UNIVERSITY OF CALIFORNIA,
IRVINE

Privacy-Preserving Consumer Rights Requests to Data Brokers

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Elina Marie Van Kempen

Dissertation Committee:
Professor Gene Tsudik, Chair
Professor Stanislaw Jarecki
Professor Habiba Farrukh

2026

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I want to thank my advisor, Gene Tsudik, for his invaluable insights into research and academic writing. I deeply appreciate his support throughout these intellectually engaging past five years, filled with fascinating projects.

I am very grateful to my committee members, Stanislaw Jarecki and Habiba Farrukh, for taking an interest in my research and offering their helpful advice. I owe particular thanks to Stanislaw Jarecki for his excellent instruction in cryptography.

I would also like to acknowledge my undergraduate professors for always being encouraging and for creating research opportunities for their students. Special thanks to Mohammad Islam for giving me my first look into research (in physics, on a topic quite different from this dissertation), and to Carolina Ilie, who never stops believing in her students and pushing them to do great things.

Thank you to my collaborators on all the various research projects: Seoyeon Hwang, Zane Karl, Giovanni Di Crescenzo, Ismat Jarin, Richard DeAmicis, Alfred Chen, Qifei Li, Giorgia Azzurra Marson, Claudio Soriente, Devriş İşler, Isita Bagayatkar, Chloe Georgiou, Pavel Frolikov, Mihir Raja, Pragya Jhunjhunwala, Rebekah Mercer, Angelo De Caro, Elli Androulaki, Kaoutar El Khiyaoui, and Ronald Petrlic. Working with all of you was a pleasure, and this dissertation is the product of that collaboration.

I am also thankful to my internship collaborators and coworkers for being so welcoming during those months and for the great times we shared in different cities. Sincere thanks to Claudio Soriente and Giorgia Azzurra Marson for guiding me through my very first PhD research project, and to Lara Kehoe Hoffman for warmly welcoming me to the CPPA and introducing me to the exciting work being done there.

I want to express my gratitude to my labmates throughout the years: Yoshimichi Nakatsuka, Sashidhar Jakkamsetti, Seoyeon Hwang, Andrew Searles, Youngil Kim, Renascence Tarafder Prapty, Pavel Frolikov, Isita Bagayatkar, Chloe Georgiou, and Paul Flammarion. The lab has always been a fun and interesting place to be. The numerous chats we had brightened my days, and I will also fondly remember our outings together.

Thank you to all my friends: at UCI, Joann Chen, Beverly Abadines Quon, Floranne Ellington, Nilab İsmailoğlu, and Ismat Jarin; and beyond, Lauren Printy, Trieu Le, Noémie Carlot, and Parul Gupta. Thank you all for staying close, and for the appreciated distractions from work.

Finally, I want to thank my family for always being supportive and raising me to be confident. Thank you to my parents, Geneviève Moebs and Erik van Kempen, my brothers, Nicolas and Mathieu, and my grandparents. Thanks to my Oma and Baba for the nice weekends spent together throughout my PhD, and for their unwavering pride in me. Lastly, thanks to my cat, Mavis, who is unconditionally loving and a joy to come home to.

This work was supported in part by funding from NSF Award SATC-1956393. Financial support was also provided through Teaching Assistant positions at UCI, and the Graduate Dean's Dissertation Fellowship.

Portion of Chapter 3 is a reprint as it appears in "Consumer Beware! Exploring Data Brokers' CCPA Compliance" [86] in the 47th IEEE Symposium on Security and Privacy, used with permission from the Institute of Electrical and Electronics Engineers. Portion of Chapter 5 is a reprint as it appears in "PIVA: Privacy-Preserving Identity Verification Methods for Accountless Users via Private List Intersection and Variants" [51] in the 29th European Symposium on Research in Computer Security, used with permission from Springer Nature. I thank my co-authors on these works for their permission to include this material in this dissertation, and the anonymous reviewers of both venues for their constructive feedback.

VITA

Elina Marie Van Kempen

EDUCATION

Doctor of Philosophy in Computer Science **2026**
University of California, Irvine *Irvine, California*

Master of Science in Computer Science **2024**
University of California, Irvine *Irvine, California*

Bachelor of Science in Physics and Applied Mathematics **2021**
State University of New York at Oswego *Oswego, New York*

PROFESSIONAL AND RESEARCH EXPERIENCE

Graduate Researcher **2021-2026**
University of California, Irvine *Irvine, California*

Applied Scientist Intern **2026**
Amazon *New York, New York*

Technologist Intern **2026**
California Privacy Protection Agency *California*

Research Intern, Decentralized Trust Group **2024**
IBM Research Zurich *Zurich, Switzerland*

Security Research Intern **2022**
NEC Laboratories Europe *Heidelberg, Germany*

TEACHING EXPERIENCE

Teaching Assistant (10 quarters) **2021–2026**
University of California, Irvine *Irvine, California*

REFEREED CONFERENCE PUBLICATIONS

Transaction Tracing for Efficient Audits in Privacy-Preserving Ledgers **2026**
European Symposium on Research in Computer Security (ESORICS)

Dead Men Tell No Tales: Assessing Post-Mortem Data Protection in GenAI Chatbots **2026**
IEEE Workshop on Technology and Consumer Protection (ConPro)

Consumer Beware! Exploring Data Brokers' CCPA Compliance **2026**
IEEE Symposium on Security and Privacy (S&P)

LISA: LIghtweight single-server Secure Aggregation with a public source of randomness **2025**
IEEE International Conference on Distributed Computing Systems (ICDCS)

Robust Secure Aggregation For Co-located IoT Devices With Corruption Localization **2024**
International Symposium on Cyber Security, Cryptology and Machine Learning (CSCML)

PIVA: **Privacy-Preserving Identity Verification Methods for Accountless Users via Private List Intersection and Variants** **2024**
European Symposium on Research in Computer Security (ESORICS)

Under Pressure: Effectiveness and Usability of the Apple Pencil as a Biometric Authentication Tool **2024**
Symposium on Usable Security and Privacy (USEC)

IN SUBMISSION

FedPoP: Federated Learning Meets Proof of Membership

Supporting Private and Efficient Consumer Requests to Data Brokers

Let My Data Go: Data Brokers' Compliance with Opt-Out & Deletion Requests

SSI-PSI: Authorized Private Set Intersection based on Self-Sovereign Identity

BOOKS

Classical Mechanics: Problems and Solutions **2022**
CRC Press

ABSTRACT OF THE DISSERTATION

Privacy-Preserving Consumer Rights Requests to Data Brokers

By

Elina Marie Van Kempen

Doctor of Philosophy in Computer Science

University of California, Irvine, 2026

Professor Gene Tsudik, Chair

Data brokers collect and sell people's personal information, often without their knowledge, to advertisers, employers, insurers, government agencies, and other third parties, creating risks that range from unwanted profiling to identity theft and stalking. Privacy regulations such as European Union's General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), and Brazil's Lei Geral de Proteção de Dados Pessoais (LGPD) are meant to provide consumers with agency over their data, granting them rights to access, delete, or opt out of the sale of their data. Effectively exercising these rights, however, is contingent on verifying that the person making the request is who they claim to be. This dissertation shows that identity verification, rather than protecting consumers, becomes a privacy threat.

We establish this systematically through two large-scale studies covering the entire California data broker registry. The first study submits data access requests to all 500+ registered brokers: over 40% never respond, pointing to a systemic failure to comply. The second study targets opt-out and deletion, the two most commonly exercised rights, and finds a similar pattern of non-compliance. More than one-fifth of brokers demand identity verification for opt-out requests, despite the CCPA explicitly prohibiting it. Across both studies, brokers consistently request a variety of personal information they do not already hold, for identity

verification purposes: consumer requests are a new source of data collection.

This dissertation addresses this threat directly and proposes two systems that make identity verification private by design. PIVA lets a consumer prove their identity to a broker using a private set intersection variant, private list intersection, guaranteeing that the broker learns no new personal information about the consumer through the request. PIVOT further reduces consumer burden, allowing a consumer to submit a single request that reaches every registered broker, through an intermediary that performs proxy re-encryption on the consumer's behalf and remains oblivious to the consumer's personal information. Both systems are implemented and benchmarked for efficiency, demonstrating that privacy-preserving identity verification is practical at scale.

Chapter 1

Introduction

More than 1,000 data brokers worldwide collect and sell people's personal information without their knowledge or consent [31]. Thus, millions of individuals' personal information is sold online: not only is this highly privacy invasive, but it also creates risks such as identity theft and stalking [34, 62, 75, 88]. While privacy laws such as the CCPA [17] and GDPR [37] give consumers rights to access, delete, or opt out of the sale of their data, exercising these rights is complicated by the fact that consumers are typically unaware which data brokers possess data about them. Accordingly, a few U.S. states have begun mandating that data brokers register publicly [18, 42, 80, 1], so that consumers are at least made aware of which companies may be collecting their information and whom to contact when exercising any right of access, deletion or opt-out.

However, since consumers do not know which specific brokers may have collected personal information about them, it is safest to assume that any registered broker could hold their data, creating a substantial burden if a consumer wishes to exercise their rights across the entire registry. In addition, brokers receiving a data request must verify a consumer's identity before honoring their request, to prevent unauthorized disclosures to malicious actors [17, 37].

The current "identity verification" approach, common to data brokers and other companies alike, is to request several pieces of personal information from the consumer and match them against existing database records [33, 72, 82, 12, 14]. This process itself can be privacy invasive.

The first half of this dissertation investigates the behavior and CCPA compliance of all California-registered data brokers. In the first study, I used my own personal data to submit access requests to data brokers. Over 40% of brokers failed to reply to the request, demonstrating widespread non-compliance with the law. Moreover, fewer than 5% of brokers reported having any personal information about me, yet I was required to provide my information to all of them during the identity verification process, highlighting the paradox of having to give up one's privacy in order to (attempt to) exercise one's rights. The second study focuses on the two most commonly exercised rights: deletion and opt-out [19]. Using synthetic identities, we again submitted requests to all registered brokers. Once again, many brokers failed to reply, and over 20% required identity verification when processing opt-out requests, in flagrant non-compliance with the CCPA [16, 17, 25].

As observed in both studies, privacy problems arise directly from the current methods of verifying consumer identity when processing a data request. Unlike businesses that a consumer has actually used, and thus knows that personal information was previously provided to obtain services, a consumer contacting a data broker must share their personal information blindly, without knowing whether the broker holds any data about them at all. This process is highly invasive, and, should the consumer encounter a malicious or negligent broker, the shared data could potentially be illegally reused for commercial purposes [86].

To remedy the situation, the second half of this dissertation proposes privacy-preserving methods for identity verification during data requests. PIVA introduces Private List Intersection and its variants, building on known Private Set Intersection techniques. With PIVA, a consumer and a broker can privately compute the intersection of their respective lists of

the consumer's personal information. The broker learns only information about the consumer that it already possessed, still being able to match this information to validate the consumer's identity.

While PIVA enables private identity verification, a consumer must nevertheless contact each broker individually. Several commercial services aim to relieve consumers of this burden by submitting requests on their behalf to a list of brokers, usually numbering in the hundreds, for a fee [76, 10, 5]. In addition, California recently became the first state to offer such a service to consumers for free [20, 27], although this service is naturally limited to California residents. In both cases, however, the consumer must place full trust in the intermediary, whether a commercial service or a government agency: data is shared directly with this intermediary, meaning it learns all provided consumer personal information. Intermediaries themselves become attractive targets for attackers.

To address this issue, we introduce PIVOT, a system that allows a consumer to submit requests to a large list of data brokers through a single communication with an intermediary. The intermediary handles all subsequent communication with the brokers, including identity verification, building on PIVA. Crucially, the intermediary is oblivious: aside from a consumer identifier, it learns no personal information about the consumer. It relies on proxy re-encryption to carry out Private List Intersection computations with each broker on the consumer's behalf.

1.1 Summary of Contributions

This dissertation makes the following contributions:

- An investigation of data broker behavior and compliance with the CCPA during data access requests, submitted to all 543 data brokers registered in California at the time of the study (2024).

- An investigation of data broker behavior during data deletion and opt-out requests, submitted to all 535 data brokers registered in California at the time of the study (2025).

- PIVA, a protocol allowing consumers to privately verify their identity with a data broker, or any other business with which they do not hold an account. The protocol is implemented and evaluated.

- PIVOT, a system allowing a consumer to send requests to all data brokers through a single communication with an oblivious intermediary. The system is implemented and evaluated.

All implementations are publicly available.

1.2 Statement of Collaborative Work

All work presented in this dissertation was collaborative. A summary of individual contributions follows.

Chapter 3, based on "Consumer Beware! Exploring Data Brokers' CCPA Compliance" [86], was my own idea, conceived after working on PIVA (Chapter 5). I designed the study and used my own personal data to carry it out. I was the main contributor throughout the study's design, execution, and the writing and submission of the paper. Isita Bagayatkar, Pavel Frolikov, and Chloe Georgiou all contributed to data collection and paper writing. Gene Tsudik provided guidance on the project and on writing the paper.

Chapter 4, based on the project "Let My Data Go: Data Brokers' Compliance with Opt-Out & Deletion Requests," is a follow-up to the project presented in Chapter 3. I had the pleasure of working with Pragya Jhunjhunwala and Mihir Raja, who each carried out the data collection phase for one of the two methods under my guidance. I helped troubleshoot

any issues they encountered and carried out the data analysis and paper writing. Gene Tsudik initiated the project and provided guidance throughout.

Chapter 5 is based on "PIVA: Privacy-Preserving Identity Verification Methods for Accountless Users via Private List Intersection and Variants" [51]. Everyone was involved in designing the protocols. Zane Karl implemented the proof of concept. Together with Seoyeon Hwang, I worked on formally defining the system and on the security proofs. Stanislaw Jarecki and Gene Tsudik provided insight on the cryptographic tools used. Stanislaw Jarecki designed the extension to malicious security. Gene Tsudik initiated the project.

Chapter 6 is based on the project "Supporting Private and Efficient Consumer Requests to Data Brokers." I proposed the project, after having worked on projects from Chapter 5 and Chapter 3. Gene Tsudik provided insight and guidance on the system design. I led the system design and implementation, as well as the paper writing. Pavel Frolikov and Chloe Georgiou helped design and formalize the threat models and system extensions, and contributed to the paper writing. Gene Tsudik provided guidance throughout.

1.3 Organization of the Dissertation

Chapter 2 provides background and related work relevant to all subsequent chapters, in particular a summary of the relevant regulations and current techniques for making data requests. Chapter 3 presents the first study, in which data access requests were submitted to all California-registered data brokers using my personal information. Chapter 4 follows up with an investigation of data brokers' responses to opt-out and deletion requests using synthetic data. Chapter 5 introduces PIVA, a privacy-preserving protocol for identity verification when making data requests. Chapter 6 proposes PIVOT, a system allowing consumers to make privacy-preserving data requests to many data brokers at once, via an oblivious intermediary. Finally, Chapter 7 concludes this dissertation.

Chapter 2

Background and Related Work

2.1 Background

2.1.1 Privacy Regulations

The General Data Protection Regulation (GDPR), enacted in 2016 by the European Union, established comprehensive privacy regulations and rights for European consumers [37], and inspired new privacy legislation internationally. In particular, the California Consumer Privacy Act (CCPA), voted on in 2018 and amended in 2020 by the California Privacy Rights Act (CPRA), grants California residents certain rights over the personal information collected about them by businesses [17]. These rights are commonly known as *the rights to know/access, opt-out of the sale of, or delete personal information collected by a business.* The Act is accompanied by the CCPA Regulations [16].

Personal information. CCPA defines personal information as information that:

> "identifies, relates to, describes, ..., or could reasonably be linked, directly or indirectly, with a particular consumer or household." (CCPA §1798.140(v))

This includes identifiers (names or usernames) and other consumer data, such as addresses, biometric information, geolocation information, and network activity. It excludes publicly available information, i.e., public government records and consumer information made public by the consumer or another person, assuming no restrictions placed by the consumer. CCPA further distinguishes *sensitive* personal information, which includes an individual's SSN, driver's license or passport number, genetic information, racial or ethnic information, and account credentials (CCPA §1798.140(ae)).

Consumer rights. CCPA grants consumers several rights over their personal information:

- **Access** (CCPA §1798.110): the right to know what personal information a business has collected about them.
- **Deletion** (CCPA §1798.105): the requesting consumer's personal information collected by a business must be erased.
- **Opt-out** of sale or sharing of personal information (CCPA §1798.120): although a business may retain consumer data in its databases, it must stop all sharing and selling of that data. Businesses are explicitly not allowed to verify consumer identity for opt-out requests (CCPA Regs. §7026(c)), i.e., the opt-out process is meant to avoid "unnecessary friction" [25].

Consumers exercise these rights by submitting a "Verifiable Consumer Request" (VCR) (CCPA §1798.140(ak)). Businesses must verify the identity of the requesting consumer, usually by matching several pieces of personally identifiable information (PII) provided by the consumer against previously collected data.

Identity verification. CCPA encourages identity verification via an existing password-protected account, however, it prohibits businesses from requiring a consumer to create an account solely to submit a VCR (CCPA §1798.130(a)(2)(A)). For non-account holders,

CCPA Regulations (CCPA Regs. §7062) require a business to verify a consumer's identity to a:

- "reasonable degree of certainty," when a user requests to know categories of personal information, by matching at least two "reliable" data points provided by the consumer with data points possessed by the business;
- "reasonably high degree of certainty," when a user requests to know specific pieces of personal information, by matching at least three "reliable" data points, along with a signed declaration under penalty of perjury.

CCPA states that a business may not use the PII collected during identity verification for any other purpose (CCPA §1798.130(a)(7)).

Response timelines. Once a business verifies the consumer's identity, it must confirm receipt of an access or deletion request within 10 business days (CCPA Regs. §7021(a)), and respond within 45 calendar days (extendable by another 45 days with notice to the consumer) (CCPA §1798.130(a)(2)(A)). For opt-out requests, a business must stop selling/sharing the consumer's personal information within 15 business days, and must wait 12 months before again requesting consent to share that consumer's information (CCPA §1798.135). CCPA Regulations further require businesses to explicitly confirm that a request was processed, e.g., for opt-out (CCPA Regs. §7025(b)):

> "...a business must provide a means by which the consumer can confirm that their request to opt-out of sale/sharing has been processed by the business. For example, the business may display on its website "Opt-Out Request Honored" in accordance with section 7025, subsection (b)(6), and display in the consumer's privacy settings through a toggle or radio button that the consumer has opted out of the sale/sharing of their personal information."

and, for deletion (CCPA Regs. §7023(b)):

> "In responding to a request to delete, a business shall inform the consumer whether it has complied with the consumer's request."

In addition, responses to access requests must be delivered by mail or electronically, "in a readily usable format that allows the consumer to transmit this information from one entity to another entity without hindrance" (CCPA §1798.130(a)(2)(A)).

Data Broker Registration law. Businesses must comply with CCPA if they satisfy one of the following (CCPA §1798.140(d)): (1) gross annual revenue over $25 million, (2) handling personal information of over 100,000 California residents, or (3) deriving more than 50% of revenue from the sale of California residents' personal information. The Data Broker Registration law [18] establishes further requirements specifically for data brokers, defined as:

> "a business that knowingly collects and sells to third parties the personal information of a consumer with whom the business does not have a direct relationship" (Data Broker Registration §1798.99.80(c)).

Unlike other businesses, which collect data about their own customers, data brokers collect data from individuals who have never used their services. Data brokers must register annually[1] with the California Privacy Protection Agency (CPPA) (Data Broker Registration §1798.99.82), currently for a $6,000 fee plus a 2.99% electronic payment surcharge (Data Broker Registration Regs. §7600). Data brokers that fail to register may be fined $200 per day of non-compliance, and must undergo regular third-party audits starting in 2028 (Data Broker Registration §1798.99.86(c)). The list of all registered data brokers is publicly available [19]. Aside from California, other states have introduced their own data broker registration legislation, e.g., Vermont [42], Oregon [1], and Texas [80]. Furthermore, data broker registration is proposed federally [85].

[1]The California data broker registry is available at https://cppa.ca.gov/data_broker_registry/

2.1.2 Submitting Consumer Requests in Practice

There are several ways for a consumer to exercise the rights above.

Direct contact. Consumers can contact a business or a data broker directly. Each entity must specify, in its privacy policy, at least two ways to submit a request [17]. This method is the most flexible in terms of which entities are contacted and how, though it is tedious and time-consuming, particularly when repeated across many data brokers.

Third-party services. Several for-profit services offer to submit requests on a consumer's behalf for a fee, e.g., DeleteMe [5], Aura [10], Incogni [76], and Optery [64]. In practice, these services require consumers to supply plaintext PII up front (full name, address, email, phone, date of birth, and sometimes information about relatives), which they then forward to individual data brokers on the consumer's behalf, where identity verification proceeds as usual. This method entails giving one's personal information to yet another 3rd party of potentially dubious trustworthiness, i.e., it may get hacked and/or engage in collecting or selling consumer data. Coverage and efficacy also vary: a recent study of 10 such services found major differences in which data brokers they cover, and that most fail to remove the majority of a user's records [47].

The DROP system. In 2023, California passed the Delete Act [27], creating the *Deletion Request and Opt-out Platform (DROP)* [20], a centralized, state-run data broker opt-out and deletion platform. DROP launched (for consumers) in January 2026: a California resident can submit a single, unified delete or opt-out request that applies to **all** registered data brokers, by providing several identifiers. "Basic" information includes name(s), date of birth, ZIP code(s), email address(es), and phone number(s); optional information includes Mobile Advertising IDs (MAIDs), Connected TV IDs, and Vehicle Identification Numbers (VINs). Each registered data broker must access DROP at least once every 45 days and act on any stored consumer identifiers, starting in August 2026 [20, 27]. DROP is of course

restricted to California residents, so consumers elsewhere must still rely on direct contact or third-party services.

The Delete Act states [27]:

> "...[T]he accessible deletion mechanism shall permit a consumer to securely submit information in one or more privacy-protecting ways..."

However, several important questions remain open: will PII be stored privately? How will identity verification be performed? And how will consumer PII be shared with data brokers? Proposed regulations indicate that lists of *hashed* identifiers will be stored and shared with data brokers [20], but no details are currently provided about any identity verification process or specific hash function(s) to be used. Without a *salt*, even hashed identifiers are easily guessable due to their low entropy [60]. Because DROP was only recently launched and data brokers are not yet required to use it, there has been no independent evaluation of its efficacy or usability.

2.2 Related Work

Throughout this thesis, we use the term **VCR** to refer to privacy rights requests broadly, and the term **consumer** to refer to an individual whose personal information is collected, regardless of the specific legal framework (CCPA, GDPR, etc.) in the context of which they occur. Prior work relevant to this thesis falls into two broad categories: empirical studies auditing compliance with privacy regulations (Section 2.2.1), and privacy-preserving techniques for identity verification (Section 2.2.2).

2.2.1 Compliance Auditing Studies

Most prior compliance work studied data-access VCRs submitted to popular websites or apps [59, 58, 6, 14, 89, 49, 55, 72, 12]. Some of these studies ensured that targeted entities

already had data about the researcher, by picking popular services, creating accounts where possible, and using them over time [14, 49, 55, 72]; this would be very complicated to do with data brokers, as they are not consumer-facing. Others focused on entities known to already hold researcher data [59, 58, 89], and examined vulnerabilities in the request process itself, including impersonation attacks [59, 58].

In 2023, Samarin et al. [72] submitted CCPA "right-to-know" (i.e., data access) requests to 109 Android app developers and reported their findings. The overall non-response rate was 19%. The same study showed alarming under-reporting of many types of collected information. Adhatarao et al. [6] attempted to submit VCRs using only an IP address for identification; all such attempts were unsuccessful.

Boniface et al. [12] and Urban et al. [83] evaluated VCR submission to third-party tracking services: the former focused on the security of the identity-verification and request process rather than outcomes, while the latter used stored cookies for identification and found that only 58% of tracking services replied at all.

Borem et al. [13] studied data-export procedures at 5 major service providers (Amazon, Facebook, Google, Spotify, and Uber). 33 participants with ≥ 2 years of use history each requested their data and found the resulting exports confusing and overwhelming, due to large amounts of information and little explanation of what was collected or why. Recently, He et al. [47] analyzed the pricing, coverage, and efficacy of 10 commercial PII removal services (see also Section 2.1.2), finding large differences in data broker coverage and generally poor removal rates.

Closest to the compliance studies in this thesis, Take et al. [78] tested CCPA/GDPR compliance of 20 people-search websites (a subclass of data brokers, e.g., Intellus, TruthFinder, and Whitepages), comparing VCR responses to paid user reports. Results showed that submitting requests was not straightforward, and the process varied even across requests made

to the same data broker, depending on who was submitting them. However, the study did not compare outcomes across the two legal frameworks, did not report response timelines, and its qualitative analysis of only 20 entities limits how far its findings generalize to the broader data broker landscape. Chapters 3 and 4 extend this line of work with the first large-scale, systematic studies of **all** California-registered data brokers, covering access requests and opt-out/deletion requests, respectively.

Table 2.1 summarizes how these two studies compare to prior compliance work, in terms of the number of entities to which VCRs were submitted, with the total number of entities involved in each study in parentheses.

2.2.2 Privacy-Preserving Identity Verification and PSI-Based Protocols

Privacy-invasive authentication in practice. Several of the compliance studies above show that the identity-verification process itself is typically privacy-invasive, requiring consumers to supply substantial PII, up to and including copies of a government-issued ID or a signed affidavit [58, 14, 12, 89, 49, 72, 55, 12, 82, 83, 14, 59]. This holds for popular websites, smartphone apps, and data brokers alike. Chapter 3 also finds that most data brokers have no data about the requesting consumer in the first place, meaning the verification process itself often becomes a new source of PII collection. In addition, many data brokers never responded to requests, in violation of the CCPA. All this increases the concerns of subsequent data misuse by untrustworthy and/or non-compliant data brokers.

Identity verification for accountless users. Verifying the identity of a consumer without an existing account is particularly hard, and supplying more PII does not always help: Adhatarao et al. [6] show that IP addresses, though PII, are insufficient for identity verification since they can be spoofed or shared by multiple users. Boniface et al. [12] propose a user-

generated cookie containing an email address and public key, used by the service provider to issue a challenge at verification time. Jordan et al. [52] propose VICEROY, where a private-public key-pair is generated at each web session by a trusted execution environment (TEE) on the consumer device; the public key is shared with the server and the private key later signs the consumer's requests. Neither approach works for entities the consumer has never interacted with before, such as data brokers.

To the best of our knowledge, prior to Chapter 5, there has been no prior work that investigated privacy-preserving identity verification specifically for consumers who have never previously interacted with the service provider or data broker in question.

Table 2.1: Comparison of this thesis's compliance studies to relevant prior results.

	# Entities	Law	Request type	Resp. rate	Consumer	Year
DBRs						
(Chapter 3)	454 (543)	CCPA	Access	57%	Real	2024–2025
(Chapter 4)	322, 358 (535)	CCPA	Delete, Opt-out	30–86%	Synthetic	2025–2026
People Search Websites						
Take et al. [78]	20	CCPA, GDPR	Access, Delete	82–100%	Real	2024
Tracking services						
Boniface et al. [12]	25 (30)	GDPR	–	–	–	2018–2019
Urban et al. [83]	36 (39)	GDPR	Access	58%	Real	2018
Common websites						
Martino et al. [58]	40	GDPR	Access	93%	Real	2021
Adhatarao et al. [6]	109 (124)	GDPR	Access	57%	Real (IP addresses)	2021
Bufalieri et al. [14]	317 (341)	GDPR	Access	71%	Pseudonymous/Real	2020
Martino et al. [59]	55	GDPR	Access	93%	Real	2019
Boniface et al. [12]	27 (50)	GDPR	–	–	–	2018–2019
Wong et al. [89]	229 (230)	GDPR	Portability	75%	Real	2018
Herrmann et al. [49]	119 (120)	GDPR	Access	77%	Synthetic	2016
Smartphone apps						
Samarin et al. [72]	109 (160)	CCPA	Access	81%	Synthetic	2023
Kroger et al. [55]	216–224 (225)	GDPR	Access	77–83%	Real	2015–2019
Herrmann et al. [49]	144 (150)	GDPR	Access	55%	Synthetic	2016

Chapter 3

Consumer Beware! Exploring Data Brokers' CCPA Compliance

Abstract

Data brokers collect and sell the personal information of millions of individuals, often without their knowledge or consent. The California Consumer Privacy Act (CCPA) grants consumers the legal right to request access to, or deletion of, their data. To facilitate these requests, California maintains an official registry of data brokers. However, the extent to which these entities comply with the law is unclear.

This chapter presents the first large-scale, systematic study of CCPA compliance of all 543 officially registered data brokers. Data access requests were manually submitted to each broker, followed by in-depth analyses of their responses (or lack thereof). Above 40% failed to respond at all, in an apparent violation of the CCPA. Data brokers that responded requested personal information as part of their identity verification process, including details they had

not previously collected. Paradoxically, this means that exercising one's privacy rights under CCPA introduces new privacy risks.

Our findings reveal rampant non-compliance and lack of standardization of the data access request process. These issues highlight an urgent need for stronger enforcement, clearer guidelines, and standardized, periodic compliance checks to enhance consumers' privacy protections and improve data broker accountability.

3.1 Introduction

Data brokers ($\mathcal{DBR}$s) operate largely hidden from public view: collecting, aggregating, and selling personal information (PI) of consumers without their knowledge or consent. These entities systematically harvest data from various sources: public records, online activities, social media profiles, and even other $\mathcal{DBR}$s. They routinely analyze this collected data to determine sensitive information such as purchasing behavior, financial status, and health conditions. Then, $\mathcal{DBR}$s monetize this information by selling it to various third parties, including companies, government agencies, and individuals. These sales are often completed without informing the consumer. One notorious and all-too-familiar class of $\mathcal{DBR}$s corresponds to so-called "people search" websites: online platforms that aggregate and market individual consumers' PI. Often, some PI is available for free, and additional information is available for gradually increasing prices. Usually, there is a prominent pitch to pay a fee in order to see detailed PI, with variable pricing schemes. Thus, any malicious actor can gain access to, and misuse, consumer PI to mount identity theft, fraud, or phishing attacks.

$\mathcal{DBR}$s operate without current relationships with consumers whose data they process, unlike traditional data aggregators (such as credit reporting agencies) with which consumers knowingly share their data to obtain presumably useful services. This fundamental difference results in consumers being unaware that their PI is being collected, analyzed, and made available for sale. Actual implications of such unmitigated access to consumers' PI remain

poorly understood by the general public.

To remedy the situation, several jurisdictions enacted data protection laws, e.g., GDPR [37], CCPA [17], and LGPD [45]. Though specific provisions differ, these laws aim to give individuals greater control over their PI. Notably, they grant consumers the rights to access, correct, and delete PI held by either businesses or other organizations. Under CCPA, consumers exercise these rights by making a *Verifiable Consumer Request (VCR)*:

> "a request made by a consumer ..., and that the business can verify, using commercially reasonable methods, ... to be the consumer about whom the business has collected personal information" (CCPA §1798.140(ak)).

The CPPA specifically targets $\mathcal{DBR}$s. Besides CCPA compliance, $\mathcal{DBR}$s must abide by the Data Broker Registration law, which requires each $\mathcal{DBR}$ to register annually with the CPPA. The resulting registry is made publicly available on the CPPA website[19].

Composing and submitting a VCR is burdensome for consumers because there is no standard process for doing so. Each $\mathcal{DBR}$ has its own method for consumers to submit a VCR, such as by filling out a form on the website, sending an email, making a phone call, or even having to go through a multi-step process. The necessary steps are typically (supposed to be) contained within a $\mathcal{DBR}$s' privacy policy.

Identity verification adds another layer of complexity. A $\mathcal{DBR}$ must verify the requesting consumer's identity before releasing the data in order to prevent data breaches. However, this verification process is not standardized and is taxing for the average consumer.

Accessing consumer data is especially challenging when interacting with $\mathcal{DBR}$s. Unlike other business entities, which users knowingly use, there is no way of knowing beforehand whether a $\mathcal{DBR}$ collected PI of a particular consumer. Thus, the consumer is forced to *blindly* make VCRs to a long list of $\mathcal{DBR}$s. The identity verification process is also somewhat Kafkaesque:

How can a consumer prove their identity to a $\mathcal{DBR}$ that may, or may *not*, have their PI?

Throughout this chapter, we distinguish between: (1) Personal Information (**PI**), i.e., any information a $\mathcal{DBR}$ may have about a consumer, and (2) Personally Identifiable Information (**PII**), which is a subset of PI that specifically identifies a consumer, e.g., name, address, or Social Security Number (SSN). This distinction is important for understanding privacy implications of CCPA enforcement. For example, the consumer's device model is PI, and not PII, because many consumers share the same device type; however, a driver's license number is PII since it uniquely identifies an individual. As shown later, this requirement to provide PII for identity verification creates an unintended privacy paradox when exercising one's CCPA rights.

3.1.1 Research Questions

Motivated by curiosity about the current state of CCPA compliance and the practical challenges consumers face when exercising their privacy rights, this work seeks to answer the following three research questions:

RQ1. What challenges are encountered in the VCR submission process, especially in the identity verification step?

VCR submission is not a streamlined process. $\mathcal{DBR}$s can simplify it by adopting user-friendly privacy policies and employing reasonable VCR submission methods. We measured the time I needed to find each $\mathcal{DBR}$'s contact information on their privacy policy, as well as the time needed to submit a VCR, along with the amount of PII needed for the $\mathcal{DBR}$ to verify my identity.

RQ2. To what extent do $\mathcal{DBR}$s comply with CCPA after VCR submission?

The goal is to discover what fraction of registered $\mathcal{DBR}$s comply, fully or in part, with CCPA requirements described in Section 3.2. After measuring VCR response rates and response timelines, we consider the factors that correlate with CCPA compliance.

RQ3. What kind of data do $\mathcal{DBR}$s collect and how is it shared with the consumer following a VCR?

After receiving the PI from $\mathcal{DBR}$s, we classify both the type of PI collected and its accuracy. We consider whether this data is sent securely to consumers, and whether it is provided in a standard and usable format (e.g., TXT, CSV, PDF, or JSON) as required by CCPA, rather than in some proprietary, obscure, or otherwise unparsable format. That PI must also be provided "...in a format that is easily understandable for the average consumer" (CCPA §1798.130 (a)(3)(B)(iii)).

3.1.2 Study Overview

This chapter reports on a comprehensive and systematic study of the $\mathcal{DBR}$ ecosystem. The study involved all 543 $\mathcal{DBR}$s duly registered in California, listed under the CPPA. We examined six key aspects of the VCR submission process: (1) consumer burden of composing VCRs, (2) variability in identity verification, (3) response time fluctuations, (4) quality of responses (i.e., content), (5) what actual PI is being collected, and (6) privacy issues, if any, that arise in the process of requesting PI.

To begin the study, I submitted VCRs to all 543 $\mathcal{DBR}$s, using my personal information, according to their individual guidelines, if those were available. We then measured their response timeline and analyzed common practices. We found that only 57% responded to VCRs, meaning that 43% are blatantly non-compliant.

We found that the majority of $\mathcal{DBR}$s did not have any PI about me. However, I still had to send PII to every $\mathcal{DBR}$ as part of their identity verification process.

Although some prior work studied CCPA and/or GDPR compliance, its focus was on entities with which the consumer has a direct relationship [59, 58, 6, 14, 12, 89, 49, 55, 72]. Furthermore, previous research on $\mathcal{DBR}$s examined only a small subset of $\mathcal{DBR}$s: people search websites [78]. Other types of $\mathcal{DBR}$s have not been investigated. A previous study evaluated the usability of data access and removal of 20 such websites [78], offering a rather narrow view of the $\mathcal{DBR}$ ecosystem.

3.1.3 Organization

Background and related work are present in Chapter 2. Additional background on data brokers' location is given in Section 3.2. The methodology is described in Section 3.3, including ethical considerations in 3.3.2. Section 3.4 follows with results. Finally, privacy issues and study outcomes are described in Section 3.5.

3.2 Background

Background on CCPA legislation is given in Chapter 2. In this section, we summarize the business location of data brokers registered in 2024.

3.2.1 $\mathcal{DBR}$s Location

95% $\mathcal{DBR}$s are registered to an address within the United States, with only 28 $\mathcal{DBR}$s registered abroad. Table 3.1 specifies the geographical distribution of $\mathcal{DBR}$s to which VCRs could be submitted, by country.

Within the United States, $\mathcal{DBR}$s are registered in 40 states. A large number are registered in three states: California, New York, and Florida, with 103, 74, and 46 registered $\mathcal{DBR}$s, respectively. Figure 3.1 shows a map of all $\mathcal{DBR}$s in the United States. Since no $\mathcal{DBR}$s are registered in Alaska or Hawaii, these states are not shown on the map.

Table 3.1: $\mathcal{DBR}$s by country of registration.

Country	$\mathcal{DBR}$ **Count**
United States	515
United Kingdom	10
Canada	5
Germany, Denmark	2 each
Malaysia, India, Poland, Sweden, Israel, New Zealand, Namibia, Italy, United Arab Emirates	1 each
Total	543

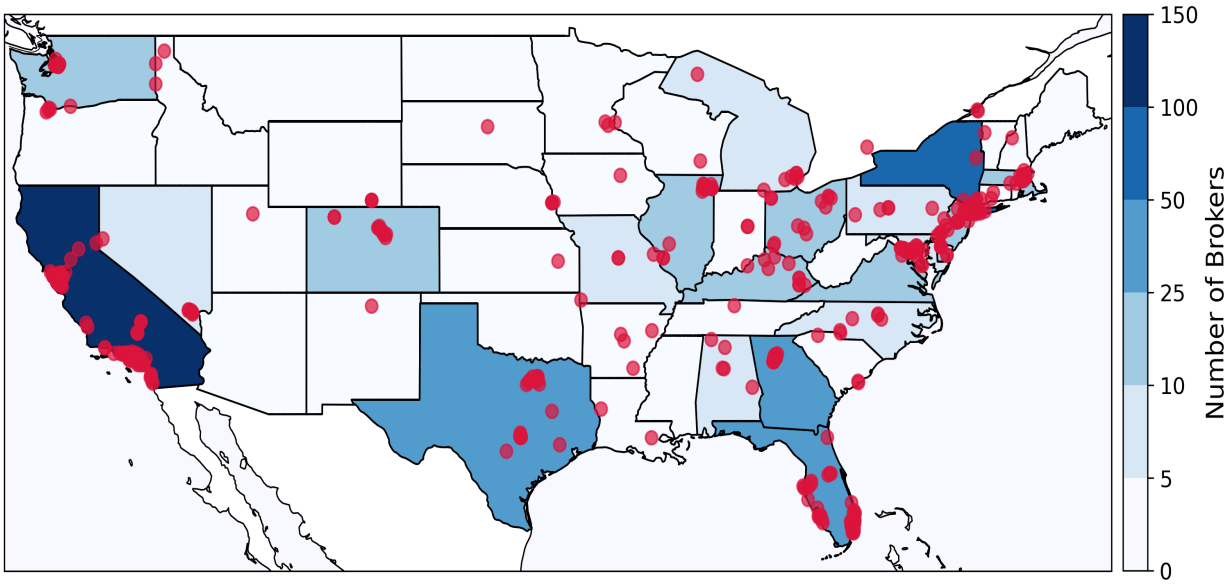


Figure 3.1: Map of US $\mathcal{DBR}$s.

3.3 Methodology

Recall that our goal is to assess $\mathcal{DBR}$s' CCPA compliance. To this end, we submitted data access VCRs[1] to all registered $\mathcal{DBR}$s. We chose not to submit data deletion or opt-out requests, since by doing so we would learn less information about $\mathcal{DBR}$s' data collection and response processes. In other words, a $\mathcal{DBR}$ could simply reply that relevant PI found in their database will be deleted, whether or not such data actually exists, with no proof of deletion, or similarly claim that they would stop selling any data whether they have this

[1]From this point forward, VCR refers specifically to data access requests.

information or not. Submitting a VCR guarantees a database lookup by the (compliant) $\mathcal{DBR}$, which must then reply with collected PI or with a statement that it has none.

