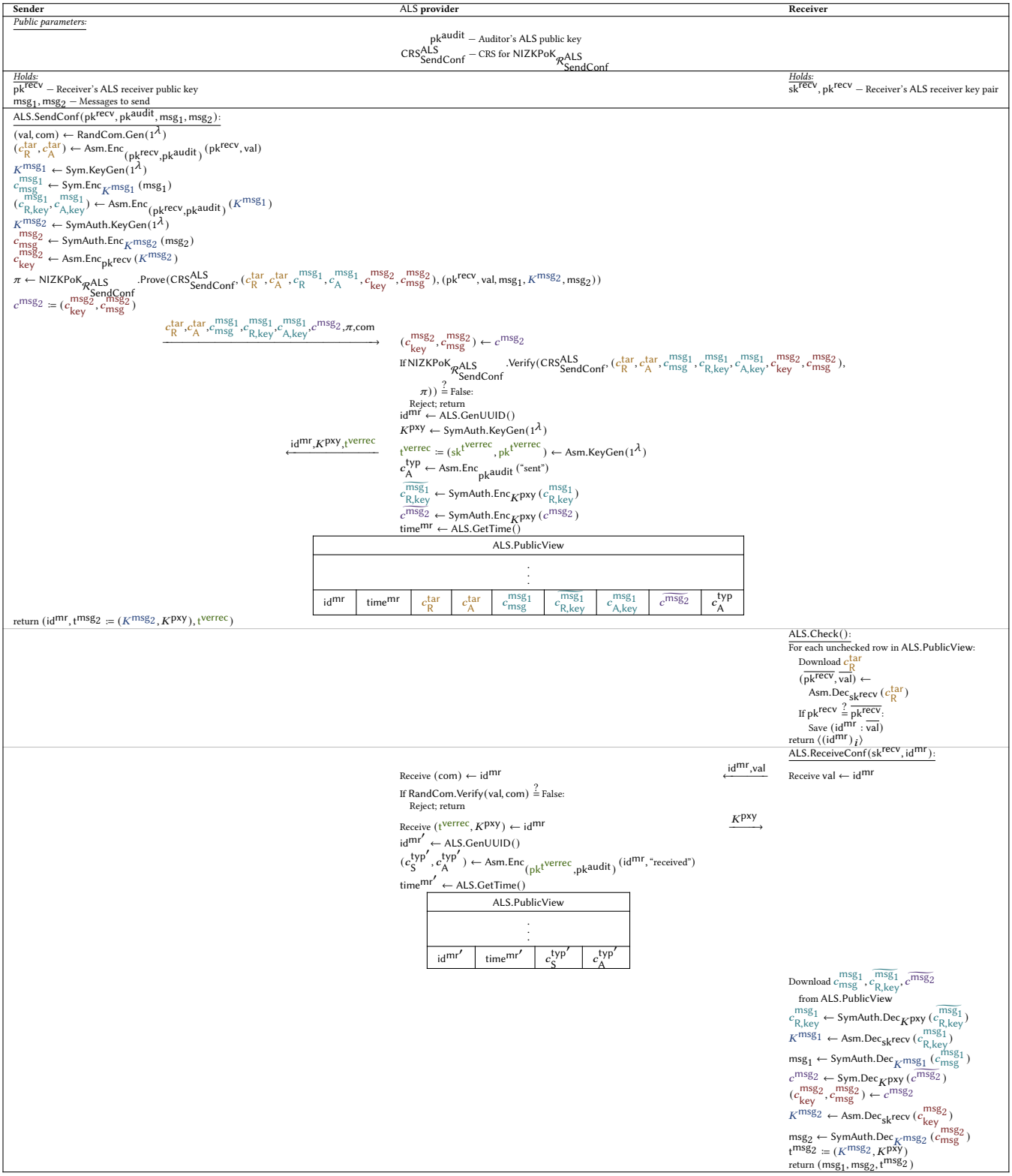

Figure 7: Implementation of ALS.SendConf, ALS.Check, and ALS.ReceiveConf. The figure illustrates the interactions between the message sender, the ALS provider, and the receiver, along with the corresponding function calls initiated by each entity.

the receiver and auditor public keys, namely $(c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}) \leftarrow \mathsf{Asm.Enc}_{(\mathsf{pk}^{\mathsf{recv}},\mathsf{pk}^{\mathsf{audit}})}(K^{\mathsf{msg}_1})$. Then generates a symmetric encryption with authentication key $K^{\mathsf{msg}_2} \leftarrow \mathsf{SymAuth.KeyGen}(1^\lambda)$ and encrypts the level-2 message $c^{\mathsf{msg}_2}_{\mathsf{msg}} \leftarrow \mathsf{SymAuth.Enc}_{K^{\mathsf{msg}_2}}(\mathsf{msg}_2)$. Then encrypts K^{msg_2} under the receiver's public key $c^{\mathsf{msg}_2}_{\mathsf{key}} \leftarrow \mathsf{Asm.Enc}_{\mathsf{pk}^{\mathsf{recv}}}(K^{\mathsf{msg}_2})$. The sender sets $x := (\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, c^{\mathsf{msg}_2}_{\mathsf{msg}}, c^{\mathsf{msg}_2}_{\mathsf{key}})$ and $w := (\mathsf{msg}_1, \mathsf{msg}_2, K^{\mathsf{msg}_1}, K^{\mathsf{msg}_2})$. Then generates a NIZKPoK proof π by running $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{ALS}}_{\mathsf{Send}}}.\mathsf{Prove}(\mathsf{CRS}^{\mathsf{ALS}}_{\mathsf{Send}}, x, w)$ and sends $(c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, c^{\mathsf{msg}_2} := (c^{\mathsf{msg}_2}_{\mathsf{key}}, c^{\mathsf{msg}_2}_{\mathsf{msg}}), \pi)$ to ALS provider. ALS provider verifies the $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{ALS}}_{\mathsf{Send}}}$ proof π for $\mathsf{CRS}^{\mathsf{ALS}}_{\mathsf{Send}}$ and statement x. If verification fails, ALS provider rejects the interaction; otherwise, it samples a unique message record identifier $\mathsf{id}^{\mathsf{mr}} \leftarrow \mathsf{ALS.GenUUID}()$. Next, sends $\mathsf{id}^{\mathsf{mr}}$ to the sender. Then encrypts the type of the message for the auditor $c^{\mathsf{typ}}_{\mathsf{A}} \leftarrow \mathsf{Asm.Enc}_{\mathsf{pk}^{\mathsf{audit}}}(\text{“sent”})$. Finally, records $\mathsf{id}^{\mathsf{mr}} : (\text{“sent”})$ and publishes $(\mathsf{id}^{\mathsf{mr}}, c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, c^{\mathsf{msg}_2}, c^{\mathsf{typ}}_{\mathsf{A}})$ to ALS.PublicView. Sender outputs $(\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2} := K^{\mathsf{msg}_2})$.

— $\mathsf{SendConf}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2) \rightarrow (\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2}, \mathsf{t}^{\mathsf{verrec}})$: The sender samples a commitment and its opening $(\mathsf{com}, \mathsf{val}) \leftarrow \mathsf{RandCom.Gen}(1^\lambda)$, which will later be used to authorize the receiver to obtain the level-2 message. Then encrypts the receiver's public key $\mathsf{pk}^{\mathsf{recv}}$ and the commitment opening val under the receiver and auditor public keys, i.e., $(c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}) \leftarrow \mathsf{Asm.Enc}_{(\mathsf{pk}^{\mathsf{recv}},\mathsf{pk}^{\mathsf{audit}})}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{val})$. Next generates symmetric encryption key $K^{\mathsf{msg}_1} \leftarrow \mathsf{Sym.KeyGen}(1^\lambda)$, encrypts the level-1 message $c^{\mathsf{msg}_1}_{\mathsf{msg}} \leftarrow \mathsf{Sym.Enc}_{K^{\mathsf{msg}_1}}(\mathsf{msg}_1)$, then encrypts K^{msg_1} under the receiver and auditor public keys, namely $(c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}) \leftarrow \mathsf{Asm.Enc}_{(\mathsf{pk}^{\mathsf{recv}},\mathsf{pk}^{\mathsf{audit}})}(K^{\mathsf{msg}_1})$. Then generates a symmetric encryption with authentication key $K^{\mathsf{msg}_2} \leftarrow \mathsf{SymAuth.KeyGen}(1^\lambda)$ and encrypts the level-2 message $c^{\mathsf{msg}_2}_{\mathsf{msg}} \leftarrow \mathsf{SymAuth.Enc}_{K^{\mathsf{msg}_2}}(\mathsf{msg}_2)$. Then encrypts K^{msg_2} under the receiver public key $c^{\mathsf{msg}_2}_{\mathsf{key}} \leftarrow \mathsf{Asm.Enc}_{\mathsf{pk}^{\mathsf{recv}}}(K^{\mathsf{msg}_2})$. The sender sets $x = (\mathsf{com}, \mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, c^{\mathsf{msg}_2}_{\mathsf{msg}}, c^{\mathsf{msg}_2}_{\mathsf{key}})$ and $w = (\mathsf{val}, \mathsf{msg}_1, \mathsf{msg}_2, K^{\mathsf{msg}_1}, K^{\mathsf{msg}_2})$. Then generates a NIZKPoK proof π by running $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{ALS}}_{\mathsf{SendConf}}}.\mathsf{Prove}(\mathsf{CRS}^{\mathsf{ALS}}_{\mathsf{SendConf}}, x, w)$ and sends $(c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, c^{\mathsf{msg}_2} := (c^{\mathsf{msg}_2}_{\mathsf{key}}, c^{\mathsf{msg}_2}_{\mathsf{msg}}), \pi, \mathsf{com})$ to ALS provider. ALS provider verifies the $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{ALS}}_{\mathsf{SendConf}}}$ proof π for $\mathsf{CRS}^{\mathsf{ALS}}_{\mathsf{SendConf}}$ and statement x. If verification fails, ALS provider rejects the interaction; otherwise, it samples a unique message record identifier $\mathsf{id}^{\mathsf{mr}} \leftarrow \mathsf{ALS.GenUUID}()$. It then generates a symmetric encryption with authentication key K^{pxy}, which enforces interaction from the receiver while ensuring that level-2 audit requires at least one of the sender–receiver pair. Next, ALS provider samples an asymmetric key pair for verifying receipt of the message for the sender $\mathsf{t}^{\mathsf{verrec}} := (\mathsf{sk}^{\mathsf{t}^{\mathsf{verrec}}}, \mathsf{pk}^{\mathsf{t}^{\mathsf{verrec}}}) \leftarrow \mathsf{Asm.KeyGen}(1^\lambda)$ and sends $(\mathsf{id}^{\mathsf{mr}}, K^{\mathsf{pxy}}, \mathsf{t}^{\mathsf{verrec}})$ to the sender. Then encrypts the type of the message for the auditor $c^{\mathsf{typ}}_{\mathsf{A}} \leftarrow \mathsf{Asm.Enc}_{\mathsf{pk}^{\mathsf{audit}}}(\text{“sent”})$. Next encrypts the receiver's part of K^{msg_1} ciphertext, that is $\widetilde{c^{\mathsf{msg}_1}_{\mathsf{R,key}}} \leftarrow \mathsf{SymAuth.Enc}_{K^{\mathsf{pxy}}}(c^{\mathsf{msg}_1}_{\mathsf{R,key}})$ and both ciphertexts of msg_2, that is, let $c^{\mathsf{msg}_2} := (c^{\mathsf{msg}_2}_{\mathsf{key}}, c^{\mathsf{msg}_2}_{\mathsf{msg}})$, then $\widetilde{c^{\mathsf{msg}_2}} \leftarrow \mathsf{SymAuth.Enc}_{K^{\mathsf{pxy}}}(c^{\mathsf{msg}_2})$. Finally, records $\mathsf{id}^{\mathsf{mr}} : (\text{“sent”}, \mathsf{com}, K^{\mathsf{pxy}}, \mathsf{t}^{\mathsf{verrec}})$ and publishes $(\mathsf{id}^{\mathsf{mr}}, c^{\mathsf{tar}}_{\mathsf{R}}, c^{\mathsf{tar}}_{\mathsf{A}}, c^{\mathsf{msg}_1}_{\mathsf{msg}}, \widetilde{c^{\mathsf{msg}_1}_{\mathsf{R,key}}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}, \widetilde{c^{\mathsf{msg}_2}}, c^{\mathsf{typ}}_{\mathsf{A}})$ to ALS.PublicView. Sender outputs $(\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2} := (K^{\mathsf{msg}_2}, K^{\mathsf{pxy}}), \mathsf{t}^{\mathsf{verrec}})$.

— $\mathsf{Check}(\mathsf{sk}^{\mathsf{recv}}) \rightarrow \langle(\mathsf{id}^{\mathsf{mr}})_i\rangle$: The receiver for each unchecked row in ALS.PublicView downloads $c^{\mathsf{tar}}_{\mathsf{R}}$. If row was a result of SendConf, then tries to decrypt $(\overline{\mathsf{pk}^{\mathsf{recv}}}, \overline{\mathsf{val}}) \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{recv}}}(c^{\mathsf{tar}}_{\mathsf{R}})$ otherwise just $\overline{\mathsf{pk}^{\mathsf{recv}}} \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{recv}}}(c^{\mathsf{tar}}_{\mathsf{R}})$, and verifies if the message was addressed to them by checking if $\overline{\mathsf{pk}^{\mathsf{recv}}}$ is equal to $\mathsf{pk}^{\mathsf{recv}}$. If true and if row was a result of SendConf records $\mathsf{id}^{\mathsf{mr}} : (\overline{\mathsf{val}})$, otherwise just records $\mathsf{id}^{\mathsf{mr}}$. Outputs $\langle(\mathsf{id}^{\mathsf{mr}})_i\rangle$.

— $\mathsf{Receive}(\mathsf{sk}^{\mathsf{recv}}, \mathsf{id}^{\mathsf{mr}}) \rightarrow \{(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2}), \perp\}$: The receiver downloads $c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{R,key}}$ and c^{msg_2} associated with $\mathsf{id}^{\mathsf{mr}}$ from ALS.PublicView. First, receives $K^{\mathsf{msg}_1} \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{recv}}}(c^{\mathsf{msg}_1}_{\mathsf{R,key}})$, and level-1 message $\mathsf{msg}_1 \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{msg}_1}}(c^{\mathsf{msg}_1}_{\mathsf{msg}})$, then receives $K^{\mathsf{msg}_2} \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{recv}}}(c^{\mathsf{msg}_2}_{\mathsf{key}})$, and level-2 message $\mathsf{msg}_2 \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{msg}_2}}(c^{\mathsf{msg}_2}_{\mathsf{msg}})$. Receiver outputs $(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2} := K^{\mathsf{msg}_2})$.

— $\mathsf{ReceiveConf}(\mathsf{sk}^{\mathsf{recv}}, \mathsf{id}^{\mathsf{mr}}) \rightarrow \{(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2}), \perp\}$: The receiver receives $(\mathsf{val}) \leftarrow \mathsf{id}^{\mathsf{mr}}$ which was previously recorded during Check. Sends $(\mathsf{id}^{\mathsf{mr}}, \mathsf{val})$ to ALS provider. ALS provider receives $(\mathsf{com}) \leftarrow \mathsf{id}^{\mathsf{mr}}$. Verifies if receiver is authorized to receive the message by checking if val is opening to com by running $\mathsf{RandCom.Verify}(\mathsf{val}, \mathsf{com})$, if verification fails ALS provider aborts and receiver outputs $\perp$. Otherwise, ALS provider receives $(\mathsf{t}^{\mathsf{verrec}}, K^{\mathsf{pxy}}) \leftarrow \mathsf{id}^{\mathsf{mr}}$ sends K^{pxy} to receiver, samples $\mathsf{id}^{\mathsf{mr}'} \leftarrow \mathsf{ALS.GenUUID}()$ for receive confirmation message record and encrypts type of the message record together with the message record identifier to which this message is received by running $(c^{\mathsf{typ}'}_{\mathsf{S}}, c^{\mathsf{typ}'}_{\mathsf{A}}) \leftarrow \mathsf{Asm.Enc}_{(\mathsf{pk}^{\mathsf{t}^{\mathsf{verrec}}},\mathsf{pk}^{\mathsf{audit}})}(\mathsf{id}^{\mathsf{mr}}, \text{“received”})$ under receipt verification key $\mathsf{pk}^{\mathsf{t}^{\mathsf{verrec}}}$ and auditor's public key $\mathsf{pk}^{\mathsf{audit}}$, respectively. Finally, records $\mathsf{id}^{\mathsf{mr}} : (\text{“received”})$ and publishes $(\mathsf{id}^{\mathsf{mr}'}, c^{\mathsf{typ}'}_{\mathsf{S}}, c^{\mathsf{typ}'}_{\mathsf{A}})$ to ALS.PublicView.

After receiving K^{pxy}, receiver downloads $c^{\mathsf{msg}_1}_{\mathsf{msg}}, \widetilde{c^{\mathsf{msg}_1}_{\mathsf{R,key}}}$ and $\widetilde{c^{\mathsf{msg}_2}}$ associated with $\mathsf{id}^{\mathsf{mr}}$ from ALS.PublicView. First, receives $c^{\mathsf{msg}_1}_{\mathsf{R,key}} \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{pxy}}}(\widetilde{c^{\mathsf{msg}_1}_{\mathsf{R,key}}})$, then $K^{\mathsf{msg}_1} \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{recv}}}(c^{\mathsf{msg}_1}_{\mathsf{R,key}})$, and level-1 message $\mathsf{msg}_1 \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{msg}_1}}(c^{\mathsf{msg}_1}_{\mathsf{msg}})$, then receives $c^{\mathsf{msg}_2} \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{pxy}}}(\widetilde{c^{\mathsf{msg}_2}})$, then $K^{\mathsf{msg}_2} \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{recv}}}(c^{\mathsf{msg}_2}_{\mathsf{key}})$, and level-2 message $\mathsf{msg}_2 \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{msg}_2}}(c^{\mathsf{msg}_2}_{\mathsf{msg}})$. Receiver outputs $(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2} := K^{\mathsf{msg}_2})$.

— $\mathsf{AuditOpenMsg1}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}) \rightarrow \{\mathsf{msg}_1, \perp\}$: The auditor locates in ALS.PublicView message record identified by $\mathsf{id}^{\mathsf{mr}}$, if no such record exists outputs $\perp$, otherwise auditor downloads $c^{\mathsf{msg}_1}_{\mathsf{msg}}, c^{\mathsf{msg}_1}_{\mathsf{A,key}}$ and decrypts $K^{\mathsf{msg}_1} \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{audit}}}(c^{\mathsf{msg}_1}_{\mathsf{A,key}})$ and then $\mathsf{msg}_1 \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{msg}_1}}(c^{\mathsf{msg}_1}_{\mathsf{msg}})$. Outputs msg_1.

— $\mathsf{AuditOpenMsg2}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2}) \rightarrow \{\mathsf{msg}_2, \perp\}$: The auditor first receives $\mathsf{t}^{\mathsf{msg}_2}$ handed to them by one of the entities.

