Estimating Incidental Collection in Foreign Intelligence Surveillance: Large-Scale Multiparty Private Set Intersection with Union and Sum

Anunay Kulshrestha and Jonathan Mayer, *Princeton University*

https://www.usenix.org/conference/usenixsecurity22/presentation/kulshrestha

This paper is included in the Proceedings of the 31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Open access to the Proceedings of the 31st USENIX Security Symposium is sponsored by USENIX.

Estimating Incidental Collection in Foreign Intelligence Surveillance: Large-Scale Multiparty Private Set Intersection with Union and Sum

Anunay Kulshrestha
Princeton University

Jonathan Mayer
Princeton University

Abstract

Section 702 of the Foreign Intelligence Surveillance Act authorizes U.S. intelligence agencies to intercept communications content without obtaining a warrant. While Section 702 requires targeting foreigners abroad for intelligence purposes, agencies "incidentally" collect communications to or from Americans and can search that data for purposes beyond intelligence gathering. For over a decade, members of Congress and civil society organizations have called on the U.S. Intelligence Community (IC) to estimate the scale of incidental collection. Senior intelligence officials have acknowledged the value of quantitative transparency for incidental collection, but the IC has not identified a satisfactory estimation method that respects individual privacy, protects intelligence sources and methods, and imposes minimal burden on IC resources.

In this work, we propose a novel approach to estimating incidental collection using secure multiparty computation (MPC). The IC possesses records about the parties to intercepted communications, and communications services possess country-level location for users. By combining these datasets with MPC, it is possible to generate an automated aggregate estimate of incidental collection that maintains confidentiality for intercepted communications and user locations.

We formalize our proposal as a new variant of private set intersection, which we term multiparty private set intersection with union and sum (MPSIU-Sum). We then design and evaluate an efficient MPSIU-Sum protocol, based on elliptic curve cryptography and partially homomorphic encryption. Our protocol performs well at the large scale necessary for estimating incidental collection in Section 702 surveillance.

1 Introduction

When a nation conducts surveillance directed outside its own borders and at foreign intelligence targets, how often does it intercept communications involving its own people? For over a decade, that seemingly simple factual question has been a flashpoint in United States national security law.

Section 702 of the Foreign Intelligence Surveillance Act (FISA) authorizes agencies in the U.S. Intelligence Community (IC) to collect communications inside the U.S. when targeting foreigners abroad [2, 36, 62]. Section 702, unlike conventional law enforcement and FISA procedures for obtaining communications content, does not require applying to a court for a warrant demonstrating probable cause and particularity for a specific target. Instead, the IC obtains annual program approvals from the Foreign Intelligence Surveillance Court (FISC), then directs communications services in the U.S. to facilitate surveillance of foreign intelligence targets.

The structure and implementation of Section 702 have prompted significant controversy, especially over "incidental" collection of communications to and from U.S. citizens and other persons protected by constitutional privacy guarantees. The statutory framework and FISC orders permit agencies to query and use these communications for purposes beyond foreign intelligence, without obtaining a warrant as ordinarily required by the Fourth Amendment to the U.S. Constitution.

For over a decade, members of Congress (on a bipartisan basis) and civil society groups have repeatedly urged the IC to estimate the scale of incidental collection [5, 7, 8, 11, 14–18]. The IC's leadership has acknowledged the importance of an empirical estimate for public transparency [6, 9, 10, 12, 21]. Because the IC often lacks information about non-target parties to intercepted communications, however, it cannot readily compute an estimate. After years of exploring estimation methods, the IC has not identified a method that it considers adequate for respecting individual privacy, protecting intelligence sources and methods, and avoiding burdensome manual analysis. Section 2 provides further detail on Section 702 of FISA, incidental collection, and the estimation challenge.

In this work, we propose a novel path forward for estimating incidental collection using secure multiparty computation (MPC). The IC possesses records of the parties to intercepted communications, but may know little about non-target parties. Communications services possess country-level user location for business purposes, but may know little about intercepted communications. By combining these datasets with MPC, it

is possible to generate an aggregate estimate of incidental collection that maintains the secrecy of targets and intercepts, maintains the confidentiality of user locations, and involves no manual investigation of users. Section 3 formalizes the computation and privacy guarantees as a new variation of private set intersection, which we term multiparty private set intersection with union and sum (MPSIU-Sum).

We design and evaluate a novel MPSIU-Sum protocol, which is practical at the large scale necessary for estimating incidental collection. Section 4 provides preliminaries for protocol construction, including on elliptic curve cryptography and partially homomorphic encryption. Section 5 then contributes a new MPSI protocol, building on the efficient Apple PSI protocol [28], which we use as an intermediate step. Section 6 presents our MPSIU-Sum protocol. Section 7 empirically evaluates performance. Section 8 offers optimizations. Section 9 discusses extensions, including for additional malicious security and differential privacy. Section 10 synthesizes related work on private set intersection and secure sum. Finally, Section 11 concludes with directions for transitioning our proposed estimation method into practice.

2 Background and Motivation

We begin by providing background on Section 702 of FISA, which is the motivation for our work. We briefly describe the constitutional and statutory legal frameworks for U.S. law enforcement and intelligence interception of electronic communications, and we explain why Section 702 is so different from prior authorities. Next, we describe the "incidental" collection and "U.S. person query" issues that have especially prompted concern about Section 702. Finally, we discuss the challenge of estimating the scale of incidental collection, which has been an open policy problem for over a decade.

2.1 U.S. Surveillance Law and Section 702

There are four primary areas of law that regulate electronic surveillance by the U.S. government. The Fourth Amendment protects both people in the U.S. and U.S. persons (i.e., citizens and permanent residents) abroad, and it covers communications content.[1] The Electronic Communications Privacy Act (ECPA) sets procedures for domestic law enforcement access to data. FISA provides a framework for foreign intelligence surveillance conducted in the U.S. Finally, Executive Order (EO) 12333 addresses extraterritorial intelligence collection.

While a comprehensive review of U.S. surveillance law is beyond the scope of this project (see [53, 74]), there is an important interplay between law and technology that led to Section 702 and the estimation challenge we address [36]. In the pre-2000s era, before modern online services and global networks, there was a general bright-line rule for domestic collection of communications content: the government had to obtain a warrant, reviewed by a judge and supported by probable cause and particularity. If the government sought content outside the U.S., by contrast, no warrant was needed—ECPA and FISA did not apply, and the Fourth Amendment and EO 12333 required (at most) limited non-judicial procedures.

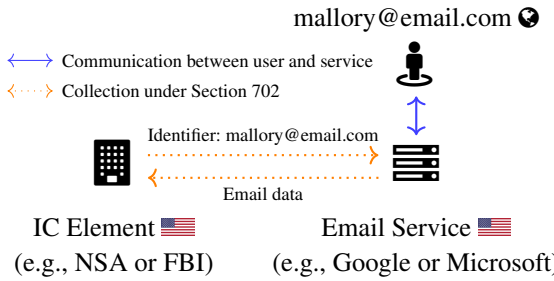


Figure 1: An example of Section 702 surveillance. Mallory uses mallory@email.com, is outside the U.S., and is not a U.S. person. They do, however, use a U.S.-based email service. Section 702 authorizes the IC to collect their email for intelligence purposes by sending a directive to the service.

The Internet created an opportunity and a dilemma for the IC. Popular U.S.-based online services became platforms for worldwide communication. Telecommunications services in the U.S. also became international network hubs. The IC could obtain foreign content from these services through domestic legal process, instead of burdensome collection abroad. But under current law, obtaining that data required a warrant—unlike procedures for extraterritorial surveillance.

The Bush administration and Congress responded with a hybrid procedure in Section 702, enacted as part of the FISA Amendments Act of 2008.[2] Section 702 would allow the IC to collect content from U.S.-based online services and telecommunications networks, when targeting foreigners abroad, without obtaining a warrant. But Section 702 also created a role for the judiciary: the FISC, a special court that adjudicates FISA matters, would conduct an annual review of procedures. The court would have to determine that the procedures were consistent with the Fourth Amendment and Section 702, and it could address instances of noncompliance.

In order to make Section 702 surveillance more concrete, consider the following example, which we depict in Figure 1. Suppose that an IC element, such as the National Security Agency (NSA) or the Federal Bureau of Investigation (FBI), seeks emails to and from Mallory.[3] The agency has determined that surveilling Mallory could yield foreign intelligence and that they are neither located in the U.S. nor a U.S. person. The agency also knows that Mallory uses the email address `mallory@email.com`, which is hosted by a major

[1] We use the term "U.S. person," which is defined in FISA, for brevity. Courts have interpreted the Fourth Amendment to have similar extraterritorial applicability based on a person's U.S. citizenship or permanent residency.

[2] The Protect America Act of 2007 briefly preceded the FISA Amendments Act with a similar hybrid procedure, before sunsetting in 2008.

[3] An IC "element" is a federal agency or a component of a federal agency that Congress or the President has designated as part of the IC.

email provider such as Google or Microsoft.[4] After the IC completes its annual review with the FISC, agency officials approve targeting Mallory. An IC element serves a Section 702 directive on the email provider and specifies Mallory's address as an identifier for collection.[5] The email provider is then compelled to disclose messages to and from the address.[6]

Section 702 was exceptionally controversial when enacted, and it remains a sticking point in the U.S. and abroad. In a pair of decisions, for example, the Court of Justice of the European Union determined that Section 702 provided such limited protections for Europeans that it would invalidate certain commercial data flows to the U.S. [13,22]. Meanwhile, Presidents Obama and Trump both signed bills reauthorizing Section 702, and the IC maintains that it is among the most important national security authorities [62]. The provision is currently scheduled to sunset on December 31, 2023.

We take no position in this work on the merits of Section 702 and whether it strikes an appropriate balance between national security and civil liberties. As we discuss in the following sections, we focus narrowly on a specific Section 702 issue, incidental collection, and our aim is to enable quantitative estimates that would inform the issue.

2.2 Incidental Collection and U.S. Person Queries

Perhaps the greatest controversy related to Section 702, at least in the U.S., is incidental collection. The concept is straightforward: Americans talk to foreigners. If an IC element targets a foreigner outside the U.S. for surveillance under Section 702, the agency may collect communications to or from persons inside the U.S. or U.S. persons abroad. These are people protected by the Fourth Amendment, whose communications content the government could ordinarily only collect with a warrant. The possibility of incidental collection is compounded by the scale of Section 702 surveillance (hundreds of thousands of targets per year [65]) and the fact that foreigners using U.S. services may be more likely to communicate with persons in the U.S. and U.S. persons abroad.

As above, we offer an example for clarity, which we depict in Figure 2. Suppose that Alice, who is not a foreign intelligence target, exchanges messages with Mallory using the same email provider. Alice is located within the U.S. and a U.S. person. When an IC element obtains Mallory's email, it incidentally collects messages to and from Alice.

[4]We focus on "downstream" collection via email providers for simplicity and because the IC has acknowledged that type of Section 702 surveillance. The estimation method that we propose generalizes to other types of surveillance, including "upstream" collection from telecommunications networks, so long as the IC possesses identifiers for individual senders and recipients.

[5]The IC element that sends the directive or the selector to the email provider might differ from the IC element seeking to collect Mallory's emails.

[6]Note that communications "identifiers" may not neatly map to participating persons. We discuss this conceptual distinction further in Section 3.

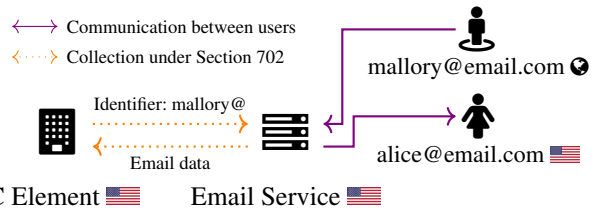


Figure 2: An example of Section 702 incidental collection. The target, Mallory, is neither in the U.S. nor a U.S. person. They send a message to Alice, who is located in the U.S. and a U.S. person. As a result, the IC incidentally collects on Alice.

After incidentally collecting communications, IC elements can query the data with U.S. person identifiers for both intelligence and law enforcement purposes [36].[7] These types of queries, referred to by the IC as "U.S. person queries" and by critics as "backdoor searches," have been a focal point for Section 702 reauthorization and reform. Opponents of Section 702 characterize these queries as an end-run around constitutional privacy protections. The FISC and other courts have consistently allowed U.S. person queries, and Congress has approved the practice—though it nearly instituted a warrant requirement for U.S. person queries in a 2018 reauthorization.

2.3 Estimating Incidental Collection

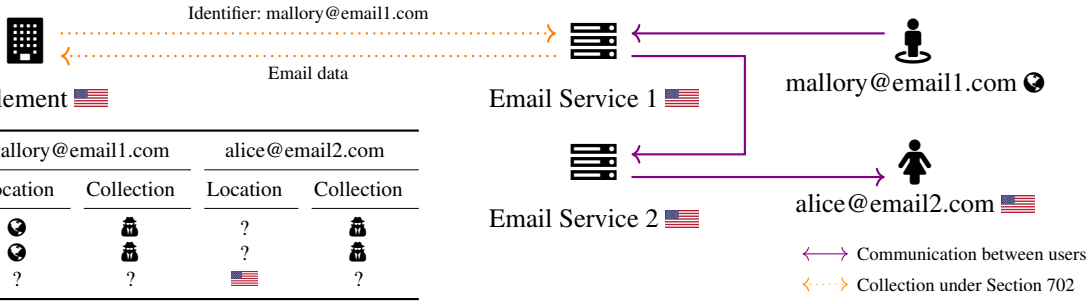
In response to the incidental collection controversy, legislators and civil society groups have urged the IC to quantitatively estimate the issue. Understanding the scale of incidental collection, they argue, is essential for evaluating whether to reauthorize Section 702, what reforms may be important, and the authority's Fourth Amendment "reasonableness."

Initial versions of the legislation that became Section 702, passed by the House and reported out of the Senate Judiciary Committee and the Senate Select Committee on Intelligence, would have required recurring estimates of incidental collection [3, 4]. The Bush administration took the position that estimates would be "impossible" [63], so Congress reduced the requirement to a nudge: if the IC established procedures for estimating incidental collection, it would have to provide the procedures and estimates to the FISC and Congress [2].