3.3.1 Study Design

We investigated all 543 registered $\mathcal{DBR}$s in the California Data Broker Registry during 2024 [19] and submitted VCRs to exercise our "right to know/access". I myself, California resident since 2021, submitted all VCRs to ensure consistency and privacy.

Submitting VCRs. Each $\mathcal{DBR}$ in the registry provided a URL to their website's homepage. After finding a privacy policy, I located each $\mathcal{DBR}$'s instructions for submitting VCRs. I used each $\mathcal{DBR}$'s preferred VCR submission method to provide $\mathcal{DBR}$s with the best circumstances and maximize the likelihood of getting a response. In situations where the $\mathcal{DBR}$-provided forms were non-functional, I used an alternative email contact method listed in the privacy policy. A standard (consistent) email template was used across all emailed VCRs (Figure 3.2).

In a few cases, when neither a form-based nor an email-based contact method worked, the only option was a phone call. Although CCPA requires $\mathcal{DBR}$s to provide at least a toll-free telephone number for submitting VCRs (CCPA §1798.130(a)(1)(A)), doing it by phone is not optimal since it leaves no proof or record of submission. Additionally, callbacks are easy to miss. Whenever a phone call led to voicemail, I left a message stating that I wanted to submit a VCR, along with my phone number for $\mathcal{DBR}$s to call back.

To accurately track all VCRs and responses, a dedicated email address was created solely for the VCR submission purpose. If $\mathcal{DBR}$s requested alternative email addresses (e.g., a business email), I provided them.

I duly provided all PII requested by $\mathcal{DBR}$s for identity verification purposes. If a copy of a

Subject: Request for Information Under CCPA "Right to Know/Access"

To whom it may concern,

I am writing to you in order to exercise my rights under the California Consumer Privacy Act (CCPA) "right to know."

Please kindly provide the following information pertaining to my personal data:

- The categories of personal information collected **and** a copy of all specific pieces of information collected about me.
- The categories of sources of my personal information.
- The purposes for collecting my personal information.
- The categories of third parties with whom my personal information is shared or sold.
- The categories of my personal information that was sold or shared with third parties.

Please let me know if you need any further information from me.

Thank you for your prompt reply.

Sincerely,
<Name>

Figure 3.2: Template of the access request email sent to data brokers to exercise the CCPA "right to know."

government-issued ID was needed, I sent a redacted version unless a full copy was specifically requested.

Metrics. We measured VCR submission time for each $\mathcal{DBR}$: the timer started when I landed on the $\mathcal{DBR}$'s homepage and stopped when a form was submitted, an email was sent, or a phone call ended. Thus, we measured the time needed to: (1) locate the privacy policy on the $\mathcal{DBR}$'s website, (2) identify the VCR submission method within that policy, and (3) perform the actual VCR submission. The data collection period lasted roughly 7 months total, from late 2024 to early 2025.

We recorded PII elements requested by each $\mathcal{DBR}$ for VCR submission and identity verification. To evaluate CCPA compliance, we tracked the response rate, timeline of responses, and the content of responses. We recorded the dates when: (1) the original request was sent, (2) the request receipt acknowledgment was received, and (3) the response (if any)

was received. If a $\mathcal{DBR}$ requested additional PII, we recorded the time I needed to provide this information. In order to precisely and fairly assess response timeline, we exclude time taken to submit the requested additional PII from our analysis. Thus, when a $\mathcal{DBR}$ requests additional information, the $\mathcal{DBR}$ is not penalized for the time I took to reply to the $\mathcal{DBR}$.

Exclusions (Data Cleaning). 89 out of 543 registered $\mathcal{DBR}$s were excluded from the full study:

Duplicates. 50 corresponded to duplicate entries, i.e., different registry entries leading to the same website, email address, or form. We only submitted one request for each duplicate registration.

Noncompliant. 8 $\mathcal{DBR}$s' websites were completely inaccessible during the study period. 7 $\mathcal{DBR}$s lacked any privacy policy or contact information to submit VCRs. 11 $\mathcal{DBR}$s for which email-based VCRs could not be delivered. 2 $\mathcal{DBR}$s only provided phone contacts, and there was no way to leave a voicemail; in each case, I hung up after 5 minutes of phone silence. Finally, 1 $\mathcal{DBR}$ required notarization, although CCPA regulations (Tit. 11 Div. 6 §7060(e)) [16] state that: "...a business may not require a consumer to provide a notarized affidavit to verify their identity unless the business compensates the consumer for the cost of notarization" [16].

Lack of Security. 2 $\mathcal{DBR}$s with non-HTTPS websites and forms were excluded for security and privacy reasons.

Unable to complete VCR. 4 $\mathcal{DBR}$s required a website cookie ID, which was not present on my

device(s). 3 $\mathcal{DBR}$s requested information that I did not have, e.g., organizational position, website URL, or medical professional status.

"No longer a $\mathcal{DBR}$" (but registered anyway). 1 $\mathcal{DBR}$ announced on its main web page that it had exited consumer data sales.

After accounting for all exclusions, the analysis focused on the remaining 454 $\mathcal{DBR}$s, representing approximately 84% of the total.

3.3.2 Ethical considerations

Our Institutional Review Board officially declared that this study constitutes non-human subject research. I (using my own identity to compose VCRs) did so voluntarily, motivated primarily by the research goals. To preserve privacy, only I had access to $\mathcal{DBR}$ responses. I could choose to withdraw from the study at any time and not submit VCRs to certain websites if I felt uncomfortable. In particular, as mentioned above, I chose not to make VCRs to $\mathcal{DBR}$s that had non-HTTPS websites out of concern for privacy.

Also, recall that we observed that some $\mathcal{DBR}$s are registered more than once, i.e., multiple $\mathcal{DBR}$s' registrations led to the same VCR submission form or email. We avoided contacting them multiple times to be respectful of their resources. If we accidentally contacted a $\mathcal{DBR}$ twice, we only considered the first VCR for all further analysis.

Finally, besides the scientific goal of this study, I had genuine and legitimate interest in learning what PI was collected by $\mathcal{DBR}$s.

3.4 Results

We now present quantitative results addressing **RQ1**, **RQ2**, and **RQ3**. Note that RQ1 is broken down into Sections 3.4.1 and 3.4.2.

3.4.1 RQ1: Submitting VCRs

As detailed in Section 3.3.1, we submitted VCRs by form, email, or phone call, as outlined in the $\mathcal{DBR}$s' privacy policy.

VCR Submission Methods. Figure 3.3 depicts various submission methods encountered and their proportions. Since there is no CCPA-standardized method to submit VCRs, each $\mathcal{DBR}$ chooses what it prefers, which complicates VCR submissions for consumers. I managed to submit most VCRs (92.5%) directly by email or by form.

Forms, denoted form-VCRs, were the most common method. While seemingly straightforward (just complete and submit), forms presented their own challenges mainly because they are not standardized, with each $\mathcal{DBR}$ using a different interface and requiring different information. The second common VCR method was email, denoted email-VCR.

We note that $\mathcal{DBR}$s can purchase CCPA compliance services from privacy-as-a-service companies. OneTrust[63] was the most popular service, used by 56 $\mathcal{DBR}$s: about 25% of form-VCRs were "powered by OneTrust". Curiously, despite using OneTrust, form fields varied quite a bit among the 56 $\mathcal{DBR}$s' form-VCRs.

For 7 $\mathcal{DBR}$s, VCRs had to be submitted by phone, denoted phone-VCRs. This corresponds to cases where $\mathcal{DBR}$s only posted their physical address and phone number, leaving no way to contact them electronically.

27 submissions involved multiple steps. Upon submitting email-VCRs to 8 $\mathcal{DBR}$s, I received

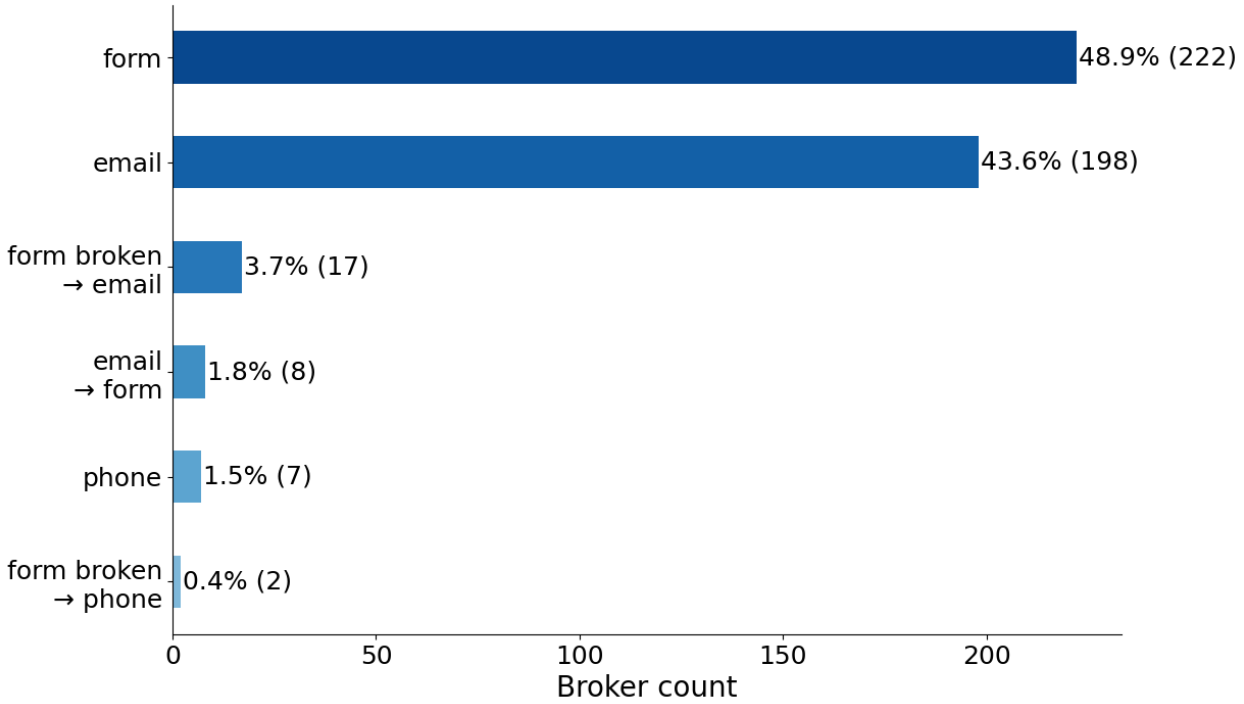


Figure 3.3: VCR submission methods.

links to resubmit through an online form. This occurred either because the $\mathcal{DBR}$ preferred email-initiated requests or because the form was not found in the $\mathcal{DBR}$'s privacy policy. Notably, certain $\mathcal{DBR}$s combine their "request to know" and "request to opt-out/delete" forms into one. This practice leads to confusion because the combined forms are frequently labeled only as "request to opt-out/delete". We believe that most consumers would not open a form titled "delete my data" when attempting to access their data.

19 forms could not be submitted, primarily due to broken links in the privacy policy. When such forms were encountered, I emailed or called the $\mathcal{DBR}$. Some forms failed to submit due to an "invalid CAPTCHA", even though no CAPTCHA was displayed on the webpage. In another irritating case, the form required a frequent shopper ID (confirmed to be a string of digits by the $\mathcal{DBR}$), yet the field only accepted email address formats.

Finally, in a particularly time-consuming request case, I had to call a $\mathcal{DBR}$ after being unable to submit the request form, since no email contact was specified in that $\mathcal{DBR}$'s CCPA

privacy policy. After leaving my contact information by voicemail, I received an email from the $\mathcal{DBR}$ asking to fill in a form (the link to which was broken) or to provide additional information by email. Despite replying with the required details and reporting on the form's malfunction, no further response was received.

VCR Submission Time. In total, 9 hours and 57 minutes were spent submitting VCRs to all 454 $\mathcal{DBR}$s. This corresponds to an average of 79 seconds to find the VCR method and to submit a VCR. Figure 3.4 shows the time to submit email-VCRs and form-VCRs. Phone and multi-step procedures are excluded from the Figure.

It took, on average, longer to submit form-VCRs than email-VCRs, with averages of 82 and 66 seconds, respectively. Shapiro-Wilk tests reject the null hypothesis that email-VCR and form-VCR submission times are normally distributed (p-value < 0.001), and a permutation test suggests a statistically significant difference in submission times (p-value < 0.005). Note that, while forms were filled out manually, emails were copy-pasted from a template. Submitting OneTrust forms took roughly as long as other forms.

Submitting phone-VCRs was quite time-consuming. An average phone call lasted 4 minutes and 37 seconds. It typically included talking with a company representative or leaving a voice message if no one was available. We were surprised by phone representatives being totally unaware of CCPA and confused by the nature of our requests. This was troubling since we made sure to use the phone number explicitly specified by $\mathcal{DBR}$s in their privacy policy.

Key Takeaways - RQ1 Part 1. Submitting VCRs is a complex process. While email-VCRs are one of the most commonly used methods, their open-ended nature places a burden on consumers, who must properly word their emails to ensure that $\mathcal{DBR}$s do not simply

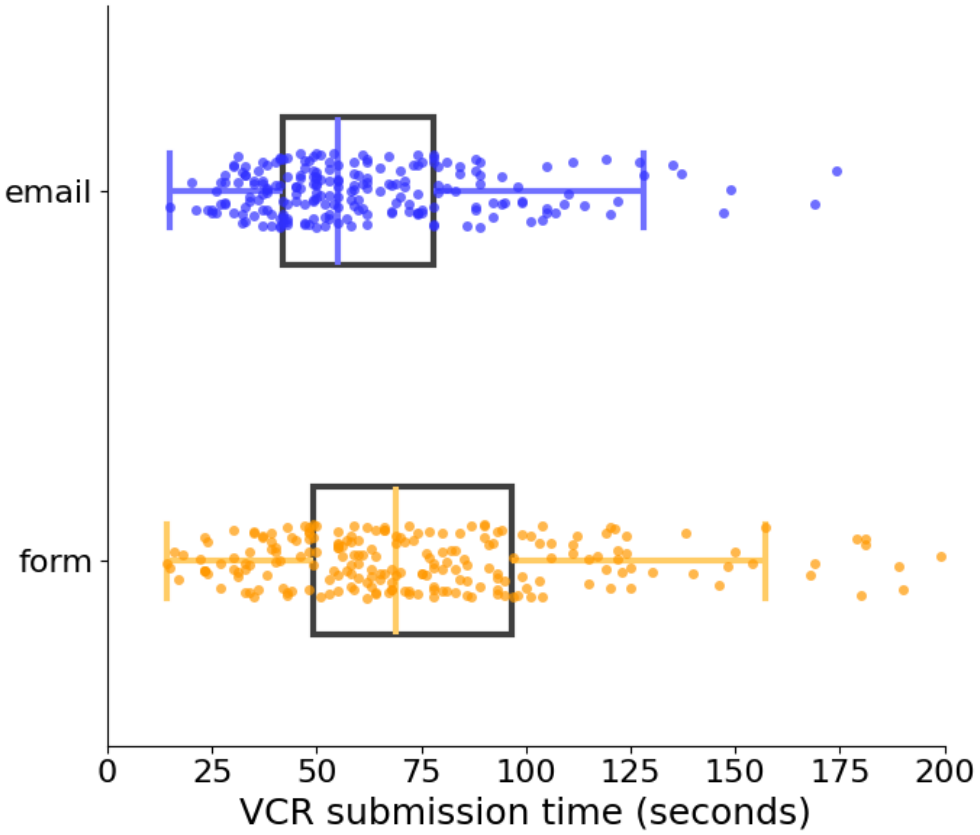


Figure 3.4: Time needed to submit VCRs.

dismiss their requests, and verify that the emailed VCR does not bounce. In comparison, form-VCRs are more straightforward, though more time-consuming. Multi-step processes incur an unjustified burden since consumers must keep track of (and complete) several steps, which are usually due to $\mathcal{DBR}$s' inadequate privacy policies or outdated contact information. Finally, phone-VCRs are plain unhelpful. Even when the calls were answered, employees demonstrated an alarming lack of understanding about the nature of the inquiry.

3.4.2 RQ1: Identity Verification Process

Recall that $\mathcal{DBR}$s have to verify the consumer identity via VCRs before granting them access to their PI. To this end, they ask the consumer to provide a set of PII, which is presumably matched with previously collected PII.

Type of PII Requested We first categorize different types of PII requested by $\mathcal{DBR}$s to perform identity verification. Table 3.2 shows, for each PII category, the number of $\mathcal{DBR}$s that requested it.

Table 3.2: Types of PII elements required for VCR submissions.

PII type	**Count (out of 454)**
Email	444
Name	431
Address	Partial: 70, Full: 122
Phone number	103
Date of birth	Partial: 8, Full: 33
MAID	26
Official ID	Redacted: 13, Full: 4
SSN	Last 4 digits: 9, Full: 7
Signature/Signed affidavit	15
LinkedIn URL	11
Employer	11
IP address	8
Utility bill (redacted)	6
Multiple-Choice Questionnaire	5
Selfie	5
Cookie	5
Previous addresses	4
Driver's license number	2
Recently visited locations	2
Grocery store frequent shopper ID	1
Gender	1
Social media usernames	1
Marital status	1

Email address and name were requested by over 95% of $\mathcal{DBR}$s. Next, home address and phone number were requested by 42% and 23%, respectively. $\mathcal{DBR}$s likely use home address to confirm California residency and CCPA jurisdiction. Sometimes, only partial address information was required, e.g., city, zip code, or state. 21 $\mathcal{DBR}$s asked for professional information, e.g., the LinkedIn URL, and/or the name of the employer.

32 $\mathcal{DBR}$s asked for PII that is difficult to retrieve for the average consumer, such as cookie values, Mobile Advertising Identifier (MAID), or IP addresses. To obtain cookie information

on Google Chrome, one needs to `Inspect` the webpage, click on `Application`, and find relevant cookie values under `Storage`. On Android devices, user MAID is accessible within the phone's settings. However, for iPhone users, Apple does not directly provide access to MAID. Instead, iOS users must download a third-party app to view their MAID, which is a bit disconcerting.

Five $\mathcal{DBR}$s requested unusual information: marital status, gender, a list of recently visited locations, and a grocery store frequent shopper ID. Another five $\mathcal{DBR}$s presented a multiple-choice questionnaire asking the consumer to select their PI from a list. A screenshot of a sample questionnaire is shown in Figure 3.5.

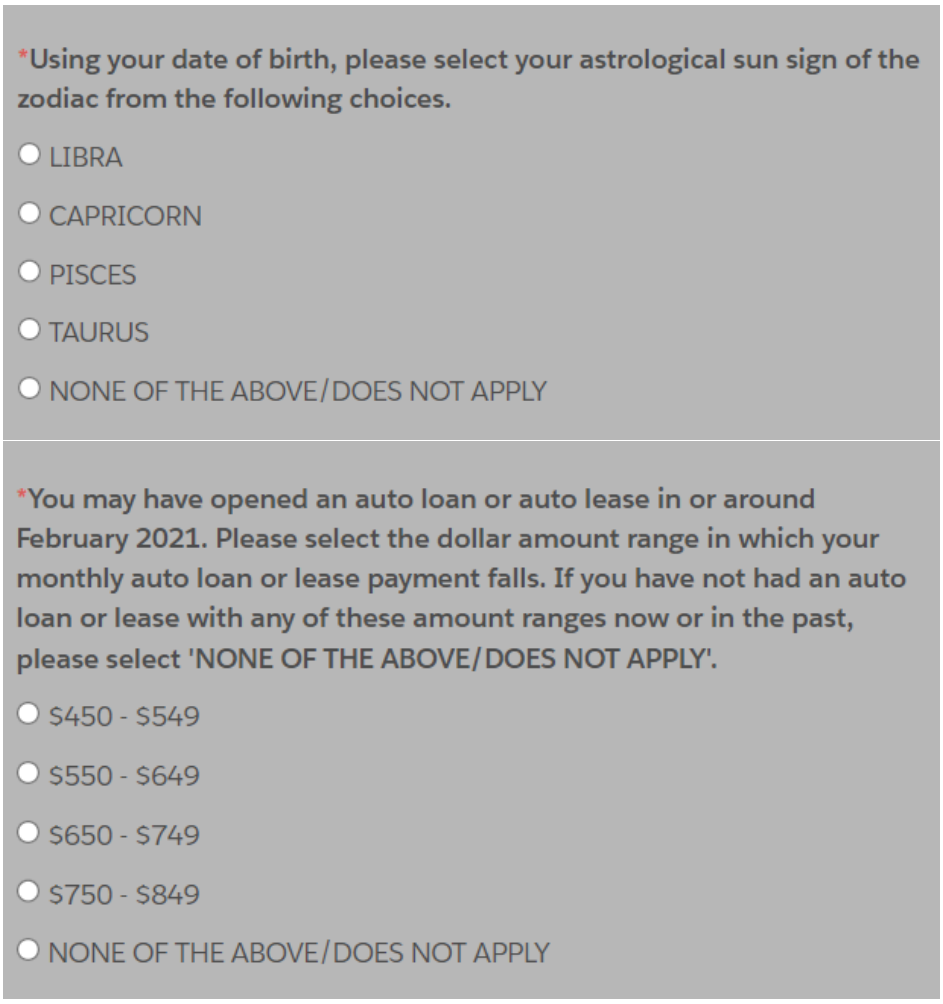


Figure 3.5: Questions from a questionnaire in a form-VCR.

Finally, 45 $\mathcal{DBR}$s asked for sensitive PII, as defined by CCPA. This included: full or partial SSN, full or partially redacted copy of a government ID, driver's license number, and biometrics, e.g., a signature or a selfie. This includes 4 $\mathcal{DBR}$s that used third-party systems requiring the consumer to take a live selfie along with a picture of a government ID.

Number of PII Elements in the VCR. The number of PII elements requested for identity verification varied depending on the VCR submission method. Since phone-VCRs were a small minority and mostly unsuccessful, a minimal amount of PII was shared through them, usually only phone number and name. Most email-VCRs require 2 PII elements, while about half of form-VCRs require 4 or more. A Mann-Whitney U test indicates that form-VCRs requested significantly more PII (p-value < 0.001) than email-VCRs. OneTrust forms did not ask for more PII elements than other $\mathcal{DBR}$ forms.

Submitting email-VCRs involved sharing my email address, from which the VCR was sent, and full name, signed at the end of the email. Seven $\mathcal{DBR}$s specified (in their privacy policy) additional PII elements to include in the email, and 23 requested additional information after the initial email, in order to continue with identity verification. In contrast, form-VCRs typically required all PII elements up front, at the initial submission time.

Additional Verification. Besides providing PII, a consumer needs to take additional steps to successfully submit form-VCRs. For instance, 101 $\mathcal{DBR}$s required confirming access to the supplied email address or phone number through a link or one-time code, or requested a copy of a recent utility bill to prove California residence. Many $\mathcal{DBR}$s with form-VCRs mandated ticking a box to acknowledge (under penalty of perjury) that I am the same consumer for whom the request was being made. Nevertheless, 13 $\mathcal{DBR}$s asked for a separate *signed affidavit*.

Moreover, 53% of form-VCRs were CAPTCHA-protected. While CAPTCHAs are used to prevent malicious bot activity, CAPTCHA solving is considered difficult for humans. Furthermore, bots are known to be faster and better than humans at this task [73, 15]. Access to 2 $\mathcal{DBR}$s' form-VCRs resulted in multiple CAPTCHAs: reCAPTCHA and a text-based or math-based CAPTCHA. Table 3.3 shows the frequency of CAPTCHA types observed on forms.

OneTrust forms overwhelm consumers with additional verification: they all require CAPTCHA completion, using either reCAPTCHAs or text-based CAPTCHAs. 77% require confirming access to the supplied email address. In contrast, only 37% of *non*-OneTrust forms are CAPTCHA-protected. Compared to other form-VCRs, OneTrust forms require significantly more additional verifications before submission (Mann-Whitney U test p-value < 0.001). I also needed to solve a text-based CAPTCHA every time I wished to access a OneTrust portal after VCR submission, e.g., to verify the request's status.

Table 3.3: CAPTCHA type frequency in CAPTCHA-protected form-VCRs.

CAPTCHA type	**Percentage**
reCAPTCHA	72%
text-based	20%
hCAPTCHA	8%
math-based	3%

Key Takeaways - RQ1 Part 2 The identity verification process for consumers is burdensome and intrusive. While most $\mathcal{DBR}$s require only a few PII elements, the type of these elements varies significantly. Some ask for PII that is inconvenient to obtain, such as mobile advertiser IDs or signed affidavits. The process can be highly privacy-invasive, with certain $\mathcal{DBR}$s requesting highly sensitive PII, e.g., SSNs or government IDs. Finally, $\mathcal{DBR}$s often implement additional verification steps (e.g., email confirmation, CAPTCHA solving) before allowing VCR submissions, primarily to prevent spam, yet further complicating the VCR

submission process for consumers.

3.4.3 RQ2: $\mathcal{DBR}$ responses

A substantial fraction of $\mathcal{DBR}$s (195 out of 454) **never responded** to VCR requests. Only 51.5% of the $\mathcal{DBR}$s responded on time, i.e., within 45 calendar days. Although no $\mathcal{DBR}$ asked for a permitted extension, 5.3% responded after the 45-day limit, thus failing to comply with the prescribed timeline.

We use the term *responding* $\mathcal{DBR}$s to denote those that replied to a VCR, whether on time or late. The rest, even if they initially acknowledged the VCR or corresponded to get more PII for identity verification, are called *non-responding* $\mathcal{DBR}$s.

Most responding $\mathcal{DBR}$s reported having no PI about me. Only 22 $\mathcal{DBR}$s provided some PI, detailed in Section 3.4.4. Figure 3.6 shows various replies from responding $\mathcal{DBR}$s.

14 $\mathcal{DBR}$s could not verify my identity. In 7 cases, their response arrived almost instantly after submitting a VCR form. These $\mathcal{DBR}$s did not ask for additional information to verify the identity. 8 $\mathcal{DBR}$s answered with irrelevant information, such as stating that I was now "opted out" of future sales or communication. Remarkably, one $\mathcal{DBR}$ refused to fulfill the request, falsely claiming that I am not a California resident.

Response rate. Since a significant fraction of VCRs were unanswered, we examined factors associated with receiving a response, such as $\mathcal{DBR}$ location, VCR submission method, number of PII elements requested, and sensitivity of the PII supplied. Note that the analysis of number of PII elements and sensitivity of PII supplied (Figures 3.7 and 3.8) excludes multi-step VCRs.

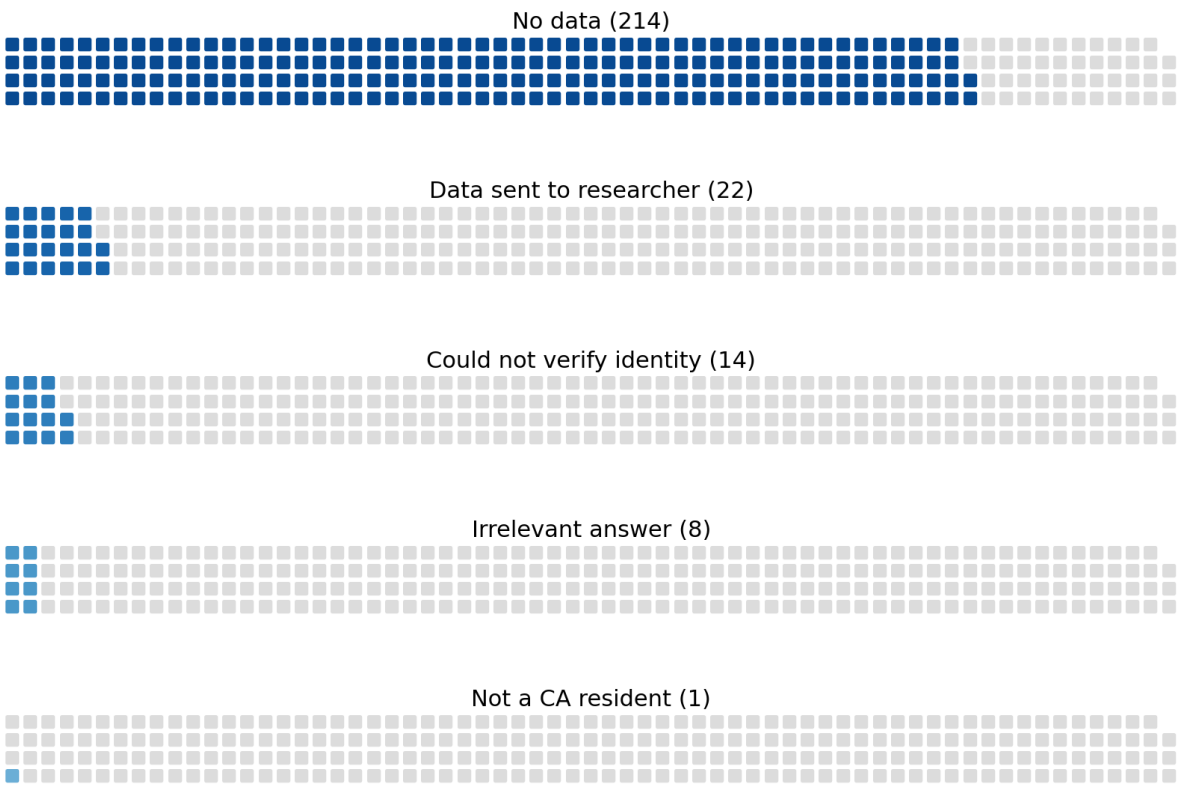


Figure 3.6: Distribution of answer categories received from $\mathcal{DBR}$s.

Geographical location. First, we analyzed response rates depending on $\mathcal{DBR}$s' location. We optimistically expected that $\mathcal{DBR}$s located in the US, especially those in California, would have a better grasp of CCPA and would therefore be more likely to answer. 56.6% US-based $\mathcal{DBR}$s and 60% non-US $\mathcal{DBR}$s responded to VCR requests. However, the latter are geographically dispersed and few in number: 25 total.

Table 3.4 provides $\mathcal{DBR}$s' response rate per state, in states where a VCR could be submitted to at least 10 $\mathcal{DBR}$s. Comparing three states with the highest $\mathcal{DBR}$ concentration (California, New York, and Florida), the response rate for California, at 58.1%, is only 5.6 and 1.3 percentage points higher than that of the other two, respectively. In other words, California $\mathcal{DBR}$s perform only marginally better than the US average of 56.6%.

VCR submission method. Beyond geographical factors, we hypothesized that $\mathcal{DBR}$s that used form-VCRs would be more likely to respond to a VCR. The rationale was that $\mathcal{DBR}$s

Table 3.4: Response rates of $\mathcal{DBR}$s in US states.

State	$\mathcal{DBR}$ Count	Resp. rate
California	93	58.1%
New York	61	52.5%
Florida	37	56.8%
Texas	23	56.5%
Massachusetts	20	70.0%
Georgia	16	62.5%
Virginia	14	50.0%
Colorado	14	50.0%
New Jersey	14	71.4%
Illinois	12	58.3%
Washington	11	72.7%
Ohio	10	70.0%

with established VCR submission processes would be more inclined to allocate both time and personnel to address these requests. Figure 3.7 shows $\mathcal{DBR}$s' answers depending on the VCR submission method.

Aligning with our assumptions, form-VCRs yielded a higher response rate of 71.5%. Email-VCRs, the second most common VCR submission method, yielded a response rate of 42.4%. This suggests that the majority of these $\mathcal{DBR}$s *only* meet a minimum regulatory requirement by providing an email address, without actually responding to received requests. Finally, phone-VCRs performed worse: only 42.9% received a response. $\mathcal{DBR}$s consistently failed to return calls after voicemails were left. In one notable instance, upon my request to exercise my CCPA rights, a $\mathcal{DBR}$ representative abruptly transferred the call to voicemail. No follow-up communication was received.

Predictably, 19 $\mathcal{DBR}$s with broken forms that had to be contacted via email or phone exhibited low response rates: 41.2% responded to emails, and none responded to phone calls. 6 of 8 $\mathcal{DBR}$s that sent a form to complete after an initial email contact responded to the

VCR, and 2 did not answer after they specifically pointed to a form.

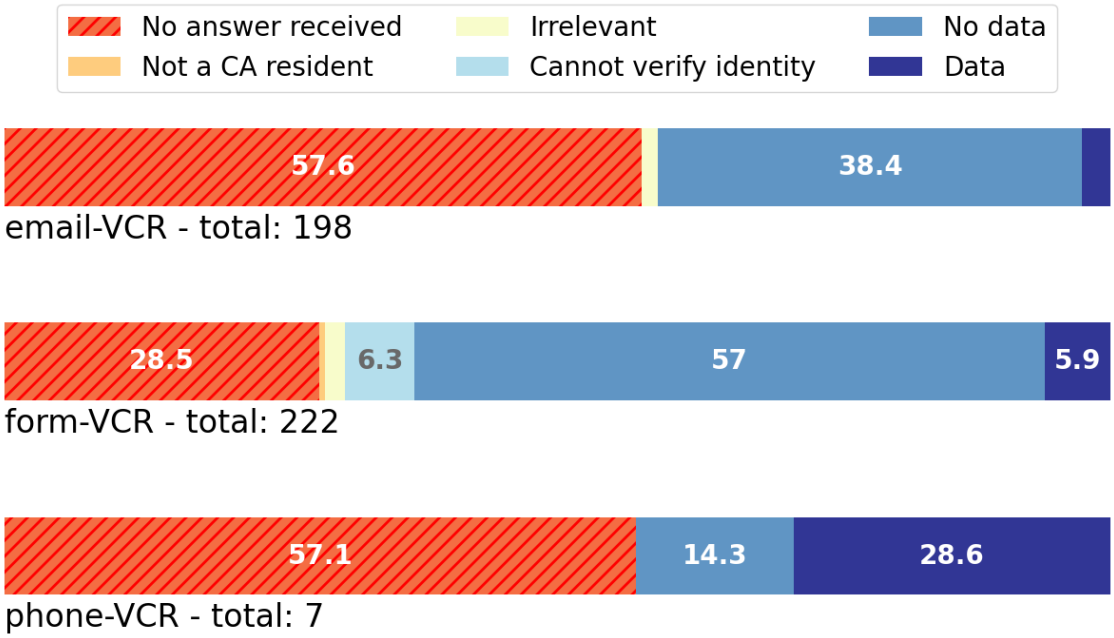


Figure 3.7: $\mathcal{DBR}$ responses for email, form, and phone VCR submissions (in %).

Number of PII elements requested. To verify the requester's identity, $\mathcal{DBR}$s ask for PII to match with their own records. As confirmed in Section 3.4.2, due to the lack of standards in CCPA legislation each $\mathcal{DBR}$ asks for different types of PII.

Figure 3.8 shows the response rate for email-VCRs and form-VCRs depending on the number of PII elements requested for identity verification. 168 email-VCRs were given 2 PII elements: email address and name, and were completed in 36.3% of cases only. 30 email-VCRs that required more than 2 PII elements were answered at a higher rate (76.7%). The number of PII elements requested in form-VCRs had no effect on the response rate.

Sensitive PII requested. Finally, we checked whether $\mathcal{DBR}$s that asked for sensitive PII responded to VCRs. Following CCPA's definition of "sensitive personal information", Table 3.5 lists the seven elements of sensitive PII occasionally requested, and $\mathcal{DBR}$s' response rate

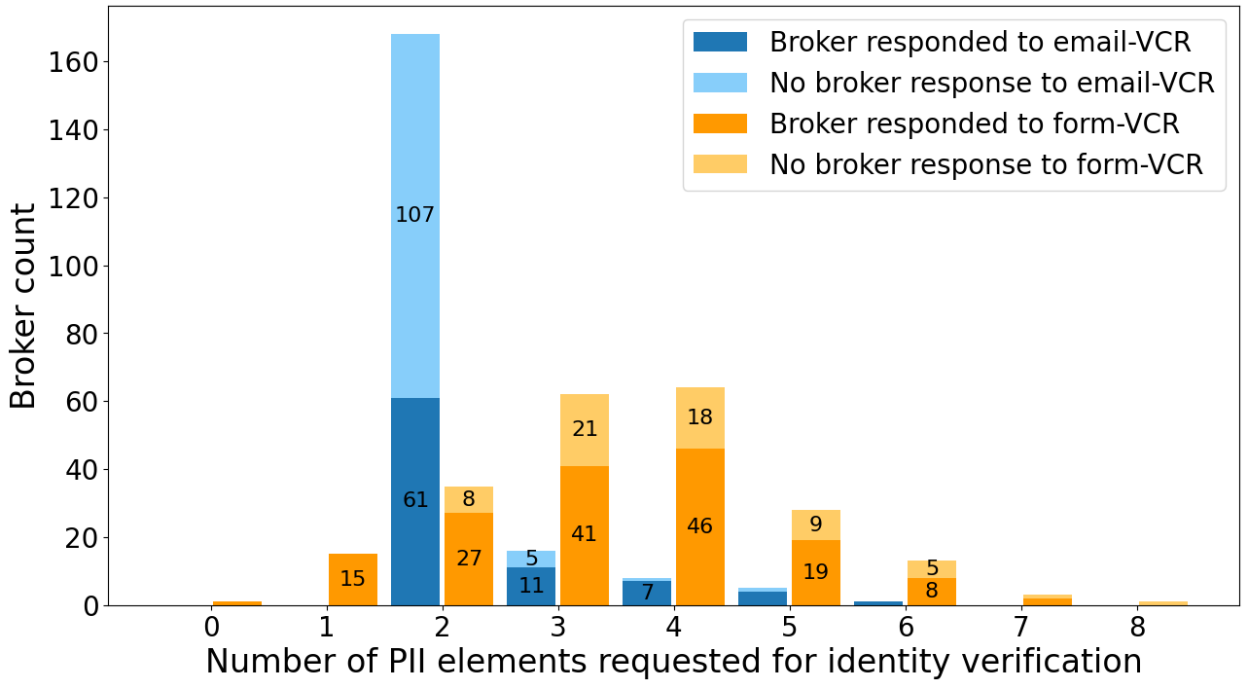


Figure 3.8: Number of responding $\mathcal{DBR}$s given increasing number of PII elements requested

when requiring these sensitive PII elements. Most $\mathcal{DBR}$s that required a full or redacted copy of a government ID completed the VCR. Meanwhile, only 2 of 6 $\mathcal{DBR}$s asking for a full SSN responded. This result is especially concerning:

> *Who now has access to this information? Does exercising one's CCPA rights in turn put a consumer at risk of identity theft?*

Two $\mathcal{DBR}$s that requested the last 4 digits of my SSN or that were sent a redacted copy of a government ID responded saying that they could not verify my identity: sharing sensitive PII was ineffective in these cases.

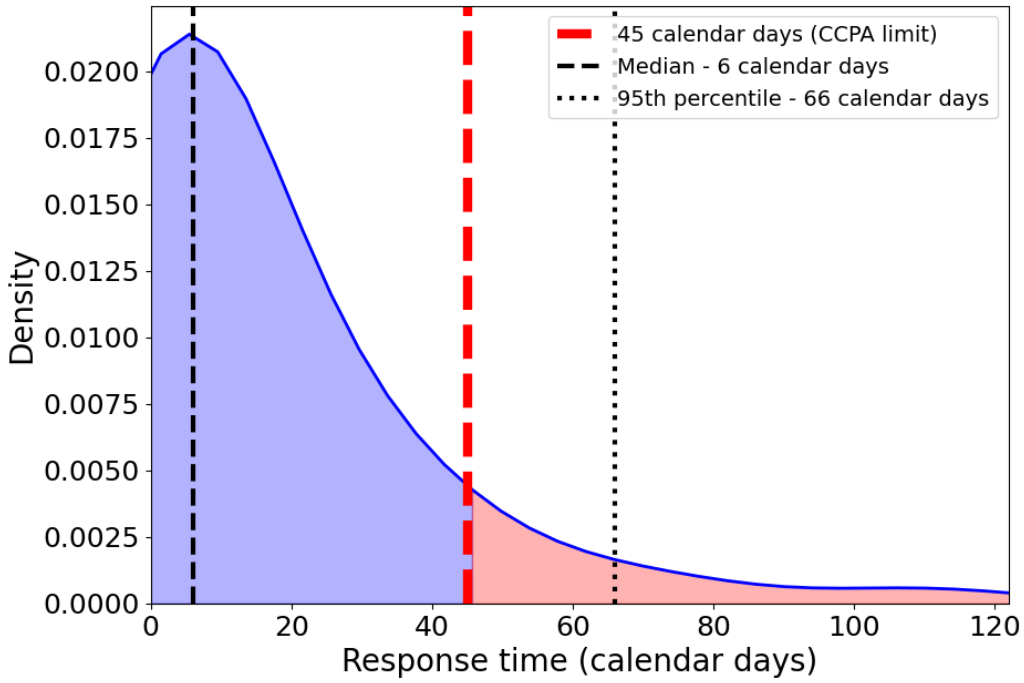
Response Timeline. Figure 3.9a shows the distribution of the response timeline, in calendar days, between VCR submission and $\mathcal{DBR}$ response, for the 259 responding $\mathcal{DBR}$s. Recall that CCPA requires $\mathcal{DBR}$s to respond within 45 calendar days. Half of the responding $\mathcal{DBR}$s answered the VCR within the first 6 days, and 90.7% responded within the allotted 45

Table 3.5: $\mathcal{DBR}$s response rate when sensitive PII is asked.

