Humanitarian Aid Distribution with Privacy-Preserving Assessment Capabilities

Christian Knabenhans
EPFL
christian.knabenhans@epfl.ch

Lucy Qin
Georgetown University
lucy.qin@georgetown.edu

Justinas Sukaitis
International Committee of the Red Cross
jsukaitis@icrc.org

Vincent Graf Narbel
International Committee of the Red Cross
vgraf@icrc.org

Carmela Troncoso
EPFL, MPI-SP
carmela.troncoso@mpi-sp.org

Abstract

In times of crisis, humanitarian organizations bring aid to those affected (e.g., water, food, medical supplies, cash assistance). Prior works introduced privacy-preserving systems for digitizing the aid distribution process, increasing their efficiency and security [8, 17, 25]. These solutions, by design, do not allow humanitarian organizations to collect metrics about the aid distribution process. Such assessments (e.g., the proportion of aid distributed to a minority) are crucial to enable the organizations to improve their operations, to perform their duty of care, and to enable transparency and accountability towards recipients, donors, and the public in general.

In partnership with the International Committee of the Red Cross (ICRC), we identify *assessments* relevant to humanitarian aid deployments and these assessments' security and privacy requirements.

We introduce a generic framework that augments existing privacy-preserving humanitarian aid distributions with such assessments. This framework enables the collection of aggregate statistics about the aid distribution process without compromising the privacy of recipients, and without requiring any changes to the existing protocols. To realize our framework we introduce *one-time functional encryption* (1FE), for which we propose efficient realizations from standard cryptographic primitives. We design and implement two variants of our framework: a more efficient one, secure against semi-honest adversaries; and a more robust one, secure against malicious adversaries.

We also introduce the novel notions of *threat model agility* and *graceful degradation*. These notions enable us to model the unstable environment of humanitarian aid distribution, where the capabilities of the adversary may change suddenly (e.g., when a militia takes over a region in conflict), invalidating the threat model under which the system was originally deployed. We believe these notions are of independent interest for other privacy-preserving applications deployed in unstable environments.

Keywords

Assessments, Humanitarian Aid, Functional Encryption

Proceedings on Privacy Enhancing Technologies 2026(2), 336–353

https://doi.org/10.56553/popets-2026-0051

1 Introduction

Humanitarian aid organizations provide aid to people affected by crises, such as armed conflicts, natural disasters, or pandemics. This aid can take various forms, such as cash, food, water, or medical supplies. Humanitarian aid distribution has historically relied on manual processes, which can be slow, error-prone, and costly. Digitalizing aid distribution can help to scale up humanitarian operations, and to reduce their cost, increasing their capability to create impact. However, digitalization brings new privacy and security challenges [13]. These challenges are particularly acute in the context of humanitarian aid, where aid recipients are particularly vulnerable, and where activities are subject to a high degree of scrutiny by the public, the media, and donors. This is compounded by very strong adversaries, ranging from corrupt officials to armed groups which do not observe legal boundaries or agreements, and by the fact that the recipients themselves may not be able to protect their own privacy, as they might lack resources or digital literacy.

Previous works proposed digital solutions for humanitarian aid distribution in collaboration with the International Committee of the Red Cross (ICRC) [8, 17, 25]. These works provide strong privacy guarantees, but do not (by design) allow for the collection and evaluation of metrics about the distribution process. Such metrics are essential for humanitarian organizations, as aid distribution is a complex and dynamic process that requires detailed advance planning. Metrics allow humanitarian aid organizations to evaluate prior deployments and measure the effectiveness of their operational decisions (e.g., they enable organizations to understand how particular subpopulations are impacted by an aid distribution program). In some cases, information about recipients is necessary for aid organizations to perform additional duties of care, such as performing welfare checks on senior recipients that have not claimed aid for a long time. Lastly, metrics of the aid distribution process are critical for transparency. Humanitarian organizations regularly communicate their activities as a measure of accountability to the communities they serve [9]. This strengthens trust and allows for direct community feedback. Humanitarian organizations also share metrics with donors in order to communicate the impact of donations and incentivize future donations, which may be critical to the long-term sustainability of their work.

In this work, we design protocols that enable the collection of relevant metrics for assessments about the aid distribution process,

while still providing strong privacy guarantees for the aid recipients. Our contributions are as follows.

Modeling assessments We collaborate with the ICRC to formally model the correctness, security, and privacy requirements of assessments for humanitarian aid distribution (§§2 and 3). To address the fact that aid is distributed in inherently unstable environments, we introduce the novel notions of *threat model agility* and *graceful degradation* (§3.2), which minimize potential harm in the presence of a dynamic adversary. These notions allow us to model the unique challenges of the humanitarian aid setting (§3.2), and we believe that they are of independent interest for other privacy-preserving applications deployed in unstable environments.

Blueprints for privacy-preserving assessments Introducing assessments without breaking the privacy guarantees of previous designs [8, 17, 25] is challenging in the humanitarian setting. Naively collecting metrics about recipients can lead to harm for recipients (e.g., discrimination or tracking) and has implications for the humanitarian organization (e.g., compliance with regulations). Further, aid recipients typically only have low-end hardware and restricted communication channels, which prevents the use of many advanced privacy-preserving cryptographic protocols and of centralized solutions. We design a generic blueprint for private assessment protocols in §4. Along the way, we introduce the notion of *one-time functional encryption* (1FE); we believe this primitive is of independent interest for other privacy-preserving applications.

Lightweight protocols We provide several protocols realizing the 1FE functionality, using either secure two-party computation or threshold homomorphic encryption (§5). We discuss how to integrate our protocols in existing privacy-preserving aid distribution systems, implement prototypes, and evaluate their efficiency (§6). Finally, we discuss the limitations of harm prevention in the presence of assessment capabilities (§7).

2 Digitalized humanitarian aid distribution

We first recall the model and requirements of humanitarian aid distribution that Wang et al. identify in their collaboration with International Committee of the Red Cross [25] as a means of contextualizing our contributions. Subsequent works complemented this model with a privacy-preserving registration scheme [8] and with flexible payments [17]. Our work complements all these variants with an *assessment* phase that computes, in a privacy-preserving manner, additional information about the distribution process in order to assess the distribution process, improve future deployments, and fulfill their duty of care.

The aforementioned works [8, 17, 25] consider the following parties:

- *Recipients* $\mathcal{P}_i$, which are households wishing to receive aid;
- A *registration station* $\mathcal{R}$, which enrolls recipients into the program;
- A *distribution station* $\mathcal{D}$, which dispenses aid;
- An external *auditor* $\mathcal{A}$ which audits the process.

Humanitarian aid distribution consists of three phases: registration, distribution, and auditing (described below and illustrated in Fig. 1). A humanitarian aid distribution program unfolds over a number of *distribution periods* (often a-priori upper-bounded); periods range from a few days to several months.

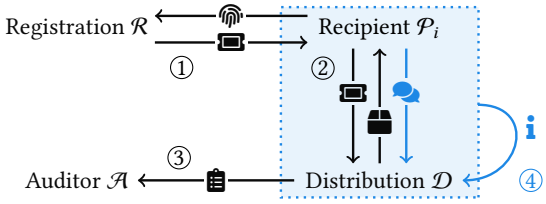

Figure 1: Parties and interactions in a digitalized humanitarian aid distribution system. Steps representing the assessment phase are shown in blue.

① **Registration** A recipient $\mathcal{P}_i$ must register once with a registration station $\mathcal{R}$ in order to be enrolled in the aid distribution program. Registration may require recipients to present identifying information (, e.g., identity documents, biometrics) to the registration station. The registration station checks the legitimacy of the request, and decides on the *aid entitlement* (whether they are eligible and the amount of aid they are entitled to) of the recipient depending on their needs. $\mathcal{R}$ then gives the recipient tokens for the corresponding entitlement (one token per period). This token can either be stored on a physical smartcard given to the recipient, or it might be a digital token to be loaded on the recipient's phone. The registration station is typically staffed by the ICRC personnel, and registration typically does not have stringent time constraints.

② **Distribution** In each distribution period, a recipient $\mathcal{P}_i$ can redeem their token for the current period at the distribution station $\mathcal{D}$. The distribution station verifies the token, and if valid, provides the recipient with the aid corresponding to their entitlement. Aid can take various forms, such as cash, vouchers, or in-kind goods (e.g., food, hygiene kits, household items). A distribution station is typically staffed by the ICRC personnel or local short-term contractors, and a distribution may be time-constrained (e.g., due to a cease-fire, or a temporary respite during a disaster).

③ **Audit** After a distribution has taken place, an independent auditor $\mathcal{A}$ audits the distribution process. The auditor receives a claimed amount of distributed aid from the distribution station. The auditor then verifies that the claimed amount of distributed aid is consistent with the inventory of aid (e.g., by inspecting a local warehouse and checking that the amount of physical aid consumed since the last distribution matches the claimed total). In the case of the ICRC, audits are mainly carried out by the Internal Audit Unit, which is independent of the department implementing aid distribution programs, and as such does not have direct access to fine-grained data from those programs.

Deployment requirements. We recall the deployment requirements that previous work [25] elicit based on their discussions with the ICRC. Our protocol also satisfies these requirements.

Scalability. *The aid distribution process should scale to a real-world deployment setting (tens of thousands of recipients over tens to hundreds of distribution periods).*

Efficient registration and distribution. *The aid distribution process should be fast, i.e., registration and distribution should not take*

much more time than the physical interaction between the recipient and the human dispensing aid.

Computationally weak recipients. *Recipients only have access to computationally weak devices.*

Weak network connectivity. *Recipients may not have any network connection. The registration and distribution station does have access to a network connection, but it may be slow and/or unreliable.*

We formalize these processes as ideal functionalities in Fig. 2; black text indicates processes present in [25], and blue text indicates modifications necessary to implement assessments (see §3).

3 Privacy-preserving assessments of humanitarian aid distribution processes

Existing security models for digitized humanitarian aid distribution [17, 25] aim to prevent an adversary from learning anything that is not strictly necessary to provide recipients with aid and to audit the distribution process. In particular, these models do not provide any means to collect and evaluate metrics.

We introduce and model an *assessment phase* for aid distribution processes (④ in Fig. 1). Such assessments are necessary for humanitarian organizations to (i) understand the effectiveness of their programs to inform future distributions; (ii) provide transparency and report impact as a measure of accountability with respect to the public, funding agencies and donors. Assessments are computed on aid recipients' data that might reveal information about their whereabouts, movement patterns, minority status, or health status. Thus, assessment computations must be carefully designed so that they do not reveal information that can result in harms for aid recipients. We collaborated with employees from the ICRC to understand the assessments relevant to humanitarian organizations and the requirements and constraints of assessment processes. This includes members of their Data Protection Office, as well as managers responsible for the implementation of aid distribution programs and staff members that deploy such programs in the field.

Assessments workflow Through months of formal discussions, we determine that assessments are typically needed after each distribution period, in order to assess the past period and inform the next one. During each period, the distribution station collects data about the distribution process (💬 in Fig. 1), which might include information about the recipients, along with information about whether they successfully received their aid. After a distribution period has concluded, the distribution station computes a pre-specified set of metrics over recipients' data for the past distribution period (④ in Fig. 1) in order to get an assessment output ℹ.

Types of assessments In our discussions with employees from the ICRC, we identify assessment relevant to their work. These fall into two categories:

Assessments for operational improvements and accountability. These assess questions pertaining to equitable aid distribution, e.g., "What is the percentage of mobility-impaired recipients that successfully received their aid?" or "Are recipients from ethnic minorities claiming their aid?". Such assessments help to understand the effectiveness of the aid distribution process. In turn, this understanding enables humanitarian organizations to improve future deployments, and to ensure that the aid is distributed fairly and equitably. These

$\mathcal{F}_{\text{RegSetup}}$

Initialize empty maps Period, Ent, Tr and Info, and empty sets $(\text{Tags}_\text{p})_{\text{p}\in[\text{P}]}$.

$\mathcal{F}^{\mathcal{O}}_{\text{Registration}}$ on input req from $\mathcal{P}_i$

1. Call $o \leftarrow \mathcal{O}(\text{req})$. If $o = \bot$, output $\bot$ to $\mathcal{P}_i$ and $\mathcal{R}$; otherwise, parse o as an entitlement ent and recipient information info.
2. For $\text{p} \in [\text{P}]$:
   (a) Generate a random tag $\text{tag}_{i,\text{p}} \leftarrow \{0,1\}^{\lambda_{\text{tag}}}$.
   (b) Set $\text{Period}[\text{tag}_{i,\text{p}}] \coloneqq \text{p}$, $\text{Ent}[\text{tag}_{i,\text{p}}] \coloneqq \text{ent}$ and $\text{Info}[\text{tag}_{i,\text{p}}] \coloneqq \text{info}$, and add $\text{tag}_{i,\text{p}}$ to Tags_p.
   (c) Set $\text{ent\_sum}_\text{p} \coloneqq 0$
3. Store $\text{Tr}[\text{tag}_{i,\text{p}'}] \coloneqq (\text{tag}_{i,1}, \ldots, \text{tag}_{i,\text{p}})$.

Output $((\text{tag}_\text{p})_{\text{p}\in[\text{P}]}, \text{ent}, \text{info})$ to $\mathcal{P}_i$.

$\mathcal{F}_{\text{RegFinish}}$

1. For $\text{p} \in [\text{P}]$:
   Initialize a map $\text{Dist}_\text{p}[\text{tag}^{(i)}] \coloneqq 0$ for each $\text{tag}^{(i)} \in \text{Tags}_\text{p}$.
2. Set $\texttt{mode} \coloneqq \texttt{distribution}$ and $\text{p} \coloneqq 1$.

$\mathcal{F}_{\text{Distribution}}$ on input $\text{tag}^{(i)}$ from $\mathcal{P}_i$

1. Abort if $\texttt{mode} \neq \texttt{distribution}$.
2. If $\text{Period}[\text{tag}^{(i)}] = \text{p} \wedge \text{Ent}[\text{tag}^{(i)}] = \text{ent} \wedge \text{Dist}_\text{p}[\text{tag}^{(i)}] \neq 1$:
   $\text{Dist}_\text{p}[\text{tag}^{(i)}] \coloneqq 1$, output $(\text{tag}^{(i)}, \text{ent}^{(i)})$ to $\mathcal{D}$.
   Otherwise output $\bot$ to $\mathcal{D}$.