The earliest request for an estimate of incidental collection after Congress enacted Section 702 was in July 2011, when a pair of Senators sought context for an upcoming reauthorization [5]. The Director of National Intelligence (DNI) responded that an estimate would not be possible [6]. The Senators asked the IC Inspector General in mid-2012 [8] and a larger bipartisan group wrote to the DNI again several months after [7]. Both requests received similar responses [9, 10].

In 2014 the Privacy and Civil Liberties Oversight Board (PCLOB), an independent agency, issued a report on Section

[7]The Section 702 statute does not restrict query purposes. FISC-approved procedures generally require that a query be "reasonably likely to retrieve foreign intelligence information" or, for the FBI, "evidence of a crime" [64].



	mallory@email1.com		alice@email2.com	
Knowledge State	Location	Collection	Location	Collection
IC Element	🌐	🕵	?	🕵
Email Service 1	🌐	🕵	?	🕵
Email Service 2	?	?	🇺🇸	?

Figure 3: A more detailed example of Section 702 incidental collection, in which Mallory and Alice use different email services. The IC knows it has collected an email to alice@email2.com, but not the user's location. Alice's email service knows they are in the U.S., but not that they have been surveilled. We propose using MPC to privately compute statistics about incidental collection.

702 [67]. PCLOB noted that the amount of incidental collection was "one of the biggest open questions" about the authority. Because there was an "impasse" about how to estimate incidental collection, PCLOB recommended alternative transparency statistics that would offer "partial insight."

Civil society groups began their own campaign for an estimate of incidental collection in October 2015 [11]. The DNI's staff did not directly respond, instead offering a status update on related oversight recommendations [12, 14].

The bipartisan leadership of the House Judiciary Committee next took up the issue, requesting an estimate in April 2016 in advance of another Section 702 reauthorization [15]. This time, the DNI personally oversaw an interagency process to generate an estimate. Members sent a follow-up letter in December 2016 [16], the incoming DNI confirmed he would prioritize an estimate at a February 2017 Senate hearing [20], members sent another letter in April 2017 [18]—and then, to widespread surprise, the new DNI announced at a June 2017 hearing that there would be no estimate [21]. Generating an estimate would be "infeasible," he explained, because it would require manual identification of persons in the U.S. and U.S. persons—both a burden on intelligence analysts and an additional privacy intrusion. "[I]f someone out there knows how" to acceptably estimate incidental collection, the DNI noted, both he and the NSA Director would "welcome the advice." Legislators and civil society groups responded with outrage at the change in position [17], but the DNI reaffirmed the following month that an estimate was presently "impossible" and the effort to generate an estimate had concluded [19].

To this day, the IC has not generated an estimate of the scale of Section 702 incidental collection. Our goal in this work is to demonstrate a possible path forward, accepting the IC's public invitation to propose novel estimation methods.

3 Problem Formulation

We formulate the problem of estimating Section 702 incidental collection in two stages.[8] First, we explain our conceptual approach and its limitations. We propose combining collection data held by the IC and location data held by communications services. Second, we offer a formalization of the problem, including ideal functionalities and a threat model.

3.1 Conceptual Approach

The fundamental challenge for estimating Section 702 incidental collection is that the IC (intentionally) does not collect location or nationality data for individuals whose communications are incidentally collected. The foundation of our approach is a recognition that communications services, such as email providers and social media platforms, *do* possess relevant location data—and that data could be combined with IC data about communications collected under Section 702.[9]

Communications services maintain user location data for a range of routine business purposes, including providing service, marketing, business analytics, personalized content, legal compliance, and shareholder reporting. This location data can originate from a variety of sources, such as device sensors (e.g., GPS and Wi-Fi positioning), network connectivity (e.g., IP geolocation or cell site location), or account information (e.g., a mailing address or selected country). We make no assumptions about the type of location data or precision beyond country-level granularity.[10] Our approach requires only that a communications service possess a set of identifiers (e.g., email addresses, telephone numbers, or usernames) that it believes are used by persons located in the United States.

Figure 3 depicts a motivating example.[11] Suppose an intelligence agency targets Mallory for Section 702 email surveillance, incidentally collecting messages to and from Alice.

[8] We focus on Section 702 and incidental collection involving the U.S. because of the issue's sustained controversy. Our approach generalizes to other legal authorities, such as Executive Order 12333, and other countries.

[9] This work emphasizes online communications services as MPC participants, because these services likely possess high-quality country-level location data from direct relationships with users. Other entities that could map communications identifiers to countries, such as broadband providers, e-commerce platforms, and financial services, could also participate.

[10] Data quality can vary by location method, especially in IP geolocation. Section 9 discusses an extension to account for varying location confidence.

[11] Note that Figure 3 does not depict knowledge about U.S. person status or the mapping between persons and identifiers, which we discuss in Section 3.2.

Alice likely uses a popular email service, and that service knows that Alice accesses the service from the United States.

A simple information sharing arrangement between the IC and communications services would not be viable. The IC could not disclose identifiers affected by incidental collection, because that data would reveal classified intelligence sources and methods. Communications services could not disclose user locations, because that would breach user privacy and run afoul of ECPA—which generally forbids sharing customer records with a government agency absent legal process [1].

The IC could attempt to determine Alice's location or nationality through open-source investigation or commercial data (e.g., [57]).[12] Public information about Alice's email address may be unavailable, however, and acquired data may be questionable. As the IC has noted, this approach would also be burdensome for analysts and further intrude on privacy.

We propose using MPC to estimate Section 702 incidental collection without these privacy pitfalls. The IC would maintain secrecy for surveillance activities, and communications services would maintain confidentiality for user locations.

Our approach would generate two aggregate transparency statistics: 1) a count of identifiers that are affected by incidental collection and are used by a person in the U.S., and 2) a count of intercepted communications where a sending or receiving identifier is used by a person in the U.S. The IC could integrate these statistics into its annual surveillance transparency report, which already provides public counts for Section 702 orders, targets, and U.S. person queries [65].

We developed and scrutinized this approach through extensive unclassified consultation with intelligence professionals who had senior experience at the Office of the Director of National Intelligence, the National Security Agency, the Federal Bureau of Investigation, and the Central Intelligence Agency. We also benefited from the expertise of individuals with oversight experience at the Senate Select Committee on Intelligence, the House Permanent Select Committee on Intelligence, and the Privacy and Civil Liberties Oversight Board. We additionally received valuable input from civil liberties groups, privacy law scholars, and security researchers. We gratefully acknowledge these essential contributions, and we emphasize that the approach we propose in this work has not been endorsed by any component of the U.S. government.

3.2 Limitations

Before formalizing our proposal as an MPC problem, we note several important limitations to the conceptual direction.

- The approach that we propose would generate an estimate of incidental collection, not a definitive count. Communications services may possess incomplete or inaccurate data about identifiers or locations, and changing protocol participants may significantly change output. When presenting an estimate, explanation and context would be essential.
- As the IC's leadership has acknowledged the importance of estimating incidental collection, and because the IC already generates annual transparency statistics for Section 702, we assume that an aggregate estimate of incidental collection would not risk intelligence sources and methods. If releasing a figure would be problematic, the IC could add noise or provide an interval for the value. We discuss a differential privacy extension to our protocol in Section 9.
- Our proposal would estimate incidental collection on persons located in the U.S., but would not account for U.S. persons abroad. Online services generally do not hold nationality data, and we do not assume its availability. Legislators and civil society groups requesting estimates have noted that quantification based on location would be valuable [11, 14, 17], and the IC already uses location as a proxy for nationality in Section 702 querying procedures [65].
- We propose counting identifiers rather than persons. Mapping identifiers to persons is a challenge: a user may have multiple email addresses, for example, or multiple users may share an address. Calls for estimating incidental collection have also accepted this limitation [11, 14, 17], and IC transparency statistics for certain queries of Section 702 collection already count identifiers rather than persons [65].
- The estimated count of incidentally collected communications that we propose treats a communication to multiple recipients as equivalent to a communication to each recipient. This approach simplifies the MPC and enables quantifying the average amount of incidental collection per affected identifier (by dividing the two protocol outputs). To the extent this approach involves duplicate counting—a matter of perspective—we note that IC transparency statistics for certain Section 702 queries already include duplicates [65].
- PSI protocols can introduce probabilistic inaccuracies in computation. We describe how false negatives can occur in our MPSI and MPSIU-Sum constructions, and we empirically evaluate false negatives in MPSIU-Sum benchmarks.

3.3 Formalization

IC element $\mathcal{P}_0$ holds communications identifiers X_0 that were non-target senders or recipients of communications collected under Section 702.[13] $\mathcal{P}_0$ also holds integer values V that are counts of collected communications for identifiers in X_0. Communications services $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ hold sets $X_1, \ldots, X_{n-1}$ of identifiers they believe are used by persons located in the U.S.

$\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ run an MPSIU-Sum protocol with $\mathcal{P}_0$ as the delegate for output (Figure 4). At the conclusion of the protocol, $\mathcal{P}_0$ learns $|I|$ and $\sum_{x \in I} V[x]$ where $I = X_0 \bigcap (\bigcup_{i=1}^{n-1} X_i)$. $|I|$ is an estimated count of identifiers used by persons in the

[12] We do not evaluate these methods for estimating incidental collection, so we take no position on the complex accuracy, burden, and privacy tradeoffs.

[13] We assume that one IC element would coordinate protocol participation for the IC. Related formalizations could include multiple IC elements, an independent oversight entity as delegate, or no government component at all.

$\mathcal{F}_{\mathsf{MPSI}}$
Parties: $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ with delegate $\mathcal{P}_0$.
Inputs: $X_i \subseteq \{0,1\}^*$ held by party $\mathcal{P}_i$.
Outputs: $\mathcal{P}_0$ receives $I = \bigcap_{i=0}^{n-1} X_i$. Others receive nothing.

$\mathcal{F}_{\mathsf{MPSIU\text{-}Sum}}$
Parties: $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ with delegate $\mathcal{P}_0$.
Inputs: $X_i \subseteq \{0,1\}^*$ held by party $\mathcal{P}_i$ and associated values $V \subseteq \mathbb{F}_q$ held by the delegate.
Outputs: $\mathcal{P}_0$ receives $|I|$ and $\sum_{x \in I} V[x]$ where $I = X_0 \bigcap (\bigcup_{i=1}^{n-1} X_i)$. Other parties receive nothing.

Figure 4: Ideal functionalities for Multiparty Private Set Intersection ($\mathcal{F}_{\mathsf{MPSI}}$) and Multiparty Private Set Intersection with Union and Sum ($\mathcal{F}_{\mathsf{MPSIU\text{-}Sum}}$). Note that the $\mathcal{F}_{\mathsf{MPSIU\text{-}Sum}}$ definition includes both cardinality and sum outputs.

U.S. that were affected by incidental collection. $\sum_{x \in I} V[x]$ is an estimated count of communications to or from persons in the U.S. that were affected by incidental collection.

3.4 Threat Model

We aim to preserve confidentiality for both the IC and users, with malicious security against information disclosure.

Protecting Intelligence Sources and Methods. The communications services $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ do not learn new information about the identifiers in X_0 or counts in V, because those values reflect specific instances of Section 702 surveillance.

Protecting User Privacy. The IC element $\mathcal{P}_0$ does not learn new information about identifiers in $X_1, \ldots, X_{n-1}$, other than from protocol output, because those values reflect the locations of persons using identifiers. The services $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ do not learn new information about identifiers in $X_1, \ldots, X_{n-1}$.

The MPSI and MPSIU-Sum protocols that we present achieve these objectives, providing security against a malicious $\mathcal{P}_0$ or any colluding subset of $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$. We provide a simulation-based security proof for both protocols (Sections 5.3 and 6.3), and we offer an extension for malicious security against any proper subset of participants (Section 9). Our constructions also have the following security properties.

- The protocols do not prevent intentional false positives, false negatives, inaccurate set cardinality, or inaccurate sum computation, because MPC participants can generally cheat with input. Malicious participants could also manipulate the protocols to induce errors. Semi-honest participants with truthful input will not induce errors, other than a probabilistic risk of false negatives (Sections 5.2, 6.2, and 7). Surveillance transparency reports already depend on the IC and communications services for trustworthy counting, so this property is consistent with current practices.
- The protocols do not prevent a participant from intentionally revealing known information through an out-of-band or repurposed in-band channel, as is generally true of MPC. This property is also consistent with the status quo.
- The aggregate cardinality and sum output from MPSIU-Sum may reveal information about $X_1, \ldots, X_{n-1}$ to $\mathcal{P}_0$.[14] Adding random noise into the protocol can mitigate that risk and achieve differential privacy (Section 9).

4 Preliminaries

Before presenting our MPSI and MPSIU-Sum protocols, we describe data structure and cryptographic primitives that are foundational for the constructions. Our notation here and throughout the balance of the paper generally follows conventions for the Apple PSI protocol that we extend, so that readers can better compare the protocols and associated proofs [28].

Hashmap Generation. Our constructions use hashmaps that rely on a collision-resistant cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^{l'}$. For a hashmap of size $m = 2^l$ with $l < l'$, the l-bit index of a string s is the unsigned integer representation of an l-bit prefix of $H(s)$. We denote this as $\mathsf{index}(s, l)$ and omit l when it is clear from context.

Elliptic Curves. All arithmetic over elliptic curves is performed on NIST P-256 [31], unless otherwise specified.

Hashing to Elliptic Curves. We use the `map_to_curve` functionality from the IETF Hashing to Elliptic Curves Internet-Draft, because the resulting hash function $H_E : \{0,1\}^* \to E$ can be modeled as a random oracle [40].