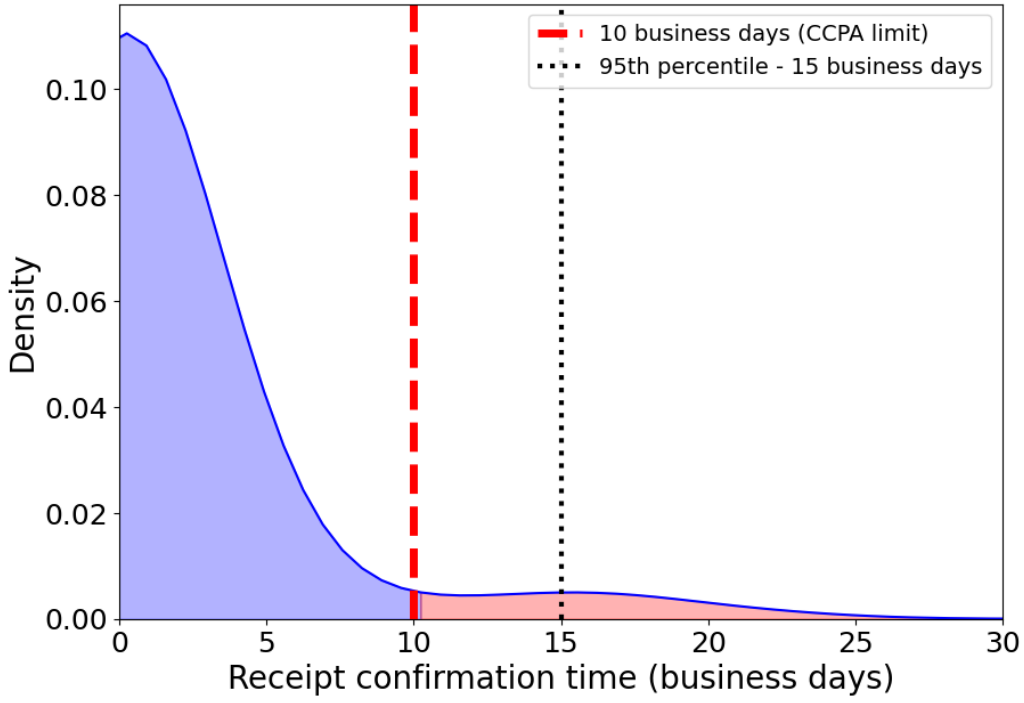
Sensitive PII	**Completed VCRs**
Last 4 digits of SSN	5/9 (55%)
Full SSN	2/6 (33%)
Redacted ID	11/13 (85%)
Full ID	3/4 (75%)
Driver's License Number	1/2 (50%)
Signature/signed affidavit	10/15 (67%)
Selfie	4/5 (80%)
Total	29/45 (64%)

days. $\mathcal{DBR}$s also have to confirm receipt of a VCR within 10 business days. We considered that every form-VCR was immediately confirmed as received, through the form submission confirmation message. For all other VCRs, we assume that the $\mathcal{DBR}$'s first email to me was confirmation of VCR receipt. Figure 3.9b shows the distribution of time, in business days, between VCR submission and confirmation of receipt. 93.6% of responding $\mathcal{DBR}$s confirmed receipt of the VCR within the allotted 10 business days. In fact, 81.8% confirmed receipt of the VCR within one business day, typically via an automatic reply system. Of the 195 $\mathcal{DBR}$s that did *not* respond, 69 confirmed receipt of the VCR without following up.

Response contents. Figure 3.10 shows the response distribution of response times categorized by the type of answer provided. $\mathcal{DBR}$s that had PI about me took longer to answer: this could be due to either a more thorough identity verification process, or the effort to find the collected PI. On the other hand, $\mathcal{DBR}$s that could not verify my identity or replied with irrelevant data (about opt-out or deletion rights) answered more expeditiously. However, it is uncertain whether their answers can be trusted, e.g., did they reply just to quickly "complete" (i.e., get rid of) the VCR?



(a) $\mathcal{DBR}$ response time



(b) $\mathcal{DBR}$ VCR receipt confirmation time

Figure 3.9: Distribution of VCR completion and receipt confirmation time.

OneTrust. OneTrust form-VCRs yielded a response rate of 79%, notably higher than the 69% rate of non-OneTrust form-VCRs. Therefore, while OneTrust form-VCRs required ad-

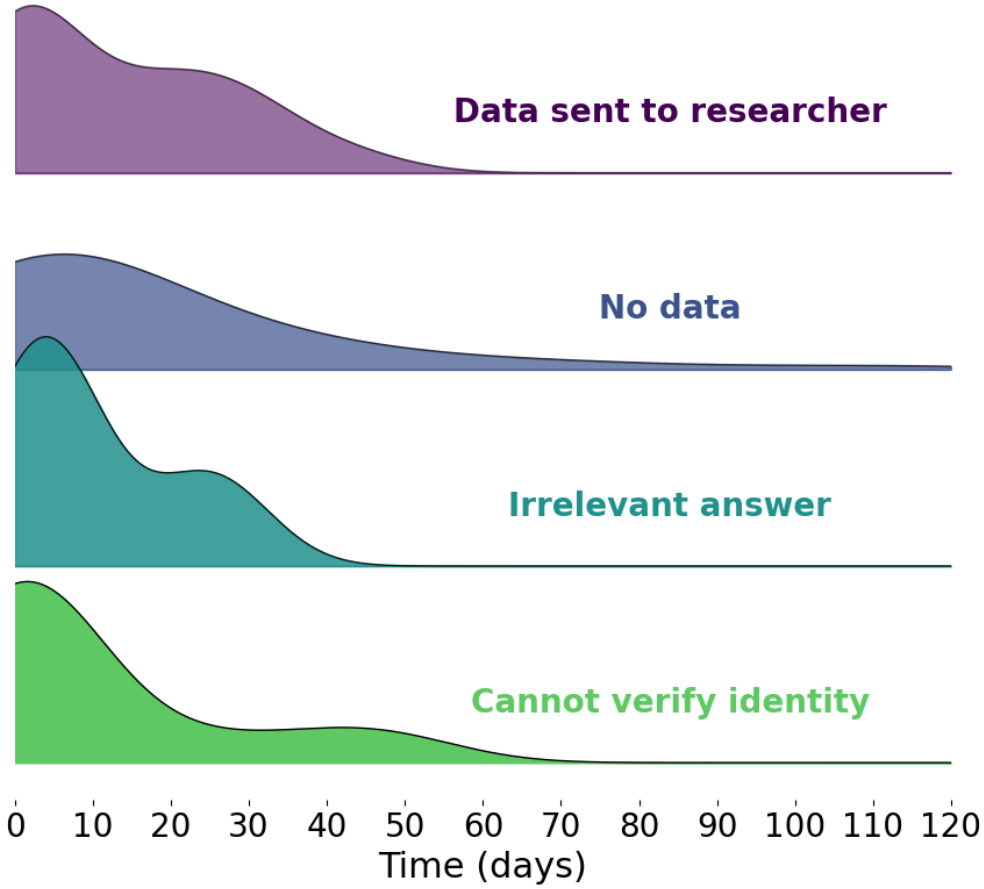


Figure 3.10: $\mathcal{DBR}$ response time depending on answer content.

ditional verifications (CAPTCHA, email confirmations), they had a higher "success" rate. We conclude that the use of such an automated system, preferably standardized, might help $\mathcal{DBR}$s meet compliance requirements.

Key Takeaways - RQ2. Measuring response rate revealed widespread non-compliance. A significant fraction of $\mathcal{DBR}$s failed to respond to VCRs, with only about half answering within the prescribed 45-day timeframe. Specifically, email and phone-VCRs were answered only in 42% of cases. Form-VCRs performed better, with a 71% response rate, potentially indicating that providing a form demonstrates a stronger commitment to CCPA compliance.

Alarmingly, even $\mathcal{DBR}$s requesting PII exhibited low response rates.

Among responding $\mathcal{DBR}$s, most adhered to CCPA-mandated 45-day limit. The pattern exposed a stark divide: $\mathcal{DBR}$s tend to either fully comply with or completely disregard CCPA obligations. Overall, these findings paint a rather disappointing picture of CCPA compliance.

3.4.4 RQ3: PI received from $\mathcal{DBR}$s

We analyze received PI to evaluate accessibility, accuracy, PI leakage, and tangentially, mapping PI requested in VCRs to the sensitivity of information $\mathcal{DBR}$s revealed. Recall that 22 $\mathcal{DBR}$s provided PI about me. These $\mathcal{DBR}$s' registered addresses span 11 states within the US, and three countries (USA, Germany, Canada). Table 3.6 presents details regarding the VCR process and the PI received from these $\mathcal{DBR}$s. The following PI was obtained:

- *Confidential PI (2).* A credit reporting agency sent a credit report, and a major data analytics company sent past and current car insurance information: policy numbers, dates, coverage, and listed drivers. This information is very sensitive.
- *Behavior inference data (3).* These $\mathcal{DBR}$s provided such data for advertising purposes, e.g., `"Samba TV → Ad Exposure → Electronics"`.
- *Publicly available professional information (6).* These $\mathcal{DBR}$s provided a copy of my public LinkedIn profile.
- *Physical addresses and cookie values (2).* These $\mathcal{DBR}$s provided a limited amount of PI, only giving one or two home addresses and some website-specific cookie ID.
- *Data from the VCR (5).* These provided information that they likely just obtained from the VCR itself. These include only my name and email address, and VCR-provided IP address, cookie ID, and mobile advertiser ID.
- *Categories of collected PI (3).* These provided the categories, not the actual PI. Thus, we cannot verify whether these $\mathcal{DBR}$s actually collected my PI or if a generic answer

was given.

PI Delivery. Two $\mathcal{DBR}$s that sent the most confidential PI did so by postal mail: the PI itself or a printed download link to the PI was sent to my home address. This measure ensured that only the consumer with the address matching that in the $\mathcal{DBR}$s' databases would receive the PI. It maintains data security as long as $\mathcal{DBR}$s are confident in the accuracy of their databases. Incidentally, the same two $\mathcal{DBR}$s required a higher-than-average number of PII elements and sensitive PII for identity verification.

Other $\mathcal{DBR}$s sent the PI via email or OneTrust portals. Three had PI directly accessible through their website, e.g., one prompted me to press a "show me the data" button, after which I could download a CSV file containing behavior inference data. $\mathcal{DBR}$s usually provided the PI as PDF, CSV, or XLSX files. One $\mathcal{DBR}$ sent files as a series of nested HTML files with non-descriptive titles, making sense of which is likely to be unintuitive for the average consumer.

Accuracy of PI. Behavior inference data received from 4 $\mathcal{DBR}$s was vague, with multiple age ranges or contradictory interests. Some $\mathcal{DBR}$s sent thousands of data points, and others dozens. Certain inferences were inaccurate, e.g., two $\mathcal{DBR}$s provided data inferring that I am Hispanic or Spanish-speaking, which is not the case.

PI Leakage. Alarmingly, the $\mathcal{DBR}$ from which I received car insurance information included sensitive PI about the following consumers:

1. My family members, included as drivers in my policy: this might be acceptable, since I entered this information myself when starting the policy.
2. My housemate, on whose policy I was listed as an excluded driver (an individual within

the same household specifically listed as not covered under the car insurance policy). We consider this to be a privacy leak. I was sent a third-person's PII, of which I had no prior knowledge.

Specifically, I received the policy number, start and end dates, coverage limits, and **7 out of 8 characters** of the housemate's driver's license number. While the first half of the report redacted the first 5 characters of the 8-character-long California driver's license number (e.g. XXXXX678), the second half of the report redacted the last 4 characters (e.g. A123XXXX). Accordingly, I obtained 7-out-of-8 characters (e.g., A123X678) and would only need to guess the 5th character, a decimal digit, to obtain my housemate's full driver's license number. We note that CCPA prohibits disclosing any consumer's "...driver's license number or other government issued identification number [in response to a request to know]" (CCPA Regs. §7024(d)).

Key Takeaways - RQ3. While a few $\mathcal{DBR}$s shared confidential PI or advertising behavioral inference data, the majority of PI shared was not particularly meaningful. Most $\mathcal{DBR}$s simply returned the PII provided by me, or at best, copied LinkedIn profile data.

$\mathcal{DBR}$s typically transmitted PI via email or OneTrust portal. Notably, two $\mathcal{DBR}$s that shared the most sensitive PII opted for postal mail delivery, demonstrating enhanced security measures. This suggests that $\mathcal{DBR}$s tailored their delivery methods to the sensitivity of the PII transmitted. File formats, with only one exception, were easily accessible and transmittable, following CCPA requirements. Surprisingly, one broker leaked sensitive PII about a third party.

3.5 Discussion

Given the study results, we now consider privacy tradeoffs in the VCR process, unexpected and noteworthy incidents, and finally recommendations stemming from these, for consumers, data brokers, and policymakers.

3.5.1 The Privacy Paradox

The goal of privacy laws is to give consumers greater control over their PI. Ironically, exercising one's CCPA-given privacy rights introduces new privacy risks and vulnerabilities. I sent a large amount of PII to hundreds of $\mathcal{DBR}$s, 95% of which either did not respond or did not previously collect any PI about me. In one email informing that no PI was collected about me, a $\mathcal{DBR}$ even stated, about the email address and name given to submit the VCR:

> "In fact, we never had this information until receiving it in your email below."

This highlights current privacy issues stemming from exercising one's CCPA rights. It is especially concerning when $\mathcal{DBR}$s request sensitive PII in VCRs. As noted in Section 3.4.3, 6 $\mathcal{DBR}$s asked for my full SSN, and 4 of them did not respond to the VCR. This raises serious concerns about identity theft risks created by the VCR process, which is intended to protect consumer privacy. Not only does the consumer submitting the request become vulnerable, but privacy threats to other individuals also emerge, i.e., as mentioned earlier, I received sensitive PI about another consumer. On a related note, prior work shows that an impersonator could easily receive another consumer's PI, since the PII required for identity verification is often public or easily obtainable [66, 14, 59].

This leaves two imperfect choices:

1. $\mathcal{DBR}$s should require a copious amount of (potentially sensitive) PII to more accurately identify the consumer, which is detrimental to consumer privacy.

OR

2. $\mathcal{DBR}$s should require less PII, thus favoring consumer privacy. However, they would be more prone to data breaches due to ease of impersonation resulting from a simplified identity verification process.

3.5.2 Unexpected and unintended outcomes

Even though I only asked to exercise my PI access rights, many $\mathcal{DBR}$s, along with responding to VCRs, announced that they were putting my information on their opt-out list. They either assume that I care about my privacy and anticipate an opt-out/deletion request, or they hope to not receive any more VCRs from me by proactively adding me to their opt-out list.

Several factors may explain why such a small percentage of $\mathcal{DBR}$s shared collected PI with me. Recall that only 22 $\mathcal{DBR}$s, 4.6% of all contacted $\mathcal{DBR}$s, and 8.1% of all responding $\mathcal{DBR}$s, shared PI which they collected about me. Beyond non-compliance, there are several potential reasons. CCPA only requires $\mathcal{DBR}$s to share PI that was collected within the past 12 months (CCPA §1798.130(a)(2)(B)). If PI was collected earlier, $\mathcal{DBR}$s might not have to disclose it. In addition, if the collected PI is considered "public" in any way, it is not considered PI under CCPA, and $\mathcal{DBR}$s do not have to disclose it (CCPA §1798.140(v)(2)).

Finally, we had no way of verifying the veracity of responding $\mathcal{DBR}$s that claimed they had no PI.

3.5.3 Noteworthy incidents

The study revealed several noteworthy incidents:

Noncompliance. Four $\mathcal{DBR}$s added my email address to their marketing/newsletter list, which clearly violates CCPA (CCPA §1798.130(a)(7)). Two of three major US credit report-

ing agencies (Experian, Equifax, TransUnion) did not respond to our request. One $\mathcal{DBR}$ that definitely had my exact current address (it was one of the multiple-choice questionnaire options) did not answer the VCR.

Inaccessible VCRs. One $\mathcal{DBR}$ required me to provide cookie values from a specified website. The website was down. Although we communicated this to the $\mathcal{DBR}$, no reply was received. One $\mathcal{DBR}$ required email access confirmation by clicking a link in an email, even though no link was included in that email.

Oddities. A couple of $\mathcal{DBR}$s used email obfuscation in their privacy policy, making it harder for both bots and humans to send emails. We also observed that employees of some $\mathcal{DBR}$s visited my LinkedIn profile, probably for identity verification purposes.

3.5.4 Recommendations

We now make a few improvement recommendations for all three stakeholders involved: policymakers, $\mathcal{DBR}$s, and consumers.

Policymakers. Policymakers need to mandate standardization of both the VCR submission process and the maximal set of information that $\mathcal{DBR}$s should be allowed to request for identity verification purposes. More importantly, increased enforcement is needed through random audits and legal actions against non-compliant $\mathcal{DBR}$s. While CPPA already fined some $\mathcal{DBR}$s for failing to register [23, 22, 24], we are unaware of any $\mathcal{DBR}$s facing legal repercussions for failing to process consumer requests.

Plus, there needs to be a standardized process for consumers to lodge grievances against unresponsive or otherwise non-compliant $\mathcal{DBR}$s. Finally, consumers need to be informed

about a history of such grievances against each $\mathcal{DBR}$ (e.g., why bother submitting yet one more VCR to a $\mathcal{DBR}$ that has numerous recent complaints).

***$\mathcal{DBR}$*s.** Of course, we strongly recommend that $\mathcal{DBR}$s comply with CCPA. We also suggest that $\mathcal{DBR}$s provide direct links to relevant VCR forms or email contacts as part of their CPPA registration, instead of providing a generic link to their privacy policy. This may help $\mathcal{DBR}$s to not confuse different rights: $\mathcal{DBR}$s should not opt-out consumers from the sale of their data if such a request was not made.

Furthermore, $\mathcal{DBR}$s must make the VCR submission process easy for consumers. If asking for online identifiers is essential (e.g., via cookie ID or MAID), obtaining such information should be trivial. Finally, $\mathcal{DBR}$s must train designated employees to properly handle responding to phone-VCRs and email-VCRs.

Consumers. Consumers should exercise caution when submitting VCRs and, if problems are encountered, file official grievances using the CPPA complaint form [21].

3.5.5 Limitations

Throughout this study, I sent VCRs to $\mathcal{DBR}$s. We acknowledge that having multiple individuals submit VCRs would yield more data. This is the subject of future work. In a similar vein, studying CCPA compliance for non-existent (or deceased) individuals might yield a better overall picture. However, that would also trigger more ethical issues. Comparing $\mathcal{DBR}$s' compliance with other privacy laws, e.g., GDPR, would also be interesting. However, GDPR unfortunately does not mandate $\mathcal{DBR}$ registration.

Our findings likely underestimate the full extent of non-compliance in the $\mathcal{DBR}$ ecosystem, since our methodology only assessed entities that have already fulfilled the basic legal CCPA

obligation of $\mathcal{DBR}$ registration. Unregistered data brokers (which by definition are already failing to comply with CCPA) were beyond the scope of this study.

Our results, in particular, the number of $\mathcal{DBR}$s that had data about me and their responses, might have been influenced by my specific characteristics and online behavior. I am under 30 years of age, privacy-conscious, and not very active on social media platforms. This naturally limited the amount of data collected by $\mathcal{DBR}$s. It is likely that $\mathcal{DBR}$s would have more data on older individuals, and/or those with richer financial histories (e.g., loans, debts, real estate, etc.), as well as ones who are more active on social media.

3.6 Conclusion

This comprehensive study of 543 California-registered data brokers reveals concerning patterns of CCPA (non-) compliance. A substantial portion of data brokers (43%) completely ignored VCRs. This widespread non-compliance undermines CCPA's efficacy and highlights significant enforcement gaps in current privacy regulation. Our findings expose a troubling paradox: exercising privacy rights under CCPA introduces new privacy vulnerabilities. Consumers must provide personal information – sometimes including sensitive PII, such as SSNs, government IDs, and biometric data – to data brokers who may never respond. This creates a privacy catch-22 where consumers must risk exposing additional personal data to potentially untrustworthy entities to learn what information these brokers already (do not) possess. The lack of standardization in verification procedures places the burden on consumers to navigate each broker's unique process, providing various levels of personally identifiable information and creating substantial barriers to accessing their own data.

Table 3.6: PI Received from $\mathcal{DBR}$s in Response to VCRs.

	Type of PI received	**PI transmission method**	**File format**	**#PII (verif.)**	**Time (days)**	**Location**
1	Credit report	Postal mail	N/A	6*	0	Georgia
2	Car insurance information	Link to PI sent via postal mail	pdf	7*	1	Georgia
3	Behavior inference for advertising	Email attachment	pdf	4	44	California
4		Expiring link	xlsx	4	29	Virginia
5		Directly website accessible	csv	0	0	New York
6	LinkedIn profile data	Email attachment	pdf	2	0	California
7		Email attachment	pdf	8	1	New York
8		Email attachment	pdf	3	5	Washington
9		Email attachment	json	2	17	Washington
10		Password-protected email attachment	xlsx	5	26	Delaware
11		Email content	N/A	4	1	California
12	Physical addr. and cookies	Expiring link	csv	7*	12	California
13	Website cookie ID	Directly website accessible	N/A	2	14	Canada
14	Email address	Expiring link	html	4	34	Virginia
15	Shortened first name	Directly website accessible	N/A	3	1	Colorado
16	IP address information	Email attachment	json	3	4	Nevada
17	Cookie ID and MAID	OneTrust Portal	N/A	5	32	Germany
18	Name, email address	OneTrust Portal	pdf	4*	22	New York
19	Categories of PI	Phone-call	N/A	2	0	Utah
20		OneTrust Portal	txt	4	0	Texas
21		Email attachment	xlsx, json	5	24	Massachusetts
22	History of services provided	OneTrust Portal	pdf	5	20	Georgia

* indicates sensitive PII was requested

Chapter 4

Let My Data Go: Data Brokers' Compliance with Opt-Out & Deletion Requests

Abstract

Data brokers are a largely American phenomenon. They collect vast amounts of personal information about most adult U.S. consumers, mainly without the latter's knowledge or consent. Accumulated data can be sold to anyone, including employers, landlords, insurance agencies, banks, governments (local, state, federal, and even foreign), as well as various malicious actors. This, in turn, enables discrimination, surveillance, identity theft, and stalking.

Recent regulations – such as the California Consumer Privacy Act (CCPA) modeled after EU's General Data Protection Regulation (GDPR) – were introduced to bolster consumer privacy, e.g., the rights to: (1) opt-out of the sharing or selling one's personal information, (2) delete one's personal information, and

(3) obtain a copy of that information. However, exercising these rights is not easy, as shown by our comprehensive study of the data broker ecosystem. We submitted both opt-out and deletion requests (using synthetic consumer identities) under the CCPA to all California-registered data brokers and investigated their responses and lack thereof.

While the majority seem to be compliant, a significant fraction is not and many failed to reply to (and/or acknowledge) consumer requests. Furthermore, some data brokers require intrusive consumer identity verification in order to exercise one's opt-out rights, which is explicitly disallowed by the CCPA. There is also great disparity in the request submission process among data brokers as well as an extremely heavy (time & effort) overall consumer burden. This motivates an urgent need for streamlining and standardization of the consumer interface, stronger enforcement, and meaningful consequences for (especially sustained) non-compliance.

4.1 Introduction

Data brokers are business entities that collect a wide variety of consumers' personal information, usually without the latter's knowledge or consent, and sell that information to anyone willing to pay for it. It is estimated that there are over one thousand data brokers in the world, the majority of which are concentrated in the United States [31, 68]. In fact, the entire concept was "born" in the United States during the mid-to-late 1980-s.[1]

Anyone can buy consumer information from data brokers, including private individuals, as well as foreign and domestic businesses and governments, including law enforcement and intelligence agencies. Data collected by data brokers range from relatively benign contact information to much more sensitive data, such as health, financial, and even precise geo-

[1]Perhaps unsurprisingly, it originated in the state of Florida [39].

location.

In recent times, a more dangerous trend has emerged: the U.S. Department of Homeland Security (DHS), particularly its Immigration and Customs Enforcement (ICE) branch, has been purchasing information about American consumers from data brokers. This has been widely reported, e.g., by the American Civil Liberties Union (ACLU) and the Electronic Frontiers Foundation (EFF) [79]. Although government agencies are not allowed to collect data about U.S. population, apparently they are not prohibited from buying it from commercial entities.

Thus, data brokers pose a significant threat to consumers whose personal information is accumulated, then bought, and accessed by numerous entities, some of which certainly do not have consumers' best interest in mind, to put it mildly. Meanwhile, many (perhaps most) consumers are unaware of (1) who collects their personal information, (2) what exactly is being collected, and (3) what and to whom information is sold.

Recent privacy regulations across the world give consumers more agency over their personal information and grant them certain rights, e.g. access one's personal information, delete it, or opt-out of its sale and sharing. These regulations notably include European Union's General Data Protection Regulation (GDPR) [37], California Consumer Privacy Act (CCPA) [17], and Brazilian Lei Geral de Proteção de Dados (LGPD) [45], Furthermore, some regulations incorporate specific provisions for data brokers. In particular, California's Data Brokers Registration law [18] mandates that every data broker doing business in the state much register annually and pay a registration fee. The resulting list of registered data brokers is made publicly available [19].

With such a public list, residents of relevant jurisdictions can contact these data brokers and exercise their legal rights. However, exercising one's rights can be complicated: contacting every single data broker is time-consuming, since there are many, plus methods of contact and

user interfaces for submitting consumer requests vary widely. Moreover, getting a response to a consumer request is not guaranteed [86, 78].

Very recent previous work analyzed compliance and behavior of data brokers when submitting consumer data access requests [86], and found that more than 40% did not respond to well-formed requests. Also, very few had data about the subject consumer, i.e., the researcher who conducted the study. Although an access request yields important information about what kind of consumer information is collected by data brokers, delete and opt-out requests are (by far) much more numerous. In 2024, there were 58 and 99 million, delete and opt-out requests, respectively, compared to just 4 million access requests. (These numbers are based on the aggregation of most recent reported metrics in the broker registry [19].) Therefore, compliance with delete and opt-out requests seems to be more important since they are much more frequent.

Motivated by the above, this chapter reports on a comprehensive study of California-registered data brokers in the context of delete and opt-out requests. Results of this study can be viewed as a natural follow-on to [86]. It aims to answer the following research questions:

- RQ1: What are the differences, if any, in the submission process of opt-out and delete requests? For example, in terms of submission methods, information requested, user interfaces, burden, etc.

- RQ2: To what extent do data brokers comply with CCPA in terms of delete and opt-out requests? Are there any trends in terms of differences in compliance with one vs. the other?

- RQ3: What is the overall consumer burden involved in submitting each request type to all data brokers?

We note that delete and opt-out request types are semantically different: the latter requires

no identity verification and is thus submittable by anyone, on anyone's behalf. In contrast, deletion requests require identity verification. In the study described below, We systematically probe data brokers with both request types using synthetic (non-existent) consumer identities and analyze their responses. Results show (1) significant non-compliance, (2) wide variety of requirements for submitting requests, (3) erroneous responses, and (4) very heavy burden for an average consumer to interact with data brokers.

This work aims to make three contributions:

1. It demonstrates a lack of transparency in data broker request processing, where the consumer is left unsure whether their request was even considered.

2. It yields evidence of recurrent CCPA non-compliance, whether due to ignorance, negligence, or malfeasance. (We make no conclusion about specific non-compliance reasons).

3. It highlights the inordinate amount of effort a consumer must invest into dealing with numerous data brokers, via a range of submissions methods, and a variety of user interfaces.

Organization: Background and related work are given in Chapter 2. Section 4.2 describes the setup and preliminaries of the study. Then, Section 4.3 discusses and analyzes study results. Next, Section 4.4 summarizes limitations and Section 4.5 outlines future work directions.

4.2 Methodology

This section described our study methodology, including timeline, identities, $\mathcal{DBR}$s selection/exclusion, and ethical considerations.

4.2.1 Study Timeline

We submitted opt-out and deletion requests to all $\mathcal{DBR}$s registered in the state of California, as of early October 2025. The bulk of the submission process lasted roughly two months, starting in early October, and ending in early December, 2025. Two researchers, based in California, split the work: one was tasked with submitting delete, and the other – opt-out, requests. Each researcher worked on their own schedule and at their own speed. Consequently, the rate of request submission varied greatly between the two researchers and their individual request submission timelines were not in any way dependent on each other. This was done for two reasons: (1) synchronizing their workloads would have been infeasible due to external factors, and (2) independent timelines naturally inhibit correlation between their respective activities, as would be observed by $\mathcal{DBR}$s.

The responses started trickling in from the very beginning. Most $\mathcal{DBR}$s answered either on the day of, or a few days after, receiving the request. Latest answers were received about 60 days after delete, and about 100 days after opt-out, requests were submitted. Further response timeline detailed are presented in Sections 4.3.4 and 4.3.4.

Note that the discussion of overall time it took to submit both request types appears later in Section 4.3.6.

4.2.2 Use of Synthetic (Fictitious) Identities

We used two distinct synthetic (non-existent or fictitious) identities, one for each request type. We believe that synthetic identities are preferable to real ones, since submitting requests (to hundreds of $\mathcal{DBR}$s) using the information about the latter would impact those real individuals. Orthogonally, unlike closely related recent work in [86], we chose not to use the researchers' (i.e., this study and chapter authors') own information, since a simple web search on researchers' names would hint at their affiliations, publication histories and research profiles. That would likely expose the purpose of this study, which would have

been obviously undesirable: we wanted $\mathcal{DBR}$s to be unaware of the study's purpose, since otherwise they could treat our requests differently (whether better or worse) than organic ones.

We used two identities, as opposed to just one, since we wanted the delete and opt-out requests not to be easily correlated by $\mathcal{DBR}$s. Of course, we could have used more than two identities – in the extreme, one distinct identity per request. However, given the large number of queries, creating as many realistic-sounding identities along with accompanying data (dates-of-birth, addresses, phone #-s, email) would have been prohibitively burdensome.

The details of the two synthetic identities are shown in Tables 4.1 and 4.2. The names correspond to non-existent individuals. We carefully selected the names by thoroughly searching for each and made sure that there is no one with that name, in order to avoid any impact on any real person.

As reflected in the tables, dates-of-birth and phone numbers are fictitious, though their area codes match to those assigned to the geographical areas that correspond to the addresses we used. However, the email addresses are real in the sense that we created them in order to be able to confirm email account access if that was needed during the request submission process. Finally, Mobile Advertising IDs (MAIDs) are fake. A MAID is a sequence of 32 hexadecimal digits generated by the OS of a mobile device (e.g., a smartphone or tablet) used by advertisers to track activity, app usage, and location data, even across devices. It is resettable by users.

We chose one female- and one male-seeming identity for the sake of gender diversity. We also opted for having a large-ish age gap between them (over 30 years), for age diversity. Physical addresses included Northern (Sacramento) and Southern (Riverside) California, for geographical diversity. Finally, the types of physical addresses that we used were motivated by socioeconomic diversity: the Riverside address corresponds to a real trailer park, while

the (real) street name in Sacramento is located in a solidly middle-class neighborhood. In both addresses, zip-codes are real.

Table 4.1: Synthetic Identity I (used for all deletion requests).

Item	**Value**
Name	Patrick Puzzolente
Email	patrickpuzzolente@gmail.com
Date of Birth	July 7, 1993
Phone # (fake)	951-893-3150
Address (real, trailer park)	220 E. La Cadena Dr, Riverside, CA 92507
MAID (fake)	F2E4C0A7-11B8-4009-A6D4-3C9B7F0E2A11

Table 4.2: Synthetic Identity II (used for all opt-out requests).

Item	**Value**
Name	Annamaria Krahenschrei
Email	annamariakrahenschrei@gmail.com
Date of Birth	May 24, 1960
Phone # (fake)	916-646-2451
Address (fake #, real street)	6452 Valetta Way, Sacramento, 95820
MAID (fake)	6A3F1B9E-0D2C-4A51-9E3B-8F2C1A7D9B4E

4.2.3 $\mathcal{DBR}$ Selection/Exclusion

Using the California Data Broker Registry [19], we aimed to contact all 535 $\mathcal{DBR}$s registered at the time of the study, in late 2025. However, some $\mathcal{DBR}$s had to be excluded from the study due to inability to submit requests, for various reasons, such as:

- Website issues, mainly due to broken URLs
- Duplicate registrations of the same business
- Requirements to provide information (for identity verification purposes) that we either could not, or did not want to, invent, e.g., cookie values or Social Security N umbers (SSNs)

- Issues with forms that prevented successful submission, e.g., fields or CAPTCHA-s that were mis-configured.
- Lack of contact information in the privacy policy, either for all requests or specifically missing methods for deletion requests.
- Other: this includes individual errors, e.g., being forced to log into an account, or email bouncing back.

We wound up submitting a total of 322 deletion, and 358 opt-out, requests. These numbers correspond to successfully contacted $\mathcal{DBR}$s. The difference ($36 = 358 - 322$) is, for the most part, due to the fact that some $\mathcal{DBR}$ websites only had opt-out forms available and omitted deletion rights. In addition, CCPA, as mentioned earlier, mandates no identity verification for opt-out requests, while it is required for deletion requests: we did not submit to $\mathcal{DBR}$ asking for PII we could not invent. Therefore, we could submit 36 more of opt-out requests. Table 4.3 summarizes $\mathcal{DBR}$ exclusions.

Table 4.3: Data Brokers Excluded from the study.

Reason for exclusion	**#$\mathcal{DBR}$s (delete)**	**#$\mathcal{DBR}$s (opt-out)**
Website down	8	14
Duplicate	25	25
Asked too much PII	85	82
Form issue(s)	33	46
No contact info given	45	7
Other	17	3

For each $\mathcal{DBR}$, we accessed the link indicated in the $\mathcal{DBR}$ registry that contains details on how consumers can exercise their CCPA rights, including how to delete their personal information. This link usually leads to one of: (1) a $\mathcal{DBR}$'s privacy policy, or (2) a CCPA-specific privacy policy, or (3) a form where the consumer can exercise their privacy rights. We submitted deletion and opt-out requests by form whenever possible. Whenever no form

was available, we submitted requests by email. The email templates that we used are shown in Figures 4.1 and 4.2. The two email templates are differently worded, for obvious reasons.

Subject: CCPA opt-out request

Hello,

Under the CCPA opt-out right, I request that you stop selling or sharing my personal information.

Let me know if you need additional information.

Thank you for processing this request.

Sincerely,
Annamaria Krahenschrei

Figure 4.1: Opt-out request email template.

Subject: Request for deletion of personal information under CCPA

To whom it may concern,

I am writing to request the deletion of my personal information under the CCPA. Please let me know if you need any further information.

Thank you for your timely response.

Best regards,
Patrick Puzzolente

Figure 4.2: Delete request email template.

We recorded dates of all request submissions, confirmation of request receipt by $\mathcal{DBR}$s (if any) and their final responses. These responses were later categorized and analyzed.

4.2.4 Ethics

Our Institutional Review Board (IRB) deemed this study to fall into the "Non-Human Subject Research" category. We contacted $\mathcal{DBR}$s only once per request type (twice total) and did not invent sensitive personal information when it was required, e.g., SSNs or copies of government-issued ID cards. Instead, we opted not to submit requests to $\mathcal{DBR}$s that request such information.

One potential issue is our use of two fictitious identities (along with related information for each) for request submission. This can be viewed as a kind of deception. However, as mentioned in Section 4.2.2, we made sure that these identities did not correspond to any real person, whether living or dead. Also, their names and other information were carefully constructed to appear realistic. We believe that this was justified, since the alternatives, e.g., using identities of real people, including those of authors, would have been ethically problematic and/or would have leaked the purpose of submitted requests.

4.3 Results and Analysis

We now present results obtained from submitting requests to $\mathcal{DBR}$s as described in Section 4.2.

4.3.1 Results Summary: Compliance Rates

We observed the following instances of non-compliance, summarized in Table 4.4:

1. Lack of response after a request to delete: as required by CCPA Regulations [16] and mentioned in Chapter 2, $\mathcal{DBR}$s must notify the consumer of the status of their request for deletion after processing it. We found that a majority of $\mathcal{DBR}$s did not answer after an email-based request submission, and a majority of $\mathcal{DBR}$s considered the confirmation page (after form submission) as a response.

2. Identity verification during opt-out requests: again, CCPA Regulations [16] specify that businesses should not verify the identity of consumers when they submit opt-out requests (see Chapter 2). Recently, Ford Motor Company was fined $375,703 due to excessive identity verification in their opt-out request submission process. Excessive verification involves asking for more information that is actually needed to identify the consumer in the $\mathcal{DBR}$'s database [25]. It occurs when $\mathcal{DBR}$s ask for too many items

of personal information, or attempt to verify consumer's access to their email address or phone number.

3. Late answers deletion requests: $\mathcal{DBR}$s must process the request and respond within 45 calendar days. They can request an extension for another 45 days, however, none did so in this study.

In addition, we observed the following instances of potential non-compliance summarized in Table 4.4:

1. Lack of answers after opt-out requests: while automatic form submissions can be considered a confirmation of opt-out, $\mathcal{DBR}$s contacted by email and who did not reply did not show any confirmation that the opt-out request was taken into account.

2. Late answers after opt-out requests: $\mathcal{DBR}$s must process any opt-out request within 15 business days. Some $\mathcal{DBR}$s responded to our request after that deadline: it is unlikely that the personal information would be opted-out before the 15 business days deadline.

3. Potentially excessive identity verification: While these $\mathcal{DBR}$s did not specify that they needed to verify the consumer's identity, they asked for a greater amount of personal information that is necessary for identifying the consumer for opt-out requests.

4. Erroneous answers for deletion requests: $\mathcal{DBR}$s are supposed to announce actions taken to the consumer after processing a deletion request. Many $\mathcal{DBR}$s replied that data was deleted, even though the consumer was synthetic and the data obviously does not and did not exist.

5. Requiring access to email address to learn the status of an opt-out request: while some $\mathcal{DBR}$s (correctly) did not require a consumer to verify access to their email address when submitting an opt-out request, $\mathcal{DBR}$s that use the OneTrust platform need the consumer to verify access to their email address before being able to view

the response. While the personal information is already opted-out, this introduces an additional burden on the consumer if they want to check the $\mathcal{DBR}$'s answer.

Table 4.4: Instances of non-compliance

Issue	**Request**	**Percentage**
Non-compliance		
Lack of answer	Deletion	70%
Identity verification	Opt-out	22%
Late answer	Deletion	1.2%
Potential non-compliance		
Lack of answer	Opt-out	14%
Late answer	Opt-out	12%
Identity verification	Opt-out	39%
Erroneous answer	Deletion	2.5%
Email verif. to view answer	Opt-out	11%

4.3.2 Request Submission: Lack of Uniformity

Previous work [86] has shown that submitting access requests is very time-consuming and does not follow a standardized process: each $\mathcal{DBR}$ is free to have its own means for a consumer to submit access requests. Since there are (respectively) 15× and 25× more deletion and opt-out requests than access requests, we hoped that submitting the most popular types of request would be more streamlined. However, we witnessed the same ad-hoc submission process for delete and opt-out requests. While a California resident could now use the convenient DROP platform [20], consumers in other jurisdictions (e.g. Vermont, Texas or Oregon) cannot do so and must submit individual requests to each $\mathcal{DBR}$.

