Fall 10-1-2021

Data exploitation and privacy protection in the era of data sharing

Lihi Idan
Yale University Graduate School of Arts and Sciences, lihi.idan@gmail.com

Abstract

Data Exploitation and Privacy Protection in the Era of Data Sharing

Lihi Idan

2021

As the amount, complexity, and value of data available in both private and public sectors has risen sharply, the competing goals of data privacy and data utility have challenged both organizations and individuals. This dissertation addresses both goals.

First, we consider the task of *interorganizational data sharing*, in which data owners, data clients, and data subjects have different and sometimes competing privacy concerns. A key challenge in this type of scenario is that each organization uses its own set of proprietary, intraorganizational attributes to describe the shared data; such attributes cannot be shared with other organizations. Moreover, data-access policies are determined by multiple parties and may be specified using attributes that are not directly comparable with the ones used by the owner to specify the data.

We propose a system architecture and a suite of protocols that facilitate dynamic and efficient interorganizational data sharing, while allowing each party to use its own set of proprietary attributes to describe the shared data and preserving confidentiality of both data records and attributes. We introduce the novel technique of *attribute-based encryption with oblivious attribute translation (OTABE)*, which plays a crucial role in our solution and may prove useful in other applications. This extension of attribute-based encryption uses semi-trusted proxies to enable dynamic and oblivious translation between proprietary attributes that belong to different organizations. We prove that our OTABE-based framework is secure in the standard model and provide two real-world use cases.

Next, we turn our attention to utility that can be derived from the vast and growing amount of data about individuals that is available on social media. As social networks

(SNs) continue to grow in popularity, it is essential to understand what can be learned about personal attributes of SN users by mining SN data.

The first SN-mining problem we consider is how best to predict the voting behavior of SN users. Prior work only considered users who generate politically oriented content or voluntarily disclose their political preferences online. We avoid this bias by using a novel type of Bayesian-network (BN) model that combines demographic, behavioral, and social features. We test our method in a predictive analysis of the 2016 U.S. Presidential election. Our work is the first to take a semi-supervised approach in this setting. Using the Expectation-Maximization (EM) algorithm, we combine labeled survey data with unlabeled Facebook data, thus obtaining larger datasets and addressing self-selection bias.

The second SN-mining challenge we address is the extent to which Dynamic Bayesian Networks (DBNs) can infer dynamic behavioral intentions such as the intention to get a vaccine or to apply for a loan. Knowledge of such intentions has great potential to improve the design of recommendation systems, ad-targeting mechanisms, public-health campaigns, and other social and commercial endeavors. We focus on the question of how to infer an SN user's *offline* decisions and intentions using only the *public* portions of her *online* SN accounts.

Our contribution is twofold. First, we use BNs and several behavioral-psychology techniques to model decision making as a complex process that both influences and is influenced by static factors (such as personality traits and demographic categories) and dynamic factors (such as triggering events, interests, and emotions). Second, we explore the extent to which temporal models may assist in the inference task by representing SN users as sets of DBNs that are built using our modeling techniques. The use of DBNs, together with data gathered in multiple waves, has the potential to improve both inference accuracy and prediction accuracy in future time slots. It may also shed light on the extent to which different factors influence the decision-making process.

Data Exploitation and Privacy Protection in the Era of Data Sharing

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Lihi Idan

Dissertation Director: Joan Feigenbaum

December 2021

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Joan Feigenbaum, for her thoughtful guidance and for letting me choose my own path and develop my personal research interests. I would also like to thank Professor David Gelernter for fascinating teaching experiences and for great conversations that led me to some of the ideas developed in this thesis.

The research in this dissertation was supported in part by grants CNS-1409599 and CNS-1407454 from the US National Science Foundation and grant 2016-3834 from the William and Flora Hewlett Foundation.

This dissertation is dedicated to my dear father, Dr. Asher Idan, for his support, his encouragement, and his eternal optimism.

Chapter 1

Introduction

In many rule-of-law societies, citizens have a right to privacy. Even in the US, where the Bill of Rights does not contain the word "privacy," the Fourth Amendment[1] prohibits unreasonable government intrusions into citizens' homes and belongings, and it has led to many court decisions that are privacy enhancing. As the number of news stories about hacks and data breaches clearly demonstrates, people are very concerned about loss of control over their personal information.

As the Internet (and, in particular, Big Tech firms) have become dominant in citizens' daily lives, however, privacy has ceased to be a right and became a commodity. Unlike most commodities, privacy is not bought with money but rather with limited or even no ability to use common and essential services. In order to use Facebook, Uber, or even a credit card, people reveal personal data that are subsequently shared with other organizations without their explicit permission. Sometimes people explicitly *agree* that service providers may share their personal information with other organizations, because it is the only obvious way to complete transactions that they need to complete in order to get on with their daily lives. In both of these common scenarios, the information revealed al-

[1] "The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized."

lows some of our most personal attributes to be inferred. It has been argued that people's willingness to use services that collect and share their personal information means that they accept the privacy risks entailed in doing so and regard them simply as the price of receiving service for which they do not have to pay money. We question the soundness of this argument, because many people do not know what information they routinely reveal or with whom it is shared.

For example, healthcare providers create electronic medical records (EMRs) for all patients and share them with other organizations for various purposes. Do patients know with whom their EMRs are shared and why? The Health Insurance Portability and Accountability Act (HIPAA [1]) limits access to EMRs by parties outside of the healthcare organizations that create them. However, it imposes very weak restrictions on access by those inside of the organizations; specifically, HIPAA requires only that internal access be granted on a "need-to-know basis." This requirement is routinely understood to allow internal access not only by doctors and nurses, as may seem intuitive to many patients, but by many other types of in-house employees such as administrators and IT-staff members. Limits on external access do not prevent sharing with "business associates" such as billing companies, insurance companies, collections agencies, practice-management companies, IT consultants, and many more.

It is also unclear how seriously and consistently HIPAA rules are enforced. Numerous HIPAA violations are reported every year.[2] The actual number of violations may be significantly higher. Some violations go unnoticed, and some may not be reported because of fear of lawsuits or bad publicity.

Another domain in which people may overestimate the extent to which their personal information is protected is that of credit reports. The Fair Credit Reporting Act (FCRA [36]) empowers consumer-reporting agencies such as Experian and Equifax to

[2]See, *e.g.*, the example of this Ohio hospital: `https://healthitsecurity.com/news/ohio-hospital-hipaa-violation-goes-unnoticed-for-over-a-decade`.

collect and maintain financial information about individuals and to create credit reports that lenders and other merchants can use to gauge individuals' creditworthiness. Contrary to many people's intuitive understanding, the FCRA allows consumer-reporting agencies to grant access to a person's credit report without that person's explicit consent, as long as the report is to be used for a "permissible purpose." This term is not defined, and the rule is not enforced in a uniform, global manner – a state of affairs that has led to several serious data breaches and privacy violations [16, 37].

Perhaps the most extreme example of the commodified nature of privacy occurs on social networks (SNs), where people voluntarily share a lot of personal information. In principle, the Cambridge Analytica scandal [2] made people aware of the fact that the information that they explicitly share on SNs can be given to the SNs' "partners," including but not limited to advertisers; one could argue, therefore, that active SN users accept this practice. However, it is unlikely that they know the extent to which information that they voluntarily publish (as well as their decisions *not* to publish certain information) can be used to infer their private decisions and intentions.

This dissertation addresses two complementary challenges posed by our commodified information environment: First, we provide a data-sharing framework that protects the privacy of data owners, data users, and data subjects; second, we provide SN-mining techniques that enable governments, companies, and other organizations to infer SN users' offline decisions and intentions from the users' publicly observable SN data.

1.1 Dissertation overview

In Chapter 2, we consider the task of interorganizational data sharing, in which data owners, data clients, and data subjects have different and sometimes competing privacy concerns. One real-world scenario in which this problem arises concerns law-enforcement use of phone-call metadata: The data owner is a phone company, the data clients are law-

enforcement agencies, and the data subjects are individuals who make phone calls. A key challenge in this type of scenario is that each organization uses its own set of proprietary intraorganizational attributes to describe the shared data; such attributes cannot be shared with other organizations. Moreover, data-access policies are determined by multiple parties and may be specified using attributes that are not directly comparable with the ones used by the owner to specify the data.

We propose a system architecture and a suite of protocols that facilitate dynamic and efficient interorganizational data sharing, while allowing each party to use its own set of proprietary attributes to describe the shared data and preserving confidentiality of both data records and proprietary intraorganizational attributes. We introduce the novel technique of *attribute-based encryption with oblivious attribute translation (OTABE)*, which plays a crucial role in our solution. This extension of attribute-based encryption uses semi-trusted proxies to enable dynamic and oblivious translation between proprietary attributes that belong to different organizations; it supports hidden access policies, direct revocation, and fine-grained, data-centric keys and queries. We prove that our OTABE-based framework is secure in the standard model and provide two real-world use cases.