If $\mathsf{id}^{\mathsf{mr}}$ was the result of SendConf, auditor parses $\mathsf{t}^{\mathsf{msg}_2}$ as $(K^{\mathsf{msg}_2}, K^{\mathsf{pxy}})$, then downloads $\widetilde{c^{\mathsf{msg}_2}}$ from ALS.PublicView for $\mathsf{id}^{\mathsf{mr}}$. Runs $(_, c^{\mathsf{msg}_2}_{\mathsf{msg}}) \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{pxy}}}(\widetilde{c^{\mathsf{msg}_2}})$, if authentication fails, auditor outputs $\perp$. Otherwise receives level-2 message $\mathsf{msg}_2 \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{msg}_2}}(c^{\mathsf{msg}_2}_{\mathsf{msg}})$, if authentication fails outputs $\perp$.

If $\mathsf{id}^{\mathsf{mr}}$ was the result of Send or Log, auditor parses $\mathsf{t}^{\mathsf{msg}_2}$ as K^{msg_2} then downloads c^{msg_2} associated with $\mathsf{id}^{\mathsf{mr}}$ from ALS.PublicView. Receives level-2 message $\mathsf{msg}_2 \leftarrow \mathsf{SymAuth.Dec}_{K^{\mathsf{msg}_2}}(c^{\mathsf{msg}_2}_{\mathsf{msg}})$, if authentication fails outputs $\perp$. Outputs msg_2.

— $\mathsf{VerifyMsgRecv}(\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{verrec}}) \rightarrow \{b, \perp\}$: The sender locates in ALS.PublicView message record identified by $\mathsf{id}^{\mathsf{mr}}$, if no such record exists outputs $\perp$, otherwise for each unchecked record after $\mathsf{id}^{\mathsf{mr}}$ downloads $c^{\mathsf{typ}'}_{\mathsf{S}}$, tries to decrypt $(\widetilde{\mathsf{id}^{\mathsf{mr}}}, \text{“}\widetilde{\text{received}}\text{”}) \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{verrec}}}(c^{\mathsf{typ}'}_{\mathsf{S}})$, if $(\widetilde{\mathsf{id}^{\mathsf{mr}}}, \text{“}\widetilde{\text{received}}\text{”})$ is equal to $(\mathsf{id}^{\mathsf{mr}}, \text{“received”})$ sets $b \coloneqq 1$, otherwise sets $b \coloneqq 0$. Outputs b.

— $\mathsf{AuditVerifyMsgRecv}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}) \rightarrow \{b, \perp\}$: The auditor locates in ALS.PublicView message record identified by $\mathsf{id}^{\mathsf{mr}}$, if no such record exists outputs $\perp$, otherwise for each unchecked record after $\mathsf{id}^{\mathsf{mr}}$ downloads $c^{\mathsf{typ}'}_{\mathsf{S}}$, tries to decrypt $(\widetilde{\mathsf{id}^{\mathsf{mr}}}, \text{“}\widetilde{\text{received}}\text{”}) \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{audit}}}(c^{\mathsf{typ}'}_{\mathsf{A}})$, if $(\widetilde{\mathsf{id}^{\mathsf{mr}}}, \text{“}\widetilde{\text{received}}\text{”})$ is equal to $(\mathsf{id}^{\mathsf{mr}}, \text{“received”})$ sets $b \coloneqq 1$, otherwise sets $b \coloneqq 0$. Outputs b.

— $\mathsf{AuditVerifyMsgSent}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}) \rightarrow \{b, \perp\}$: The auditor locates in ALS.PublicView message record identified by $\mathsf{id}^{\mathsf{mr}}$, if no such record exists outputs $\perp$, otherwise downloads $c^{\mathsf{typ}}_{\mathsf{A}}$, tries to decrypt $\text{“}\widetilde{\text{sent}}\text{”} \leftarrow \mathsf{Asm.Dec}_{\mathsf{sk}^{\mathsf{verrec}}}(c^{\mathsf{typ}}_{\mathsf{A}})$, if $\text{“}\widetilde{\text{sent}}\text{”}$ is equal to “sent” sets $b \coloneqq 1$, otherwise sets $b \coloneqq 0$. Outputs b.

F.5 ALS Security Analysis

Lemma 1. *The construction presented in Section F.4 satisfies all* Auditable Log Service *security properties defined in Section F.2, P1–P10.*

Arg. In the analysis below, we assume that the setup procedure is run in a trusted manner, i.e. by a set of entities which honestly follow the protocol instructions.

It is easy to see that the scheme enjoys P1: Correctness for each communication flow.

For P2: Sender anonymity, the sender's identity is protected due to the assumption of an anonymous communication channel (see Section 5.1). Messages are received "on demand" via the Check, Receive, and ReceiveConf interfaces, which mitigates information leakage (e.g., through shoulder surfing or timing observation). Additionally, the ALS send layer operates independently of any sender-specific information, in particular, it does not rely on the sender's identity or any sender-specific (also private) information.

P3: Receiver confidentiality, similarly to sender anonymity, the receiver's confidentiality is protected due to the assumption of an anonymous communication channel (see Section 5.1). The ALS provider remains oblivious to the receiver's identity, as the receiver's address is never revealed to it. Only the intended receiver can find out that they are the addressee, since the receiver's address (its public key) is encrypted under the receiver's public key. To enable message receival, a one-time identification mechanism is used: the sender sends an encrypted opening (intended for the receiver) to a commitment known to ALS provider. This allows ALS provider to verify that a requesting entity is indeed the legitimate addressee of the message, without requiring access to the receiver's long-term ALS public key. Furthermore, the receiver's public key is also encrypted under the auditor's public key. This ensures that, the auditor can recover the receiver's public key while it remains hidden from other entities.

For P4: Message confidentiality, both msg_1 and msg_2 are encrypted under symmetric keys which are again encrypted under public keys of the receiver and the auditor. Consequently, only the intended receiver and the auditor are able to access the contents of these messages. Importantly, the auditor requires an additional opening token to access msg_2, which can only be obtained from either the sender or the receiver. This restriction ensures that the auditor cannot unilaterally decrypt all transmitted data. Moreover, since all messages processed by ALS provider is encrypted, any observer of the ALS.PublicView, including ALS provider, cannot derive any meaningful information about the transferred messages, beyond leakage inherent to the encryption scheme, namely the length of the plaintext (see Definition 12) and the fact that some communication has occurred.

We argue the P5: Message status confidentiality as follows. The message record confirming that another message record was received is meaningful only to the sender, auditor, as it contains information encrypted under their respective public keys, and receiver who claimed the message. For all other entities, this message record reveals only an information that *some* message record was received, by the properties of the encryption scheme it does not reveal which of the previously submitted message records was received.

P6: Assured availability of ALS is ensured by its deployment in a high-availability environment, incorporating redundancy, fault tolerance, and distributed infrastructure. Moreover, as long as the underlying communication channels and hosting infrastructure remain operational, honest entities can reliably access ALS interfaces and receive their messages. This assumption is standard and orthogonal to the cryptographic guarantees provided by the scheme.

For P7: Integrity protection, we rely on the assumption that ALS provider is an honest-but-curious entity that faithfully executes the protocol. In particular, it correctly processes send and receive requests without modifying, injecting, or deleting the proxied and stored data.

P8: Receival control follows directly from the construction.

Lastly, the remaining two properties, P9: Level-1 auditability and P10: Level-2 auditability, follow from the verification procedure enforced by ALS provider. In particular, for a msg_1 and msg_2 to be published in ALS.PublicView (i.e., accepted by ALS provider), the associated NIZK proofs must verify successfully. Therefore, a sender cannot submit a message with an incorrectly constructed auditability layer, as such a message would be rejected by ALS provider. Consequently, all accepted messages necessarily satisfy the required auditability guarantees. □

G IMC Technical Details

This section contains technical details of Identity Management Component (IMC).

G.1 IMC Interface

We now present the interface of Identity Management Component (IMC) and the corresponding security requirements.

Definition 4 (Identity Management Component). *Given a certificate scheme* Crt = (KeyGen, Certify, Verify, Decode) *and an NP-hard relation* $\mathcal{R}$ *the* Identity Management Component IMC *is a tuple of PPT algorithms:*

— IMC.GenSysParams$(1^\lambda) \to (\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{sys}})$*: Takes as input the security parameter* 1^λ *and returns a pair* $(\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{sys}})$*, where* $\mathsf{sk}^{\mathsf{sys}}$ *is the private system key and* $\mathsf{pk}^{\mathsf{sys}}$ *is the corresponding public system key.*

— IMC.GenMemberParams$(1^\lambda) \to (\mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}})$*: Takes as input the security parameter* 1^λ *and returns a pair* $(\mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}})$*, where* $\mathsf{sk}^{\mathsf{mem}}$ *is the private member key and* $\mathsf{pk}^{\mathsf{mem}}$ *is the corresponding public member key.*

— IMC.GenTicket$(\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{mem}}, \tau) \to$ ticket*: Takes as input a private system key* $\mathsf{sk}^{\mathsf{sys}}$*, a public member key* $\mathsf{pk}^{\mathsf{mem}}$*, and a nametag* τ*, and returns a ticket* ticket *that can be used by the whistleblower to generate report pseudonyms.*

— IMC.VerifyTicket$(\mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{ticket}, \tau) \to b$*: Takes as input a public system key* $\mathsf{pk}^{\mathsf{sys}}$*, a public member key* $\mathsf{pk}^{\mathsf{mem}}$*, a ticket* ticket*, and an identity tag* τ*, and returns* 1 *if and only if* ticket *is a valid ticket for the nametag* τ *and the public member key* $\mathsf{pk}^{\mathsf{mem}}$*. Otherwise, the procedure returns* 0*.*

— IMC.GenRepnym$(\mathsf{sk}^{\mathsf{mem}}, \mathsf{ticket}, \rho) \to$ repnym*: Takes as input a private member key* $\mathsf{sk}^{\mathsf{mem}}$*, a valid ticket* ticket*, a report tag* ρ*, and returns a report pseudonym* repnym*. The function can be used to generate report pseudonyms for distinct values of the report tag.*

— IMC.CertifyRepnym$(\mathsf{pk}^{\mathsf{sys}}, \mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{repnym}, \tau, \rho, \mathsf{ticket}, \mathsf{cert}^{\mathsf{ent}}, \mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}}, \mathsf{pol}^\tau, \mathsf{pol}^\psi) \to (\mathsf{cert}^{\mathsf{repnym}}, \boldsymbol{s}^\tau, \boldsymbol{s}^\psi)$*: Takes as input a public key of the identity system* $\mathsf{pk}^{\mathsf{sys}}$*, private and public member keys* $\mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}}$*, a report pseudonym* repnym*, a nametag* τ*, a report tag* ρ*, a valid ticket* ticket*, a valid certificate* cert *under* $\mathsf{pk}^{\mathsf{ent}}$ *for a nametag* τ*, secret and public keys of the certificate's receiver* $\mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}}$*, a policy* pol^τ *indicating number of nametag decryption shares and a nametag decryption threshold, a policy* pol^ψ *indicating number of linkability decryption shares and a linkability decryption threshold and returns a certificate for the report pseudonym* $\mathsf{cert}^{\mathsf{repnym}}$ *that consists of three parts* $(\mathsf{cert}^\pi, \mathsf{cert}^\tau, \mathsf{cert}^\psi)$ *along with shareset for nametag* $\boldsymbol{s}^\tau$ *and a shareset for linkability* $\boldsymbol{s}^\psi$*. The three-part certificate* $\mathsf{cert}^{\mathsf{repnym}}$ *certifies that (*)* repnym *is a valid report pseudonym for some valid ticket* ticket *bound to the report tag* ρ *and some nametag* τ *issued for a member key* $\mathsf{pk}^{\mathsf{mem}}$ *and (*) given* cert^τ *and sufficient number of shares in* $\boldsymbol{s}^\tau$ *it is possible to extract the nametag* τ *associated with the report pseudonym under policy for tag disclosure* pol^τ*, as well as given* cert^ψ *and sufficient number of shares in* $\boldsymbol{s}^\psi$ *it is possible to extract trapdoor* ψ *that allows to link all pseudonyms generated using a single valid ticket under policy for trapdoor disclosure* pol^ψ *(*) there exists a certificate* $\mathsf{cert}^{\mathsf{ent}}$ *verifiable under* $\mathsf{pk}^{\mathsf{ent}}$ *for the nametag* τ*, and a key* $\mathsf{pk}^{\mathsf{ent}}$ *such that* $(\mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}})$ *are in a NP-hard relation* $\mathcal{R}$ *and the pseudonym certificate contains a proof of knowledge of the string* $\mathsf{sk}^{\mathsf{ent}}$ *uniquely bound to the pseudonym* repnym*.*

— IMC.VerifyRepnymCert$(\mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{cert}}, \mathsf{repnym}, \rho, \mathsf{cert}^{\mathsf{repnym}}, \mathsf{pol}^\tau, \mathsf{pol}^\psi) \to b$*: Takes as input a public system key* $\mathsf{pk}^{\mathsf{sys}}$*, a public key of a certificate authority* $\mathsf{pk}^{\mathsf{cert}}$*, a report pseudonym* repnym*, a report tag* ρ*, a certificate* $\mathsf{cert}^{\mathsf{repnym}}$ *and returns* 1 *if and only if* $\mathsf{cert}^{\mathsf{repnym}}$ *is a valid report certificate for the report pseudonym* repnym*.*

— IMC.VerifyNametagShare$(\mathsf{cert}^\tau, s^\tau) \to b$ *(or* IMC.VerifyLinkabilityShare$(\mathsf{cert}^\psi, s^\psi) \to b$*): Takes as input a valid repnym cert* cert^τ *(or* cert^ψ*) and a share* s^τ *for the nametag (or a share* s^ψ *for linkability) and outputs* 1 *if the share is a valid share for the certificate* $\mathsf{cert}^{\mathsf{repnym}}$*, otherwise outputs* 0*.*

— IMC.ExtractNametag$(\mathsf{cert}^{\mathsf{repnym}}, \boldsymbol{s}^\tau) \to \tau$*: Takes as input a valid* repnym *certificate* $\mathsf{cert}^{\mathsf{repnym}}$ *encrypted under policy* pol^τ *and returns a tag* τ *if and only if* $\boldsymbol{s}^\tau$ *contains sufficient number of shares to decrypt.*

— IMC.ExtractLinkability$(\mathsf{cert}^{\mathsf{repnym}}, \boldsymbol{s}^\psi) \to (\tau, \psi)$*: Takes as input a valid* repnym *certificate* $\mathsf{cert}^{\mathsf{repnym}}$ *encrypted under policy* pol^ψ *and returns a linkability information* ψ *if and only if* $\boldsymbol{s}^\psi$ *contains sufficient number of shares to decrypt.*

— IMC.GenEntry$(\mathsf{sk}^{\mathsf{mem}}, \mathsf{repnym}, \mathsf{ticket}, \rho, msg) \to \epsilon$*: Takes as input a valid member's secret key* $\mathsf{sk}^{\mathsf{mem}}$*, a valid report pseudonym* repnym*, a report ticket* ticket*, a report tag* ρ*, a message msg, and returns an entry* ϵ *for the message msg.*

— IMC.VerifyEntry$(\mathsf{pk}^{\mathsf{sys}}, \mathsf{repnym}, \epsilon, msg) \to b$*: Takes as input a public system key* $\mathsf{pk}^{\mathsf{sys}}$*, a valid report pseudonym* repnym*, an entry* ϵ*, and a message msg, and returns* 1 *if and only if* ϵ *is a valid entry for the message msg under the report pseudonym* repnym*.*

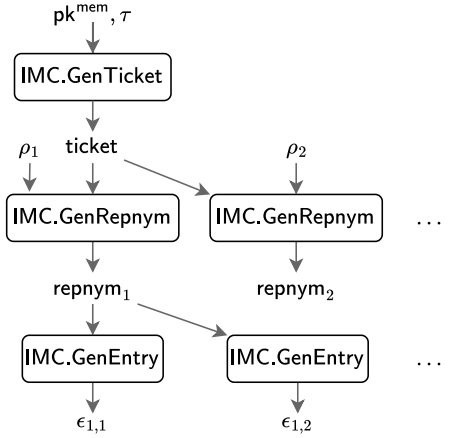

Figure 8: The IMC **interface enables credential issuance for** ES **and** RS**. When interacting with** ES**, a whistleblower, identified by a temporary member key** $\mathsf{pk}^{\mathsf{mem}}$ **and a nametag** τ**, obtains a ticket** ticket**. This** ticket **allows the whistleblower to derive multiple report pseudonyms** repnym**s, each uniquely tied to a report via a report tag** ρ**. Using a** repnym **and its associated authentication data, the whistleblower can authenticate and link multiple messages under the same pseudonym by creating a corresponding entry** ϵ **for each message.**

G.2 IMC Security Properties

The IMC satisfies the following security properties:

Whistleblower		**Organization**
Holds: $\mathsf{pk}^{\mathsf{sys}}_{Org}$ — Organization's IMC public key $\mathsf{pk}^{\mathsf{recv}}_{Org}$ — Organization's ALS public key τ — Whistleblower's identity tag		*Holds:* $\mathsf{sk}^{\mathsf{sys}}_{Org}, \mathsf{pk}^{\mathsf{sys}}_{Org}, \mathsf{CRS}^{\mathsf{IMC}}$ — key pair and CRS generated by $\mathsf{IMC.GenSysParams}(1^\lambda)$ $\mathsf{sk}^{\mathsf{recv}}_{Org}, \mathsf{pk}^{\mathsf{recv}}_{Org}$ — ALS key pair generated by $\mathsf{ALS.GenReceiverParams}(1^\lambda)$
$\mathsf{sk}^{\mathsf{mem}}_W, \mathsf{pk}^{\mathsf{mem}}_W \leftarrow \mathsf{IMC.GenMemberParams}(1^\lambda)$ $\mathsf{sk}^{\mathsf{recv}}_W, \mathsf{pk}^{\mathsf{recv}}_W \leftarrow \mathsf{ALS.GenReceiverParams}(1^\lambda)$ Send $\mathsf{msg}_1 \coloneqq \mathsf{initUser}, \mathsf{msg}_2 \coloneqq (\tau, \mathsf{pk}^{\mathsf{mem}}_W, \mathsf{pk}^{\mathsf{recv}}_W)$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{Org}$	$\rightarrow$	If τ not describing any person related to the organization: Send $\mathsf{msg}_1 \coloneqq \mathsf{initUserResponse}, \mathsf{msg}_2 \coloneqq$ "id not related" via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_W$ return $\mathsf{ticket} \leftarrow \mathsf{GenTicket}(\mathsf{sk}^{\mathsf{sys}}_{Org}, \mathsf{pk}^{\mathsf{mem}}_W, \tau)$
Receive $\mathsf{msg}_1 = \mathsf{initUserResponse}, \mathsf{msg}_2 = \mathsf{ticket}$ If $\mathsf{IMC.VerifyTicket}(\mathsf{pk}^{\mathsf{sys}}_{Org}, \mathsf{pk}^{\mathsf{mem}}_W, \mathsf{ticket}, \tau) \stackrel{?}{=} \mathsf{False}$: complain to auditor, return Save ticket	$\leftarrow$	Auth-Send $\mathsf{msg}_1 \coloneqq \mathsf{initUserResponse}, \mathsf{msg}_2 \coloneqq \mathsf{ticket}$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_W$

Figure 9: Protocol for obtaining a ticket during the whistleblower's initial contact with the organization. Upon a request from the whistleblower, the organization verifies identity of the whistleblower. If the declared identity indicates a person related to the organization, the organization issues a ticket for the declared identity.