Diffie-Hellman Random Self-Reduction. In a group of order q with generator G where the DDH problem is hard, define an operation DH.Reduce for random scalars $\beta, \gamma \xleftarrow{\$} \mathbb{F}_q$.

$$\mathsf{DH.Reduce}(L, T, P) = (\beta \cdot T + \gamma \cdot G, \beta \cdot P + \gamma \cdot L)$$

Recall that (L, T, P) is a DH tuple if and only if $L = \alpha \cdot G$ and $P = \alpha \cdot T$ for some $\alpha \in \mathbb{F}_q$. Naor and Reingold show that DH.Reduce reduces DH tuples to DH tuples uniformly sampled in $E(\mathbb{F}_q)$ and non-DH tuples to random values uniformly sampled in $E(\mathbb{F}_q)$ [28, 60]. Suppose $(T', P') \leftarrow \mathsf{DH.Reduce}(L, T, P)$. If (L, T, P) is a DH tuple with $L = \alpha \cdot G$,

$$\begin{aligned} P' &= \beta \cdot P + \gamma \cdot L = \beta \cdot (\alpha \cdot T) + \gamma \cdot (\alpha \cdot G) \\ &= \alpha \cdot (\beta \cdot T + \gamma \cdot G) = \alpha \cdot T'. \end{aligned}$$

Otherwise, if (L, T, P) is a non-DH tuple, then T' and P' are uniformly random as they are linear combinations of

[14] A malicious $\mathcal{P}_0$ could exploit the property by encoding X_0 items into V values. That gambit would run afoul of information disclosure restrictions in ECPA [1], and regardless, the same noise mitigation would apply.

Key-Aggregation

Public Parameters: EC group $E(\mathbb{F}_q)$ with generator G.
Parties: $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$.
Inputs: Party $\mathcal{P}_i$ holds $\mathsf{sk}_i \in \mathbb{F}_q$ for $0 \leq i \leq n-1$.
Outputs: Aggregated public key apk.

1. (All Parties) Compute $\mathsf{pk}_i \leftarrow \mathsf{sk}_i \cdot G$ and broadcasts pk_i to all other parties.
2. (All Parties) Compute $\mathsf{apk} \leftarrow \sum_{i=0}^{n-1} \mathsf{pk}_i$.

Figure 5: Sub-protocol Key-Aggregation is run at the start of MPSIU-Sum to compute an aggregated public key apk.

T and P with uniformly random coefficients β and γ [60]. Furthermore, if (L, T_1, P_1) and (L, T_2, P_2) are DH tuples, so is (L, T_1+T_2, P_1+P_2) as $P_1+P_2 = \alpha \cdot (T_1+T_2)$ with $L = \alpha \cdot G$. Similarly, the sum of a DH tuple and a non-DH tuple is a non-DH tuple, and the sum of a DH tuple and a random tuple is a random tuple. We use these DH tuple additive properties to extend the Apple PSI protocol to the multiparty setting [28].

Authenticated Encryption with Associated Data. Our constructions require an authenticated encryption with associated data (AEAD) scheme, which ensures both confidentiality and integrity of encrypted data. We denote a scheme with key space $\mathcal{K}_{\mathsf{AEAD}}$ as $(\mathsf{AEAD.Enc}, \mathsf{AEAD.Dec})$ and instantiate it with AES-GCM in our implementation. We derive AEAD keys from elliptic curve points as (KDF) using SHA256.

Distributed ElGamal Cryptosystem (DEG). The ElGamal cryptosystem is an asymmetric partially homomorphic encryption scheme. For semantic security, the ElGamal cryptosystem is initialized over a cyclic group in which the DDH problem is hard. We define the scheme in an elliptic curve group $E(\mathbb{F}_q)$, with generator G, for practical efficiency gains [39, 50]. A distributed version of the cryptosystem allows parties to jointly generate an aggregated public key apk such that data encrypted under apk can only be decrypted jointly by all parties. In particular, no single party (e.g., the delegate $\mathcal{P}_0$) can decrypt ciphertexts encrypted under apk.

DEG: Public Key Aggregation. Secret keys are integers in $\mathbb{F}_q$. The (ideal) aggregated secret key is the sum (modulo q) of individual secret keys generated by each party. The public key associated with a secret key sk is the elliptic curve point $\mathsf{sk} \cdot G$. It follows that the aggregated public key apk is simply the sum of individual public keys (see Figure 5). We denote the public key space as $\mathcal{K}_{\mathsf{EG}}$.

DEG: Partial Homomorphism. A message $m \in \mathbb{F}_q$ is represented by the elliptic curve point $m \cdot G$, which preserves additive homomorphism. The encryptions of messages m_1, m_2 under public key apk are $(r_1 \cdot G, r_1 \cdot \mathsf{apk} + m_1 \cdot G), (r_2 \cdot G, r_2 \cdot \mathsf{apk} + m_2 \cdot G)$ for $r_1, r_2 \xleftarrow{\$} \mathbb{F}_q$ respectively. It follows that

$$(r_1 \cdot G, r_1 \cdot \mathsf{apk} + m_1 \cdot G) + (r_2 \cdot G, r_2 \cdot \mathsf{apk} + m_2 \cdot G)$$

Joint-Decryption

Public Parameters: EC group $E(\mathbb{F}_q)$ with generator G, number of CRT moduli c, ciphertext space $\mathcal{C}$.
Parties: $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ with delegate $\mathcal{P}_0$.
Inputs: Party $\mathcal{P}_i$ holds $\mathsf{sk}_i \in \mathbb{F}_q$ for $0 \leq i \leq n-1$ and ElGamal ciphertext $\mathsf{ct} = (C_1, C_2) \in \mathcal{C}$.
Outputs: Party $\mathcal{P}_0$ learns the plaintext encrypted in ct.

1. (All Parties) Compute
$$d_i \leftarrow \mathsf{sk}_i \cdot C_1$$
2. (Party $\mathcal{P}_i$ for $0 < i < n$) Send d_i to $\mathcal{P}_0$.
3. (Delegate $\mathcal{P}_0$) Computes
$$T \leftarrow C_2 - \sum_{i=0}^{n-1} d_i$$

And the discrete log of T (base G) using BSGS.

Figure 6: Sub-protocol Joint-Decryption is run at the end of MPSIU-Sum, allowing $\mathcal{P}_0$ to learn the computed sum.

$$= ((r_1+r_2) \cdot G, (r_1+r_2) \cdot \mathsf{apk} + (m_1+m_2) \cdot G).$$

DEG: Decryption. Notice that decryption in this scheme amounts to solving the Discrete Log Problem (DLP) over $E(\mathbb{F}_q)$. We use a precomputed table and the baby-step giant-step (BSGS) algorithm to find m given $m \cdot G$ [71, 72]. We discuss a Chinese Remainder Theorem-based version of the ElGamal cryptosystem which makes decryption tractable for large integer values in Supplementary Information [54]. Note that MPSIU-Sum (Figure 13) requires the decryption of only a *single* ElGamal ciphertext (Figure 6). Using 2 CRT moduli of at least 16 bits each and a small precomputed table (e.g., of size 2^{16}), the BSGS algorithm computes the discrete log m of $m \cdot G$ well under 500 ms for $m < 2^{32}$ on commodity hardware.

5 Multiparty Private Set Intersection (MPSI)

In this section, we construct an MPSI protocol as a step toward our MPSIU-Sum protocol. We begin by describing a variant of the Apple PSI protocol, which is based on the insight that DH random self-reduction can be used to both detect matches and protect associated data for a match [28]. We then generalize the protocol to the multiparty setting, yielding a practical MPSI protocol. We prove protocol correctness, analyze false negatives, and provide a simulation-based proof of security.

PSI. Assume that delegate $\mathcal{P}_0$ holds set X_0 and party $\mathcal{P}_1$ holds set X_1. $\mathcal{P}_0$ and $\mathcal{P}_1$ would like to jointly compute $X_0 \cap X_1$ and output to $\mathcal{P}_0$, without revealing other information.

Informally, our PSI construction tracks the Apple PSI protocol: $\mathcal{P}_0$ creates a table M of values in X_0 blinded with key α, $\mathcal{P}_1$ computes a DH random self-reduction for each value in X_1 with the corresponding value in M, $\mathcal{P}_1$ encrypts associated

PSI

Public Parameters: EC group $E(\mathbb{F}_q)$ with generator G, hash function $H_E : \{0,1\}^* \to E$, map size $m = 2^l$.
Parties: Delegate $\mathcal{P}_0$ and party $\mathcal{P}_1$.
Inputs: $\mathcal{P}_0$ holds set $X_0 \subset \{0,1\}^*$, $\mathcal{P}_1$ holds set $X_1 \subset \{0,1\}^*$.
Outputs: $\mathcal{P}_0$ receives $X_0 \cap X_1$, $\mathcal{P}_1$ receives nothing.

1. (Delegate $\mathcal{P}_0$) Generates $\mathsf{sk} \xleftarrow{\$} \mathcal{K}_{\mathsf{AEAD}}$ and $\alpha \xleftarrow{\$} \mathbb{F}_q$, and sets $L = \alpha \cdot G$. Initializes map M of size m. For every $w_j \in X_0$, sets $M_{\mathsf{index}(w_j)} \leftarrow \{\alpha \cdot H_E(w_j), \mathsf{Enc}(\mathsf{sk}, w_j)\}$. For every unmodified index $0 \le j < m$ in M, sets $M_j \leftarrow \{r_1 \cdot G, \mathsf{Enc}(\mathsf{sk}, r_2)\}$ for $r_1, r_2 \xleftarrow{\$} \mathbb{F}_q$. Sends M, L to $\mathcal{P}_1$.
2. (Party $\mathcal{P}_1$) Initializes map R of size m. For every $w \in X_1$, computes $j \leftarrow \mathsf{index}(w)$ and sets $R_j \leftarrow \mathsf{DH.Reduce}(L, H_E(w), M_{j,0})$. For every unmodified index $0 \le j < m$ in R, sets $R_j \leftarrow \{Q', S'\}$ for $Q', S' \xleftarrow{\$} E$. Initializes array B of size m. For $0 \le j < m$, sets $B_j \leftarrow \{R_{j,0}, \mathsf{AEAD.Enc}(\mathsf{KDF}(R_{j,1}), M_{j,1})\}$. Sends R to $\mathcal{P}_0$.
3. (Delegate $\mathcal{P}_0$) For $0 \le j < m$, computes $K_j \leftarrow \mathsf{KDF}(\alpha \cdot B_{j,0})$, $d_j \leftarrow \mathsf{AEAD.Dec}(K_j, B_{j,1})$ and the output $D \leftarrow \{d_j : d_j \neq \perp, 0 \le j < m\}$.

Figure 7: A formalization of the complete PSI protocol.

data with a symmetric key derived from the DH self-reduction, and finally $\mathcal{P}_0$ attempts decryption of the associated data. Our protocol differs in that $\mathcal{P}_0$ creates a hashmap instead of a Cuckoo table for multiparty coordination, $\mathcal{P}_0$ provides encrypted associated data to enable the sum in MPSIU-Sum, $\mathcal{P}_1$ generates a new hashmap R from every value in M to protect $|X_1|$ and for multiparty coordination, and $\mathcal{P}_1$ permutes the array B that it sends to $\mathcal{P}_0$ to protect individual matches in MPSIU-Sum. We formalize the PSI protocol in Figure 7.

For simplicity, assume no hashmap collisions (see Section 5.2). Assuming the DDH problem is hard over $E(\mathbb{F}_q)$, the discrete log problem is also hard over $E(\mathbb{F}_q)$, so M does not reveal anything about the elements of X_0 to a computationally-bounded $\mathcal{P}_1$. $\mathcal{P}_0$ can only decrypt elements from $\mathcal{P}_1$ that are in $X_0 \cap X_1$, such that $\mathcal{P}_0$ can recover the key derived from $\mathcal{P}_1$'s self-reduction, and does not learn other information about X_1.

PSI $\to$ MPSI. We generalize the PSI protocol to the multiparty setting by using the additive properties of DH tuples. $\mathcal{P}_0$ and $\mathcal{P}_1$ initialize the protocol as above. The parties $\mathcal{P}_2, \ldots, \mathcal{P}_{n-1}$ then sequentially pass parameter L, hashmap M, and hashmap R, updating R to incorporate each party $\mathcal{P}_i$'s set X_i. If a party has a set item for a hashmap index, it computes a DH self-reduction with M (as in PSI) and adds the computed value to the value already in R. If a party does does not have an item for an index, it sets a random value in R.

The conclusion of the protocol is the same as in PSI. The last party $\mathcal{P}_{n-1}$ constructs an array B, encrypting the associated data from $\mathcal{P}_0$ with keys derived from the values in R. $\mathcal{P}_{n-1}$ permutes B and sends it to $\mathcal{P}_0$. $\mathcal{P}_0$ completes the protocol, decrypting ciphertexts associated with DH tuples and yielding the intersection $\bigcap_{i=0}^{n-1} X_i$. We present an overview of the protocol in Figure 8 and the formal protocol in Figures 9, 10, 11, and 12. Section 9 discusses possible extensions.

5.1 Correctness

Theorem 1. *Assuming semi-honest participants, false positives are not possible in MPSI.*

Proof. Let $I = \bigcap_{i=0}^{n-1} X_i$ and suppose, on the contrary, that $x \notin I$ but $x \in D$ (in Delegate-Finish) with $\mathsf{index}(x) = j$. As $\mathcal{P}_0$ was able to decrypt $B_{j,1}$ using key $\alpha \cdot B_{j,0}$, it follows that $x \in X_0$ and $M_{j,0} = \alpha \cdot H_E(x)$. Before Blind-Encrypt, $(L, R_{j,0}, R_{j,1})$ must have formed a DH tuple as $B_{j,1}$ was encrypted using $R_{j,1} = \alpha \cdot B_{j,0} = \alpha \cdot R_{j,0}$. Since $x \notin I$, it follows that $x \notin X_i$ for some $i > 0$. There are two possibilities:

- $y \in X_i$ such that $y \neq x$ but $\mathsf{index}(y) = \mathsf{index}(x) = j$. $\mathcal{P}_i$ adds $\mathsf{DH.Reduce}(L, H_E(y), M_{j,0})$ to R_j. As $H_E(y) \neq H_E(x)$, $(L, H_E(y), M_{j,0})$ is not a DH tuple, R_j will be a random tuple due to the self-reduction property.
- There exists no $y \in X_i$ such that $\mathsf{index}(y) = j$. $\mathcal{P}_i$ explicitly sets R_j to a random tuple.