The material in Chapter 2 has been published in [58, 59].

In Chapter 3, we introduce our novel approach to SN mining. Increasing use of social media in political campaigns raises the question of whether one can predict the voting behavior of SN users. Prior work on this problem considered only the minority of users who generate politically oriented content or voluntarily disclose their political preferences online, thus introducing substantial bias. We avoid this bias by using a novel Bayesian-network (BN) model that combines demographic, behavioral, and social features. We test our approach in a predictive analysis of the 2016 US Presidential election. Our model is highly extensible and facilitates the use of incomplete datasets. Furthermore, our work is the first to apply a semi-supervised approach to this task: Using an expectation-maximization (EM) algorithm, we combine labeled survey data with unlabeled Facebook

data, thus obtaining larger datasets as well as addressing self-selection bias.

The material in Chapter 3 has been published in [57].

In Chapter 4, we generalize our BN approach to SN mining by asking what can be learned about personal attributes of SN users (beyond their voting behavior) by mining their publicly observable SN data. Previous work in this area focused on the inference of time-invariant attributes such as personality, demographic categories, and ideology. By contrast, we study the extent to which BNs and Dynamic Bayesian Networks (DBNs) can infer *dynamic behavioral intentions* in areas such as health care and finance. Knowledge of such intentions has great potential to improve the design of recommendation systems, ad-targeting mechanisms, public-health campaigns, and other social and commercial endeavors.

Our contribution to this type of intention inference is twofold. First, we use BNs and several behavioral-psychology techniques to model decision making as a complex process that both influences and is influenced by static factors (such as personality traits and demographic categories) and dynamic factors (such as triggering events, interests, and emotions). Second, we explore the extent to which temporal models may assist in the inference task by representing SN users as sets of DBNs that are built using our modeling techniques. The use of DBNs, together with data gathered in multiple waves, has the potential to improve both inference accuracy and prediction accuracy in future time slots. It may also shed light on the extent to which different factors influence the decision-making process.

The material in Chapter 4 has been submitted for publication and is under review.

Finally, in Chapter 5, we present conclusions and directions for further research.

1.2 Technical background

In this dissertation, we make extensive use of three technical building blocks: attribute-based encryption, Bayesian networks, and dynamic Bayesian networks. We give brief introductions to these building blocks here and provide pointers to more extensive treatments.

1.2.1 Attribute-based encryption

An attribute-based encryption (ABE) scheme is a type of public-key encryption scheme. In an ABE scheme, plaintexts are encrypted under access policies that are composed of attributes. In addition, potential decryptors receive secret keys, composed of attributes as well; typically, these are *user-centric* attributes in the sense that they represent decryptors' identities or roles within the enterprise that is using the ABE scheme. For instance: IS-IN-CS-DEPARTMENT, CLEARANCE-LEVEL=HIGHEST, *etc*. One or more *trusted authorities* (TAs) issue secret keys to potential decryptors that are based on the decryptors' attribute sets. A user will be able to decrypt a ciphertext if and only if the set of attributes that her key represents satisfies the access policy under which the plaintext was encrypted. This is called *ciphertext-policy ABE*.

By contrast, *key-policy ABE* works in reverse: Users' keys are based on access policies, and plaintexts are encrypted under sets of attributes. A key-policy ABE scheme that involves multiple TAs is called a *multi-authority, key-policy ABE* (MA-KP-ABE) scheme.

MA-KP-ABE is our technical starting point in PRShare, a framework for privacy-preserving, interorganizational data sharing that we present in Chapter 2. We start with the single-authority KP-ABE scheme due to Rouselakis and Waters [96] and enhance it with novel, oblivious-translation capabilities, thus introducing a new type of encryption scheme: *attribute-based encryption with oblivious attribute translation* (OTABE).

Further information about ABE can be found in [17, 47].

1.2.2 Bayesian networks

A *Bayesian network* (BN) is a directed graphical model that captures a subset of the independence relations of a given joint probability distribution. Each BN is represented as a directed acyclic graph (DAG), where nodes in the graph represent random variables and edges represent statistical dependencies between variables. Specifically, a key property of BNs is that each variable is conditionally independent of its non-descendants, given its parents in the network. The parameters of a BN are all of the nodes' conditional probability distributions, which are often represented as tables (cpts). Cpts quantify the effects of the node's parents on each of the nodes' states.

The main objective of a BN is to model the posterior conditional probability distribution of a variable or a set of variables of interest after observing new data ("evidence"). BNs may be constructed either manually, using knowledge of the underlying domain (*e.g.* "priors"), or automatically using domain-specific datasets.

In Chapters 3 and 4, we exploit some of the main advantages of BNs for the purpose of inferring different behavioral intentions of SN users. BNs are highly suitable for this purpose, because they naturally encode relations between different variables, rather than focusing solely on relations between features and the target variable. This property is essential when dealing with SN data or, more generally, with a data set that contains many correlations among features. Furthermore, BNs allow us to incorporate prior information in parameters and to learn parameters from both data and priors.

Further information about BNs can be found in [54, 63].

1.2.3 Dynamic Bayesian networks

BNs are static models and cannot represent dynamic processes. A *dynamic Bayesian network* (DBN) is a sequence of BNs that adds three components: temporal variables, temporal edges, and temporal evidence. The i^{th} BN in the sequence represents the i^{th} *time slice* of the DBN. In order to build a DBN, one must specify both its intra-slice structure and corresponding cpts and its inter-slice cpts. Intra-slice cpts represent dependency relations within a single time slice. Inter-slice cpts represent temporal dependency relations, *i.e.*, relations between variables from different time slices.

DBNs can model temporal relations between different variables and thus are capable of capturing influences over time; this makes them excellent tools for modeling time-varying dependencies. Furthermore, DBNs are more versatile than BNs in terms of the queries that they support and can be used to conduct richer types of analysis. These types include inference of one or more variables of interest, prediction of future states of one or more variables, identification of key determinants of different variables in the network, and backward reasoning and retroactive analysis of historical values of variables. By enabling the analyst to assess how a given variable changes over time, DBNs provide a capability that static BNs cannot.

In Chapter 4, we present a new conceptual model of SN users that is based on DBNs. We go on to develop a new methodology for inference of dynamic attributes in SN settings. We demonstrate the usefulness of this conceptual representation on five dynamic attributes, each of which is a different behavioral intention.

Further information about DBNs can be found in [84].

Chapter 2

Privacy-preserving data sharing

2.1 Introduction

As the amount, complexity, and value of data available in both private and public sectors has risen sharply, data management and access control have challenged many organizations. Even more challenging are management and access control in *interorganizational* data sharing. Each organization would like to minimize the amount of sensitive information disclosed to other organizations, including both information about the data and information about the organization's work methodologies and role structure.

2.1.1 Problem description

We consider scenarios in which multiple organizations need to share data while each organization uses its own set of *proprietary metadata* to describe the shared data. In these scenarios, data records contain a *payload*, which is the actual data, and a set of *metadata attributes* that describe the payload. Although organizations may agree to share the payload, each uses a different set of metadata attributes, taken from its own professional domain, to describe this payload. Data must be shared in a controlled manner that protects the confidentiality of each organization's proprietary attributes *and* prevents unauthorized

users from accessing the payload.

Typically, one organization, the *data owner*, maintains a set of data records that are potentially useful to other organizations, called the *data clients*. Each data record contains sensitive information about an individual, the *data subject*. *Data users*, who are employees of a data client, may need access to data records stored by the data owner to perform their assigned tasks. Each user must have the proper authorization to access the payloads of the specific set of records needed for a given task. Our framework also features a third type of organization, *data intermediaries*, that enrich data with additional information that is needed for the client's tasks but is available only to the intermediary. Each organization ORG_i maintains its own vocabulary VOC_i that contains the overall set of domain-specific, intraorganizational attributes used in its operations. VOC_i includes both proprietary, sensitive attributes and attributes that can be shared with other organizations. ORG_i uses a different set of attributes, $ATT_{i,j} \subseteq VOC_i$, to describe each shared payload p_j.

For example, the data owner may be an email service provider (ESP). The data records represent email messages. Each email record is composed of a payload, which is the content of the email message, and metadata attributes about the payload such as sender, receiver, and date. Some attributes, *e.g.*, the email message's receiver, are sensitive; therefore, the ESP will share them with other organizations only when required to do so and only in a controlled manner. Each email message is created by one of the ESP's customers, who are the data subjects; it is then stored and cataloged using attributes that represent the message's metadata as collected by the ESP. Clients may be law-enforcement (LE) agencies, in which agents (data users) need access to email records in order to perform investigations. Intermediaries may include government agencies such as the IRS, which could provide tax records associated with the email addresses that appear in the messages' metadata attributes.