$\mathcal{F}_{\text{DistFinish}}$

1. Set $\text{p} \coloneqq \text{p} + 1$. If $\text{p} > \text{P}$, set $\texttt{mode} \coloneqq \texttt{end}$.

$\mathcal{F}_{\text{Audit}}$

If $\text{ent\_sum} \coloneqq \sum_i \text{ent}^{(i)}$ matches the physical inventory at the end of the distribution period, output $\top$ to $\mathcal{A}$, and $\bot$ otherwise.

$\mathcal{F}_{\text{Simple}}$ on input $f : \mathbb{I}^{\leq \text{N}} \to \mathbb{O}$ from $\mathcal{D}$

1. Abort if $\text{p} - 1 \notin [\text{P}]$.
2. Set $x_{\text{p}-1} \coloneqq \left\{ \text{Info}[\text{tag}] \;\middle|\; \begin{array}{l} \text{tag} \in \text{Tags}_{\text{p}-1} \\ \wedge\, \text{Dist}[\text{tag}] \neq 0 \end{array} \right\}$
3. Output $f(x_{\text{p}-1})$ and $|x_{\text{p}-1}|$ to $\mathcal{D}$.

$\mathcal{F}_{\text{Full}}$ on input $f : \mathbb{I}^{\text{N}} \times ((\{0,1\})^{\text{N}})^{\text{P}} \to \mathbb{O}$ from $\mathcal{D}$

1. Abort if $\text{p} - 1 \notin [\text{P}]$.
2. For $\text{p}' = 1$ to $\text{p} - 1$:
   $x_{\text{p}'} \coloneqq \left\{ (\text{Info}[\text{tag}], (\text{Dist}[\text{tag}'])_{\text{tag}' \in \text{Tr}[\text{tag}]}) \right\}_{\text{tag} \in \text{Tags}_{\text{p}'}}$
3. Output $\{(\text{Dist}[\text{tag}'])_{\text{tag}' \in \text{Tr}[\text{tag}]}\}_{\text{tag} \in \text{Tags}_{\text{p}'}}$ to $\mathcal{A}$.
4. Output $f(x_1, \ldots, x_{\text{p}-1})$ and $|x_1|, \ldots, |x_{\text{p}-1}|$ to $\mathcal{D}$.

Figure 2: Ideal functionalities for aid distribution, parameterized by a security parameter λ, a maximum number of recipients N, and a maximum number of periods P. $\mathbb{I}$ is the recipient information space, and $\mathbb{O}$ is the space of assessment outcomes. Steps for assessments are shown in blue.

assessments also serve to complement internal and external reports (e.g., reports for donors or the public at large).

Assessments to support duty of care activities. These answer questions such as: "What is the address of all senior recipients that have not claimed aid for the last five distribution periods?". Having this information enables the organization to provide additional services to the population, beyond aid distribution, such as performing welfare checks.

Security and privacy requirements. Recipients' information, such as when and where they collect the aid, can be highly sensitive and might cause harm if revealed (e.g., reveal the whereabouts of political refugees). Except the information needed to prove legitimacy at registration, recipients' data must be kept private to themselves at all times.

Assessment privacy. *A distribution station only learns the assessment output, but does not learn additional information about individual recipients other than the information leaked by the assessment output itself.*

Assessments must be unforgeable as they influence the distribution process, affect duties of care, and foster transparency and trust in the ICRC.

Assessment unforgeability. *The assessment output is an accurate reflection of the distribution process.*

3.1 Modeling assessments

Assessments differ in their purpose, expressivity, and in the amount of input data they require. In order to accommodate these differences, we abstract these use cases into two types of assessments which we translate into ideal functionalities (shown in blue) in Fig. 2. We use ideal functionalities (rather than games) as they allow us to more directly map our formal model to requirements.

Simple assessments capture most operational and accountability-oriented assessments. They require information about aid distribution within *a single* distribution period (e.g., the fraction of recipients from a minority that successfully claimed aid in a given period). Formally, a simple assessment function f is a function from the space of recipient data $\mathbb{I}$ to a domain $\mathbb{O}$; the assessment for the p-th distribution is $f(\{x_1, \ldots, x_k\})$, where x_i is the information of a recipient $\mathcal{P}_i$ who successfully claimed aid in a given period. Most simple assessment functions of interest can be expressed as linear functions (e.g., counts, means, weighted averages).

Full assessments output information about *one or more* distribution periods up until the current one. Full assessments can be used to fulfill duty of care (e.g., output the contact of all recipients that did not claim aid for the last 3 periods), and to compute complex operational- and accountability-oriented assessments (e.g., the fraction of recipients from a minority that claimed aid in all periods up to the current one). The output of full assessments can be influenced by the information of all recipients (including those that have not participated in some distribution periods). Formally, a full assessment function f takes as input all inputs $x_i \in \mathbb{I}$ across periods P, along with bits $d_{i,\mathsf{p}}$ indicating whether the distribution for recipient $\mathcal{P}_i$ in period p was successful. Full assessments of interest can typically be expressed as linear or quadratic functions (see §6.1 for an example).

3.2 Threat model

Our main goal is to allow humanitarian organizations to collect information about the aid distribution process, while protecting the privacy of recipients. We now describe the threat model against which such protection should be provided and how it deviates from previous work on digitalized aid distribution.

Recipients. As recipients want to protect their own privacy, we model recipients as honest. However, to model the risk of coercion or bribery (e.g., by local authorities), we allow a fraction t of recipients to be statically and passively corrupted by the adversary.

Registration station. The registration station is typically staffed by the ICRC personnel, and registration only happens once. We therefore model this station as honest-but-curious, i.e., it follows the protocol, but it may try to learn additional information about the recipients. We note that this assumption is somewhat necessary, as assuming a fully malicious registration station does not allow for a meaningful aid distribution functionality [8, 17, 25].

Auditor. The auditor is a trusted independent party. It is tasked with auditing the distribution process. As in previous work, we model it as honest-but-curious [8, 17, 25].

Distribution station. The distribution station may be staffed by the ICRC personnel, or by external providers (e.g., local short-term contractors), and might be performed over longer period of times. While these personnel could in principle be considered honest-but-curious, humanitarian aid distribution is deployed in unstable environments in which the capabilities of the adversary may evolve over time (e.g., a militia taking control of a region of conflict deciding not to respect previous agreements with humanitarian organizations and requesting access to data that can compromise the privacy of recipients which before was considered confidential).

One way to address this reality is to design for the strongest possible threat model: a fully malicious distribution station. This is the route taken by the authors of previous privacy-preserving aid distribution systems upon which we build [8, 17, 25]. For relatively simple functionalities (e.g., an all-or-nothing information disclosure model [25] or a straightforward fundamental leakage pattern [17]), fully malicious security can be achieved without a significant impact on efficiency, and thus designing solely for this strong threat model is feasible. However, when the desired functionality requires the disclosure of fine-grained information—as it is the case of assessments—designing for the strongest threat model may lead to overly inefficient protocols that would impact the efficacy of the aid-distribution process. To maximize the effectiveness of humanitarian aid distribution, we consider two threat models, which differ in the adversarial capabilities granted to the distribution station.

In the first threat model, we assume the distribution station to be honest-but-curious. This corresponds to a deployment where the risk of an armed take-over or coercion of the distribution is low, and where the risks are opportunistic threats to the privacy of recipients (e.g., by internal staff or external providers). In the second threat model, we model the distribution station as actively malicious. Our goal is to enable the ICRC to assess the risk for each deployment scenario, and to maximize the efficiency of the

system in that scenario, thereby maximizing the efficacy of the aid distribution process and minimizing the risk of harm.

Yet, deploying the weaker protocol may pose serious risks in the event that the adversary changes from honest to malicious. We thus introduce two new requirements that ensure that the protocols we design are robust (even if not perfectly secure) against such changes:

Threat model agility. *In the case of a change from a weaker to a stronger threat model, it is possible to either (i) strengthen an already deployed protocol to be robust against the new, stronger threat model, or (ii) safely shut down the assessments protocol without causing significant harm.*

Graceful degradation. *When an adversary is stronger than assumed in the threat model (either temporarily or durably), recipients do not experience significant harm, i.e., their safety does not fail catastrophically, but might worsen slightly.*

Threat model agility is a *system property*; when showing that a protocol satisfies this property, we analyze either (i) what additional setup is needed to deploy a strengthened protocol, or (ii) whether the protocol can be shut down, and what are the harms resulting of such a shutdown. In contrast, Graceful degradation is a *security property*; when showing that a protocol satisfies this property, we analyze the additional leakage that results from the stronger threat model and bound it against the original threat model.

4 Building privacy-preserving assessments

We describe a generic framework for privacy-preserving assessments, which can be integrated into existing privacy-preserving aid distribution systems. We show that standard cryptographic primitives are insufficient for our use. In order to realize privacy-preserving assessments, we introduce *one-time* functional encryption (§4.1), in which a function f can only be evaluated once. We then use this building block to construct protocols for simple and full assessments, for both semi-honest (§4.2) and malicious (§4.3) threat models.

An insecure strawman solution.A natural building block for conducting secure assessments may appear to be functional encryption (FE). In FE, a trusted dealer generates a public key pk along with a secret key sk_f for a function f. Each party encrypts a message pt_i under pk to produce a ciphertext ct_i. Then, a party that holds sk_f can evaluate f on ciphertexts ct_i of multiple parties to receive $f(\mathsf{pt}_1, \ldots, \mathsf{pt}_n)$, without learning additional information about the inputs pt_i. There exist direct constructions of FE, but for efficiency reasons, FE-like functionalities are also often realized by secure multi-party computation (MPC) with a designated output party. We show why such a primitive is, by itself, insufficient for our setting.

Applying FE in our setting would look as follows: the trusted dealer is the registration station $\mathcal{R}$, which sends the secret key sk_f for a function f to the distribution station $\mathcal{D}$. During distribution, recipients encrypt their input and send them to $\mathcal{D}$. After a distribution period is complete, $\mathcal{D}$ evaluates f on the encrypted inputs and recovers the plaintext assessment results.

A malicious distribution station $\mathcal{D}$ can attack the privacy and unforgeability of assessments (see below), and this construction does not satisfy the Graceful degradation requirement.

Attack on assessment privacy. A malicious $\mathcal{D}$ can run the FE scheme multiple times with different inputs. It can then learn how inputs from different recipients affect the resulting output, revealing information about the recipient. For example, if it is able to observe how the output of the FE scheme changes as new inputs from recipients are added, it can learn which recipient's input caused the output to change—and therefore learn the input of this recipient. In the worst case, a malicious $\mathcal{D}$ can evaluate the FE functionality repeatedly for each recipient's input separately to learn their inputs.

Attack on assessment unforgeability. A malicious distribution station can also arbitrarily withhold or duplicate encrypted inputs before a FE evaluation. This introduces issues of correctness and integrity. This attack is possible even if the FE scheme is non-malleable and secure against active adversaries.

Note that even advanced FE schemes are vulnerable to these attacks. To thwart the attack on Assessment privacy, it is critical that the FE scheme used to construct the protocol be *one-time*, i.e., the function cannot be evaluated more than once per period. In order to address this limitation of standard FE, we formalize the notion of *one-time functional encryption* (1FE), which we will use as an intermediate target in our constructions. Thwarting the attack on Assessment unforgeability requires removing the ability of the adversary to adaptively choose the inputs to be evaluated. We will address this challenge in our protocol against a malicious adversary (§4.3), using *predicate* 1FE. We give concrete instantiations of 1FE from standard cryptographic primitives in §5.

4.1 One-time functional encryption

As a core building block for our protocols, we introduce the notion of *one-time functional encryption* (1FE) (Fig. 3). As in standard functional encryption, a 1FE scheme consists of the following four algorithms:

- $\mathcal{F}_{\mathsf{1FE.Setup}}$ samples a public key $\mathsf{pk}_{\mathsf{1FE}}$ and a master secret key $\mathsf{msk}_{\mathsf{1FE}}$;
- $\mathcal{F}_{\mathsf{1FE.KeyGen}}$ samples an evaluation key $\mathsf{sk}_{\mathsf{1FE}}$ for a function f;
- $\mathcal{F}_{\mathsf{1FE.Enc}}$ encrypts a plaintext pt under $\mathsf{pk}_{\mathsf{1FE}}$, and outputs a ciphertext ct;
- $\mathcal{F}_{\mathsf{1FE.Eval}}$ takes a set of ciphertexts along with $\mathsf{sk}_{\mathsf{1FE}}$, and outputs the evaluation of f on the corresponding plaintexts.

In 1FE, we additionally require that each plaintext be encrypted using a given tag id, and restrict 1FE.Eval to allow only a single evaluation per tag.

Relationship to standard FE. Our definition is a strict generalization of standard (multi-input) functional encryption. We consider the FE setting with a single public key (rather than a public key per input party), which is a better fit for our setting. In addition, we augment the 1FE definition in the two following ways to support our use case.

Plaintext inputs We augment the evaluation functionality to accept plaintext inputs. By itself, this does not expand the expressivity of the scheme, but it allows for more expressive predicates (since these are computed on public data). In the cryptographic primitives we use to realize 1FE (§5), plaintext inputs also enable more efficient constructions.

Predicates We allow a predicate φ to be associated with each function f, which must be satisfied by the inputs in order for the

$\mathcal{F}_{\text{1FE.Setup}}(1^\lambda)$

1. Initialize $\mathsf{SKeys} \coloneqq \emptyset$; $\mathsf{Id} \coloneqq \emptyset$; $\mathsf{Inputs} \coloneqq [\,]$.
2. Output $(\mathsf{pk}_{\text{1FE}}, \mathsf{msk}_{\text{1FE}}) \leftarrow (\{0,1\}^\lambda)^2$ to calling party.

$\mathcal{F}_{\text{1FE.KeyGen}}(\mathsf{msk}, f, \varphi)$

1. If $\mathsf{msk} \neq \mathsf{msk}_{\text{1FE}}$, abort.
2. Sample $\mathsf{sk}_{\text{1FE}} \leftarrow \{0,1\}^\lambda$.
3. Set $\mathsf{SKeys} \coloneqq \mathsf{SKeys} \cup \{(f, \varphi, \mathsf{sk}_{\text{1FE}})\}$.
4. Output sk_{1FE} to calling party.

$\mathcal{F}_{\text{1FE.Enc}}(\mathsf{pk}, \mathsf{id}, \mathsf{pt})$

1. If $\mathsf{pk} \neq \mathsf{pk}_{\text{1FE}}$, abort.
2. Sample $\mathsf{ct} \leftarrow \{0,1\}^\lambda$ and set $\mathsf{Inputs}[\mathsf{ct}] \coloneqq (\mathsf{pt}, \mathsf{id})$.
3. Output ct to calling party.