The majority of deletion and opt-out requests could be submitted via a form, 75% and 80% respectively. Less than a third were submitted by email. Results are shown in Figure 4.3. While email request submission is faster for the consumer (since the same message template can be used for all email-using $\mathcal{DBR}$s), it sometimes requires a second step of providing

more information to verify consumer identity or identify the consumer. In contrast, form-based submission asks for all required information at the same time: this usually means that more items of information need to be provided to the $\mathcal{DBR}$, and each $\mathcal{DBR}$ may ask for different types of personal information. In addition, forms regularly entail solving Captchas to prevent bots, which translates into an added burden for consumers [15, 73].

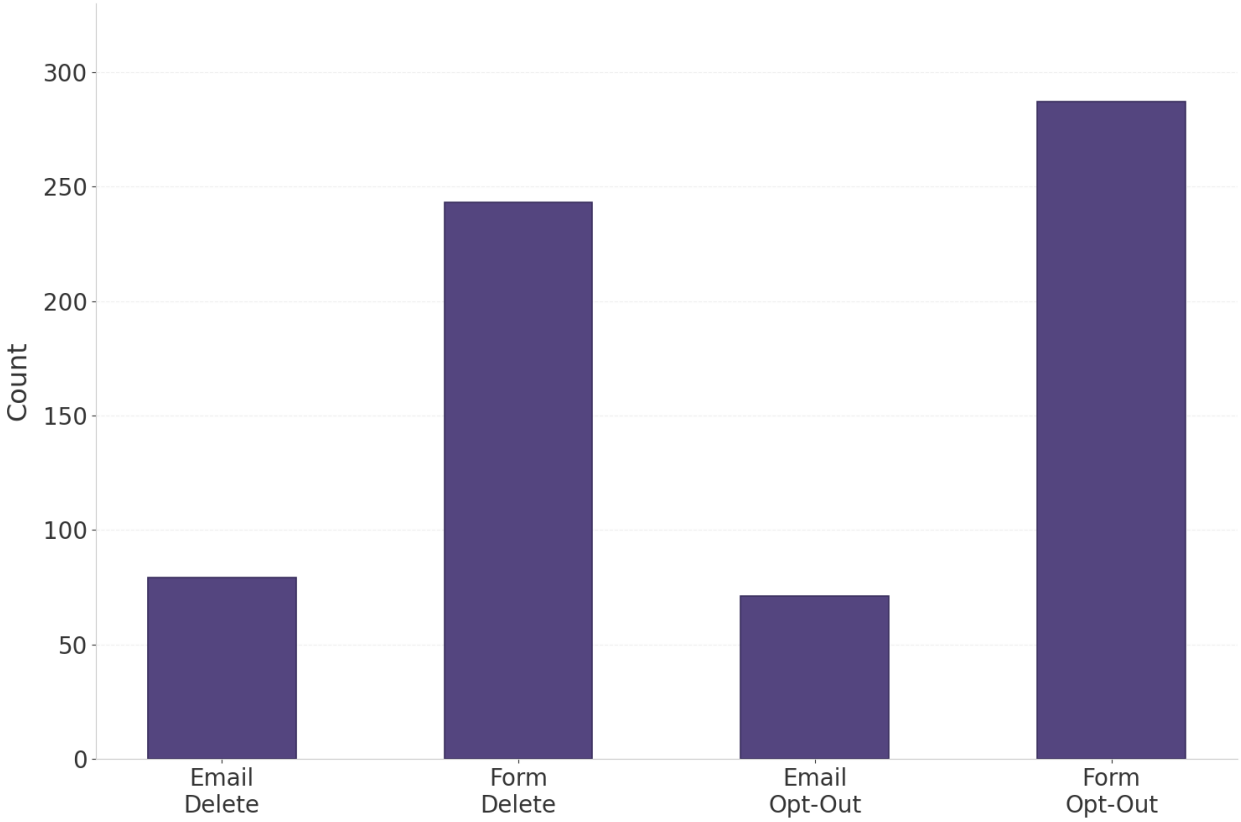


Figure 4.3: Method of Request Submission

4.3.3 Identity verification process

This section describes the details, and types, of identity verification encountered for in submitting each request type.

Deletion Requests

To request deletion of their personal information, consumers must submit a VCR. $\mathcal{DBR}$s must verify the identity of the consumer, and then process the request, i.e., delete any

personal information relating to that consumer. To verify consumer identity, $\mathcal{DBR}$s ask the consumer to provide some items of personal information in order to match it with what is stored in their databases. If enough personal information matches, consumer identity is considered "verified" and the $\mathcal{DBR}$ can proceed with data deletion [16]. $\mathcal{DBR}$s may ask for varying types (and amounts of) personal information. Figure 4.4 summarizes personal information requested by $\mathcal{DBR}$s contacted using a form. For $\mathcal{DBR}$s contacted via email, only the full name and email address of the synthetic consumer were provided.

Some $\mathcal{DBR}$s asked for information that we chose not to falsify because we either could not (e.g., device cookie values), or did not want to, do so for ethical reasons, e.g., SSNs.

Opt-out Requests

To opt out of sharing and further collection of their personal information, a consumer must submit some information as part of their request, though a VCR is not required (see Chapter 2): even though $\mathcal{DBR}$s are not supposed to request any personal information for verification purposes from consumers who submit opt-out requests, they can ask for personal information needed to identify the consumer in their databases. Nonetheless, We found that many $\mathcal{DBR}$s attempted to *verify* consumer identity during opt-out requests by requesting additional information.

4.3.4 Answers and Lack Thereof

For both deletion and opt-out requests, most $\mathcal{DBR}$s did not explicitly confirm having processed requests after a form- or email-based submission.

Delete Processing

As noted in Chapter 2, $\mathcal{DBR}$s have 45 calendar days to inform the consumer about the result of their deletion request. Among the 322 $\mathcal{DBR}$s contacted, only 97 responded with

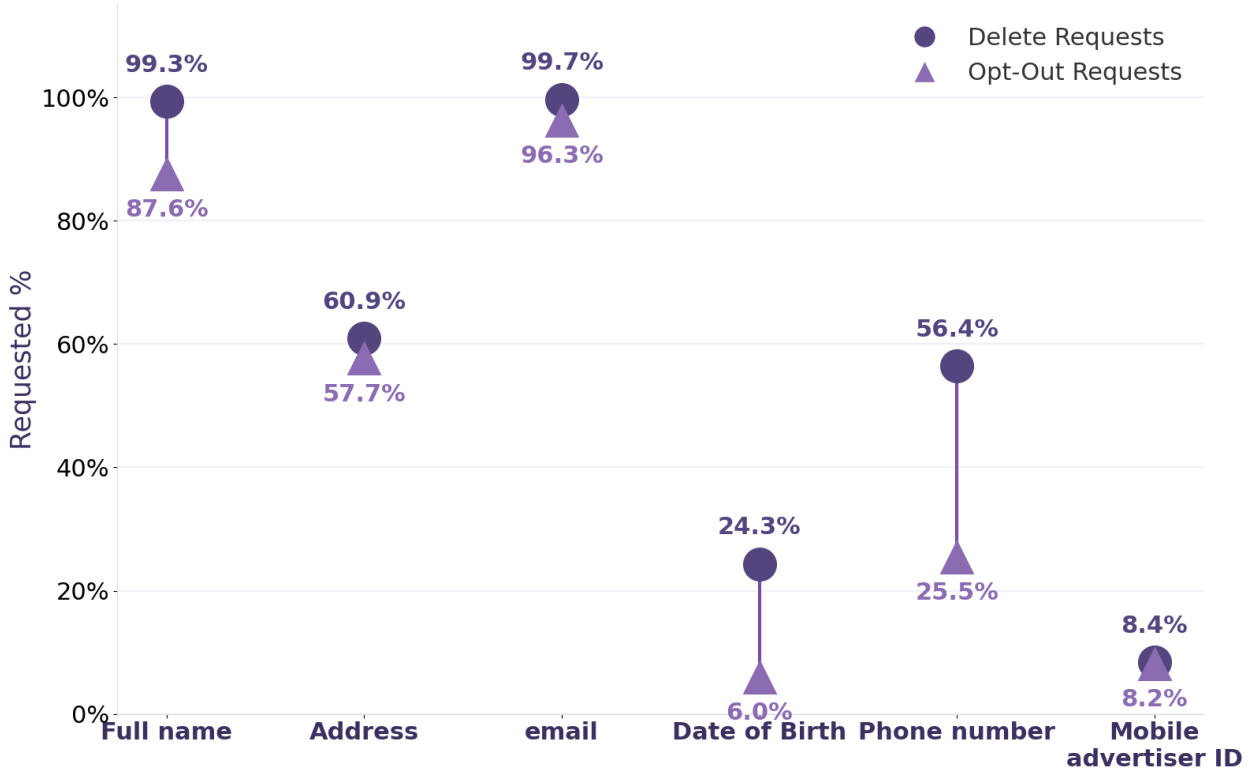


Figure 4.4: Information required for request Submission

the result of the deletion requests. We summarize their answers in Figure 4.5. Some $\mathcal{DBR}$s might assume that the form submission confirmation message is equivalent to the response of confirmation for the consumer. This may be true depending on the $\mathcal{DBR}$s' deletion process. If a $\mathcal{DBR}$ deletes consumer's data immediately, a message stating "your data was deleted" suffices. Otherwise, the consumer is left in the dark as far as the outcome of their request. It is impossible to tell the difference between the two cases.

The response timeline for deletion requests was generally encouraging, with the large majority of $\mathcal{DBR}$s replying within the prescribed 45-day limit (as shown in Figure 4.6), though 5% of responding $\mathcal{DBR}$s exceeded it. Half of responding $\mathcal{DBR}$s replied within the first 3 calendar days.

Opt-out Processing

$\mathcal{DBR}$s need to act upon an opt-out request within 15 business days after its receipt. The

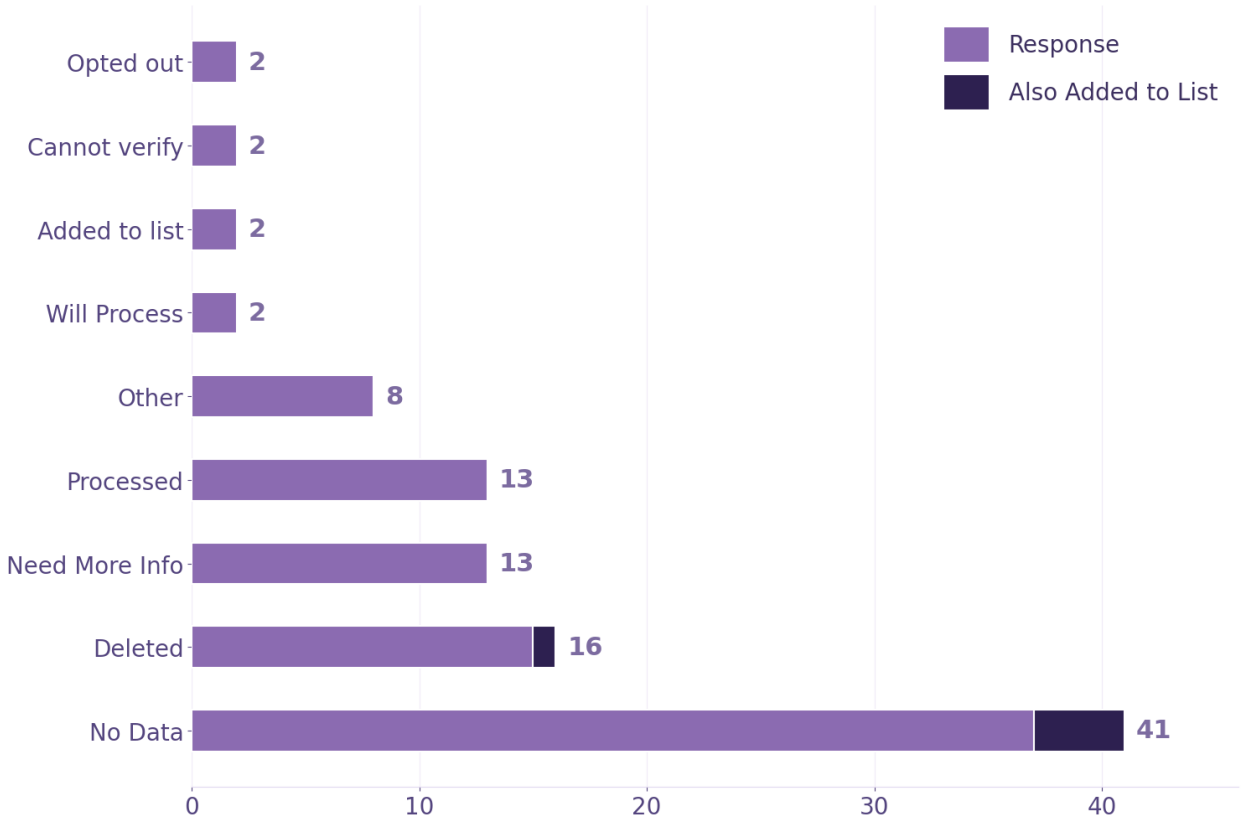


Figure 4.5: Answers for deletion requests.

law does not mandate them to directly respond to consumers. Instead, it states that a consumer must have a way to confirm that they have been opted out [16]. However, without a response from $\mathcal{DBR}$s and considering the low compliance with deletion requests observed in this study as well as the similarly poor compliance with access requests from [86], it is unclear whether opt-out requests are actually processed appropriately. 34 $\mathcal{DBR}$s confirmed, by email, that they processed the request. 66 $\mathcal{DBR}$s announced that no data was found about the consumer, of which 11 also added the consumer's information to an "opt-out list". The summary of responses appears in Figure 4.7.

The timing of responses for opt-out requests was appreciably worse than that for delete, with a large fraction of $\mathcal{DBR}$s (30%) exceeding the prescribed 15-business-day limit, as shown in Figure 4.8. Still, half of them answered within 1 business day.

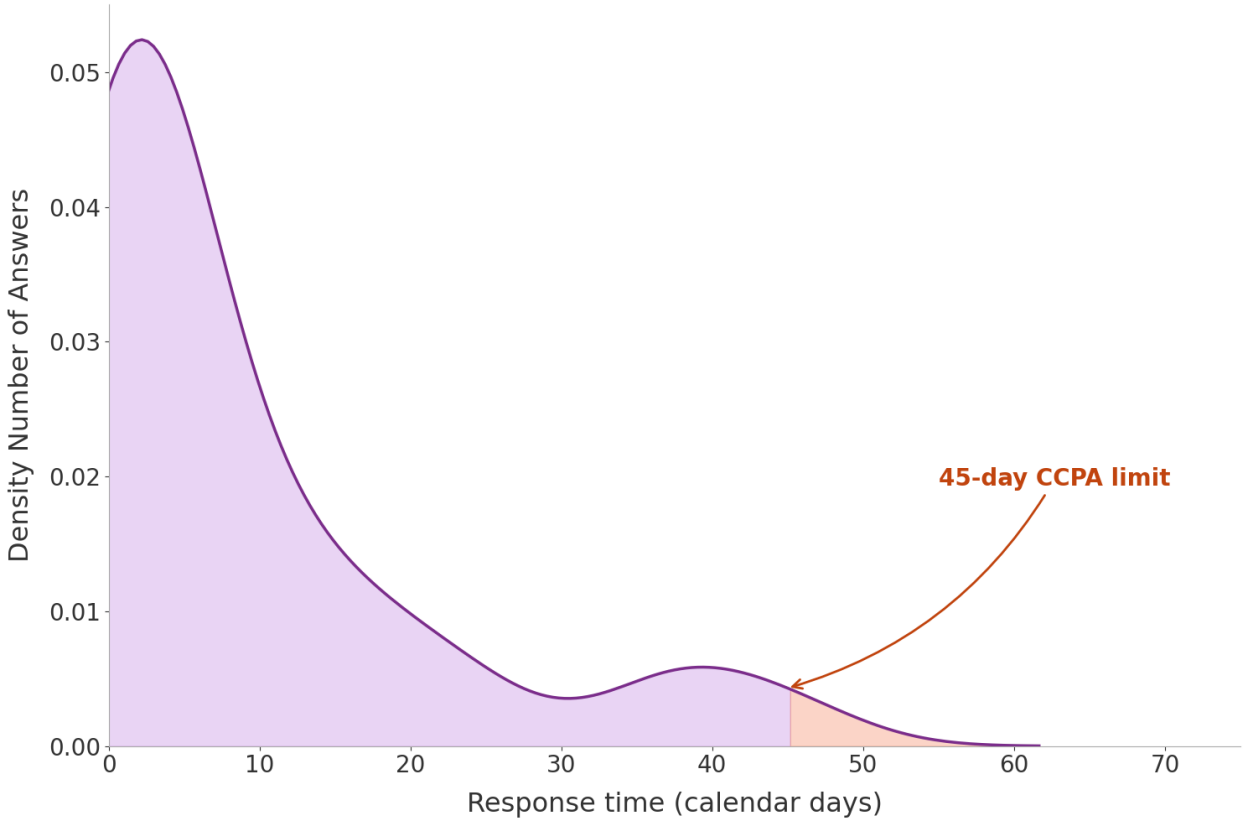


Figure 4.6: Timeline for deletion requests.

4.3.5 Individual Non-Compliance

Aside from the major non-compliance trends described above, we observed the following isolated behaviors.

After submitting opt-out and deletion requests, three $\mathcal{DBR}$s began sending marketing emails to the email addresses corresponding to our synthetic identities. This practice is clearly forbidden by CCPA, which states that any personal information shared during a data request can only be used to fulfill the request, i.e. identification and verification of the consumer [17]. Any further use of the personal information is clearly forbidden.

One $\mathcal{DBR}$, responding to an opt-out request, sent a link by email which led to a simple webpage stating "You have already chosen to opt-in". Neither the email nor the webpage provided any information about actually opting out.

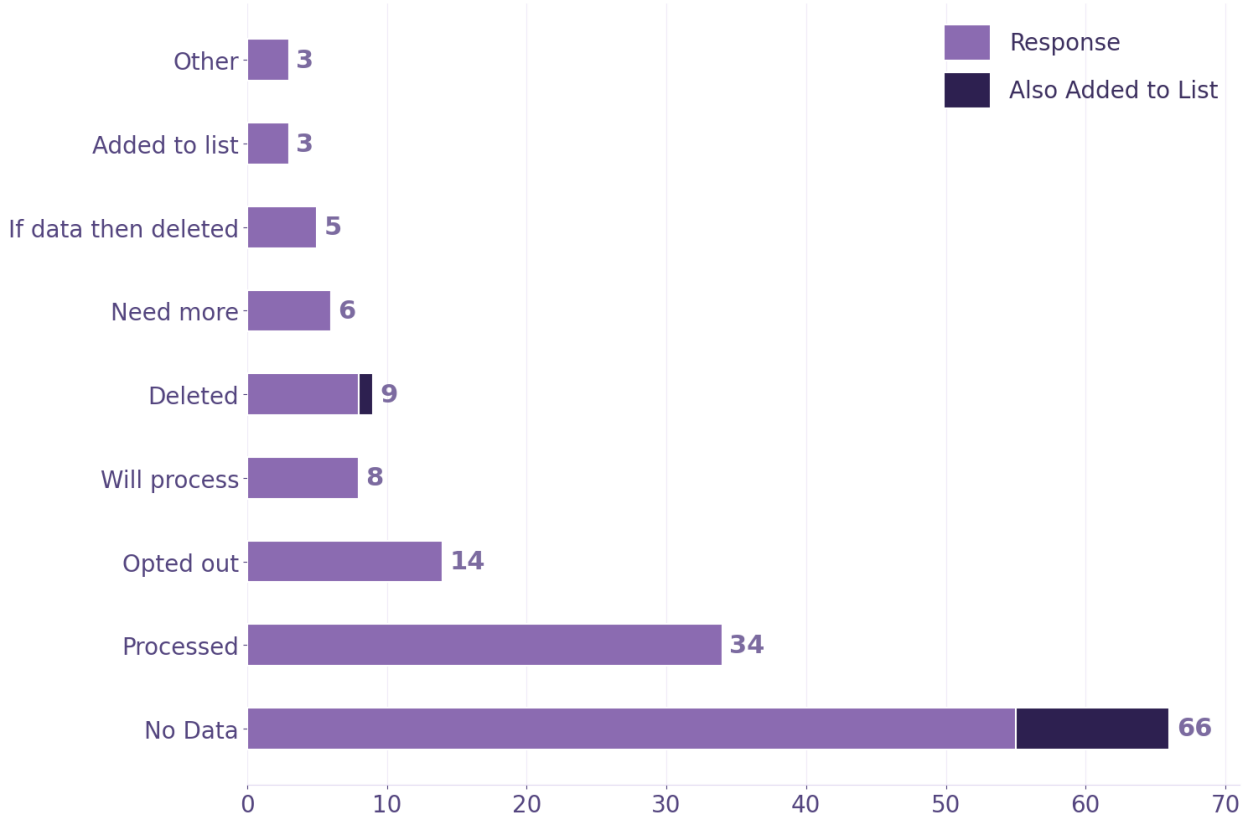


Figure 4.7: Answers for opt-out requests.

Finally, along with its confirmation of data deletion, one $\mathcal{DBR}$ provided (via email) a sample of its consumer records, which (alarmingly) included information about real people.

4.3.6 Time to Submit All Requests

Two researchers (co-authors of this study) conducted all requests submissions: one for each request type. Recall that a total of 358 opt-out, and 322 delete, requests were submitted. Other registered $\mathcal{DBR}$s could not be contacted for several reasons, mentioned earlier.

The submission of all opt-out requests took approximately 20 hours, which corresponds to 3.3 minutes per $\mathcal{DBR}$. In contrast, the deletion request took about 25 hours in total, yielding a rate of 4.6 minutes per $\mathcal{DBR}$. The difference of just over one minute between the two rates is mainly due to two factors: (1) procedures for submitting deletion requests were a bit harder to find on the $\mathcal{DBR}$s' websites (they were often not in the same place where deletion

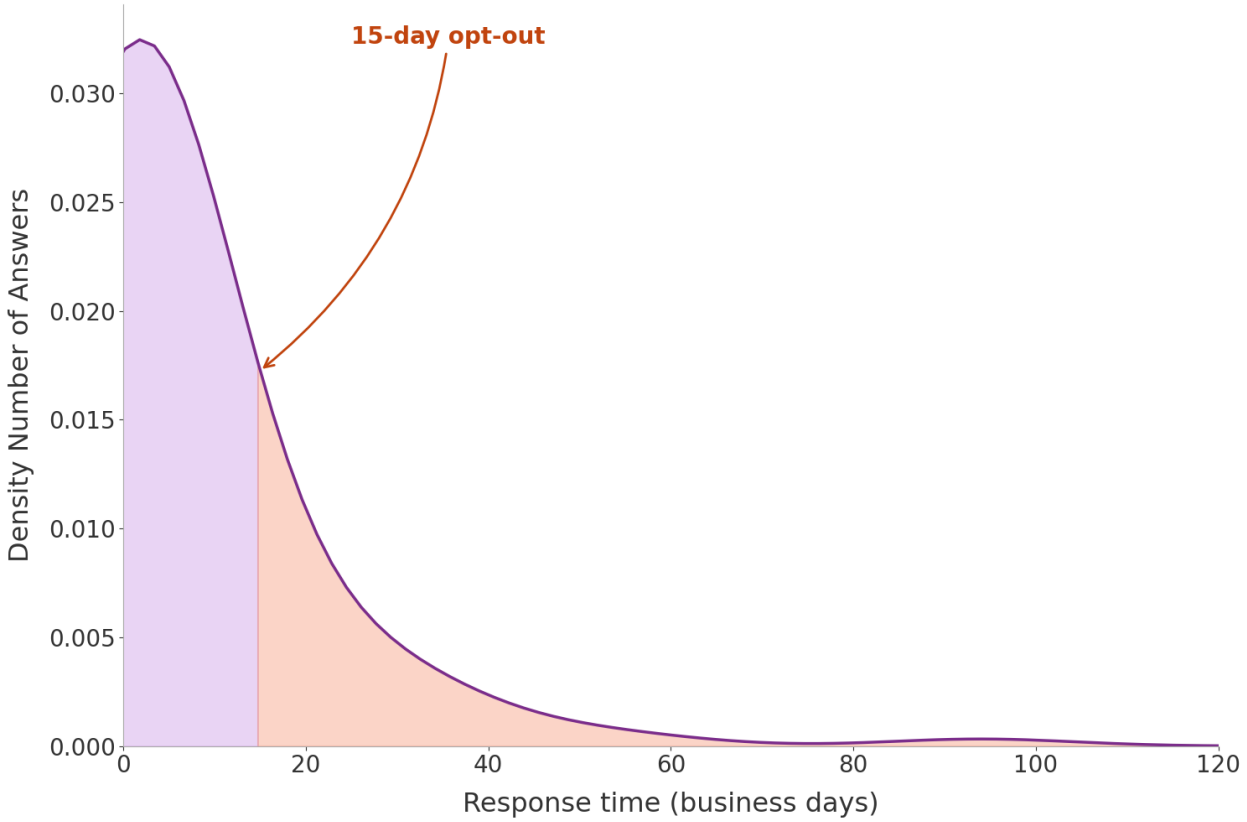


Figure 4.8: Timeline for opt-out requests.

requests procedures were described), and (2) deletion requests required identity verification, while opt-out requests generally did not.

One factor that has likely influenced the overall time consumed to submit requests is the demographic of the researchers who performed this work. Both are Computer Science students in their early- to mid-20s who are naturally quite tech-savvy.

4.3.7 Comparison With a Prior Study of Data Access Requests

A closely related study was recently conducted by Van Kempen et al. [86]. It focused on data access requests submitted under CCPA to all California-registered $\mathcal{DBR}$s in late 2024. At that time, 543 $\mathcal{DBR}$s were registered and requests were made to 454 of those. In contrast, we submitted opt-out and deletion requests to 358 and 322 $\mathcal{DBR}$s, respectively. This represents substantial differences in terms of numbers of contacted $\mathcal{DBR}$s.

Since [86] used a real person's (one of its authors') information to submit data access requests, more $\mathcal{DBR}$s could be contacted. However, our study encountered more limitations since we used synthetic identities and thus could not (or did not want to) fabricate certain information, e.g. cookie values or SSNs. We believe that this accounts for most of the gap in request submission numbers.

Compared to [86] where 48.9% of $\mathcal{DBR}$s provided form submissions, many more $\mathcal{DBR}$s did so in this study: 75% and 80%. We believe that there were two reasons for this:

1. Requests to delete/opt-out are easier to automate than requests to access. Therefore, use of a form streamlines the process for $\mathcal{DBR}$s.
2. Over the past couple of years, more $\mathcal{DBR}$s adopted form submissions for all types of consumer requests.

Beyond the numbers of contacted $\mathcal{DBR}$s, response rates in our study were also lower than that in the prior study of access requests. This is likely because $\mathcal{DBR}$s assume that a form submission confirmation page/message for a delete/opt-out request represents a response, and thus do not separately confirm receipt of a request.

In terms of responses, $\mathcal{DBR}$s followed the same answer timeline pattern in both studies: a vast majority responded within the first few days, and very few $\mathcal{DBR}$s answered late.

4.3.8 Answers to Research Questions

Recall the research questions posed in Section 4.1. We now describe the extent to which the study answered them.

RQ1: What are the differences, if any, in the submission processes of opt-out and delete requests? The former involve less consumer burden and are thus faster to

submit since they require no identity verification. In general, opt-out requests needed less personal information (see Figure 4.4). However, a significant percentage of $\mathcal{DBR}$s still asked to verify consumer identity, thus not complying with CCPA. Also, deletion requests are more final, since $\mathcal{DBR}$s can still keep (previously collected) information about a given consumer after an opt-out request. Within each type of request, we found that most brokers rely on submission forms for opt-out or deletion requests rather than email. These forms require providing a different set of PII to each broker.

RQ2: To what extent do brokers comply with CCPA in terms of delete and opt-out requests? As described in Chapter 2, significant non-compliance was observed. The majority of $\mathcal{DBR}$s did not respond to our deletion requests, and many did not confirm our opt-out requests. Furthermore, potential acts of non-compliance were common. Results are summarized in Table 4.4.

RQ3: What is the overall consumer burden involved in submitting each request type to all $\mathcal{DBR}$s? Since every $\mathcal{DBR}$ has its own delete/opt-out request submission process (see Section 4.3.2), a consumer needs to access every $\mathcal{DBR}$'s webpage and find request submission directions. With well over 300 $\mathcal{DBR}$s, this represents a very heavy burden (in terms of time and effort) for an average consumer. It is also highly repetitive and thus extremely tedious. While California consumers can now use the DROP [20] portal, those in other jurisdictions can either contact each $\mathcal{DBR}$ separately or pay a third-party service to do so on their behalf.

4.4 Limitations

Since we used synthetic identities, we cannot be sure whether $\mathcal{DBR}$s actually complied with our requests. $\mathcal{DBR}$s could have ignored requests if they deemed it fraudulent. In the best case, $\mathcal{DBR}$s should have recorded our opt-out requests even though they should not have found any information in their databases corresponding to the fictitious individuals. In other words, they should have kept "state" to prevent themselves from collecting any information about that individual in the future.

As far as deletion requests, the best outcome would have been an explicit notification by each $\mathcal{DBR}$ about their inability to find any information for the fictitious individual. However, this is a fundamental limitation of our study: although it uncovered various types of non-compliance, it did not evaluate the actual *efficacy* of submitting delete and opt-out requests. Doing that would require using real identities.

4.5 Future Work

The study presented here is by no means the "end of the road." The new California DROP system mandates $\mathcal{DBR}$s to explicitly communicate the status of each request to the requesting consumer. This feature, if treated in good faith (by $\mathcal{DBR}$s) and implemented well by CPPA, would be highly beneficial to consumers. One intuitive direction for future work is to wait some time after August 2026, when $\mathcal{DBR}$s will be mandated to actually participate in the CPPA's DROP portal, and assess its efficacy.

Another direction is to repeat the study presented in this chapter with identities that are neither synthetic nor those of the researchers conducting the study. The main idea would be to evaluate deletion and opt-out (by submitting requests as we did) and then later verify that those requests were acted upon by submitting data access requests. We obviously could not do the latter, since we used non-existent identities. However, using identities of real

people can be problematic, e.g., recruitment and remuneration, as well as potential ethical issues. An alternative is to use identities of dead people, though this is likely even more problematic.

As mentioned in Section 4.3.6, the two researchers who submitted both request types are not representative of the general population, due to being quite young and technologically adept. A more realistic assessment of overall time and effort would use a representative population sample.

Finally, in the long-term, a more frictionless user interface to $\mathcal{DBR}$s should be developed. While the CPPA's DROP portal is a step in the right direction, it is not privacy agile. A consumer must enter lots of personal information into the portal. Also, DROP operates in an asynchronous manner, since $\mathcal{DBR}$s are only required to fetch accumulated (on the portal) delete and opt-out requests with a frequency of at most 45 days. It is not hard to imagine a system that offers more privacy for consumers and operates in a real-time fashion.

4.6 Conclusions

While current regulations, exemplified by CCPA, appear to give consumers more control over their data, exercising one's rights is still a highly laborious and opaque process. As the results of the study described in this chapter show, one must submit many types of forms (and/or emails) and what comes back (if anything at all) is mostly in the form of generic text. Deletion is harder than opt-out and sometimes requires sending a considerable amount of additional personal information to $\mathcal{DBR}$s.

Chapter 5

PIVA: Privacy-Preserving Identity Verification Methods for Accountless Users via Private List Intersection and Variants

Abstract

Several prominent privacy regulations (e.g., CCPA and GDPR) require service providers to let consumers request access to, correct, or delete, their personal data. Compliance necessitates verification of consumer identity. This is not a problem for consumers who already have an account with a service provider since they can authenticate themselves via a successful account log-in. However, there are no such methods for accountless consumers, even though service providers routinely collect data about casual consumers, i.e., those without accounts. Currently, in order to access their collected data, accountless consumers are asked

to provide Personally Identifiable Information (PII) to service providers, which is privacy-invasive.

To address this problem, we propose PIVA*: Privacy-Preserving Identity Verification for Accountless Users*, a technique based on *Private List Intersection (PLI)* and its variants. First, we introduce PLI, a close relative of *private set intersection (PSI)*, a well-known cryptographic primitive that allows two or more mutually suspicious parties to compute the intersection of their private input sets. PLI takes advantage of the (ordered and fixed) list structure of each party's private set. As a result, PLI is more efficient than PSI. We also explore PLI variants: PLI-cardinality (PLI-CA), threshold-PLI (t-PLI), and threshold-PLI-cardinality (t-PLI-CA), all of which yield less information than PLI. These variants are progressively better suited for addressing the accountless consumer authentication problem.

We prototype PIVA and compare its performance against techniques based on regular PSI and garbled circuits (GCs). Results show that proposed PLI and PLI-CA constructions are more efficient than GC-based techniques, in terms of both computation and communication overheads. While GC-based t-PLI and t-PLI-CA execute faster, proposed constructs greatly outperform the former in terms of bandwidth, e.g., our t-PLI protocol consumes 16× less bandwidth. We also show that proposed protocols can be made secure against malicious adversaries, with only moderate increases in overhead. These variants outperform their GC-based counterparts by at least one order of magnitude.

5.1 Introduction

Notable privacy regulations – including the European Union General Data Protection Regulation (GDPR) [37], California Consumer Privacy Act (CCPA) [17], and Brazilian Lei Geral

de Proteção de Dados (LGPD) [45] – mandate that service providers which collect any consumer-specific data must grant consumers access to that data [1]. To allow consumers to access, correct, or delete their personal data, a service provider must employ secure identity verification methods.

For instance, CCPA [17] introduces the notion of a Verifiable Consumer Request (VCR) which is defined as "a request that is made by a consumer [...], and that the business can verify, using commercially reasonable methods, pursuant to regulations adopted by the Attorney General pursuant to paragraph (7) of subdivision (a) of Section 1798.185 to be the consumer about whom the business has collected personal information." (CCPA §1798.140 (ak)). CCPA also states that "the business may require authentication of the consumer that is reasonable in light of the nature of the personal information requested" (CCPA §1798.130 (a)(2)(A)).

For a consumer who has an account with a service provider, authentication translates into account log-in via, e.g., username and password, and two-factor authentication, such as email or SMS verification. In fact, GDPR guidelines [36] state that a user log-in, using a password, "should be sufficient to authenticate a data subject". CCPA also mentions that, for consumers with accounts, "the business may require the consumer to use that account to submit a verifiable consumer request" [17].

On the other hand, for *account-less* users (i.e., those without accounts with a service provider) authentication options are limited. Without an account, and hence without any shared secrets (such as passwords), service providers tend to ask consumers to reveal some amount of Personally Identifiable Information (PII) in the form of: (1) simply entering it [59, 58, 72, 12, 14], (2) providing cookies from prior interactions [82, 83], (3) showing government-

[1] While GDPR and LGPD apply to any entity processing residents' data in their respective regions, CCPA only applies to for-profit businesses that have a gross revenue above $25 million, handle the data of more than 100,000 California residents, or generate 50% or more of their revenue by selling California residents' PII (in 1798.140 (d)) [17].

issued IDs [59, 58, 72, 12, 82, 83, 14], or (4) producing notarized documents [72, 82, 83, 14]. A recent user study [77] that studies user experience with removing PII from the Internet shows that current removal procedures are not thorough, transparent, or responsive. This naturally prompts user concerns about service providers acquiring new PII in the nebulous process of authenticating accountless users.

Motivated by the above, this chapter proposes PIVA*: Privacy-Preserving Identity Verification for Accountless Users*. PIVA involves two parties: an accountless user $\mathcal{C}$ who possesses their PII as an ordered/labeled list and a service provider $\mathcal{S}$ that holds a collection of such PII (ordered/labeled lists) for each accountless user. It allows $\mathcal{S}$ to authenticate $\mathcal{C}$ if sufficiently many list elements are matched as the intersection of their respective PII lists. Meanwhile, $\mathcal{S}$ is prevented from learning any new $\mathcal{C}$ PII and vice-versa.

As a basic building block, we use *Private List Intersection (PLI)*, a variant of a well-known cryptographic primitive called *private set intersection (PSI)*. While PSI allows two parties to jointly compute the intersection of their private *sets*, PLI computes the intersection of the private *lists*, i.e., it takes list order into account. We also consider PLI variants that reveal strictly less information than PLI: (1) *PLI-Cardinality (PLI-CA)*, which outputs only the cardinality of the intersection list; (2) *threshold-PLI (t-PLI)*, which outputs the intersection only if its cardinality exceeds a threshold t; and (3) *threshold-PLI-Cardinality (t-PLI-CA)*, which outputs the intersection cardinality only if it exceeds a threshold t. We implement these protocols and evaluate their efficiency. All variants outperform the GC-based implementation in terms of bandwidth, and the PLI and PLI-CA execute faster. The entire implementation is publicly available at [2].

Finally, we propose a modified version of each protocol that achieves security against malicious (as opposed to HbC: Honest-but-Curious) adversaries. These modifications result in relatively low additional communication and computation costs. In particular, the asymptotic overhead of each modified protocol remains the same. Notably, our work yields the

first threshold-based PSI and PSI-CA protocols secure in the malicious model, which might be of independent interest.

Organization: Background is given in Chapter 2. After reviewing related work in Section 5.2, Section 5.3 presents the preliminary background, and Section 5.4 describes system and threat models. Next, we present PIVA and PLI variants in Section 5.5; their security analyses appear in Section 5.5.4. We describe the implementation and evaluation in Section 5.6. Finally, variants of proposed protocols secure in the malicious model are presented in Section 5.7.

5.2 Related Work: Private Set Intersection and Its Variants

Private Set Intersection (PSI) protocols, e.g. [71, 48, 70, 67], allow two parties to compute the intersection $X \cap Y$ of their private input sets X and Y, without revealing any other information. Works on PSI include PSI-cardinality (PSI-CA), e.g. [32, 35, 81], which reveals only cardinality $|X \cap Y|$ of the intersection, and hides the intersecting elements, and threshold-PSI (t-PSI), e.g. [93, 65, 43], which reveals the intersection only if $|X \cap Y| \geq t$ for some fixed threshold t.

Table 5.1: Comparison of the computational complexity of PLI and PSI variants as a function of n, the size of input lists or sets.[2]

	Comp. Cost	**Active Security**
PSI [71, 48, 70, 67]	$O(n)$	✓
PLI (ours)	$O(n)$	✓
t-PSI [93]	$O(n)$	✗
t-PSI [65]	$O(n^2)$	✗
t-PSI [43]	$O(t)$	✗
t-PLI (ours)	$O(n^3)$	✓

	Comp. Cost	**Active Security**
PSI-CA [32, 35]	$O(n)$	✗
PSI-CA [81]	$O(n \log n)$	✗
PLI-CA (ours)	$O(n)$	✓
t-PLI-CA (ours)	$O(n^3)$	✓

[2]To the best of our knowledge, there is no prior work on t-PSI-CA. [92] presents a protocol they call "t-PSI-CA", but it reveals $X \cap Y$ if $|X \cap Y| \geq t$, and $|X \cap Y|$ otherwise, whereas our t-PLI-CA reveals

Literature on PSI is very rich, but instead of citing all work in PSI literature, we summarize state-of-the-art PSI results and compare them to our PLI protocols in Table 5.1. Asymptotically, our PLI and PLI-CA protocols match the $O(n)$ computation cost of prior PSI and PSI-CA protocols, while our t-PLI has $O(n^3)$ cost which is higher than t-PSI of [93, 65, 43]. However, all our protocols achieve *active security*, i.e. security against malicious participants, at the same asymptotic costs (and a small increase to concrete costs), whereas $O(n)$ actively secure PSI is only known as the basic PSI variant, and not for PSI-CA or t-PSI. Actively secure (t-)PLI(-CA) can be implemented using generic constant-round actively secure computation using "cut-and-choose" over garbled circuits (see e.g. [53]), but the "cut-and-choose" technique increases protocol costs by two orders of magnitude, rendering them impractical.

5.3 Preliminaries

We use the notation $[n]$ to represent the set $\{1, 2, \ldots, n\}$. Where applicable, we denote by $x_{i,1}$ and $x_{i,2}$ the first and second elements of a tuple x_i with two elements, for $\forall i \in [n]$, i.e., $x_i := (x_{i,1}, x_{i,2})$, where $i = 1, ..., n$.