Whistleblower		**CA**
Holds: τ — identity tag		*Holds:* $\mathsf{sk}^{\mathsf{Crt}}_{CA}, \mathsf{pk}^{\mathsf{Crt}}_{CA}$ — key pair generated by $\mathsf{Crt.KeyGen}(1^\lambda)$
$(\mathsf{sk}^{\mathsf{ent}}_W, \mathsf{pk}^{\mathsf{ent}}_W) \leftarrow\!\!\$\ \mathcal{R}$		
Request identity verification	$\xrightarrow{\tau, \mathsf{pk}^{\mathsf{ent}}_W}$	If τ is not correct identification credential of the whistleblower: inform the whistleblower about rejection, return
Save $\mathsf{cert}^{\mathsf{ent}}$	$\xleftarrow{\mathsf{cert}^{\mathsf{ent}}}$	$\mathsf{cert}^{\mathsf{ent}} \leftarrow \mathsf{Crt.Certify}_{\mathsf{sk}^{\mathsf{Crt}}_{CA}}(\{identity : \tau, key : \mathsf{pk}^{\mathsf{ent}}_W\})$

Figure 10: Procedure for acquiring an identity certificate from a Certificate Authority. The Certificate Authority is tasked with verifying that the applicant truly owns the claimed identity, in a manner comparable to the process used when issuing a qualified certificate [55].

P11: Correctness. We say that IMC ensures *correctness* if for every pair of tag τ and report identifier ρ chosen by a polynomial-time entity, and correctly generated system keys $(\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{sys}}) \leftarrow \mathsf{IMC.GenSysParams}(1^\lambda)$ and member keys $(\mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}}) \leftarrow \mathsf{IMC.GenMemberParams}(1^\lambda)$ we have the following. Given $\mathsf{ticket} \leftarrow \mathsf{IMC.GenTicket}(\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{mem}}, \tau)$, the procedure $\mathsf{IMC.VerifyTicket}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{ticket}, \tau)$ returns 1. Furthermore, given $\mathsf{repnym} \leftarrow \mathsf{IMC.GenRepnym}(\mathsf{sk}^{\mathsf{mem}}, \mathsf{ticket}, \rho)$, and a valid $\mathsf{cert}^{\mathsf{ent}}$ for the pair $(\tau, \mathsf{pk}^{\mathsf{mem}})$ issued by an authority holding a secret key $\mathsf{sk}^{\mathsf{ent}}$, and $(\mathsf{cert}^{\mathsf{repnym}}, s^\tau, s^\psi) \leftarrow \mathsf{IMC.CertifyRepnym}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{repnym}, \tau, \rho, \mathsf{ticket}, \mathsf{cert}^{\mathsf{ent}}, \mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}}, \mathsf{pol}^\tau, \mathsf{pol}^\psi)$, the procedure $\mathsf{IMC.VerifyRepnymCert}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{cert}}, \mathsf{repnym}, \rho, \mathsf{cert}^{\mathsf{repnym}}, \mathsf{pol}^\tau, \mathsf{pol}^\psi)$ returns 1. The verification procedures $\mathsf{IMC.VerifyNametagShare}(\mathsf{cert}^\tau, s^\tau)$ and $\mathsf{IMC.VerifyLinkabilityShare}(\mathsf{cert}^\psi, s^\psi)$ output 1 and the extraction procedure $\mathsf{IMC.ExtractNametag}(\mathsf{cert}^\tau, s^\tau)$ returns τ with probability 1, and $\mathsf{IMC.ExtractLinkability}(\mathsf{cert}^\psi, s^\psi)$ returns ψ with probability 1. Finally, given any message *msg* chosen by a polynomial-time entity, and $\epsilon \leftarrow \mathsf{IMC.GenEntry}(\mathsf{sk}^{\mathsf{mem}}, \mathsf{repnym}, \mathsf{ticket}, \rho, msg)$, the verification procedure $\mathsf{IMC.VerifyEntry}(\mathsf{repnym}, \epsilon, msg)$ returns 1.

P12: Ticket unforgeability. It is infeasible to forge a ticket that validates under the system's public key, without an access to the system's secret key, even if an adversary has seen a polynomial number of valid tickets and report pseudonyms.

P13: Report certificate unforgeability. It is infeasible to forge a report certificate that verifies under the system public key and the certificate authority public key $\mathsf{pk}^{\mathsf{ent}}$ without access to (i) a valid report pseudonym for a tag τ, (ii) a member public key $\mathsf{pk}^{\mathsf{mem}}$, (iii) a secret key $\mathsf{sk}^{\mathsf{ent}}$, and (iv) a valid certificate under $\mathsf{pk}^{\mathsf{cert}}$ for the same tag τ and public key $\mathsf{pk}^{\mathsf{ent}}$, where $(\mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}}) \in \mathcal{R}$, even if the adversary has observed a polynomial number of valid report certificates. Additionally, a certificate cannot be considered valid unless it includes (i) confirmation that cert^τ, together with a threshold number of shares, suffices to reconstruct the nametag, and (ii) confirmation that cert^ψ, together with a threshold number of shares, suffices to reconstruct the linkability trapdoor.

P14: Entry unforgeability. It is infeasible to forge an entry that validates under a valid report pseudonym repnym without an access to member's secret key $\mathsf{sk}^{\mathsf{mem}}$, even if an adversary has seen a polynomial number of valid entries for distinct messages.[23]

P15: Report pseudonym uniqueness. It is infeasible to find two distinct pairs (ticket, ρ) of a ticket and a report tag, for which

[23]This property aims to protect against anyone being able to reuse the report pseudonym without being a proper owner of the report pseudonym.

one is able to forge a single valid and certifiable report pseudonym repnym, given an access to an generating tickets oracle.

P16: Unlinkability of report pseudonyms. Given an access to the bindings and tags observed by the system, as well as certificates and signatures generated for distinct report pseudonyms, it is infeasible to link a single report pseudonym to a tag or report tag bound to the report pseudonym without an access to to the member's secret key or decryption shares associated with the report pseudonyms.

G.3 IMC Technical Overview

In this section, we present technical overview of a concrete instantiation of IMC. Our construction primarily relies on Non-Interactive Blind Signatures for Random Messages (NIBS) as introduced by Hanzlik [26], together with a generic framework based on Non-Interactive Zero-Knowledge proofs (NIZKs) and the existence of pseudorandom functions (PRFs) proposed by Baldimtsi et al. [3, 4].

The NIBS primitive enables an entity holding a asymmetric key pair $(\mathsf{sk}, \mathsf{pk})$ to obtain from a system authority a presignature σ on a random message m. The holder of sk can subsequently rerandomize both the presignature σ and the message m to obtain a new pair (σ', m') that verifies as having been issued by the system authority, while remaining *unlinkable* to the original pair (σ, m). A tagged extension of this primitive, referred to as Tagged-NIBS (TNIBS), allows the inclusion of a public tag τ in the signing process.

In this case, the rerandomized pair (σ', m') is valid only together with the corresponding tag τ. The default construction by Hanzlik is based on Structure-Preserving Signatures on Equivalence Classes [25] which are built on bilinear pairings. To avoid reliance on such assumptions, we instead adopt the (batched) generic construction of Baldimtsi et al. based on NIZKs and PRFs, which forms the foundation of our IMC design.

We briefly review the generic batched NIBS construction. The user's public key pk is defined as a commitment to a random PRF key K, while the secret key sk consists of K. To issue a presignature, the system samples a random string r, computes a signature on the tuple (pk, r), and returns both the signature and the nonce r to the user. In the batched variant, rerandomization proceeds as follows: to derive the i-th message m'_i, the user computes $m'_i = \mathsf{PRF}_K(r \,\|\, i)$. The corresponding signature is generated as a NIZK proof of knowledge of a valid presignature on (pk, r), such that pk commits to K and $\mathsf{PRF}_K(r \,\|\, i)$ is the pseudorandom function evaluation of $r \,\|\, i$ under key K.

Building on this framework, the IMC is instantiated as follows. The primary goal of the IMC is to allow a user to obtain a *ticket* that can later be transformed into a *report pseudonym* that is unlinkable to the original ticket, yet cannot be obtained without possession of a valid ticket. Concretely, we interpret the NIBS presignature (i.e., the system's signature on (pk, r)) as the ticket, and the rerandomized message m' as the report pseudonym.

Several modifications are required to satisfy additional requirements of the IMC. First, the user must be able to derive distinct pseudonyms for different report tags ρ. To this end, instead of evaluating the PRF on $r \,\|\, i$, the user evaluates it on $r \,\|\, \rho$ and adapts the NIZK statement to assert that the value was computed using a valid report tag ρ. The pseudorandomness of the PRF guarantees that, for a fixed ticket and report tag ρ, only a single pseudonym can be derived, and pseudonyms corresponding to different tags remain unlinkable.

Second, all pseudonyms must be inherently bound to a unique nametag τ. We achieve this requirement by transforming the generic construction into a tagged one as follows: the system is asked to issue a signature on (pk, τ, r) instead of on (pk, r), and the user is asked to augment the NIZK proof to certify that the signed message includes τ. Consequently, each derived pseudonym m' is cryptographically linked to the corresponding nametag.

Third, the nametag must be validated by a trusted certificate authority (CA). Given a certificate scheme Crt and a signature scheme Sig, the user generates an auxiliary key pair $(\mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}})$ and obtains from the CA a certificate on the pair $(\tau, \mathsf{pk}^{\mathsf{ent}})$. To produce a certificate for a pseudonym, the user signs pk using $\mathsf{sk}^{\mathsf{ent}}$ and incorporates into the NIZK statement a proof that the witness contains a signature verifiable under $\mathsf{pk}^{\mathsf{ent}}$, which itself was certified together with τ by the CA.

Fourth, we require that the nametag τ is decryptable from the certificate under a policy pol^{τ}, and that the linkability information ψ that allows to link distinct pseudonyms is decryptable under policy pol^{ψ}. We first set the randomness r as the linkability trapdoor ψ. Next, we encrypt the nametag and the trapdoor under their respective policies, and embed the resulting ciphertexts into the certificate. The access keys to the ciphertext are secret shared using respective policies. Next, a linkability verification tag $v^{\psi} = \mathsf{PRF}_r(\mathsf{ver_tag} \,\|\, r')$ is computed, given a fixed constant $\mathsf{ver_tag} \in \{0,1\}^{\lambda}$ and a fresh random string $r' \leftarrow\!\!\$\ \{0,1\}^{\lambda}$. The linkability verification tag along with the randomness r' are also added to the certificate. The NIZK statement must be extended to assert the correct computation of the ciphertexts, their shares and the linkability information. Whenever it is required to link all pseudonyms stemming from a single ticket, one can first extract r from the selected ciphertext, and then verify which of the linkability verification tags were computed using the same r.

Finally, to create a fresh entry on a message *msg*, the user is asked to sample a fresh randomness s, and then compute $\mathsf{PRF}_K(r \,\|\, s \,\|\, msg)$ and create a NIZK proof that the key K and the randomness r used to compute the value are consistent with the ticket. In our implementation, we set the decryption policies to contain a number of receivers and a threshold (n, t) and restrict the receivers to decrypt the respective tag only (or learn any information about the tag) if at least t of them explicitly consent for the decryption (even if they have already queried decryption of a polynomial number of tags in the IND-CCA sense).

G.4 IMC Implementation

In this section, we present details of our IMC construction.

Let $\mathcal{R}^{\mathsf{IMC}}$ denote an NP-hard relation describing all pairs (x, w), where for each statement $x = (\mathsf{repnym}, \rho, \mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{cert}}, c^{\tau}, c^{\psi}, \mathsf{pol}^{\tau}, \mathsf{pol}^{\psi}, vs^{\tau}, vs^{\psi}, r', v^{\psi})$ there exists a witness $w = (K, \tau, \sigma, r, \mathsf{cert}^{\mathsf{ent}}, \sigma_{\mathsf{cert}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{ent}})$ such that:

(1) $\mathsf{pk}^{\mathsf{mem}}$ is a commitment to K,
(2) σ is a signature on $(\mathsf{pk}^{\mathsf{mem}}, \tau, r)$ that validates under $\mathsf{pk}^{\mathsf{sys}}$,
(3) $\mathsf{repnym} = \mathsf{PRF}_K(r \,\|\, \rho)$,

(4) cert is a certificate for a pair $(\tau, \mathsf{pk}^{\mathsf{ent}})$ that validates under $\mathsf{pk}^{\mathsf{cert}}$,
(5) σ_{cert} is a signature on repnym that validates under $\mathsf{pk}^{\mathsf{ent}}$,
(6) the ciphertexts c^τ, c^ψ encrypt τ and the trapdoor r under the keys K^τ and K^ψ and the respective verification keys vs^τ, vs^ψ were correctly generated for the nametag and trapdoor $\tau, \psi = r$ under policies $\mathsf{pol}^\tau, \mathsf{pol}^\psi$,
(7) $v^\psi = \mathsf{PRF}_r(\mathsf{ver_tag} \,\|\, r')$.

Let $\mathcal{R}^{\mathsf{IMC}^*}$ denote an NP-hard relation describing all pairs (x, w), where for each statement $x = (msg, \mathsf{repnym}, \epsilon, s)$ there exists a witness $w = (K, r, \rho)$ such that:

(1) $\epsilon = \mathsf{PRF}_K(r \,\|\, s \,\|\, msg)$ and $\mathsf{repnym} = \mathsf{PRF}_K(r \,\|\, \rho)$.

The IMC consists of the following algorithms.

— $\mathsf{IMC.GenSysParams}(1^\lambda) \to (\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{sys}})$: Run $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{IMC}}}.\mathsf{Setup}(1^\lambda)$ to obtain $\mathsf{CRS}^{\mathsf{IMC}}$. $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{IMC}^*}}.\mathsf{Setup}(1^\lambda)$ to obtain $\mathsf{CRS}^{\mathsf{IMC}^*}$, and $\mathsf{Sig.KeyGen}(1^\lambda)$ to generate a signing key pair $(\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{sys}})$ for ticket issuance. Publish $\mathsf{CRS}^{\mathsf{IMC}}$, $\mathsf{CRS}^{\mathsf{IMC}^*}$ and output $(\mathsf{sk}^{\mathsf{sys}} \coloneqq \mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{sys}} \coloneqq \mathsf{pk}^{\mathsf{sys}})$.

— $\mathsf{IMC.GenMemberParams}(1^\lambda) \to (\mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}})$: Run $\mathsf{RandCom.Gen}(1^\lambda)$ to obtain a commitment c to a random key K. Set $\mathsf{pk}^{\mathsf{mem}} = c$, and set $\mathsf{sk}^{\mathsf{mem}} \coloneqq K$. Output $(\mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}})$.

— $\mathsf{IMC.GenTicket}(\mathsf{sk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{mem}}, \tau) \to \mathsf{ticket}$: Sample $r \leftarrow\!\!\$\ \{0,1\}^\lambda$, compute signature $\sigma \leftarrow \mathsf{Sig.Sign}_{\mathsf{sk}^{\mathsf{sys}}}(\mathsf{pk}^{\mathsf{mem}}, \tau, r)$. Output $\mathsf{ticket} \coloneqq (\sigma, r)$.

— $\mathsf{IMC.VerifyTicket}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{ticket}, \tau) \to b$: Parse ticket as (σ, r), iff σ verifies as a valid signature under the system's public key $\mathsf{pk}^{\mathsf{sys}}$ on $(\mathsf{pk}^{\mathsf{mem}}, \tau, r)$ set $b \coloneqq 1$ and $b \coloneqq 0$ otherwise. Output b.

— $\mathsf{IMC.GenRepnym}(\mathsf{sk}^{\mathsf{mem}}, \mathsf{ticket}, \rho) \to \mathsf{repnym}$: Parse ticket as (σ, r) and $\mathsf{sk}^{\mathsf{mem}}$ as K, compute $\mathsf{repnym} \coloneqq \mathsf{PRF}_K(r \,\|\, \rho)$. Output repnym.

— $\mathsf{IMC.CertifyRepnym}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{sk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{repnym}, \tau, \rho, \mathsf{ticket}, \mathsf{cert}^{\mathsf{ent}}, \mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}}, \mathsf{pol}^\tau = (n^\tau, t^\tau), \mathsf{pol}^\psi = (n^\psi, t^\psi)) \to (\mathsf{cert}^{\mathsf{repnym}}, \boldsymbol{s}^\tau, \boldsymbol{s}^\psi)$: Parse ticket as (σ, r). Given a certificate $\mathsf{cert}^{\mathsf{ent}}$ for a pair $(\tau, \mathsf{pk}^{\mathsf{ent}})$ such that $(\mathsf{sk}^{\mathsf{ent}}, \mathsf{pk}^{\mathsf{ent}})$ was generated by $\mathsf{Sig.KeyGen}(1^\lambda)$, compute signature $\sigma_{\mathsf{cert}} \leftarrow \mathsf{Sig.Sign}_{\mathsf{sk}^{\mathsf{ent}}}(\mathsf{repnym})$, generate random keys K^τ and K^ψ using $\mathsf{Sym.KeyGen}(1^\lambda)$. Sample random binary string $r' \leftarrow\!\!\$\ \{0,1\}^\lambda$, calculate linkability verification tag $v^\psi = \mathsf{PRF}_r(\mathsf{ver_tag} \,\|\, r')$, given a fixed constant ver_tag. Encrypt τ and the trapdoor r under the keys K^τ and K^ψ to obtain c^τ, c^ψ. Secret-share the keys under policies $\mathsf{pol}^\tau, \mathsf{pol}^\psi$, using VSS.Share, obtaining sets of shares $\boldsymbol{s}^\tau, \boldsymbol{s}^\psi$ (in the form (s_i, i)) and verification keys vs^τ, vs^ψ. Set $x \coloneqq (\mathsf{repnym}, \rho, \mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{cert}}, c^\tau, c^\psi, \mathsf{pol}^\tau, \mathsf{pol}^\psi, vs^\tau, vs^\psi, r', v^\psi)$, and $w \coloneqq (\mathsf{sk}^{\mathsf{mem}}, \tau, \sigma, r, \mathsf{cert}^{\mathsf{ent}}, \sigma_{\mathsf{cert}}, \mathsf{pk}^{\mathsf{mem}}, \mathsf{pk}^{\mathsf{ent}})$. Generate a NIZKPoK proof π, by running $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{IMC}}}.\mathsf{Prove}(\mathsf{CRS}^{\mathsf{IMC}}, x, w)$. Output $\mathsf{cert}^{\mathsf{repnym}} \coloneqq (\mathsf{cert}^\pi \coloneqq \pi, \mathsf{cert}^\tau \coloneqq (c^\tau, vs^\tau), \mathsf{cert}^\psi \coloneqq (c^\psi, vs^\psi, r', v^\psi), \boldsymbol{s}^\tau, \boldsymbol{s}^\psi)$.

— $\mathsf{IMC.VerifyRepnymCert}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{cert}}, \mathsf{repnym}, \rho, \mathsf{cert}^{\mathsf{repnym}}, \mathsf{pol}^\tau, \mathsf{pol}^\psi) \to b$: Parse $\mathsf{cert}^{\mathsf{repnym}}$ as $(\mathsf{cert}^\pi, \mathsf{cert}^\tau, \mathsf{cert}^\psi)$, where $\mathsf{cert}^\pi = \pi$, $\mathsf{cert}^\tau = (c^\tau, vs^\tau)$ and $\mathsf{cert}^\psi = (c^\psi, vs^\psi, r', v^\psi)$. Set $x \coloneqq (\mathsf{repnym}, \rho, \mathsf{pk}^{\mathsf{sys}}, \mathsf{pk}^{\mathsf{cert}}, c^\tau, c^\psi, \mathsf{pol}^\tau, \mathsf{pol}^\psi, vs^\tau, vs^\psi, r', v^\psi)$ and $b \coloneqq \mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{IMC}}}.\mathsf{Verify}(\mathsf{CRS}^{\mathsf{IMC}}, x, \pi)$. Output b.