$\mathcal{F}_{\text{1FE.Eval}}(\mathsf{sk}, f, \varphi, (\mathsf{ct}_i)_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]})$

1. If $(f, \varphi, \mathsf{sk}) \notin \mathsf{SKeys}$ or $\varphi((\mathsf{ct}_i)_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]}) = 0$, abort.
2. For $i \in [m]$:
   If $\mathsf{Inputs}[\mathsf{ct}_i] = \perp$, abort.
   Otherwise, set $(\mathsf{pt}_i, \mathsf{id}_i) \coloneqq \mathsf{Inputs}[\mathsf{ct}_i]$.
3. If id_i are not all the same, abort.
4. If $(f, \varphi, \mathsf{id}) \in \mathsf{Id}$, abort.
5. Set $\mathsf{pt}' \coloneqq f(\mathsf{pt}_1, \ldots, \mathsf{pt}_{m+n})$ and $\mathsf{Id} \coloneqq \mathsf{Id} \cup \{\mathsf{id}\}$.
6. Output pt' to calling party.

Figure 3: Ideal functionalities for one-time multi-input predicate functional encryption with plaintext inputs. We write $\mathcal{F}(\text{args})$ to denote a party invoking functionality $\mathcal{F}$ with arguments args**.**

evaluation to succeed. This predicate is *public*, in the sense that it only depends on the ciphertexts corresponding to private inputs, not on the underlying plaintexts.

Supported functions Our definition does not explicitly restrict the class of functions that a 1FE scheme should support, and we provide instantiations of 1FE for arbitrary arithmetic or boolean circuits in §5.

We design protocols for 1FE with various expressivity and efficiency profiles in §5. In the rest of this section, we assume these are implemented and leverage the 1FE primitive to design protocols for privacy-preserving assessments. For the sake of exposition, in this section we also assume that there is a single function to be evaluated. We discuss a straightforward extension of our protocols to handle multiple assessment functions in §B.2.

In this section, we show that our protocols securely realize the ideal functionalities for assessments (Fig. 2) in the $\mathcal{F}_{\text{1FE}}$-hybrid model, i.e., where protocols invoke the 1FE ideal functionality as a subroutine. Our proofs proceed by showing that the views of an adversary in the ideal world and in this hybrid world are (computationally) indistinguishable. We note that we target simulation security, but not Universally Composable (UC) security.

4.2 Secure assessments against passive stations

We design protocols secure against honest-but-curious distribution stations, for simple and complex assessments (Figs. 4a and 4b).

4.2.1 Protocol for simple assessments. In our protocol for simple assessments (shown in Fig. 4a), we directly leverage 1FE to evaluate the assessment function (in essence, we use the strawman construction, but use 1FE instead of FE). The registration station acts as a dealer and generates a public key pk_{1FE} along with a secret key sk_{1FE} for an assessment function f; it then distributes the public key to all recipients, and the secret key to the distribution station. During a distribution phase (after running the protocol required to claim their aid, but just before receiving their aid), each recipient $\mathcal{P}_i$ sends their inputs to the one-time functional encryption functionality $\mathcal{F}_{\text{1FE.Enc}}$, and sends the resulting ciphertext to the distribution station $\mathcal{D}$. At the end of a distribution period, $\mathcal{D}$ forwards all ciphertexts to the 1FE evaluation functionality $\mathcal{F}_{\text{1FE.Eval}}$ to learn the assessment output.

Lemma 4.1. *$\Pi^{\text{hbc}}_{\text{Simple}}$ (Fig. 4a) securely realizes $\mathcal{F}_{\text{Simple}}$ against honest-but-curious distribution stations in the $\mathcal{F}_{\text{1FE}}$-hybrid model.*

Proof sketch. This follows directly from the 1FE functionality: for any passive adversary against $\mathcal{F}_{\text{Simple}}$, the views of an adversary interacting with $\mathcal{F}_{\text{Simple}}$ and $\Pi^{\text{hbc}}_{\text{Simple}}$ in the $\mathcal{F}_{\text{1FE.Eval}}$-hybrid model are equivalent by construction. □

4.2.2 Protocol for full assessments. In some cases, it is important to evaluate *full* assessments over the inputs of *all* recipients, regardless of whether they interacted with the distribution station or not. For example, the protocol should reveal the contact information of all senior recipients that have not claimed aid for the last five distribution periods, in order to perform a welfare check. In this example, the assessment function also depends on the success or failure of the distribution process for each recipient, over multiple earlier distribution periods. Extending the protocol from simple to full assessments runs into three challenges.

Challenge 1: Distribution status The distribution station should provide inputs about the success or failure of the aid distribution for each recipient. To accommodate this fact, we require the distribution station to provide this input to the 1FE functionality, by leveraging the plaintext inputs capability of our 1FE scheme.

Challenge 2: Non-interactivity The distribution station should only learn something about a recipient if and only if the recipient did *not* interact with the station. This requires recipients to register their (1FE-encrypted) information upfront with the registration station. The registration station can forward these encrypted inputs to the evaluating party. During distribution, there is no longer a need for recipients to communicate any additional information to the distribution station $\mathcal{D}$. Note that this shifts the communication between recipients and the distribution station from the distribution phase to the registration phase. This might be a boon in distribution scenarios where the registration phase can proceed at a slower pace than the distribution phase (e.g., for a registration phase spread over multiple days, and a distribution including perishable goods and/or under time pressure).

(a) Protocols for simple assessments; $\Pi^{\text{hbc}}_{\text{Simple}}$ (in black) is against an honest-but-curious distribution station, and $\Pi^{\text{mal}}_{\text{Simple}}$ (in black and red) is against a malicious distribution station.

AssessmentsRegSetup(f)

$\mathcal{R} : (\text{pk}_{\text{1FE}}, \text{msk}_{\text{1FE}}) \leftarrow \mathcal{F}_{\text{1FE.Setup}}(1^\lambda); \text{sk}_{\text{1FE}} \leftarrow \mathcal{F}_{\text{1FE.KeyGen}}(\text{msk}_{\text{1FE}}, f)$

$\mathcal{A} : (\text{pk}_{\text{PKE}}, \text{sk}_{\text{PKE}}) \leftarrow \text{PKE.Setup}(1^\lambda)$

$\mathcal{A} : (\text{vk}_{\text{SIG}}, \text{sk}_{\text{SIG}}) \leftarrow \text{SIG.Setup}(1^\lambda)$

$\mathcal{A} \rightarrow \mathcal{R} : \text{pk}_{\text{PKE}}, \text{vk}_{\text{SIG}}$

AssessmentsReg

$\mathcal{R} \rightarrow \mathcal{P}_i : \text{pk}_{\text{1FE}}, \text{pk}_{\text{PKE}}, \text{vk}_{\text{SIG}}$

$\mathcal{P}_i$: For $\text{p} \in [\text{P}]$:
  For $\ell \in [\text{ent}_i]$:
    $\text{stag}_{i,\text{p},\ell} \leftarrow \{0,1\}^{\lambda_{\text{tag}}}$

$\mathcal{P}_i \rightarrow \mathcal{R} : \{\text{PKE.Enc}(\text{pk}_{\text{PKE}}; \text{stag}_{i,\text{p},\ell})\}_{\text{p}\in[\text{P}],\ell\in[\text{ent}_i]}$

AssessmentsRegFinish

$\mathcal{R} \rightarrow \mathcal{D} : \text{sk}_{\text{1FE}}$

$\mathcal{R} \rightarrow \mathcal{A} : \{\text{PKE.Enc}(\text{pk}_{\text{PKE}}; \text{stag}_{i,\text{p},\ell})\}_{i\in[\text{N}],\text{p}\in[\text{P}],\ell\in[\text{ent}_i]}$

AssessmentsCollect

$\mathcal{P}_i : \text{ct}^{\text{1FE}}_{i,\text{p}} \leftarrow \mathcal{F}_{\text{1FE.Enc}}(\text{pk}_{\text{1FE}}; \text{info}_i; \text{p})$

$\mathcal{P}_i : \text{ct}^{\text{PKE}}_{i,\text{p},1} \leftarrow \text{PKE.Enc}(\text{pk}_{\text{PKE}}; (\text{stag}_{i,\text{p},1}, \text{ct}^{\text{1FE}}_{i,\text{p}}))$:

$\mathcal{P}_i$: For $\ell \in [\text{ent}_i - 1]$:
  $\text{ct}^{\text{1FE}}_{i,\text{p},\ell} \leftarrow \mathcal{F}_{\text{1FE.Enc}}(\text{pk}_{\text{1FE}}; \perp_f; \text{p})$
  $\text{ct}^{\text{PKE}}_{i,\text{p},\ell} \leftarrow \text{PKE.Enc}(\text{pk}_{\text{PKE}}; (\text{stag}_{i,\text{p},\ell}, \text{ct}^{\text{1FE}}_{i,\text{p},\ell}))$

$\mathcal{P}_i \rightarrow \mathcal{D} : \text{ct}^{\text{1FE}}_{i,\text{p}} \,/\, \text{ct}^{\text{PKE}}_{i,\text{p},1}, \ldots, \text{ct}^{\text{PKE}}_{i,\text{p},\text{ent}_i}$

AssessmentsEval(f)

$\mathcal{D} \rightarrow \mathcal{A} : (\text{ct}^{\text{PKE}}_{i,\text{p},\ell})_{i\in[k],\ell\in[\text{ent}_i]}$

$\mathcal{A}$: For $i \in [\text{E}] : (\text{stag}_i, \text{ct}^{\text{1FE}}_i) \coloneqq \text{PKE.Dec}(\text{sk}_{\text{PKE}}; \text{ct}^{\text{PKE}}_i)$

$\mathcal{A}$: If $|\{\text{stag}_i\}_i| = \text{E} \wedge \{\text{stag}_i\}_i \subseteq \text{Tags}_{\text{p}} \wedge \text{Audit}_{\text{p}}(\text{E}) = 1$:

- Compute a canonical representation m of the set of 1FE inputs $\{\text{ct}^{\text{1FE}}_i\}_{i\in[\text{E}]}$.
- $\sigma \coloneqq \text{SIG.Sign}(\text{sk}_{\text{SIG}}; m)$.
- $\mathcal{A} \rightarrow \mathcal{D} : \sigma$

$\mathcal{D} : \text{res} \leftarrow \mathcal{F}_{\text{1FE.Eval}}(\text{sk}_{\text{1FE}}, f^*, \varphi_{\text{SIG}}, \text{p}, \text{ct}^{\text{1FE}}_{1,\text{p}}, \ldots, \text{ct}^{\text{1FE}}_{k,\text{p}}, \sigma)$

$\mathcal{D}$ outputs res.

(b) Protocols for full assessments; $\Pi^{\text{hbc}}_{\text{Full}}$ (in black) is against an honest-but-curious distribution station, and $\Pi^{\text{mal}}_{\text{Full}}$ (in black and red) is against a malicious distribution station.

AssessmentsRegSetup(f)

$\mathcal{R} : (\text{pk}_{\text{1FE}}, \text{msk}_{\text{1FE}}) \leftarrow \mathcal{F}_{\text{1FE.Setup}}(1^\lambda); \text{sk}_{\text{1FE}} \leftarrow \mathcal{F}_{\text{1FE.KeyGen}}(\text{msk}_{\text{1FE}}, f)$

$\mathcal{A} : (\text{pk}_{\text{PKE}}, \text{sk}_{\text{PKE}}) \leftarrow \text{PKE.Setup}(1^\lambda)$

$\mathcal{A} : (\text{vk}_{\text{SIG}}, \text{sk}_{\text{SIG}}) \leftarrow \text{SIG.Setup}(1^\lambda)$

$\mathcal{A} \rightarrow \mathcal{R} : \text{pk}_{\text{PKE}}, \text{vk}_{\text{SIG}}$

AssessmentsReg

$\mathcal{R} \rightarrow \mathcal{P}_i : \text{pk}_{\text{1FE}}, \text{pk}_{\text{PKE}}, \text{vk}_{\text{SIG}}$

$\mathcal{P}_i$: For $\text{p} \in [\text{P}]$:
  $\text{ct}^{\text{1FE}}_{i,\text{p}} \leftarrow \mathcal{F}_{\text{1FE.Enc}}(\text{pk}_{\text{1FE}}; \text{info}_i; \text{p})$
  $\text{ct}^{\text{PKE}}_{i,\text{p},1} \leftarrow \text{PKE.Enc}(\text{pk}_{\text{PKE}}; (\text{tag}_{i,\text{p},\ell}, \text{ct}^{\text{1FE}}_{i,\text{p}}))$
  For $\ell \in [\text{ent}_i - 1]$:
    $\text{ct}^{\text{1FE}}_{i,\text{p}} \leftarrow \mathcal{F}_{\text{1FE.Enc}}(\text{pk}_{\text{1FE}}; \perp_f; \text{p})$
    $\text{ct}^{\text{PKE}}_{i,\text{p},\ell} \leftarrow \text{PKE.Enc}(\text{pk}_{\text{PKE}}; (\text{tag}_{i,\text{p},\ell}, \text{ct}^{\text{1FE}}_{i,\text{p},\ell}))$

$\mathcal{P}_i \rightarrow \mathcal{R} : (\text{ct}^{\text{1FE}}_{i,\text{p},\ell})_{\text{p}\in[\text{P}],\ell\in[\text{ent}_i]}$

AssessmentsRegFinish

$\mathcal{R} \rightarrow \mathcal{D} : \text{sk}_{\text{1FE}}$

$\mathcal{R} \rightarrow \mathcal{A} : \{\text{ct}^{\text{PKE}}_{i,\text{p},\ell}\}_{i\in[\text{N}],\text{p}\in[\text{P}],\ell\in[\text{ent}_i]}$

AssessmentsCollect

As part of the standard distribution process, $\mathcal{P}_i$ sends $\{\text{tag}_{i,\text{p},\ell}\}_{\ell\in[\text{ent}_i]}$ to $\mathcal{D}$.