In either case, R_j is set to a random tuple by $\mathcal{P}_i$. Even if subsequent parties add DH tuples to this random tuple (if they have x in their sets), the result will not be a DH tuple, except with negligible probability. Therefore, $(L, R_{j,0}, R_{j,1})$ cannot be a DH tuple before Blind-Encrypt, yielding a contradiction. □

Theorem 2. *Assuming semi-honest participants, false negatives in MPSI are only caused by collisions in hashmap M.*

Proof. Let $I = \bigcap_{i=0}^{n-1} X_i$, $x \in I$ with $\mathsf{index}(x) = j$. A false negative implies that $\mathcal{P}_0$ was unable to decrypt $B_{j,1}$ with key $\alpha \cdot B_{j,0}$ in Delegate-Finish. It follows that $(L, R_{j,0}, R_{j,1})$ was not a DH tuple before Blind-Encrypt. As $x \in I$, no party explicitly set R_j to a random tuple. It follows that some party, say $\mathcal{P}_i$, added a non-DH tuple $(L, H_E(y), M_{j,0})$ to R_j during its turn. There are two possibilities:

- $x \neq y \implies H_E(y) \neq H_E(x)$ but $\mathsf{index}(y) = \mathsf{index}(x) = j$, which implies a collision in M.
- $x = y$ and $M_{j,0} \neq \alpha \cdot H_E(x)$, which is only possible if $\exists y' \in X_0$ such that $\mathsf{index}(y') = \mathsf{index}(x) = j$ and $M_{j,0} = \alpha \cdot H_E(y')$, which also implies a collision in M. □

5.2 False Negatives

Expected Number of Filled Slots. Assume a cryptographic hash function yields uniformly random indices. After w

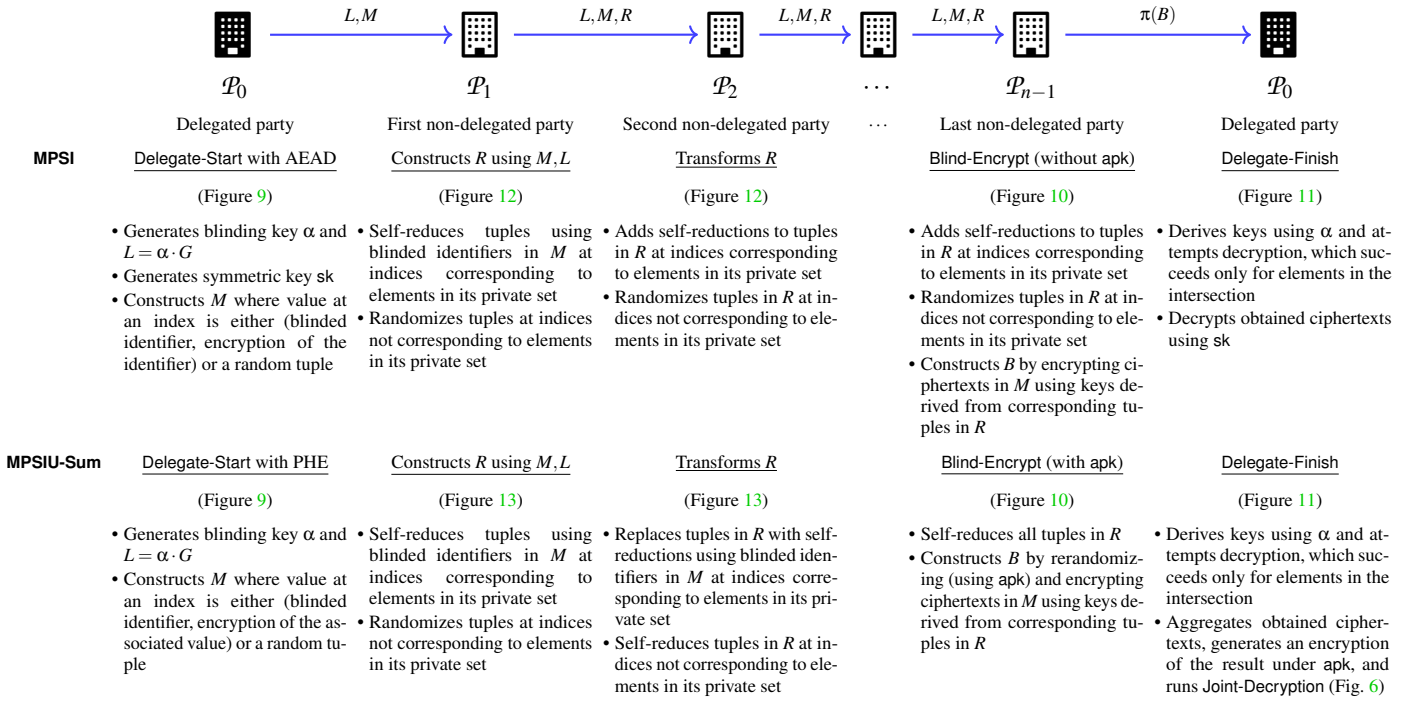


Figure 8: An illustration of the MPSI and MPSIU-Sum protocols, omitting the Key-Aggregation and Joint-Decryption sub-protocols.

Delegate-Start

Public Parameters: EC group $E(\mathbb{F}_q)$ with generator G, hash function $H_E : \{0,1\}^* \to E$, map size $m = 2^l$.
Parties: Delegate $\mathcal{P}_0$.
Inputs: $\mathcal{P}_0$ holds set $X_0 \subset \{0,1\}^*$, associated values $V \subset \{0,1\}^*$, encryption key ek, encryption function Enc.
Outputs: Blinding key α, EC point L, and hashmap M.

1. Generates $\alpha \xleftarrow{\$} \mathbb{F}_q$ and sets $L = \alpha \cdot G$.
2. For every $w_j \in X_0, v_j \in V$, sets $M_{\mathsf{index}(w_j)} \leftarrow \{\alpha \cdot H_E(w_j), \mathsf{Enc}(\mathsf{ek}, v_j)\}$.
3. For every unmodified index $0 \le j < m$ in M, sets $M_j \leftarrow \{r_1 \cdot G, \mathsf{Enc}(\mathsf{ek}, r_2)\}$ for $r_1, r_2 \xleftarrow{\$} \mathbb{F}_q$.

Figure 9: Delegate $\mathcal{P}_0$ runs sub-protocol Delegate-Start at the beginning of MPSI and after Key-Aggregation in MPSIU-Sum.

unique items are inserted into a hashmap of size m, the probability that a slot is empty is $\left(1 - \frac{1}{m}\right)^w$. For large m, using the identity $\lim_{m\to\infty}\left(1 - \frac{1}{m}\right)^m = e^{-1}$, the probability that a given slot is empty is approximately $e^{\frac{-w}{m}}$.[15] The expected number of filled slots when w unique items are inserted into a hashmap of size m is

$$\mathsf{ExpFilled}(m,w) = m\left(1 - \left(1 - \frac{1}{m}\right)^w\right) \approx m(1 - e^{\frac{-w}{m}}).$$

[15] This is the same probability that a given bit will not be set in a Bloom filter with just one hash function. The expected number of hash collisions is usually not a quantity of interest in analyses of Bloom filters [44].

Blind-Encrypt

Public Parameters: EC group $E(\mathbb{F}_q)$, size $m = 2^l$.
Parties: Last party $\mathcal{P}_{n-1}$.
Inputs: Hashmaps R and M, ElGamal public key apk (optional).
Outputs: Array B.

1. Initializes array B of size m. For $0 \le j < m$, sets

$$\mathsf{ct}_j \leftarrow \begin{cases} \mathsf{EG.AddZero}(\mathsf{apk}, M_{j,1}) & \text{if apk was provided} \\ M_{j,1} & \text{otherwise} \end{cases}$$

$$B_j \leftarrow \{R_{j,0}, \mathsf{AEAD.Enc}(\mathsf{KDF}(R_{j,1}), \mathsf{ct}_j)\}$$

2. Samples permutation π over $\{0, \ldots, m-1\}$ and shuffles $B \leftarrow \pi(B)$.

Figure 10: The last party $\mathcal{P}_{n-1}$ runs sub-protocol Blind-Encrypt before delegate $\mathcal{P}_0$ runs Delegate-Finish. For brevity, we combine the MPSI and MPSIU-Sum steps into Blind-Encrypt.

We analyze false negatives induced by the delegate and non-delegates separately here, as some non-delegate errors are recoverable in MPSIU-Sum (Section 6.2).

Errors Induced by Delegate. Suppose $I = \bigcap_{i=0}^{n-1} X_i$ and $|I| = s_I$. Every collision of index$(\cdot)$ with these s_I elements during Delegate-Start causes a false negative. The expected number of false negatives induced by $\mathcal{P}_0$ is $e_0 = |I| \cdot (1 - \frac{1}{m}) \cdot \mathsf{ExpFilled}(m, |X_0| - |I|)$.

Delegate-Finish

Public Parameters: EC group $E(\mathbb{F}_q)$, array size $m = 2^l$.
Parties: Delegate $\mathcal{P}_0$.
Inputs: Blinding key α and array B.
Outputs: Set of AEAD plaintexts D.

1. For $0 \le j < m$, computes $K_j \leftarrow \mathsf{KDF}(\alpha \cdot B_{j,0})$ and $d_j \leftarrow \mathsf{AEAD.Dec}(K_j, B_{j,1})$.

$$D \leftarrow \{d_j : d_j \neq \perp, 0 \le j < m\}$$

Figure 11: Delegate $\mathcal{P}_0$ runs sub-protocol Delegate-Finish to conclude MPSI and before Joint-Decryption in MPSIU-Sum.

Errors Induced by Non-Delegates. A false negative is induced by a non-delegate if $\mathsf{index}(\cdot)$ causes a collision among the $|I| - e_0$ indices filled by the delegate $\mathcal{P}_0$. The expected number of such collisions is

$$e_i = \frac{|I| - e_0}{m} \cdot \mathsf{ExpFilled}(m, |X_i| - |I|).$$

If $|X_i| = s$ for all $i > 0$ and $f_0 = \mathsf{ExpFilled}(m, |X_0| - s_I)$, the total number of expected false negatives is given by

$$\begin{aligned}
&e_0 + (n-1)e_i \\
&= e_0 + \frac{(n-1)(s_I - e_0)}{m}\mathsf{ExpFilled}(m, s - s_I) \\
&= s_I(1 - \frac{1}{m})\mathsf{ExpFilled}(m, |X_0| - s_I) \\
&+ s_I \frac{(n-1)\mathsf{ExpFilled}(m, |X_0| - s_I))}{m^2}\mathsf{ExpFilled}(m, s - s_I) \\
&= s_I \cdot f_0\Big(1 - \frac{1}{m} + \frac{n-1}{m^2}\mathsf{ExpFilled}(m, s - s_I)\Big).
\end{aligned}$$

As $s_I \ll s$, the expected false negative rate is less than

$$\begin{aligned}
&f_0\Big(1 - \frac{1}{m} + \frac{n-1}{m^2}\mathsf{ExpFilled}(m, s)\Big) \\
&= f_0\Big(1 - \frac{1}{m} + \frac{n-1}{m}(1 - e^{\frac{-s}{m}})\Big) \\
&= (1 - e^{\frac{s_I - |X_0|}{m}})\Big(m - 1 + (n-1)(1 - e^{\frac{-s}{m}})\Big).
\end{aligned}$$

5.3 Security

The security properties of MPSI follow from DDH and discrete log hardness and AEAD semantic security. Intuitively, parties $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ cannot read values in M because they are encrypted by α or sk. Delegate $\mathcal{P}_0$ can only decrypt values in B that are DH tuples because of a set intersection match.

Multiple security proofs are available for the Apple PSI protocol [26, 28]. We extend the simulator-based arguments here with proof sketches, and we provide complete security proofs in the Supplementary Information [54].

Theorem 3. *A static computationally-bounded malicious adversary $\mathcal{A}$ that corrupts $\mathcal{P}_i$ for $i \in C \subset \{1, \ldots, n-1\}$ learns no information about $\{X_j : 1 \le j \le n-1, j \notin C\}$ in MPSI.*

MPSI

Public Parameters: EC group $E(\mathbb{F}_q)$ with generator G, hash function $H_E : \{0,1\}^* \to E$, map size $m = 2^l$.
Parties: $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ with delegate $\mathcal{P}_0$.
Inputs: $\mathcal{P}_i$ holds set $X_i \subset \{0,1\}^*$ for $0 \le i < n$.
Outputs: Delegate $\mathcal{P}_0$ receives the result of the protocol, other parties receive nothing.

1. (Delegate $\mathcal{P}_0$) Generates $\mathsf{sk} \xleftarrow{\$} \mathcal{K}_{\mathsf{AEAD}}$, runs $(\alpha, L, M) \leftarrow \mathsf{Delegate\text{-}Start}(X_0, X_0, \mathsf{sk}, \mathsf{AEAD.Enc})$ and publishes M, L to all parties.
2. (a) (Party $\mathcal{P}_1$) Initializes map R of size m. For all $w \in X_1$, computes $j \leftarrow \mathsf{index}(w)$ and sets $R_j \leftarrow \mathsf{DH.Reduce}(L, H_E(w), M_{j,0})$.
   (b) (Party $\mathcal{P}_i$ for $1 < i \le n-1$) For all $w \in X_i$, computes $j \leftarrow \mathsf{index}(w)$ and sets $R_j \leftarrow R_j + \mathsf{DH.Reduce}(L, H_E(w), M_{j,0})$.
3. (Party $\mathcal{P}_i$ for $1 \le i \le n-1$) For $0 \le j < m$, if R_j was not modified in step 2, $\mathcal{P}_i$ sets $R_j \leftarrow \{Q', S'\}$ for $Q', S' \xleftarrow{\$} E$.
4. (Party $\mathcal{P}_i$ for $1 \le i < n-1$) Sends R to $\mathcal{P}_{i+1}$.
5. (Last party $\mathcal{P}_{n-1}$) Runs $B \leftarrow \mathsf{Blind\text{-}Encrypt}(R, M)$ and sends B to delegate $\mathcal{P}_0$.
6. (Delegate $\mathcal{P}_0$) Retrieves $I \leftarrow \mathsf{Delegate\text{-}Finish}(\alpha, B)$.

Figure 12: The MPSI protocol, extending the PSI protocol.