Design goals: Each organization wishes to maintain its proprietary view of the shared data

and to keep that view confidential. This means that the set $ATT_{i,j}$ of attributes that ORG_i maintains on each shared payload must be hidden from the other organizations.

Another requirement that must be accommodated is the use of multiple vocabularies. The owner uses vocabulary VOC_1 to store and query the shared data, an intermediary uses a different vocabulary VOC_2 to enrich the shared data, and the client uses a third vocabulary VOC_3 to query and process the data, to manage access control, and to issue data-access authorizations to its employees. Therefore, our framework must provide a mechanism that *dynamically and obliviously* transforms attributes of shared data from one vocabulary to another. Note that that this problem cannot be solved simply by requiring any set of organizations that may need to share data to agree on a shared, standard vocabulary. Such a standardization effort would require the organizations to know both the names and values of attributes used by other organizations. However, our premise is that the values of many attributes used internally by organizations are sensitive and cannot be exposed to other organizations. Furthermore, in many natural use cases (see Subsection 2.2.2), transformations require auxiliary information, such as up to date statistics or lists. Such information is known only at the point at which a user requests a specific data record and may need to be supplied by an intermediary that is not known by the data owner at the time that the owner encrypts the data.

Finally, because attributes could reveal sensitive aspects of organizations' activities, regulators and data subjects should expect sharing of both payloads and attributes to be kept to a minimum. To facilitate minimal exposure of sensitive information, an interorganizational data-sharing framework should offer a data-centric access-control mechanism. Such a mechanism will allow a user to access a payload only if it is essential for the completion of one of her tasks; in addition, it will allow the user to learn only the subset of that payload's attributes that are needed for the task.

2.1.2 Starting point: attribute-based encryption

Attribute-based encryption (ABE) is a natural starting point in the design of our framework. In our terminology, the encryptor is the data owner, users are data clients' employees (data users), and trusted authorities (TAs) both inside and outside the data client determine users' access policies. An ABE scheme grants an individual user a key that permits him to decrypt a ciphertext if and only if the key matches certain attributes specified during the ciphertext's creation. ABE enables fine-grained access control, which is essential in a privacy-preserving data-sharing framework. It provides one-to-many encryption, which can significantly increase the scalability of encryption and key management – properties that are necessary for interorganizational data sharing. ABE policy-access formulae are highly expressive, because they can be specified with binary or multivalued attributes, using AND, OR, and threshold gates.

Existing ABE schemes, however, have several properties that make them unsuitable for our framework.

In existing ABE schemes, encryptors, users, and TAs all use the same vocabulary. This means that these schemes cannot be used off-the-shelf in our framework, where a crucial element of the problem description is that participating organizations may belong to different business sectors or professional domains and thus use different vocabularies. In particular, a data client's TAs and employees use a different vocabulary from that of the data owner. In ABE terms, this implies that attributes used in access policies (and keys) issued by the TAs to data users might belong to a different vocabulary from the one used by the owner to encrypt and store ciphertexts. Unless a suitable transformation is made between the keys and the ciphertexts, decryption will fail even if the ciphertext satisfies the user's access policy. Such a transformation must separately consider each attribute in the ciphertext and change it into a valid attribute from the users' keys' vocabulary. To protect both data subjects' privacy and organizations' proprietary views, the original attribute must

remain hidden from the user and the new attribute must remain hidden from the encryptor. Existing ABE schemes cannot support this requirement.

Moreover, existing ABE schemes are generally used for role-based access control and thus have user-centric vocabularies (attributes that describe decryptors' traits) that reflect organizational structure and roles. The use of user-centric attributes, coupled with the single-vocabulary assumption, implies that the encryptor (data owner) must be exposed to the roles of potential decryptors (clients' data users) and the organizational structure that they fit into. Many organizations are reluctant to share such sensitive information.

2.1.3 Main contributions

We present a new system architecture and a suite of protocols for interorganizational data sharing that support privacy of both data (*payload hiding*) and organizational vocabularies (*attribute hiding*). We introduce *Attribute-Based Encryption With Oblivious Attribute Translation (OTABE)*, in which a semi-trusted proxy *translates* the attributes under which a data record's payload was encrypted into the attributes under which it can be decrypted by authorized users. The proxy performs the translation without learning the underlying plaintext data. Moreover, translation is *oblivious* in the sense that the attributes under which the record is encrypted remain hidden from the proxy. This novel cryptographic technique enables mutually untrusted parties not only to *use different vocabularies* of attributes to describe the shared data but also to share proprietary metadata attributes in a controlled manner that protects the attributes' confidentiality (*attribute privacy*). Furthermore, attributes and policies can be *dynamically reconfigured*, in the sense that updates are done dynamically, without the need for re-encryption, and offline. No previous proposed ABE scheme achieves all of these properties.

We provide a concrete OTABE scheme and prove it selectively secure in the standard model. We then use it in our design of an efficient, expressive, and flexible interorgani-

zational data-sharing framework that we call PRShare. In addition to the direct benefits of OTABE described above, PRShare provides several other capabilities that are desirable in real-world interorganizational data-sharing applications, including *efficient and direct revocation, protection from key-abuse attacks*, and *hidden access policies*. In order to obtain these features, we leverage our OTABE scheme's translation technique. OTABE also enables *division of trust* (multiple independent authorities authorize data access) and *data centricity* (access policies contain data-related, rather than user-related, attributes), both of which enhance privacy protection in PRShare. Finally, because of the unique structure of OTABE ciphertexts, a single owner's database can serve multiple clients without knowing the clients' identities at encryption time. Furthermore, the owner does not to need to authorize or serve clients' data-access queries. Previous ABE schemes achieved some of these desirable properties of our OTABE construction, but none achieved all of them.

Before proceeding to our technical results, we note that our approach is not suitable for *all* data-sharing applications. For example, it is not intended for scenarios in which the data subject participates directly in the user's request for data about her and could be asked to grant explicit consent. In general, data subjects in the scenarios we consider will not even be aware of the specific uses that are made of data about them. Similarly, our approach is not intended for scenarios in which there are clear, efficiently decidable, and universal rules that govern which users can access which portions of the data; existing access-control mechanisms suffice in such scenarios. Our techniques are useful in scenarios in which there are legally mandated, general principles that govern access to sensitive data, but instantiating those principles in the form of efficiently decidable rules requires data-specific and dynamically changing knowledge. We give two examples in Subsection 2.2.2.

Table 2.1: Properties of Proxy-Assisted ABE Schemes

Scheme	Type	MA	Access policy	DAT	Proxy's role	DR	SM	HP
[50]	KP-ABE	✗	LSSS	✗	Outsourced decryption	✗	RCCA	✗
[73]	CP-ABE	✗	AND gates	✗	Delegation of decryption rights	✗	SCPA	✗
[116]	KP-ABE	✗	LSSS	✗	Revocation management	✗	SCPA	✗
[104]	KP-ABE	✗	LSSS	✗	Revocation management	✓	SCPA	✗
[117]	CP-ABE	✗	AND gates	✗	Revocation management	✗	SCCA	✗
[15]	CP-ABE	✓	LSSS	✗	Outsourced decryption	✗	SRCPA	✓
[72]	CP-ABE	✗	LSSS	✗	Delegation of decryption rights	✗	SCCA	✗
[70]	CP-ABE	✗	AND gates	✗	Outsourced decryption, encryption	✗	SCPA	✗
[67]	CP-ABE	✗	LSSS	✗	Outsourced decryption	✗	SCPA	✗
OTABE	KP-ABE	✓	LSSS	✓	Attribute translation	✓	SCPA	✓

2.2 Background and motivation

2.2.1 Related work

Existing privacy-preserving data-sharing schemes fall into two general categories: centralized and decentralized. The former category includes the works of Dong *et al.* [32], X. Liu *et al.* [76], Popa *et al.* [91] and Vinayagamurthy *et al.* [109]. The major advantage of the centralized approach is efficiency; disadvantages include single points of failure and the lack of division of trust. Decentralized solutions can be found in the work of Fabian *et al.* [35], Froelicher *et al.* [41], C. Liu *et al.* [75], and Nayak *et al.* [86]. Decentralized solutions avoid single points of failure, but they often have limited efficiency or scalability.

The original motivation for PRShare was enhancement of privacy protections in surveillance processes. Previous work in this area includes that of Kamara [62] and Kroll *et al.* [66]; they proposed cryptographic protocols that protect the privacy of known surveillance targets. Segal *et al.* [101, 102] focused on unknown (*i.e.*, not yet identified) targets and provided cryptographic protocols that protect privacy of innocent bystanders in two

commonly used surveillance operations: set intersection and contact chaining. Frankle *et al.* [40] used secure, multiparty computation and zero-knowledge protocols to improve the accountability of electronic surveillance.