AssessmentsEval(f)

$\mathcal{D} \rightarrow \mathcal{A} : T_{\text{p}} \coloneqq \left\{\text{tag}_{i,\text{p},\ell} \mid \text{Dist}_{\text{p}}[\text{tag}_{i,\text{p},\ell}] = 1\right\}$

$\mathcal{A}$: For $i \in [\text{E}] : (\text{tag}_i, \text{ct}^{\text{1FE}}_i) \coloneqq \text{PKE.Dec}(\text{sk}_{\text{PKE}}; \text{ct}^{\text{PKE}}_i)$

$\mathcal{A}$: Computes a canonical representation of the 1FE inputs

$$m_{\text{p}} \coloneqq ((\text{ct}^{\text{1FE}}_i)_{i\in[\text{E}]}, (\mathbb{1}[\text{tag}_i \in T_{\text{p}}])_{i\in[\text{E}]})$$

such that ct^{1FE}_i matches the distribution outcome value.

$\mathcal{A}$: If $\text{Audit}_{\text{p}}(|T_{\text{p}}|) = 1$: $\sigma_{\text{p}} \coloneqq \text{SIG.Sign}(\text{sk}_{\text{SIG}}; m_{\text{p}})$

$\mathcal{A} \rightarrow \mathcal{D} : m_{\text{p}}, \sigma_{\text{p}}$

$\mathcal{D} : \text{res} \leftarrow \mathcal{F}_{\text{1FE.Eval}}(\text{sk}_{\text{1FE}}, f^*, \varphi_{\text{SIG}}, \text{p}, (m_p)_{p\in[\text{p}]}, (\sigma_p)_{p\in[\text{p}]})$

$\mathcal{D}$ outputs res.

$$f^*((x_{1,1}, \ldots, x_{k,1}), \ldots, (x_{1,\text{p}}, \ldots, x_{k,\text{p}}), \sigma_1, \ldots, \sigma_{\text{p}}) = f((x_{1,1}, \ldots, x_{k,1}), \ldots, (x_{1,\text{p}}, \ldots, x_{k,\text{p}}))$$

$$\varphi_{\text{SIG}}((x_{1,1}, \ldots, x_{k,1}), \ldots, (x_{1,\text{p}}, \ldots, x_{k,\text{p}}), \sigma_1, \ldots, \sigma_{\text{p}}) = 1 \Leftrightarrow \forall i \in [\text{p}] : \text{SIG.Vfy}(\text{vk}_{\text{SIG}}; (x_{1,i}, \ldots, x_{k,i}), \sigma_i) = 1$$

Figure 4: Protocols for simple (4a) and full assessments (4b). Additions necessary to achieve security against a malicious distribution station are shown in red. P is the number of periods, N is the number of recipients, and E is the total sum of entitlement distributed in a given period.

Challenge 3: Matching The distribution station needs to match a given ciphertext (as computed by a recipient during registration) and the distribution statuses for different periods of that same recipient (as known by the distribution station). We leverage pseudorandom tags which are sent by recipients to the distribution station when claiming aid, and which are already present in the privacy-preserving aid distribution protocol without assessments that we build upon [25]. Concretely, each recipient $\mathcal{P}_i$ generates pseudorandom tags $(\mathsf{tag}_{i,\mathsf{p}})_{\mathsf{p}\in[\mathsf{P}]}$ (one per period), and sends them to the registration station, along with P 1FE encryptions $(\mathsf{ct}^{\mathsf{1FE}}_{i,\mathsf{p}})_{\mathsf{p}\in[\mathsf{P}]}$ of their (singular) input. During distribution, a $\mathcal{P}_i$ sends its tag $\mathsf{tag}_{i,\mathsf{p}}$ to the distribution station when claiming aid. After a distribution period, the distribution station sends all tags $\{\mathsf{tag}_{i,\mathsf{p}}\}_{i\in[k]}$ received in the past period to the auditor $\mathcal{A}$. The auditor can match a ciphertext $\mathsf{ct}^{\mathsf{1FE}}_i$ with the corresponding distribution status for each period up to the current one, and answers with the matched 1FE inputs $((\mathsf{ct}^{\mathsf{1FE}}_{i,\mathsf{p}})_{i\in[\mathsf{N}]}, (d_{i,1})_{i\in[\mathsf{N}]}, \ldots, (d_{i,\mathsf{p}})_{i\in[\mathsf{N}]})$, where $d_{i,\mathsf{p}}$ is a bit denoting whether $\mathcal{P}_i$'s distribution process was successful in period p.

Note that the distribution station learns the distribution traces $(d_{i,1}, \ldots, d_{i,\mathsf{p}})$ for all recipients, however, this protocol still provides *privacy* for recipients, since for the distribution station recipients and such traces are *unlinkable*. This leakage also appears in some instantiations of Wang et al.'s protocols [25], notably their phone-based instantiations relying on either Bluetooth Low Energy (BLE) or WiFi (without MAC rerandomization) for data transmission. Our protocol introduces an additional information leakage towards the auditor, who also learns the distribution traces of all recipients. Note that this additional leakage is inherent to the assessment functionality: if assessments requires linking across multiple periods, then this linking operation needs to be performed in the protocol.

Lemma 4.2. *$\Pi^{\mathsf{hbc}}_{\mathsf{Full}}$ (Fig. 4b) securely realizes $\mathcal{F}_{\mathsf{Full}}$ against an honest-but-curious distribution station in the $\mathcal{F}_{\mathsf{1FE}}$-hybrid model.*

Proof sketch (full proof in §C.1). A distinguisher between $\mathcal{F}_{\mathsf{Full}}$ and $\Pi^{\mathsf{hbc}}_{\mathsf{Full}}$ in the $\mathcal{F}_{\mathsf{1FE}}$-hybrid model yields an attacker against the IND-CCA2-security of PKE. The adversary's view in both worlds are computationally indistinguishable. □

4.2.3 Threat model agility. $\Pi^{\mathsf{hbc}}_{\mathsf{Simple}}$ does not unconditionally fulfill the Threat model agility requirement: as recipients only communicate with the distribution station, if the station becomes actively malicious, recipients have no way to know that they should now refuse to engage in the assessment process. On the other hand, $\Pi^{\mathsf{hbc}}_{\mathsf{Full}}$ does fulfill this requirement, since the distribution station needs to communicate with the auditor, which is able to safely shut down the protocol. In order to make both protocols fulfill the Threat model agility requirement, it suffices that the protocol implementing the 1FE functionality implement a "kill-switch" functionality, which allows the ICRC headquarters (or a trusted observer of the geopolitical situation) to shut down the assessment process in case of sudden change in distribution circumstances. Our protocols for 1FE (§5) naturally lend themselves to such a functionality (see §B.5).

4.2.4 Graceful degradation. In both protocols in this section, an honest-but-curious distribution only learns the intended assessment output. As we discuss in §3.2, because aid distribution is sometimes deployed in inherently unstable regions, the assumption that the distribution station is not actively malicious might be invalidated. For example, if an armed militia takes over the distribution station. If the distribution station is malicious (or even covert), protocols designed for a weaker threat model may no longer securely realize their corresponding ideal functionalities. Contrary to the strawman solution based on FE, the privacy violation of recipients' information is bounded thanks to our use of 1FE. To see why, we recall the attack on recipient privacy for the strawman scheme: an actively malicious $\mathcal{D}$ can run the evaluation multiple times with different ciphertext inputs in order to learn additional information about recipients' private information. When 1FE is in place, $\mathcal{D}$ can only ever learn a *single* function evaluation's worth of information. This leakage can be additionally mitigated by either thresholding the function (§B.3) and/or by leveraging differential privacy (§B.4).

4.3 Protocols against active stations

As explained in §4.2.4, $\Pi^{\mathsf{hbc}}_{\mathsf{Simple}}$ and $\Pi^{\mathsf{hbc}}_{\mathsf{Full}}$ are not secure against actively malicious distribution stations. In this section, we design protocols for both simple and full assessments that are secure in such a threat model. To achieve this, we introduce a sub-protocol that cryptographically binds the distribution station's message to its physical aid inventory (§4.3.2), which ensures that any malicious behavior of the distribution station is discoverable by the auditor.

4.3.1 Threat Model. We consider a fully malicious distribution station, which can actively deviate from the protocol, and in particular is able to drop or duplicate recipient messages before forwarding them on to the auditor or to the 1FE evaluation functionality. We note that the 1FE functionality prevents even an actively malicious distribution station from crafting completely new inputs, as well as invoking the 1FE functionality more than once per tag. This threat model corresponds to a deployment with a risk of an armed take-over or coercion of the distribution station.

4.3.2 Binding Cryptographic Material to Inventory. In order to force an actively malicious distribution station to behave honestly, it suffices to ensure that the inputs to the 1FE are consistent with the real-world distribution outcome. For both simple and full assessments, we leverage the fact that the aid distribution process inherently involves audits of physical aid inventory to cryptographically bind 1FE inputs to this auditable quantity.

Our binding protocols share the following high-level structure: During registration, each recipient $\mathcal{P}_i$ generates random tags for each period, encrypts them (under the auditor's public key), and sends the resulting ciphertexts to the registration station. These tags will later allow the auditor to verify that a given ciphertext is intended for a given period. After the registration finishes, the registration station forwards all ciphertexts to the auditor.

After a distribution period, the auditor knows the following information (through messages from recipients and from the stations):

- the 1FE inputs (either only ciphertexts, or both ciphertexts and public plaintext inputs) for the current period (controlled by the malicious distribution station);
- the random tag associated with each 1FE input;
- the recipient entitlement associated with each 1FE input;
- the physical inventory of distributed aid.

Using this information, the auditor checks whether (i) all 1FE ciphertexts originate from recipients; (ii) no 1FE input corresponding to a recipient has been duplicated; (iii) all 1FE ciphertexts correspond to the current period; (iv) the total entitlement corresponding to the 1FE inputs matches the physical inventory. If all these conditions are met, the auditor certifies the set of inputs by signing this set, and sends the signature to the distribution station. Finally, the distribution station can use this signature to invoke the 1FE evaluation functionality, with the restriction that a signature verification predicate must be satisfied on the inputs.

This framework is general enough to accommodate both simple and full assessments. However, in a straightforward instantiation of this framework, the auditor learns the breakdown of entitlements for each period. This is a deviation from the protocol of Wang et al. [25], where the auditor only learns the total entitlement distributed. While this might not lead to concrete harms in practice, we deem it undesirable, and we strive to achieve a leakage profile that is as close as possible to Wang et al. In order to do so, we split the entitlement sent by the recipient in the collection phase into equal units (which are the same across all recipients), and provide one real and $\mathsf{ent} - 1$ dummy ciphertexts (encrypting a special dummy symbol $\perp_f$ that is discarded by f).

4.3.3 Simple assessments. The protocol for simple assessments is shown in Fig. 4a. In the registration phase, each recipient $\mathcal{P}_i$ generates a tag $\mathsf{stag}_{i,\mathsf{p}}$ per period p, encrypts it under the auditor's public key, and sends this ciphertext to the registration station $\mathcal{R}$. After registration, $\mathcal{R}$ sends these ciphertexts to the auditor, which decrypts and stores all tags ordered by their period. During collection, recipients send their encrypted 1FE input $\mathsf{ct}^{\mathsf{1FE}}_{i,\mathsf{p}}$ along with their secret tag $\mathsf{stag}_{i,\mathsf{p}}$ and their entitlement ent_i to $\mathcal{A}$, via the distribution station $\mathcal{D}$. Before the evaluation, $\mathcal{A}$ checks whether the secret tags are distinct and consistent with the current period, and that the physical inventory matches the alleged quantity of distributed aid, signs the set of 1FE inputs, and sends the signature to $\mathcal{D}$.

Lemma 4.3. $\Pi^{\mathsf{mal}}_{\mathsf{Simple}}$ *(Fig. 4a) securely realizes* $\mathcal{F}_{\mathsf{Simple}}$ *against a malicious distribution station in the* $\mathcal{F}_{\mathsf{1FE}}$*-hybrid model.*

Proof sketch (full proof in §C.2). We first go through a series of hybrids where we replace all ciphertexts with random values and sending ciphertexts by secure channels. The ideal and real worlds are equivalent up until the event where the distribution station $\mathcal{D}$ distributes aid to a recipient but either (i) drops that recipient's message to the 1FE functionality but forwards their message to $\mathcal{A}$, or (ii) forwards the recipient's message to the 1FE functionality but drops it for $\mathcal{A}$, or (iii) drops both recipient messages. In cases (i) and (ii), the signature predicate will not be satisfied, and the 1FE functionality will abort. In case (iii), there will be a discrepancy between the total entitlement registered with the external party and the physical inventory. The distribution station will not be able to guess a valid secret tag that it can pair with an entitlement and user ciphertext to make up the discrepancy. Thus, the auditor will notice the discrepancy and will not notarize the inputs. □

4.3.4 Full assessments. The protocol for full assessments is shown in Fig. 4b. In the distribution phase, no additional data needs to be collected, as recipients' encrypted inputs are already collected during registration. Upon a successful audit, the auditor sends a signature of the 1FE inputs for the current period to the distribution station. This is the only change needed to achieve security against a malicious distribution station.

Lemma 4.4. $\Pi^{\mathsf{mal}}_{\mathsf{Full}}$ *(Fig. 4b) securely realizes* $\mathcal{F}_{\mathsf{Full}}$ *against a malicious distribution station in the* $\mathcal{F}_{\mathsf{1FE}}$*-hybrid model.*

Proof sketch (full proof in §C.3). This follows from the IND-CCA2-security of PKE, and the SUF-CMA-security of SIG. □

5 Realizing 1FE

Constructing 1FE requires (i) guaranteeing the one-time property, (ii) achieving security against an actively malicious and adaptive adversary, and (iii) supporting expressive classes of functions and public predicates. Challenge (i) is related to one-time programs and one-time memory [10, 27] and proof-of-erasure [22]: 1FE can be seen as a one-time program where (part of) the program is additionally required to remain hidden from the adversary. Such constructions rely either on quantum primitives, strong assumptions on storage limitations of the adversary, or hardware-backed one-time memory, all of which are impractical in our setting. More practical approaches to realizing 1FE could plausibly rely on trusted hardware, or on a standard FE scheme together with a trusted party. However, both of these approaches are inadequate in our setting, as we discuss in the following strawman constructions.

Strawman construction #1: trusted hardware One approach to realize the 1FE functionality $\mathcal{F}_{\mathsf{1FE}}$ is to leverage hardware trust to realize the functionality $\mathcal{F}_{\mathsf{1FE}}$ using a Trusted Execution Environment (TEE). Approaches relying on hardware trust have been proposed for one-time programs [22, 27]. However, these approaches rely on non-volatile, non-reprogrammable memory, which is not available in commercially available TEEs [27, App. A]. Further, enforcing the one-time property requires either sealing the hardware memory, which in some cases requires the help of the hardware vendor, or communication with external trusted parties to ensure the one-time property (distributed agreement or monotonic counters) [1, 2, 4, 18]. Beyond the one-time property, in order to realize the functional encryption functionality of 1FE, the TEE would need to contain a secret key for a standard asymmetric encryption scheme, and an adversary should not be able to learn this key, nor any information being processed (in the clear) by the TEE. Unfortunately, TEEs are very vulnerable to physical attacks [5, 16], and in our use case the adversary is likely to have prolonged physical access to the TEE.