Proof Sketch. $\mathcal{A}$ receives no output. Simulator $\mathcal{S}$ generates an incoming view for $\mathcal{A}$ that is computationally indistinguishable from a real execution. $\mathcal{S}$ generates M, L as in the Apple PSI proof [28]. $\mathcal{S}$ aborts if $\mathcal{A}$ aborts. If $j \in C$ such that $j > 1$, $\mathcal{S}$ also generates $R \xleftarrow{\$} E^{2 \times m}$. Even if $\mathcal{A}$ aborts after receiving R from an honest non-delegate, without knowledge of α, $\mathcal{A}$ cannot distinguish DH and non-DH tuples in the real R. □

Theorem 4. *A static computationally-bounded malicious adversary $\mathcal{A}$ that corrupts $\mathcal{P}_i$ for $i \in C \subset \{1, \ldots, n-1\}$ learns no information about X_0 in MPSI.*

Proof Sketch. The proof proceeds similarly to Theorem 3. Simulator $\mathcal{S}$ generates M, L as in the Apple PSI proof [28]. $\mathcal{S}$ aborts if $\mathcal{A}$ aborts. Even if $\mathcal{A}$ aborts before sending R or B, intractability of the DDH problem over E and semantic security of the AEAD cipher imply that $\mathcal{A}$ cannot computationally distinguish M, L in the simulation and a real execution. □

Theorem 5. *A static computationally-bounded malicious $\mathcal{P}_0$ learns no more information about $\{X_j : 1 \le j \le n-1\}$ than is revealed by the result of MPSI.*

Proof Sketch. Simulator $\mathcal{S}_0$ interacts with a corrupted $\mathcal{P}_0$. In MPSI, delegate $\mathcal{P}_0$ receives B and the protocol output but not R. If $\mathcal{P}_0$ aborts, $\mathcal{S}_0$ aborts. If $\mathcal{P}_0$ aborts before receiving B, it cannot compute the protocol result and does not learn

MPSIU-Sum

Public Parameters: EC group $E(\mathbb{F}_q)$ with generator G, hash function $H_E : \{0,1\}^* \to E$, map size $m = 2^l$.
Parties: $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ with delegate $\mathcal{P}_0$.
Inputs: $\mathcal{P}_i$ holds set $X_i \subset \{0,1\}^*$ for $0 \le i < n$ and $\mathcal{P}_0$ holds associated values V.
Outputs: Delegate $\mathcal{P}_0$ receives the result of the protocol, other parties receive nothing.

1. (All parties) Choose $\mathsf{sk}_i \xleftarrow{\$} \mathbb{F}_q$ and run $\mathsf{apk} \leftarrow \mathsf{Key\text{-}Aggregation}(\mathsf{sk}_0, \ldots, \mathsf{sk}_{n-1})$.
2. (Delegate $\mathcal{P}_0$) Runs $(\alpha, L, M) \leftarrow \mathsf{Delegate\text{-}Start}(X_0, V, \mathsf{apk}, \mathsf{EG.Enc})$ and publishes M, L to all parties.
3. (Party $\mathcal{P}_1$) Initializes map R of size m.
4. (Party $\mathcal{P}_i$ for $1 \le i \le n-1$) For all $w \in X_i$, sets $j \leftarrow \mathsf{index}(w)$ and $R_j \leftarrow \mathsf{DH.Reduce}(L, H_E(w), M_{j,0})$.
5. (a) (Party $\mathcal{P}_1$) For $0 \le j < m$, if R_j was not modified in step 4, sets $R_j \leftarrow \{Q', S'\}$ for $Q', S' \xleftarrow{\$} E$.
   (b) (Party $\mathcal{P}_i$ for $1 < i \le n-1$) For $0 \le j < m$, if R_j was not modified in step 4, sets $R_j \leftarrow \mathsf{DH.Reduce}(L, R_{j,0}, R_{j,1})$.
6. (Party $\mathcal{P}_i$ for $1 \le i < n-1$) Sends R to $\mathcal{P}_{i+1}$.
7. (Last party $\mathcal{P}_{n-1}$) Runs $B \leftarrow \mathsf{Blind\text{-}Encrypt}(R, M, \mathsf{apk})$ and sends B to delegate $\mathcal{P}_0$.
8. (Delegate $\mathcal{P}_0$) Runs $C_{\text{sum}} \leftarrow \mathsf{EG.Add}(\mathsf{Delegate\text{-}Finish}(\alpha, B))$ and broadcasts C_{sum} to all other parties.
9. (All parties) Run $\mathsf{Joint\text{-}Decryption}(C_{\text{sum}}, (\mathsf{sk}_0, \ldots, \mathsf{sk}_{n-1}))$ and delegate $\mathcal{P}_0$ learns the plaintext result.

Figure 13: The MPSIU-Sum protocol, extending MPSI.

any information about $\{X_j : 1 \le j \le n-1\}$. Otherwise, $\mathcal{S}_0$ learns α, M and real output of the protocol I_R. $\mathcal{S}_0$ constructs B such that the output matches in both real and simulated executions. As β, γ in DH.Reduce are drawn uniformly at random and the AEAD cipher is semantically secure, B is identically distributed in the two views regardless of whether decryption succeeds. □

6 MPSIU-Sum from MPSI

We now turn to constructing an MPSIU-Sum protocol. We begin by extending MPSI to MPSIU, which includes a union operation across $X_1, \ldots, X_{n-1}$. We then further extend the design to an MPSIU-Sum protocol, which includes a sum operation on elements in V. We provide an intuitive explanation of each step and a formalization of MPSIU-Sum, followed by proof of correctness, analysis of false negatives, and proof of security.

MPSI → MPSIU. Notice that both the MPSI and MPSIU problems require computation of $X_0 \bigcap S$ where

$$S \leftarrow \begin{cases} \bigcap_{i=1}^{n-1} X_i & \text{in MPSI} \\ \bigcup_{i=1}^{n-1} X_i & \text{in MPSIU.} \end{cases}$$

In MPSI, every party $\mathcal{P}_i$ for $i > 1$ either self-reduces a tuple or randomizes it. The self-reduction guarantees that tuples corresponding to elements of S—which are in each non-delegated party's set—remain DH tuples until $\mathcal{P}_{n-1}$ runs Blind-Encrypt. Randomization ensures that tuples not corresponding to elements of S become non-DH tuples eventually, except with negligible probability. As a consequence, $\mathcal{P}_0$ cannot decrypt AEAD ciphertexts corresponding to these non-DH tuples.

Adapting MPSI to MPSIU requires two modifications to this paradigm. First, to compute the intersection with union, every non-delegated party $\mathcal{P}_i$ holding $w \in X_i$ with $j = \mathsf{index}(w)$ sets $R_j \leftarrow \mathsf{DH.Reduce}(L, H_E(w), M_{j,0})$. Second, instead of randomizing tuples at indices where there is no possible match, a party self-reduces those tuples. These changes ensure that if $w \in X_i$ for any non-delegate party $\mathcal{P}_i$ and $w \in X_0$, R_j will form a DH tuple (except for hashmap collisions).

MPSIU → MPSIU-Sum. MPSI and the sketch of MPSIU allow delegate $\mathcal{P}_0$ to learn elements in the intersection $X_0 \bigcap S$, by decrypting AEAD ciphertexts containing elements. $\mathcal{P}_0$ can only decrypt a ciphertext if it is associated with a DH tuple.

We modify the MPSIU sketch in two ways to yield MPSIU-Sum. First, we replace $\mathcal{P}_0$'s associated data. Instead of encrypting elements in X_0 under a symmetric AEAD scheme (e.g., AES), $\mathcal{P}_0$ encrypts values in V under an additively homomorphic scheme (e.g., ElGamal). Second, we hide the protocol's intermediate output B from $\mathcal{P}_0$ by using an aggregated public key apk. This step ensures $\mathcal{P}_0$ can decrypt AEAD ciphertexts in B associated with DH tuples and can add the inner ElGamal ciphertexts, but it cannot decrypt those ciphertexts. $\mathcal{P}_{n-1}$ also adds 0 to each ciphertext so $\mathcal{P}_0$ cannot undo B's permutation. The parties then jointly decrypt the sum with output to $\mathcal{P}_0$.

Figure 8 provides an overview of MPSIU-Sum. We formalize the protocol in Figure 13. For brevity, we incorporate by reference the MPSI sub-protocols Delegate-Start in Figure 9, Blind-Encrypt in Figure 10 and Delegate-Finish in Figure 11. We discuss possible extensions for the protocol in Section 9.

6.1 Correctness

Theorem 6. *Assuming semi-honest participants, false positives are not possible in MPSIU-Sum.*

Proof. Let $U = \bigcup_{i=1}^{n-1} X_i$ and suppose, on the contrary, that $x \notin (X_0 \bigcap U)$ but $x \in D$ (in Delegate-Finish) with $\mathsf{index}(x) = j$. $\mathcal{P}_0$ was able to decrypt $B_{j,1}$ using key $\alpha \cdot B_{j,0}$, so $x \in X_0$ and $M_{j,0} = \alpha \cdot H_E(x)$. It follows that before Blind-Encrypt, $(L, R_{j,0}, R_{j,1})$ formed a DH tuple as $B_{j,1}$ was encrypted using $R_{j,1} = \alpha \cdot B_{j,0} = \alpha \cdot R_{j,0}$. Because $x \notin U$, $x \notin X_i$ for all $i > 0$. There are two possibilities:

- $y \in X_i$ for some $i > 0$ such that $y \neq x$ but $\mathsf{index}(y) = \mathsf{index}(x) = j$. $\mathcal{P}_i$ sets $R_j \leftarrow \mathsf{DH.Reduce}(L, H_E(y), M_{j,0})$. As $H_E(y) \neq H_E(x)$, $(L, H_E(y), M_{j,0})$ is not a DH tuple. R_j will be set to a random tuple from self-reduction.
- There exists no $y \in U$ such that $\mathsf{index}(y) = j$. $\mathcal{P}_1$ explicitly sets R_j to a random tuple, and the remaining parties $\mathcal{P}_i$ for $i > 1$ self-reduce R_j to another random tuple.

In either case, R_j is set to a random tuple by every $\mathcal{P}_i$ for $i > 0$. Therefore, $(L, R_{j,0}, R_{j,1})$ cannot be a DH tuple before Blind-Encrypt, yielding a contradiction. □

Theorem 7. *Assuming semi-honest participants, MPSIU-Sum false negatives are only caused by collisions in hashmap M.*

Proof. Let $U = \bigcup_{i=1}^{n-1} X_i, I = X_0 \bigcap U, x \in I$ with $\mathsf{index}(x) = j$. A false negative implies $\mathcal{P}_0$ could not decrypt $B_{j,1}$ with key $\alpha \cdot B_{j,0}$ in Delegate-Finish. It follows that $(L, R_{j,0}, R_{j,1})$ was not a DH tuple before Blind-Encrypt. Since $x \in U$, there must be $x \in X_i$ for some $i > 0$. It follows that $\mathcal{P}_i$ set $R_j \leftarrow (L, H_E(x), M_{j,0})$ during its turn. There are two possibilities:

- Party X_k, $i < k < n$ with $y \neq x$, $y \in X_k$ set $R_j \leftarrow (L, H_E(y), M_{j,0})$ and $\mathsf{index}(x) = \mathsf{index}(y) = j$, which implies a collision in M.
- $M_{j,0} \neq \alpha \cdot H_E(x)$. We know that $x \in I \implies x \in X_0$ so there must be $y \in X_0$ such that $y \neq x$ and $\mathsf{index}(x) = \mathsf{index}(y) = j$, which also implies a collision in M. □

6.2 False Negatives

The false negative analysis for MPSI (Section 5.2) applies to MPSIU-Sum. Notice that in MPSI, if there is a hashmap collision, the corresponding slot of M or R is corrupted. If $x \in I$, $y \notin I$, and $\mathsf{index}(x) = \mathsf{index}(y)$, the output will omit x.

MPSIU-Sum can, in some instances, recover from hashmap collisions. If a subsequent party has x in their set and no other value that hashes to $\mathsf{index}(x)$, the party will replace the corrupted value in R. As a consequence, the expected false negative rate for MPSI is an upper bound for MPSIU-Sum.

6.3 Security

We provide proofs of malicious security for MPSIU-Sum. For brevity, we combine proofs that non-delegate parties $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ learn no information about $X_1, \ldots, X_{n-1}$ or X_0.

Theorem 8. *A static computationally-bounded malicious adversary $\mathcal{A}$ that corrupts $\mathcal{P}_i$ for $i \in C \subset \{1, \ldots, n-1\}$ learns no information about $\{X_j : 1 \leq j \leq n-1, j \notin C\}$ in MPSIU-Sum.*

Theorem 9. *A static computationally-bounded malicious adversary $\mathcal{A}$ that corrupts $\mathcal{P}_i$ for $i \in C \subset \{1, \ldots, n-1\}$ learns no information about X_0 in MPSIU-Sum.*

Proof. We denote the view of $\mathcal{A}$ in the real world as $\mathsf{view}_{\mathcal{A}}^{\text{MPSIU-Sum}}$. Simulator $\mathcal{S}$ interacts with $\mathcal{A}$. As non-delegates receive no output, it suffices to show that $\mathcal{S}$ generates an incoming view that is computationally indistinguishable from $\mathsf{view}_{\mathcal{A}}^{\text{MPSIU-Sum}}$. In MPSIU-Sum, $\mathcal{P}_i$ for $i > 0$ receives L, M, pk_s for $s \neq i$ (during Key-Aggregation), and an ElGamal ciphertext C_{sum} (during Joint-Decryption). If $i > 1$ and $i \in C$, $\mathcal{A}$ also receives R. Define simulator $\mathcal{S}$ with inputs $\{X_i : i \in C\}$. If $\mathcal{A}$ aborts, $\mathcal{S}$ aborts. $\mathcal{A}$ may also corrupt inputs $\{X_i : i \in C\}$.

$\mathcal{S}$ chooses $X_0 \overset{\$}{\subset} \{0,1\}^*, V \overset{\$}{\subset} \mathbb{Z}^+, \{\mathsf{pk}_s\}_{s \neq i} \overset{\$}{\leftarrow} \mathcal{K}_{\mathsf{EG}}, z \leftarrow \mathbb{Z}^+$.
$\mathsf{apk} \leftarrow \mathsf{EG.AggKeyGen}(\mathsf{pk}_0, \ldots, \mathsf{pk}_{n-1})$.
$(\alpha, L, M) \leftarrow \mathsf{Delegate\text{-}Start}(X_0, V, \mathsf{apk}, \mathsf{EG.Enc})$.
$C_{\text{sum}} \leftarrow \mathsf{EG.Enc}(\mathsf{apk}, z)$.
$\mathcal{S}$ sends $L, M, \{\mathsf{pk}_s\}_{s \notin C}, C_{\text{sum}}$ to $\mathcal{A}$.
If $i > 1$ and $i \in C$, $\mathcal{S}$ also sends $R \overset{\$}{\leftarrow} E^{2 \times m}$.