— $\mathsf{IMC.VerifyNametagShare}(\mathsf{cert}^\tau, s^\tau) \to b$ (or $\mathsf{IMC.VerifyLinkabilityShare}(\mathsf{cert}^\psi, s^\psi) \to b$): Derive vs^τ (or vs^ψ) from cert^τ (or cert^ψ), parse s^τ (or s^ψ) as (s_i, i) and set $b \coloneqq \mathsf{VSS.Ver}(s_i, vs^\tau, i)$ (or $\mathsf{VSS.Ver}(s_i, vs^\psi, i)$). Output b.

— $\mathsf{IMC.ExtractNametag}(\mathsf{cert}^{\mathsf{repnym}}, \boldsymbol{s}^\tau) \to \tau$: Derive the ciphertext c^τ from $\mathsf{cert}^{\mathsf{repnym}}$, then $K^\tau \leftarrow \mathsf{VSS.Recon}(\boldsymbol{s}^\tau)$. Use K^τ and c^τ to decrypt and output τ.

— $\mathsf{IMC.ExtractLinkability}(\mathsf{cert}^{\mathsf{repnym}}, \boldsymbol{s}^\psi) \to \psi$: Derive the ciphertext c^ψ from $\mathsf{cert}^{\mathsf{repnym}}$, then $K^\psi \leftarrow \mathsf{VSS.Recon}(\boldsymbol{s}^\psi)$. Use K^ψ and c^ψ to decrypt and output ψ.

— $\mathsf{IMC.GenEntry}(\mathsf{sk}^{\mathsf{mem}}, \mathsf{repnym}, \mathsf{ticket}, \rho, msg) \to \epsilon$: Parse ticket as $(_, r)$. Sample a random seed $s \leftarrow \{0,1\}^\lambda$, create a signature $\sigma = \mathsf{PRF}_K(r \,\|\, s \,\|\, msg)$. Set $x \coloneqq (msg, \mathsf{repnym}, \sigma, s), w \coloneqq (\mathsf{sk}^{\mathsf{mem}}, r, \rho)$. Generate a NIZKPoK proof π^{ent}, by running $\mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{IMC}^*}}.\mathsf{Prove}(\mathsf{CRS}^{\mathsf{IMC}}, x, w)$. Output $\epsilon \coloneqq (\sigma, \pi^{\mathsf{ent}}, s)$.

— $\mathsf{IMC.VerifyEntry}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{repnym}, \epsilon, msg) \to b$. Parse ϵ as $(\sigma, \pi^{\mathsf{ent}}, s)$, set $x \coloneqq (msg, \mathsf{repnym}, \sigma, s)$ and $b \coloneqq \mathsf{NIZKPoK}_{\mathcal{R}^{\mathsf{IMC}^*}}.\mathsf{Verify}(\mathsf{CRS}^{\mathsf{IMC}^*}, x, \pi)$. Output b.

G.5 IMC Security Analysis

Lemma 2. *The construction presented in the Section G.4 satisfies all Identity Management Component security properties P11–P16.*

Arg. In the analysis below, we assume that the execution of the setup procedure is trusted. It is easy to see that the scheme satisfies P11: Correctness. The P12: Ticket unforgeability is ensured by unforgeability of the signature scheme, specifically the signature $\mathsf{Sig.Sign}_{\mathsf{sk}^{\mathsf{sys}}}(\mathsf{pk}^{\mathsf{mem}}, \tau, r)$. P13: Report certificate unforgeability comes from the soundness of the NIZKPoK system and the relation $\mathcal{R}^{\mathsf{IMC}}$ that enforces existence of a valid ticket and a certificate for a nametag associated with the ticket as well as a key that is used to sign the repnym. Additionally the conditional extractability (i.e., the possibility to receive it only with an access to the threshold of their shares) of nametag and the linkability is ensured by the privacy and verifiability properties of the verifiable secret sharing scheme (see Definition 21) and the soundness of the NIZKPoK system and the relation $\mathcal{R}^{\mathsf{IMC}}$ that enforces generation of a valid verification keys for the shares. Additionally, the indistinguishability property of the pseudorandom function and the usage of additional randomness r' in the construction ensure that without collection of shares any two report pseudonyms are not linkable. By the deterministic nature of the pseudorandom function, the soundness of the NIZKPoK system and the relation $\mathcal{R}^{\mathsf{IMC}}$, given the linkability trapdoor, all pseudonyms created using the same ticket are linkable. For the next two properties, we assume that the key and ticket used to generate the pseudonym are kept away from the access of the adversary. Thus, the P14: Entry unforgeability and P15: Report pseudonym uniqueness come from the indistinguishability of the pseudorandom function as well as soundness of the NIZKPoK system and the relation $\mathcal{R}^{\mathsf{IMC}^*}$ which enforces the usage of the same secret key and ticket that was used to create the repnym. P16: Unlinkability of report pseudonyms is ensured by the indistinguishability property of the pseudorandom function. □

H System Design

Building on the components introduced in the previous sections, we now present the overall system design. This design is structured to satisfy the system and design goals outlined earlier.

H.1 System Description

Notation. Since the system involves multiple entities holding values of similar types but with different roles and usages, we introduce the following unified notation. We denote values as

$$\mathrm{type}^{\mathrm{usage}}_{\mathrm{holder,\,index}},$$

where type specifies the kind of value, usage indicates its intended purpose or role within the protocol, holder identifies the entity to whom the value belongs, and index distinguishes multiple values of the same type and usage held by the same entity. Where this is unnecessary we omit writing holder and index.

Furthermore, in the system description below, as well as in Figures 11 and 12, we use the following shorthand notation to describe protocol executions. Let the auditor hold $(\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}})$ generated by ALS.AuditorSetup(1^λ), and let the receiver hold $(\mathsf{sk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{recv}})$ generated by ALS.Setup(1^λ). Suppose a sender wishes to transmit messages msg_1 and msg_2 to the receiver identified by $\mathsf{pk}^{\mathsf{recv}}$ using ALS. We write

$$\text{Send } \mathsf{msg}_1, \mathsf{msg}_2 \text{ via ALS.Send to } \mathsf{pk}^{\mathsf{recv}}$$

to denote that the sender executes

$$(\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2}) \leftarrow \text{ALS.Send}(\mathsf{pk}^{\mathsf{recv}}, \mathsf{pk}^{\mathsf{audit}}, \mathsf{msg}_1, \mathsf{msg}_2).$$

Correspondingly, we write

$$\text{Receive } \mathsf{msg}_1, \mathsf{msg}_2$$

to denote that the receiver performs the following sequence of operations:

$$\langle \ldots, \mathsf{id}^{\mathsf{mr}}, \ldots \rangle \leftarrow \text{Check}(\mathsf{sk}^{\mathsf{recv}}),$$

$$(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{t}^{\mathsf{msg}_2}) \leftarrow \text{ALS.Receive}(\mathsf{sk}^{\mathsf{recv}}, \mathsf{id}^{\mathsf{mr}}).$$

We use analogous notation for executions of ALS.SendConf and ALS.ReceiveConf, as well as for logging operations via ALS.Log.

Moreover, we introduce notation for sending and logging authenticated messages. Let the sender additionally hold $(\mathsf{sk}^{\mathsf{sig}}, \mathsf{pk}^{\mathsf{sig}}) \leftarrow \text{Sig.KeyGen}(1^\lambda)$. We write

$$\text{Auth-Send } \mathsf{msg}_1, \mathsf{msg}_2 \text{ via ALS.Send to } \mathsf{pk}^{\mathsf{recv}}$$

to denote that the sender creates authenticated messages msg'_1 and msg'_2 by attaching signatures on msg_1 and msg_2, respectively, together with the public key $\mathsf{pk}^{\mathsf{sig}}$, and then executes

$$\text{Send } \mathsf{msg}'_1, \mathsf{msg}'_2 \text{ via ALS.Send to } \mathsf{pk}^{\mathsf{recv}}.$$

In a corresponding line, we write Receive $\mathsf{msg}'_1, \mathsf{msg}'_2$ to denote that the receiver verifies the included signatures using $\mathsf{pk}^{\mathsf{sig}}$ upon receipt. We use a similar notation to send and receive authenticated messages via ALS.SendConf and ALS.ReceiveConf and to log authenticated messages via ALS.Log.

About Receipt Confirmation. In the system design presented below, communication between entities, except the internal RS communication, relies on the Send procedure. In deployments that require stronger delivery guarantees or additional interaction assurance, the SendConf procedure can be used as a drop-in replacement.

H.1.1 System Setup. The system setup consists of a set of procedures executed once by the participating entities before the WS can be used to submit reports. This phase initializes ALS, IMC, and the certification scheme Crt.

The auditor runs ALS.AuditorSetup(1^λ) to obtain $(\mathsf{sk}^{\mathsf{audit}}, \mathsf{pk}^{\mathsf{audit}})$ and publishes $\mathsf{pk}^{\mathsf{audit}}$. ALS provider runs $(\mathsf{sparams}, \mathsf{pparams}, \mathsf{state}) \leftarrow \text{ALS.Setup}(1^\lambda, \mathsf{pk}^{\mathsf{audit}})$ to initialize an ALS instance. As a result of the procedure, the secret parameters sparams are shared among the ALS provider's infrastructure and the public parameters pparams and public part of state (i.e., ALS.PublicView) are made publicly available.

Each managing operator MO_i and organization representative Org run ALS.GenReceiverParams(1^λ) to obtain $(\mathsf{sk}^{\mathsf{recv}}_{MO_i}, \mathsf{pk}^{\mathsf{recv}}_{MO_i})$ and $(\mathsf{sk}^{\mathsf{recv}}_{Org}, \mathsf{pk}^{\mathsf{recv}}_{Org})$, respectively. Additionally, Org runs IMC.GenSysParams(1^λ) to obtain $(\mathsf{sk}^{\mathsf{sys}}_{Org}, \mathsf{pk}^{\mathsf{sys}}_{Org})$ along with CRS values needed to run the system. To ensure that the system parameters are generated honestly, the organization representative delegates generation of the CRS to a trusted environment, e.g., governed by the managing operators. Each MO_i publishes $\mathsf{pk}^{\mathsf{recv}}_{MO_i}$, while Org publishes $\mathsf{pk}^{\mathsf{recv}}_{Org}$, $\mathsf{pk}^{\mathsf{sys}}_{Org}$ and the CRS is published by the trusted environment. The managing operators jointly generate a shared broadcast key by executing ALS.GenReceiverParams(1^λ) as a multi-party computation protocol, obtaining the key pair $(\widehat{\mathsf{sk}^{\mathsf{recv}}_{\mathbf{MO}}}, \widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}})$, they publish $\widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}}$.

Additionally, each operator O_i runs ALS.GenReceiverParams(1^λ) to obtain $(\mathsf{sk}^{\mathsf{recv}}_{O_i}, \mathsf{pk}^{\mathsf{recv}}_{O_i})$ and sends $\mathsf{pk}^{\mathsf{recv}}_{O_i}$ to the managing operators. The certification authority CA runs Crt.KeyGen(1^λ) to obtain $(\mathsf{sk}^{\mathsf{crt}}_{CA}, \mathsf{pk}^{\mathsf{crt}}_{CA})$ and publishes $\mathsf{pk}^{\mathsf{crt}}_{CA}$.

H.1.2 Enrollment Phase. At the first contact between the whistleblower W and Org, W obtains a ticket that will later allow them to authorize their reports. The details of this phase are depicted in the Figure 9.

W runs ALS.GenReceiverParams(1^λ) and IMC.GenMemberParams(1^λ) to obtain (one-time introductory) $(\mathsf{sk}^{\mathsf{recv}}_{W,\mathrm{int}}, \mathsf{pk}^{\mathsf{recv}}_{W,\mathrm{int}})$ and $(\mathsf{sk}^{\mathsf{mem}}_{W}, \mathsf{pk}^{\mathsf{mem}}_{W})$, respectively. Then W selects a tag τ containing information uniquely identifying the person[24] and sends necessary information to Org, by sending $\mathsf{msg}_1 \coloneqq \mathsf{initUser}, \mathsf{msg}_2 \coloneqq (\tau, \mathsf{pk}^{\mathsf{mem}}_{W}, \mathsf{pk}^{\mathsf{recv}}_{W,\mathrm{int}})$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{Org}$.

Org receives the messages $\mathsf{msg}_1, \mathsf{msg}_2$ and checks if τ corresponds to some person related to the organization. If the verification output is false, then Org returns an error message to W, by Auth-Send $\mathsf{msg}_1 \coloneqq \mathsf{initUserResponse}, \mathsf{msg}_2 \coloneqq (\mathrm{error})$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{W,\mathrm{int}}$, where msg_2 contains an error message indicating that the identity validation gave negative result.

If the output is correct, then Org runs ticket $\leftarrow \text{GenTicket}(\mathsf{sk}^{\mathsf{sys}}_{Org}, \mathsf{pk}^{\mathsf{mem}}_{W}, \tau)$ and returns message to W by Auth-Send $\mathsf{msg}_1 \coloneqq \mathsf{initUserResponse}, \mathsf{msg}_2 \coloneqq (\mathrm{ticket})$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{W,\mathrm{int}}$.

[24] Such as person's national identification number and full name. What data uniquely identifies a person should be established by each Member State.

W receives the messages and runs verification procedure $\mathsf{IMC.VerifyTicket}(\mathsf{pk}^{\mathsf{sys}}_{Org}, \mathsf{pk}^{\mathsf{mem}}_{W}, \mathsf{ticket}, \tau)$. If the output of the verification procedure is negative, then W uses the transcript of the communication with Org as evidence to complain to the auditor.

At any point prior to submitting a report, W must contact a CA to obtain a certificate $\mathsf{cert}^{\mathsf{ent}}$ for the nametag τ and the public key $\mathsf{pk}^{\mathsf{ent}}_{W}$, where $(\mathsf{sk}^{\mathsf{ent}}_{W}, \mathsf{pk}^{\mathsf{ent}}_{W})$ is generated by $\mathsf{Sig.KeyGen}(1^{\lambda})$ (see Figure 10). Note that the communication between W and CA does not take place through ALS.

H.1.3 Create Phase. The create phase is depicted in Figure 11.

Create and send report. This is the first procedure of the initial processing phase, in which W creates a report and submits it to the RS.

W starts with generating a fresh pair of ALS keys for communication $(\mathsf{sk}^{\mathsf{recv}}_{W}, \mathsf{pk}^{\mathsf{recv}}_{W}) \leftarrow \mathsf{ALS.GenReceiverParams}(1^{\lambda})$. W chooses a report tag ρ containing correctly formatted category of the report and a freshly generated random binary string of length λ,[25] and runs $\mathsf{IMC.GenRepnym}(\mathsf{sk}^{\mathsf{mem}}_{W}, \mathsf{ticket}, \rho)$ to obtain report pseudonym repnym. Next, given the certificate $\mathsf{cert}^{\mathsf{ent}}$ for $(\tau, \mathsf{pk}^{\mathsf{ent}}_{W})$ and the policies $\mathsf{pol}^{\mathsf{rep}} = (n, t^{\mathsf{rep}})$, $\mathsf{pol}^{\tau} = (n, t^{\tau})$, and $\mathsf{pol}^{\psi} = (n, t^{\psi})$, where n denotes the total number of managing operators, the thresholds t^{rep}, t^{τ}, and t^{ψ} specify, respectively, the number of managing operators required to reconstruct the report contents, disclose the identity, and allow linking all repnyms originating from the same whistleblower. The whistleblower runs $\mathsf{IMC.CertifyRepnym}(\mathsf{pk}^{\mathsf{sys}}_{Org}, \mathsf{sk}^{\mathsf{mem}}_{W}, \mathsf{pk}^{\mathsf{mem}}_{W}, \mathsf{repnym}, \tau, \rho, \mathsf{ticket}, \mathsf{cert}^{\mathsf{ent}}, \mathsf{sk}^{\mathsf{ent}}_{W}, \mathsf{pk}^{\mathsf{ent}}_{W}, \mathsf{pol}^{\tau}, \mathsf{pol}^{\psi})$ to obtain certificates $\mathsf{cert}^{\mathsf{repnym}} = (\mathsf{cert}^{\pi}, \mathsf{cert}^{\tau}, \mathsf{cert}^{\psi})$, and sets of decryption key shares $\boldsymbol{s}^{\tau}, \boldsymbol{s}^{\psi}$. Next, W constructs a triple $(c^{\mathsf{rep}}, \boldsymbol{s}^{\mathsf{rep}}, vs^{\mathsf{rep}})$ consisting of the ciphertext of the report content, set of shares of the content, and the verification key for those shares. Then, W transmits the certificates, report ciphertext, report verification key, report identifier, and their response ALS public key to the managing operators' group address by Send $\mathsf{msg}_1 \coloneqq (\mathsf{sendRepPrep}, \mathsf{repnym})$ and $\mathsf{msg}_2 \coloneqq (\mathsf{cert}^{\tau}, \mathsf{cert}^{\psi}, c^{\mathsf{rep}}, vs^{\mathsf{rep}}, \mathsf{cert}^{\pi}, \rho, \mathsf{pk}^{\mathsf{recv}}_{W})$ via ALS.Send to $\widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}}$. Δ^{ack} timer starts.[26]

After that, W transmits by Send $\mathsf{msg}_1 \coloneqq (\mathsf{sendRep}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\boldsymbol{s}^{\tau}[i], \boldsymbol{s}^{\psi}[i], \boldsymbol{s}^{\mathsf{rep}}[i], \epsilon_i)$ via ALS.Send to each $\mathsf{pk}^{\mathsf{recv}}_{MO_i}$, where ϵ_i is computed over the concatenated content of msg_1 and msg_2, i.e., $\mathsf{IMC.GenEntry}(\mathsf{sk}^{\mathsf{mem}}_{W}, \mathsf{repnym}, \mathsf{ticket}, \rho, \mathsf{msg}_1 \| \mathsf{msg}_2)$.

Receive report. Each MO_i first receives a message sent to $\widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}}$ containing the certificates, report ciphertext, report verification key, report identifier, and the whistleblower's response key and they start timer δ^{lag}. [27] If the message is not received before the expiration of δ^{lag}, MO_i initiates audit.

Each MO_i then proceeds with verification. First, ϵ_i is verified using IMC.VerifyEntry with respect to the system key $\mathsf{pk}^{\mathsf{sys}}$, pseudonym repnym, and the the direct message to this MO_i. Next, repnym and its associated certification data are verified against $\mathsf{pk}^{\mathsf{sys}}_{Org}$ and $\mathsf{pk}^{\mathsf{ent}}_{CA}$, as well as the policies pol^{τ} and pol^{ψ}, using the IMC.VerifyRepnymCert function. Furthermore, each share is verified against its corresponding certificate or verification key using IMC.VerifyNametagShare, IMC.VerifyLinkabilityShare, and VSS.Ver. Finally, each MO_i verifies that the received report tag ρ was not already used in the system. If any verification step fails, the managing operator Auth-Logs this event via ALS.Log, informs the other operators, and rejects the report by triggering the rejectRep action. Otherwise, MO_i broadcasts its vote via ALS and waits δ^{sync} for votes from the other managing operators. If at least one vote is negative, MO_i Auth-Logs rejectRep and audit is initiated.[28]

Send acknowledgment. Each MO_i saves the collected report shares for future reference in the following form $\mathsf{repnym} : (\mathsf{cert}^{\tau}, vs^{\tau}, \mathsf{cert}^{\psi}, vs^{\psi}, c^{\mathsf{rep}}, s^{\mathsf{rep}}_{i}, vs^{\mathsf{rep}}, i, \mathsf{pk}^{\mathsf{recv}}_{W})$ and returns message to W by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{sendAck}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\perp)$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{W}$. At the moment of sending the first message, the feedback timer Δ^{feed} starts.