Strawman construction #2: trust and FE Moving away from trusted hardware, one could try leveraging a standard functional encryption scheme, and rely on an honest-but-curious (stateful) external party to enforce the one-time property. Here, this party would be solely responsible for enforcing the one-time property of the functionality, with FE guaranteeing confidentiality. However, in this approach the untrusted party learns the output of the assessment, which is not acceptable in general. Further, cryptographic FE constructions only support linear and quadratic functions, which limits the set of assessment functions we can support.

Construction blueprint In the following sections, we overcome these challenges by relying on an honest-but-curious party $\mathcal{H}$ who

does not collude with the distribution station. We reduce the task of realizing 1FE to a secure multi-party computation protocol between the distribution station and the untrusted helper, with inputs provided by recipients. We leverage $\mathcal{H}$ to enforce the one-time property. Further, we involve $\mathcal{H}$ as a helper in a secure multi-party computation protocol, which allows us to achieve security even against a fully malicious distribution station, as $\mathcal{H}$ is honest-but-curious. $\mathcal{H}$ does not need to be able to communicate with the recipients, but only (occasionally) with the distribution station in order to help with the evaluation of the assessment function. While distributed trust is hard to achieve in general, in our setting, $\mathcal{H}$ can be instantiated either by the ICRC headquarters (which is independent of the distribution teams and located in Geneva), or by another humanitarian organization. Either of these entities are assumed to be are able to communicate with the distribution station and to be honest-but-curious, and not colluding with a malicious distribution station. For example, headquarters of the ICRC or another organization are physically not located on the distribution territory, and thus not susceptible to armed take-overs.

Using this blueprint, we provide two concrete constructions for 1FE in our setting (from two-party secure multiparty computation in §5.1, and from threshold homomorphic encryption in §5.2), with different computation-communication trade-offs. Both constructions make black-box use of the underlying cryptographic primitives. We show that our protocols securely realize the 1FE ideal functionality (targeting simulation security as in §4.1). Our proofs proceed through a series of hybrids, starting at the ideal functionality and ending in the real-world protocol, and showing that each hybrid is indistinguishable from the previous one.

Protocol $\Pi^{\mathsf{2PC}}_{\mathsf{1FE}}$

$\Pi^{\mathsf{2PC}}_{\mathsf{1FE.Setup}}(1^\lambda)$

1. $\mathcal{R}$ samples $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, sends sk to $\mathcal{H}$.
2. $\mathcal{R}$ outputs $(\mathsf{pk}_{\mathsf{1FE}}, \mathsf{msk}_{\mathsf{1FE}}) = (\mathsf{pk}, \mathsf{sk})$.

$\Pi^{\mathsf{2PC}}_{\mathsf{1FE.KeyGen}}(\mathsf{msk}_{\mathsf{1FE}} = \mathsf{sk}, f, \varphi)$ for caller $\mathcal{H}$

1. $\mathcal{H}$ initializes an empty set S and outputs $\mathsf{sk}_{\mathsf{1FE}} = \bot$.

$\Pi^{\mathsf{2PC}}_{\mathsf{1FE.Enc}}(\mathsf{pk}, \mathsf{pt}, \mathsf{id})$ for caller $\mathcal{P}$

1. $\mathcal{P}$ secret-shares pt into $([\mathsf{pt}]_0, [\mathsf{pt}]_1)$.
2. $\mathcal{P}$: $\mathsf{ct} \coloneqq \mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{id} \parallel [\mathsf{info}_i]_0)$
3. $\mathcal{P}$ outputs $(\mathsf{ct}, [\mathsf{pt}]_1)$

$\Pi^{\mathsf{2PC}}_{\mathsf{1FE.Eval}}(\mathsf{sk}_{\mathsf{1FE}} = \bot, f, \varphi, \mathsf{id}, (\mathsf{ct}_i)_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]})$ for caller $\mathcal{D}$

1. $\mathcal{D}$ parses each ct_i as $(\mathsf{ct}_i^{\mathsf{PKE}}, \mathsf{sh}_{1,i})$, and sends $(\mathsf{ct}_i^{\mathsf{PKE}})_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]}$ to $\mathcal{H}$.
2. $\mathcal{H}$: If $\varphi(\mathsf{ct}_1, \ldots, \mathsf{pt}_{m+n}) = 0$, abort.
3. $\mathcal{H}$: For $i \in [m]$: $\mathsf{id}_i \parallel \mathsf{sh}_{0,i} \coloneqq \mathsf{PKE.Dec}(\mathsf{sk}; \mathsf{ct}_i)$
4. $\mathcal{H}$: If id_i are not all the same id, abort.
5. $\mathcal{H}$: If $\mathsf{id} \in S$, abort; otherwise, add id to S.
6. $\mathcal{H}$ and $\mathcal{D}$ call a 2PC protocol for f with public values $(\mathsf{pt}_{m+j})_{j\in[n]}$, with respective input shares $(\mathsf{sh}_{0,i})_{i\in[m]}$ and $(\mathsf{sh}_{1,i})$, and such that only $\mathcal{D}$ recovers the output.

Figure 5: 1FE protocol from 2PC.

5.1 1FE from 2PC

We show the protocols realizing the 1FE functionality in Figure 5. During the setup and key generation phases, the helper $\mathcal{H}$ generates a keypair for a public-key encryption scheme, and initializes an empty set which will be used to track which ids have been called. Encrypting of a plaintext value is done by splitting the value into two shares, and encrypting the shares for one party under the helper's public key. Finally, $\mathcal{D}$ holding 1FE-encrypted inputs can send the encrypted shares to $\mathcal{H}$, who will then decrypt them and run a 2PC protocol with $\mathcal{D}$ to evaluate the function on the inputs.

Lemma 5.1. *$\Pi^{\mathsf{2PC}}_{\mathsf{1FE}}$ securely implements $\mathcal{F}_{\mathsf{1FE}}$ against an honest-but-curious distribution station.*

Proof sketch (full proof in §C.4). The security of the protocols follows from the IND-CPA security of the PKE scheme and the honest-but-curious security of the 2PC protocol. □

For a malicious distribution station, using a maliciously-secure 2PC protocol is not sufficient to guarantee the security of the protocol. First, because it is unclear how the helper alone can verify that a client's authenticated share was used by the distribution station without additional interaction with said recipient. Second, because it is unclear how such a protocol can support *predicates* on both the helper's and the distribution stations' shares or garbled inputs without leaking information about the function being evaluated.

5.2 1FE from threshold HE

Our setting naturally includes multiple parties (the recipients) that can be trusted to some extent (as their privacy relies on their behaving correctly). In addition, these parties are also inherently stateful, and the aid distribution mechanism provides a natural continually incrementing counter in the form of the period number. We leverage these insights to devise a protocol $\Pi^{\mathsf{FHE}}_{\mathsf{1FE}}$ (Fig. 6) that securely realizes the functionality $\mathcal{F}_{\mathsf{1FE}}$ using distributed trust.

In the setup phase, the dealer generates a keypair for a HE scheme, and distributes shares of the secret key to the decrypting parties (i.e., the recipients). The dealer also generates an evaluation key for the assessment function f and sends it to the distribution station. During distribution, each recipient encrypts its input under the public key and sends the ciphertext to the distribution station. The distribution station relays these ciphertexts to the helper $\mathcal{H}$ (potentially along with a signature from the auditor), which evaluates the function on the ciphertexts and the plaintext inputs, and sends the resulting ciphertext to the distribution station. In the next distribution period, a threshold of recipients partially decrypts, until the distribution station recovers the output. These recipients must not necessarily be the same as the one who provided inputs in the previous period, as long as the quorum threshold is met. We now discuss three concrete threshold HE schemes:

Linear functions A (t, N)-variant of the ElGamal scheme allows for the evaluation of linear functions, and partial decryption can be done by each recipient independently.

Protocol $\Pi^{\mathsf{FHE}}_{\mathsf{1FE}}$

$\Pi^{\mathsf{FHE}}_{\mathsf{1FE.Setup}}(1^\lambda)$

1. $\mathcal{R}$ samples $(\mathsf{sk}_{\mathsf{PKE}}, \mathsf{pk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$.
2. $\mathcal{R}$ sends $\mathsf{sk}_{\mathsf{PKE}}$ to $\mathcal{H}$.
3. $\mathcal{R}$ samples $(\mathsf{sk}_{\mathsf{HE}}, \mathsf{pk}_{\mathsf{HE}}) \leftarrow \mathsf{HE.KeyGen}(1^\lambda)$, secret-shares $([\mathsf{sk}_{\mathsf{HE}}]_i)_{i\in[\mathsf{N}]}$ and sends $[\mathsf{sk}_{\mathsf{HE}}]_i$ to $\mathcal{P}_i$.
4. $\mathcal{R}$ outputs $\mathsf{pk}_{\mathsf{1FE}} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{pk}_{\mathsf{HE}})$, $\mathsf{msk}_{\mathsf{1FE}} = (\mathsf{sk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{HE}})$.

$\Pi^{\mathsf{FHE}}_{\mathsf{1FE.KeyGen}}(\mathsf{msk}_{\mathsf{1FE}} = (\mathsf{sk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{HE}}), f, \varphi)$

1. $\mathcal{R}$ sends $\mathsf{ek} \leftarrow \mathsf{HE.KeyGen}(\mathsf{sk}, f)$ to $\mathcal{H}$.
2. $\mathcal{H}$ initializes an empty set S.
3. $\mathcal{R}$ outputs $\mathsf{sk}_{\mathsf{1FE}} = \perp$.

$\Pi^{\mathsf{FHE}}_{\mathsf{1FE.Enc}}(\mathsf{pk}_{\mathsf{1FE}} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{pk}_{\mathsf{HE}}), \mathsf{pt})$

1. $\mathcal{P}$: $\mathsf{ct}^{\mathsf{HE}} \leftarrow \mathsf{HE.Enc}(\mathsf{pk}; \mathsf{pt})$.
2. $\mathcal{P}$: $\mathsf{ct}^{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{id} \parallel \mathsf{ct}^{\mathsf{HE}})$
3. $\mathcal{P}$ outputs $\mathsf{ct}^{\mathsf{PKE}}$.

$\Pi^{\mathsf{2PC}}_{\mathsf{1FE.Eval}}(f, \varphi, \mathsf{id}, (\mathsf{ct}_i)_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]})$

1. $\mathcal{D}$ sends $(\mathsf{ct}_i)_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]}$ to $\mathcal{H}$.
2. $\mathcal{H}$ aborts if $\varphi(\mathsf{ct}_1, \ldots, \mathsf{ct}_m, \mathsf{pt}_{m+1}, \ldots, \mathsf{pt}_{m+n}) = 0$.
3. $\mathcal{H}$: For $i \in [m]$: $\mathsf{id}_i \parallel \mathsf{ct}_i^{\mathsf{HE}} \coloneqq \mathsf{PKE.Dec}(\mathsf{sk}; \mathsf{ct}_i)$
4. $\mathcal{H}$: If id_i are not all the same id, abort.
5. $\mathcal{H}$: If $\mathsf{id} \in S$, abort; otherwise, add id to S.
6. $\mathcal{H}$: $\mathsf{ct}^{\mathsf{HE}} \coloneqq \mathsf{HE.Eval}(\mathsf{ek}, f, (\mathsf{ct}_i^{\mathsf{HE}})_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]})$ and sends $\mathsf{SIG.Sign}(\mathsf{sk}_{\mathsf{SIG}\mathcal{H}}; \mathsf{ct}^{\mathsf{HE}})$ to $\mathcal{D}$.
7. $\mathcal{D}$ runs a partial decryption protocol with a threshold of t decrypting parties $\mathcal{P}_i$, and receives partial decryptions $\{\mathsf{pdec}_i\}_{i\in[t]}$.
8. $\mathcal{D}$ outputs $\mathsf{HE.PartDecFin}(\mathsf{ct}, \{\mathsf{pdec}_i\}_{i\in[t]})$.

Figure 6: 1FE protocol from threshold HE.

Degree-1 functions A (t, N)-variant of the Freeman10 encryption scheme [12, Chapter 4.6],[11] (a prime-order version of the BGN scheme) allows for the evaluation of degree-2 polynomials. While partial decryption can be done by each recipient independently, decryption requires a round of interaction between the threshold parties. We translate this requirement into two successive (non-overlapping) threshold of parties, as we have no guarantee that any given recipient will be present for two consecutive periods.

General functions Threshold Fully Homomorphic Encryption (thFHE) allows evaluating arbitrary functions. However, thFHE has shown tricky to instantiate in practice without strong limitations on the partial decryption process [3, 6, 19]. [21] only requires the threshold of parties to know the identifiers of the other parties in the threshold group. In our setting, this could be instantiated by having a *pre-distribution* phase which only serves to collect the party identifiers of parties participating in the threshold decryption; for example, a person going through the queue of recipients to collect their party identifiers. As soon as a threshold of them is reached, and before the first surveyed party has claimed aid and left the queue, the identifiers can be communicated to the recipients by the distribution station in order to perform the threshold decryption.

Lemma 5.2. *$\Pi^{\mathsf{FHE}}_{\mathsf{1FE}}$ securely implements $\mathcal{F}_{\mathsf{1FE}}$ in the presence of an actively malicious distribution station.*

Proof sketch (full proof in §C.5). Security follows from the IND-CCA2-security of PKE, the IND-CPA-security of HE, the security of HE.PartDecFin, and the SUF-CMA-security of SIG. □

6 Evaluation for relevant humanitarian tasks

We implement, benchmark, and evaluate two representative assessment functions under deployment parameters provided by the ICRC. Table 1 shows the communication costs of our full protocol instantiated with $\Pi^{\mathsf{2PC}}_{\mathsf{1FE}}$ and $\Pi^{\mathsf{FHE}}_{\mathsf{1FE}}$, for simple and full assessment functions. We then discuss how our assessment protocols can be integrated with the privacy-preserving aid distribution protocol of Wang et al. [25] without introducing any changes on this protocol.