Attribute-based encryption was introduced by Sahai and Waters [99]. Their work was followed by many ciphertext-policy ABE and key-policy ABE constructions, including those in [11, 17, 47, 88, 96]. Chase [25] introduced multi-authority ABE, and Nishide *et al.* [87] introduced ABE with hidden access policy. ABE has been applied in a wide range of domains, including fine-grained data-access control in cloud environments [112], health IT [5, 71], and security of blockchains and Internet-Of-Things devices [93, 115].

We now give a high-level explanation of some crucial differences between the role of proxies in OTABE and their roles in previous works.

An OTABE scheme provides an algorithm $Translate()$ which allows a semi-trusted proxy to translate one or more of the attributes under which a data record's payload is encrypted without learning the underlying plaintext. Moreover, translation can be done obliviously, in the sense that the attributes under which the payload is encrypted remain hidden from the proxy who translates them. The proxy learns only the attributes' new values.

Two common responsibilities of proxies in ABE are *outsourced decryption*, which was introduced by Green *et al.* [50], and *revocation management*, which was used by Yu *et al.* [116, 117]. In both cases, proxies are used for efficiency; they assume much of the computational cost of decryption or revocation and lighten other parties' loads. The attribute-translation protocols in OTABE are *not* designed to reduce the client's or the owner's computational loads. Similarly, outsourced-decryption and revocation-management proxies are not designed to enable oblivious translation between organizational vocabularies or to support dynamically reconfigurable attributes – two of OTABE's primary goals. Simply put, proxies used for outsourced decryption and revocation management and those in

OTABE serve completely different primary purposes.[1]

The use of proxies for *ciphertext delegation* was introduced by Sahai *et al.* [98]. Proxies in this scenario take ciphertexts that are decryptable under policy P_1 and transform them into ciphertexts that are decryptable under policy P_2. However, P_2 must be stricter than and use the same vocabulary as P_1; here, "stricter" means than P_2 permits the decryption of a subset of the ciphertexts that could be decrypted under the original policy P_1 used by the encryptor. Neither of these restrictions applies to the proxies in OTABE.

In *attribute-based proxy re-encryption* (ABPRE), which was introduced by Liang *et al.* [73], a proxy re-encrypts a ciphertext encrypted under access structure AS_1 to one that can be decrypted under access structure AS_2 without learning the plaintext. There is a surface similarity between ABPRE and OTABE in that proxies in both transform ciphertexts encrypted by data owners under AS_1 into ciphertexts decryptable by clients under AS_2. However, the entity that issues re-encryption keys to proxies in ABPRE requires knowledge of the vocabularies of both owner and client; to create re-encryption keys, she must know AS_1 *and* AS_2. Thus, unlike OTABE, ABPRE does not support multiple vocabularies and can not provide attribute privacy.

In an ABPRE scheme, re-encryption keys are issued to a proxy on a per-access-policy basis. In order to perform re-encryption, the entire access policy must be changed so that the new policy contains no attributes that appear in the original policy. Neither of these restrictions applies to OTABE, in which re-encryption-key issuing and re-encryption itself can be done on a per-attribute basis. The responsibility for determining the new attribute set and performing the re-encryption is divided among multiple parties from different trust domains. Each party performs a partial re-encryption that uses only the attributes that belong to its trust domain and does so in a controlled manner that results in a final, full re-encryption that satisfies the data owner's requirements. This decentralized approach

[1] A direct-revocation mechanism, partially managed by the proxy, is a natural byproduct of attribute translation, as described in Subsection 2.4.2, but it is not the primary goal of OTABE.

allows OTABE to support multiple vocabularies, provide attribute privacy, and enable dynamically reconfigurable translation policies that do not require re-initialization of the system or re-encryption of records by the owner.

Finally, in ABPRE, the proxy must know the ciphertext's original access policy in order to perform the re-encryption. OTABE proxies, by contrast, perform *oblivious* translation and re-encryption; they do not learn the original set of attributes or the original access structure under which the plaintext was encrypted.

A detailed comparison between our scheme and other proxy-assisted ABE schemes is shown in Table 4.2. MA denotes multi authority, DAT denotes dynamic attribute translation, DR denotes direct revocation, SM denotes security model, HP denotes hidden policy. In addition, SCCA,SCPA and SRCPA stand for selective CCA, CPA and RCPA, respectively.

2.2.2 Use cases

In order to motivate the notion of OTABE and illustrate its applicability in real-world scenarios, we provide two examples.

Law-enforcement agencies: The Electronic Communications Privacy Act (ECPA) [33] was passed to protect the privacy rights of ISPs' customers with respect to disclosure of their personal information. The ECPA limits LE access to email and other communication records in a manner that is consistent with the Fourth Amendment. However, it has several "loopholes." For example, the ECPA classifies an email message that is stored on a third party's server for more than 180 days as "abandoned." As a result, LE agencies can request that both the metadata and the content of those email messages be turned over without the need for judicial review.

Unrestrained government access to communication data is clearly undesirable. However, given national-security and public-health concerns, expecting LE and intelligence

agencies never to access *any* data held by communication companies such as ESPs is unrealistic. A more realistic goal is to deploy a policy that restricts such data sharing to the minimum needed in order to perform the task at hand, as defined by multiple trusted entities. OTABE provides a mechanism that can enforce such policies and protect the confidential information of all organizations and agencies that participate in the data-sharing protocols.

In OTABE terms, the data owner is the ESP, and the data subjects are people who send and receive email messages. The data are email records. Each email record contains a payload, which is the content of an email message, encrypted under a set of metadata attributes, *e.g.*, sender's and receiver's email addresses, date, subject line, *etc*. The client is an LE agency, such as the FBI or a municipal police department, and the intermediaries may be other LE agencies, non-LE government agencies, or private companies. The data users are LE agents employed by the client.

Clearly, email records can be useful to LE agencies, but an agent should be able to decrypt only those records whose metadata attributes constitute *probable cause* in the context of a specific investigation. The entities who determine probable cause on a per-investigation basis are the TAs. Each TA is motivated by a different set of interests and goals. A TA may be part of the judicial branch, the ESP, the LE agency, or another external entity.

Not all of the attributes used by the ESP to store email records can be shared with the LE agency, because some of them reveal both private information about the ESP's customers or proprietary information of the ESP itself. Similarly, the attributes used by the LE agency to access and process records and to issue access policies cannot be shared with the ESP, because they reveal confidential information about the LE agency's investigations. Furthermore, some of the attributes that are used by the parties do not belong to the same vocabulary. For instance, the attribute "appears-in-watchlist" is frequently used in keys issued to LE agents, but it is meaningless to the ESP. Such attributes must undergo dynamic

adaptation to ensure that agents' keys match an email message's attributes. OTABE allows the ESP and LE agency to use their own vocabularies while keeping the email messages' content and metadata confidential.

TAs are likely to grant an agent who is investigating a crime access to email records in which either the sender or the receiver is on the agency's watchlist. The LE agency's proxy can translate the ESP's sender and receiver attributes into the LE agency's "on-watchlist" attribute in an oblivious fashion, thus maintaining both the confidentiality of the watchlist and the privacy of data subjects' email addresses. In addition, an agent might want to check whether the sender or receiver appears on other agencies' lists, *e.g.*, a list of investigations ongoing at LEA-2, which is another LE agency. Because details of LEA-2's ongoing investigations cannot be shared with the client, the translation of the attributes sender and receiver will be done obligiously by LEA-2's intermediary proxy.

Similarly, the access policy of an agent who is investigating cyber fraud may enable access to email records in which the subject line matches a "suspicious" pattern. The definition of "suspicious" may be determined according to a dynamically updated list of keywords. Using this keyword list, the client's proxy can obliviously translate the attribute "subject line," maintained by the ESP, into the attribute "is-suspicious-subject," maintained by the client and used in the agent's access policy. Neither the agent nor the proxy is able to read the actual subject line, and the data subject's privacy is maintained.

Note that, in both of these investigations, dynamic translations are needed, because watchlists and lists of suspicious keywords change over time. They enforce the requirement that an agent cannot access payloads without probable cause, but they do not reveal to the ESP confidential information about watchlists and ongoing investigations.