Additively Homomorphic ElGamal Encryption. Below, we define the additively homomorphic variant of the ElGamal encryption scheme [30]. Compared to the original ElGamal [41] which is multiplicatively homomorphic, this variant uses g^m in place of the plaintext m in encryption, where g is a generator of a group G of prime order q.

Definition 5.1. *Additively homomorphic ElGamal encryption [30] is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{PreDec}, \mathsf{Test})$*, where:*

- $\mathsf{Setup}(1^\lambda) \to (G, q, g)$*: for security parameter* λ*, it outputs parameters* (G, q, g)*, where* g *is a generator of group* G *with prime order* q *s.t.* $|q| \geq 2\lambda$.

$|X \cap Y|$ if $|X \cap Y| \geq t$ and $\perp$ otherwise.

- $\mathsf{KeyGen}(G, q, g) \rightarrow (sk, pk)$: *outputs a pair of secret and public keys, where* $sk \leftarrow_{\$} Z_q$ *and* $pk = (G, q, g, h)$ *for* $h = g^{sk}$.

- $\mathsf{Enc}_{pk}(m) \rightarrow c$: *On message* $m \in Z_q$ *and public key* $pk = (G, q, g, h)$, *it outputs ciphertext* $c := (g^w, h^w g^m)$, *where* $w \leftarrow_{\$} Z_q$.

  (We write $\mathsf{Enc}_{pk}(m; w)$ *to denote randomness* w *as an explicit input.)*

- $\mathsf{PreDec}_{sk}(c) \rightarrow M$: *On secret key* sk *and ciphertext* $c = (c_1, c_2)$, *output* $M := c_2(c_1)^{-sk}$. *(Note that if* $(c_1, c_2) = \mathsf{Enc}_{pk}(m)$ *then* $M = g^m$.*)*

- $\mathsf{Test}(M, m)$: *Decide if the pre-decryption value* M *corresponds to plaintext target* m *by outputting* 1 *if* $M = g^m$ *and* 0 *otherwise.*

Note that this version of ElGamal does not allow for efficient decryption, but it allows for efficient testing if a ciphertext encrypts a given plaintext m. In our PLI protocol and its (t-)PLI(-CA) variants, the decryptor will only test whether the ciphertext encrypts $m = 0$, by checking if $c_2(c_1)^{-sk} = 1$.

Shamir Secret Sharing. Shamir secret sharing [74] is a t-out-of-n threshold secret sharing scheme based on polynomial interpolation over finite fields, with a threshold t. i.e., When a secret s is shared using Shamir secret sharing, s can be reconstructed only with $\geq t$ shares, while the possession of $< t$ shares reveals no information about s.

Definition 5.2. *Shamir secret sharing scheme [74] over field* $\mathbb{F}$ *is defined by algorithms* $\mathsf{Share}, \mathsf{Reconstruct}$ *s.t.*

- $\mathsf{Share}_{(t,n)}(s) \rightarrow (s_1, ..., s_n)$: *generates* n *shares of a secret* $s \in \mathbb{F}$, *assuming* $n < |F|$. *It chooses* $(t-1)$ *random coefficients* $a_i \leftarrow_{\$} \mathbb{F}$, *for* $i \in [t-1]$, *defines* $(t-1)$*-degree polynomial* $f(x) = s + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1}$ *over* $\mathbb{F}$, *and sets* $s_1 := f(1), s_2 := f(2), ..., s_n := f(n)$.

- $\mathsf{Reconstruct}(s_{i_1}, ..., s_{i_m}) \rightarrow s$: *reconstructs a secret from* $m \geq t$ *shares* $s_{i_1}, ..., s_{i_m}$. *It interpolates any* t *out of these* m *shares into a* $(t-1)$*-degree polynomial* f' *and outputs* $s' := f'(0)$ *as the reconstructed secret.*

Note that $\mathsf{Reconstruct}$ guarantees to output $s' = s$ if the m shares it receives as input were generated by $\mathsf{Share}(s)$.

Reed-Solomon Codes and Berlekamp-Welch Decoding. Shamir secret sharing scheme can be seen as a special case of Reed-Solomon (RS) coding scheme [69], where the number of errors is zero. (n, t)-RS codes are a group of error correction codes, where its encoder adds some parity bits to encode t messages into n messages (called *symbols* in coding theory) so that its decoder can correct up to $\frac{(n-t)}{2}$ errors. Specifically, t messages are assigned as the coefficients of a polynomial p of degree $t-1$ over a finite field $\mathbb{F}$, and the encoding of t messages is the evaluation of $p(x)$ over some n evaluation points, $x_1, ..., x_n$, i.e., $\mathsf{Encode}(m_1, ..., m_t) = (p(x_1), ..., p(x_n))$, where $p(x) := m_1 + m_2 x + ..., m_t x^{t-1}$. For decoding a codeword $c := (c_1, ..., c_n)$ with n symbols, the original method interpolates polynomials over all possible t points over n points, $(x_1, c_1), ..., (x_n, c_n)$, and outputs the t coefficients of the majority polynomial p.

The Berlekamp-Welch (BW) algorithm [11] is an efficient decoding algorithm that corrects up to $\lfloor \frac{n-t}{2} \rfloor$ errors in RS codes. It not only recovers the polynomial p over $\mathbb{F}$ which represents the original messages, but also outputs the error locator polynomial err by solving n linear equations, where the x-intercepts of err represent the locations where the errors occurred among n symbols. We denote this by $\mathsf{BW}_{(k,n)}(c_1, c_2, ..., c_n) \rightarrow (p, err)$, where $k := \lceil \frac{n+t}{2} \rceil$, the minimum number of correct symbols in the input codeword. We use this algorithm to efficiently reconstruct the secret shared by the Shamir secret sharing, where the input shares potentially include some invalid shares, i.e., errors.

5.4 System & Threat Models

The system model includes two mutually distrusting parties: a client ($\mathcal{C}$) and a server ($\mathcal{S}$), where $\mathcal{S}$ initiates the protocol to verify $\mathcal{C}$'s identity and grants access to $\mathcal{C}$ once the verification succeeds. Each party has a list of PII of size n, $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$, respectively. We assume that the order of attributes in lists is public and agreed between parties in advance.

Considering that $\mathcal{S}$ may have a large database of clients, we assume that $\mathcal{S}$ locates $\mathcal{C}$ in its database using a unique public string, e.g. in the context of people search websites, using the URL of the webpage about $\mathcal{C}$.

In Section 5.5, we consider a static *Honest-but-Curious* (HbC) adversary who corrupts either $\mathcal{C}$ or $\mathcal{S}$ before the protocol starts, and aims to learn maximal information about the other party's private input, but otherwise follows the protocol. In Section 5.7, we consider *active*, a.k.a. *malicious* adversaries, i.e. we assume that an adversary who corrupts either party can run an arbitrary protocol in place of the prescribed one, and we show that our protocols realize the intended ideal functionality even in the presence of malicious adversaries.

5.5 PIVA Design

In this section, we present four protocol variants for PIVA, i.e. PLI, PLI-CA, t-PLI, and t-PLI-CA. We describe each variant below, summarizing their security properties with ideal functionalities described in Fig. 5.3 in Subsection 5.5.3. Security proofs are shown in Section 5.5.4.

5.5.1 Private List Intersection (PLI) and PLI-Cardinality (PLI-CA)

PLI: Recall that (one-sided) PSI takes private *sets* as input and outputs their intersection set to one of the parties. PLI takes private *lists* as input instead. i.e. the inputs are two lists, X and Y, of size n, contributed by $\mathcal{C}$ and $\mathcal{S}$, respectively. Entries in list X and Y are arbitrary bitstrings or a special symbol $\perp$ which designates that the corresponding party does not have any value at this position.

The protocol computes and sends to $\mathcal{S}$ the list intersection $X \cap Y$, defined as the set of indexes where the two lists contain the same values, i.e.

$$X \cap Y \;=\; \{\, i \;\mid\; X[i] = Y[i] \;\wedge\; X[i] \neq \perp \,\}$$

(See also the Ideal Functionality for PLI in Fig. 5.3 in Section 5.5.3.) In the context of our application, each position in list X or Y represents a specific type of PII, e.g., home address, driver's license number, mother's maiden name, etc, and we assume that the ordered list of these n attributes is fixed and public.

After obtaining the intersection, $\mathcal{S}$ decides whether the $\mathcal{C}$-provided PII is sufficient to verify their identity corresponding to the PII held by $\mathcal{S}$. As part of PLI, $\mathcal{S}$ learns no new PII about $\mathcal{C}$. This follows GDPR guidelines [36] stating that information requested by a service provider for verification purposes:

> "must be proportionate to the type of data processed, the damage that could occur, etc. in order to avoid excessive data collection",

We construct a PLI protocol using the additively homomorphic ElGamal variant of Section 5.3. We assume that values contained in input lists X and Y are integers in Z_q. Since $q \geq 2\lambda$ where λ is the security parameter, longer values can be hashed into Z_q using a

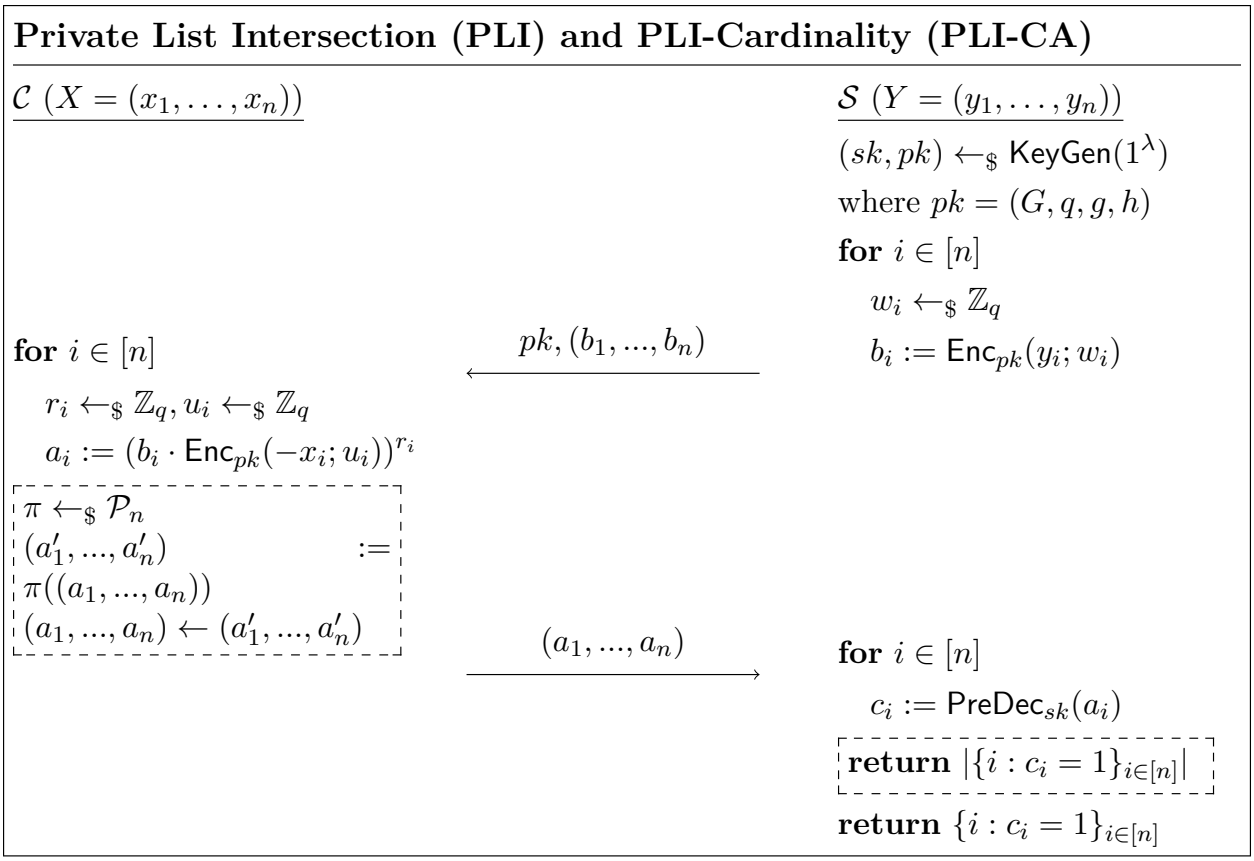


Figure 5.1: PLI and PLI-CA protocols based on ElGamal in Def. 5.1, where $\cdot$ in a_i computation is the element-wise product, i.e., $x \cdot y = (x_1 y_1, x_2 y_2)$, for some tuples $x = (x_1, x_2)$ and $y = (y_1, y_2)$, and $\mathcal{P}_n$ is the set of all pseudorandom permutations of size n. PLI-CA protocol includes the steps in the dashed box, while PLI protocol does not.

collision-resistant hash. If $X[i]$, resp. $Y[i]$, is an empty symbol $\perp$, the corresponding party replaces it with a random value in Z_q. (Using a random value at some index prevents either accidental or malicious attempt at creating a match at that index, except for negligible probability $1/q$.)

The PLI protocol is shown in Fig. 5.1, and it works as follows. $\mathcal{S}$ first generates a private-public key pair (sk, pk) for ElGamal encryption, and encrypts each element y_i under the public key. i.e., $b_i := \mathsf{Enc}_{pk}(y_i; w_i) = (g^{w_i}, h^{w_i} g^{y_i})$, where w_i is chosen at random in $\mathbb{Z}_q$, for all $i \in [n]$. Then, $\mathcal{S}$ sends its public key, $pk = (G, q, g, h)$, and the ciphertexts of its input elements, $(b_1, ..., b_n)$, to $\mathcal{C}$.

Upon receipt, $\mathcal{C}$ encrypts the additive inverse of each element x_i in its list, using pk. i.e., $\mathsf{Enc}_{pk}(-x_i; u_i) = (g^{u_i}, h^{u_i} g^{-x_i})$, where u_i is chosen at random in $[q-1]$, for all $i \in [n]$. Then, $\mathcal{C}$ multiplies the resulting values by b_i's in the list order and re-randomizes them with a random r_i to hide its input values. i.e., For a randomly chosen $r_i \in G$, $\mathcal{C}$ computes $a_i := ((b_{i,1} \cdot g^{u_i})^{r_i}, (b_{i,2} \cdot h^{u_i} g^{-x_i})^{r_i})$ for all $i \in [n]$. Lastly, $\mathcal{C}$ sends $(a_1, ..., a_n)$ to $\mathcal{S}$ in order.

Now, $\mathcal{S}$ partially decrypts the received ciphertexts to see if they are encryptions of zero. i.e., $\mathcal{S}$ computes $c_i := \mathsf{PreDec}_{sk}(a_i) = \frac{a_{i,2}}{a_{i,1}^{sk}}$, and checks if it is equal to 1 or not, for each a_i. Finally, $\mathcal{S}$ outputs all i's s.t. $c_i = 1$, which is the intersection $X \cap Y$. This is because a_i is the encryption of $(y_i - x_i)r_i$ as follows:

$$
\begin{aligned}
a_i &= ((b_{i,1} \cdot g^{u_i})^{r_i}, (b_{i,2} \cdot h^{u_i} g^{-x_i})^{r_i}) \\
&= ((g^{w_i} \cdot g^{u_i})^{r_i}, (h^{w_i} g^{y_i} \cdot h^{u_i} g^{-x_i})^{r_i}) \\
&= (g^{(w_i+u_i)r_i}, h^{(w_i+u_i)r_i} g^{(y_i-x_i)r_i}) \\
&= \mathsf{Enc}_{pk}((y_i - x_i)r_i; R_i), \text{ where } R_i := (w_i + u_i)r_i.
\end{aligned}
$$

Consequently, only if $y_i = x_i$, $\mathcal{S}$ gets $c_i = g^{(y_i-x_i)r_i} = g^0 = 1$. Otherwise, it gets a random c_i and learns nothing about x_i's. We provide the proof of the following theorem in Section 5.5.4. The PLI protocol is similar to the first protocol in [61].

Theorem 5.1. *The protocol presented in Fig. 5.1 securely computes the list intersection in the presence of HbC adversaries under the decisional Diffie-Hellman (DDH) problem. It requires $O(n)$ communication and computation costs for both sides, where n is the input list size.*

PLI-CA: PLI reveals to $\mathcal{S}$ which list elements match and, hence, which do not. $\mathcal{S}$ might use this information to confirm whether the PII previously collected for a given $\mathcal{C}$ is accurate. To prevent this, we suggest using PLI-cardinality (PLI-CA), which outputs only the cardinality

of the intersection, i.e., it reveals only the number of intersecting elements, while keeping their locations private. Using PLI-CA for PIVA is suitable because it suffices for the service provider ($\mathcal{S}$) to learn how many elements are in the intersection to decide whether this is enough to verify the identity of a consumer ($\mathcal{C}$).

We build a PLI-CA protocol (in Fig. 5.1 with the dashed box) atop the PLI protocol. To hide the locations of matching elements, $\mathcal{C}$ randomly shuffles the computation results using a pseudorandom permutation π and sends the permuted results. $\mathcal{S}$ now outputs the number of elements where $c_i = 1$. Because of the pseudorandom permutation, indices where $c_i = 1$ do not reveal the real locations of intersecting elements to $\mathcal{S}$. Similar to PLI, we derive the following theorem, and present the proof in Section 5.5.4.

Theorem 5.2. *The protocol presented in Fig. 5.1 including the procedure of the dashed box securely computes the cardinality of list intersection in the presence of HbC adversaries under the DDH problem. It requires $O(n)$ communication and computation costs for both sides, where n is the input list size.*

5.5.2 Threshold PLI (t-PLI) and t-PLI-Cardinality (t-PLI-CA)

While PLI and PLI-CA output their results to $\mathcal{S}$ no matter what, threshold PLI (t-PLI) and t-PLI-Cardinality (t-PLI-CA) output the intersection to $\mathcal{S}$ only if the number of common data exceeds some agreed-upon (fixed a priori) threshold t, i.e., $|X \cap Y| \geq t$. Therefore, it does not reveal anything to $\mathcal{S}$ if either party does not have sufficient PII.

Note that the threshold t can be determined by either agreement between parties, or by some legal standard. For instance, CCPA considers this, mentioning the "degree of certainty" required for identity verification. It sets "reasonable degree of certainty" as matching at least two pieces of information and "reasonably high degree of certainty" as matching at least three pieces of information and obtaining a signed declaration, under penalty of perjury [16].

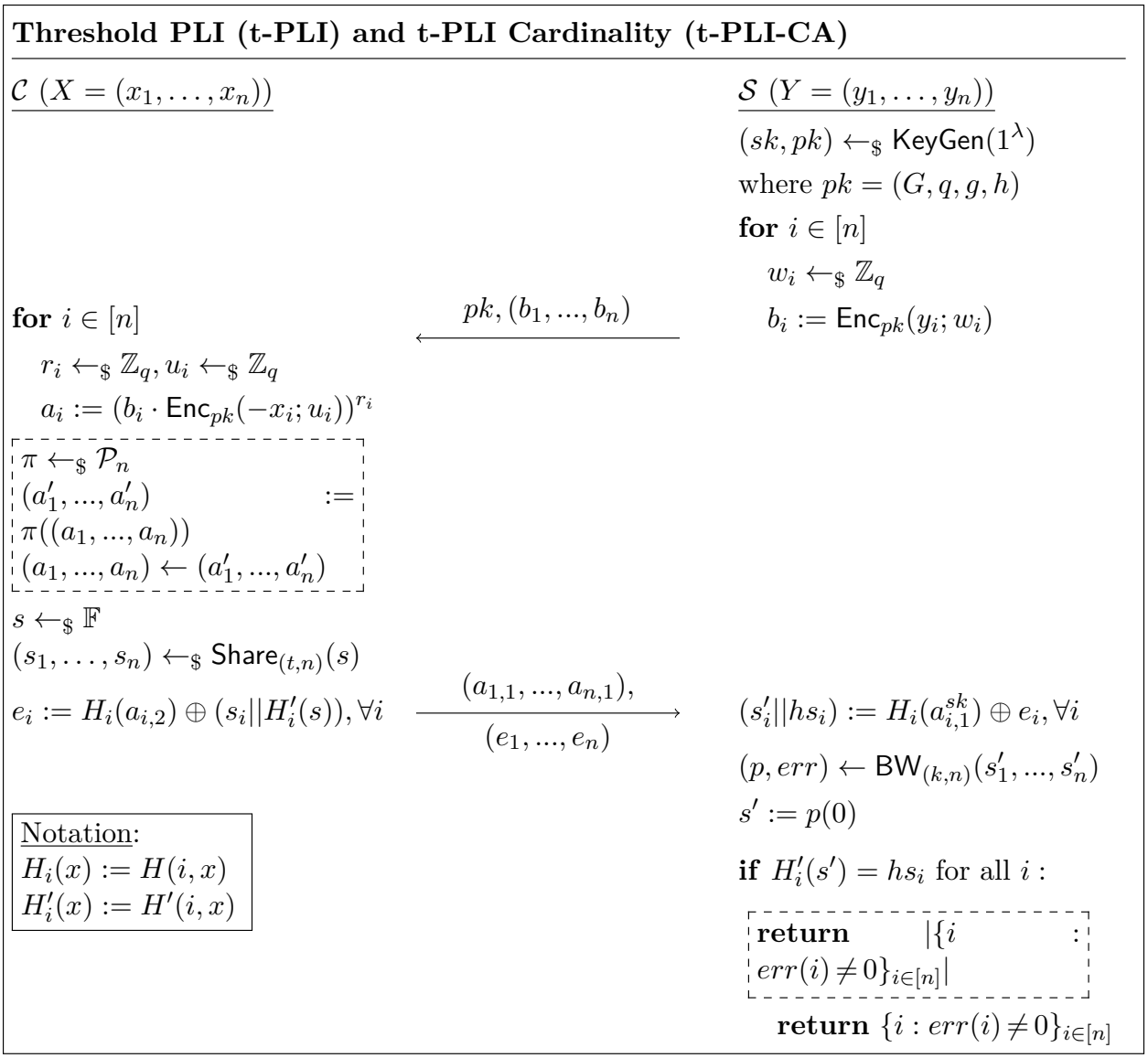


Figure 5.2: t-PLI and t-PLI-CA protocols using Shamir secret sharing (Def. 5.2) and Berlekamp-Welch decoder BW (Section 5.3). Here $\oplus$ denotes XOR, $||$ denotes string concatenation, and $\mathcal{P}_n$ is the set of all permutations of size n. t-PLI-CA protocol includes the steps in the dashed box, while t-PLI protocol does not.

t-PLI: We add t-out-of-n Shamir Secret Sharing to the PLI protocol in Section 5.5.1, i.e., after computing a_i's, $\mathcal{C}$ randomly chooses a secret value s and runs the Share algorithm, see Definition 5.2, to generate sharing $(s_1, ..., s_n)$ of s. Then, $\mathcal{C}$ xor's each i-th share with the hash of the second element of a_i's, such that $e_i := H(i, a_{i,2}) \oplus (s_i || H'(i, s))$, for each i, where H, H' are Random Oracle hash functions where $H : \{0,1\}^* \to \{0,1\}^{|\mathbb{F}|+2\lambda}$ and

$H' : \{0,1\}^* \to \{0,1\}^{2\lambda}$, where $|\mathbb{F}|$ is the size of bitstrings encoding elements of field $\mathbb{F}$ used in Shamir Secret Sharing. We denote $H(i,x)$ and $H'(i,x)$ by $H_i(x)$ and $H'_i(x)$, respectively. Lastly, $\mathcal{C}$ sends the first component of a_i's, $(a_{i,1})_i$, and e_i's to $\mathcal{S}$.

After receiving it, $\mathcal{S}$ computes $H_i(a_{i,1}^{sk})$ with its private key sk and XORs to each e_i for each i. Then, it gets the shares and hash values by dividing the computed strings, i.e., $(s'_i||hs_i) := H_i(a_{i,1}^{sk}) \oplus e_i$. Then, $\mathcal{S}$ derives s' using the Reconstruct algorithm, or the Berlekamp-Welch algorithm BW for better performance, with $(s'_1, ..., s'_n)$ as input shares. Finally, if the hash of s' with the position i matches the received hash value for all i, $\mathcal{S}$ outputs its elements with the indices of correct shares. Fig. 5.2 shows the protocol above.

Note that s'_i is the same as the original share s_i if $x_i = y_i$. Therefore, if at least $k := \lceil \frac{n+t}{2} \rceil$ obtained s'_i are correct, $\mathcal{S}$ can use BW algorithm to efficiently recover the original polynomial p, and the error locating polynomial err. However, even if between t and $k-1$ shares are correct, $\mathcal{S}$ can still recover p and locate errors by examining all the possible subsets of size t of the set $\{s'_i\}_i$ of size n. As a result, the following theorem can be derived, with a proof given in Section 5.5.4.

Theorem 5.3. *The protocol presented in Fig. 5.2 securely computes the list intersection only when its cardinality exceeds the threshold t in the presence of HbC adversaries under the DDH problem. It requires $O(n)$ communication bits, and $O(tn)$ $\mathcal{C}$-side computation, where n is the input list size and t is the threshold for outputting the result. $\mathcal{S}$-side requires $O(n^3)$ computation if the cardinality of intersection is greater than or equal to $\lceil \frac{n+t}{2} \rceil$, or $O(C(n,t))$ computation, otherwise, where $C(n,t)$ is the number of t-combinations over n elements.*

t-PLI-CA: We combine t-PLI and PLI-CA to further enhance privacy. At the end of t-PLI-CA, $\mathcal{S}$ learns only the cardinality of the intersection, and only if it exceeds threshold t, i.e., $\mathcal{S}$ outputs $|X \cap Y|$ only if $|X \cap Y| \geq t$.

Similarly to PLI-CA built atop the PLI construction, we construct t-PLI-CA by adding a random shuffling to the t-PLI construction. i.e., To prevent $\mathcal{S}$ from learning the indices of the matching elements, given by the error locating polynomial, $\mathcal{C}$ randomly permutes $(a_1, ..., a_n)$. As a result, $\mathcal{S}$ outputs the subtraction of the number of x-intercepts of the error locating polynomial *err* from the list size n. Recall that $\mathcal{S}$ can find the correct s' only when the number of matching elements exceeds the threshold, and the x-intercepts of *err* represent the wrong shares. Thus, the output means the cardinality of the intersection of the private lists. The detailed protocol is given in Fig. 5.2 with the dashed box, and the security proof of the following theorem is presented in Section 5.5.4.

Theorem 5.4. *The protocol presented in Fig. 5.2 including the procedure of the dashed box securely computes the cardinality of list intersection only when it exceeds the threshold t in the presence of HbC adversaries under the DDH problem. It requires $O(n)$ communication bits, and $O(tn)$ $\mathcal{C}$-side computation, where n is the input list size and t is the threshold for outputting the result. $\mathcal{S}$-side requires $O(n^3)$ computation if the cardinality of intersection is greater than or equal to $\lceil \frac{n+t}{2} \rceil$, or $O(C(n,t))$ computation, otherwise, where $C(n,t)$ is the number of t-combinations over n elements.*

5.5.3 Ideal Functionalities for PLI and its variants

Parameters: List size n
Ideal Functionality of (a) PLI, (b) PLI-CA, (c) t-PLI, (d) t-PLI-CA:
1. Wait for inputs $Y = (y_1, y_2, ..., y_n)$ from $\mathcal{S}$ and $X = (x_1, x_2, ..., x_n)$ from $\mathcal{C}$
2. Compute (a),(c): the list intersection $\mathsf{out}_{\mathcal{S}} = X \cap Y$
   (b),(d): the cardinality of list intersection $\mathsf{out}_{\mathcal{S}} = |X \cap Y|$
3. Send $\perp$ to $\mathcal{C}$ and (a),(b): send $\mathsf{out}_{\mathcal{S}}$ to $\mathcal{S}$
   (c),(d): if $|X \cap Y| \geq t$ send $\mathsf{out}_{\mathcal{S}}$ to $\mathcal{S}$ else send $\perp$ to $\mathcal{S}$

Figure 5.3: Ideal Functionality of PLI and its (t-)PLI(-CA) variants

Fig. 5.3 shows the ideal functionality of each PLI variant introduced above, i.e., private list intersection (PLI), PLI-Cardinality (PLI-CA), threshold PLI (t-PLI), and t-PLI-Cardinality (t-PLI-CA). Each functionality takes as input lists X and Y of size n from $\mathcal{C}$ and $\mathcal{S}$ re-

spectively, and computes either the intersection (PLI and t-PLI) or the cardinality of the intersection (PLI-CA and t-PLI-CA) of the input lists, and this output is revealed to $\mathcal{S}$ party either unconditionally (PLI and PLI-CA) or only if the intersection size is above t (t-PLI and t-PLI-CA).

5.5.4 Security Proofs for PIVA Protocols

Proof of Theorem 5.1 (PLI security):

Correctness: Assume an execution of the protocol Π with honest $\mathcal{C}$ and honest $\mathcal{S}$. $\mathcal{S}$ computes $c_i := \frac{a_{i,2}}{a_{i,1}^{sk}}$, which is equivalent to $(g^{y_i - x_i})^{r_i}$ as follows:

$$c_i := \frac{a_{i,2}}{a_{i,1}^{sk}} = \frac{(b_{i,2} \cdot h^{u_i} g^{-x_i})^{r_i}}{((b_{i,1} \cdot g^{u_i})^{r_i})^{sk}} = \frac{(h^{w_i} g^{y_i} h^{u_i} g^{-x_i})^{r_i}}{((g^{w_i})^{sk} (g^{u_i})^{sk})^{r_i}}$$
$$= \frac{(h^{w_i} h^{u_i} g^{y_i - x_i})^{r_i}}{(h^{w_i} h^{u_i})^{r_i}} = \frac{(h^{w_i} h^{u_i})^{r_i} (g^{y_i - x_i})^{r_i}}{(h^{w_i} h^{u_i})^{r_i}} = (g^{y_i - x_i})^{r_i}$$

This becomes 1, if y_i is equal to x_i, or looks random in G, otherwise, as the random r_i is not known to $\mathcal{S}$. Therefore, $\mathcal{S}$ outputs $X \cap Y$, while $\mathcal{C}$ outputs $\perp$.

Server privacy: For corrupted $\mathcal{C}$, simulator SIM_C can be constructed as follows: SIM_C chooses n random values $(z_1, ...z_n)$ in Z_q and encrypts them under pk, i.e. it sets $b_i = \mathsf{Enc}_{pk}(z_i)$ for all i, and it sends pk and $(b_1, ..., b_n)$ to $\mathcal{C}$. (Note that SIM_C could use $\mathcal{C}$'s input X in the simulation, but the above algorithm does not need this input.) Because of the IND-CPA security of ElGamal encryption under the hardness assumption of the decisional Diffie-Hellman (DDH) problem, the ciphertexts produced by SIM_C are indistinguishable from the ones produced by $\mathcal{S}$ in the real protocol execution.

Client privacy: For corrupted $\mathcal{S}$, simulator SIM_S can be constructed as follows: Given Y and $X \cap Y$,

- SIM_S receives the public key pk and $(b_1, ..., b_n)$ from $\mathcal{S}$, and it sets $z_i = 0$ for all $i \in X \cap Y$ and picks $z_i \leftarrow_\$ Z_q$ for all $i \notin X \cap Y$.

- SIM_S computes $a_i = Enc_{pk}(z_i)$ for all i, and sends $(a_1, ..., a_n)$ to $\mathcal{S}$.

It follows that $\mathcal{S}$'s view in the interaction with SIM_S matches the interaction with the real-world $\mathcal{C}$: In both cases, each a_i is either a random encryption of 1, if $i \in X \cap Y$, or an encryption of a random value in Z_q, if $i \notin X \cap Y$.

Proof of Theorem 5.2 (PLI-CA security):

Correctness:

Assume a protocol execution where both client and server are honest. The client gets output $\perp$ from the protocol Π. The server computes $c_i = \frac{a_{i,2}}{a_{i,1}^{sk}}$. Since

$$\begin{aligned}
\frac{(b_{i,2} \cdot h^{u_i} g^{-x_i})^{r_i}}{((b_{i,1} \cdot g^{u_i})^{r_i})^{sk}} &= \frac{(h^{w_i} g^{y_i} h^{u_i} g^{-x_i})^{r_i}}{((g^{w_i})^{sk} (g^{u_i})^{sk})^{r_i}} \\
&= \frac{(h^{w_i} h^{u_i} g^{y_i - x_i})^{r_i}}{(h^{w_i} h^{u_i})^{r_i}} \\
&= \frac{(h^{w_i} h^{u_i})^{r_i} (g^{y_i - x_i})^{r_i}}{(h^{w_i} h^{u_i})^{r_i}} \\
&= (g^{y_i - x_i})^{r_i}
\end{aligned}$$

Similar to Theorem 5.1, c_i is 1 if $y_i = x_i$, or is random, otherwise. Thus, $|\{c_i : c_i = 1\}| = |X \cap Y|$, which the server outputs.

Server privacy:

We construct a simulator SIM_C. The simulator SIM_C generates encryption keys and chooses n random values $(z_1, ..., z_n)$, which it encrypts under pk such that $s_i = \mathsf{Enc}_{pk}(z_i)$ for all i. It sends to the corrupt client the public key pk and the encrypted random values $(s_1, ..., s_n)$.

Because of the IND-CPA security of ElGamal encryption, under the hardness assumption of the decisional Diffie-Hellman problem, $(s_1, ..., s_n)$ is indistinguishable from the real protocol view of $(b_1, ..., b_n)$. The corrupt client gets output $\perp$.

Since the view of the (corrupted) $\mathcal{C}$ is the same as the one in PLI, server privacy is also met in PLI-CA.

Client privacy: Considering the corrupted $\mathcal{S}$, a simulator SIM_S can be similarly constructed. In addition to SIM_S in Theorem 5.1, SIM_S sends $\pi(A_1, ..., A_n)$ to $\mathcal{S}$, for a randomly chosen π in $\mathcal{P}_n$. For some $\pi, \pi' \in \mathcal{P}_n$, $\mathcal{S}$'s view in the interaction with SIM_S is indistinguishable from the output of the protocol execution with real $\mathcal{C}$, as $\pi\{\{1\}_{i\in X\cap Y}, \{z_i\}_{i\notin X\cap Y}\} \stackrel{c}{\equiv} \pi'\{\{1\}_{i\in X\cap Y}, \{(y_i - x_i)r_i\}_{i\notin X\cap Y}\}$.

Proof of Theorem 5.3 (t-PLI security):

Correctness: Assume an execution of the protocol Π with honest $\mathcal{C}$ and honest $\mathcal{S}$. For all $i \in [n]$, $\mathcal{S}$ computes s'_i as follows: $(s'_i||hs_i) := H_i(a_{i,1}^{sk}) \oplus e_i = H_i(a_{i,1}^{sk}) \oplus H_i(a_{i,2}) \oplus s_i, \forall i$. Since $H_i(a_{i,1}^{sk}) = H_i(((b_{i,1} \cdot g^{u_i})^{r_i})^{sk}) = H_i(((g^{w_i} \cdot g^{u_i})^{sk})^{r_i}) = H_i((h^{w_i} \cdot h^{u_i})^{r_i})$ and $H_i(a_{i,2}) = H_i((b_{i,2} \cdot h^{u_i} g^{-x_i})^{r_i}) = H_i((h^{w_i} g^{y_i} \cdot h^{u_i} g^{-x_i}))^{r_i}) = H_i((h^{w_i} \cdot h^{u_i})^{r_i} (g^{y_i - x_i})^{r_i})$, $H_i(a_{i,2}) = H_i(a_{i,1}^{sk})$, if $y_i = x_i$. Otherwise, $H_i(a_{i,2})$ is a hash of a random element in G. Consequently, if $y_i = x_i$, $s'_i = s_i$, or is random, otherwise. Depending on the cardinality of intersection $I := \{i \mid s'_i = s_i\}_{i\in[n]}$, three possible cases exist:

1. $|I| \geq k := \lceil \frac{n+t}{2} \rceil$: $\mathcal{S}$ can apply BW algorithm to recover (p, err). $\mathcal{S}$ outputs $\{y_i : err(i) \neq 0\}_{i\in[n]}$, which is the set of input elements where their corresponding shares are correct, i.e., indices of intersecting elements.

2. $t \leq |I| < k$: $\mathcal{S}$ can examine every subset of size t, and check which subset reconstructs s' such that $H_i(s') = hs_i$.

3. $|I| < t$: $\mathcal{S}$ can neither reconstruct s' nor learn anything about x_i's.

Server privacy:

We construct a simulator SIM_C. The simulator SIM_C generates encryption keys and chooses n random values $(z_1, ...z_n)$, which it encrypts under pk such that $s_i = \mathsf{Enc}_{pk}(z_i)$ for all i. It sends to the corrupt client the public key pk and the encrypted random values $(s_1, ..., s_n)$. Because of the IND-CPA security of ElGamal encryption, under the hardness assumption of the decisional Diffie-Hellman problem, $(s_1, ..., s_n)$ is indistinguishable from the real protocol view of $(b_1, ..., b_n)$. The corrupt client gets output $\perp$.

Since the view of the (corrupted) $\mathcal{C}$ is the same as the one in PLI (and PLI-CA), server privacy is also met in t-PLI.

Client privacy: Considering corrupted $\mathcal{S}$, a simulator SIM_S can be constructed as below. Given Y and $|X \cap Y|$,

- SIM_S receives the public key pk and $(b_1, ..., b_n)$ from $\mathcal{S}$. It sets $z_i = 0$ if $y_i \in X \cap Y$, and $z_i \leftarrow^{\$} G$, otherwise.

- SIM_S computes $A_i = \mathsf{Enc}_{pk}(z_i)$ for all i.

- SIM_S generates $S \leftarrow \$$, and computes the shares $(S_1, ..., S_n) \leftarrow \mathsf{Share}_{(t,n)}(S)$.

- SIM_S computes $E_i = H_i(A_{i,2}) \oplus H'_i(S_i)$ for all i.

- SIM_S sends $(A_{1,1}, ..., A_{n,1})$ and $(E_1, ..., E_n)$ to $\mathcal{S}$.

Comparing $\mathcal{S}$'s view in the interaction with SIM_S and in the real execution of Π with $\mathcal{C}$, first, the tuples, $(a_{1,1}, ..., a_{n,1})$ and $(A_{1,1}, ..., A_{n,1})$, are indistinguishable because of the IND-CPA security of ElGamal encryption, under the hardness of DDH problem. Then, the tuples, $(e_1, ..., e_n)$ and $(E_1, ..., E_n)$ are also indistinguishable, as $(a_{1,2}, ..., a_{n,2})$ and $(A_{1,2}, ..., A_{n,2})$ are

indistinguishable because of the IND-CPA security of ElGamal encryption, and through the security of one-time pad encryption with the randomly generated shares guaranteed by Shamir secret sharing scheme.

Proof of Theorem 5.4 (t-PLI-CA security):

Correctness: Similar to Theorem 5.3, $H_i(a_{i,2}) = H_i(a_{i,1}^{sk})$ if $y_i = x_i$, or is otherwise random. Thus, if the number of intersecting elements $|I| \geq t$, $\mathcal{S}$ can apply the Berlekamp-Welch algorithm or examine every subset to obtain the number of errors and obtain $|\{s_i' : err(i) \neq 0\}_i| = |X \cap Y|$, which the server outputs. If $|I| < t$, $\mathcal{S}$ outputs $\perp$.

Server privacy: Since the view of the (corrupted) $\mathcal{C}$ is the same as the one in PLI (and PLI-CA and t-PLI), server privacy is also met in t-PLI-CA.

Client privacy: Due to the similarity of the protocols, the construction of the simulator SIM_S, considering the corrupted $\mathcal{S}$, is similar to the one in Theorem 5.3. In addition to SIM_S in Theorem 5.3, the following step is modified: SIM_S computes $A_i = \pi(\mathsf{Enc}_{pk}(z_i))$ for all i. As in Theorem 5.3, $\mathcal{S}$'s view (even without this modification) in the interaction with SIM_S and the output of the protocol execution with real $\mathcal{C}$ are indistinguishable, and adding pseudorandom permutations randomly chosen in $\mathcal{P}_n$ does not change it.