Note that $\mathsf{view}_{\mathcal{A}}^{\text{MPSIU-Sum}}$ does not contain α or secret keys $\{\mathsf{sk}_s\}_{s \notin C}$ corresponding to $\{\mathsf{pk}_s\}_{s \notin C}$. As the Discrete Log Problem is computationally hard over E, $L = \alpha \cdot G$ is identically distributed in $\mathsf{view}_{\mathcal{A}}^{\text{MPSIU-Sum}}$ and in the simulator-generated view. Similarly, without knowledge of sk_s, pk_s is identically distributed. For any $w \in (\bigcup_{i \in C} X_i)$ such that $j = \mathsf{index}(w)$, $\mathcal{A}$ knows $L = \alpha \cdot G$ and can compute $H_E(w)$ but cannot distinguish the tuple $(L, H_E(w), M_{j,0})$ from $(L, H_E(w), \alpha \cdot H_E(w))$ due to the intractability of the DDH problem over E. Semantic security of the ElGamal cryptosystem implies that the ciphertexts $M_{j,1}$ and C_{sum} are also indistinguishable in the two views. Therefore, M is identically distributed in the two views. If $i > 1$ and $i \in C$, then without knowledge of α, $\mathcal{A}$ cannot distinguish between DDH and non-DH tuples in the real R. Because $\mathcal{S}$ samples the ideal R uniformly at random from $E^{2 \times m}$, $\mathcal{S} \overset{c}{\equiv} \mathsf{view}_{\mathcal{A}}^{\text{MPSIU-Sum}}$ as required. □

Theorem 10. *A static computationally-bounded malicious $\mathcal{P}_0$ learns no more information about $\{X_j : 1 \leq j \leq n-1\}$ than is revealed by the result of MPSIU-Sum.*

Proof. Simulator $\mathcal{S}_0$ interacts with a corrupted $\mathcal{P}_0$. In MPSIU-Sum, delegate $\mathcal{P}_0$ receives B, pk_s (during Key-Aggregation), d_s (during Joint-Decryption) for $s > 0$, and the protocol output. Define simulator $\mathcal{S}_0$ with input X_0. Note that $\mathcal{S}_0$ learns α, M, sk_0, $C_{\text{sum}} = (C_1, C_2)$, and the real output of the protocol I_R. If $\mathcal{P}_0$ aborts, $\mathcal{S}_0$ aborts. If $\mathcal{P}_0$ aborts before receiving B, it cannot compute the protocol result and thus does not learn any information about $\{X_j : 1 \leq j \leq n-1\}$. If $\mathcal{P}_0$ aborts after receiving B but before Joint-Decryption, it cannot compute the intersection sum but learns the cardinality of the intersection, which is also revealed by the result of MPSIU-Sum. $\mathcal{P}_0$ may also use a corrupted input X_0'. $\mathcal{S}_0$ programs the random oracle $H_E(\cdot)$ and extracts input provided by $\mathcal{P}_0$, as in the Apple PSI security proof [28]. We first show how $\mathcal{S}_0$ constructs $B, \{\mathsf{pk}_s\}_{s>0}, \{d_s\}_{s>0}$ such that the output matches in both real and simulated executions.

$\mathcal{S}_i$ chooses $\{\mathsf{sk}_s\}_{s \neq i} \overset{\$}{\leftarrow} \mathbb{F}_q$, and sets $\mathsf{pk}_s \leftarrow \mathsf{sk}_s \cdot G$, $\mathsf{apk} \leftarrow \mathsf{EG.AggKeyGen}(\mathsf{pk}_0, \ldots, \mathsf{pk}_{n-1})$.

Operation	MPSI		MPSIU-Sum			
	$\mathcal{P}_0$	$\mathcal{P}_i$ for $0 < i \leq n-1$	$\mathcal{P}_0$	$\mathcal{P}_1$	$\mathcal{P}_2, \ldots, \mathcal{P}_{n-2}$	$\mathcal{P}_{n-1}$
Key-Aggregation	–	–	1	1	1	1
Delegate-Start	$m+1$	–	$m(2c+1)+1$	–	–	–
Computation on R	–	$2(m+\lvert X_i\rvert)$	–	$2(m+\lvert X_1\rvert)$	$4m$	$4m$
Blind-Encrypt	–	–	–	–	–	$2mc$
Delegate-Finish	m	–	m	–	–	–
Joint-Decryption	–	–	c	c	c	c
Total	$2m+1$	$2(m+\lvert X_i\rvert)$	$(2m+1)(c+1)+1$	$2(m+\lvert X_1\rvert)+c+1$	$4m+c+1$	$(2m+1)(c+2)-1$

Table 1: Computation cost in elliptic curve point multiplications for MPSI, MPSIU-Sum, and sub-protocols.

	MPSI	MPSIU-Sum
delegate $\mathcal{P}_0$	$m+1$	$(n-1)+(m+1)(2c+1)$
$\mathcal{P}_1, \ldots, \mathcal{P}_{n-2}$	$2m$	$(n-1)+2m(c+1)$
$\mathcal{P}_{n-1}$	m	$(n-1)+m(2c+1)$

Table 2: Communication cost (egress) of MPSI and MPSIU-Sum in EC points. $m = |M| = |R| = |B|$ with c CRT moduli.

If $x \in I_R$ such that $\mathsf{index}(x) = j$, $\mathcal{S}_0$ chooses $z \xleftarrow{\$} E$ and sets $B_j \leftarrow \{z, \mathsf{AEAD.Enc}(\mathsf{KDF}(\alpha \cdot z), \mathsf{EG.AddZero}(\mathsf{apk}, M_{j,1}))\}$. Otherwise, $\mathcal{S}_0$ chooses $z_1, z_2 \xleftarrow{\$} E$ and sets $B_j \leftarrow \{z_1, \mathsf{AEAD.Enc}(\mathsf{KDF}(z_2), \mathsf{EG.AddZero}(\mathsf{apk}, M_{j,1}))\}$. $\mathcal{S}_i$ sets $d_s \leftarrow \mathsf{sk}_s \cdot C_1$ and sends $B, \{\mathsf{pk}_s\}_{s>0}, \{d_s\}_{s>0}$.

If $j \in \{\mathsf{index}(x) : x \in I_R\}$, $\mathcal{S}_0$ sets B_j such that $\mathcal{P}_0$ successfully decrypts the corresponding AEAD ciphertexts in Delegate-Finish. Otherwise, $\mathcal{S}_0$ sets B_j so decryption fails. $\mathcal{S}_0$ simulates Key-Aggregation using $\{\mathsf{pk}_s\}_{s>0}$ such that apk is constructed correctly. Using secret key shares $\{\mathsf{sk}_s\}_{s>0}$, $\mathcal{S}_0$ sets $\{d_s\}_{s>0}$ so that decryption of C_{sum} in Joint-Decryption succeeds.

We now show that $B, \{\mathsf{pk}_s\}_{s>0}, \{d_s\}_{s>0}$ are identically distributed in the real and simulated views. The argument for B is identical to that for Theorem 5, which we provide in the supplement [54]. Parties choose $\{\mathsf{sk}_s\}_{s>0}$ uniformly at random, as does $\mathcal{S}_0$, implying $\{d_s\}_{s>0}$ are also constructed identically by exponentiation of C_1. It follows that $\mathcal{S}_0 \overset{c}{\equiv} \mathsf{view}_0^{\mathsf{MPSIU\text{-}Sum}}$. □

7 Performance Evaluation

MPSI and MPSIU-Sum require total computation linear in $m = |M| = |R| = |B|$ and in the number of participants n. Similarly, the total communication is also linear in m and n. Both protocols require only a single round for every non-delegated party and the runtime of each party is independent of the number of parties n. The delegated party $\mathcal{P}_0$ uses a broadcast channel while every other participant communicates with only two parties (except during Key-Aggregation). Table 1 provides computation cost in elliptic curve (EC) point multiplications, which are generally the most expensive operations. Table 2 presents communication cost in EC points.

MPSIU-Sum is more computationally expensive than MPSI for two primary reasons. First, every party in MPSIU-Sum applies DH.Reduce to every hashmap slot, which costs 4 EC point multiplications each. Randomizing unmodified slots in MPSI, by contrast, requires only 2 EC point multiplications each. Second, MPSIU-Sum uses ElGamal encryption, which requires $2c$ EC point multiplications per encryption where c is the number of CRT moduli. The ciphertext expansion factor $F > 2c$ also increases communication cost in MPSIU-Sum.

Implementation. We implemented MPSI and MPSIU-Sum in Go (1.17.2) using the standard library implementation of NIST P-256, space-optimized bitmaps [55], a userspace CSPRNG [32], $c = 2$, and 33-byte compressed EC points. The implementation consists of approximately 2,000 lines of code and is available at https://github.com/citp/mps-operations.

Benchmarks. We ran benchmarks on a 128-core AMD EPYC 7742 @ 2.76 GHz with 1,024 GB RAM and parties participating locally in serial order. The private input set for each party consisted of random 12-byte strings. Table 3 presents the benchmark results with varying set and hashmap sizes.

Our motivating use case, estimating incidental collection, would involve running MPSIU-Sum about once annually for the IC's transparency report. Participants would have access to high-performance servers and connectivity with high bandwidth and low latency. Note also that the protocol easily parallelizes for each participant and streaming across participants.

Predicting performance for our use case is difficult, because IC and communications service inputs are not public. As very rough figures, based on IC transparency reports and service usage disclosures, IC input could include tens to hundreds of millions of items and communications service input could include hundreds of millions to billions of items. Even with very conservative assumptions—input sets with tens of billions of elements and a hashmap with over a trillion indices—our benchmarks show that MPSIU-Sum would remain practical.[16]

8 Optimizations

Concurrent Execution. For simplicity, we describe serial versions of MPSI and MPSIU-Sum. Both protocols could be adapted for concurrent computation with a star topology. In

[16] If performance were prohibitive, the parties could run MPSIU-Sum on a random subset of hashmap indices and extrapolate to the entire hashmap.

$\|X_0\| = \|V_0\|$	$\|X_i\|, i > 0$	$\|M\|$	MPSI					MPSIU-Sum				
			$\mathcal{P}_0$	$\mathcal{P}_1$	$\mathcal{P}_2, \ldots, \mathcal{P}_{n-2}$	$\mathcal{P}_{n-1}$	*FNR*	$\mathcal{P}_0$	$\mathcal{P}_1$	$\mathcal{P}_2, \ldots, \mathcal{P}_{n-2}$	$\mathcal{P}_{n-1}$	*FNR*
2^{20}	2^{20}	2^{24}	80	41	49	79	$\sim 10\%$	81	52	74	145	$\sim 8\%$
2^{20}	2^{20}	2^{25}	152	72	79	144	$\sim 5\%$	191	76	111	270	$\sim 4\%$
2^{21}	2^{21}	2^{25}	156	89	89	153	$\sim 10\%$	183	103	151	303	$\sim 8\%$
2^{21}	2^{21}	2^{26}	268	158	181	307	$\sim 5\%$	299	154	236	534	$\sim 4\%$
2^{22}	2^{22}	2^{26}	299	201	214	339	$\sim 10\%$	364	193	265	602	$\sim 8\%$

Table 3: Runtime (in seconds) and false negative rate (FNR) for MPSI and MPSIU-Sum with four parties.

MPSI, for example, parties could concurrently compute either a random tuple or a self-reduction (depending on their input) for each index in R. A non-delegate could operate as a hub and receive tuples for each index from all other non-delegates. It could then add the tuples for each index and run Blind-Encrypt on the resulting hashmap. This design would preserve the necessary property: the sum of DH tuples is a DH tuple and if at least one addend is a random non-DH tuple, the sum is also a random non-DH tuple. A similar construction for MPSIU-Sum would use the additive inverse property of EC point addition to replace the value at a hashmap index.

In this concurrent execution model, assuming co-located parties, the MPSI total runtime would be primarily determined by $\mathcal{P}_0$'s runtime. Similarly, the MPSIU-Sum total runtime would be primarily determined by $\mathcal{P}_{n-1}$'s runtime.

Faster Intersection Computation. In MPSI, $\mathcal{P}_0$ encrypts items in X_0 as associated data and then learns I by recovering plaintext. We use this construction as an intuitive step toward MPSIU-Sum. A more efficient MPSI or MPSIU construction could omit the associated data and permutation of B, such that $\mathcal{P}_0$ would learn I from which indices in B contain DH tuples.

Faster Cardinality Computation. Similarly, if $\mathcal{P}_0$ should only learn intersection cardinality, a more efficient MPSI-CA or MPSIU-CA construction could omit associated data such that $\mathcal{P}_0$ learns $|I|$ from the count of DH tuples in $\pi(B)$.

9 Extensions

In this section, we briefly describe optional extensions for MPSI and MPSIU-Sum. While we do not believe these extensions are essential for estimating incidental collection, we present them to enable additional use cases and because our constructions may be of independent research interest.

Additional Location Data. Multiple communications services may have location information about the person using an identifier, and that information can differ in recency and precision. It is possible to construct protocols that account for levels of location confidence, by decomposing the interactions between confidence levels into runs of MPSI and MPSIU-Sum.

MPSI-Sum. MPSI can be easily extended to an MPSI-Sum protocol, like MPSIU-Sum, by omitting the union operations.

Differential Privacy. The sum and cardinality output from MPSIU-Sum could reveal information about non-delegate sets, as discussed when formalizing the threat model (Section 3.4).

Our protocol allows for easy addition of calibrated noise during the sum computation, in order to achieve differential privacy [38, 77]. Any party could add noise to the ciphertext for a hashmap index, $\mathcal{P}_0$ could add noise to the aggregated homomorphic ciphertext during Delegate-Finish, and other parties could add noise to the aggregated ciphertext as an additional step in Joint-Decryption. $\mathcal{P}_0$ could provide a sensitivity for the sum operation under homomorphic encryption, or parties could independently estimate sensitivity.