6.1 Representative assessment functions

For our evaluation, we choose two functions representative of real-world use cases discussed in §3: f_1 improves operations effectiveness and accountability; f_2 supports duty of care beyond aid distribution.

f_1: Count of successful aid claims by members of a minority. This simple assessment function counts the number of recipients with a given attribute that successfully claimed aid in a given period. These types of assessments are needed for disaggregated donor reports, (e.g., the proportion of all recipients which are part of a sexual and gender minority group that successfully claimed aid). In order to adhere to the Graceful degradation requirement, we add a threshold predicate such that the function only outputs a count if the number of total recipients which successfully claimed aid is above a certain threshold. This threshold should be sufficiently large to ensure that individual aid recipients cannot be re-identified. To implement this conditional thresholded sum, we set the recipient information info_i to be an indicator bit and consider

$$f_1(\mathsf{info}_1, \ldots, \mathsf{info}_k) \coloneqq \textstyle\sum_{i\in[k]} \mathsf{info}_i \text{ if } k \geq \ell \text{ else } -1$$

for a threshold ℓ. The symbol for a dummy count is $\perp_{f_1} \coloneqq 0$.

f_2: Reveal information to enable duty of care. This function reveals information about recipients with a relevant attribute that have not claimed aid during a large period of time, e.g., reveal the address of all recipients which have not claimed aid in the least 5 periods and whose `is_senior` attribute is set. The ICRC uses this information to perform welfare checks. We model this query as a private-attribute conditional disclosure function, with the additional requirement that a recipient's information is only disclosed if their private information additionally satisfies some predicate (`is_senior` = 1 in the example above). A recipient's information info_i contains its contact information i_i (encoded as a vector $\mathbb{Z}_q^n$) and an indicator bit b_i. The assessment function is

$$f_2((\mathsf{i}_i, \mathsf{b}_i)_{i\in[k]}, d_{1,1}, \ldots, d_{k,\mathsf{p}}) \coloneqq \left(\mathsf{i}_i \cdot \mathsf{b}_i \cdot \textstyle\prod_{j=\mathsf{p}-4}^{\mathsf{p}} (1 - d_{i,j})\right)_{i\in[\mathsf{N}]},$$

which is a quadratic *full assessment* function (the distribution outcome $d_{i,j}$ is a constant). The symbol $\perp_{f_2}$ for dummy contact information is the zero vector.

		HbC-2PC	HbC-thHE	Mal-thHE
Simple	$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{P}_i$	$2\lvert\mathsf{info}_i\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert$	$2\cdot\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\mathsf{pdec}\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert$	$2\cdot\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\mathsf{pdec}\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert$
	$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{H}$	$\mathsf{M}(\lvert\mathsf{info}_i\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert) + \lvert\mathsf{out}\rvert + \lvert f\rvert$	$\mathsf{M}\cdot\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\mathsf{out}\rvert$	$(\mathsf{M}+1)\cdot\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\sigma\rvert$
	$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{A}$	–	–	$\mathsf{M}\cdot\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\sigma\rvert$
Full	$\mathcal{R}\leftarrow\!\!\not\!\!\rightarrow\mathcal{P}_i$	$\mathsf{P}\cdot\mathsf{ent}_i\cdot(2\lvert\mathsf{info}_i\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert)$	$\mathsf{P}\cdot\mathsf{ent}_i\cdot(2\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert)$	$\mathsf{P}\cdot\mathsf{ent}_i\cdot(2\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert)$
	$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{P}_i$	–	$\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\mathsf{pdec}\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert$	$\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\mathsf{pdec}\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert + \lvert\sigma\rvert$
	$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{H}$	$\mathsf{M}(1 + \lvert\mathsf{info}_i\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert) + \lvert\mathsf{out}\rvert + \lvert f\rvert$	$\mathsf{M}\cdot(1 + \lvert\mathsf{ct}_{\mathsf{HE}}\rvert) + \lvert\mathsf{out}\rvert$	$(\mathsf{M} + \lvert\mathsf{out}\rvert)\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + 2\lvert\sigma\rvert$
	$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{A}$	$\mathsf{C}\lambda + \mathsf{N}(2\lvert\mathsf{info}_i\rvert + \lvert\mathsf{ct}_{\mathsf{PKE}}\rvert)$	$\mathsf{C}\lambda + \mathsf{N}\lvert\mathsf{ct}_{\mathsf{HE}}\rvert$	$\mathsf{C}\lambda + \mathsf{N}\lvert\mathsf{ct}_{\mathsf{HE}}\rvert + \lvert\sigma\rvert$

Table 1: Communication costs of our protocols for simple / full assessments. $\lvert\mathsf{ct}_{\mathsf{PKE}}\rvert$, $\lvert\mathsf{ct}_{\mathsf{HE}}\rvert$, $\lvert\mathsf{pdec}\rvert$, $\lvert\sigma\rvert$, and $\lvert\mathsf{out}\rvert$ are the sizes of a PKE ciphertext, thHE ciphertext, partial HE decryption, signature, and assessment output, respectively. N and M are the total number of registered recipients/recipients that showed up to the aid distribution in a given period, respectively. $\mathsf{C} \le \mathsf{M}\cdot\max_i \mathsf{ent}_i$ is the total amount of successfully claimed entitlement in a given period. Costs in the distribution phase are highlighted.

			[25]	+ { HbC-2PC,	HbC-thHE,	Mal-thHE }
f_1	Comm.	$\mathcal{R}\leftarrow\!\!\not\!\!\rightarrow\mathcal{P}_i$	96 B	-	-	-
		$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{P}_i$	≈50 kB	96 B	353 B	353 B
		$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{H}$	-	1 MB	1 MB	1 MB
		$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{A}$	700 kB	-	-	1 MB
	Comp.	$\mathcal{P}_i$ 🖴/📱	≈0.8 s	1 ms	32 ms	128 ms
		$\mathcal{D}$ 💻	≈0.07 s	0.005 s	0.45 s	0.52 s
		$\mathcal{H}$ 💻	-	0.005 s	5 s	29 s
		$\mathcal{A}$ 💻	?	-	-	32 s
f_2	Comm.	$\mathcal{R}\leftarrow\!\!\not\!\!\rightarrow\mathcal{P}_i$	96 B	4.8 kB	16 kB	16 kB
		$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{P}_i$	≈50 kB	-	224 B	288 B
		$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{H}$	-	(270) 4.4 MB	1 MB	1 MB
		$\mathcal{D}\leftarrow\!\!\not\!\!\rightarrow\mathcal{A}$	700 kB	1.0 MB	1.35 MB	1.35 MB
	Comp.	$\mathcal{P}_i$ 🖴/📱	≈0.8 s	-	12 ms	24 ms
		$\mathcal{D}$ 💻	≈0.07 s	(1.22) 0.09 s	6 s	25 s
		$\mathcal{H}$ 💻	-	(1.22) 0.09 s	15 s	50 s
		$\mathcal{A}$ 💻	?	23 s	26 s	33 s

Table 2: Communication and computation costs for the assessment functions f_1 and f_2. Preprocessing offline costs are shown in (Gray). Icons indicate the communication/computation medium. Costs in the distribution phase are highlighted. The leftmost column in gray shows the costs of the protocols of [25] (distribution only, no assessment capabilities) for reference. See §A for details.

6.2 Evaluation

We implement and benchmark all six protocols in this paper (simple and full assessments, 1FE based on 2PC and threshold HE, semi-honest and actively malicious adversary)in Rust. We use hybrid encryption with an ECDH-KEM and an AES-based DEM for PKE. For the 2PC protocol, we use the MP-SPDZ [14] implementation of the `Semi` variant of BMR garbled circuits [15]. We implement threshold variants of the ElGamal and BGN/Freeman10 HE schemes.

Informed by coarse statistics of previous distribution deployments by the ICRC, we select the following parameters which represent a mid-large distribution operation: $\mathsf{N} = 10\,000$ recipients over $\mathsf{P} = 10$ periods, with entitlements $\mathsf{ent}_i \in \{1,\dots,5\}$ for a total of $\mathsf{E} = 20\,000$ aid units. In a given period, $\mathsf{M} = 9\,000$ recipients successfully claim aid (totalling $\mathsf{C} = 18\,000$ units of aid). We set a threshold of $t = \frac{\mathsf{N}}{5} = 2\,000$ recipients corrupted by the adversary, and achieve $\lambda = 128$ bit of computational security and 40 bits of statistical security for the 2PC-based protocols.

We consider a setup simulating a typical deployment setting for the ICRC: we use a mid-range laptop (2018 Thinkpad T480 with an 8-core Intel i5 CPU @ 1.6 GHz and 16 GB of RAM) for the registration and distribution stations and the auditor. We use a low-end smartphone (2016 Doogee X5 with a 4-core ARM Cortex-A7 CPU @ 1.3 GHz and 1 GB of RAM) for recipients. As we use a single core of each device for recipient benchmarks, these devices offer comparable computational and connection capabilities to devices used by recipients in real-world deployments. We do not benchmark our protocols on smartcards, as this implementation poses a big engineering challenge and is out of the scope of this work. Nevertheless, we stress that our protocols are as lightweight as possible, and the complexity of the recipient-side computation is comparable to that of private aid distribution schemes without assessments [25] that have been implemented on smartcards (e.g., at most one ECDH and BGN encryption for our protocols, versus an ECDSA signature and an ElGamal commitment in [25]). Further, note that the threat model and assumptions on restricted network connectivity are the same regardless of the computation medium available to recipients.

We show communication and computation costs of our protocols in Table 2, together with the additional costs of running the basic distribution protocols in [25] (first column). These results demonstrate that our protocols are practical for real-world deployments (more information in §A). Note that the critical efficiency metrics (namely, the communication between recipients and stations during distribution, and the computation time for recipients) do not depend directly on the number of recipients N or periods P. The non-critical efficiency metrics (communication between distribution station and helper, computation time at distribution station and helper) depend linearly on the number of recipients and periods. Our protocols can thus scale to even to much larger deployments than the typical deployment setup given to us by the ICRC.

Communication during distribution The communication between the recipients $\mathcal{P}_i$ and the registration or distribution $\mathcal{D}$ station (highlighted in Tables 1 and 2) needs to be small to satisfy the

Efficient registration and distribution requirement. Communication is one-way (from recipients to the station), in one round. The recipients only send $\tilde{O}(|\mathsf{info}_i|)$ bits, so the communication cost is at most a few hundred bytes. In any common channel that could implement the link between $\mathcal{P}_i$ and $\mathcal{D}$ (200 kbit s^{-1} for NFC, 2 Mbit s^{-1} for Bluetooth, or 400 Mbit s^{-1} for USB), transmitting this information takes dozens of milliseconds at most. The data is also small enough to be encoded in a QR code that recipients could show to a scanner.

Communication outside of distribution For the assessment phase for f_1, $\mathcal{D}$ needs to communicate around a megabyte of data to an external party. For f_2, the HbC-2PC solution incurs an additional cost of several hundreds of megabytes in an offline phase. This offline phase can be pre-computed by the $\mathcal{D}$ before the start off the field, in parallel to the distributions. For the thHE-based protocols, $\mathcal{D}$ needs to send only around a megabyte, at the expense of additional (but still tractable) computation. The communication between $\mathcal{D}$ and external parties happens either over a GSM cell connection (10 kbit s^{-1}), a satellite connection (400 kbit s^{-1}), or a 3G connection (2 Mbit s^{-1}). In the worst case, it requires few minutes to evaluate the outcome of an assessment. We stress that this operation does not need to be done real time, but rather in a separate phase that can be done at any time after the distribution phase. We describe how the distribution station's communication can be streamed to accommodate low-bandwidth links in §B.7.

Computation The recipients can generate their 1FE ciphertexts in less than a second of computation, which fulfills the Efficient registration and distribution requirement.

Energy consumption Active Bluetooth communication requires between 0.1 W (Bluetooth Low Energy) and 1 W (classic Bluetooth). A phone CPU has a typical power consumption of 0.1 W to 0.5 W [26]. A very conservative estimate of the added power consumption of our protocols on the recipient's phone is thus 1.5 W s, which corresponds to roughly 0.0004% of a full 10 W h phone battery charge (including computation and sending packets at distribution time).

Simple/full assessments For simple assessments, the registration does not send any additional data to the recipients. Full assessments require the registration station to send $\tilde{O}(\mathsf{P} \cdot \mathsf{ent}_i)$ bits to the recipients during registration. For both simple and full assessments, recipients send a constant number of bits to the distribution station when claiming their aid. For full assessments, the distribution station needs to additionally send $\mathsf{E} \cdot \lambda$ bits to the auditor, where E is the total number of distributed aid units.

The cost of malicious security Malicious security requires communicating at most dozens of kilobytes of random tags between recipients and the registration station (less than a second when using an NFC link). The recipients themselves do not need to store these tags in full. They can instead store a short seed and use a pseudo-random function to generate the tags when needed.

6.3 Integration with aid distribution systems

System integration Our protocols can easily run in parallel to existing privacy-preserving aid distribution systems without requiring changes. For example, to integrate with Wang et al.'s system [25], our assessment registration protocol can be run before or after the registration phase of [25]. In the distribution phase, the assessment data collection runs after Wang et al.'s distribution protocol. The assessment evaluation protocol can be run after each distribution period, or once at the end of the program.

Security properties Our protocols do not break the security properties of aid distribution schemes they can be integrated with [25]. First, we do not modify the core distribution and audit protocol, and thus the overall system inherits the security properties in [25]. Second, for registration and distribution, our protocols do not introduce any leakage beyond the leakage inherent to the assessment functionality. This, combined with the use of independent keying material in the registration, distribution, and assessment protocols, ensures that the overall protocol achieves *registration privacy* and *distribution privacy*. The output of simple assessments does not introduce any leakage with respect to the information revealed by the aid distribution protocols. When computing full assessments, we leak the distribution traces (i.e., pseudonymized traces of whether a recipient successfully claimed aid, over multiple periods). We elaborate on this aspect in §4.2.2, and stress that this is a fundamental, necessary leakage if full assessments are desired. To account for this leakage, the ICRC must conduct an analysis of the harms that the disclosure of this information could cause before using such assessments. Finally, even after integration, assessments can be shut down (Threat model agility) without stopping aid distribution; thanks to Graceful degradation, recipients only incur bounded harm for an adversary which is stronger than initially assumed.