W awaits responses; if, by the time Δ^{ack}, it receives n consistent sendAck messages (whether approvals or rejections), or if Δ^{ack} expires and all received sendAck messages are consistent, W continues; otherwise, W may initiate an audit. [29]

Assign operator. Each MO_i derives $O_j \leftarrow f^{\mathsf{assignOp}}(\mathsf{repnym}, \rho)$ and then transfers their share of the report data, the share verification key, the report ciphertext, and the W's response key[30] to O_j by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{assignOp}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (s^{\mathsf{rep}}_{i}, vs^{\mathsf{rep}}, c^{\mathsf{rep}}, \mathsf{pk}^{\mathsf{recv}}_{W})$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{O_j}$.

O_j waits to receive t tuples with the same ciphertext and verification key, if any of the shares is inconsistent, O_j Auth-Logs a complaint and initiates an audit. Otherwise, O_j reconstructs the key, decrypts the report content, and proceeds with handling of the report.

H.1.4 Handle Phase. In the handle phase, O_j performs the necessary human-driven handling of the report, i.e., the investigation. This process cannot be controlled by the RS itself and lies beyond its scope; it is therefore left unspecified. However, requiring O_j to oversee the report's progression by performing certain actions within the RS makes it possible to introduce certain audit guarantees into the system. These actions include logging the state of the

[25] The randomness included in the report tag is sufficient to guarantee that the pseudonym repnym is unique for each report. However, it is not sufficient to limit the amount of reports that can be submitted by a single whistleblower to the system and thus enhance denial-of-service attacks protection in the system. To limit the number of reports that can be submitted on an organization to l, one can ask the whistleblower to include in each report tag a value $\mathsf{PRF}_{\mathsf{sk}^{\mathsf{mem}}}(i)$, with $i \in \{1, \ldots, l\}$, and provide a NIZK proof that the value was computed on a value $i \in \{1, \ldots, l\}$ using a key attested by the organization.

[26] The Directive explicitly states that acknowledgment must be sent within a specified period *of receipt*. Note that this requirement can be directly satisfied using SendConf. However, we abstract away from this detail and instead measure the acknowledgment deadline from the time the report is sent.

[27] Note that, due to possible parallel access to ALS and an unknown sender, the notion of a "session" between a whistleblower and the Reporting System is defined by repnym and is authorized by ϵ.

[28] See Section H.2 for a description of the two audit modes available to the auditor. Depending on the severity of the case, both modes enable an affected entity to obtain verifiable evidence of their actions (i.e., submitted inputs), which may be used in legal proceedings.

[29] The appearance of sendAck message records in ALS.PublicView constitutes receiving acknowledgment in the sense of the Directive.

[30] Note that all fields, except the individual shares, are sent redundantly. If only a single managing operator were responsible for sending these fields, they could silently replace some of them. Requiring all managing operators to send these values ensures that any inconsistency can be immediately detected by the operator.



Figure 11: Overview of the create phase in the report workflow. After successful enrollment, a whistleblower can create and submit a report. Managing operators are responsible for receiving and validating the submission. If validation fails, the report is rejected; otherwise, managing operators sent the acknowledgment to whistleblower, and collectively assign an operator responsible for the subsequent handle phase.

investigation (through ALS.Log), making decisions regarding competence and priority, and managing follow-up actions associated with the report, such as transferring the report, requesting further information, disclosing identity, and concluding the report.

Decide competence and choose priority. O_j Auth-Logs a message via ALS.Log of the form $\text{msg}_1 \coloneqq$ (decideAndChoosePriority, repnym), $\text{msg}_2 \coloneqq$ (justMsg), where justMsg provides the rationale for the decision.[31]

Request identity disclosure. O_j broadcasts an identity disclosure request to all managing operators by Auth-Send $\text{msg}_1 \coloneqq$ (reqIDDisclsoure, repnym), $\text{msg}_2 \coloneqq$ (justMsg, $\text{pk}^{\text{recv}}_{\text{E}}$) via ALS.Send to $\widehat{\text{pk}^{\text{recv}}_{\textbf{MO}}}$, where justMsg contains the justification for

[31] decideAndChoosePriority serves as a connector following each action to capture the impact of the latest results (discoveries) on the state of the report. If the latest results are not significant for the investigation, logging this action may be omitted.

identity disclosure as well as indication whether the request concerns disclosure of the identity tag or the linkability information (in the latter case, the only allowed recipient is the selected operator of the report), and $\mathsf{pk}_{\mathsf{E}}^{\mathsf{recv}}$ is the public key of the intended receiver.[32] At the moment of sending the request the δ^{sync} timer starts.

Decide identity disclosure request. Each MO_i validates justMsg of the identity disclosure request with respect to $f^{\mathsf{idDiscReq}}$ and then broadcasts its decision by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{decideIdentityDisclosure}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\mathsf{pk}_{\mathsf{E}}^{\mathsf{recv}}, \mathsf{outMsg})$ via ALS.Send to $\widehat{\mathsf{pk}_{\mathbf{MO}}^{\mathsf{recv}}}$, where outMsg represents the managing operator's evaluation of the request. If a managing operator evaluates the request negatively, they broadcast a negative vote; otherwise, they broadcast a positive vote.

Reject identity disclosure. If O_j receive at least one rejection before δ^{sync} expires they proceed with rejection. If O_j believes that the rejection is a malicious attempt to stop deanonymization, they initiate an audit. O_j transmits via Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{rejectIDDisclosure}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\mathsf{justMsg})$ to $\mathsf{pk}_{O_j}^{\mathsf{recv}}$, where justMsg contains the reason for rejecting identity disclosure.

Send identity disclosure. If after δ^{sync} all votes are positive, each MO_i transmits by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{sendIDDisclosure}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (c^{\tau}, s_i^{\tau}, vs^{\tau})$ or $\mathsf{msg}_2 \coloneqq (c^{\psi}, s_i^{\psi}, vs^{\psi})$ via ALS.Send to $\mathsf{pk}_{O_j}^{\mathsf{recv}}$, where msg_2 contains MO_i's shares of the identity tag or the linkability information attached to the report, depending on the indication included in the identity disclosure request. Furthermore, MO_i informs W about disclosure through *Send feedback*.

Receive identity disclosure. If the receiver entity E does not receive at least t consistent tuples before δ^{sync} expires they initiate and audit. Otherwise, after shares are collected, E receives the identity. If the identity is received as linkability information ψ, the selected operator can verify which of the certificates of pseudonyms cert^{ψ} are conformant with ψ (for more details, see Section G.3). If any inconsistent tuples are received, E initiates an audit.

Request report (partial) transfer. O_j broadcasts a report transfer request to all managing operators by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{reqTransRep}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\mathsf{justMsg}, \mathsf{pk}_{\mathsf{E}}^{\mathsf{recv}})$ via ALS.Send to $\widehat{\mathsf{pk}_{\mathbf{MO}}^{\mathsf{recv}}}$, where justMsg contains the justification for report (partial) transfer and $\mathsf{pk}_{\mathsf{E}}^{\mathsf{recv}}$ is the public key of the intended receiver E. At the moment of sending the request the δ^{sync} timer starts.

Decide report (partial) transfer request. Each MO_i validates the justification justMsg of the report transfer request with respect to f^{transReq} and then broadcasts its decision by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{decideTransferReport}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\mathsf{pk}_{\mathsf{E}}^{\mathsf{recv}}, \mathsf{outMsg})$ via ALS.Send to $\widehat{\mathsf{pk}_{\mathbf{MO}}^{\mathsf{recv}}}$, where outMsg represents MO_i's evaluation of the request. If at least one MO_i decision is negative before δ^{sync} expires O_j aborts (partial) report transfer and Auth-Logs $\mathsf{msg}_1 \coloneqq (\mathsf{decideAndChoosePriority}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\perp)$. If O_j believes that the rejection is a malicious attempt to stop report (partial) transfer, they initiate an audit.

Transfer Report. If the decision of the managing operators is positive, O_j collects (or a selected/requested subset in the case of a partial transfer) history of the report repHist and transmits it by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{transRep}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\mathsf{repHist})$ via ALS.Send to $\mathsf{pk}_{\mathsf{E}}^{\mathsf{recv}}$. [33] In justified cases,[34] O_j additionally informs W about report transfer through *Send feedback*.

Receive transferred report. The receiver entity E receives the report data from ALS.

Send feedback. O_j transmits by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{sendFeed}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\mathsf{feedMsg})$ via ALS.Send to $\mathsf{pk}_{W}^{\mathsf{recv}}$, where feedMsg is prepared feedback message to the W.

Receive feedback. If W does not see the response on ALS.PublicView[35] before Δ^{feed} expires, they initiate audit, otherwise W receives feedMsg from ALS.

Send further information request. O_j transmits a further information request to the W via Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{sendFurInfoReq}, \mathsf{repnym})$ and $\mathsf{msg}_2 \coloneqq (\mathsf{reqMsg}, \mathsf{pk}_{O_j}^{\mathsf{recv}})$ via ALS.Send to $\mathsf{pk}_{W}^{\mathsf{recv}}$, where reqMsg contains the justification for the request, O_j's questions, etc., and $\mathsf{pk}_{O_j}^{\mathsf{recv}}$ allows the W to respond directly to O_j. At the moment of sending the message, the further information timer Δ^{info} starts.

Receive further information request & send further information. The whistleblower receives feedMsg from ALS. If they cannot or do not want to answer they abort, otherwise respond by Send $\mathsf{msg}_1 \coloneqq (\mathsf{sendFurInfo}, \mathsf{repnym})$ and $\mathsf{msg}_2 \coloneqq (\mathsf{furInfMsg}, \epsilon)$ via ALS.Send to $\mathsf{pk}_{W}^{\mathsf{recv}}$, where $\epsilon \leftarrow \mathsf{IMC.GenEntry}(\mathsf{sk}^{\mathsf{mem}}, \mathsf{repnym}, \mathsf{ticket}, \rho, \mathsf{furInfMsg})$.

Receive further information. If O_j does not see the response to their request on ALS.PublicView before Δ^{info} expires, they log by Auth-Log $\mathsf{msg}_1 \coloneqq (\mathsf{receiveFurInfo}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq$ ("no response") via ALS.Log. Otherwise, O_j receives furInfMsg and ϵ, then verifies $\mathsf{IMC.VerifyEntry}(\mathsf{pk}^{\mathsf{sys}}, \mathsf{repnym}, \epsilon, \mathsf{furInfMsg})$. If output of the verification is negative, O_j logs by Auth-Log $\mathsf{msg}_1 \coloneqq (\mathsf{receiveFurInfo}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq$ ("entry verification failed") via ALS.Log. If the verification procedure was successful, the operator logs by Auth-Log $\mathsf{msg}_1 \coloneqq (\mathsf{receiveFurInfo}, \mathsf{repnym}), \mathsf{msg}_2 \coloneqq (\perp)$ via ALS.Log.

Read report. The content of the report is made available to an authorized entity by being sent to them via ALS during *Assign operator*.

Write to report. O_j appends content to the report by Auth-Log via ALS.Log.

[32] Note that the operator themselves may be the intended receiver.

[33] When a report is transferred to another instance of the RS, $\mathsf{pk}_{\mathsf{E}}^{\mathsf{recv}}$ is set to the broadcast key of the target instance's managing operators. The operator then verifiably secret-shares repHist among these target operators, effectively replicating the report component of the whistleblower submission process. Linkability shares and identity-disclosure shares, however, remain with the originating RS.

[34] [Art.11(6)] implies that in the case of a report transfer due to lack of competence, the whistleblower must be informed, whereas in the case of a transfer for investigative purposes, this is not required.

[35] The whistleblower is expected to periodically monitor ALS.

Operator		Managing Operators		Operator/External Entity
Holds: $\widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}}$ – **MO**'s ALS broadcast public key repnym – Report pseudonym $\mathsf{pk}^{\mathsf{recv}}_{\mathsf{E}}$ – Entity's ALS receiver public key justMsg – Message containing justification for identity disclosure		*Each $MO_i \in MO$ holds:* $\widehat{\mathsf{sk}^{\mathsf{recv}}_{\mathbf{MO}}}$ – **MO**'s ALS broadcast private key repnym $\rightarrow$ $\{\mathsf{cert}^{\tau}, s_i^{\tau}, \mathsf{cert}^{\psi}, s_i^{\psi}, c^{\mathsf{rep}}, s_i^{\mathsf{rep}}, vs^{\mathsf{rep}}, i, \mathsf{pk}^{\mathsf{recv}}_{W}\}$) – repnym to values mapping saved during create phase		*Holds:* $\mathsf{sk}^{\mathsf{recv}}_{\mathsf{E}}$ – Entity's ALS receiver private key $\mathsf{pol}^{\tau} = (n^{\tau}, t^{\tau})$ – Identity disclosure policy
Auth-Send msg_1 = (reqIDDisclsoure, repnym), msg_2 = (justMsg, $\mathsf{pk}^{\mathsf{recv}}_{\mathsf{E}}$) via ALS.Send to $\widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}}$	$\rightarrow$	*Each MO_i*: Receive msg_1 = (reqIDDisclsoure, repnym) msg_2 = (justMsg, $\mathsf{pk}^{\mathsf{recv}}_{\mathsf{E}}$) Auth-Log (decideIdentityDisclosure, repnym, $\mathsf{pk}^{\mathsf{recv}}_{\mathsf{E}}$, outMsg) via ALS.Log If justMsg is not satisfactory for ID disclosure to $\mathsf{pk}^{\mathsf{recv}}_{\mathsf{E}}$ with respect to $\mathcal{J}^{\mathsf{idDiscReq}}$: Proceed to `rejectedIDDisclosure`; return Inform whistleblower about identity disclosure via `sentFeed` Auth-Send msg_1 = (sendIDDisclosure, repnym), msg_2 = $(\mathsf{cert}^{\tau}, s_i^{\tau})$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{\mathsf{E}}$	$\rightarrow$	Receive msg_1 = (sendIDDisclosure, repnym), msg_2 = $(\mathsf{cert}^{\tau}, s_i^{\tau})$ $\mathsf{IMC.VerifyNametagShare}(\mathsf{cert}^{\tau}, s_i^{\tau}) \stackrel{?}{=} \text{False}$: Complain to auditor about sender After receiving t^{τ} tuples with identical cert^{τ}: $\tau \leftarrow \mathsf{IMC.ExtractNametag}(\mathsf{cert}^{\tau}, (\{s_i^{\tau}\}_{i=1}^{t^{\tau}}))$ Complain to auditor if any two mismatching tuples are delivered.

Figure 12: Overview of the Request Identity Disclosure, Validate Identity Disclosure Request, Send & Receive Identity Disclosure actions which involve disclosing identity tag of the whistleblower. Initially, an operator prepares an identity disclosure request for disclosure of the identity tagd and sends it to the managing operators. The managing operators may either accept or reject the request. Whenever the request is accepted, the managing operators perform identity disclosure by sending their shares to a designated entity. Otherwise the operator confirms rejection of the request and takes back ownership of the report processing.

Send final outcome. Each MO_i Auth-Sends to the W $\mathsf{msg}_1 \coloneqq (\mathsf{sendFinOut}, \mathsf{repnym})$, $\mathsf{msg}_2 \coloneqq (\mathsf{outMsg})$ via ALS.Send to $\mathsf{pk}^{\mathsf{recv}}_{W}$, where outMsg contains summary of the investigation. At the moment of sending the first message, a timer δ^{sync} starts.

Receive final outcome. If W does not receive t final outcome messages or at least one is inconsistent before δ^{sync} expires, they may initiate an audit, otherwise, the final outcome is considered as received.

H.1.5 Archive Phase.

Archive report. All managing operators trigger the archiving of the report by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{archiveRep}, \mathsf{repnym})$, $\mathsf{msg}_2 \coloneqq (\perp)$ via ALS.Send to $\widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}}$. From this point on, the report becomes available for audit and read access, and no further modifications are permitted. At the moment of archiving the report, the close report timer Δ^{cls} starts.

H.1.6 Close Phase.

Close report. After Δ^{cls} expires all managing operators trigger the closing of the report by Auth-Send $\mathsf{msg}_1 \coloneqq (\mathsf{closeRep}, \mathsf{repnym})$, $\mathsf{msg}_2 \coloneqq (\perp)$ via ALS.Send to $\widehat{\mathsf{pk}^{\mathsf{recv}}_{\mathbf{MO}}}$. All entities involved in processing the report remove all level-2 messages as well as keys enabling receival of level-2 messages from ALS from their memory. From this point on, only level-1 messages related to the report remain available for reading and audit.

H.2 Audit Procedures

Process compliance audit. The auditor A is given a report pseudonym repnym (or a set of pseudonyms) to investigate. A begins by inspecting the ALS.PublicView and executing $\mathsf{ALS.AuditOpenMsg1}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}})$ to receive all message records $\langle(\mathsf{id}^{\mathsf{mr}})_i\rangle$ that contain the requested pseudonym(s) in level-1 messages, alternatively, the set $\langle(\mathsf{id}^{\mathsf{mr}})_i\rangle$ may be provided directly to A as part of the request. Next, A verifies whether the sequence of actions, their timing, and their authentication (for messages that are expected to originate from operator or managing operator) comply with the procedures specified in Figure 6. Any deviation from these requirements results in a finding of non-compliance.

Individual report audit. The auditor A is given a list of message record identifiers together with their opening tokens, $\langle(\mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2})_i\rangle$.[36] The auditor inspects the ALS.PublicView and executes $\mathsf{ALS.AuditOpenMsg2}(\mathsf{sk}^{\mathsf{audit}}, \mathsf{id}^{\mathsf{mr}}, \mathsf{t}^{\mathsf{msg}_2})$ for each requested identifier $\langle(\mathsf{id}^{\mathsf{mr}})_i\rangle$ to obtain the corresponding messages $\langle(\mathsf{msg}_2)_i\rangle$.

Depending on the request and context, a *Process compliance audit* may be conducted in parallel. A then investigates the received messages $\langle(\mathsf{msg}_2)_i\rangle$ in accordance with the prescribed procedures, for example to confirm or refute disputed message contents between involved entities.

For brevity, we move the details of security analysis of the system to Section I.

I System Security Analysis

In the analysis below, we assume a single probabilistic polynomial-time adversary that can control a subset of entities in the system (see the adversary assumption in the Section 5.1), by executing the operations of the controlled entities on the adversary's own machine. Thus, a total running time of the adversary together with its controlled entities or any entity in the system is guaranteed to be polynomial. Additionally, whenever we write in the analysis that some goal is ensured by a property of IMC or ALS, we assume that the analysis given in the component analysis can be placed in the environment of the adversary that controls the full system. A fully formal analysis should include a study of composability of the subsequent building blocks in the system with formal frameworks such as Universal Composability [11], which we postpone to future work.

[36] The provided list may represent only a subset of the full repnym's message history.