5.6 Implementation & Evaluation

The performance of each protocol was tested on a MacBook Pro with a 2.3 GHz Dual-Core Intel i5 processor with 16 GB RAM. We implemented each protocol in `C`, using the OpenSSL v3.1.2 library. The Fisher-Yates shuffle [38], instantiated with a cryptographically secure pseudo-random number generator provided by OpenSSL, served as a secure shuffle for the PLI-CA and t-PLI-CA protocols. To evaluate the performance of our constructions, we implemented each protocol using Yao's garbled circuit (GC) [90] provided by MP-SPDZ

[53].

For the homomorphic encryption schemes, we use the Elliptic Curve-based ElGamal (EC-ElGamal) with a 224-bit key provided by OpenSSL, after comparing it to the standard and the additively homomorphic ElGamal with the equivalent security level (i.e., with a 2048-bit key). From our comparisons, using EC-ElGamal in our PLI implementation has around 2× better bandwidth and 20× better latency than using the other ElGamal schemes. The list elements of the participants were 256-bit integers.

To evaluate, we measured the communication and computation costs for:

1. increasing size n of input lists, and compared it to the GC implementation.

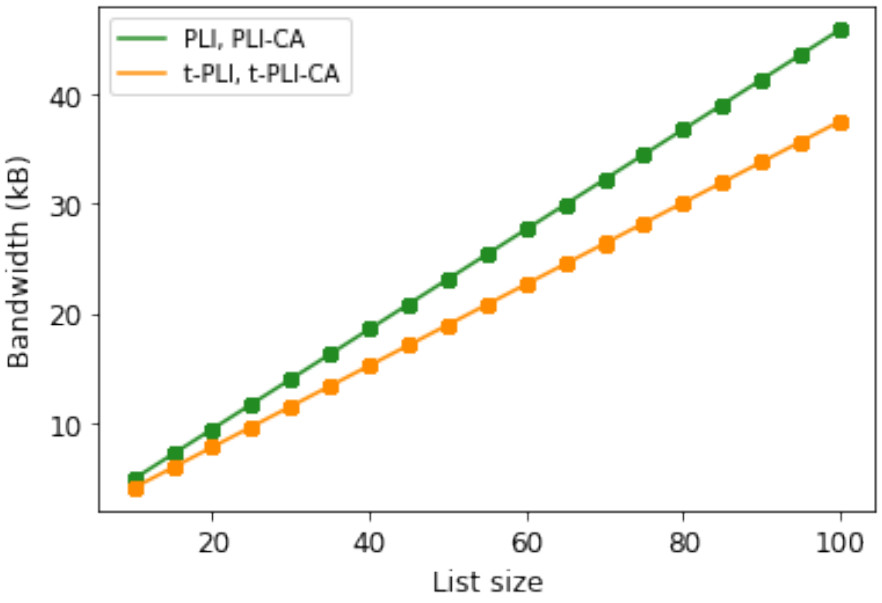
2. increasing intersection cardinality $|X \cap Y|$, to observe the effects of low or high number of matching elements in the case of t-PLI and t-PLI-CA.

3. increasing reconstruction threshold t, and analyzed the effect of the threshold choice on the execution time of t-PLI and t-PLI-CA.

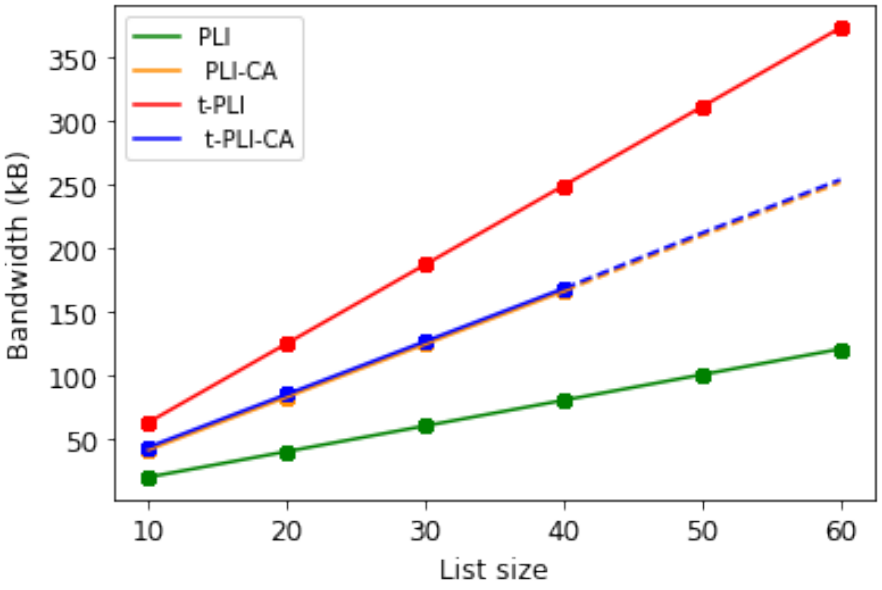
For each instantiation of the protocol, we present the average of 10 code executions as a result. Our implementation code is publicly available at [2].

5.6.1 Bandwidth Evaluation

For 224-bit list inputs, Fig. 5.4a showcases the bandwidth required to run each protocol for increasing values of the list size, n. We observe that the bandwidth used in the PLI and PLI-CA protocols is the same, which is consistent with the protocol constructions in Section 5.5. This is because the messages sent in PLI-CA and PLI are equivalent, only permuted. The same is observed for t-PLI and t-PLI-CA. The threshold PLI protocols require less bandwidth, e.g. 11.5 kB for $n = 30$, than the PLI and PLI-CA protocols, which need 14 kB for $n = 30$. This is due to $\mathcal{C}$ sending only the first element of each encrypted value $a_{i,1}$ and a



(a)



(b)

Figure 5.4: Bandwidth consumed by all protocols using (a) EC-ElGamal and (b) GC in MP-SPDZ [53]. In (b), due to errors occurring for higher n, projected values are shown with dashed lines. t-PLI results coincide with t-PLI-CA.

one-time pad of the hash of the second element $H_i(a_{i,2})$ in the threshold protocols, instead of sending $(a_{i,1}, a_{i,2})$ in the non-threshold protocols.

Compared to MP-SPDZ garbled circuits with 32-bit integer inputs, the bandwidth required

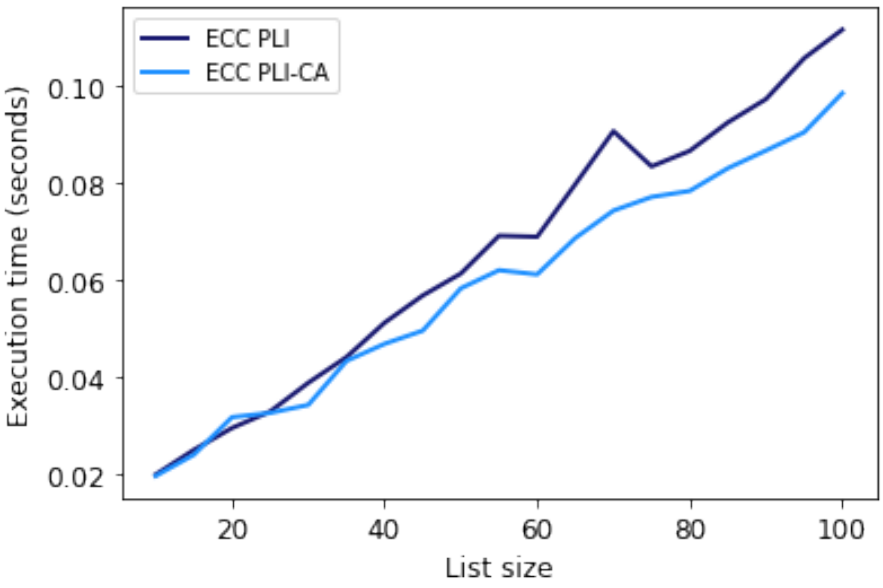
by PLI constructions with EC-ElGamal encryptions is lower. For example, for $n = 30$, the GC versions of PLI, PLI-CA, t-PLI, and t-PLI-CA require 60.4 kB, 124.8 kB, 187.4 kB and 127.0 kB, respectively (shown in Fig. 5.4b). Since the evaluation through MP-SPDZ outputs errors for higher values of n, we present the dashed lines for the projected values for $n = 50$ and $n = 60$.

5.6.2 Execution Time Evaluation

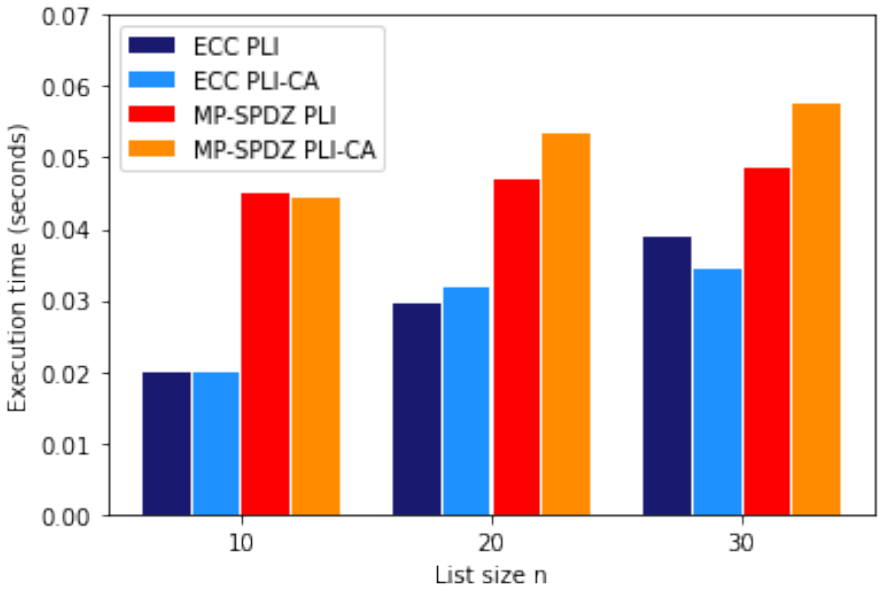
We first evaluated the running time of each protocol using EC-ElGamal as the list size n increased from 10 to 100, with step size 5. Shown in Fig. 5.5a, the execution time of PLI and PLI-CA increases linearly as n increases. Compared to the GC implementations, our constructions are more efficient in terms of execution time (Fig. 5.5b). For example, for $n = 30$, our PLI and PLI-CA protocols took 0.039 and 0.034 seconds on average, while the ones using GC took 0.048 and 0.057 seconds to complete.

Now, fixing the cardinality $|X \cap Y|$ to 80% of n and the threshold t to $\frac{1}{3}n$, we evaluate the execution time of t-PLI and t-PLI-CA, with results displayed in Fig. 5.6a. Because of the Berlekamp-Welch (BW) protocol, the execution time is $O(n^3)$, which is also reflected in the figure. For $n = 30$, t-PLI and t-PLI-CA take on average 0.32 and 0.31 seconds, while for $n = 100$, t-PLI and t-PLI-CA take on average 11.4 and 11.1 seconds. For t-PLI and t-PLI-CA, the GC approach is always faster than ours, e.g., at $n = 30$, it takes on average 0.055 seconds to run GC-based tPLI implementation.

Lastly, we compare the execution time for t-PLI and t-PLI-CA as $|X \cap Y|$ increases and as t increases, for a fixed $n = 50$. In Fig. 5.6b, with a t fixed to $\frac{1}{3} \cdot 50 \approx 16$, the protocol is very fast when $|X \cap Y|$ does not allow efficient reconstruction of the secret, i.e. when the BW algorithm fails. On average, when the number of matches is below $k = \lceil \frac{n+t}{2} \rceil$ (here $k = 33$), execution for both t-PLI and t-PLI-CA takes 0.19 seconds. For $|X \cap Y| \geq 35$, the BW algorithm succeeds, and the t-PLI and t-PLI-CA protocols take a linearly increasing



(a)



(b)

Figure 5.5: Execution time of PLI and PLI-CA protocols (a) for increasing list size, and (b) compared to the GC implementation in MP-SPDZ [53]

amount of time. This is because the BW algorithm starts by assuming the maximum number of errors $\lfloor \frac{n-t}{2} \rfloor$, and tries to recover the secret. If it fails to recover the secret, it decreases the assumed number of errors by one unit, tries again, and repeats until successful recovery. Thus, for a higher intersection cardinality $|X \cap Y|$, there are fewer errors, and running the

BW algorithm requires more trials.

We observe a similar effect in Fig. 5.6c, where the execution time first decreases linearly as the threshold t increases, i.e., as $\lfloor \frac{n-t}{2} \rfloor$ decreases. Then, when t gets too high and efficient reconstruction is impossible, the BW algorithm fails and the t-PLI and t-PLI-CA protocols take about 0.19 seconds to complete.

5.7 Security against malicious participants

Each of our protocol variants enables moderate-cost upgrades to security against malicious participants, i.e., parties that arbitrarily diverge from the protocol. Recall that PLI should reveal nothing to $\mathcal{C}$ while $\mathcal{S}$ should learn (only) $X \cap Y$. This would be violated if a malicious $\mathcal{C}$ learns anything about $\mathcal{S}$'s Y and/or makes $\mathcal{S}$ output I s.t. $i \in I$ even if $X[i] \neq Y[i]$, or if a malicious $\mathcal{S}$ learns more about $\mathcal{C}$'s X beyond $X \cap Y$. We argue that this will not happen, i.e., our protocols can become maliciously secure, at a moderate increase to protocol costs. We present the modifications below along with proof outlines.

Security against Malicious Server: All four protocol variants offer security for the client in the face of a malicious server, if the protocol is amended in two ways. First, each input x_i and y_i is replaced by resp. $\bar{x}_i = H_i^q(x_i)$ and $\bar{y}_i = H_i^q(y_i)$ where H^q is a Random Oracle (RO) hash onto Z_q and $H_i^q(x) = H^q(i, x)$. Second, $\mathcal{S}$ appends a zero-knowledge proof of knowledge (ZKPoK) of the discrete logarithm $sk = \mathsf{DL}_g(h)$ to its message $(pk, (b_1, ..., b_n))$, where $pk = (G, q, g, h)$.[3] This modification adds 1 (multi)exponentiation for $\mathcal{C}$ and $\mathcal{S}$ and 4λ bits to bandwidth, thus increasing protocol costs by only a small additive factor.

Security of PLI (in Fig. 5.1) against Malicious Client: can be achieved by pre-processing inputs via RO hash as above, and adding ZKPoK of (δ_i, r_i) s.t. $a_{i,1} = g^{\delta_i}(b_{i,1})^{r_i}$

[3] All our ZKPoK proofs are variants of Schnorr's proof of discrete logarithm, made non-interactive given an RO hash. They add 2λ bits to bandwidth and 1 (multi-)exponentiation to the computation of both parties per statement, see [29].

for all $i \in [n]$ to $\mathcal{C}$'s message ($\delta_i = u_i r_i$ for honest $\mathcal{C}$). In the protocol, $\mathcal{S}$ outputs I s.t. $i \in I$ iff $a_{i,2} = a_{i,1}^{sk}$, but if $(h, a_{i,1}, b_{i,1}, b_{i,2}) = (g^{sk}, g^{\delta_i} b_{i,1}^{r_i}, g^{w_i}, h^{w_i} g^{\bar{y}_i})$, this is equivalent to $a_{i,2} = h^{\delta_i}(b_{i,2} g^{-\bar{y}_i})^{r_i}$, i.e., $g^{\bar{y}_i} = b_{i,2}(a_{i,2}^{-1} h^{\delta_i})^{1/r_i}$. This modification adds n (multi-)exponentiations for both $\mathcal{C}$ and $\mathcal{S}$ and $4\lambda n$ bits to bandwidth, resulting in PLI protocol with two-sided security against malicious parties, with bandwidth and computation costs that are larger than the HbC-secure version of PLI, by resp. factors $\times 1.5$ and $\times 1.4$.

Security of t-PLI (in Fig. 5.2) against Malicious Client: is accomplished by the same ZKPoK as for PLI above. In particular, $\mathcal{C}$ does not need to prove anything regarding the e_i values or the secret-sharing encoded by them. Since this modification is the same as above, it incurs the same overhead *compared to the PLI protocol*, i.e. it does not affect Berlekamp-Welch decoding costs.

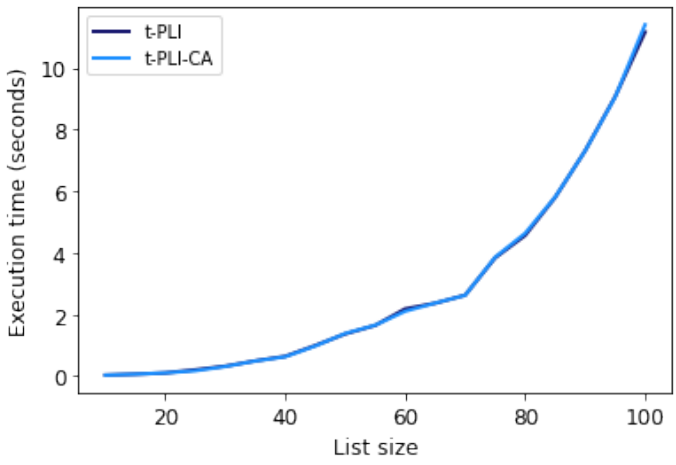
Security of X-CA protocols against Malicious Client: The only difference between our (t-)PLI-CA and (t-)PLI protocols is that $\mathcal{C}$ permutes ciphertexts a_i before processing them further. Since all our ZKPoK's above involve only $a_{i,1}$ values, we just add a ZK *proof of shuffle*, e.g. [46]. Taking the RO-based non-interactive version of the 3-round proof-of-shuffle due to Furukawa [40] adds $15n$ exponentiations to the total computation cost and $6n\lambda$ bits to bandwidth, which together with the changes above increases the bandwidth and computation costs of the HbC versions of these protocols by resp. factors $\times 2.25$ and $\times 4.4$.

Proof Outlines for Security against Malicious Participants:

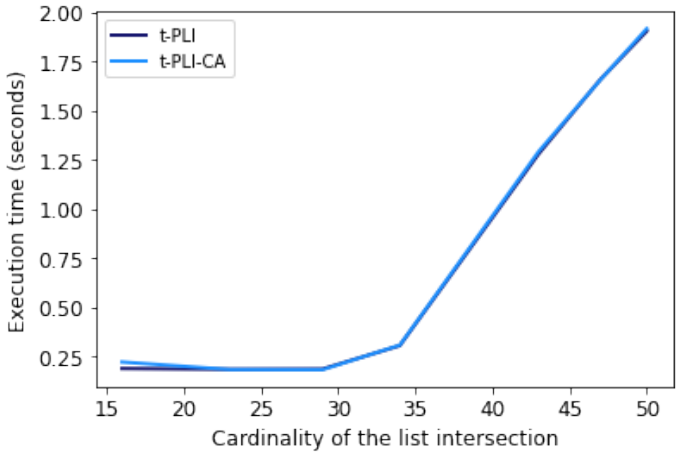
For any malicious server $\mathcal{S}^*$, simulator SIM_S can (1) emulate H^q, capturing $\mathcal{S}^*$'s queries to H^q, (2) on $\mathcal{S}$'s message, extract $sk = \mathsf{DL}_g(pk)$ from the ZKPoK and sets $M_i = \mathsf{PreDec}_{sk}(b_i)$ for each i, and (3) form $\mathcal{S}^*$'s effective input Y into either protocol, by setting $Y[i]$, for each i, to y s.t. $\mathcal{S}^*$ queried (i, y) to H and $M_i = g^{\bar{y}_i}$ for $\bar{y}_i = H_i^q(y)$, and setting $Y[i] = \perp$ if there is no such H^q query. After sending Y to PLI functionality and getting output out_S back, SIM_S can then simulate honest client's message as in the proofs of Theorems 5.1-5.4 above.

For malicious client $\mathcal{C}^*$, SIM_C extracts δ_i, r_i for $i \in [n]$ from $\mathcal{C}^*$'s message, "pre-decrypts" each $a_{i,2}$ as $M_i = b_{i,2}(a_{i,2}^{-1}h^{\delta_i})^{1/r_i}$, and forms $\mathcal{C}^*$'s effective input X into PLI, by setting $X[i]$, for each i, to x s.t. $\mathcal{C}^*$ received $\bar{x}_i = H_i^q(x)$ from RO H^q and $M_i = g^{\bar{x}_i}$, and setting $X[i] = \perp$ if $\mathcal{C}^*$ made no such H^q query. An honest $\mathcal{S}$ will output $I = X \cap Y$. In the simulation, SIM_C verifies the same constraint as in the protocol for $\bar{x}_i = H(i, x)$, hence $i \in I$ if $y_i = x_i$. SIM_C misses any match where the constraint is satisfied for $\bar{y}_i = H_i^q(y)$ but $\mathcal{C}^*$ does not query H_i^q on y. Since this can happen only with probability n/q, the simulated and real views are computationally indistinguishable.

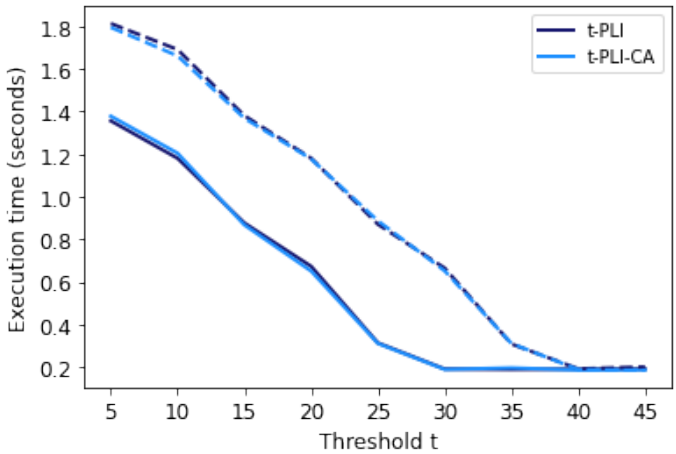
For the modified t-PLI, SIM_C will extract (δ_i, r_i)'s as above, but it modifies the pre-decryption and derivation of $\mathcal{C}^*$'s effective input X: SIM_C searches through $\mathcal{C}^*$'s queries to H_i^q, H_i', and H_i, to determine if there exists s s.t., for some subset of at least t indexes i, $\bar{x}_i = H_i^q(x_i)$, $hs_i = H_i'(s)$, and $ha_i = H_i(a_{i,2})$ satisfy that (1) $(ha_i \oplus e_i)[R] = hs_i$, where $z[L]$ and $z[R]$ stand for resp. the $\mathbb{F}$ and $\{0,1\}^{2\lambda}$ components of z; (2) $g^{\bar{x}_i} = b_{i,2}(a_{i,2}^{-1}h^{\delta_i})^{1/r_i}$; and (3) $s_i = (ha_i \oplus e_i)[L]$ lie on t-degree polynomial which interpolates to s. If SIM_C identifies such a subset, it forms X from all (i, x_i) pairs found above and sets all other values in X to $\perp$. If no such subset of at least t indexes is found, then SIM_C sets X to $(\perp)^n$. SIM_C efficiently and correctly simulates the real protocol because if H', H are RO's then constraint $(ha_i \oplus e_i)[R] = hs_i$ can be satisfied at most by a single (ha_i, hs_i) pair, except for negligible probability. On the other hand, if fewer than $t + 1$ matches exist in the real protocol, the hs_i components recovered by $\mathcal{S}$ are indistinguishable from random, and thus the real protocol execution on X s.t. $|X \cap Y| < t$ is indistinguishable from the simulation where SIM_C sets X to an all-empty sequence $(\perp)^n$.



(a)



(b)



(c)

Figure 5.6: Execution time of the t-PLI and t-PLI-CA protocols for (a) increasing list size, (b) increasing cardinality of intersection list, for $n = 50$ and $t = 16$, and (c) increasing threshold, for $n = 50$, $|X \cap Y| = 44$ (solid lines) and $|X \cap Y| = 39$ (dashed lines)

5.8 Conclusion

This chapter proposes PIVA, privacy-preserving identity verification for accountless users, that includes four protocols: PLI, PLI-CA, t-PLI, and t-PLI-CA. We analyze each protocol's security and performance via a prototype implementation. Proposed PLI and PLI-CA protocols are more efficient than their GC-based counterparts in terms of communication and computation costs. Whereas the proposed t-PLI and t-PLI-CA protocols incur much less bandwidth than GC-based ones and are faster than current threshold PSI protocols. All proposed protocols can be upgraded to be secure in the malicious model with a small increase in costs.

Chapter 6

Supporting Private and Efficient Consumer Requests to Data Brokers

Abstract

Under data privacy regulations, such as GDPR and CCPA, consumers are legally entitled to access, delete, or opt-out from collection of, personal information that so-called data brokers collect about them. However, the process of exercising these rights remains non-uniform, inefficient, and privacy-invasive. A consumer must submit individual requests to hundreds of data brokers, each requiring different – and often extensive – identity verification procedures. In January 2026, California launched a centralized facility that allows consumers to formulate a single (unified) request to all registered data brokers. While this simplifies the process, it raises significant privacy, scalability, and security concerns. Current commercial services (e.g., DeleteMe and Incogni) automate such requests for a fee, although they require consumers to provide all personal information upfront to yet another entity of uncertain trustworthiness.

Motivated by the above, this chapter presents PIVOT, an efficient cryptographic scheme that enables privacy-agile data access, deletion, and opt-out requests through an untrusted intermediary. PIVOT combines proxy re-encryption with additively homomorphic encryption so that data brokers can verify consumer identity without the intermediary learning any personally identifiable information (PII). A consumer encrypts their PII once, and the intermediary facilitates consumer identity verification with all data brokers. Data brokers learn only whether their databases contain matching records corresponding to a given consumer, without acquiring any new PII beyond what they already possess. PIVOT ensures both consumer and broker confidentiality, as well as verification soundness, requiring no user interaction after the initial request.

6.1 Introduction

Prominent current data privacy regulations, such as the European Union's General Data Protection Regulation (GDPR) [37], California Consumer Privacy Act (CCPA) [17] and Brazilian Lei Geral de Proteção de Dados Pessoais (LGPD) [45], grant consumers the right to access, delete, and correct their personal information collected by entities such as website operators and other providers. Under these regulations, the latter must verify consumer identity before processing such requests in order to prevent unauthorized disclosure of information or other potentially harmful actions.

For entities with which consumers have accounts (e.g., social media platforms or retailers), identity verification is straightforward: consumers are authenticated using their normal account credentials [17, 16]. However, whenever consumers lack accounts, e.g., those who merely browse websites or whose data is collected through third-party tracking, verification becomes much more difficult. Without account-based authentication, entities rely on matching consumer-provided Personally Identifiable Information (**PII**) against their records

[12, 58, 86]. This approach varies widely across business entities, resulting in awkward and burdensome verification procedures that are fragmented and privacy-invasive for consumers.

A data broker is a very particular business entity type which collects, aggregates, and sells consumer information, usually without any direct consumer interaction. The data broker ecosystem includes both small and obscure operators as well as established (mainly US-based) credit-rating agencies, such as Experian, Transunion, Equifax, Lexis, and Acxiom [19]. They compile profiles from social media, public records (e.g., court cases, property ownership changes, as well as car and drivers' license registrations), and even from other $\mathcal{DBR}$s.[1] All of that is usually done **without** consumers' knowledge or consent [8]. Though some $\mathcal{DBR}$s support user accounts (e.g., Experian), the overwhelming majority does not. Also, historically, $\mathcal{DBR}$s are mostly an American phenomenon with roots going back to pre-Internet years – 1980s [39].

In the context of most business types, consumers knowingly use offered services. In contrast, consumers typically do not know which $\mathcal{DBR}$s have their PII. Moreover, even for those $\mathcal{DBR}$s that do, consumers do not know the nature of that PII. Therefore, in order to comprehensively exercise their privacy rights, a consumer must submit a separate request to each $\mathcal{DBR}$, which can be numerous, e.g., California maintains a $\mathcal{DBR}$ registry with well over 500 $\mathcal{DBR}$s [19]. Alarmingly, each request requires supplying a potentially distinct set of PII to each $\mathcal{DBR}$ so that they can perform required-by-law identity verification. This holds even for those $\mathcal{DBR}$s that have no PII about the consumer. This is because a consumer has no way of knowing *a priori* whether a given $\mathcal{DBR}$ has any PII about them.

A recent study on CCPA compliance of $\mathcal{DBR}$s [86] reported that one $\mathcal{DBR}$, after receiving extensive consumer PII for verification purposes, responded with:

`"...[W]e never had this information until receiving it in your email below..."`

[1] Hereafter we use "$\mathcal{DBR}$s"instead of "data brokers" for brevity.

Thus, the act of exercising privacy rights can itself lead to new data collection by entities that previously held no PII about the consumer. Furthermore, this is not a one-time burden. $\mathcal{DBR}$s continuously acquire new consumer PII. Each request provides only an ephemeral snapshot of the data that a given $\mathcal{DBR}$ possesses. Privacy-conscious consumers must periodically repeat this burdensome process with hundreds of $\mathcal{DBR}$s to maintain any lasting control over their PII [86]. Several commercial services, including DeleteMe [5], Incogni [76], and Aura [10], offer to submit data deletion requests on behalf of consumers to hundreds of $\mathcal{DBR}$s, for a monthly or yearly fee. However, these services require consumers to provide all of their PII upfront to yet another entity of uncertain trustworthiness. Storing all consumer PII at such a single intermediary elevates the risk of eventual leakage. Also, a recent study shows that these services have inconsistent $\mathcal{DBR}$ coverage and underwhelming efficacy [47].

Since 2020, California Privacy Protection Agency (CPPA, recently renamed CalPrivacy) has been the California state government body tasked with CCPA enforcement. Recently (in 2023), it proposed the *Delete Request and Opt-out Platform* (DROP) [27, 20]. This is a free state-run platform that just launched in January 2026. It allows consumers to submit a single deletion or opt-out request and requires all registered California $\mathcal{DBR}$s to process such requests. While DROP appreciably improves consumer accessibility, several concerns remain. **First,** DROP only handles deletion and opt-out requests, while information access requests are not supported. **Second,** DROP treats (stores, etc.) consumer identifiers as hashes, which represents insufficient privacy protection for low-entropy consumer data, such as names, dates of birth, as well as physical and email addresses. **Third,** neither DROP nor the aforementioned commercial services address one fundamental problem: during identity verification, $\mathcal{DBR}$s may learn new PII even about consumers whose data they do not possess.

As long as $\mathcal{DBR}$s continue to operate at scale, consumers need an effective, efficient, and privacy-agile means of exercising their legal privacy rights without incurring a heavy burden. An ideal solution would minimize the amount of time and effort consumers would spend

navigating complex verification procedures. To ensure widespread adoption and practicality, such a system must also avoid imposing excessive operational overhead on $\mathcal{DBR}$s.

Motivated by the above, this chapter introduces PIVOT, a cryptographic scheme that enables privacy-preserving identity verification for accountless consumer data requests. Recall that the overwhelming majority of Verifiable Consumer Requests (VCRs) in the context of $\mathcal{DBR}$s correspond to accountless consumers.

PIVOT provides a general framework adaptable to any jurisdiction requiring identity verification for data access, deletion, or opt-out requests. In it, a consumer makes a single request to an intermediary which, in turn, facilitates identity verification with all registered $\mathcal{DBR}$s. Notably, this intermediary learns no PII about the consumer, while $\mathcal{DBR}$s learn no new PII beyond what they already possess. PIVOT supports all three types of consumer requests: access, deletion, and opt-out. At the core of PIVOT, one important feature is a novel combination of (surprisingly practical) proxy re-encryption and additively homomorphic encryption techniques.

Anticipated Contributions:

- **Unified Consumer Effort:** Consumers provide their PII to the intermediary **once**. This process supports any number of $\mathcal{DBR}$s without any further interactions involving the consumer.
- **Strong Privacy Guarantees:** PIVOT ensures that: (1) the intermediary learns no consumer PII during the verification process, and (2) $\mathcal{DBR}$s learn only whether their databases contain matching records. No new PII is acquired beyond what they already possess.
- **Comprehensive Request Support:** PIVOT supports all three request types: delete, opt-out, and information access.
- **Open-Source Implementation:** A publicly available implementation of PIVOT [87], evaluated in Section 6.8.

Organization: Section 6.2 provides the background and overviews related work. Section 6.3 introduces our cryptographic building blocks. Section 6.4 describes PIVOT preliminaries, followed by Section 6.5 which presents the complete scheme. Next, Section 6.6 provides security analysis. Section 6.7 describes the implementation, Section 6.8 addresses its performance, and Section 6.9 discusses extensions and variants.

6.2 Background & Related Work

Additional background and related work common to previous Chapters are given in Chapter 2.

6.2.1 Privacy-Enhancing Techniques

Prior research yielded some privacy-preserving methods to mitigate privacy risks of current identity verification methods (see Chapter 2). The protocol presented in Chapter 5 enables privacy-preserving identity verification for accountless consumers using PLI; however it remains inconvenient for the consumer who must perform the same PLI-based process with every $\mathcal{DBR}$.

PIVOT also uses PLI, and provides a streamlined and privacy-preserving identity verification process which is significantly less burdensome for consumers.

Delegated PSI is a class of PSI protocols where the "cloud" privately and obliviously computes the intersection of two parties' sets. This way, computation overhead for the two parties is greatly reduced, as compared to regular PSI.

It appears that [54] is the first to propose a delegated PSI protocol, based on RSA encryption with quadratic complexity. [56] presented one based on symmetric key encryption, where both parties and the cloud must be online throughout. More recent work uses Bloom filters [4, 3], polynomial encodings [50], or PRFs [91]. These protocols either lack efficiency [50], or

require users to directly communicate with each other [4, 3, 91].

[94] is the approach most similar to the one in this chapter. It uses proxy re-encryption to securely compute the PSI. However, that scheme is limited in its privacy properties, since it makes plaintext guessing attacks easy. In the context of relatively low-entropy consumer PII, guessing attacks must be mitigated. Also, the setting of [94] is the case when two peer parties want to compute a PSI with each other, aided by the cloud. In contrast, we focus on a setting where a consumer wants to securely and efficiently compute **many PSI instances with many $\mathcal{DBR}$s**.

6.3 Building Blocks

Throughout the chapter, we denote $x^{(\alpha)}$ and $x^{(\beta)}$ the first and the second value of a tuple $x = (a, b)$, i.e. $x^{(\alpha)} = a$ and $x^{(\beta)} = b$.

6.3.1 Bilinear pairings

A map $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map if $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups with the same prime order q and if it satisfies the following properties:

(1)Bilinearity: $\forall\, a, b \in \mathbb{Z}_q$ and $g \in \mathbb{G}_1$,$e(g^a, g^b) = e(g, g)^{ab}$.

(2)Non-degeneracy: With g a generator of $\mathbb{G}_1$, $e(g, g) \neq 1_{\mathbb{G}_2}$.

(3)Computability: The pairing operation e is efficient.

6.3.2 Private List Intersection (PLI)

PLI is a variant of Private Set Intersection (PSI) where values are labeled and ordered [51]. Since in PLI one-to-one comparison of list elements is possible, computation of the intersection is more straightforward than in PSI. The ideal functionality $\mathcal{F}_{\mathsf{PLI}}$ with inputs

from a sender S (in this context, the consumer) and a receiver R (in this context, a broker) is defined in Figure 6.1.

Ideal Functionality of PLI

Parameters: List size n

Ideal Functionality of PLI:

1. Wait for inputs $Y = (y_1, y_2, ..., y_n)$ from receiver R (in PIVOT, a broker) and $X = (x_1, x_2, ..., x_n)$ from sender S (in PIVOT, a consumer)
2. Compute the list intersection $\mathsf{out}_R = X \cap Y$
3. Send $\perp$ to S and send out_R to R

Figure 6.1: Idealized PLI Functionality: $\mathcal{F}_{\mathsf{PLI}}$ from [51]

6.3.3 Proxy re-encryption (PRE)

Proxy re-encryption (PRE) is a cryptographic primitive that allows a designated proxy to convert a ciphertext encrypted for Alice to a ciphertext encrypted for Bob, given a special PRE key provided by Alice. During PRE, the proxy does not learn the plaintext. A unidirectional PRE scheme ensures that PRE can only be done from Alice to Bob – and not from Bob to Alice – with the same PRE key.

PIVOT builds on the unidirectional PRE scheme from [9], which is based on the bilinear Diffie-Hellman assumption. The original scheme is presented below, with modifications described in Definition 6.3 to provide additive homomorphism.

Definition 6.1 (Unidirectional Proxy re-encryption [9])**.** *This unidirectional* PRE *scheme consists of a tuple of six algorithms:*

$(\mathsf{KeyGen}, \mathsf{ReEncKeyGen}, \mathsf{Enc}_1, \mathsf{Enc}_2, \mathsf{ReEnc}, \mathsf{Dec})$

Global parameters are (g, Z) *where* $g \in \mathbb{G}_1$ *and* $Z = e(g, g)$*:*

[PRE*$_1$*] $\mathsf{KeyGen}\ (g, Z) \rightarrow (sk_A, pk_A)$*: A user A's secret key is of the form* $sk_a = (a_1, a_2)$*, and its corresponding public key is* $pk_A = (Z^{a_1}, g^{a_2})$*.*

[PRE$_2$] $\mathsf{ReEncKeyGen}\ (a_1, g^{b_2}) \to rk_{A\to B}$*: A user A delegates decryption to a user B by publishing the* PRE *key* $rk_{A\to B} = (g^{b_2})^{a_1} = g^{a_1 b_2}$*, computed from B's public key and A's secret key.*

[PRE$_3$] $\mathsf{Enc}_1\ (pk_A, m) \to c_{a,1}$*: To encrypt a message* $m \in \mathbb{G}_2$ *such that it can only be decrypted by user A (first-level encryption), pick random* $r \in \mathbb{Z}_q$ *and output ciphertext* $c_{a,1} = (Z^{a_1 r}, mZ^r)$*. To achieve proxy invisibility, i.e., it is not possible to determine whether the ciphertext was re-encrypted, output* $c_{a,2} = (Z^{a_2 r}, mZ^r)$ *at no security loss.*

[PRE$_4$] $\mathsf{Enc}_2\ (Z^{a_1}, m) \to c_{a,R}$*: To encrypt a message* $m \in \mathbb{G}_2$ *such that it can be re-encrypted to A's delegates (second-level encryption), pick random* $r \in \mathbb{Z}_q$ *and output ciphertext* $c_{a,R} = (g^r, m(Z^{a_1})^r) = (g^r, mZ^{a_1 r})$*.*

[PRE$_5$] $\mathsf{ReEnc}\ (c_{a,R}, rk_{A\to B}) \to c_{b,1}$*: Given* $rk_{A\to B}$*, one can change a second-level ciphertext destined to A into a first-level ciphertext destined to B. From* $c_{a,R} = (g^r, mZ^{ar})$*, compute* $e(g^r, rk_{A\to B}) = Z^{r a_1 b_2}$*. Output* $c_{b,1} = (Z^{r a_1 b_2}, mZ^{a_1 r}) = (Z^{b_2 r'}, mZ^{r'})$ *where* $r' = a_1 r$*.*

[PRE$_6$] $\mathsf{Dec}\ (sk_a, c_{a,i}) \to m$*: To decrypt a first-level ciphertext* $c_{a,i} = (c_{a,i}^{(\alpha)}, c_{a,i}^{(\beta)})$ *where* $i \in [1, 2]$*, compute* $m = \frac{c_{a,i}^{(\beta)}}{(c_{a,i}^{(\alpha)})^{1/a_i}}$*, to decrypt a second-level ciphertext* $c_{a,R} = (c_{a,R}^{(\alpha)}, c_{a,R}^{(\beta)})$*, compute* $m = \frac{c_{a,R}^{(\beta)}}{e(c_{a,R}^{(\alpha)}, g)^{a_1}}$*.*

Definition 6.2 (Extended Decisional Bilinear Diffie–Hellman (eDBDH) assumption)**.** *Let* $\mathbb{G}_1$ *be a cyclic group of prime order q with generator g,* $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ *bilinear, and* $Z := e(g, g)$*. Choose* $a, b, c, d \leftarrow \mathbb{Z}_q$ *uniformly at random. Given:* $\left(g,\ g^a,\ g^b,\ g^c,\ Z^{bc^2},\ Z^d\right)$*, decide whether* $d \equiv abc \pmod q$*. The eDBDH assumption says that no PPT adversary can distinguish* $(g, g^a, g^b, g^c, Z^{bc^2}, Z^{abc})$ *from* $(g, g^a, g^b, g^c, Z^{bc^2}, Z^t)$ *for uniform* $t \in \mathbb{Z}_q$ *with non-negligible advantage [9].*

Theorem 6.1 (PRE security)**.** *Under the eDBDH assumption in* $(\mathbb{G}_1, \mathbb{G}_2, e)$*, re-encrypted ciphertexts produced by the scheme are secure. Moreover, under the Discrete Log assumption*

in G, recovering a user's secret key from delegations is infeasible. Hence, the scheme is secure as long as eDBDH and Discrete Log assumptions for the secret key hold.