7 Final remarks

Informed by a collaboration with the ICRC, this work extends prior solutions for privacy-preserving aid distribution by offering a mechanism to collect information critical to the operation of humanitarian aid organizations. Our solution minimizes additional leakage compared to the privacy guarantees of previous works, and can operate in volatile threat environments. Yet, we would like to caution about the dangers of integrating and using similar reporting mechanisms. Even if done in a privacy-preserving way, excessive collection and disclosure of metrics may harm the individuals and communities that the privacy-preserving design aimed to protect in the first place. In this work, we have carefully considered what metrics to compute and their associated risks, and any deployment must do the same to avoid unintended harms.

Although this work targets humanitarian aid distribution, the problem of gathering operational information within a privacy-preserving application is of broader relevance. Public institutions and non-profits are increasingly interested in deploying privacy-preserving applications [7, 20, 23, 24], for which they may wish to report metrics on the efficacy and impact of their work. Proactively incorporating mechanisms for reporting aggregate statistics may help support these organizations when deploying privacy-preserving designs, and assuage common concerns that such institutions face. The one-time functional encryption primitive we introduce might be of interest in these scenarios.

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Maurice Bailleu, Jörg Thalheim, Pramod Bhatotia, Christof Fetzer, Michio Honda, and Kapil Vaswani. SPEICHER: Securing LSM-based Key-Value stores using shielded execution. In *17th USENIX Conference on File and Storage Technologies (FAST 19)*, pages 173–190, Boston, MA, February 2019. USENIX Association. URL: https://www.usenix.org/conference/fast19/presentation/bailleu.

[2] Pramod Bhatotia, Markulf Kohlweiss, Lorenzo Martinico, and Yiannis Tselekounis. Steel: Composable hardware-based stateful and randomised functional encryption. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 709–736, Virtual Event, May 10–13, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-75248-4_25.

[3] Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 565–596, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-96884-1_19.

[4] Marcus Brandenburger, Christian Cachin, Matthias Lorenz, and Rüdiger Kapitza. Rollback and forking detection for trusted execution environments using lightweight collective memory. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 157–168, 2017. doi:10.1109/DSN.2017.45.

[5] David Cerdeira, Nuno Santos, Pedro Fonseca, and Sandro Pinto. SoK: Understanding the prevailing security vulnerabilities in TrustZone-assisted TEE systems. In *2020 IEEE Symposium on Security and Privacy*, pages 1416–1432, San Francisco, CA, USA, May 18–21, 2020. IEEE Computer Society Press. doi:10.1109/SP40000.2020.00061.

[6] Siddhartha Chowdhury, Sayani Sinha, Animesh Singh, Shubham Mishra, Chandan Chaudhary, Sikhar Patranabis, Pratyay Mukherjee, Ayantika Chatterjee, and Debdeep Mukhopadhyay. Efficient threshold FHE with application to real-time systems. *IACR Cryptol. ePrint Arch.*, page 1625, 2022. URL: https://eprint.iacr.org/2022/1625.

[7] David Archer, Amy O'Hara, Rawane Issa, and Stephanie Straus. Sharing Sensitive Department of Education Data Across Organizational Boundaries Using Secure Multiparty Computation. Technical report, Georgetown University Massive Data Institute, May 2021.

[8] Kasra Edalatnejad, Wouter Lueks, Justinas Sukaitis, Vincent Graf Narbel, Massimo Marelli, and Carmela Troncoso. Janus: Safe biometric deduplication for humanitarian aid distribution. In *2024 IEEE Symposium on Security and Privacy*, pages 655–672, San Francisco, CA, USA, May 19–23, 2024. IEEE Computer Society Press. doi:10.1109/SP54263.2024.00116.

[9] United Nations High Commissioner for Refugees (UN HCR). Operational guidance on accountability to affected people. https://www.unhcr.org/sites/default/files/2022-12/UNHCR-AAP_Operational_Guidance.pdf, 2020. Accessed: 2024-11-18.

[10] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 39–56, Santa Barbara, CA, USA, August 17–21, 2008. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-85174-5_3.

[11] Chloé Hébant, Duong Hieu Phan, and David Pointcheval. Decentralized evaluation of quadratic polynomials on encrypted data. In Zhiqiang Lin, Charalampos Papamanthou, and Michalis Polychronakis, editors, *ISC 2019*, volume 11723 of *LNCS*, pages 87–106, New York City, NY, USA, September 16–18, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-30215-3_5.

[12] Kevin John Henry. The theory and applications of homomorphic cryptography. Master's thesis, University of Waterloo, 2008.

[13] Anja Kaspersen and Charlotte Lindsey-Curtet. The digital transformation of the humanitarian sector. https://blogs.icrc.org/law-and-policy/2016/12/05/digital-transformation-humanitarian-sector/, 2016. Accessed: May 10, 2026.

[14] Marcel Keller. MP-SPDZ: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020. doi:10.1145/3372297.3417872.

[15] Marcel Keller and Avishay Yanai. Efficient maliciously secure multiparty computation for RAM. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 91–124, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-78372-7_4.

[16] Mengyuan Li, Yuheng Yang, Guoxing Chen, Mengjia Yan, and Yinqian Zhang. SoK: Understanding design choices and pitfalls of trusted execution environments. In Jianying Zhou, Tony Q. S. Quek, Debin Gao, and Alvaro A. Cárdenas, editors, *ASIACCS 24*, Singapore, July 1–5, 2024. ACM Press. doi:10.1145/3634737.3644993.

[17] Eva Luvison, Sylvain Chatel, Justinas Sukaitis, Vincent Graf Narbel, Carmela Troncoso, and Wouter Lueks. A low-cost privacy-preserving digital wallet for humanitarian aid distribution. In *IEEE Security and Privacy*, 2025. URL: https://arxiv.org/abs/2410.15942, arXiv:2410.15942.

[18] Sinisa Matetic, Mansoor Ahmed, Kari Kostiainen, Aritra Dhar, David Sommer, Arthur Gervais, Ari Juels, and Srdjan Capkun. ROTE: Rollback protection for trusted execution. In Engin Kirda and Thomas Ristenpart, editors, *USENIX Security 2017*, pages 1289–1306, Vancouver, BC, Canada, August 16–18, 2017. USENIX Association.

[19] Christian Mouchet, Elliott Bertrand, and Jean-Pierre Hubaux. An efficient threshold access-structure for rlwe-based multiparty homomorphic encryption. *J. Cryptol.*, 36(2):10, 2023. URL: https://doi.org/10.1007/s00145-023-09452-8, doi:10.1007/S00145-023-09452-8.

[20] National Science and Technology Council. National Strategy to Advance Privacy-Preserving Data Sharing and Analytics. Technical report, Executive Office of the President of the United States, March 2023. URL: https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Strategy-to-Advance-Privacy-Preserving-Data-Sharing-and-Analytics.pdf.

[21] Truong Son Nguyen, Tancrède Lepoint, and Ni Trieu. Mario: Multi-round multiple-aggregator secure aggregation with robustness against malicious actors. *IACR Cryptol. ePrint Arch.*, page 1428, 2024. URL: https://eprint.iacr.org/2024/1428.

[22] Daniele Perito and Gene Tsudik. Secure code update for embedded devices via proofs of secure erasure. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *ESORICS 2010*, volume 6345 of *LNCS*, pages 643–662, Athens, Greece, September 20–22, 2010. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-15497-3_39.

[23] Anjana Rajan, Lucy Qin, David W. Archer, Dan Boneh, Tancrede Lepoint, and Mayank Varia. Callisto: A Cryptographic Approach to Detecting Serial Perpetrators of Sexual Misconduct. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–4, Menlo Park and San Jose CA USA, June 2018. ACM. URL: https://dl.acm.org/doi/10.1145/3209811.3212699, doi:10.1145/3209811.3212699.

[24] United Nations Committee of Experts on Big Data and Data Science for Official Statistics. The United Nations Guide on Privacy-Enhancing Technologies for Official Statistics. Technical report, United Nations, 2023. URL: https://unstats.un.org/bigdata/task-teams/privacy/guide/.

[25] Boya Wang, Wouter Lueks, Justinas Sukaitis, Vincent Graf Narbel, and Carmela Troncoso. Not yet another digital ID: Privacy-preserving humanitarian aid distribution. In *2023 IEEE Symposium on Security and Privacy*, pages 645–663, San Francisco, CA, USA, May 21–25, 2023. IEEE Computer Society Press. doi:10.1109/SP46215.2023.10179306.

[26] Ming Yan, Chien Aun Chan, André F. Gygax, Jinyao Yan, Leith Campbell, Ampalavanapillai Nirmalathas, and Christopher Leckie. Modeling the total energy consumption of mobile network services and applications. *Energies*, 12(1), 2019. URL: https://www.mdpi.com/1996-1073/12/1/184, doi:10.3390/en12010184.

[27] Lianying Zhao, Joseph I. Choi, Didem Demirag, Kevin R. B. Butler, Mohammad Mannan, Erman Ayday, and Jeremy Clark. One-time programs made practical. In Ian Goldberg and Tyler Moore, editors, *FC 2019*, volume 11598 of *LNCS*, pages 646–666, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-32101-7_37.

A Practicality of our Protocols

We give a more detailed explanation of our results in Table 2 and how they demonstrate the practicality of our protocols.

Our protocols are intended to be run *in addition* to a base protocol for privacy-preserving aid distribution, such as the one of Wang et al. [25]. The costs for Wang et al. are shown in gray in the first column of Table 2. To obtain these numbers, we take as reference the reported performance from Section 7 and Figure 3 in [25]. For the recipients, the runtime on smartcards and on phones is roughly the same (779 ms and 800 ms), and thus we report a single number on the table. The rightmost three columns show the additional costs introduced by our assessment protocols.

We note that the phones that we use to benchmark our protocols are less powerful than the phones considered in [25] (4-core ARM Cortex-A7 CPU @ 1.3 GHz and 1 GB of RAM in our case, versus 6 ARM Cortex-A53 cores @ 1.6 GHz plus 2 ARM Cortex-A73 cores @ 1.8 GHz, and 4 GB of RAM in [25]). Similarly, we use a less powerful laptop for the distribution station than the one used in [25] (Intel Core i5-7200U @ 1.6 GHz in our case, versus Intel Core i7-9700K @ 2.8 GHz in [25]).

Registration For simple assessments, our protocols do not require any additional communication or computation compared to the base protocol of [25]. For full assessments, our protocols require up to 16 kB of communication between each recipient and the registration station. Even with the worst bandwidth available to recipients (200 kbit s^{-1} for NFC on phones, or 848 kbit s^{-1} for smartcards), this corresponds to sub-second additional communication time. Furthermore, the registration phase is not time-critical, as recipients typically register well in advance of the distribution phase and not in bulk.

Distribution The base protocol of [25] requires significant communication between each recipient and the distribution station (we report the number of 50 kB for a blocklist of 512 entries [25, Section 7], but this can go up to several megabytes for larger blocklists [25, Figure 3]). In contrast, our protocols require a few hundred bytes of additional communication at most. The computation time of our protocols for recipients is less than a second in all cases, which is comparable to the 0.8 s required by the base protocol of [25]. Similarly, our protocols require the distribution station to compute for less than a second (and in the case of our HbC-2PC instantiation, less than one tenth of the computation of the base protocol of [25]). Thus, applying our assessment protocol on top of the base protocol of [25] still fulfills the Efficient registration and distribution requirement, as the computation time is still below the time of any physical interaction happening at distribution time.

Assessments Evaluation Our protocols require communication between the distribution station and the helper $\mathcal{H}$, and between the distribution station and the auditor $\mathcal{A}$ in order to evaluate the assessment function. We recall that these steps do not happen often, and do not require real time like distribution. Our protocols require a communication overhead of 1 MB to 4.4 MB between the distribution station and the helper $\mathcal{H}$, as well as 1 MB to 1.35 MB between the distribution station and the auditor $\mathcal{A}$ for our maliciously-secure variants. This is on the same order of magnitude as the communication required by the base protocol of [25] between the distribution station and the auditor (700 kB), which is not a significant overhead for a satellite communication. Even though this phase is not time-critical, or not limited in bandwidth, we describe further optimizations to reduce the communication costs in §B.7.

B Extensions

In this appendix, we present several extensions to our framework, which can be used to adapt our assessment protocols to different deployment scenarios and requirements. We note that these extensions are all modular and we take care to present implementations which are orthogonal to each other, such that any subset of extensions can be combined to suit the specific requirements of a deployment.

B.1 Assessments for external use

In some cases, the assessment outcome should not only be revealed to the distribution station, but should be shared with specific units of the ICRC or with other third parties.

B.1.1 Honest-but-curious distribution stations. In the threat model of honest-but-curious distribution stations, the distribution station can relay the assessment outcome to the relevant parties. The proof of security closely follows the proofs of Lemmas 5.1 and 5.2.

B.1.2 Malicious distribution stations. For malicious distribution stations, we require the help of the parties involved in the 1FE evaluation protocol in order to ensure that the assessment outcome is correctly relayed.

For our 1FE construction based on threshold HE (§5.2), the distribution station should provide a proof that the assessment outcome is the correct final decryption computed from partial decryptions provided by the recipients. This can be done by leveraging a public-key infrastructure (PKI) during registration, with each recipient being assigned a signing key, and the registration station knowing the corresponding verification key. We show this construction in Fig. A7.

Given a set of partial decryptions, the distribution station can then compute a zero-knowledge proof that these partial decryptions have been signed by one of the signing keys, and forward the partial decryptions along with the proof to the output party O. This party can then verify the proof and recompute the final decryption using the partial decryptions in order to obtain the assessment outcome.

Proof. We start in the real world and proceed in a series of indistinguishable hybrids.

Hybrid 1 The simulator replaces all PKE ciphertexts by random values and implements a secure channel between $\mathcal{P}$ and $\mathcal{H}$. This hybrid is indistinguishable from the real world by the IND-CCA2-security of PKE.

Hybrid 2 The simulator replaces the proof π with a simulated proof. Hybrid 2 is computationally indistinguishable from Hybrid 1 by the simulation-extractability of the proof system.

Hybrid 3 The simulator replaces all signatures with random strings, and programs HE.PartDecFin to use the output ciphertext provided by $\mathcal{H}$. Hybrid 2 is indistinguishable from Hybrid 1 by the SUF-CMA-security of SIG.