Lemma 3. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Report Processing Security Goals (goals G1–G8).*

Arg. We first consider G1: Process correctness. The whistleblower, the managing operators and the operators are obliged to log (or send) each action's result via ALS. A single confirmation message is required to claim an operation finished by the whistleblower or by an operator. Whenever the whistleblower suspects that the process is incompliant (e.g., the process is delayed), they can trigger an audit of the RS procedures. To claim an operation (or a voting procedure) finished by the managing operators, a synchronization mechanism is introduced. In each action executed by the managing operators, the synchronization mechanism enforces the managing operators to post at least t confirmation messages (with an additional requirement that all posted messages are consistent). With an assumption that the synchronization time of the managing operators is δ^{sync}, and there are at most f faulty and malicious nodes, and the number of managing operators is at least $2f + 1$, it is guaranteed that the actions are claimed finished, or any of the honest and correctly working nodes triggers an audit.

The G2: Processing recordability, is ensured by including an IMC's report pseudonym, as well as action name, and the timestamp in the level-1 ALS message. The report pseudonym is unique and bounded to a single report.

The G3: Report data time-bounded persistence is ensured by including all data associated with the action on the report in the level-2 ALS messages and asking the senders and receivers of these messages to remove data access tokens when the report is closed.

The G4: Report confidentiality is ensured by asking the entities interacting with the RS to generate fresh ALS key pairs, and to send messages via ALS that preserves the P4: Message confidentiality property. The whistleblower is additionally obliged to use a fresh key pair for communication with an organization representative during ES as well as for each submitted report. Additionally, the whistleblower secret-shares the report data among the managing operators, and the identity of the whistleblower is protected using IMC's certificate mechanism, as the identity or the linkability information on the pseudonyms can be retrieved only after a sufficient number of shares is collected.

The G5: Process compliance auditability is ensured by the P9: Level-1 auditability provided by the ALS functionality.

The G6: Individual report auditability is ensured by the P10: Level-2 auditability provided by the ALS functionality as well as trough protected access to the level-2 messages enforced by the implementation of the opening tokens (see the property P10: Level-2 auditability).

The G7: Artifacts authenticated and bound to reports is ensured by asking all operators and managing operators to send authenticated and signed messages via ALS.

The G8: Timely response is achieved by appropriately aligning the synchronization timers for managing operators and the deadlines of WS procedures with those mandated by the Directive, as well as by enforcing the obligation to log all actions in ALS. □

Lemma 4. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Procedures Involving Message Exchange with the System (goals G9–G10).*

Arg. The G9: Guaranteed delivery of messages exchanged with the system is ensured by P6: Assured availability of ALS. The G10: Designated receiver is ensured by P1: Correctness and P3: Receiver confidentiality of ALS. □

Lemma 5. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Create, Send & Receive Report (goals G11 and G13).*

Arg. The G11: Submittability is ensured since every correctly formatted report is sent via ALS and thus automatically made available to the managing operators. The G12: Unframeability is ensured by the unforgeability properties of an IMC certificate for a report pseudonym. The G13: Process valid reports ensures that all valid and only valid reports are processed in the system, as guaranteed by the properties of the IMC certificate. Certificate can only be derived when the whistleblower first obtains a ticket from the Enrollment System and then verifies their identity against a trusted certificate authority. □

Lemma 6. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Send & Receive Acknowledgement (goals G14–G15).*

Arg. The requirement to send acknowledgments via ALS and the procedural requirement to send acknowledgment only when the report is claimed valid along with managing operators synchronization mechanism are used to force that the acknowledgment is being sent (only) for valid reports effectively ensuring G14: Valid reports acknowledgment.

The G15: Acknowledgment unforgeability is ensured since messages sent by the managing operators are authenticated, thus the acknowledgment is unforgable. □

Lemma 7. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Reject Report (goal G16).*

Arg. The requirement to log rejection result via ALS, the procedural requirement to reject only when the report is claimed invalid along with managing operators synchronization mechanism ensure G16: Reject invalid reports. □

Lemma 8. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Send & Receive Final Outcome (goal G17).*

Arg. The requirement to send final outcome via ALS, the procedural include conclusive statement in the final outcome message, along with managing operators synchronization mechanism ensure G17: Contains valid information. □

Lemma 9. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Assign Operator (goals G18 and G19).*

Arg. The managing operators assign an operator by applying the f^{assignOp} function and broadcasting their vote via ALS. Since the managing operators are obliged to synchronize before they fix their vote, exit when there exists a pair of not consistent votes, there are at most f faulty and malicious managing operators, and the number of managing operators is at least $2f + 1$, this implies that any unanimous voting result is equal to the output of the f^{assignOp} function effectively ensuring G18: Assign operator correctness. The managing operators are obliged to send their shares to a selected operator, and any inconsistency between the shares is detected by that operator using the verifiability of the verifiable secret sharing scheme. If discrepancies are found, the operator calls for an audit. Because of this, the shares must be correct, thus fulfilling G19: Data access. □

Lemma 10. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Conclude Report (goal G20).*

Arg. The same argument holds as for Lemma 8. □

Lemma 11. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Decide & Choose Competence (goal G21).*

Arg. The requirement to decide competence and choose priority, together with the fact that any action performed on a report by an operator is univocal with recording it in ALS, ensures G21: Decided action is logged. □

Lemma 12. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Send Further Information Request, Send & Receive Further Information (goals G22 and G23).*

Arg. If a request does not contain a justification, the whistleblower requests an audit. The auditor can then use a Level-2 audit to verify whether the request for further information included a justification, thereby ensuring G22: Further information request contains justification. The G23: Further information is unframeable is ensured by P14: Entry unforgeability of the IMC. □

Lemma 13. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Send & Receive Feedback (goal G24).*

Arg. The same argument holds as for G22: Further information request contains justification of Lemma 12. □

Lemma 14. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Request Identity Disclosure, Validate Identity Disclosure Request, Send & Receive Identity Disclosure (goals G25–G27).*

Arg. The whistleblower who submitted a report is identifiable due to the properties of the certificate associated with the report pseudonym in the IMC. At the same time, the whistleblower's identity can only be disclosed in a controlled manner, namely through reconstruction from identity shares or linkability information shares provided by the managing operators to an authorized receiver, thereby ensuring G25: Identity protection. Recall that, under the honest managing operators assumption (see Section 5.1), identity reconstruction requires the participation of at least one honest managing operator. Consequently, unauthorized identity disclosure cannot occur. Additionally, due to P13: Report certificate unforgeability, the linkability information included in the ciphertext, can be used to link all related reports stemming from the same ticket, whereas they cannot be linked without collecting a sufficient amount of linkability information shares. For G26: Identity disclosure request, the same argument applies as for G22: Further information request contains justification of Lemma 12 which includes a clear indication whether the request concerns identity disclosure of the full identity tag or only the linkability information. The G27: Validation of identity disclosure requests is ensured by the managing operators synchronization mechanism. □

Lemma 15. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Request (Partially) Transfer Report, Transfer & Receive Transferred Report (goals G28 and G29).*

Arg. The same arguments hold as for Lemma 14. □

Lemma 16. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies Security Requirements for Write to Report (goal G30).*

Arg. The same argument holds as for G22: Further information request contains justification of Lemma 12. □

Lemma 17. *Given the assumptions listed in Section 5.1, the implementation of* ALS *presented in Section F.4, the implementation of* IMC *presented in Section G.4, the system design presented in Section H satisfies System Design Goals (goals DG1–DG3).*

Arg. The DG1: Denial of service protection is ensured by P6: Assured availability and is enhanced in the extended version of the system by applying a limit to the number of submitted reports (see the remark on report tag generation in Section H.1.3). The implementation of the IMC and ALS components as well as the

system design based on well-studied cryptographic primitives ensures DG2: Evolutionary design. The DG3: Separation of duties is ensured by properly assigning independent operators to each report through the f^{assignOp} function and providing only a limited subset of information on the report to the managing operators. □

J Funnel Details

Funnel details are presented in Tables 1 to 4.

Table 1: Complete list of procedures derived from the Directive [21]. The "Required by the Dir. (Internal/External)" column indicates whether the Directive explicitly mandates a procedure for internal and external reporting, respectively. The "Tech." column specifies whether a requirement can be addressed solely through technical means ("Yes") or requires human judgment ("No").

No.	Procedure	Description	Required by the Dir. (Internal/External)	Tech.	Comment	Relevant Articles
1	Submit report containing information on breaches	Create a report and send it to Reporting System.	Yes/Yes	Yes	Proposed system is online-platform based; oral reporting can be handled in many ways, e.g. by having a trusted operator record the message for the whistleblower or by using speech recognition software.	[Art.2], [Art.4], [Art.7], [Art.8(2)], [Art.9(1a)], [Art.11(1)], [Art.11(2b)], [Art.16(1)]
2	Receive report	Receive the report.	Yes/Yes	Yes	–	[Art.8(2)], [Art.11(1)]
3	Accept report	See below.	See below.	See below.	See below.	See below.
3.1	Check the legibility of the text	Verify whether the text is syntactically correct and conveys a meaningful message.	No/No	No	–	–
3.2	Decide if the report falls within one of the areas specified in the scope of State law	Decide if the report's subject falls in one of the categories specified in the national law (e.g. public procurement, financial services, etc.).	Yes/Yes	No	–	[Art.2]
3.3	Decide if the report was submitted by a person who is, was, or is going to be in a work relationship with the organization	Decide whether the person is related to the organization.	Yes/Yes	Yes	–	[Art.4]
3.4	Decide if the reported matter is clearly minor	Decide whether the reported matter is clearly minor.	No/Yes	No	–	[Art.11(3)]
4	Record the report	Store the report and all actions associated with it.	Yes/Yes	Yes	–	[Art.12(1b)], [Art.18(1)]
5	Inform about the rejection of the report	Inform the whistleblower that the report was not accepted.	Yes/Optional	Yes	–	[Art.11(3)], [Art.11(4)]
6	Designate an impartial person or competent authority for following-up the report	Select a single person for handling the report.	Yes/Yes	Yes	–	[Art.9(1c)], [Art.11(1)]
7	Send acknowledgment of receipt	Send an acknowledgment of receipt.	Yes/Yes	Yes	–	[Art.9(1b)], [Art.11(2b)]
8	Decide if the acknowledgment of receipt shall be provided	–	No/Yes	No	This functionality was removed from the proposed whistleblowing system because it violates the security of the whistleblower. It was originally introduced due to concerns that sending an acknowledgment could break confidentiality or jeopardize the investigation; however, the proposed system is resilient to those risks.	[Art.11(2b)]
9	Receive the acknowledgment of receipt	(see procedure name)	Yes/Yes	Yes	–	[Art.9(1b)], [Art.11(2b)]
10	Follow-up report	Assess the accuracy of the allegations made in the report, investigates, addresses the breach, etc.	Yes/Yes	Yes	–	[Art.9(1d)], [Art.9(1e)], [Art.11(2c)]
10.1	Decide if the organization has competence to follow up on the breach	(see procedure name)	No/Yes	No	–	[Art.11(6)]
10.2	Decide if the report requires further follow-up	(see procedure name)	No/Optional	No	–	[Art.11(3)]
10.3	Decide about the priority of follow-up reports	(see procedure name)	No/Yes	No	–	[Art.11(5)]
11	Ask for further information from the whistleblower	Send a message to the whistleblower with a request for more information.	Yes/Yes	Yes	–	[Art.9(1c)], [Art.12(4c)]
12	Provide report feedback	Send feedback to the whistleblower.	Yes/Yes	Yes	This function can also be used to inform the whistleblower in case of transferring the report or disclosing the identity.	[Art.9(1c)], [Art.11(1d)], [Art.11(5)], [Art.11(6)]
13	Provide final feedback	Send final feedback to the whistleblower.	No/Yes	Yes	–	[Art.11(2e)]

No.	Procedure	Description	Required by the Dir. (Internal/External)	Tech.	Comment	Relevant Articles
14	(Partially) Transfer / Export report	Transfer the current state of the report to an external entity, whistleblower is informed.	No/Yes	Yes	After transferring the report, the investigation in the original organization does not have to end and may continue.	[Art.11(6)]
15	Identity disclosure (de-anonymization)	The identity of the whistleblower is revealed to a designated entity, whistleblower is informed.	Yes/Yes	Yes	The disclosed identity of the whistleblower can be delegated to an external entity without the operator or managing operators learning it.	[Art.16(2)], [Art.16(3)], [Art.23]
16	Review of the procedures by Auditor	Review performed procedures and check the statuses of ongoing report handling.	No/Yes	Yes	–	[Art.14]
17	Delete report data once requirements are fulfilled	Remove the report data when sufficient time has passed or after meeting deletion requirements.	Yes/Yes	Yes	–	[Art.17], [Art.18(1)]

Table 2: Complete list of protected assets derived from the Directive [21]. The "Relevant Article" column indicates whether a source article applies to internal ("int") or external ("ext") reporting; if neither is specified, it applies to both. "SBP" denotes "Security Best Practice." "SBP derived [Art.x(y)]" indicates that [Art.x(y)] allows for interpretation, and we adopt a conservative interpretation providing the highest level of security for the whistleblower. "SBP derived" indicates that an asset is not explicitly stated in the Directive but follows from security best practices.

Entity	Name	Description	Relevant Articles
Whistleblower	Entitlement.WhistleblowerWorkRelated	Only persons in work-relation with the organization can report on it.	[Art.4], [Art.5(9)]
	Entitlement.Confidentiality	The process of verifying one's entitlement status does not reveal any information to unauthorized entities.	[Art.16(1)]
	Identity.Unframeability	Only Whistleblower may interact with the System on their own behalf.	SBP derived [Art.4(1)]
	Identity.Confidentiality	A Whistleblower's identity remains confidential unless it has been officially disclosed to specified entities.	[Art.16]
	Report.ReportedIssueBreachEULaw	The reported information on breaches concerns a breach of EU law.	[Art.2], [Art.6(1a)]
	Report.ConcernedPartiesIdentities	The identities of entities concerned in the Report remain confidential, except to entities responsible for investigating it.	int: [Art.9(1a)]; ext: [Art.12(1a)]
	Report.Submittability	A Whistleblower can submit a Report to the System.	[Art.5(3)], [Art.8(2)]
	Report.Unforgeability	A Report in the name of another Whistleblower cannot be forged.	SBP derived [Art.4(1)]
	Report.Confidentiality	The confidentiality of a Report is maintained.	int: [Art.9(1a)]; ext: [Art.12(1a)]
	Report.Integrity	The integrity of a Report is maintained.	int: [Art.9(1a)]; ext: [Art.12(1a)]
	Report.Completeness	The completeness of a Report is maintained.	int: [Art.9(1a)]; ext: [Art.12(1a)]
	Report.ActivityConfidentiality	The confidentiality of activities related to a Report is maintained.	SBP derived [Art.16]
	Report.Separation	Reports, unless repetitive, are processed separately.	SBP derived [Art.11(4)]
	Report.Record	The Report and the log of activities performed in relation to it are stored in the System for the duration of their preservation period.	[Art.17], [Art.18(1)]
	Report.TimelyProcessing	The processing of the Report is carried out in a timely manner.	int: [Art.9(1b)], [Art.9(7f)]; ext: [Art.11(2b)], [Art.11(2d)]
	Report.Auditable	The correctness of Report processing is auditable.	[Art.14], [Art.16(4)], [Art.18(1)]
	Report.Unlinkability	Reports originating from the same Whistleblower regarding different cases are unlinkable.	[Art.16(1)]
	Report.Linkability	Reports originating from the same Whistleblower regarding the same case are linkable.	SBP derived [Art.16(1)]
	Report.IdentityAttached	A Report, after meeting specified conditions, can always reveal the identity of the Whistleblower.	[Art.16(2)], [Art.16(3)], [Art.16(4)]
	Acknowledgement.Correctness	A Whistleblower receives an Acknowledgement after submitting a Report.	int: [Art.9(1b)]; ext: Adopted[37] [Art.11(2b)]
	Acknowledgement.Unforgeability	An Acknowledgement cannot be forged.	SBP derived
	Acknowledgement.Confidentiality	An Acknowledgement does not reveal unnecessary information.	SBP derived
	Acknowledgement.TimelyDelivery	The Acknowledgement is provided within seven days.	int: [Art.9(1b)]; ext: Adopted[38][Art.11(2b)]
	Followup.Confidentiality	The confidentiality of follow-up is maintained.[39]	int: [Art.9(1a)], [Art.9(1c)]; ext: [Art.11(2c)]; [Art.16(1)]
	Followup.TimelyDelivery	Follow-up feedback is provided within a reasonable timeframe.	int: [Art.9(1f)]; ext: [Art.11(2d)]
System	Uptime.Inflow	The System can process large inflow of Reports.	int: [Art.9(2)]; ext: [Art.11(5)], [Art.12(2)]
Law Enforcement	Transfer.Completeness	The completeness of transferred data is preserved.	int: SBP derived; ext: [Art.11(1f)], [Art.11(6)]; [Art.18(1)]
	Transfer.Integrity	The integrity of transferred data is preserved.	int: SBP derived; ext: [Art.11(1f)], [Art.11(6)]; [Art.18(1)]
	Transfer.Confidentiality	The confidentiality of transferred data is preserved.	int: SBP derived; ext: [Art.11(1f)], [Art.11(6)]; [Art.18(1)]
	IdentityDisclosure.Policy	The identity of the Whistleblower may be disclosed only according to policy.	[Art.16(2)]
	IdentityDisclosure.Designated	The outcome of Identity Disclosure is delivered only to specified entities.	[Art.16(3)]

[37]We design acknowledgment receipt such that it cannot jeopardize the protection.

[39]In the meaning of the Directive, "follow-up" procedures include any actions performed by the system in response to a submitted report. Recall that, throughout this work, instead of "follow-up", we use the term Handle Phase.

Table 3: Complete list of threats associated with system procedures, derived from the Directive [21]. Each threat is mapped to the corresponding security goals ("Security Goals" column) and the respective protection mechanisms ("Protection" column). Each protection specifies the components' names and their associated security properties.