Insurance companies: Consumer reporting agencies (CRAs) collect and share credit-related information about consumers. This information is used by credit-card issuers, mortgage lenders, insurance companies, *etc.* to assess creditworthiness of consumers. The three

Table 2.2: Summary of notations and symbols

Notation	Description
$(M)_S$	encryption of M under a set of attributes S
$[x]_y$	encryption of x under the key y
ORG_S	set of proxies involved in translation of $C = (M)_S$
DEC_S	set of parties involved in decryption of $C = (M)_S$
P_{org_j}	the proxy operating on behalf of organization org_j
S_m	mutable attributes
S_{im}	immutable attributes
S_p	the set of attributes' labels that P_{org_p} is allowed to translate
$pub_{\Pi}(x)$	the public key of entity x, created by a public-key scheme Π
K_x	a symmetric shared key between org_{owner} and organization org_x
$org(k)$	the organization who is allowed to translate attribute att_k
$E_j(L)$	encryption of auxiliary information L by organization org_j
$F(K, x)$	pseudorandom function keyed with symmetric key K
$F(x)[0]$	the first argument of the output of the evaluation of F on x

largest CRAs in the US are Experian, TransUnion, and Equifax.[2] The Fair Credit Reporting Act (FCRA) [36] regulates the collection, dissemination, and use of credit-related information. The FCRA gives companies the right to access consumers' credit reports. This access is not limited to reports on the company's customers; it may include reports on large sets of potential customers. In order to create pre-screened offers and market them to potential customers, an insurance company is allowed to access consumers' credit reports and to share information with credit-card issuers, banks, other insurance companies, *etc*. However, access rights to credit reports are limited by the FCRA to information for which an insurance company has a *permissible purpose*. OTABE can be used to formalize and enforce this vague concept in a manner that protects both consumers' privacy and proprietary information of insurance companies and CRAs.

In OTABE terms, the data owner is a CRA, and the data subjects are consumers. Data records are credit reports, owned by the CRA. Each record is encrypted under the set of attributes that describe the report, *e.g.*, the phone number, credit score, and driver's license

[2] In September of 2017, Equifax announced a data breach that exposed the personal information of 147 million people and cost the company hundreds of millions of dollars in compensation to affected people [16, 37].

number (DLN) of the data subject, credit-utilization ratio, date and time of the report's creation, CRA-internal statistics, *etc*.

Insurance companies are the data clients. Data users are insurance-company employees who use credit reports to make decisions about which insurance products to offer consumers and how to price them. In order to comply with the FCRA's "permissible-purpose" requirement, employees should only access credit reports on a "need-to-know" basis. An employee can only access those records whose associated attributes are relevant to her task, as determined by a set of TAs. TAs may include the CRA, a government entity, or various parties within the insurance company. Other organizations, such as credit-card issuers, government entities, banks, and other insurance companies may serve as intermediaries by "enriching" data supplied by a CRA in a privacy-preserving manner.

As in the LE scenario, each organization wants to protect the confidentiality of its proprietary information. For instance, the CRA does not want to reveal unnecessary identifying information about its customers, an insurance company does not want to reveal how it makes business decisions regarding which consumers are considered "qualified" for pre-screened offers, *etc*. Also as in LE, different organizations may use different vocabularies. Consider the attribute "number of accidents," which is used by insurance companies to screen potential customers. This attribute cannot be used by CRAs, because they do not maintain such information in their credit reports. OTABE supports all of these requirements.

Assume that each report is encrypted under the following attributes:
CREDIT-UTILIZATION-RATIO, CREDIT-SCORE, PHONE-NUMBER, DLN, and DATE.
Employee U in the car-insurance department is assigned the task of finding qualified potential customers and tailoring pre-screened car-insurance offers, using information found in their credit reports.

The TAs determine that, for this task, a qualified customer is defined by the following policy:

CREDIT-SCORE $\geq X \wedge$ #ACCIDENTS $\leq Y \wedge$ IS-BLACKLISTED=FALSE $\wedge$ IS-CREDIT-RATIO-LESS-THAN-AVERAGE=TRUE

The intermediaries in this case are financial business partners of the insurance company, *e.g.*, banks and credit-card issuers, and the Department of Motor Vehicles (DMV).

To complete her task, U submits to the CRA a query that requests the reports of all consumers whose credit scores are greater than X. The CRA then sends each matching record to two intermediaries: the DMV and a credit-card issuer.

For each record, the DMV's proxy obliviously translates the DLN attribute into #ACCIDENTS, which is found in the subject's driving record. The credit-card issuer's proxy obliviously translates the numeric CREDIT-UTILIZATION-RATIO attribute into a binary attribute IS-CREDIT-RATIO-LESS-THAN-AVERAGE by obliviously comparing the consumer's utilization ratio with the average utilization ratio of the issuer's customers. The insurance company's proxy obliviously translates the PHONE-NUMBER attribute into the attribute IS-BLACKLISTED, using a dynamically updated list of individuals who were blacklisted by the insurance company or one of its business associates for, *e.g.*, failure to pay.

When U receives a record, she will be able to decrypt the credit report, read its contents, and learn the subjects' identifying information if and only if the record's post-translation attributes satisfy her access policy.

Data privacy is achieved, because only authorized users can decrypt a given credit report. Attribute privacy is achieved, because attributes used by each organization remain hidden to the extent required. Moreover, sensitive information about consumers whose records *are* decrypted is also protected. For example, a user may learn that a consumer's number of accidents is below a certain threshold but not learn the exact number. Finally, these translations demonstrate OTABE proxies' ability to translate *dynamically*, because the list and the average change over time, and *obliviously*, because neither the attributes

nor the data are revealed to them.

2.3 Attribute-based encryption with oblivious attribute translation

2.3.1 Terminology

Attributes: Our scheme uses multi-valued attributes, denoted by $\langle label, operator, value \rangle$. Note that this representation is different from the ones found in typical ABE schemes, which use "descriptive" (essentially binary) attributes. We denote by att_k^L and att_k^V the label and value of an attribute att_k. Translation of an attribute can be done either by changing the attribute's value (*i.e.*, replacing $value$ with $value^*$) or by replacing both the attribute's label and its value with $label^*$ and $value^*$, respectively.

In PRShare, attributes' labels are partitioned into two sets: mutable, denoted S_m, and immutable, denoted S_{im}. Immutable attributes are ones that cannot be translated by any party in the system. Intuitively, they are the attributes that are shared by the owner and the client. Mutable attributes, on the other hand, are ones that can be translated by a semi-trusted proxy at some point after their initialization by the owner.

Hidden access policy: We introduce an OTABE scheme with hidden access policy by ensuring that the set of attributes used to encrypt a message is hidden from the CSP, the proxies, and the data users. We use the term "hidden access policy" for compatibility with the terminology used in existing CP-ABE work, in which access policies are attached to the ciphertexts.

In such a scenario, a data user cannot learn the attributes that are attached to a ciphertext but is able to determine which attributes are needed to perform the decryption. The hidden-access-policy feature is used to enhance privacy. However, if the owner and client

wish to reveal the ciphertexts' attributes to the users or wish to speed up decryption at the expense of some privacy, they can turn off this feature without having to alter the encryption, translation, and decryption operations. This follows from the modular design of the system, as discussed in Subsection. 2.5.1. Note that the hidden-access-policy feature does not enable the creation of trivial policies (*i.e.*, those that always allow a user to decrypt every record she receives). This is because a key must satisfy *all* TAs' policies in order to succeed in decrypting, and the data owner can always serve as a TA or delegate to a TA that it trusts not to permit decryptions that it wishes to forbid.

In general, PRShare is designed to achieve a high level of privacy while allowing flexible and expressive data-sharing protocols. In real-world scenarios, however, organizations have different priorities. Some may favor privacy, but others may favor functionality and thus prefer to allow their data users broader access to information about the shared data at the expense of privacy. PRShare is able to support both approaches: It is highly modular, and each privacy guarantee relies on a different low-level feature that can be removed or changed to fit the organization's privacy-functionality trade-offs while maintaining the rest of the privacy guarantees.

(Informal) Definition: Let M be a data record's payload encrypted under a set $S \subseteq \mathcal{U}_1$ of attributes, resulting in a ciphertext C. We refer to the set S as the set of original attributes under which M is encrypted. Let $T : \mathcal{U}_1 \rightarrow \mathcal{U}_2$ be a translation function from the universe $\mathcal{U}_1$ of attributes to the universe $\mathcal{U}_2$ of attributes, and let Q_j be the set of original attributes that a semi-trusted proxy j is allowed to translate. An ABE scheme supports ***oblivious attribute translation by semi-trusted proxy*** j if, given C, Q_j, and T, for all $s \in Q_j$, the proxy is able to compute $T(s)$ without:

- learning anything about M,
- learning anything about the attributes in $S \setminus Q_j$, or
- learning the labels or the values of attributes in $S \cap Q_j$.

Formal security definitions are given in Subsection 2.5.2.

2.3.2 Algorithms

An MA-OTABE scheme consists of the following algorithms:

GlobalSetup$(\lambda) \Rightarrow (PK)$: The global-setup algorithm takes as input a security parameter λ and outputs global parameters PK.

AuthoritySetup$(PK) \Rightarrow (PK_i, MSK_i)$: Each authority runs the authority-setup algorithm with PK as input to produce its own public key PK_i and master secret key MSK_i.

Encrypt$(M, PK, S, \{PK_i\}_{i \in Aut}) \Rightarrow (CT)$: The encryption algorithm takes as input a message M, a set S of attributes, and the public parameters. It outputs the ciphertext CT.