6.3.4 Additively Homomorphic Encryption

An additively homomorphic encryption (AHE) scheme supports arithmetic addition of multiple ciphertexts, i.e., an AHE scheme has the property that $\mathsf{Enc}(k, x) \oplus \mathsf{Enc}(k, y) = \mathsf{Enc}(k, x + y)$, where $\mathsf{Enc}()$ is the encryption function, $\oplus$ is some operator, and k is an encryption key.

6.4 System and Threat Models

Notation used in the rest of the chapter is summarized in Table 6.1.

6.4.1 Three-Party Setting

Prior PLI methods assume direct communication between a consumer ($\mathcal{C}$) and a $\mathcal{DBR}$ ($\mathcal{BR}_j$). PIVOT extends this to a three-party setting where an intermediary ($\mathcal{MED}$) mediates between $\mathcal{C}$ and $\mathcal{DBR}$s. $\mathcal{C}$ submits their PII once to $\mathcal{MED}$, and the latter orchestrates identity verification with every $\mathcal{DBR}$ on $\mathcal{C}$'s behalf, while learning no $\mathcal{C}$ PII.

6.4.2 Participants and Threat Model

All $\mathcal{DBR}$s and $\mathcal{MED}$ are assumed to be honest-but-curious, i.e., they follow the rules, though they may attempt to learn PII beyond what is necessary. This assumption stems from real-world constraints: $\mathcal{MED}$, whether a government-provided service or a paid service, is trusted by the consumers, and it is impossible to verify whether a $\mathcal{DBR}$ is telling the truth without audits. We consider collusion to be out of scope since entities can communicate consumer data outside PIVOT. An extension of PIVOT that tolerates malicious $\mathcal{DBR}$s and $\mathcal{MED}$ is sketched out in Section 6.9.3. On the other hand, it is reasonable to assume that $\mathcal{C}$ might be malicious and attempt to bypass identity verification by providing fake or manipulated

Table 6.1: Notation Summary

Notation	Description
Parties	
$\mathcal{C}$	Consumer
$\mathcal{MED}$	Intermediary
$\mathcal{BR}$	Set of all registered brokers
$\mathcal{BR}_j$	A specific broker j
Consumer Identifiers and Data	
$\mathsf{ID}_{\mathcal{C}}$	$\mathcal{C}$'s public identifier (e.g., email, name+DOB)
$\mathcal{P}_{\mathcal{C}}$	$\mathcal{C}$'s complete list of PII attributes $(x_1, x_2, \ldots, x_N)$
$\mathcal{P}_{\mathcal{C}_j}$	$\mathcal{C}$'s list of PII attributes $(x_1, x_2, \ldots, x_n)$ for IV with $\mathcal{BR}_j$
Broker Data	
$\mathcal{P}_{\mathcal{BR}_j}$	$\mathcal{BR}_j$'s list of PII attributes $(y_1, y_2, \ldots, y_n)$ for a $\mathcal{C}$
Labels and Metadata	
$\mathsf{L}_{\mathcal{BR}}$	Union of all PII field labels across all $\mathcal{BR}$
$\mathsf{L}_{\mathcal{BR}_j}$	Set of PII field labels required by $\mathcal{BR}_j$ for IV
ℓ_i	Label of PII element i
Size Parameters	
m	Size of $\mathcal{BR}$ (total # of brokers)
N	Size of $\mathsf{L}_{\mathcal{BR}}$ (# of unique PII labels across $\mathcal{BR}$)
n	Size of $\mathcal{P}_{\mathcal{BR}_j}$ (number of PII elements in a $\mathcal{BR}_j$'s list)
$[x]$	Set of integers $\{1, 2, \ldots, x\}$
Cryptographic Primitives	
g	Generator of cyclic group $\mathbb{G}_1$
Z	Equals $e(g, g)$ where e is the bilinear pairing
e	Bilinear pairing map
Keys	
sk_A	Secret key of party A; format: (a_1, a_2)
pk_A	Public key of party A; format: (Z^{a_1}, g^{a_2})
$rk_{A \to B}$	PRE key from party A to party B
Ciphertexts	
c_i	$\mathcal{C}$'s ciphertext of $-x_i$ encrypted under consumer's key
$\tilde{c}_i$	$\mathcal{BR}_j$'s ciphertext of y_i encrypted under $\mathcal{BR}_j$'s key
a_i	Combined and blinded ciphertext sent to $\mathcal{BR}_j$ by $\mathcal{MED}$
d_i	$\mathcal{BR}_j$'s partially decrypted result; equals $1_{\mathbb{G}_2}$ if $x_i = y_i$

PII. In more detail, the parties play the following roles:

($\mathcal{C}$) Consumer: wants to submit a data request (of type: delete, opt-out, or access) to every $\mathcal{DBR}$.[2] For simplicity's sake, we assume that $\mathcal{C}$ makes the same type of request to

[2]Consumers retain the ability to contact $\mathcal{DBR}$s individually through traditional request mechanisms.

all $\mathcal{DBR}$s. $\mathcal{C}$ interacts only with the intermediary, and never contacts $\mathcal{DBR}$s directly. $\mathcal{C}$ is assumed to have a labeled list of their own PII $\mathcal{P}_{\mathcal{C}}$ and a unique public identifier $\mathsf{ID}_{\mathcal{C}}$. Sharing $\mathsf{ID}_{\mathcal{C}}$ in plaintext is a functional requirement rather than a design choice: a verifiable consumer request must identify the individual it concerns, and a $\mathcal{DBR}$ must locate candidate records before any verification can take place (Section 6.4.3) [17]. Current practices reflect this, e.g., DROP shares consumer identifiers with all registered $\mathcal{DBR}$s as hashes [20], which, given their low entropy, offer marginal protection over plaintext [60]. The privacy downside is that a $\mathcal{DBR}$ that has no records for $\mathcal{C}$ Alice nonetheless learns $\mathsf{ID}_{\mathcal{C}}$. It thus learns that Alice exists and is exercising her privacy rights. Section 6.9.2 presents a variant of PIVOT that avoids this leakage by ensuring that only $\mathcal{DBR}$s that have $\mathsf{ID}_{\mathcal{C}}$ receive the request.

PIVOT ensures that $\mathcal{C}$ learns nothing about $\mathcal{DBR}$ datasets beyond what $\mathcal{DBR}$s explicitly return. Identity theft, whereby an adversary impersonates $\mathcal{C}$ using stolen PII, is out of scope: it defeats *any* verification scheme based on PII matching (including direct requests, commercial removal services, and DROP), and PIVOT neither helps nor hurts such an adversary.

($\mathcal{MED}$) Intermediary: acts as a proxy between consumers and $\mathcal{DBR}$s. Its responsibilities include receiving deletion, opt-out, or access requests from $\mathcal{C}$ and contacting $\mathcal{DBR}$s on $\mathcal{C}$'s behalf. $\mathcal{MED}$ handles all identity verification procedures for $\mathcal{C}$. $\mathcal{MED}$ learns $\mathsf{ID}_{\mathcal{C}}$ and, for each $\mathcal{DBR}$, only a single bit: whether that $\mathcal{DBR}$'s database contains any record for $\mathsf{ID}_{\mathcal{C}}$ (plus the final status of each request). Beyond $\mathsf{ID}_{\mathcal{C}}$, no consumer PII is ever revealed to $\mathcal{MED}$. This role resembles current commercial services, such as DeleteMe or Incogni, which submit multiple deletion requests on consumers' behalf. As an honest-but-curious entity, $\mathcal{MED}$ may attempt to learn consumer PII during interactions with $\mathcal{C}$ and $\mathcal{DBR}$s.

($\mathcal{BR}$) Brokers: We use $\mathcal{BR}$ to denote the set of all $\mathcal{DBR}$s of size m, while $\mathcal{BR}_j$ is a specific $\mathcal{DBR} \in \mathcal{BR}$. A given $\mathcal{BR}_j$ has a database of consumer PII. We denote $\mathcal{P}_{\mathcal{BR}_j}$ the list of PII elements corresponding to a specific $\mathcal{C}$. A $\mathcal{BR}_j$ must declare the exact types of PII elements

$\mathsf{L}_{\mathcal{BR}_j}$ that it requires for consumer identity verification purposes.

Upon receiving a request from $\mathcal{MED}$ for $\mathcal{C}$, $\mathcal{BR}_j$ performs identity verification. If the identity verification succeeds, $\mathcal{BR}_j$ processes $\mathcal{C}$'s request:

- *Deletion:* remove $\mathcal{C}$'s data.
- *Opt-out:* prevent future collection or sale of $\mathcal{C}$'s data.
- *Access:* return all consumer data currently held.

Brokers may attempt to learn new consumer PII beyond what they already possess.

Scope

We do not consider Denial-of-Service (DoS) attacks whereby $\mathcal{MED}$ refuses to forward requests or $\mathcal{DBR}$s ignore them. We also do not attempt to enforce the veracity of $\mathcal{DBR}$ responses. A $\mathcal{DBR}$ can always claim that $\mathsf{ID}_{\mathcal{C}}$ is not in its database, or report an empty intersection during identity verification. Ensuring that $\mathcal{DBR}$s faithfully process requests is therefore a compliance and enforcement concern. Our focus is on $\mathcal{C}$ privacy and minimizing $\mathcal{C}$ interaction when exercising their privacy rights.

6.4.3 Assumptions

Communication Channels: All communication between participants occurs over authenticated and encrypted channels, e.g., TLS/HTTPS.

$\mathcal{DBR}$Registration: A list of $\mathcal{DBR}$s is publicly available, e.g., by legally mandated registration, e.g., as in CCPA. Each $\mathcal{DBR}$ declares which PII elements are needed to verify consumer identity and publishes this along with its public key.

Public Identifier: Similar to the proposed CPPA DROP system, each $\mathcal{C}$ has a public identifier $\mathsf{ID}_{\mathcal{C}}$ (at least based on their name and email address) that is shared in plaintext.

$\mathcal{MED}$ forwards $\mathsf{ID}_{\mathcal{C}}$ to $\mathcal{DBR}$s, who use it to search their databases before performing identity verification. In Section 6.9, we provide a protocol extension to prevent a $\mathcal{DBR}$ from learning $\mathsf{ID}_{\mathcal{C}}$ if $\mathcal{C}$ is not in $\mathcal{DBR}$'s database.

Asynchronous Operation: $\mathcal{MED}$ maintains state for each $\mathcal{C}$, tracking the status of their request with each $\mathcal{DBR}$. The timing of $\mathcal{DBR}$s' responses will vary, e.g., CCPA and GDPR allow up to 90 days (including extensions) to process consumer requests. For **Access** requests, $\mathcal{MED}$ also stores encrypted data returned by $\mathcal{DBR}$s until the consumer retrieves it.[3]

Identity Verification Requirements: Under CCPA, entities must verify that a consumer making a data request is the individual about whom the business has collected PII:

> "...[A] request made by a consumer ..., and that the business can verify, using commercially reasonable methods, ... to be the consumer about whom the business has collected personal information.." (CCPA §1798.140(ak)).

In practice, verification procedures vary widely across $\mathcal{DBR}$s and, notably, do not involve strong identification mechanisms, such as verification of government-issued IDs. Many $\mathcal{DBR}$s require minimal effort: a consumer provides basic PII (e.g., name and email address) with no attempts to verify that they correspond to the individual they claim to be.

The proposed CCPA DROP system validates California residency, as stated in [20]:

> "...[C]onsumers will have their California residency verified by the Agency prior to submitting a deletion request ..." (§7620(a))

Verification of California residency is separate and distinct from verifying consumer identity. Since opt-out requests do not require identity verification [16], DROP only verifies eligibility

[3] $\mathcal{MED}$ determines its own data retention policy based on scalability and privacy considerations. A bounded retention period (e.g., 90-180 days) balances practical deployment constraints with minimizing the period during which encrypted consumer data resides with $\mathcal{MED}$.

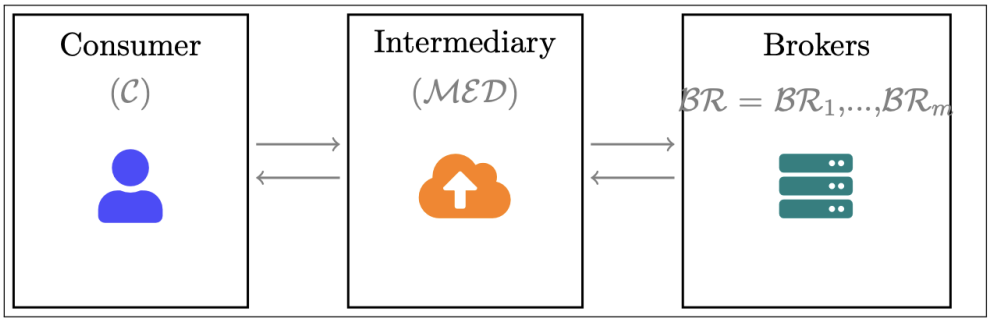


Figure 6.2: PIVOT scheme Participants.

either through Login.gov [57] or by asking name, date of birth, California address, and either a phone number or an email address [28].

Identity verification remains the responsibility of each $\mathcal{DBR}$. Similarly, PIVOT does not prescribe or enforce any particular verification policy. $\mathcal{MED}$ can verify $\mathcal{C}$'s possession of its identifiers (e.g., email address, phone number) by sending an email or SMS challenge.

6.4.4 System Requirements

PIVOT must satisfy the following requirements:

Correctness. A legitimate (well-formed) consumer identity that corresponds to at least one record in a $\mathcal{DBR}$'s database must result in a successful identity verification.

Consumer Efficiency. A consumer must be able to submit data requests to all $\mathcal{DBR}$s in one effort, ideally with a single (logical) message to the intermediary.

Privacy. Two privacy requirements must be satisfied, based on the adversary model in Section 6.4.2.

1. **Consumer Privacy:**
    - The intermediary does not learn consumer PII from the identity verification process.

- $\mathcal{DBR}$s do not learn new consumer PII beyond what they already store in their databases.

2. $\mathcal{DBR}$ **Privacy:** Both the consumer and the intermediary learn nothing about the content or characteristics of $\mathcal{DBR}$s' databases, aside from the content sent in responses to legitimate requests.

6.5 PIVOT Description

6.5.1 Consumer Experience

From $\mathcal{C}$'s perspective, PIVOT is straightforward and requires minimal effort. $\mathcal{C}$ begins by visiting $\mathcal{MED}$'s web portal using HTTPS, where they are presented with a form containing labeled fields for various PII elements, e.g., name, email address, phone number, date of birth, etc. This form includes the union of all PII elements that all registered $\mathcal{DBR}$s require for identity verification. $\mathcal{C}$ fills in as many fields as they choose – and/or are able – to provide and selects the desired request type: **Delete, Access**, or **Opt-out**.

Once $\mathcal{C}$ clicks submit, their browser performs all cryptographic operations locally. Each PII value is hashed and blinded client-side before leaving their device. The browser simultaneously generates `PRE` keys for each registered $\mathcal{DBR}$. Afterwards, the browser sends the encrypted data and all `PRE` keys to $\mathcal{MED}$. At this point, $\mathcal{C}$'s work is done. The identity verification process and all communication with $\mathcal{DBR}$s happen entirely through $\mathcal{MED}$. $\mathcal{C}$ simply waits for notifications as $\mathcal{DBR}$ responses trickle in.

Each notification indicates whether a particular $\mathcal{DBR}$ successfully verified $\mathcal{C}$'s identity and processed the request. For **Delete** and **Opt-out** requests, the notification confirms whether the action was completed or informs of verification failure. For **Access** requests, $\mathcal{DBR}$s encrypt $\mathcal{C}$'s data under $\mathcal{C}$'s public key and send it to $\mathcal{MED}$, where $\mathcal{MED}$ is accordingly unable to read this encrypted data. When $\mathcal{C}$ visits $\mathcal{MED}$'s portal to retrieve their data,

their browser automatically decrypts it.

6.5.2 PIVOT Details

PLI enables privacy-preserving identity verification: $\mathcal{BR}_j$ can verify $\mathcal{C}$'s identity by checking which PII elements received match those it possesses, without learning any new PII elements [51].

To achieve PLI with the help of an oblivious $\mathcal{MED}$, we use additively homomorphic PRE. $\mathcal{MED}$ can re-encrypt ciphertexts received from $\mathcal{C}$ to $\mathcal{BR}_j$'s public key, allowing $\mathcal{BR}_j$ to decrypt them without $\mathcal{MED}$ learning any plaintext value. The PRE scheme proposed by Ateniese et al. [9] does not allow additive homomorphic operations, and thus is unsuitable for PIVOT. Therefore, we modify it slightly, as follows:

Definition 6.3 (Additively homomorphic proxy re-encryption)**.** *This additively homomorphic unidirectional* PRE *scheme is a tuple of seven algorithms:*

$$(\mathsf{KeyGen}, \mathsf{ReEncKeyGen}, \mathsf{Enc}_1, \mathsf{Enc}_2, \mathsf{ReEnc}, \mathsf{Add}, \mathsf{ParDec})$$

Global parameters (g, Z) *where* $g \in \mathbb{G}_1$ *and* $Z = e(g, g)$*:*

[$\boldsymbol{A}_1$] $\mathsf{KeyGen}\ (g, Z) \rightarrow (sk_A, pk_A)$*: A user's A secret key is of the form* $sk_a = (a_1, a_2)$*, and its corresponding public key is* $pk_A = (Z^{a_1}, g^{a_2})$*.*

[$\boldsymbol{A}_2$] $\mathsf{ReEncKeyGen}\ (a_1, g^{b_2}) \rightarrow rk_{A \rightarrow B}$*: A user A delegates decryption to a user B by publishing the* PRE *key* $rk_{A \rightarrow B} = (g^{b_2})^{a_1} = g^{a_1 b_2}$*, computed from B's public key and A's secret key.*

[$\boldsymbol{A}_3$] $\mathsf{Enc}_1\ (pk_A, m) \rightarrow c_{a,1}$*: To encrypt a message* $m \in \mathbb{Z}_q$ *such that it can only be decrypted by user A, pick random* $r \in \mathbb{Z}_q$ *and output ciphertext* $c_{a,1} = (Z^{a_1 r}, Z^m Z^r)$*. To achieve proxy invisibility, output* $c_{a,2} = (Z^{a_2 r}, Z^m Z^r)$ *at no security loss.*

[$\boldsymbol{A}_4$] Enc_2 $(Z^{a_1}, m) \to c_{a,R}$: To encrypt a message $m \in \mathbb{Z}_q$ such that it can be re-encrypted to A's delegates, pick random $r \in \mathbb{Z}_q$ and output ciphertext $c_{a,R} = (g^r, Z^m(Z^{a_1})^r) = (g^r, Z^m Z^{a_1 r})$.

[$\boldsymbol{A}_5$] ReEnc $(c_{a,R}, rk_{A\to B}) \to c_{b,1}$: Given $rk_{A\to B}$, one can change a second-level ciphertext destined to A into a first-level ciphertext destined to B. From $c_{a,R} = (g^r, Z^m Z^{ar})$, compute $e(g^r, rk_{A\to B}) = Z^{ra_1b_2}$. Output $c_{b,1} = (Z^{ra_1b_2}, Z^m Z^{a_1 r}) = (Z^{b_2 r'}, Z^m Z^{r'})$ where $r' = a_1 r$.

[$\boldsymbol{A}_6$] Add $(c_1, c_2) \to (c_{12})$: To add two first-level ciphertexts encrypted under the same key[4] $c_1 = (c_1^{(\alpha)}, c_1^{(\beta)})$ and $c_2 = (c_2^{(\alpha)}, c_2^{(\beta)})$, compute $c_{12} = (c_1^{(\alpha)} \cdot c_2^{(\alpha)}, c_1^{(\beta)} \cdot c_2^{(\beta)})$.

[$\boldsymbol{A}_7$] ParDec $(sk_a, c_{a,i}) \to m$: To partially decrypt a first level ciphertext $c_{a,1} = (c_{a,1}^{(\alpha)}, c_{a,1}^{(\beta)})$, compute $Z^m = \frac{c_{a,1}^{(\beta)}}{(c_{a,1}^{(\alpha)})^{1/a_1}}$, to partially decrypt a first-level proxy-invisible or re-encrypted ciphertext to user A: $c_{a,2} = (c_{a,2}^{(\alpha)}, c_{a,2}^{(\beta)})$, compute $Z^m = \frac{c_{a,2}^{(\beta)}}{(c_{a,2}^{(\alpha)})^{1/a_2}}$, and to partially decrypt a second-level ciphertext $c_{a,R} = (c_{a,R}^{(\alpha)}, c_{a,R}^{(\beta)})$, compute $Z^m = \frac{c_{a,R}^{(\beta)}}{e(c_{a,R}^{(\alpha)}, g)^{a_1}}$.

Remark. Our introduction of an additively homomorphic operation ([$\mathbf{A}_6$] Add), whereby the plaintext becomes the exponent of Z, changes decryption semantics. Direct decryption becomes computationally infeasible, since it would require solving an instance of the bilinear discrete logarithm problem. However, testing for zero (i.e., whether the encrypted plaintext $m = 0$) is efficient, doable by simply checking whether $Z^m = 1_{\mathbb{G}_2}$. PIVOT takes advantage of this exact property.

The security of the additively homomorphic proxy re-encryption scheme follows the proof of the (non-additively homomorphic) proxy re-encryption scheme by [9], modifying it to put the message m in the exponent of Z.

[4] E.g., a re-encrypted ciphertext can be added to a proxy-invisible, first-level ciphertext.

Technical Overview

$\mathcal{C}$ contacts $\mathcal{MED}$ to request service and sends its plaintext identifier $\mathsf{ID}_{\mathcal{C}}$. After receiving $\mathsf{ID}_{\mathcal{C}}$, $\mathcal{MED}$ composes a union $\mathcal{U}$ of all PII types required by $\mathcal{BR}$ in their identity verification processes.

Since every possible needed PII element is included in $\mathcal{U}$, $\mathcal{C}$ can provide all needed PII only once. Along with the PII, $\mathcal{MED}$ provides a list of all $\mathcal{BR}$ public keys. $\mathcal{C}$ encrypts (Enc_2) the PII elements under *its own* public key, and computes a `PRE` key for each $\mathcal{BR}_j$. All these values are sent to $\mathcal{MED}$ in a single logical message.

We note that computing a `PRE` key for each $\mathcal{BR}_j$ represents a non-negligible burden for $\mathcal{C}$, though it is much more efficient than encrypting all PII values under each $\mathcal{BR}_j$'s key: $\mathcal{O}(nm)$ vs $\mathcal{O}(n+m)$. This overhead is quantified in Section 6.8.

At this point, $\mathcal{MED}$ has all $\mathcal{C}$'s PII elements as second-level ciphertexts (which it cannot decrypt) and `PRE` keys from $\mathcal{C}$ for each $\mathcal{BR}_j$ in $\mathcal{BR}$. $\mathcal{MED}$ stores these values and proceeds with contacting each $\mathcal{BR}_j$ on $\mathcal{C}$'s behalf. After checking that a specific $\mathcal{BR}_j$ has the specific $\mathsf{ID}_{\mathcal{C}}$ in its database, $\mathcal{MED}$ re-encrypts $\mathcal{C}$'s ciphertexts under $\mathcal{BR}_j$'s public key. Next, $\mathcal{MED}$ and $\mathcal{BR}_j$ proceed with PLI computation.

PLI operates as follows:

1. $\mathcal{BR}_j$ encrypts every PII element that it has for $\mathcal{C}$ under its own public key $pk_{\mathcal{BR}_j}$, and sends the resulting ciphertext vector to $\mathcal{MED}$.
2. $\mathcal{MED}$ homomorphically subtracts each PII element encrypted by $\mathcal{BR}_j$ from the corresponding $\mathcal{C}$ ciphertext. Recall that the latter has been already proxy re-encrypted for this $\mathcal{BR}_j$. If the plaintexts of the two respective PII elements match, the result is an encryption of zero under $pk_{\mathcal{BR}_j}$, which $\mathcal{BR}_j$ can easily test.
3. To prevent $\mathcal{BR}_j$ from solving a discrete logarithm problem (e.g., for small-value plain-

texts) and thus learning non-matching PII elements of $\mathcal{C}$, $\mathcal{MED}$ re-randomizes each encrypted subtraction by exponentiating it to a unique random value. $\mathcal{MED}$ then sends a vector of all re-randomized ciphertexts to $\mathcal{BR}_j$.

4. $\mathcal{BR}_j$ computes the PLI result by testing whether each ciphertext is an encryption of 0. It thus learns the intersection of its list of PII elements (corresponding to $\mathcal{C}$) with the list provided by $\mathcal{C}$. $\mathcal{BR}_j$ can not decrypt any non-matching elements. Thus, it obtains no PII about $\mathcal{C}$ beyond what it already possesses. Finally, $\mathcal{BR}_j$ uses the intersection to verify $\mathcal{C}$'s identity, according to its own policy.

6.5.3 PRE Scheme Description

We now describe using the additively homomorphic PRE scheme corresponding to Definition 6.3 above. It is also summarized in Figure 6.3.

Setup:

Each $\mathcal{BR}_j$ must register with a relevant authority and provide a public key $pk_{\mathcal{BR}_j} \leftarrow \mathsf{KeyGen}(g, Z)$. $\mathcal{MED}$ accesses the registry and retrieves all public keys. $\mathcal{MED}$ also retrieves from each $\mathcal{BR}_j$ the list of PII elements $\mathsf{L}_{\mathcal{BR}_j}$ it needs for consumer identity verification purposes and computes its union $\mathsf{L}_{\mathcal{BR}} = \bigcup_{\mathcal{BR}_j \in \mathcal{BR}} \mathsf{L}_{\mathcal{BR}_j}$. $\mathcal{MED}$ publishes both $\{pk_{\mathcal{BR}_j}\}_{\mathcal{BR}_j \in \mathcal{BR}}$ and $\mathsf{L}_{\mathcal{BR}}$ on its website.

Phase 1: $\mathcal{C}$ One-time Data Transmission:

[**P1.1**]: $\mathcal{C}$ retrieves $\{pk_{\mathcal{BR}_j}\}_{\mathcal{BR}_j \in \mathcal{BR}}$, and $\mathsf{L}_{\mathcal{BR}}$ from $\mathcal{MED}$ (e.g., through `HTTP GET`).

[**P1.2**]: For each label $\ell_i \in \mathsf{L}_{\mathcal{BR}}$, $\mathcal{C}$ encrypts each corresponding PII element x_i using the second-level encryption algorithm Enc_2 such that $c_i = \mathsf{Enc}_2(Z^{a_1}, -x_i) = (g^{r_i}, Z^{-x_i} Z^{a_1 r_i})$ where $r_i \xleftarrow{\$} \mathbb{Z}_q$ for all $i \in [N]$ with $((a_1, a_2), (Z^{a_1}, g^{a_2})) = (sk_{\mathcal{C}}, pk_{\mathcal{C}}) \leftarrow \mathsf{KeyGen}(g, Z)$ and where N is the size of $\mathsf{L}_{\mathcal{BR}}$. $\mathcal{C}$ also prepares zero-knowledge proofs $\pi_{c_i} = PoK\{r_i, -x_i :$

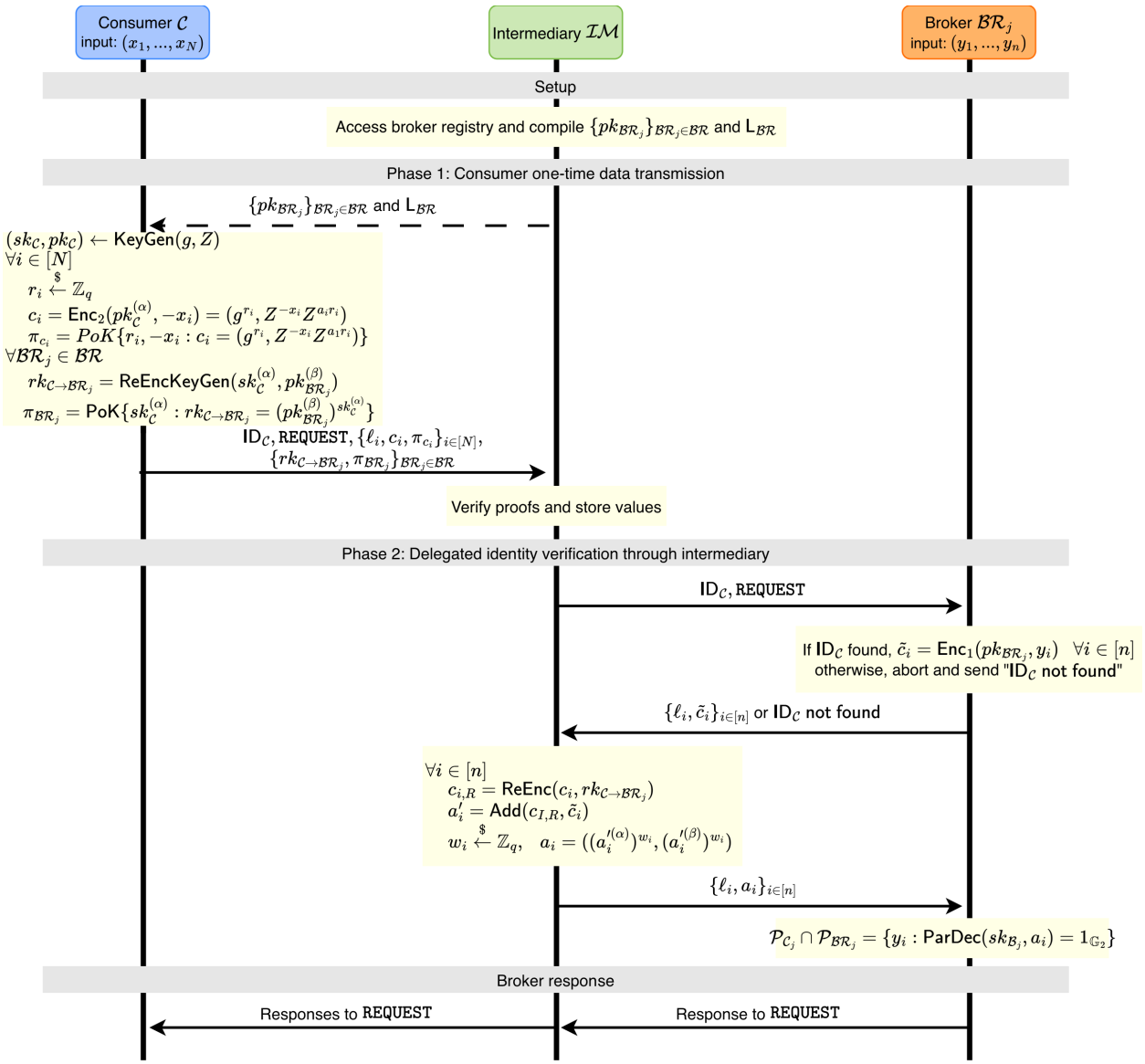


Figure 6.3: PIVOT Scheme.

$c_i = (g^{r_i}, Z^{-x_i} Z^{a_1 r_i})\}$. $\mathcal{C}$ computes PRE keys $rk_{\mathcal{C} \to \mathcal{BR}_j} \leftarrow \mathsf{ReEncKeyGen}(sk_{\mathcal{C}}^{(\alpha)}, pk_{\mathcal{BR}_j}^{(\beta)}) = (pk_{\mathcal{BR}_j}^{(\beta)})^{sk_{\mathcal{C}}^{(\alpha)}}$ for each $\mathcal{BR}_j \in \mathcal{BR}$, along with proofs $\pi_{\mathcal{BR}_j} = PoK\{sk_{\mathcal{C}}^{(\alpha)} : rk_{\mathcal{C} \to \mathcal{BR}_j} = (pk_{\mathcal{BR}_j}^{(\beta)})^{sk_{\mathcal{C}}^{(\alpha)}}\}$. $\mathcal{C}$ selects its request type REQUEST to be either Access, Opt-out or Delete, and sends $(\mathsf{ID}_{\mathcal{C}}, \mathtt{REQUEST}, \{\ell_i, c_i, \pi_{c_i}\}_{i \in [N]}, \{rk_{\mathcal{C} \to \mathcal{BR}_j}, \pi_{\mathcal{BR}_j}\}_{\mathcal{BR}_j \in \mathcal{BR}})$ to $\mathcal{MED}$. Thus, $\mathcal{MED}$ is unable to decrypt and learn anything about $\mathcal{C}$'s elements, yet it can later re-encrypt the elements under $\mathcal{BR}$' keys. We name the set of $\mathcal{C}$'s PII $\mathcal{P}_{\mathcal{C}} = \{\ell_i, c_i\}_{i \in [N]}$.

[**P1.3**]: $\mathcal{MED}$ verifies each proof and aborts if any proof does not verify. Otherwise, it stores

all received values.

At this point, $\mathcal{C}$ has completed sending its input. $\mathcal{MED}$ will communicate on its behalf with all $\mathcal{BR}_j \in \mathcal{BR}$. The following steps are repeated for each $\mathcal{BR}_j \in \mathcal{BR}$.

Phase 2: Delegated Identity Verification:

[**P2.1**]: $\mathcal{MED}$ contacts $\mathcal{BR}_j$ and sends $\mathsf{ID}_\mathcal{C}$ and the request type: "`Access`", "`Delete`", or "`Opt-out`".

[**P2.2**]: If $\mathcal{BR}_j$ does not find $\mathsf{ID}_\mathcal{C}$ in its database, it sends "$\mathsf{ID}_\mathcal{C}$ `not found`" to $\mathcal{MED}$ and aborts. If $\mathsf{ID}_\mathcal{C}$ is in its database, $\mathcal{BR}_j$ proceeds with identity verification as follows. $\mathcal{BR}_j$ encrypts all data elements $y_i \in \mathcal{P}_{\mathcal{BR}_j}$ under its own public key such that $\tilde{c}_i \leftarrow \mathsf{Enc}_1(pk_{\mathcal{BR}_j}, y_i) = (Z^{b_2\tilde{r}_i}, Z^{y_i}Z^{\tilde{r}_i})$ where $\tilde{r}_i \xleftarrow{\$} \mathbb{Z}_q$ for all $i \in [n]$ where n is the size of $\mathcal{P}_{\mathcal{BR}_j}$ and $Z^{b_2} = pk^{(\alpha)}_{\mathcal{BR}_j}$. $\mathcal{BR}_j$ sends $\{\ell_i, \tilde{c}_i\}_{i\in[n]}$ to $\mathcal{MED}$.

[**P2.3**]: $\mathcal{MED}$ receives $\{\ell_i, \tilde{c}_i\}_{i\in[n]}$ from $\mathcal{BR}_j$. $\mathcal{MED}$ retrieves $\mathcal{P}_{\mathcal{C}_j} \subseteq \mathcal{P}_\mathcal{C}$ such that $\{\ell_i\} \in \mathcal{P}_{\mathcal{C}_j} = \{\ell_i\} \in \mathcal{P}_{\mathcal{BR}_j}$. $\mathcal{MED}$ re-encrypts the values provided by $\mathcal{C}$ under $\mathcal{BR}_j$'s key, using $rk_{\mathcal{C}\to\mathcal{BR}_j}$, such that $c_{i,R} = \mathsf{ReEnc}(c_i, rk_{\mathcal{C}\to\mathcal{BR}_j})$. Now, $\mathcal{MED}$ computes $a'_i = \mathsf{Add}(c_{i,R}, \tilde{c}_i) = \mathsf{Enc}_1(pk_{\mathcal{BR}_j}, -x_i + y_i)$. If a'_i is sent to $\mathcal{BR}_j$, then $\mathcal{BR}_j$ can easily check whether $x_i = y_i$ by executing ParDec and checking whether $Z^m = 1_{\mathbb{G}_2}$. However, it may also learn values x_i when $x_i \neq y_i$ after solving for the discrete logarithm and subtracting y_i, which is feasible for small exponents. To prevent this, $\mathcal{MED}$ selects a random value $w_i \in \mathbb{Z}_q$ for each element, and re-randomizes the ciphertexts: $\mathcal{MED}$ computes $a_i = ((a'^{(\alpha)}_i)^{w_i}, (a'^{(\beta)}_i)^{w_i})$. Values $\{\ell_i, a_i\}_{i\in[n]}$ are sent to $\mathcal{BR}_j$.

[**P2.4**]: $\mathcal{BR}_j$ receives the encrypted values and for each a_i computes $d_i = \mathsf{ParDec}(sk_{\mathcal{BR}_j}, a_i)$. For each element, if $x_i = y_i$, then $d_i = 1_{\mathbb{G}_2}$. Otherwise, d_i is random and thus $x_i \neq y_i$. $\mathcal{BR}_j$ outputs $\{y_i : d_i = 1_{\mathbb{G}_2}\}_{i\in[n]} = \mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$.

$\mathcal{BR}_j$ Response:

$\mathcal{BR}_j$ uses $\mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$ to verify $\mathcal{C}$'s identity. $\mathcal{BR}_j$ responds to the data request as follows:

- "`Cannot verify identity`": $\mathcal{BR}_j$ sends this to $\mathcal{MED}$ if the identity verification was not successful, otherwise:
- "`Data deleted`": in response to a data deletion request.
- "`Opted-out`": in response to an opt-out request.
- In response to a data access request, $\mathcal{BR}_j$ encrypts $\mathcal{C}$'s data under $pk_{\mathcal{C}}$ and sends it to $\mathcal{MED}$. $\mathcal{MED}$ sends a notification to $\mathcal{C}$ to retrieve the data.

6.6 Security Analysis

PIVOT satisfies correctness, $\mathcal{C}$ privacy, and $\mathcal{BR}_j$ privacy during identity verification phases 1 and 2. We follow simulation-based definitions for privacy in the presence of honest-but-curious and malicious adversaries, as in [44, 51].

6.6.1 Correctness

PIVOT correctly executes if $\mathcal{BR}_j$ outputs the intersection of $\mathcal{MED}$'s list and $\mathcal{BR}_j$'s list $\mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$.

6.6.2 $\mathcal{C}$ Privacy

PIVOT realizes $\mathcal{C}$ privacy in the presence of honest-but-curious adversaries corrupting $\mathcal{MED}$ or $\mathcal{BR}_j$ if there exists a probabilistic polynomial-time simulator SIM such that [44, 51]:

$$\{\mathsf{SIM}_{\mathcal{MED},\mathcal{BR}_j}(1^\lambda, \mathcal{P}_{\mathcal{BR}_j}, \mathcal{F}_{\mathsf{PLI}}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}))\}_{\mathcal{P}_{\mathcal{C}_j},\mathcal{P}_{\mathcal{BR}_j},\lambda}$$
$$\stackrel{c}{\equiv} \{view^{\Pi}_{\mathcal{MED},\mathcal{BR}_j}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}, \lambda), output^{\Pi}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j})\}_{\mathcal{P}_{\mathcal{C}_j},\mathcal{P}_{\mathcal{BR}_j},\lambda}$$

Proof of Correctness

Assume an execution of PIVOT with honest $\mathcal{C}$, $\mathcal{MED}$ and $\mathcal{BR}_j$.