Threat	Description	Security Goals	Protection	Comment
General				
T.Entitlement.UnauthorizedAccess	A person not entitled to whistleblow on an organization submits a report on it. This includes both forging an entitlement and transferring one's own entitlement to another person.	G12: Unframeability, G13: Process valid reports	IMC Ticket and report pseudonym unforgeability; IMC Report certificate unforgeability; IMC Entry unforgeability	It is possible to submit a report only when a person submits a valid pseudonym obtained from a ticket previously received from an organization. It is not possible to reuse a previously submitted report entry by some other entity.
T.Entitlement.Leak	The entitlement process leaks unnecessary information from the system, including the identity of the whistleblower, among other sensitive details.	G4: Report confidentiality	IMC Unlinkability of report pseudonyms (among pseudonyms); IMC Unlinkability of report pseudonyms (to the tickets)	Report pseudonyms are not linkable to the tickets received from an organization and are not linkable to each other, thus do not reveal any information about the whistleblower.
T.Entitlement.IncorrectConclusion	The entitlement proof is accepted, rejected, or ignored without proper justification.	G2: Processing recordability, G6: Individual report auditability, G17: Contains valid information	Managing operators collective computing	Managing Operators collectively evaluate correctness of the entitlement.
T.Identity.Impersonation	Another entity uses the person's identity to interact with the whistleblowing system. This includes all cases of impersonation, whether by a system operator or any other entity. It also applies when a report is created in the system on behalf of a person.	G12: Unframeability	IMC Ticket and report pseudonym unforgeability; IMC Report certificate unforgeability; IMC Entry unforgeability; ALS Message confidentiality	It is possible to submit a report in an organization only when one submits a valid pseudonym obtained from a ticket requested previously from the organization. It is not possible to reuse a previously submitted report entry by some other entity.
T.Identity.UnlinkableIdentity	The report is submitted using an identity that authorized entities cannot associate with an identity of a real individual.	G25: Identity protection	Managing operators collective computing	Managing Operators collectively verify whether the whistleblower's nametag can be received from the submitted report without revealing the nametag.
T.Report.IncorrectConclusion	A report is accepted, rejected, or ignored without proper justification. This includes rejecting reports not meeting formal requirements, as well as later stages where a report is incorrectly classified, whether intentionally or not, as minor, illegible, or outside the scope.	G6: Individual report auditability, G16: Reject invalid reports, G20: Contains conclusive statement	ALS Level-2 auditability (complaint by: Whistleblower); Managing operators collective computing	Managing Operators collectively evaluate the report's formal requirements before accepting it. Report is sent via ALS; Whistleblower or System opens level 2 message of submit-report message record; Auditor inspects ALS PublicView.
T.Report.NotDeleted	The report details are not deleted after the end of its required preservation period.	G3: Report data time-bounded persistence	ALS Level-1 auditability (complaint by: Whistleblower)	Deletion is recorded in ALS; Auditor inspects ALS PublicView.
T.Report.IncorrectSeparation	Reports, despite concerning the same breach, are processed separately, thus doubling the investigative work. On the other hand, reports which should be processed separately, are processed as a single report.	G18: Assign operator correctness, DG3: Separation of duties	IMC Unlinkability of report pseudonyms (among pseudonyms), IMC Unlinkability of report pseudonyms (to the tickets)	A whistleblower, during report creation, can choose to provide metadata that will allow Managing Operators to assign the report to an Operator who is or was processing reports from this whistleblower or from other whistleblowers who submitted reprots in the same category.
T.Acknowledgement.Forge	An acknowledgement for a report is forged. This includes creating fraudulent acknowledgements for both existing and non-existent reports, as well as altering a genuine acknowledgement by modifying associated data, such as the whistleblower's identity or the report it refers to.	G15: Acknowledgment unforgeability	ALS Level-1 auditability (complaint by: Whistleblower); ALS Message confidentiality; ALS Receival control	The acknowledgements sent out from the system are authenticated and bound with a unique report pseudonym.
T.Acknowledgement.UnlawfulRefraining	An acknowledgment is withheld, resulting in lack of protection of whistleblower.	G5: Process compliance auditability, G14: Valid reports acknowledgment	ALS Level-2 auditability (complaint by: Whistleblower)	Report is sent via ALS; Whistleblower or System opens level 2 message of submit-report message record; Auditor inspects ALS PublicView.
T.Acknowledgement.AuxiliaryInformation	The acknowledgment reveals more information than necessary, exceeding its intended scope. In other words, the information scope of the acknowledgment exceeds its purpose. It refers to the confidentiality of auxiliary information attached to the acknowledgement.	G4: Report confidentiality	ALS Message status confidentiality; Managing operators collective computing; Managing operators collective computing	The acknowledgment includes only the pseudonymous report identifier and an acknowledgment statement. The message is delivered to a designated receiver.

Threat	Description	Security Goals	Protection	Comment
T.IdentityDisclosure.UnauthorizedAccess	An unauthorized entity learns the identity of the whistleblower through identity disclosure.	G25: Identity protection, G26: Identity disclosure request, G27: Validation of identity disclosure requests	IMC Ticket and report pseudonym unforgeability; IMC Report certificate unforgeability (Conditioned tag extractability); Managing operators collective computing	Identity disclosure material is shared by the Managing Operators only with entities approved by the Managing Operators.
T.IdentityDisclosure.Ambiguous	Identity disclosure leads to multiple identities.	G25: Identity protection	IMC Report pseudonym uniqueness; IMC Report certificate unforgeability	Report pseudonyms are uniquely bound to a single identity.
T.System.ValidReportNotProcessed	A legitimate report is not processed. In particular, the system fails to handle legitimate reports during high-volume submissions, including malicious scenarios such as denial-of-service attacks in which fake reports are mixed with valid ones.	G13: Process valid reports, G2: Processing recordability, G5: Process compliance auditability, G19: Data access	IMC Report pseudonym uniqueness; ALS Assured availability	Submitted reports are retained reliably in ALS and can be processed as soon as system resources permit. Reports constituting flooding attacks, after discovery, can be uniquely linked back to the author and punished.
T.Send				
T.Send.ClaimButWasNot	Sender claims that message was sent (including sending time) but it was not.	G9: Guaranteed delivery	ALS Level-1 auditability (complaint by: Receiver)	Message is sent via ALS; Auditor inspects ALS record.
T.Send.ClaimDifferentContent	Sender claims that a message (content-wise) was sent, but in reality a different message (content-wise) was sent.	G9: Guaranteed delivery	ALS Level-2 auditability (complaint by: Receiver)	Message is sent via ALS; Auditor inspects ALS record; Sender opens level 2 message.
T.Send.ClaimTime	Sender claims that it sent a message earlier/later than it actually did.	G2: Processing recordability, G9: Guaranteed delivery	ALS Level-1 auditability (complaint by: Receiver)	Message is sent via ALS; Auditor inspects ALS record.
T.Send.ClaimNotButWas	Receiver claims that a message was not sent but it was.	G9: Guaranteed delivery, G22: Further information request contains justification, G23: Further information is unframeable, G24: Contains feedback statement, G28: Report (partial) transfer request, G29: Validation of report-(partial)-transfer requests	ALS Level-1 auditability (complaint by: Sender)	Message is sent via ALS; Auditor inspects ALS record; action initiators follow prescribed instructions.
T.Send.ChannelUnavailability	Sender is not able to send a message.	G9: Guaranteed delivery	ALS Assured availability	Message is sent via ALS.
T.Send.Leak	Sending the message exposes data beyond the payload. In particular, it may leak content of the message and/or communication metadata (e.g., sender and receiver addresses, timestamps) to unauthorized entities.	G4: Report confidentiality, G9: Guaranteed delivery	ALS Message confidentiality; ALS Sender anonymity; ALS Receiver confidentiality; ALS Message status confidentiality	–
T.Send.Malform	During sending, the message is modified in some manner.	G9: Guaranteed delivery	ALS Integrity protection	–
T.Send.CommunicationObstructed	Sending the message from sender to receiver is obstructed, resulting in delayed, disrupted, ignored, or lost messages. The obstruction may be intentional or unintentional, and may prevent the protocol from being completed or delay its completion beyond the expected timeframe, exposing one or both entities to risk.	G9: Guaranteed delivery	ALS Assured availability	Communication is conducted through ALS.
T.Receive				
T.Receive.ClaimButWasNot	Receiver claims that they did not receive a message but they did.	G11: Submittability, G9: Guaranteed delivery	ALS Level-1 auditability (complaint by: Sender); ALS Sender-side Receipt Verification	Message is received via ALS; Auditor inspects ALS record; alternatively, Sender may verify it themself and only after initial verification contact Auditor.
T.Receive.ClaimDifferentContent	Receiver claims that one message (content-wise) was received, but in reality a different message (content-wise) was received.	G9: Guaranteed delivery	ALS Level-2 auditability (complaint by: Sender)	Message is received via ALS; Auditor inspects ALS record; Sender or Receiver opens level 2 message.
T.Receive.ClaimTime	Receiver claims that it received a report earlier/later than it actually did.	G9: Guaranteed delivery, G5: Process compliance auditability	ALS Level-1 auditability (complaint by: Sender)	Message is received via ALS; Auditor inspects ALS record.

Threat	Description	Security Goals	Protection	Comment
T.Receive.ClaimNotButWas	Sender claims that a message was not received but it was.	G9: Guaranteed delivery, G22: Further information request contains justification, G23: Further information is unframeable, G24: Contains feedback statement, G28: Report (partial) transfer request, G29: Validation of report-(partial)-transfer requests	ALS Level-1 auditability (complaint by: Receiver)	Message is received via ALS; Auditor inspects ALS record; action initiators follow prescribed instructions.
T.Receive.ChannelUnavailability	Receiver is not able to receive a message.	G9: Guaranteed delivery	ALS Assured availability	Message is received via ALS.
T.Receive.Leak	Receiving the message exposes data beyond the payload. In particular, it may leak content of the message and/or communication metadata (e.g., sender and receiver addresses, timestamps) to unauthorized entities.	G10: Designated receiver, G4: Report confidentiality	ALS Message confidentiality; ALS Sender anonymity; ALS Receiver confidentiality; ALS Message status confidentiality	–
T.Receive.Malform	During receiving, the message is modified in some manner.	G9: Guaranteed delivery	ALS Integrity protection	–
T.Receive.CommunicationObstructed	Receiving the message from sender to receiver is obstructed, resulting in delayed, disrupted, ignored, or lost messages. The obstruction may be intentional or unintentional, and may prevent the protocol from being completed or delay its completion beyond the expected timeframe, exposing one or both entities to risk.	G9: Guaranteed delivery	ALS Assured availability	Communication is conducted through ALS.
T.SystemAction				
T.SystemActions.Illegitimate	An illegitimate system action is performed, including report amendment and deletion during its required preservation period.	G1: Process correctness, G2: Processing recordability, G5: Process compliance auditability, G21: Decided action is logged, G22: Further information request contains justification, G23: Further information is unframeable, G24: Contains feedback statement, G28: Report (partial) transfer request, G29: Validation of report-(partial)-transfer requests	ALS Integrity, ALS Level-1 auditability	System actions are recorded in ALS; ALS is an append-only data structure; an auditor can inspect the procedures; action initiators follow prescribed instructions.
T.SystemActions.LegitimateUnauthorized	A system action is performed by an unauthorized entity.	G2: Processing recordability, G7: Artifacts authenticated and bound to reports, G5: Process compliance auditability, G30: Authorized access to write	ALS Integrity, Message Authentication, ALS Level-1 auditability	System actions are recorded in ALS; action logs must be authenticated by entities operating the system.
T.SystemActions.HiddenActivity	A system action is performed without being recorded.	G2: Processing recordability, G5: Process compliance auditability	ALS Integrity, ALS level-1 auditability	Performing a system action is unequivocally coupled with recording it in the ALS; an action not recorded in the ALS is not claimed valid during audit.
T.SystemActions.Leak	Performing a system action reveals crucial information, such as the identity of the whistleblower or content of the report, or auxiliary information that may enable inference capabilities, such as that a given organization is handling a report.	G4: Report confidentiality, G10: Designated receiver, G25: Identity protection	ALS Message confidentiality; ALS Sender anonymity; ALS Receiver confidentiality; ALS Message status confidentiality; IMC unlinkability of report pseudonyms	Data is input to the ALS anonymously and is protected from an access of unauthorized entities through encryption. It reveals no information to unauthorized entities except the mere fact that data has been input.
T.SystemActions.RecordDelay	Performing a system action is logged with a delay.	DG1: Denial of service protection, DG8:	ALS Assured availability	Data is input to the ALS.
Certain System Actions				

Threat	Description	Security Goals	Protection	Comment
T.NotMeetingDeadline	The action is not completed within the expected timeframe.	DG8:	ALS Level-1 auditability (complaint by: Whistleblower)	System actions are recorded in ALS; Auditor verifies whether the action timestamp is within the expected timeframe.
All actions				
T.AuditorTooPowerful	An auditor acquires access to information about individual reports exceeding the minimum required for correct operation.	G5: Process compliance auditability, G6: Individual report auditability	ALS Level-1 auditability; ALS Level-2 auditability	Auditor has a direct access to level-1 messages, which contain a minimal set of information required to trace report processing; in order to access level-2 messages, auditor needs to cooperate with one of the involved entities.

Table 4: Combined overview of the relationships between threats, assets, and procedural requirements from Tables 1 to 3. Each procedure is associated with the corresponding Reporting System actions ("Related RS Actions" column), providing a mapping to the implementation proposed in this work.

No.	Procedure	Related Assets	Related Threats	Related RS Actions	Involved actors	Comments
1	Submit report containing information on breaches	Entitlement.WhistleblowerWorkRelated, Entitlement.Confidentiality, Identity.Unframeability, Identity.Confidentiality, Report.ReportedIssueBreachEULaw, Report.ActivityConfidentiality, Report.Submittability, Report.Record	T.Entitlement.UnathorizedAccess, T.Entitlement.Leak, T.Entitlement.IncorrectConclusion, T.Identity.Impersonation, T.Identity.UnlinkableIdentity, T.Send	sendRep	Whistleblower, Managing Operators	–
2	Receive report	Entitlement.Confidentiality, Identity.Unframeability, Identity.Confidentiality, Report.ReportedIssueBreachEULaw, Report.Submittability, Report.Unforgeability, Report.Confidentiality, Report.Integrity, Report.Completeness, Report.ActivityConfidentiality, Report.Record, Uptime.Inflow	T.Entitlement.Leak, T.Report.UnauthorizedModification, T.System.ValidReportNotProcessed, T.Receive	receiveRep	Managing Operators, Operator	–
3	Accept Report	Report.Auditable, Report.ActivityConfidentiality	T.Entitlement.UnathorizedAccess, T.Entitlement.Leak, T.Entitlement.IncorrectConclusion, T.Report.IncorrectConclusion, T.System.ValidReportNotProcessed, T.SystemAction	receiveRep	Managing Operators, Operator	–
3.1	Check the legibility of the text.	Followup.Confidentiality	T.SystemAction	decCompChosPrio	Managing Operators, Operator	–
3.2	Decide if the report falls within scope of State law	Report.ReportedIssueBreachEULaw	T.SystemAction	decCompChosPrio	Managing Operators, Operator	–
3.3	Decide if the report was submitted by a work-related person	Entitlement.WhistleblowerWorkRelated, Entitlement.Confidentiality	T.SystemAction	receiveRep	Managing Operators, Operator	–
3.4	Decide if the reported matter is clearly minor	Followup.Confidentiality	T.SystemAction	decCompChosPrio	Managing Operators, Operator	–
4	Record the report	Report.Integrity, Report.Completeness, Report.Record, Report.Auditable, Report.ActivityConfidentiality	T.SystemAction	receiveRep	Managing Operators, Operator	–
5	Inform about the rejection of the report	Report.Record, Report.Auditable, Report.ActivityConfidentiality	T.Identity.Impersonation, T.Report.IncorrectConclusion, T.System.CommunicationObstructed, T.Report.NotDeleted, T.Send, T.Receive, T.NotMeetingDeadline, T.SystemAction	sendFinOut, receiveFinOut, archiveRep, closeRep	Whistleblower, Managing Operators, Operator	System closes a report before the "report closing" period.
6	Designate an impartial person or competent authority	Entitlement.Confidentiality, Identity.Unframeability, Identity.Confidentiality, Report.ActivityConfidentiality, Report.Separation, Report.Record	T.SystemAction	assignOp	Managing Operators, Operator	–
7	Send acknowledgment of receipt	Acknowledgement.Correctness, Acknowledgement.Unforgeability, Acknowledgement.Confidentiality	T.Identity.Impersonation, T.Acknowledgement.Forge, T.Acknowledgemt.UnlawfulRefraining, T.Acknowledgement.AuxuliaryInformation, T.Send, T.NotMeetingDeadline	sendAck, receiveAck	Managing Operators	System does not send ACK before deadline.
8	Decide if acknowledgment shall be provided	–	–	–	–	Acknowledgement is always sent to valid reports / acknowledgement is never sent to invalid reports.

No.	Procedure	Related Assets	Related Threats	Related RS Actions	Involved actors	Comments
9	Receive acknowledgment of receipt	Identity.Unframeability, Acknowledgement.Correctness, Acknowledgement.Unforgeability, Acknowledgement.Confidentiality	T.Identity.Impersonation, T.Acknowledgement.Forge, T.Acknowledgemt.UnlawfulRefraining, T.Acknowledgement.AuxuliaryInformation, T.Receive	receiveAck	Whistleblower, Managing Operators	–
10	Follow-up report	Report.Record, Report.Auditable, Followup.Confidentiality	T.Report.IncorrectConclusion, T.System.ValidReportNotProcessed, T.SystemAction	decCompChosPrio	Managing Operators, Operator	–
10.1	Decide competence to follow-up	Followup.Confidentiality	T.SystemAction	decCompChosPrio	Operator	–
10.2	Decide if report requires follow-up	Followup.Confidentiality	T.SystemAction	decCompChosPrio	Operator	–
10.3	Decide priority of follow-up reports	Followup.Confidentiality	T.SystemAction	decCompChosPrio	Operator	–
11	Ask for further information	Identity.Unframeability, Identity.Confidentiality, Followup.Access, Followup.Confidentiality, Uptime.Inflow	T.Identity.Impersonation, T.Send, T.Receive, T.NotMeetingDeadline	sendFurInfoReq, receiveFurInfoReq, sendFurInfo, receiveFurInfo	Whistleblower, Operator	Whistleblower timeout on further information.
12	Provide report feedback	Identity.Unframeability, Identity.Confidentiality, Followup.Access, Followup.Confidentiality, Uptime.Inflow	T.Identity.Impersonation, T.Send, T.Receive, T.NotMeetingDeadline	sendFeed, receiveFeed	Whistleblower, Operator	Operator misses feedback deadline.
13	Provide final feedback	Report.Record, Report.Auditable, Report.ActivityConfidentiality	T.Identity.Impersonation, T.Report.IncorrectConclusion, T.System.CommunicationObstructed, T.Report.NotDeleted, T.Send, T.Receive, T.NotMeetingDeadline	sendFinOut, receiveFinOut, archiveRep, closeRep	Whistleblower, Operator	System closes report early.
14	Transfer/Export report	Identity.Unframeability, Identity.Confidentiality, Report.Unforgeability, Report.Confidentiality, Report.Integrity, Report.Completeness, Report.ActivityConfidentiality, Report.Record, Report.Auditable, Uptime.Inflow, Transfer.Completeness, Transfer.Integrity, Transfer.Confidentiality	T.Send, T.Receive, T.SystemAction	reqTransRep, decideTransferReport, transRep, receiveTransRep	Whistleblower, Managing Operators, Operator, External Entity	–
15	Identity disclosure (de-anonymization)	IdentityDisclosure.Policy, IdentityDisclosure.Designated	T.Send, T.Receive, T.IdentityDisclosure.UnauthorizedAccess, T.IdentityDisclosure.Ambiguous, T.SystemAction	reqIDDisclsoure, decideIdentityDisclosure, rejectIDDisclosure, sendIDDisclosure, receiveIDDisclsoure	Whistleblower, Managing Operators, Operator, External Entity	–
16	Review procedures by Auditor	Report.Auditable	T.Report.IncorrectConclusion, T.Report.NotDeleted, T.Send, T.Receive, T.SystemAction, T.NotMeetingDeadline	receiveTransRep	Managing Operators, Operator, Auditor	–
17	Delete report data	Report.Record	T.SystemAction	closeRep	Managing Operators, Operator	–

K Details of Preliminaries

The definitions in this section are drawn or adapted from [6, 13, 33].

K.1 Digital Signature Scheme

Definition 5 (Digital signature scheme). *A* digital signature scheme Sig *consists of three PPT algorithms* (KeyGen, Sign, Verify)*, where:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$*: Takes as input security parameter* 1^λ*, and returns a pair* (sk, pk) *of keys, where* sk *is a private (signing) key and* pk *is a corresponding public (verification) key.*
- $\mathsf{Sign}_{\mathsf{sk}}(m) \to \sigma$*: Takes as input a private signing key* sk *and message* m*, returns signature* σ *over* m*.*
- $\mathsf{Verify}_{\mathsf{pk}}(\sigma) \to \{0,1\}$*: Takes as input a public key* pk*, a message* m*, and a signature* σ*, returns a bit* b*, with* $b = 1$ *meaning valid and* $b = 0$ *meaning invalid.*

Let the existential signature forgery experiment $\mathsf{SigForge}_{\mathcal{A},\mathsf{Sig}}$ be defined as Algorithm 1.