MPSIU-Sum also enables adding calibrated noise to cardinality output. A party could increase cardinality with the Apple PSI protocol synthetic match method [28], replacing associated data with 0 under homomorphic encryption to preserve the sum computation.[17] A party could reduce cardinality, in expectation, by spoiling a hashmap index[18] or an input set item. Note that cardinality sensitivity for an identifier is 1.

A party could add noise to the sum and cardinality at the same time, by generating a synthetic match and setting associated data to noise for the sum under homomorphic encryption.

Arbitrary Functions of the Intersection. Variants of our protocols could compute arbitrary functions of an intersection or intersection with union, by substituting fully homomorphic encryption (FHE) for partially homomorphic encryption. For example, $\mathcal{P}_0$ could compute the sum of squares of values associated with the intersection if it encrypts associated data with FHE and squares intersection ciphertexts before summing.

Malicious Security. As discussed in the threat model formalization (Section 3.4), MPSI and MPSIU-Sum provide confidentiality against a malicious delegate $\mathcal{P}_0$ or a malicious subset of non-delegate parties $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$. The protocols do not, however, provide malicious security against a colluding delegate and non-delegate party. A non-delegate party could reveal its internal state to the delegate, allowing the delegate to learn the intermediate intersection computation across non-malicious preceding parties for each index.

[17] Parties could otherwise add 0 under homomorphic encryption to associated data, such that synthetic and real matches would be indistinguishable.

[18] Parties would have to modify R so ordinary and spoiled indices would be indistinguishable, such as by using the EC addition method in Section 8

A simple modification to the protocols would provide confidentiality against any malicious proper subset of parties. Intuitively, each party could randomly transform the blinding key α and the values that depend on α, preserving DH tuples while ensuring that no other party possesses the key necessary to test for DH tuples. Suppose that each non-delegate party $\mathcal{P}_i$ generates private randomizer α_i. When participating in the protocol, $\mathcal{P}_i$ updates L, $M_{j,0}$, and $R_{j,1}$ for $0 < j < m$ by multiplying them with α_i. $\mathcal{P}_i$ then uses the updated values and sends them to $\mathcal{P}_{i+1}$. After all parties have participated, $\mathcal{P}_{n-1}$ holds R where an index that has an intersection (in MPSI) or intersection with union (in MPSIU-Sum) contains a DH tuple with aggregated blinding key $\alpha \cdot \alpha_1 \cdot \ldots \cdot \alpha_{n-1}$. The parties jointly compute the aggregated key using secure multiplication with output to $\mathcal{P}_0$, and the protocol completes as normal.

Additional forms of malicious security are possible. After Delegate-Start, delegate $\mathcal{P}_0$ could provide a zero-knowledge (ZK) proof of knowledge of α such that $L = \alpha \cdot G$ using a Schnorr proof [69]. Similarly, after Blind-Encrypt in MPSIU-Sum, $\mathcal{P}_{n-1}$ could provide a ZK proof that ElGamal ciphertexts in B and M encrypt the same plaintext values [25]. After Delegate-Finish in MPSIU-Sum, $\mathcal{P}_0$ could prove it computed C_{sum} correctly with homomorphic addition of ciphertexts in B, by simply broadcasting B. In Joint-Decryption of ciphertext $\mathsf{ct} = (C_1, C_2)$, $\mathcal{P}_i$ sends $\mathsf{sk}_i \cdot C_1$. $\mathcal{P}_i$ could append a ZK proof of knowledge of sk_i [69]. Parties could verify these ZK proofs and abort the protocol in case of verification failure.

Output Delegation. In some use cases, it may be desirable for a party other than $\mathcal{P}_0$ to receive output. When estimating incidental collection, for example, it may be preferable to provide output to a different IC element or to an independent agency (e.g., the Privacy and Civil Liberties Oversight Board).

Modifying MPSIU-Sum to achieve this property would be straightforward. $\mathcal{P}_0$ could run Delegate-Finish and pass D to a new delegated party $\mathcal{E}$, regardless of whether that party provided input earlier. Recall that D contains homomorphic ciphertexts, which $\mathcal{E}$ can add to compute C_{sum}. $\mathcal{E}$ can then engage in Joint-Decryption with all parties $\mathcal{P}_0, \ldots, \mathcal{P}_{n-1}$.

Adapting MPSI for output delegation is also achievable. During Delegate-Start, $\mathcal{P}_0$ and $\mathcal{E}$ could use MPC for AEAD to encrypt X_0 under key sk held by $\mathcal{E}$. Similarly, during Delegate-Finish, $\mathcal{P}_0$ and $\mathcal{E}$ could use MPC for AEAD to decrypt tuples in B and reveal plaintexts to $\mathcal{E}$. MPC circuits for AES have been studied extensively and are now practical [37, 48].

If $\mathcal{E}$ only needs to learn intersection cardinality, as in the Section 8 optimization, delegation is simple. $\mathcal{E}$ can update blinding key α, as in the malicious security extension, and count DH tuples at the end of the protocol to learn cardinality.

10 Related Work

Our MPSI and MPSIU-Sum protocols build on a rich MPC literature. Private set intersection problems have received extensive scholarly attention, and we refer the reader to [66] for a valuable overview of constructions. We focus here on multiparty set operations, variants of two-party PSI that are particularly relevant to our protocols, and secure aggregation.

Multiparty Private Set Intersection. MPSI is the best studied multiparty private set operation. Prior work has constructed MPSI protocols from a diverse range of primitives, including threshold asymmetric encryption, oblivious pseudorandom functions, symmetric private information retrieval, and Bloom filters with homomorphic encryption [45, 46, 51, 58, 76]. Some MPSI constructions provide malicious security [27, 42, 45, 61, 78]. Recent work has explored post-quantum secure MPSI protocols and protocols for quantum computers [35]. Threshold variants of MPSI have also been proposed, only revealing the intersection if its cardinality is above or below a threshold [23, 24, 30].

We construct a new MPSI protocol to achieve the scale, functionality, and malicious security necessary for estimating incidental collection. Prior work has generally only evaluated MPSI protocols on set sizes up to 2^{20}. Kolesnikov et al. provide benchmarks for inputs sets larger than 2^{20}, but their MPSI protocol is secure only against a few corrupted semi-honest parties [51]. Bay et al. provide a more efficient MPSI protocol if set sizes are much smaller (up to 256) and the number of participants is higher (up to 50) [24]. Most MPSI protocols attempt to decrease communication at the cost of increasing computation. In our use case, the trade-off is reversed or neutral because participants are well resourced in both computational capacity and network connectivity.

Multiparty Private Set Intersection Cardinality. MPSI-CA protocols, which only reveal intersection cardinality, have been studied in the unbalanced case where the delegate's set is much smaller than that of other parties [56]. Our MPSI protocol can be adapted into a faster MPSI-CA protocol than past work, at the cost of increased communication (Section 8).

Multiparty Private Set Union. Kolesnikov et al. use oblivious transfer to achieve two-party PSU, and Garimella et al. generalize their technique to compute arbitrary functions of set intersection and union in the two-party setting yielding the most efficient PSU protocol [34, 41, 52]. Mohassel et al. provide a general protocol for three-party database joins [59].

In the multiparty setting, Shishido et al. propose a direct construction for MPSU over multisets, Wang et al. provide a generic protocol for mixed set operations, and Seo et al. provide the first constant-round MPSU protocol [70, 73, 75]. No implementations or concrete benchmarks are, however, provided for these MPSU constructions.

The problem of privately intersecting the delegate's set with the union of all other sets was first described by Kissner and Song as a hypothetical application for their private set operation protocols [49]. We formalize the MPSIU problem and present a more efficient protocol than past work.

Private Set Intersection Sum. PSI-Sum protocols output a sum of associated values at an intersection. Ion et al. describe a practical two-party protocol with semi-honest security [47].

Private Set Intersection with Associated Data. PSI-AD, alternately labeled PSI or PSI with data transfer, associates data with each intersection element. Recent work studies the two-party case with small delegate set sizes using FHE [33].

Secure Multiparty Aggregation. Secure aggregation protocols allow parties holding private values to aggregate them, without revealing more information than can be learned from the aggregate value [29]. These protocols are well studied for summation and have been constructed using perturbations, secret-sharing mechanisms, and homomorphic encryption, among other primitives [68]. Some protocols offer malicious security and differentially private aggregation [43]. Many aggregation protocols that do not use encryption are vulnerable to colluding malicious parties. We use a PHE-based aggregation method in MPSIU-Sum to defend against such collusion.

11 Conclusion

In this work, we proposed a secure multiparty computation approach for estimating the scale of incidental collection under Section 702 of FISA. We designed and evaluated a scalable multiparty private set intersection with union and sum protocol that could implement our proposed approach.

Demonstrating technical feasibility is, however, just the first step. The IC may be reluctant to participate in a protocol like ours without clear direction from Congress, out of concern that it might run afoul of the intricate U.S. legal framework for government surveillance. Communications services may also be hesitant because of perceived ambiguities in privacy law and risks of private litigation. We have heard both of these perspectives while discussing our proposal with intelligence professionals and technology sector practitioners.

Moving from technical feasibility to operational reality will likely require congressional leadership. And so, in closing, we offer a public policy recommendation: Congress should create a pilot program for estimating incidental collection. The program should require IC participation and offer limited liability protection for participating communications services. Pilot programs are common in legislation, including in the annual intelligence and defense authorization bills. A trial deployment is the logical next step toward finally estimating the scale of incidental collection under Section 702.

Acknowledgments

This project would not have been possible without the expertise, candor, and generosity of countless individuals. We thank the many intelligence professionals, congressional staff members, privacy law scholars, information security researchers, technology sector practitioners, civil society advocates, and national security journalists who informed this work. We also gratefully acknowledge the participants in a 2018 workshop on estimating Section 702 incidental collection, which was supported by the MacArthur Foundation. Seny Kamara, Anne Kohlbrenner, and Aleksandra Korolova provided valuable feedback on technical aspects of the project, and Michael Specter thoughtfully shepherded the publication. This work was supported by a cryptography research grant from Ripple. All views are solely our own.

References

[1] 18 U.S.C. § 2702.

[2] 50 U.S.C. § 1881a.

[3] H.R. 3773, 110th Cong. (as passed by House, Nov. 15, 2007).

[4] S. 2248, 110th Cong. (as reported by Sen. Select Comm. on Intelligence, Oct. 26, 2007 and as reported by Sen. Comm. on the Judiciary, Nov. 16, 2007).

[5] Letter from Senators Wyden and Udall to the Director of National Intelligence. https://www.wyden.senate.gov/imo/media/doc/2011-07-14%20Clapper%20FISA%20Letter.pdf, July 2011.

[6] Letter from the Office of the Director of National Intelligence to Senators Wyden and Udall. https://www.wyden.senate.gov/imo/media/doc/2011-07-28%20DNI%20Letter.pdf, July 2011.

[7] Letter from Senator Wyden et al. to the Director of National Intelligence. https://www.wyden.senate.gov/imo/media/doc/Letter%20to%20Clapper.pdf, July 2012.

[8] Letter from Senators Wyden and Udall to the Inspectors General of the Intelligence Community and the National Security Agency. https://www.wyden.senate.gov/imo/media/doc/2012-05-04%20WydenUdall%20Letter%20to%20NSA%20on%20Estimating%20Number%20of%20Americans%20Communications%20Monitored.pdf, May 2012.

[9] Letter from the Director of National Intelligence to Senators Wyden et al. https://www.wyden.senate.gov/imo/media/doc/08-24-2012%20Letter%20from%20Clapper%20regarding%20FISA%20Reauthorization.pdf, August 2012.

[10] Letter from the Inspector General of the Intelligence Community to Senators Wyden and Udall. https://www.wired.com/images_blogs/dangerroom/2012/06/IC-IG-Letter.pdf, June 2012.

[11] Letter from Civil Society Groups to the Director of National Intelligence. https://www.brennancenter.org/sites/default/files/analysis/Coalition_Letter_DNI_Clapper_102915.pdf, October 2015.

[12] Letter from the Office of the Director of National Intelligence to Civil Society Groups. https://www.brennancenter.org/our-work/research-reports/letter-office-director-national-intelligence, December 2015.

[13] Maximillian Schrems v Data Protection Commissioner. https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:62014CJ0362, September 2015.

[14] Letter from Civil Society Groups to the Director of National Intelligence. https://www.brennancenter.org/our-work/research-reports/letter-director-national-intelligence, January 2016.

[15] Letter from House Judiciary Committee Members to the Director of National Intelligence. https://irp.fas.org/congress/2016_cr/hjc-702.pdf, April 2016.

[16] Letter from House Judiciary Committee Members to the Director of National Intelligence. https://judiciary.house.gov/sites/democrats.judiciary.house.gov/files/documents/letter%20to%20director%20clapper%20(12.16.16).pdf, December 2016.

[17] Letter from Civil Society Groups to the Director of National Intelligence. https://www.aclu.org/letter/coalition-letter-director-national-intelligence-dan-coats-decision-abandon-efforts-estimate, June 2017.

[18] Letter from the Chair and Ranking Member of the House Judiciary Committee to the Director of National Intelligence. https://republicans-judiciary.house.gov/wp-content/uploads/2017/04/040717_Letter-to-DNI-Coats.pdf, April 2017.

[19] Letter from the Director of National Intelligence to the Chair and Ranking Member of the House Judiciary Committee. https://www.intelligence.senate.gov/sites/default/files/documents/FISA%20QFRs%202017-06-07.pdf, July 2017.

[20] Senate Intelligence Committee: Daniel Coats Nomination Hearing. https://www.intelligence.senate.gov/hearings/open-hearing-nomination-daniel-coats-be-director-national-intelligence, February 2017.

[21] Senate Intelligence Committee: Open Hearing on FISA Legislation. https://www.intelligence.senate.gov/hearings/open-hearing-fisa-legislation-0, June 2017.

[22] Data Protection Commissioner v Facebook Ireland Limited, Maximillian Schrems. https://www.europarl.europa.eu/doceo/document/TA-9-2021-0256_EN.html, July 2020.