$\mathcal{BR}_j$ computes $d_i = \mathsf{ParDec}(sk_{\mathcal{BR}_j}, a_i)$, which is equal to $Z^{(-x_i+y_i)w_i}$ as follows:

$$
\begin{aligned}
d_i &= \frac{\left(Z^{-x_i} Z^{r_i} \cdot Z^{y_i} Z^{\tilde{r}_i}\right)^{w_i}}{\left(\left(Z^{b_2 r_i} \cdot Z^{b_2 \tilde{r}_i}\right)^{w_i}\right)^{1/b_2}} \\
&= \frac{\left(Z^{-x_i+y_i} Z^{r'_i}\right)^{w_i}}{\left(Z^{b_2 r'_i}\right)^{w_i/b_2}} && \text{where } r'_i = r_i + \tilde{r}_i \\
&= \frac{Z^{(-x_i+y_i)w_i} Z^{r'_i w_i}}{Z^{r'_i w_i}} \\
&= Z^{(-x_i+y_i)w_i}
\end{aligned}
$$

If $x_i = y_i$, then the result becomes $Z^0 = 1_{\mathbb{G}_2}$. Otherwise, the result looks random to $\mathcal{BR}_j$ as $\mathcal{BR}_j$ does not know w_i. Thus, $\mathcal{BR}_j$ correctly outputs $\mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$ by checking whether $d_i = 1_{\mathbb{G}_2}$. □

6.6.3 $\mathcal{BR}_j$ Privacy

PIVOT realizes $\mathcal{BR}_j$ privacy in the presence of an honest-but-curious $\mathcal{MED}$ if there exists a probabilistic polynomial-time simulator SIM such that [44]:

$$
\begin{aligned}
&\{\mathsf{SIM}_{\mathcal{MED}}(1^\lambda, \perp, \perp)\}_{\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}, \lambda} \\
&\qquad \stackrel{c}{\equiv} \{view^{\Pi}_{\mathcal{MED}}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}, \lambda), output^{\Pi}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j})\}_{\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}, \lambda}
\end{aligned}
$$

and in the presence of a malicious $\mathcal{C}$ if, following the real/ideal paradigm of [44], for every real world adversary $\mathcal{A}$ there exists a probabilistic polynomial-time simulator SIM such that for any input $\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}$

$$
\{\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{PLI}}, \mathsf{SIM}}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j})\}_{\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}, \lambda} \stackrel{c}{\equiv} \{\mathsf{REAL}_{\pi, \mathcal{A}}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j})\}_{\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}, \lambda} \quad (6.1)
$$

Proof of $\mathcal{C}$ Privacy

For honest-but-curious $\mathcal{MED}$ and $\mathcal{BR}_j$, a simulator $\mathsf{SIM}_{\mathcal{MED},\mathcal{BR}_j}$ can be constructed as follows:

Given Y and $\mathcal{F}_{\mathsf{PLI}}(\mathcal{P}_{\mathcal{C}_j}, \mathcal{P}_{\mathcal{BR}_j}) = \mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$, the simulator $\mathsf{SIM}_{\mathcal{MED},\mathcal{BR}_j}$:

- Generates key pair: $(sk_S, pk_S) \leftarrow \mathsf{KeyGen}(g, Z)$ where $sk_S = (s_1, s_2)$
- Receives $pk_{\mathcal{BR}_j}$ from $\mathcal{MED}$ and computes `PRE` key:

  $rk_{S \to B} \leftarrow \mathsf{ReEncKeyGen}(s_1, g^{b_2})$
- For all $i \in [n]$:
  - If $y_i \in \mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$: sets $z_i = -y_i$
  - Otherwise: sets $z_i = r$ where $r \xleftarrow{\$} \mathbb{Z}_q$
- Computes ciphertexts: $C_i = \mathsf{Enc}_2(Z^{s_1}, z_i)\ \forall i \in [n]$
- Sends $\{C_i\}_{i \in [n]}$ and $rk_{S \to B}$ to $\mathcal{MED}$

$\mathcal{MED}$'s view of the interaction with $\mathsf{SIM}_{\mathcal{MED},\mathcal{BR}_j}$ is the same as its view of its interaction with a real-world honest $\mathcal{C}$, because of the security of the proxy re-encryption scheme under the eDBDH and discrete logarithm assumptions. $\mathcal{BR}_j$'s view of the interaction with $\mathsf{SIM}_{\mathcal{MED},\mathcal{BR}_j}$ is the same as its view of its interaction with a real-world honest $\mathcal{C}$: in both cases, $\mathcal{BR}_j$ receives an encryption of a random value when $y_i \notin \mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$, and an encryption of 0 when $y_i \in \mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}$. □

6.7 Implementation

This section describes the implementation of PIVOT, which is publicly available at [87].

6.7.1 Configuration and Setup

PIVOT is implemented in Python 3.13.3 with three participants: $\mathcal{C}$, $\mathcal{MED}$, and $\mathcal{BR}_j$. Communication occurs over TLS-secured TCP sockets. All communication uses Python's `socket` and `ssl` modules with mutual TLS authentication via X.509v3 certificates[5].

[5] See https://www.ietf.org/rfc/rfc5280.txt

Proof of $\mathcal{BR}_j$ Privacy

For an honest-but-curious $\mathcal{MED}$, a simulator $\mathsf{SIM}_{\mathcal{MED}}$ can be constructed as follows: $\mathsf{SIM}_{\mathcal{MED}}$ chooses n random values $(z_1, ...z_n)$ and encrypts them under pk_b, i.e. $\mathsf{SIM}_{\mathcal{MED}}$ sets $\tilde{c}_i = \mathsf{Enc}_1(Z^{b_2}, z_i)$ for all i, and sends $(\tilde{c}_1, ..., \tilde{c}_n)$ to $\mathcal{MED}$. Because of the semantic security of the proxy re-encryption scheme under the hardness assumption of eDBDH and discrete logarithm, the ciphertexts produced by $\mathsf{SIM}_{\mathcal{MED}}$ are indistinguishable from the ones produced by $\mathcal{MED}$ in the real execution.

For a malicious $\mathcal{C}$, a simulator SIM is constructed as:

- SIM interacts with $\mathcal{C}$ and receives the values $\{c_i\}_{x_i \in X}$, $rk_{\mathcal{C} \to \mathcal{BR}_j}$ and the proofs $\{\pi_{c_i}\}_{i \in [n]}$ and $\pi_{\mathcal{BR}_j}$. If any proof does not verify, SIM aborts.
- SIM runs the extractor algorithm and extracts $-x_i, r_i$ and $sk_{\mathcal{C}}^{(\alpha)}$. SIM checks whether $sk_{\mathcal{C}}^{(\alpha)}$ was used when computing $\{c_i\}_{x_i \in X}$ by verifying whether $Z^{-x_i} Z^{sk_{\mathcal{C}}^{(\alpha)} r_i} = c_i^{(\beta)}$ (i.e., $a_i = sk_{\mathcal{C}}^{(\alpha)}$). If the verification is successful, SIM forwards $\{x_i\}$ to the ideal functionality. Otherwise, SIM sends random values.

It is easy to see that the output of the honest $\mathcal{BR}_j$ in the real world interacting with the malicious $\mathcal{C}$ is the same as the output of the ideal world $\mathcal{B}\bar{\mathcal{R}}_j$ with SIM interacting with the ideal functionality, due to the soundness of the proof of knowledge. □

A separate setup script (`setup.py`) performs one-time parameter generation, creating global parameters and key pairs, which are serialized to binary files. A shared configuration module (`config.py`) loads these at runtime across all participants.

The implementation runs on a MacBook Pro with an M4 processor and 24 GB of RAM, communicating over localhost.

6.7.2 Cryptographic Primitives

PIVOT uses the Charm-Crypto library [7] for cryptographic primitives. Charm offers native

implementations of bilinear pairings and group operations, along with built-in serialization support for cryptographic objects. Charm enables the implementation of the `PRE` scheme (Definition 6.1) with additively homomorphic modifications (Definition 6.3). It uses the SS512 symmetric pairing group `PairingGroup('SS512')`. For PII hashing, Charm's hash-to-group function `group.hash(pii, ZR)` is used.

6.7.3 Consumer $\mathcal{C}$

`consumer.py` (honest-but-curious) loads PII as label-value pairs (e.g., `email:user@example.com`). It encrypts each PII element via `encrypt(params, pk_c, -group.hash(pii, ZR))` and computes `PRE` keys using `rkeygen(params, pk_c, sk_c, pk_b)`. After establishing a TLS connection to $\mathcal{MED}$, it transmits encrypted data and `PRE` keys over the socket, closes the connection, and terminates.

6.7.4 Intermediary $\mathcal{MED}$

`intermediary.py` (honest-but-curious) orchestrates communication between consumers and brokers. Its core operations use `re_encrypt` for `PRE` and `add` for homomorphic ciphertext combination. Random blinding factors `group.random(ZR)` are applied to each ciphertext before transmission to brokers.

6.7.5 Broker $\mathcal{BR}_j$

Upon receiving the consumer identifier, `broker.py` (honest-but-curious) encrypts its PII elements using `fencrypt` and sends them to $\mathcal{MED}$. After receiving blinded ciphertexts, it performs partial decryption via `decrypt` and tests the identity element in `G2`.

6.8 Evaluation

PIVOT's performance is evaluated across all three participants. All experiments were conducted using the implementation described in Section 6.7. Each experiment was run 10 times and results were averaged.

Parameters: The union of PII elements N is configured to 30. A recent study of CCPA compliance [86] found that the total union of PII elements requested by all California-registered $\mathcal{DBR}$s is 23; we round up to overestimate and allow headroom for the innovative demands of $\mathcal{DBR}$s. For individual broker PII requirements, we set $n = 10$ elements per $\mathcal{BR}_j$, rounding up from the observed maximum of 8 [86]. California currently maintains a registry of 541 $\mathcal{DBR}$s [19]. We vary the number of registered $\mathcal{BR} = m \in \{100i : i \in [10]\}$.

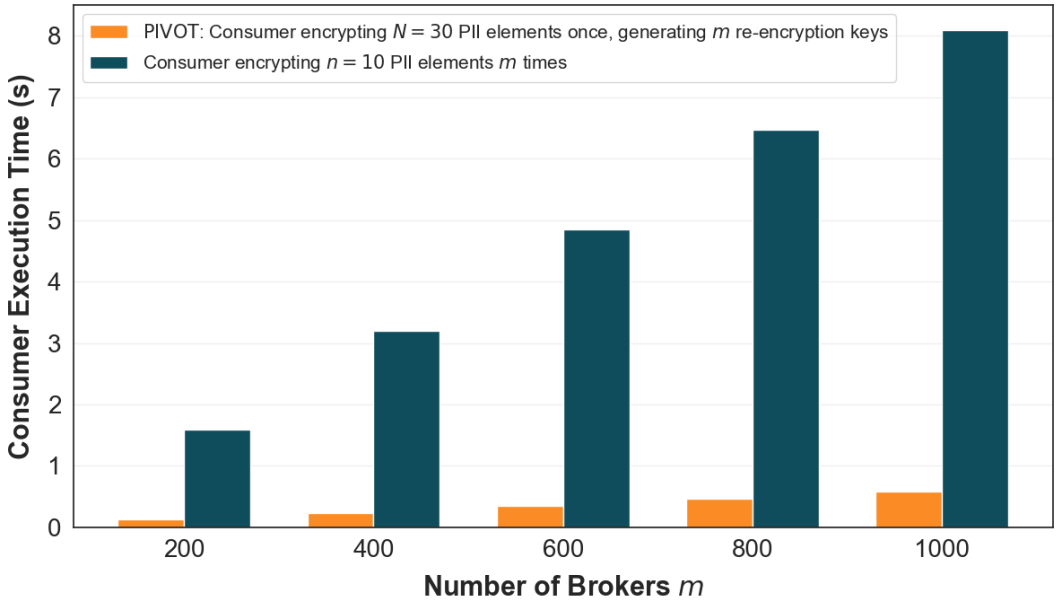


Figure 6.4: Runtime for consumers with PIVOT, compared to individually contacting $\mathcal{DBR}$s

Table 6.2: Performance with Variable # of $\mathcal{DBR}$s

	Execution time (ms)			**Bandwidth sent/received (kB)**		
$\mathcal{BR}$	100	500	1000	100	500	1000
Consumer	79	288	577	27.4/0	80.9/0	147.6/0
Intermediary	6	6	6	6.0/33.3	6.0/86.9	6.0/153.6

Table 6.3: Performance with Variable # of PII Elements

	Execution time (ms)			Bandwidth sent/received (kB)		
# PII elements/$\mathcal{BR}_j$	10	20	30	10	20	30
Intermediary	6	12	18	6.1/86.9	11.6/92.5	17.2/98.0
Broker	8	15	23	6.0/6.1	11.6/11.6	17.1/17.2

6.8.1 Consumer Performance

We first analyze $\mathcal{C}$ execution time by comparing PIVOT against a baseline whereby $\mathcal{C}$ directly encrypts PII values for each $\mathcal{BR}_j$ individually, without using $\mathcal{MED}$. Figure 6.4 shows $\mathcal{C}$ execution time as the number of $\mathcal{BR}$ varies from 200 to 1000. For 600 brokers, PIVOT achieves a **14.2**× speedup, over the naïve $\mathcal{O}(nm)$ approach, completing in 0.34 seconds compared to 4.85 seconds. Table 6.2 shows that $\mathcal{C}$ execution time scales linearly with m, while analysis determined that the overhead of adding one broker is only 0.55 ms. With 500 brokers, $\mathcal{C}$ completes all operations in 288 ms and transmits 80.9 Kbytes. At an upload speed of 3 Mbps[6], this transmission takes approximately 0.2 seconds.

6.8.2 Intermediary Performance

For each $\mathcal{C}$-$\mathcal{BR}_j$ pair, $\mathcal{MED}$'s processing time scales linearly with the number of PII elements requested by $\mathcal{BR}_j$: Tables 6.2 and 6.3 show 6 ms for $n = 10$, 12 ms for $n = 20$, and 18 ms for $n = 30$ regardless of the total number of registered brokers m. Since each interaction is independent, these operations can be parallelized.

Storage: $\mathcal{MED}$ must store encrypted $\mathcal{C}$ data. This amounts to approximately 81 Kbytes per $\mathcal{C}$. Scaling to the entire US adult population of approximately 267 million [84] would require roughly 21 terabytes of storage.

Broker Performance: Similar to $\mathcal{MED}$, $\mathcal{BR}_j$ execution time scales linearly with the

[6]A conservative estimate based on California's broadband adoption data showing that the majority of households meet at least the 25/3 Mbps standard [26].

number of PII elements requested for identity verification. Bandwidth also scales linearly, from 6.0/6.1 Kbytes sent/received for $n = 10$ to 17.1/17.2 Kbytes for $n = 30$.

6.9 Extensions & Variants

This section presents three PIVOT extensions, each strengthening privacy or security guarantees at some additional cost. Each extension is independent and can be applied on its own, depending on the deployment's threat model.

6.9.1 Better Privacy via PLI Variants

For better $\mathcal{C}$ privacy, we can use PLI variants. For instance, [51] introduces PLI-Cardinality (PLI-CA), which only outputs the cardinality of list intersection instead of matching elements. Using PLI-CA, $\mathcal{BR}_j$ would learn only the number of matching elements, thus preventing it from confirming whether its consumer PII database and data collection methods are accurate.

PLI-CA can be easily implemented in PIVOT by having $\mathcal{MED}$ remove labels and shuffle the list $\{a_i\}$ before sending its values to $\mathcal{BR}_j$. In more detail, after computing $\{\ell_i, a_i\}_{i\in[n]}$, $\mathcal{MED}$ randomly permutes $\{a_i\}_{i\in[n]}$ and does not send labels $\{\ell_i\}_{i\in[n]}$, to hide any correspondence between the output values and their order in the list. Then, $\mathcal{BR}_j$ simply outputs $|\{i : \mathsf{ParDec}(sk_{\mathcal{BR}_j}, a_i) = 1_{\mathbb{G}_2}\}| = |\{\mathcal{P}_{\mathcal{C}_j} \cap \mathcal{P}_{\mathcal{BR}_j}\}|$ and obtains the total number of matching elements.

This approach improves privacy in scenarios where knowing which PII elements matched is not necessary for fulfilling a request, and having a high number of matching PII elements is enough to verify $\mathcal{C}$'s identity.

6.9.2 Partial $\mathcal{C}$ Anonymity

Recall that $\mathcal{MED}$ first sends $\mathsf{ID}_{\mathcal{C}}$ to $\mathcal{BR}_j$, who checks whether it possesses $\mathsf{ID}_{\mathcal{C}}$ in its database. However, in the case when $\mathcal{BR}_j$ does not, then the existence of $\mathcal{C}$ is revealed to $\mathcal{BR}_j$ by sharing $\mathsf{ID}_{\mathcal{C}}$. Preventing this would provide partial anonymity to $\mathcal{C}$: only a $\mathcal{BR}_j$ that already knows $\mathsf{ID}_{\mathcal{C}}$ will receive the data request.

CPPA's proposed DROP system [20] hashes identifiers to ensure $\mathcal{C}$ privacy. However, since identifiers are low entropy (e.g., names, and physical or email addresses), $\mathcal{C}$ privacy could be easily compromised.

Instead, we envision the following interaction between $\mathcal{MED}$ and $\mathcal{BR}_j$, starting in Phase 2 of PIVOT. $\mathcal{MED}$ compiles a list $\mathcal{L}_{\mathsf{id}} = \{\mathsf{ID}_{\mathcal{C}}\}_{\mathcal{C} \in \mathcal{C}s}$, where $\mathcal{C}s$ are consumers using $\mathcal{MED}$'s services. $\mathcal{BR}_j$ similarly compiles a list $\mathcal{L}'_{\mathsf{id}} = \{\mathsf{ID}_{\mathcal{C}'}\}_{\mathcal{C}' \in \mathcal{C}s'}$, where $\mathcal{C}s'$ are all the consumers in $\mathcal{BR}_j$'s database. $\mathcal{MED}$ and $\mathcal{BR}_j$ run a standard PSI protocol, with inputs $\mathcal{L}_{\mathsf{id}}$ and $\mathcal{L}'_{\mathsf{id}}$. The PSI protocol returns $\mathcal{L}_{\mathsf{id}} \cap \mathcal{L}'_{\mathsf{id}}$ to $\mathcal{MED}$. Thus, $\mathcal{MED}$ will only make data requests on behalf of $\{\mathsf{ID}_{\mathcal{C}}\}_{\mathcal{C} \in \mathcal{L}_{\mathsf{id}} \cap \mathcal{L}'_{\mathsf{id}}}$ to $\mathcal{BR}_j$, such that $\mathcal{BR}_j$ does not learn $\mathsf{ID}_{\mathcal{C}}$ if $\mathcal{C}$ is not in its database.

Since the databases of both $\mathcal{MED}$ and $\mathcal{BR}_j$ may be very large, a single PSI execution can incur high computation and communication overhead. To lower costs, instead of doing it in real time, this process can be performed asynchronously and periodically, e.g., every day, week, or month. This follows current and proposed laws, e.g., the California DROP regulation stipulates that brokers must consult the state-run platform every 45 days.

6.9.3 Malicious $\mathcal{MED}$ & $\mathcal{BR}_j$

PIVOT can be easily amended to tolerate malicious $\mathcal{MED}$ or $\mathcal{BR}_j$. We define malicious behavior as any deviation from the scheme. As mentioned in Section 6.4.2, collusion between $\mathcal{BR}_j$ and $\mathcal{MED}$ as well as veracity of $\mathcal{BR}_j$'s final response are out of scope.

$\mathcal{C}$ privacy against malicious $\mathcal{MED}$. Since $\mathcal{C}$ only retrieves $\mathcal{BR}$ information from $\mathcal{MED}$,

it suffices to include certificates along with public keys. $\mathcal{C}$ can verify these certificates before proceeding.

$\mathcal{BR}_j$ privacy against malicious $\mathcal{MED}$. $\mathcal{MED}$ computes proofs of the form:

$$\pi_{a_i} = PoK\{w_i : a_i^{(\alpha)} = (Z^{b_2 r_i} Z^{b_2 \tilde{r}_i})^{w_i}\} \; \forall \; i \in [1, n]$$

and appends them to its second message to $\mathcal{BR}_j$.

$\mathcal{C}$ privacy against malicious $\mathcal{BR}_j$. $\mathcal{BR}_j$ appends zero-knowledge proofs of the form:

$$\pi_{\tilde{c}_i} = PoK\{\tilde{r}_i, y_i : \tilde{c}_i = (Z^{b_2 \tilde{r}_i}, Z^{y_i} Z^{\tilde{r}_i})\} \; \forall \; i \in [1, n]$$

to its message to $\mathcal{MED}$.

Security Proofs Outlines

Privacy against malicious $\mathcal{MED}$. For a malicious $\mathcal{MED}$, a simulator $\mathsf{SIM}_{\mathcal{MED}}$ verifies π_{a_i}, extracts w_i for all $i \in [n]$, and computes $D = (a_i^{(\beta)})^{1/w_i}$. $\mathsf{SIM}_{\mathcal{MED}}$ computes $Z^{-x_i} Z^{r_i} \cdot Z^{y_i} Z^{\tilde{r}_i}$ from values received from $\mathcal{C}$ and $\mathcal{BR}_j$ and checks whether it equals D. If yes, $\mathsf{SIM}_{\mathcal{MED}}$ sets $A_i = a_i$, and otherwise $A_i \leftarrow \$$, and sends all values to $\mathcal{BR}_j$. Thus the simulated and real views are indistinguishable.

Privacy against malicious $\mathcal{BR}_j$. For a malicious $\mathcal{BR}_j$, a simulator $\mathsf{SIM}_{\mathcal{BR}_j}$ extracts $\tilde{r}_i, y_i$. The proof then follows the same reasoning as for a malicious $\mathcal{C}$.

6.10 Conclusions

This chapter presented PIVOT that enables efficient and privacy-preserving consumer requests to data brokers. By combining proxy re-encryption with additively homomorphic

encryption, PIVOT reduces consumer effort to a single submission that scales to all registered data brokers. PIVOT guarantees strong privacy: neither the intermediary nor data brokers acquire any new consumer PII. PIVOT enables consumers to efficiently and comprehensively exercise their statutorily granted privacy rights.

Chapter 7

Conclusions

This dissertation examines the privacy issues that arise when consumers submit requests under privacy regulations, with a specific focus on requests made to data brokers.

Chapter 3 presented the first comprehensive evaluation of data brokers' responses to data access requests. All data brokers registered in California were sent requests to know which data was collected, under the CCPA. We found that 43% of brokers failed to respond to these requests, in non-compliance with the CCPA. In addition, brokers asked for varying amounts of PII for identity verification, including sensitive PII, yet only 5% reported having collected any personal information about the requesting consumer. Consumers must therefore give up personal information freely to even attempt a data request, and for the large majority of brokers, that information is learnt for the first time through the request. As shown, a substantial number of brokers never responded at all, even after being given highly sensitive PII: exercising one's consumer rights, meant to obtain more control over personal information, instead introduces a worrying threat to their privacy.

Chapter 4 follows up with a study of the more commonly exercised opt-out and deletion rights, again under the CCPA. Requests were once more submitted to all registered brokers,

revealing the same concerning pattern of non-compliance: more than 70% never answered deletion requests, and 22% explicitly demanded identity verification during opt-out requests, despite being prohibited from doing so. Across both studies, we observed that a consumer wishing to submit requests to every broker must individually send a form or email to each one, with each broker asking for different information – an extremely time-consuming process for the consumer.

Chapter 5 proposes four protocols that make this identity verification process privacy-preserving for consumers. Using ElGamal encryption, we build a variant of private set intersection, private list intersection (PLI), which exploits the fact that the PII required during identity verification consists of labeled, ordered lists. We also propose three variants: PLI-cardinality (PLI-CA), threshold-PLI (t-PLI), and threshold PLI-cardinality (t-PLI-CA). Each variant offers stronger privacy by revealing less information in its output. We implement these protocols and demonstrate their efficiency.

Chapter 6 uses PLI as a foundational building block. We introduce PIVOT, which addresses both the privacy risks of identity verification and the burden of contacting every broker individually. PIVOT allows consumers to make a single request to reach all brokers through an intermediary. Using proxy re-encryption, the intermediary itself never learns the consumer's PII. This protocol can further be extended, using private set intersection, to prevent brokers from learning the identifier of consumers they do not already know. The resulting system is implemented and proved efficient for the consumer and practical for brokers to deploy.

As mentioned in Chapter 4, a natural direction for future work is to evaluate the California DROP platform once it has been established and operating for at least a few months: do brokers comply more readily through this system, and does it introduce any new security or privacy risks of its own?

Another direction is to further improve PLI for identity verification, for instance including

techniques from fuzzy PSI, i.e. accommodating typos either by the consumer or in the broker's database, which could improve the success rate of identity verification. Finally, the robustness of the verification process against impersonators could be strengthened using personal information certified by a trusted third party.

Bibliography

[1] 82nd OREGON LEGISLATIVE ASSEMBLY. House Bill 2052. https://olis.oregonlegislature.gov/liz/2023R1/Downloads/MeasureDocument/HB2052/Enrolled.

[2] Repository for PIVA, 2023. https://github.com/zane-a-karl/PLI.

[3] A. Abadi, S. Terzis, and C. Dong. Feather: Lightweight multi-party updatable delegated private set intersection. *Cryptology ePrint Archive*, 2020.

[4] A. Abadi, S. Terzis, R. Metere, and C. Dong. Efficient delegated private set intersection on outsourced private datasets. *IEEE Transactions on Dependable and Secure Computing*, Volume 16, Number 4, 2017.

[5] Abine. DeleteMe: Remove Your Personal Info from Search Results. https://joindeleteme.com/. Accessed: 11-13-2025.

[6] S. Adhatarao, C. Lauradoux, and C. Santos. Why IP-based Subject Access Requests Are Denied? *arXiv preprint arXiv:2103.01019*, 2021.

[7] J. A. Akinyele, M. D. Green, and A. D. Rubin. Charm: A Framework for Rapidly Prototyping Cryptosystems. Cryptology ePrint Archive, Paper 2011/617, 2011.

[8] G. Anthes. Data brokers are watching you. *Communications of the ACM*, Volume 58, Number 1, 2014.

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. In *Proc. Network and Distributed System Security Symposium*, 2005.

[10] Aura. Aura: All-In-One Digital Safety. https://buy.aura.com/. Accessed:11-13-2025.

[11] E. R. Berlekamp and L. R. Welch. Error correction for algebraic block codes, Dec 1986.

[12] C. Boniface, I. Fouad, N. Bielova, C. Lauradoux, and C. Santos. Security analysis of subject access request procedures: How to authenticate data subjects safely when they request for their data. In *Privacy Technologies and Policy: 7th Annual Privacy Forum, APF 2019*, pages 182–209, 2019.

[13] A. Borem, E. Pan, O. Obielodan, A. Roubinowitz, L. Dovichi, M. L. Mazurek, and B. Ur. Data subjects' reactions to exercising their right of access. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2865–2882, 2024.

[14] L. Bufalieri, M. L. Morgia, A. Mei, and J. Stefa. GDPR: When the Right to Access Personal Data Becomes a Threat. In *2020 IEEE International Conference on Web Services, ICWS 2020, Beijing, China, October 19-23, 2020*, pages 75–83. IEEE, 2020.

[15] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky. How good are humans at solving CAPTCHAs? A large scale evaluation. In *2010 IEEE symposium on security and privacy*, pages 399–413. IEEE, 2010.

[16] California Legislature. California Code of Regulations. https://govt.westlaw.com/calregs/Index?transitionType=Default&contextData=%28sc.Default%29.

[17] California Legislature. TITLE 1.81.5. California Consumer Privacy Act of 2018, 2018.

[18] California Legislature. TITLE 1.81.48. Data Broker Registration, 2019.

[19] California Privacy Protection Agency. Data Broker Registry. https://cppa.ca.gov/data_broker_registry/.

[20] California Privacy Protection Agency. Proposed Regulations on Accessible Deletion Mechanism. https://cppa.ca.gov/regulations/drop.html. Accessed:11-13-2025.

[21] California Privacy Protection Agency. California privacy protection agency complaint form, 2025. Accessed: 2025-06-05.

[22] California Privacy Protection Agency. CPPA orders Florida data broker to pay fine. https://cppa.ca.gov/announcements/2025/20250508.html, May 2025.

[23] California Privacy Protection Agency. Data Broker Promoting Ability to Dig Up "Scary" Amounts of Information Agrees to Shut Down, Feb 2025.

[24] California Privacy Protection Agency. Washington Data Broker Agrees to Pay Fine for Failing to Register. https://cppa.ca.gov/announcements/2025/20250729.html, Jul 2025.

[25] California Privacy Protection Agency. Ford to change practices, pay fine for adding unnecessary friction to opt-out process, Mar 2026.

[26] California Public Utilities Commission. Home Broadband Adoption Report. https://docs.cpuc.ca.gov/PublishedDocs/Efile/G000/M563/K203/563203680.PDF, 2025. Accessed: 11-13-2025.

[27] California State Legislature. SB-362: California Delete Act of 2023. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=202320240SB362, 2023. Accessed:11-13-2025.

[28] CalPrivacy. Delete request and opt-out platform (DROP). https://consumer.drop.privacy.ca.gov/. Accessed:11-13-2025.

[29] J. Camenisch and M. Stadler. Proof systems for general statements about discrete logarithms. In *Technical Report/ETH Zurich, Dept. of Computer Science*, 1997.

[30] R. Cramer, R. Gennaro, and B. Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *Advances in Cryptology — EUROCRYPT '97*, 1997.

[31] Data Brokers Watch. Databrokerswatch.org. https://databrokerswatch.org/.

[32] E. De Cristofaro, P. Gasti, and G. Tsudik. Fast and private computation of cardinality of set intersection and union. In *CANS*, 2012.

[33] M. Di Martino, P. Robyns, W. Weyts, P. Quax, W. Lamotte, and K. Andries. Personal Information Leakage by Abusing the {GDPR}'Right of Access'. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pages 371–385, 2019.

[34] I. Dodds. You may not know them, but they certainly know you: Inside the dangerous and shady business of data brokers, Feb 2026.

[35] T. Duong, D. H. Phan, and N. Trieu. Catalic: Delegated PSI cardinality with applications to contact tracing. In *Eurocrypt*, 2020.

[36] European Data Protection Board. Guidelines 01/2022 on data subject rights - Right of access, Version 2.0, 2023.

[37] European Parliament and Council. Regulation (eu) 2016/679, general data protection regulation, 2016.

[38] R. A. Fisher and F. Yates. *Statistical tables for biological, agricultural, and medical research*. Hafner Publishing Company, 1953.

[39] M. Funk. The Man Who Trapped Us in Databases. https://www.nytimes.com/2023/09/22/magazine/hank-asher-data.html, 2023. Accessed: 11-13-2025.

[40] J. Furukawa. Efficient and Verifiable Shuffling and Shuffle-Decryption. *IEICE Transactions*, 2005.

[41] T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory*, 1985.

[42] General Assembly of the State of Vermont. No. 171. An act relating to data brokers and consumer protection. https://legislature.vermont.gov/Documents/2018/Docs/ACTS/ACT171/ACT171%20As%20Enacted.pdf.

[43] S. Ghosh and M. Simkin. The Communication Complexity of Threshold Private Set Intersection. In *CRYPTO*, 2019.

[44] O. Goldreich. *Foundations of Cryptography, Volume 2*. Cambridge university press Cambridge, 2004.

[45] Governo do Brasil. LGPD - Lei Geral de Proteção de Dados, 2023. Accessed: [5/28/2025].

[46] J. Groth. A Verifiable Secret Shuffle of Homomorphic Encryptions. Cryptology ePrint Archive, Paper 2005/246, 2005.

[47] J. He, P. Snyder, H. Haddadi, F. E. Bustamante, and G. Tyson. Measuring the Accuracy and Effectiveness of PII Removal Services. *Proc. Priv. Enhancing Technol.*, 2025(4):166–182, 2025.

[48] A. Heinrich, M. Hollick, T. Schneider, M. Stute, and C. Weinert. PrivateDrop: Practical Privacy-Preserving Authentication for Apple AirDrop. In *USENIX Security*, 2021.

[49] D. Herrmann and J. Lindemann. Obtaining personal data and asking for erasure: do app vendors and website owners honour your privacy rights? In M. Meier, D. Reinhardt, and S. Wendzel, editors, *Sicherheit 2016: Sicherheit, Schutz und Zuverlässigkeit, Beiträge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 5.-7. April 2016, Bonn*, volume P-256 of *LNI*, pages 149–160. GI, 2016.

[50] J. Hu, Z. Liu, and C. Zuo. Delegated multi-party private set intersection from secret sharing. *Cryptology ePrint Archive*, 2025.

[51] S. Hwang, S. Jarecki, Z. Karl, E. van Kempen, and G. Tsudik. PIVA: Privacy-Preserving Identity Verification Methods for Accountless Users via Private List Intersection and Variants. In *Proc. European Symposium on Research in Computer Security*, 2024.

[52] S. Jordan, Y. Nakatsuka, E. Ozturk, A. Paverd, and G. Tsudik. VICEROY: GDPR-/CCPA-compliant Enforcement of Verifiable Accountless Consumer Requests. In *Proc. Network and Distributed System Security Symposium*, 2023.

[53] M. Keller. MP-SPDZ: A Versatile Framework for Multi-Party Computation. In *ACM CCS*, 2020.

[54] F. Kerschbaum. Collusion-resistant outsourcing of private set intersection. In *Proc. Annual ACM Symposium on Applied Computing*, 2012.

[55] J. L. Kröger, J. Lindemann, and D. Herrmann. How do app vendors respond to subject access requests?: a longitudinal privacy study on iOS and Android Apps. In M. Volkamer and C. Wressnegger, editors, *ARES 2020: The 15th International Conference on Availability, Reliability and Security, Virtual Event, Ireland, August 25-28, 2020*, pages 10:1–10:10. ACM, 2020.

[56] F. Liu, W. K. Ng, W. Zhang, D. H. Giang, and S. Han. Encrypted set intersection protocol for outsourced datasets. In *Proc. IEEE International Conference on Cloud Engineering*, 2014.

[57] Login.gov. https://login.gov/.

[58] M. D. Martino, I. Meers, P. Quax, K. Andries, and W. Lamotte. Revisiting Identification Issues in GDPR 'Right Of Access' Policies: A Technical and Longitudinal Analysis. *Proc. Priv. Enhancing Technol.*, 2022(2):95–113, 2022.

[59] M. D. Martino, P. Robyns, W. Weyts, P. Quax, W. Lamotte, and K. Andries. Personal Information Leakage by Abusing the GDPR 'Right of Access'. In H. R. Lipford, editor, *Fifteenth Symposium on Usable Privacy and Security, SOUPS 2019, Santa Clara, CA, USA, August 11-13, 2019*. USENIX Association, 2019.

[60] M. Marx, E. Zimmer, T. Mueller, M. Blochberger, and H. Federrath. Hashing of personally identifiable information is not sufficient. In *Sicherheit*, 2018.

[61] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, D. Boneh, et al. Location privacy via private proximity testing. In *NDSS*, volume 11, 2011.

[62] A. Ng. Alleged shooter found minnesota lawmakers' addresses online, court docs say, Jun 2025.

[63] OneTrust LLC. OneTrust. https://www.onetrust.com/.

[64] Optery. https://www.optery.com/.

[65] E. Pagnin, G. Gunnarsson, P. Talebi, C. Orlandi, and A. Sabelfeld. TOPPool: Time-aware optimized privacy-preserving ridesharing. *Cryptology ePrint Archive*, 2021.

[66] J. Pavur and C. Knerr. Gdparrrrr: Using privacy laws to steal identities. *arXiv preprint arXiv:1912.00731*, 2019.

[67] B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai. PSI from PaXoS: fast, malicious private set intersection. In *Eurocrypt*, 2020.

[68] Privacy Rights Clearinghouse. Why are hundreds of data brokers not registering with states?

[69] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 1960.

[70] P. Rindal and P. Schoppmann. VOLE-PSI: fast OPRF and circuit-PSI from vector-OLE. In *Eurocrypt*, 2021.

[71] M. Rosulek and N. Trieu. Compact and Malicious Private Set Intersection for Small Sets. In *ACM CCS*, 2021.

[72] N. Samarin, S. Kothari, Z. Siyed, O. Bjorkman, R. Yuan, P. Wijesekera, N. Alomar, J. Fischer, C. J. Hoofnagle, and S. Egelman. Lessons in VCR Repair: Compliance of Android App Developers with the California Consumer Privacy Act (CCPA). *Proc. Priv. Enhancing Technol.*, 2023(3):103–121, 2023.

[73] A. Searles, Y. Nakatsuka, E. Ozturk, A. Paverd, G. Tsudik, and A. Enkoji. An empirical study & evaluation of modern {CAPTCHAs}. In *32nd usenix security symposium (usenix security 23)*, pages 3081–3097, 2023.

[74] A. Shamir. How to Share a Secret. *Commun. ACM*, 1979.

[75] J. Sherman. The data brokers selling us data to foreign actors, according to california, Mar 2026.

[76] Surfshark. Incogni: Automatic Personal Data Removal. https://incogni.com/. Accessed: 11-13-2025.

[77] K. Take, K. Gallagher, A. Forte, D. McCoy, and R. Greenstadt. "It Feels Like Whack-a-mole": User Experiences of Data Removal from People Search Websites. *Proc. Priv. Enhancing Technol.*, 2022.

[78] K. Take, J. Young, R. Bhalerao, K. Gallagher, A. Forte, D. McCoy, and R. Greenstadt. What to Expect When You're Accessing: An Exploration of User Privacy Rights in People Search Websites. *Proc. Priv. Enhancing Technol.*, 2024(4):311–326, 2024.

[79] S. Tewari and F. Walter-Johnson. New records detail DHS purchase and use of vast quantities of cell phone location data.

[80] Texas Business And Commerce Code. Chapter 510. data brokers. https://statutes.capitol.texas.gov/?tab=1&code=BC&chapter=BC.510&artSec=.

[81] N. Trieu, K. Shehata, P. Saxena, R. Shokri, and D. Song. Epione: Lightweight contact tracing with strong privacy. *arXiv preprint arXiv:2004.13293*, 2020.

[82] T. Urban, D. Tatang, M. Degeling, T. Holz, and N. Pohlmann. The unwanted sharing economy: An analysis of cookie syncing and user transparency under GDPR. *arXiv preprint arXiv:1811.08660*, 2018.

[83] T. Urban, D. Tatang, M. Degeling, T. Holz, and N. Pohlmann. A Study on Subject Data Access in Online Advertising After the GDPR. In C. Pérez-Solà, G. Navarro-Arribas, A. Biryukov, and J. García-Alfaro, editors, *Data Privacy Management, Cryptocurrencies and Blockchain Technology - ESORICS 2019 International Workshops, DPM 2019 and CBT 2019, Luxembourg, September 26-27, 2019, Proceedings*, volume 11737 of *Lecture Notes in Computer Science*, pages 61–79. Springer, 2019.

[84] U.S. Census Bureau. United States Population by Age and Sex. https://www.census.gov/popclock/data_tables.php?component=pyramid, 2024. Accessed: 11-13-2025.

[85] US House of Representatives. SECURE Data Act. https://financialservices.house.gov/uploadedfiles/secure_data_act.pdf.

[86] E. van Kempen, I. Bagayatkar, P. Frolikov, C. Georgiou, and G. Tsudik. Consumer Beware! Exploring Data Brokers' CCPA Compliance. *IEEE S&P 2026*, 2026.

[87] E. van Kempen, P. Frolikov, and C. Georgiou. Source code for PIVOT. https://github.com/y76/IV-Drip.

[88] S. Weisman. Data brokers fuel scams: Senate report on billions of consumer losses, Mar 2026.

[89] J. Wong and T. Henderson. The right to data portability in practice: exploring the implications of the technologically neutral GDPR. *International Data Privacy Law*, 9(3):173–191, 2019.

[90] A. C. Yao. How to Generate and Exchange Secrets (Extended Abstract). In *27th Annual Symposium on Foundations of Computer Science*, 1986.

[91] H. Zhao, W. Zhang, X. Li, S. Shang, K. Huang, and X. Zhang. Towards efficient delegated private set intersection cardinality protocol. In *Proc. International Conference on Computer Supported Cooperative Work in Design*, 2024.

[92] Y. Zhao and S. S. Chow. Are you the one to share? Secret transfer with access structure. *PETS*, 2017.

[93] Y. Zhao and S. S. M. Chow. Can You Find The One for Me? In *WPES*, 2018.

[94] Q. Zheng and S. Xu. Verifiable delegated set intersection operations on outsourced encrypted data. In *Proc. IEEE International Conference on Cloud Engineering*, 2015.