Algorithm 1 Existential Signature Forgery ($\mathsf{SigForge}_{\mathcal{A},\mathsf{Sig}}$) experiment

function $\mathsf{SigForge}_{\mathcal{A},\mathsf{Sig}}(\lambda)$
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
  $Q \leftarrow \emptyset$ ▷ Set of queried messages
  Define oracle $O(m)$:
    $\sigma \leftarrow \mathsf{Sign}_{\mathsf{sk}}(m)$
    $Q \leftarrow Q \cup \{m\}$
    return σ
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{O(\cdot)}(1^\lambda, \mathsf{pk})$
  if $\mathsf{Verify}_{\mathsf{pk}}(m^*, \sigma^*) \neq 1$ **then**
    return 0
  end if
  if $m^* \in Q$ **then**
    return 0
  end if
  return 1
end function

Definition 6 (Existential unforgeability). *A signature scheme* Sig = (Gen, Sign, Verify) *is* existentially unforgeable under an adaptive chosen-message attack *if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* $negl$ *such that:*

$$\Pr\left[\mathsf{SigForge}_{\mathcal{A},\mathsf{Sig}}(\lambda) = 1\right] \leq negl(\lambda).$$

K.2 Symmetric Encryption Scheme

Definition 7 (Symmetric encryption scheme). *A* symmetric encryption scheme Sym *consists of three PPT algorithms* (KeyGen, Enc, Dec)*, where:*

- $\mathsf{KeyGen}(1^\lambda) \to k$*: Takes as input security parameter* 1^λ*, and returns a secret key* k*.*
- $\mathsf{Enc}_k(m) \to c$*: Takes as input a key* k *and message* m*, returns a ciphertext* c*.*
- $\mathsf{Dec}_k(c) \to m \cup \{\perp\}$*: Takes as input a key* k *and ciphertext* c*, returns a message* m *or a special symbol* $\perp$ *indicating failure. It is required that for every* λ*, every key* $k \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and every* $m \in \{0,1\}^*$*, it holds that* $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$*.*

Let the indistinguishability under chosen-ciphertext attack experiment $\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Sym}}$ be defined as Algorithm 2.

Algorithm 2 IND-CCA experiment ($\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Sym}}$)

function $\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Sym}}(\lambda)$
  $k \leftarrow \mathsf{KeyGen}(1^\lambda)$
  Define encryption oracle $O_{\mathsf{Enc}}(m)$:
    return $\mathsf{Enc}_k(m)$
  Define decryption oracle $O_{\mathsf{Dec}}(c)$:
    return $\mathsf{Dec}_k(c)$
  $(m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}^{O_{\mathsf{Enc}}, O_{\mathsf{Dec}}}(1^\lambda)$
  if $|m_0| \neq |m_1|$ **then**
    return 0
  end if
  $b \xleftarrow{\$} \{0,1\}$
  $c^* \leftarrow \mathsf{Enc}_k(m_b)$
  $b' \leftarrow \mathcal{A}^{O_{\mathsf{Enc}}, O_{\mathsf{Dec}}}(\mathsf{state}, c^*)$ ▷ $\mathcal{A}$ continues to have oracle access to O_{Enc} and O_{Dec}
  if $\exists$ query $c = c^*$ to O_{Dec} **then**
    return 0
  end if
  if $b' = b$ **then**
    return 1
  else
    return 0
  end if
end function

Definition 8 (IND-CCA security). *A symmetric encryption scheme* Sym = (KeyGen, Enc, Dec) *is* indistinguishable under adaptive chosen-ciphertext attack (IND-CCA secure) *if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* $negl$ *such that:*

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Sym}}(\lambda) = 1\right] \leq \tfrac{1}{2} + negl(\lambda).$$

Let the unforgeable encryption experiment $\mathsf{Enc\text{-}Forge}_{\mathcal{A},\mathsf{Sym}}$ be defined as Algorithm 3.

Definition 9 (Unforgeable encryption). *A symmetric encryption scheme* Sym = (KeyGen, Enc, Dec) *is* unforgeable *if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* $negl$ *such that:*

$$\Pr\left[\mathsf{Enc\text{-}Forge}_{\mathcal{A},\mathsf{Sym}}(\lambda) = 1\right] \leq negl(\lambda).$$

Definition 10 (Authenticated encryption scheme). *A symmetric encryption scheme* SymAuth = (KeyGen, Enc, Dec) *is* authenticated encryption scheme *if it is IND-CCA (Definition 8) secure and unforgeable (Definition 9).*

K.3 Asymmetric Encryption Scheme

Definition 11 (Asymmetric encryption scheme). *An* asymmetric encryption scheme Asm *consists of three PPT algorithms* (KeyGen, Enc, Dec)*, where:*

- $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$*: Takes as input the security parameter* 1^λ*, and outputs a public key* pk *and secret key* sk*. The public key* pk *defines a message space* $\mathcal{M}_{pk}$*.*

Algorithm 3 Unforgeable Encryption (Enc-Forge$_{\mathcal{A},\mathsf{Sym}}$) experiment

function Enc-Forge$_{\mathcal{A},\mathsf{Sym}}(\lambda)$
  $k \leftarrow \mathsf{KeyGen}(1^\lambda)$
  $Q \leftarrow \emptyset$ ▷ Set of queried messages
  Define oracle $O_{\mathsf{Enc}}(m)$:
    $c \leftarrow \mathsf{Enc}_k(m)$
    $Q \leftarrow Q \cup \{m\}$
    return c
  $c^* \leftarrow \mathcal{A}^{O_{\mathsf{Enc}}}(1^\lambda)$
  $m^* \leftarrow \mathsf{Dec}_k(c^*)$
  if $m^* = \perp$ **then**
    return 0
  end if
  if $m^* \in Q$ **then**
    return 0
  end if
  return 1
end function

- $\mathsf{Enc}_{pk}(m) \to c$*: Takes as input a public key* pk *and message* $m \in \mathcal{M}_{pk}$*, and returns a ciphertext* c*.*
- $\mathsf{Dec}_{sk}(c) \to \{m, \perp\}$*: Takes as input a secret key* sk *and ciphertext* c*, and (deterministically) returns a message* $m \in \mathcal{M}_{pk}$ *or a special symbol* $\perp$ *indicating failure.*

It is required that, except with negligible probability over the randomness of Gen *and* Enc*, we have* $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m$ *for any message* $m \in \mathcal{M}_{pk}$*.*

Let the indistinguishability under chosen-ciphertext attack experiment $\mathsf{PubK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Asm}}$ be defined as Algorithm 4.

Algorithm 4 IND-CCA experiment ($\mathsf{PubK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Asm}}$)

function $\mathsf{PubK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Asm}}(\lambda)$
  $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
  Define decryption oracle $O_{\mathsf{Dec}}(c)$:
    return $\mathsf{Dec}_{sk}(c)$
  $(m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}^{O_{\mathsf{Dec}}}(pk)$
  if $|m_0| \neq |m_1|$ **then**
    return 0
  end if
  $b \xleftarrow{\$} \{0,1\}$
  $c^* \leftarrow \mathsf{Enc}_{pk}(m_b)$
  $b' \leftarrow \mathcal{A}^{O_{\mathsf{Dec}}}(\mathsf{state}, c^*)$
  if $\exists$ query $c = c^*$ to O_{Dec} **then**
    return 0
  end if
  if $b' = b$ **then**
    return 1
  else
    return 0
  end if
end function

Definition 12 (IND-CCA security). *A asymmetric encryption scheme* Asm = (KeyGen, Enc, Dec) *is* indistinguishable under adaptive chosen-ciphertext attack (IND-CCA secure) *if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function negl such that:*

$$\Pr\left[\mathsf{PubK}^{\mathsf{cca}}_{\mathcal{A},\mathsf{Asm}}(\lambda) = 1\right] \leq \tfrac{1}{2} + negl(\lambda).$$

K.4 Randomness Commitment Scheme

Definition 13 (Randomness commitment scheme). *A* randomness commitment scheme RandCom = (Gen, Verify) *consists of two PPT algorithms, where:*

- $\mathsf{Gen}(1^\lambda) \to (v, c)$*: Takes as input a security parameter* 1^λ*, samples a value* $v \leftarrow\!\!\$\ \{0,1\}^\lambda$*, and outputs an opening* v *and a commitment* c*.*
- $\mathsf{Verify}(v, c) \to \{0,1\}$*: Takes as input a value* v *and a commitment* c*, and outputs* 1 *if* v *is a valid opening of* c*, and* 0 *otherwise.*

Let the hiding experiment $\mathsf{Hide}_{\mathcal{A},\mathsf{RandCom}}$ be defined as Algorithm 5.

Algorithm 5 Hiding experiment ($\mathsf{Hide}_{\mathcal{A},\mathsf{RandCom}}$)

function $\mathsf{Hide}_{\mathcal{A},\mathsf{RandCom}}(\lambda)$
  $(v, c) \leftarrow \mathsf{Gen}(1^\lambda)$
  $b' \leftarrow \mathcal{A}(1^\lambda, c)$
  if $b' = v$ **then**
    return 1
  end if
  return 0
end function

Definition 14 (Hiding). *A randomness commitment scheme* RandCom = (Gen, Verify) *is* hiding *if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function negl such that:*

$$\Pr\left[\mathsf{Hide}_{\mathcal{A},\mathsf{RandCom}}(\lambda) = 1\right] \leq negl(\lambda).$$

K.5 Certificate Scheme

Definition 15 (Certificate scheme). *A* certificate scheme Crt *consists of four PPT algorithms* (KeyGen, Certify, Verify, Decode)*, where:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$*: Takes as input security parameter* 1^λ*, and returns a pair* (sk, pk) *of keys, where* sk *is a private (certifying) key and* pk *is a corresponding public verification key.*
- $\mathsf{Certify}_{\mathsf{sk}}(tbc) \to \mathsf{cert}$*: Takes as input a set* $tbc = \{(n_i, v_i)\}$ *of to be certified name-value pairs, where* n_i *is an alphanumeric field name and* $v_i \in \{0,1\}^*$ *is a field value, and uses private certifying key* sk*, returns* $\mathsf{cert} \in \{0,1\}^*$*. Field names are unique, i.e.,* $(\forall i \neq j)(n_i \neq n_j)$*. We say that* cert *is a certificate of* tbc*.*
- $\mathsf{Verify}_{\mathsf{pk}}(\mathsf{cert}) \to \{T, \perp\}$*: Takes as input a certificate* $\mathsf{cert} \in \{0,1\}^*$*, and using the public verification key* pk*, returns* T *if the certificate is valid, i.e., was certified using the private certifying key corresponding to the public key* pk*, and* $\perp$ *otherwise.*
- $\mathsf{Decode}(\mathsf{cert}) \to tbc \cup \{\perp\}$*: Takes as input a certificate* $\mathsf{cert} \in \{0,1\}^*$*, and returns a set tbc of name-value pairs, as defined above, or* $\perp$ *(when decoding fails).*

Definition 16 (Correctness). *A certificate scheme* Crt = (KeyGen, Certify, Verify, Decode) *ensures* correctness *if for every* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and for every set of name-value pairs tbc, with unique names, the following holds:*

(1) $\mathsf{Verify}_{\mathsf{pk}}(\mathsf{Certify}_{\mathsf{sk}}(tbc)) = T$, *and*
(2) $\mathsf{Decode}(\mathsf{Certify}_{\mathsf{sk}}(tbc)) = tbc$.

Let existential certificate forgery experiment $\mathsf{ECF}_{\mathcal{A},\mathsf{Crt}}$ be defined as Algorithm 6.

Algorithm 6 Existential Certificate Forgery ($\mathsf{ECF}_{\mathcal{A},\mathsf{Crt}}$) experiment

function $\mathsf{ECF}_{\mathcal{A},\mathsf{Crt}}(\lambda)$
  $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
  $Q \leftarrow \emptyset$ ▷ Set of certified messages
  Define oracle $O(tbc)$:
    $\mathsf{cert} \leftarrow \mathsf{Certify}_{\mathsf{sk}}(tbc)$
    $Q \leftarrow Q \cup \{tbc\}$
    return cert
  $\mathsf{cert}^* \leftarrow \mathcal{A}^{O(\cdot)}(1^\lambda, \mathsf{pk})$
  $tbc^* \leftarrow \mathsf{Decode}(\mathsf{cert}^*)$
  if $tbc^* = \perp$ **then return** 0
  end if
  if $\mathsf{Verify}_{\mathsf{pk}}(\mathsf{cert}^*) \neq 1$ **then return** 0
  end if
  if $tbc^* \in Q$ **then return** 0
  end if
  return 1
end function

Definition 17 (Existential unforgeability). *A certificate scheme* Crt = (KeyGen, Certify, Verify, Decode) *is* existentially unforgeable, *if for all PPT adversaries $\mathcal{A}$, there is a negligible function negl such that:*

$$\Pr\left[\mathsf{ECF}_{\mathcal{A},\mathsf{Crt}}(\lambda) = 1\right] \leq negl(\lambda),$$

K.6 Pseudorandom Function

Definition 18 (Pseudorandom Function). *An efficient, length-preserving keyed function* $\mathsf{PRF} : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ *is a* pseudorandom function *if for all PPT distinguishers $\mathcal{D}$, there exists a negligible function negl such that:*

$$\left|\Pr\left[\mathcal{D}^{\mathsf{PRF}_K(\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathcal{D}^{f(\cdot)}(1^\lambda) = 1\right]\right| \leq negl(\lambda),$$

where:

- *the first probability is taken over a uniform choice of key* $K \leftarrow \{0,1\}^\lambda$ *and the randomness of $\mathcal{D}$,*
- *the second probability is taken over a function f chosen uniformly at random from the set of all functions mapping* $\{0,1\}^\lambda$ *to* $\{0,1\}^\lambda$, *and the randomness of $\mathcal{D}$.*

K.7 Verifiable Secret Sharing

Definition 19 (Verifiable Secret Sharing). *A* verifiable secret sharing scheme VSS *consists of three PPT algorithms* (Share, Verify, Recon), *where:*

- $\mathsf{Share}(s, t, n, 1^\lambda) \to (\boldsymbol{s}, vs)$*: Takes as input a secret $s \in \mathcal{S}$, threshold t, number of parties n, and security parameter 1^λ, run by a dealer D, and outputs a vector of shares $\boldsymbol{s} = (s_1, \ldots, s_n)$ and public verification information vs.*
- $\mathsf{Verify}(s_i, vs, i) \to \{0,1\}$*: Takes as input a share s_i, verification information vs, and entity index i, and returns a bit b, with $b = 1$ meaning valid and $b = 0$ meaning invalid.*
- $\mathsf{Recon}(\{s_i\}_{i \in I}) \to \{s, \perp\}$*: Takes as input a set of shares indexed by $I \subseteq \{1, \ldots, n\}$, and outputs either a secret $s \in \mathcal{S}$ or $\perp$.*

Definition 20 (Correctness). *A VSS scheme is* correct *if for any secret $s \in \mathcal{S}$, any honest dealer D, and any set $I \subseteq \{1, \ldots, n\}$ with $|I| \geq t$, it holds that*

$$\Pr\left[\mathsf{Recon}(\{s_i\}_{i \in I}) = s\right] = 1,$$

where $(\boldsymbol{s}, vs) \leftarrow \mathsf{Share}(s, t, n, 1^\lambda)$ *and* $\mathsf{Verify}(s_i, vs, i) = 1$ *for all $i \in I$.*

Definition 21 (Security). *A* VSS *scheme is* secure *against a static adversary corrupting at most $t - 1$ parties if it satisfies:*

(1) ***Privacy:*** *For any two secrets $s_0, s_1 \in \mathcal{S}$, any adversary $\mathcal{A}$ corrupting at most $t - 1$ parties, the adversary's view (including its randomness, the shares of corrupted parties, and all messages received during the sharing phase) is computationally indistinguishable:*

$$\mathsf{View}_{\mathcal{A}}(\mathsf{Share}(s_0, t, n, 1^\lambda)) \approx_c \mathsf{View}_{\mathcal{A}}(\mathsf{Share}(s_1, t, n, 1^\lambda)).$$

(2) ***Verifiability (Consistency):*** *For any (possibly malicious) dealer D^*, and any adversary corrupting at most $t - 1$ parties, there exists a value $s^* \in \mathcal{S} \cup \{\perp\}$ such that, except with negligible probability:*
- *All honest parties either reject their shares (i.e., output $\perp$) or accept shares that are consistent with s^*, and*
- *Any set of at least t shares that are accepted by* Verify *reconstructs to s^*.*

K.8 Non-Interactive Zero-Knowledge Proofs of Knowledge

Definition 22 (Non-Interactive Zero-Knowledge Proofs of Knowledge). *Let $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation and let $L = \{x \mid \exists w$ such that $(x, w) \in \mathcal{R}\}$ be the associated language. A* non-interactive proof system *for $\mathcal{R}$ is a tuple of PPT algorithms* (Setup, Prove, Verify) *such that:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{CRS}$*: Takes as input security parameter 1^λ, and returns a common reference string* CRS.
- $\mathsf{Prove}(\mathsf{CRS}, x, w) \to \pi$*: Takes as input a common reference string* CRS*, a statement x and a witness w, and returns a proof π.*
- $\mathsf{Verify}(\mathsf{CRS}, x, \pi) \to \{0,1\}$*: Takes as input a common reference string* CRS*, a statement x and a proof π, and returns* 1 *(accept) or* 0 *(reject).*

The system is a non-interactive zero-knowledge proof of knowledge *for $\mathcal{R}$*

$$\mathsf{NIZKPoK}_{\mathcal{R}} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$$

if the following properties hold:

(1) ***Completeness:*** *For all* $(x, w) \in \mathcal{R}$,

$$\Pr\left[\mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1 \;\middle|\; \begin{array}{l} \mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda), \\ \pi \leftarrow \mathsf{Prove}(\mathsf{CRS}, x, w) \end{array}\right] = 1.$$

(2) ***Knowledge Soundness:*** *There exists a PPT extractor* Extract *such that for every PPT prover* Prove^* *there exists a negligible function* $negl(\cdot)$ *satisfying:*

$$\Pr\left[\begin{array}{l} \mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1 \;\wedge\; (x, w) \notin \mathcal{R} \;\Big|\; \\ \mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda),\; \pi \leftarrow \mathsf{Prove}^*(\mathsf{CRS}, x), \\ w \leftarrow \mathsf{Extract}^{\mathsf{Prove}^*}(\mathsf{CRS}, x) \leftarrow \mathsf{Prove}(\mathsf{CRS}, x, w) \end{array}\right] \leq negl(\lambda)$$

(3) ***Zero-Knowledge:*** *There exists a PPT simulator* Sim *such that for all* $(x, w) \in \mathcal{R}$*, the distributions*

$$\{(\mathsf{CRS}, \mathsf{Prove}(\mathsf{CRS}, x, w))\} \quad \textit{and} \quad \{(\mathsf{CRS}, \mathsf{Sim}(\mathsf{CRS}, x))\}$$

are computationally indistinguishable, where $\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda)$.