[23] Saikrishna Badrinarayanan, Peihan Miao, Srinivasan Raghuraman, and Peter Rindal. Multi-party Threshold Private Set Intersection with Sublinear Communication. In *IACR International Conference on Public-Key Cryptography*, pages 349–379, 2021.

[24] Aslı Bay, Zekeriya Erkin, Jaap-Henk Hoepman, Simona Samardjiska, and Jelle Vos. Practical Multi-Party Private Set Intersection Protocols. *IEEE Transactions on Information Forensics and Security*, 17:1–15, 2021.

[25] Stephanie Bayer and Jens Groth. Efficient Zero-Knowledge Argument for Correctness of a Shuffle. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 263–280, 2012.

[26] Mihir Bellare. A Concrete-Security Analysis of the Apple PSI Protocol. Technical report, University of California, San Diego, July 2021. https://www.apple.com/child-safety/pdf/Alternative_Security_Proof_of_Apple_PSI_System_Mihir_Bellare.pdf.

[27] Aner Ben-Efraim, Olga Nissenbaum, Eran Omri, and Anat Paskin-Cherniavsky. Psimple: Practical Multiparty Maliciously-Secure Private Set Intersection. In *ACM ASIA Conference on Computer and Communications Security*, pages 1098–1112, 2022.

[28] Abhishek Bhowmick, Dan Boneh, Steve Myers, Kunal Talwar, and Karl Tarbe. The Apple PSI System. Technical report, Apple, Inc., July 2021. https://www.apple.com/child-safety/pdf/Apple_PSI_System_Security_Protocol_and_Analysis.pdf.

[29] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[30] Pedro Branco, Nico Döttling, and Sihang Pu. Multiparty Cardinality Testing for Threshold Private Intersection. In *IACR International Conference on Public-Key Cryptography*, pages 32–60, 2021.

[31] Michael Brown, Darrel Hankerson, Julio López, and Alfred Menezes. Software Implementation of the NIST Elliptic Curves over Prime Fields. In *Cryptographers' Track at the RSA Conference*, pages 250–265, 2001.

[32] Luke Champine. Go package frand. https://pkg.go.dev/lukechampine.com/frand.

[33] Kelong Cong, Radames Cruz Moreno, Mariana Botelho da Gama, Wei Dai, Ilia Iliashenko, Kim Laine, and Michael Rosenberg. Labeled PSI from Homomorphic Encryption with Reduced Computation and Communication. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1135–1150, 2021.

[34] Alex Davidson and Carlos Cid. An Efficient Toolkit for Computing Private Set Operations. In *Australasian Conference on Information Security and Privacy*, pages 261–278, 2017.

[35] Sumit Kumar Debnath, Tanmay Choudhury, Nibedita Kundu, and Kunal Dey. Post-Quantum Secure Multi-Party Private Set-Intersection in Star Network Topology. *Journal of Information Security and Applications*, 58:102731, 2021.

[36] Laura K. Donohue. Section 702 and the Collection of International Telephone and Internet Content. *Harvard Journal of Law and Public Policy*, 38, 2015.

[37] F. Betül Durak and Jorge Guajardo. Improving the Efficiency of AES Protocols in Multi-Party Computation. In *International Conference on Financial Cryptography and Data Security*, pages 229–248, 2021.

[38] Cynthia Dwork. Differential Privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12, 2006.

[39] Taher ElGamal. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[40] Armando Faz-Hernández, Sam Scott, Nick Sullivan, Riad S. Wahby, and Christopher A. Wood. Hashing to Elliptic Curves. Internet-Draft draft-irtf-cfrg-hash-to-curve-13, Internet Engineering Task Force, November 2021. Work in Progress. https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-hash-to-curve-13.

[41] Gayathri Garimella, Payman Mohassel, Mike Rosulek, Saeed Sadeghian, and Jaspal Singh. Private Set Operations from Oblivious Switching. In *IACR International Conference on Public-Key Cryptography*, pages 591–617, 2021.

[42] Satrajit Ghosh and Tobias Nilges. An Algebraic Approach to Maliciously Secure Private Set Intersection. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 154–185, 2019.

[43] Slawomir Goryczka and Li Xiong. A Comprehensive Comparison of Multiparty Secure Additions with Differential Privacy. *IEEE Transactions on Dependable and Secure Computing*, 14(5):463–477, 2015.

[44] Fabio Grandi. On the Analysis of Bloom filters. *Information Processing Letters*, 129:35–39, 2018.

[45] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. Scalable Multi-Party Private Set-Intersection. In *IACR International Workshop on Public Key Cryptography*, pages 175–203, 2017.

[46] Roi Inbar, Eran Omri, and Benny Pinkas. Efficient Scalable Multiparty Private Set-Intersection via Garbled Bloom Filters. In *International Conference on Security and Cryptography for Networks*, pages 235–252, 2018.

[47] Mihaela Ion, Ben Kreuter, Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, David Shanahan, and Moti Yung. Private Intersection-Sum Protocols with Applications to Attributing Aggregate Ad Conversions. In *IEEE European Symposium on Security and Privacy*, pages 370–389, 2020.

[48] Marcel Keller, Emmanuela Orsini, Dragos Rotaru, Peter Scholl, Eduardo Soria-Vazquez, and Srinivas Vivek. Faster Secure Multi-Party Computation of AES and DES using Lookup Tables. In *International Conference on Applied Cryptography and Network Security*, pages 229–249, 2017.

[49] Lea Kissner and Dawn Song. Privacy-Preserving Set Operations. In *Annual International Cryptology Conference*, pages 241–257, 2005.

[50] Neal Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[51] Vladimir Kolesnikov, Naor Matania, Benny Pinkas, Mike Rosulek, and Ni Trieu. Practical Multi-Party Private Set Intersection from Symmetric-key Techniques. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1257–1272, 2017.

[52] Vladimir Kolesnikov, Mike Rosulek, Ni Trieu, and Xiao Wang. Scalable Private Set Union from Symmetric-key Techniques. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 636–666, 2019.

[53] David Kris and J. Douglas Wilson. *National Security Investigations and Prosecutions*. Thomson Reuters, 3d edition, 2021.

[54] Anunay Kulshrestha and Jonathan Mayer. Supplementary Information. Technical report, Princeton University, 2022.

[55] Daniel Lemire, Owen Kaser, Nathan Kurz, Luca Deri, Chris O'Hara, François Saint-Jacques, and Gregory Ssi-Yan-Kai. Roaring Bitmaps: Implementation of An Optimized Software Library. *Software: Practice and Experience*, 48(4):867–895, 2018.

[56] Siyi Lv, Jinhui Ye, Sijie Yin, Xiaochun Cheng, Chen Feng, Xiaoyan Liu, Rui Li, Zhaohui Li, Zheli Liu, and Li Zhou. Unbalanced Private Set Intersection Cardinality Protocol with Low Communication Cost. *Future Generation Computer Systems*, 102:1054–1061, 2020.

[57] Jonathan Mayer, Patrick Mutchler, and John C. Mitchell. Evaluating the Privacy Properties of Telephone Metadata. *Proceedings of the National Academy of Sciences*, 113(20):5536–5541, 2016.

[58] Atsuko Miyaji and Shohei Nishida. A Scalable Multiparty Private Set Intersection. In *International Conference on Network and System Security*, pages 376–385, 2015.

[59] Payman Mohassel, Peter Rindal, and Mike Rosulek. Fast Database Joins and PSI for Secret Shared Data. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1271–1287, 2020.

[60] Moni Naor and Omer Reingold. Number-Theoretic Constructions of Efficient Pseudo-Random Functions. In *Annual Symposium on Foundations of Computer Science*, pages 458–467. IEEE, 1997.

[61] Ofri Nevo, Ni Trieu, and Avishay Yanai. Simple, Fast Malicious Multiparty Private Set Intersection. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1151–1165, 2021.

[62] Office of the Director of National Intelligence. *The FISA Amendments Act: Q&A*. April 2017. https://www.dni.gov/files/icotr/FISA%20Amendments%20Act%20QA%20for%20Publication.pdf.

[63] Office of Management and Budget. Statement of Administration Policy on S. 2248. https://www.justice.gov/archive/ll/docs/sap-on-s2248.pdf, December 2007.

[64] Office of the Director of National Intelligence. Release of Documents Related to the 2020 FISA Section 702 Certifications. https://www.intel.gov/ic-on-the-record-database/results/1057-release-of-documents-related-to-the-2020-fisa-section-702-certifications, April 2021.

[65] Office of the Director of National Intelligence. Annual Statistical Transparency Report. https://www.dni.gov/files/CLPT/documents/2022_ASTR_for_CY2020_FINAL.pdf, April 2022.

[66] Benny Pinkas, Thomas Schneider, and Michael Zohner. Scalable Private Set Intersection Based on OT Extension. *ACM Transactions on Privacy and Security*, 21(2), January 2018.

[67] United States Privacy and Civil Liberties Oversight Board. *Report on the Surveillance Program Operated Pursuant to Section 702 of the Foreign Intelligence Surveillance Act*. July 2014. https://documents.pclob.gov/prod/Documents/OversightReport/823399ae-92ea-447a-ab60-0da28b555437/702-Report-2.pdf.

[68] Thilina Ranbaduge, Dinusha Vatsalan, and Peter Christen. Secure Multi-party Summation Protocols: Are They Secure Enough Under Collusion? *Transactions on Data Privacy*, 13(1):25–60, 2020.

[69] Claus-Peter Schnorr. Efficient Identification and Signatures for Smart Cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252, 1989.

[70] Jae Hong Seo, Jung Hee Cheon, and Jonathan Katz. Constant-Round Multi-Party Private Set Union using Reversed Laurent Series. In *International Workshop on Public Key Cryptography*, pages 398–412, 2012.

[71] Hossein Shafagh, Anwar Hithnawi, Lukas Burkhalter, Pascal Fischli, and Simon Duquennoy. Secure Sharing of Partially Homomorphic Encrypted IoT Data. In *ACM Conference on Embedded Networked Sensor Systems*, pages 1–14, 2017.

[72] Daniel Shanks. Class Number, a Theory of Factorization, and Genera. In *Proceedings of Symposia in Pure Mathematics*, volume 20, pages 415–440, 1971.

[73] Katsunari Shishido and Atsuko Miyaji. Efficient and Quasi-Accurate Multiparty Private Set Union. In *IEEE International Conference on Smart Computing*, pages 309–314, 2018.

[74] Daniel J. Solove and Paul M. Schwartz. *Information Privacy Law*. Aspen, 7th edition, 2021.

[75] Wenli Wang, Shundong Li, Jiawei Dou, and Runmeng Du. Privacy-Preserving Mixed Set Operations. *Information Sciences*, 525:67–81, 2020.

[76] Zhusheng Wang, Karim Banawan, and Sennur Ulukus. Multi-Party Private Set Intersection: An Information-Theoretic Approach. *IEEE Journal on Selected Areas in Information Theory*, 2(1):366–379, 2021.

[77] Qiao Xue, Youwen Zhu, Jian Wang, and Xingxin Li. Distributed Set Intersection and Union with Local Differential Privacy. In *IEEE International Conference on Parallel and Distributed Systems*, pages 198–205, 2017.

[78] En Zhang, Feng-Hao Liu, Qiqi Lai, Ganggang Jin, and Yu Li. Efficient Multi-party Private Set Intersection against Malicious Adversaries. In *ACM SIGSAC Conference on Cloud Computing Security Workshop*, pages 93–104, 2019.

A Artifact Appendix

A.1 Abstract

Multiparty Private Set Operations allow parties to privately compute the intersection of sets held by them (MPSI) or the intersection of one set with the union of all others (MPSIU). A delegated party learns the result and no other information is revealed. If set elements are associated with values, the values associated with the intersection can also be privately aggregated (MPSI-Sum or MPSIU-Sum). The implementation is in Go and can be run in containers using Docker.

A.2 Checklist

- **Algorithm:** We present novel protocols for Multiparty Private Set Intersection (MPSI) and Intersection with Union (MPSIU). We also provide support for aggregation of values associated with the intersection (MPSI-Sum and MPSIU-Sum).
- **Compilation:** Requires Go version 1.18.
- **Data set:** The program generates random data to simulate the protocols in the user-specified `data` directory.
- **Metrics:** The program appends timing results to `bench.csv` in the user-specified `results` folder.
- **Output:** The program prints output to `stdout` and appends to `bench.csv`.
- **Experiments:** Please refer to the `README` provided. The program reads configuration from `config.yml`.
- **How much disk space required (approximately)?:** Disk space requirements are proportional to the number of parties and set sizes simulated (specified in `config.yml`).
- **How much time is needed to prepare workflow (approximately)?:** Both native and Docker builds take less than a minute on commodity hardware.
- **How much time is needed to complete experiments (approximately)?:** Please refer to Table 3 of the paper.
- **Publicly available (explicitly provide evolving version reference)?:** https://github.com/citp/mps-operations
- **Code licenses (if publicly available)?:** MIT license.
- **Archived (explicitly provide DOI or stable reference)?:** https://github.com/citp/mps-operations/releases/tag/usenix22

A.3 Description

A.3.1 How to access

Clone the repository from https://github.com/citp/mps-operations. The current version (as of publication) is https://github.com/citp/mps-operations/releases/tag/usenix22.

A.3.2 Software dependencies

Go (for native build) or Docker (for containerized build).

A.4 Installation

Native. Install Go (at least version 1.18) and run

```
go build -o mps_operations
./mps_operations
```

Docker. Install Docker (at least version 20.10.12) and run

```
docker build -t mps_operations .
docker run -it -rm -name mps_operations
mps_operations
```

A.5 Evaluation and expected results

Table 3 of the paper lists execution times using large input set sizes. The program was run on a modest server using 128 cores. On personal computers, it is expected to run for a longer time. The benchmarks from Table 3 can be reproduced by setting appropriate values for set sizes x_0 and x_i in `config.yml`. For build instructions, please refer to the `README`.

A.6 Experiment customization

Please refer to `config.yml`.

A.7 Notes

- The number of parties (n) in `config.yml` does not include the delegate.
- The upper bound on associated integers (l) in `config.yml` is ignored if the protocol is MPSI or MPSIU. l is only required for MPSI-Sum and MPSIU-Sum.

A.8 Version

Based on the LaTeX template for Artifact Evaluation V20220119.