Protocol $\Pi^{\mathsf{FHE}}_{\mathsf{1FE}}$

$\Pi^{\mathsf{FHE}}_{\mathsf{1FE.Setup}}(1^\lambda)$

1. $\mathcal{R}$ samples $(\mathsf{sk}_{\mathsf{PKE}}, \mathsf{pk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$.
2. $\mathcal{R}$ sends $\mathsf{sk}_{\mathsf{PKE}}$ to $\mathcal{H}$.
3. $\mathcal{R}$ samples $(\mathsf{sk}_{\mathsf{SIG}i}, \mathsf{vk}_{\mathsf{SIG}i}) \leftarrow \mathsf{SIG.Setup}(1^\lambda)$ for each recipient $\mathcal{P}_i$ in N, and sends $\{\mathsf{vk}_{\mathsf{SIG}i}\}_{i\in[\mathsf{N}]}$ to $\mathcal{O}$.
4. $\mathcal{R}$ samples $(\mathsf{sk}_{\mathsf{HE}}, \mathsf{pk}_{\mathsf{HE}}) \leftarrow \mathsf{HE.KeyGen}(1^\lambda)$, secret-shares $([\mathsf{sk}_{\mathsf{HE}}]_i)_{i\in[\mathsf{N}]}$ and sends $[\mathsf{sk}_{\mathsf{HE}}]_i$ $\mathsf{vk}_{\mathsf{SIG}i}$ to $\mathcal{P}_i$.
5. $\mathcal{R}$ outputs $\mathsf{pk}_{\mathsf{1FE}} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{pk}_{\mathsf{HE}})$, $\mathsf{msk}_{\mathsf{1FE}} = (\mathsf{sk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{HE}})$.

$\Pi^{\mathsf{FHE}}_{\mathsf{1FE.KeyGen}}(\mathsf{msk}_{\mathsf{1FE}} = (\mathsf{sk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{HE}}), f, \varphi)$

1. $\mathcal{R}$ sends $\mathsf{ek} \leftarrow \mathsf{HE.KeyGen}(\mathsf{sk}, f)$ to $\mathcal{H}$.
2. $\mathcal{H}$ initializes an empty set S.
3. $\mathcal{R}$ outputs $\mathsf{sk}_{\mathsf{1FE}} = \perp$.

$\Pi^{\mathsf{FHE}}_{\mathsf{1FE.Enc}}(\mathsf{pk}_{\mathsf{1FE}} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{pk}_{\mathsf{HE}}), \mathsf{pt})$

1. $\mathcal{P}$: $\mathsf{ct}^{\mathsf{HE}} \leftarrow \mathsf{HE.Enc}(\mathsf{pk}; \mathsf{pt})$.
2. $\mathcal{P}$: $\mathsf{ct}^{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{id} \parallel \mathsf{ct}^{\mathsf{HE}})$
3. $\mathcal{P}$ outputs $\mathsf{ct}^{\mathsf{PKE}}$.

$\Pi^{\mathsf{2PC}}_{\mathsf{1FE.Eval}}(f, \varphi, \mathsf{id}, (\mathsf{ct}_i)_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]})$

1. $\mathcal{D}$ sends $(\mathsf{ct}_i)_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]}$ to $\mathcal{H}$.
2. $\mathcal{H}$ aborts if $\varphi(\mathsf{ct}_1, \ldots, \mathsf{ct}_m, \mathsf{pt}_{m+1}, \ldots, \mathsf{pt}_{m+n}) = 0$.
3. $\mathcal{H}$: For $i \in [m]$: $\mathsf{id}_i \parallel \mathsf{ct}_i^{\mathsf{HE}} \coloneqq \mathsf{PKE.Dec}(\mathsf{sk}; \mathsf{ct}_i)$
4. $\mathcal{H}$: If id_i are not all the same id, abort.
5. $\mathcal{H}$: If $\mathsf{id} \in S$, abort; otherwise, add id to S.
6. $\mathcal{H}$: $\mathsf{ct}^{\mathsf{HE}} \coloneqq \mathsf{HE.Eval}(\mathsf{ek}, f, (\mathsf{ct}_i^{\mathsf{HE}})_{i\in[m]}, (\mathsf{pt}_{m+j})_{j\in[n]})$ and sends $\mathsf{SIG.Sign}(\mathsf{sk}_{\mathsf{SIG}\mathcal{H}}; \mathsf{ct}^{\mathsf{HE}})$ to $\mathcal{D}$.
7. $\mathcal{D}$ runs a partial decryption protocol with a threshold of t decrypting parties $\mathcal{P}_i$, and receives partial decryptions $\{\mathsf{pdec}_i\}_{i\in[t]}$ and their signatures $\{\sigma_i\}_{i\in[t]}$.
8. $\mathcal{D}$ outputs $\mathsf{pt} \coloneqq \mathsf{HE.PartDecFin}(\mathsf{ct}, \{\mathsf{pdec}_i\}_{i\in[t]})$,ct, $\{\mathsf{pdec}_i\}_{i\in[t]}$, and a zero-knowledge proof π for the statement $\bigwedge_{i\in[t]} \bigvee_{j\in[\mathsf{N}]} \mathsf{SIG.Vfy}(\mathsf{vk}_{\mathsf{SIG}j}, \sigma_i) = 1$ with public instance $((\mathsf{pdec}_i)_{i\in[t]}, (\mathsf{vk}_{\mathsf{SIG}i})_{i\in[\mathsf{N}]})$ and witness $(\sigma_i)_{i\in[t]}$ to $\mathcal{O}$.

Figure A7: 1FE protocol from threshold HE for outputs destined for the external party $\mathcal{O}$. Differences from Fig. 6 are highlighted in blue.

Hybrid 4 The simulator replaces HE.PartDecFin with the simulator for HE.PartDecFin, which takes as input the plaintext values from the input parties . Hybrid 3 is indistinguishable from Hybrid 2 by the security of HE.PartDecFin.

Hybrid 4 is indistinguishable from $\mathcal{F}_{\mathsf{1FE}}$, since $\mathcal{H}$ is semi-honest and enforces the one-time property and predicate checks, and the output plaintexts have the same distribution in both worlds. □

B.2 Multiple assessment functions

In order to handle multiple assessment functions, only the 1FE key generation and evaluation phases must be repeated for each function; the 1FE setup and encryption phases, as well as the binding protocols for malicious stations stay unchanged. We note that for our instantiations of the 1FE evaluation phase (see §5), multiple 1FE evaluation phases can be batched in order to minimize round complexity.

B.3 Threshold functions

Leakage from the assessments (both the final result and the result of individual evaluations) can be significantly reduced by using a *threshold predicate*, i.e., by considering a predicate $\varphi_t(x_1, \ldots x_n) = 1 \Leftrightarrow n \geq t$. With this mitigation in mind, the ICRC might decide to deploy a solution that only offers optimal leakage against honest-but-curious distribution stations (but still satisfies Graceful degradation), and exactly quantify the leakage of one function evaluation in case of a mismatch between threat model and adversarial capabilities.

B.4 Differential Privacy

Differential privacy (DP) is a powerful tool to protect the privacy of individuals in a dataset while still allowing for the extraction of useful information. From a deployment perspective, our context is especially well-suited for DP, since both the set of queries (assessments) and the number of queries made on the dataset (recipient information) is bounded, which allows for a fine-grained control of the privacy-utility trade-off and a straightforward DP parameter setting. Yet, from a functionality perspective it is important to validate with the humanitarian organization that in the concrete scenario where the assessments are to be deployed, noisy values are suitable for their purposes.

DP can be added to our system in two ways: First, as a post-processing step after the function has been evaluated, which can be done in our concrete realizations (both 2PC and HE-based ones) with the help of the helper server. Second, as an input perturbation mechanism, where the recipient's input is perturbed before being encrypted. For a single function, this comes at no additional cost in our protocol, since recipients already encrypt their information separately for each period (i.e., each query), and the DP noise for each such ciphertext is independent of all others computed by the same recipient. For multiple functions however, recipients need to perturb the input independently for each function and period, which leads to an increase in the number of ciphertexts that is linear in the number of assessment functions.

B.5 Adding Kill-Switches

As conditions in a particular deployment can change suddenly, it may be the case that it is no longer safe to continue operating the aid distribution system. In some cases, the ICRC may stop the distribution process altogether. However, in order to provide an additional layer of security for the case that a distribution station is taken over and continues to operate maliciously (and collecting and evaluating assessment information), we can augment our assessment evaluation protocols with a "kill-switch" functionality. To realize this functionality, the 1FE predicate required for evaluation is additionally augmented with a second signature from a third party (e.g., the ICRC headquarters, or a trusted observer of the

geopolitical and security situation) to prevent any evaluation in case of sudden change in distribution circumstances.

B.6 Less frequent audits

Our protocols in §4 assume that the auditor carries out an audit at the end of each period, which is required to guarantee malicious security. If the audits are less frequent than once a period (or are not fully synchronized), our protocols still achieve security against *covert* distribution stations. Under this relaxed model, our protocols still fulfill the Graceful degradation property, since the one-time property of our 1FE scheme still ensures that only a single function evaluation is leaked, which limits the harm stemming from violating Assessment privacy.

B.7 Streaming communication

Sending all the data at once between the registration station, the distribution station, and the auditor might be challenging due to the Weak network connectivity. It would thus be advantageous to stream these messages, for example, relaying recipient messages to the auditor as recipients arrive at the registration station. However, naïvely streaming messages will leak some additional information to the auditor; for the example above, the auditor will be able to infer the individual entitlement of recipients (whereas it is only supposed to learn the total entitlement). It is unclear whether this leakage is acceptable in practice, but it can be mitigated by chunking and aggregating recipient's messages in fixed-size batches, thereby removing this leakage.

C Full Proofs

C.1 Proof of Lemma 4.2

Full proof of Lemma 4.2. We define a hybrid in which the simulator replaces all ciphertexts encrypted towards the auditor with a random value, and forwards the recipients' messages directly to the auditor (in essence, implementing a secure channel). The view of an adversary in the real world and in this hybrid are computationally indistinguishable by the IND-CCA2 security of PKE. Finally, the view of an adversary in this hybrid and in the ideal world are indistinguishable in the $\mathcal{F}_{\text{1FE}}$-hybrid model, thanks to the correctness of the matching procedure §4.2.2 and the assumption that the auditor is honest-but-curious. □

C.2 Proof of Lemma 4.3

Full proof of Lemma 4.3. We start in the real world and proceed in a series of indistinguishable hybrids.

Hybrid 1 The simulator replaces all ciphertexts encrypted towards the auditor with a random value, and forwards the recipients' messages (random tag and 1FE ciphertext) to the auditor. This is computationally indistinguishable from the real world by the IND-CCA2 security of PKE.

Hybrid 2 The simulator programs the $\mathcal{F}_{\text{1FE}}$ functionality to ignore the signature provided by the adversary. Instead, $\mathcal{F}_{\text{1FE}}$ rejects if the auditor outputted a $\perp$ message, and behaves as the plain $\mathcal{F}_{\text{1FE}}$ without predicates otherwise. This hybrid is computationally indistinguishable from Hybrid 1 by the SUF-CMA-security of SIG, as it is infeasible for the adversary to forge a signature for the set of inputs m_{p}.

Hybrid 3 The simulator programs the $\mathcal{F}_{\text{1FE}}$ functionality to ignore its ciphertext inputs as well, and to output f evaluated on the recipient's plaintexts. Hybrid 2 and 3 are equivalent up until the event where the distribution station $\mathcal{D}$ distributes aid to a recipient but either (i) drops that recipient's message to the 1FE functionality but forwards their message to $\mathcal{A}$, or (ii) forwards the recipient's message to the 1FE functionality but drops it for $\mathcal{A}$, or (iii) drops both recipient messages. In cases (i) and (ii), the signature predicate will not be satisfied, and the 1FE functionality will abort. In case (iii), there will be a discrepancy between the total entitlement registered with the trusted external party and the physical inventory. The distribution station will not be able to guess a valid secret tag that it can pair with an entitlement and user ciphertext to make up the discrepancy. However, the programmed $\mathcal{F}_{\text{1FE}}$ outputs $\perp$ in this case.

Finally, by construction, Hybrid 3 is indistinguishable from the ideal world in the $\mathcal{F}_{\text{1FE}}$-hybrid model. □

C.3 Proof of Lemma 4.4

Full proof of Lemma 4.4. We start in the real world and proceed in a series of indistinguishable hybrids.

Hybrid 1 The simulator replaces all ciphertexts encrypted towards the auditor with a random value, and forwards the recipients' messages (random tag and 1FE ciphertext) to the auditor. This is computationally indistinguishable from the real world by the IND-CCA2 security of PKE.

Hybrid 2 The simulator programs the $\mathcal{F}_{\text{1FE}}$ functionality to ignore the signature provided by the adversary. Instead, $\mathcal{F}_{\text{1FE}}$ only outputs the assessment output if the auditor outputted a non-$\perp$ message. This hybrid is computationally indistinguishable from Hybrid 1 by the SUF-CMA-security of SIG, as it is infeasible for the adversary to forge a signature for the set of inputs m_{p}.

Hybrid 2 is indistinguishable from the ideal world in the $\mathcal{F}_{\text{1FE}}$-hybrid model by construction. □

C.4 Proof of Lemma 5.1

Full proof of Lemma 5.1. We start in the real world and proceed in a series of indistinguishable hybrids.

Hybrid 1 The simulator replaces all PKE ciphertexts by random values and implements a secure channel between $\mathcal{P}$ and $\mathcal{H}$. This hybrid is indistinguishable from the real world by the IND-CCA2 security of PKE.

Hybrid 1 is computationally indistinguishable from $\mathcal{F}_{\text{1FE}}$: since $\mathcal{H}$ implements one-time and predicate checks and is semi-honest, any distinguisher can also be turned into an attacker against the semi-honest security of the 2PC protocol. □

C.5 Proof of Lemma 5.2

Full proof of Lemma 5.2. We start in the real world and proceed in a series of indistinguishable hybrids.

Hybrid 1 The simulator replaces all PKE ciphertexts by random values and implements a secure channel between $\mathcal{P}$ and $\mathcal{H}$. This

hybrid is indistinguishable from the real world by the IND-CCA2-security of PKE.

Hybrid 2 The simulator replaces all signatures with random strings, and programs HE.PartDecFin to use the output ciphertext provided by $\mathcal{H}$. Hybrid 2 is indistinguishable from Hybrid 1 by the SUF-CMA-security of SIG.

Hybrid 3 The simulator replaces HE.PartDecFin with the simulator for HE.PartDecFin, which takes as input the plaintext values from the input parties . Hybrid 3 is indistinguishable from Hybrid 2 by the security of HE.PartDecFin.

Hybrid 3 is indistinguishable from $\mathcal{F}_{\mathsf{1FE}}$, as $\mathcal{H}$ implements one-time and predicate checks and is semi-honest, and the output plaintexts have the same distribution in both worlds. □