KeyGen$(PK, MSK_i, A_i, u, t) \Rightarrow (SK_{i,u,t})$: The key-generation algorithm takes as input the global parameters, an access structure A_i, a master secret key MSK_i, the global identifier u of a data user who issued the key-generation request, and a task t. It outputs a decryption key $SK_{i,u,t}$.

Distribute$(I) \Rightarrow (\{C^j | j \in DEC_S\})$: This algorithm takes as input a set I of ciphertexts' ids. It outputs a set of partial ciphertexts, $\{C^j | j \in DEC_S\}$.

Translate$(PK, j = p, C^p, \{PK_i\}_{i \in Aut}) \Rightarrow (C'^p)$: The translation algorithm takes as input the global public parameters and the authorities' public parameters, a proxy's index $j = p$, and a partial ciphertext C^p. It outputs a translated partial ciphertext C'^p.

Decrypt$(PK, \{SK_{i,u,t}\}, C^u, \{C'^j | j \in ORG_S\}) \Rightarrow (M)$: The decryption algorithm takes as input the global parameters, a set of secret keys $\{SK_{i,u,t}\}_{i \in Aut}$, a partial ciphertext C^u, and a set of translated partial ciphertexts $\{C'^j | j \in ORG_S\}$. It outputs the plaintext M.

2.4 System model

Definition of attributes: We define two sets of attributes' labels: S_{owner} represents the set of attributes that the owner uses to encrypt, store, and access data records that it owns. This set is determined by the data owner. S_{client} represents the set of attributes under which keys are generated; those are the attributes that the client uses to access and process the shared data records, and they are chosen by org_{client}. Note that $S_{owner} \cap S_{client} \neq \emptyset$; this means that some attributes are shared by the client and the owner. This enables the users to retrieve data records of potential interest from the CSP using queries that are composed of shared attributes and also enables the data owner, if it wishes, to be one of the TAs. We denote the universes of attributes comprising each set by $\mathcal{U}_{owner}$ and $\mathcal{U}_{client}$.

For each data intermediary org_j in the system, we define a set of attributes' labels $S_j \subseteq S_m$. It represents the set of attributes that is governed by org_j and hence can be translated by the proxy P_{org_j} that acts on behalf of org_j.

2.4.1 System participants

Data owner: org_{owner} is responsible for encrypting each of its data records using the set $S \subseteq \mathcal{U}_{owner}$ of attributes that are most likely to appear in future queries.

Data users: Data users are employees of org_{client} who need access to data records stored by org_{owner} in order to perform daily tasks. Each user is assigned a unique global identifier and a list of tasks. Each task t has a well defined time limit tl_t. The list is dynamic in the sense that tasks can be removed or added to it during the system run. A user issues two types of queries. A *key request* is used to obtain a key that corresponds to a specific access policy. A *data query* is used to obtain data records owned by org_{owner} that are relevant to a specific task in the user's task list.

Cloud-service provider: The CSP stores the ciphertexts outsourced by org_{owner} and re-

sponds to queries submitted by data users in org_{client}.

Trusted authorities: TAs are the entities that determine the decryption policy of org_{client} and issue secret keys that are used by data users. They use attributes from $\mathcal{U}_{client}$. There must be at least two TAs, and they may be entities in org_{owner}, org_{client}, or external organization. We assume that at least one TA belongs to org_{client} and that at least one TA does not.

Proxies: Each proxy P_{org_j} represents a different organization org_j (either an intermediary or a client) and operates on behalf of that organization. The role of a proxy P_{org_j} is to translate a subset of attributes in $\mathcal{U}_{owner}$ under which a ciphertext was encrypted to the corresponding attributes in $\mathcal{U}_{client}$. To do this, the proxy uses both a generic translation algorithm that is used by all proxies in the system and an organization-specific translation function that is determined by org_j and may involve auxiliary information provided by the organization to its proxy. The generic translation algorithm is public, but the organization-specific translation function and auxiliary information are considered private to org_j and P_{org_j}. We assume that every MA-OTABE scheme includes at least one proxy (the "client proxy") that is responsible for managing org_{client}'s user-level revocation mechanism and for performing vocabulary translations.

Data subjects: Each data record owned by org_{owner} is linked to a certain individual, the data subject. A data record's payload contains personal information about the data subject, including content produced by the data subject. We assume that every data subject has a user id (UID) that varies based on the type of data used in the system. Examples of UIDs include phone numbers and email addresses.

2.4.2 Revocation mechanism

One major byproduct of OTABE is the ability to implement an efficient and direct revocation mechanism, in which revoking the keys of a set U of users does not affect the

keys of users not in U. Using the translation technique, a semi-trusted mediator can transform a ciphertext that was encrypted under a set of data-centric attributes at point A into a "personalized" ciphertext reflecting a specific data query made by a user at point B. The main idea of our revocation mechanism is the addition of global-identifier (GID) and time attributes to each key. In addition, we add a dummy GID and dummy times during encryption. These dummy attributes will be translated to suit the specific data query's time and requester only if a certain criterion is met. This creates an efficient mechanism in which most revocations are enforced automatically.

We assume that every data user receives a unique GID. The data client maintains a revocation list that contains revoked GIDs. Users whose GIDs are on the revocation list are not allowed to access any data record. Revocation-list updates are infrequent and happen only when a user completely leaves the organization. Furthermore, GIDs can be removed from the revocation list after a relatively short time, because the key-level revocation mechanism ensures that secret keys become invalid within a well known and controlled length of time from the date they were issued.

For the key-level revocation mechanism, we leverage a basic trait of an organizational task: It has a well defined time limit. This time limit is determined by the user's manager and may change while the user is working the task. In our case, the entities who choose the time limit are the TAs; this is an integral part of the per-task "probable-cause" approach. The time limit given to a specific task performed by a user becomes an attribute in the user's key. In addition, the encryptor adds to each ciphertext a dummy "time" attribute. That dummy attribute is translated by the client proxy to the current time at which the data query is submitted by the user, thus making a key-level revocation check an automatic part of any decryption attempt. In our construction, we view a "time limit" as a date. This can easily be extended to include finer-grained notions of time.

We also leverage our attribute-translation technique for the user-level revocation mechanism. It enables us to include a user-specific component in the ciphertext; this component

is adjusted according to the specific data user by the client proxy in the data-retrieval phase. Note that we treat the GID as an *additional attribute*. We incorporate the user's GID as an attribute in the user's secret keys and, in parallel, add a "placeholder" GID attribute to each ciphertext. When a user submits a data query, the placeholder attribute is translated to that specific user's GID only if she does not appear in the revocation list. This mechanism provides an efficient user-level revocation mechanism and protects the scheme from collusion attempts and key-abuse attacks.

Details of the translations used in our revocation mechanism are provided in Subsection 2.7.2.

2.4.3 Main flows

The system model consists of an encryption flow, a data flow, and a key-generation flow. We assume that the system has already been set up, resulting in the global public parameters PK and a public-key, master-secret-key pair (PK_i, MSK_i) for each trusted authority Aut_i.

Encryption flow: In order to encrypt a data record's payload M, org_{owner} first determines the set S of attributes under which M will be encrypted. $S \subseteq \mathcal{U}_{owner}$ is composed of $|S| - 2$ data-centric attributes that describe the record's metadata and two attributes that serve as "placeholders." The placeholders att_{GID} and att_{TIME} are initialized with random, "dummy" values by org_{owner} and receive their actual values from org_{client}'s proxy. Based on the attributes in S, the encryptor determines the set DEC_S of decryption parties. DEC_S contains all parties involved in the decryption of the ciphertext, *i.e.*, a data user and the set ORG_S of organizations that are allowed to translate attributes in S (represented by their proxies). ORG_S includes the client's proxy and any number of data intermediaries' proxies. After determining DEC_S, org_{owner} encrypts M under S by calling $Encrypt(M, PK, S, \{PK_i\}_{i \in Aut})$ and receives a set $\{C^j\}$ of $|DEC_S|$ partial ciphertexts.

$|DEC_S| - 1$ of the partial ciphertexts correspond to proxies and contain only attribute components. One corresponds to the data user and contains both attribute components and a data component; the latter contains the payload M itself. Note that, for each C^j, $U(C^j) \subseteq \mathcal{U}_{owner}$, where $U(C)$ is the vocabulary of attributes under which a ciphertext C is encrypted. Lastly, org_{owner} computes $Y = \{Obf(att_k) \mid att_k \in S\}$, a set of obfuscated values for immutable attributes in S, and uploads to the cloud the preprocessed ciphertext and the UID that the ciphertext is associated with.

Key-generation flow: A user u who belongs to org_{client} sends a key request to the TAs in each of the following cases: Either a new task is inserted to u's task list, or the time limit for an existing task in u's task list has expired, and her existing secret key for that task is no longer valid. The request contains a description of the task and the "ideal" access policy that u would like to obtain in the context of that task. Each authority Aut_i creates an access policy A_i based on an examination of the user's request and the nature of the specific task. It creates a GID attribute att_{GID} that contains the user's GID u. Finally, it determines tl_t, which is either a new time limit for t (if t is a new task) or an extended time limit (if t is an existing task and its time limit has expired) and uses tl_t to create a time-limit attribute att_{LIMIT}. The time-limit attribute that is embedded in a secret key must be expressed using the same units (date, month, time stamp, *etc.*) used in the time attribute att_{TIME} that is attached to the ciphertext. It then creates its secret key $SK_{i,u,t}$ by calling $KeyGen(PK, MSK_i, A'_i, u, t)$, where

$$A'_i = A_i \wedge att_{GID} \wedge att_{LIMIT} = A_i \wedge (GID == u) \wedge (TIME < tl_t).$$

Data flow: A data user u sends a data query to the CSP. It contains a conjunctive query ψ on attributes from $\mathcal{U}_{owner} \cap \mathcal{U}_{client}$. The CSP retrieves the ciphertexts that satisfy the query. For each ciphertext C, it sends $C^{j=u}$ to u and each $C^{j=p}$ to a proxy P_{org_p}. At that point, because u received only a partial ciphertext, she cannot yet use her key for

decryption. Each proxy P_{org_p} in ORG_S translates each attribute att_k such that $(att_k \in S) \wedge (att_k^L \in S_p)$ by calling $Translate(PK, j = p, C^p, \{PK_i\}_{i \in Aut})$ and computes an obfuscated value for each new attribute $att_{k'}$ that it added, creating $Y_p = \{Obf(att_{k'})\}$. The client organization's proxy also manages the user-level mechanism by performing a correct translation of att_{GID} and att_{TIME} only if u does not appear in the revocation list. Each proxy P_{org_p} then sends the translated partial ciphertext $C'^{j=p}$ and Y_p to the user. At this point, $U(C'^j)$ has changed from $\mathcal{U}_{owner}$ to $\mathcal{U}_{client}$. Because each partial ciphertext is, from the proxy's view, independent of the data component inside the ciphertext, each proxy is able to perform the translations without learning M. Moreover, the structure of each partial ciphertext ensures that P_{org_j} learns nothing about the attributes with labels that do not belong to S_j. All attribute components that correspond to attributes that the proxy can translate contain obfuscations of the attributes, rather than the attributes themselves; thus, each attribute att_k such that $(att_k \in S) \wedge (att_k^L \in S_p)$ remains hidden from the proxy, while the obfuscated value can still be used for various translation operations. The user gathers all the translated partial ciphertexts $\{C'^j | j \in ORG_S\}$ and her partial ciphertext C^u to create an aggregated ciphertext that she can decrypt using her secret key. Finally, u decrypts the payload by calling $Decrypt(PK, \{SK_{i,u,t}\}_{i \in Aut}, C^u, \{C'^j | j \in ORG_S\})$. The decryption succeeds if and only if the following three conditions hold:

- $\forall i \in Aut, TR(S) \models A_i$, where $TR(S) = Y \cup \{Y_j\}_{j \in ORG_S}$ represents the set of translated attributes, created based on the original set S of attributes.
- tl_t, the time limit for task t, has not expired. (Otherwise, att_{LIMIT} cannot be satisfied.)
- u has not been revoked, and no collusion or key-abuse attempt has been made. (Otherwise, att_{GID} cannot be satisfied.)

2.5 Security definitions

2.5.1 Goals and trust relationships

An OTABE-based framework should satisfy three security goals with respect to all PPT adversaries.

Selective security against chosen-plaintext attacks: The adversary cannot learn (in the selective-security model) the plaintext of either an original ciphertext or an aggregated, translated ciphertext.

Security against colluding parties: Let $C = (M)_S$ be a valid MA-OTABE ciphertext. No coalition of at most $|DEC_S| - 1$ parties can learn anything about M.

Attribute secrecy: The trust model that we consider in this work is different from the standard ABE trust model. Unlike the plaintext, for which we have a single security notion that applies to all the participants, we cannot apply a uniform security criterion to the attributes. Because each party plays a distinct role in the protocol, the set of attributes to which it is allowed to be exposed differs from the sets to which other parties are allowed to be exposed. We define three security requirements to ensure the secrecy of ciphertexts' attributes: hidden access policy, oblivious translation, and attribute privacy.

Hidden access policy: The set of attributes used to encrypt a message cannot be learned by the CSP, the proxies, or the data users.

Oblivious translation: The original attributes that each proxy P_{org_j} translates remain hidden from the proxy. That is, for every attribute s such that $s^L \in S_j$, the proxy P_{org_j} is able to translate s into a new attribute $s' \in \mathcal{U}_{client}$ without learning s.

Attribute privacy: Informally, the attribute-privacy requirement states that organizations that share data must be able to maintain separate views of the data that they share.

Definition 1 Given a payload space $\mathcal{M}$, a universe $\mathcal{U}_{owner}$ of attributes used by the encryptor (org_{owner}) to describe data records it owns, and a universe $\mathcal{U}_{client}$ of attributes

used by org_{client} for data usage and authorization management, we define a function $MAP : \mathcal{M} \times \mathcal{U}_{owner} \rightarrow \mathcal{U}_{client}$ that maps attributes in org_{owner}'s vocabulary (corresponding to data records' payloads $M \in \mathcal{M}$) to attributes in org_{client}'s vocabulary. An OTABE scheme achieves ***attribute privacy*** if and only if:

- For every data record's payload M and every attribute $s \in \mathcal{U}_{owner}$, if s is mutable, the encryptor does not learn $MAP(M, s)$, the translated value of the attribute s with respect to M.
- For every data record's payload M and every attribute $v \in \mathcal{U}_{client}$, if $MAP^{-1}(M, v)$ is mutable, data users and TAs do not learn $MAP^{-1}(M, v)$, the original value of the attribute v with respect to M.

The following observations about our threat model, which considers external adversaries as well as the parties presented in Subsection 2.4.1, are natural aspects of the security definitions and results presented in Subsection 2.5.2.

No organization fully trusts the other organizations: Our framework protects the owner's data records, attributes of the data held by each organization, and auxiliary information held by each organization that is used for attribute translation. We assume that the owner is honest but curious.

No organization fully trusts its proxy server: CSPs and proxies in our framework, which we assume to be honest but curious, are only given encrypted attributes and encrypted auxiliary information. Note that the use of honest but curious proxies is well established in the design of cryptographic protocols [9, 18, 21, 49, 56, 117].

The client organization does not fully trust its data users: Data users in our system, who are assumed to be malicious, can only access records that are relevant to their assigned tasks, as determined by the TAs. We assume that at least one TA is honest. Data users also cannot learn attributes of the shared data records that are held by organizations other than the data client.

2.5.2 Definitions

We start by presenting the definition of selective security for our MA-OTABE scheme. Let $E = (Setup, AuthoritySetup, Encrypt, Distribute, KeyGen, Translate, Decrypt)$ be an OTABE scheme for a set of authorities Aut, $|Aut| = K$. Consider the following OTABE game for a PPT adversary $\mathcal{A}$, a challenger $\mathcal{B}$, a security parameter λ, an attribute universe $\mathcal{U}_{owner}$, and an attribute universe $\mathcal{U}_{client}$.

Init: The adversary chooses the challenge attribute set S, where $S \subseteq \mathcal{U}_{owner}$. Based on S, the adversary chooses the challenge decryption-parties set DEC_S^*, where $DEC_S^* \subseteq DEC_S$. The adversary also chooses a subset of corrupted authorities Aut_c. We assume that all authorities but one are corrupted and denote the honest authority by Aut_h; thus, $Aut = Aut_c \cup \{Aut_h\}$. The adversary sends Aut_c, Aut_h, S, and DEC_S^* to the challenger.

Setup: The challenger runs the $Setup$ algorithm to produce the public parameters PK and, for each authority Aut_i, runs the $AuthoritySetup$ algorithm to produce PK_i and MSK_i. If Aut_i is honest, the challenger sends PK_i to the adversary. If Aut_i is corrupted, the challenger sends both PK_i and MSK_i to the adversary.

Phase 1: The adversary chooses a revocation list RL and sends it to the challenger. It may then issue any polynomial number of key requests for tuples of the form (access structure, GID, task identifier) and send them to the challenger.

Given a request (access structure=$AC \in \mathcal{U}_{client}$, GID=$u$, task=$t$), the adversary proceeds as follows. For requests issued for a corrupted authority Aut_i, the adversary runs $SK_{iut} = KeyGen(PK, MSK_i, AC, u, t)$ itself, because it has MSK_i, given to it in the setup phase. For requests issued for the honest authority Aut_h, the challenger provides the answer. It extracts the time limit tl_t from the description of task t and creates a time-limit attribute $att_{LIMIT} = \langle DATE, <, tl_t \rangle$. In addition, given the GID, u, in the request, the challenger creates a GID attribute $att_{GID} = \langle GID, ==, u \rangle$. It then creates $AC' = AC \wedge att_{LIMIT} \wedge att_{GID}$, which is an updated version of AC, and performs:

- If $S \models AC'$ and $u \notin RL$, the challenger aborts.
- If $S \models AC'$ and $u \in RL$, then S must contain $S_{GID} = u$. The challenger picks GID u', $u' \neq u$, and generates the secret key using $SK_{hu't} = KeyGen(PK, MSK_h, AC, u', t)$.
- If $S \not\models AC'$, the challenger generates the secret key using $SK_{hut} = KeyGen(PK, MSK_h, AC, u, t)$.

Challenge: The adversary submits two messages m_0 and m_1 to the challenger. In addition, for every proxy j in DEC_S^*, it sends a bit a_j to the challenger. (By default, if j represents the user, we assume $a_j = 0$.) The challenger flips a fair coin b and encrypts m_b under S: $CT = Encrypt(m_b, PK, S, \{PK_i\}_{i \in Aut})$. Assuming I_{CT} is the index corresponding to the ciphertext CT, the challenger computes a set $\{C^j | j \in DEC_S^*\}$ of partial ciphertexts using $Distribute(I_{CT})$. For each proxy $j \in DEC_S^*$, if $a_j = 1$, the challenger performs a translation of the corresponding partial ciphertext, $C'^j = Translate(PK, j, C^j, \{PK_i\}_{i \in Aut})$, resulting in a translated partial ciphertext C'^j. Finally, it sends the ciphertext C^* to the adversary:

$$C^* = \bigcup_{j \in DEC_S^*} c_j^* \qquad c_j^* = \begin{cases} C'^j & \text{if } a_j = 1 \\ C^j & \text{if } a_j = 0 \end{cases}$$

Phase 2: Phase 1 is repeated.

Guess: The adversary outputs a guess b' of b. The advantage of the adversary in this game is defined as $\Pr[b' = b] - 0.5$.

Definition 2 An MA-OTABE scheme is ***selectively secure*** if all PPT adversaries have negligible advantage with respect to λ in the selective-security game.

In the proof that our MA-OTABE construction is secure, we use a q-type assumption about prime-order bilinear groups: the ***decisional*** q***-Bilinear*** (t, n)***-threshold Diffie-***

Hellman assumption ***((q,t,n)-DBTDH)***. It is similar to the Decisional q-Bilinear Diffie-Hellman assumption (q-DBDH) used by Rouselakis and Waters [96].[3]. The (q,t,n)-DBTDH assumption can be seen as a "threshold version" of the q-DBDH assumption: Instead of sending the attacker terms that contain the element z (q-DBDH), z is broken into n shares and some of the terms that the attacker receives contain only shares $\{z_c\}_{c\in V}$ of z, where V, $|V| = t$, is chosen by the attacker.

The assumption is parameterized by a security parameter λ, a suitably large prime p, two prime-order bilinear groups $G1$ and $G2$, a bilinear map $e : G1 \to G2$, and integers q, t, and n, where $n \geq 1$ is polynomial in λ, and $t \leq n$. It is defined by a game between a challenger and an attacker. The attacker chooses a subset $V \subseteq [n]$ of t indices and sends it to the challenger. The challenger picks a group element g uniformly at random from $G1$, $q+3$ exponents $x, y, z, b_1, b_2, \ldots, b_q$ independently and uniformly at random from Z_p, and $n-1$ additional exponents $z_1, \ldots, z_{n-1}$ independently and uniformly at random from Z_p. It sets $z_n = z - \sum_{c=1}^{n-1} z_c$. Then it sends $(p, G1, G2, e)$ and the following terms to the attacker:

$$g, g^x, g^y, g^z, g^{(xz)^2}$$

$$\forall l \in [q] : g^{b_l}, g^{xzb_l}, g^{xz/b_l}, g^{x^2zb_l}, g^{y/b_l^2}, g^{y^2/b_l^2}$$

$$\forall l, f \in [q], l \neq f : g^{yb_l/b_f^2}, g^{xyzb_l/b_f^2}, g^{(xz)^2b_l/b_f}, \Psi_{l,f}$$

where

$$\Psi_{l,f} = \{g^{xz_c(b_l/b_f)} | c \in V\}.$$

The challenger flips a fair coin b. If $b = 0$, it gives the term $e(g,g)^{xyz}$ to the attacker. Otherwise, it gives the attacker a term R chosen uniformly at random from $G2$. Finally, the attacker outputs its guess b' for the value of b.

[3]For convenience, we give the details of the q-DBDH assumption in Section 2.10.

Definition 3 We say that ***the*** (q, t, n)***-DBTDH assumption holds*** if all PPT attackers have at most a negligible advantage in λ in the above security game, where the advantage is defined as $\Pr[b' = b] - 1/2$.

2.6 Construction overview

2.6.1 Main OTABE techniques

Before presenting our construction in full detail, we present a simplified version that is inspired by the 69large-universe ABE scheme of Rouselakis and Waters [96] and that illustrates basic techniques that are new to our construction. Note that the scheme in [96] is single-authority; we extend it here to a multi-authority scheme.

Ciphertext composition in [96] is given by these equations:

$$C0 = Me(g,g)^{s\alpha} \qquad C1 = g^s \qquad C2_k = g^{f_k} \qquad C3_k = (\theta^{att_k} h)^{f_k} (w)^{-s}$$

The ciphertext is composed of a data layer and an attribute layer. We refer to $C0$ and $C1$ as *data-layer components*, $C2$ and $C3$ as *attribute-layer components*, and each element in $C3$ as an *attribute component*. The data-layer component $C0$ in [96] contains the message M masked by the public key $e(g,g)^{\alpha}$ of the (single) TA. Assuming that M is encrypted under a set S of attributes, the attribute layer contains $2|S|$ components, *i.e.*, two ($C2_k$ and $C3_k$) for each attribute att_k in the ciphertext. Each pair contains a uniform, randomly chosen term f_k that is local to the specific attribute att_k. $C3_k$ also contains the attribute att_k itself. The two layers are connected by the binder term s.

The basic idea of our construction is as follows. Assume that we have a data owner, a data client, two authorities (denoted Aut_1 and Aut_2), a client proxy, and a data user u.[4] Assume that the keys given to u by Aut_1 and Aut_2 are based on the access structures

[4]For clarity, we do not use intermediaries in this simplified construction.

$att_1 \vee att_2$ and att_4, respectively.

The data owner wishes to encrypt a record M under a set $S = \{att_1, att_3\}$ of attributes, where $att_1 \in U_{owner} \cap U_{client}$, but $att_3 \notin U_{owner} \cap U_{client}$. That is, att_3 does not belong to the client's vocabulary and hence needs to undergo translation before it can be used for decryption by u, using the keys she received from the authorities. In this example, we assume that $T(att_3) = att_4$; that is, a correct translation of the attribute $att_3 \in U_{owner}$ is $att_4 \in U_{client}$.

In order to encrypt M, the owner produces a two-level ciphertext; it is similar to the one in [96] but differs in several respects.

First, instead of creating $|S|$ attribute components $C3_k$, one for each attribute, the owner creates $|S| * |DEC_S|$ attribute components $C3_{k,j}$, one for each pair (attribute, decryption party), where DEC_S represents the set of parties that participate in the decryption of the ciphertext (*decryption-parties* set). In this example $|DEC_S| = 2$ because there are two decryption parties: the user and the client proxy.

Second, we use the binder term s differently from the way it is used by Rouselakis and Waters in [96]. In [96], the binder term is used in the data layer and in each attribute component. By contrast, we use secret sharing to break s into $|DEC_S|$ shares: each attribute component $C3_{k,j}$ contains only one share of the binder term, the one that corresponds to the decryption party j. In this example, there are two decryption parties: the user and the client proxy.

Third, recall that each attribute component in [96] contains the actual attribute to which it corresponds. In our OTABE scheme, however, each attribute component contains the output of a given transformation that is applied to the attribute. This enables the proxy to translate the attribute *obliviously* without knowing its label or value. In our construction, the transformation is a keyed PRF, but, as explained below, OTABE can accommodate a richer set of transformations in order to better serve each organization's business logic.

Fourth, we use another uniformly randomly chosen term, l_k. Like f_k, l_k is local to

the attribute component in which it appears. It is used to double blind the attribute part ($\theta^{att_k}h$) of each attribute component, using $d_k = f_k * l_k$ as a blinding factor; in this way, f_k can be used by the proxy as a token for oblivious translation.

Because of the composition of the ciphertext, the proxy is able to translate the attribute $att_3 \in U_{owner}$ into a new attribute $att_4 \in U_{client}$. The proxy uses the attribute component $C3_{att_3,proxy}$, an obfuscated version of the original attribute att_3, the tokens given to it by the $Encrypt()$ algorithm, and Equation 1 in the $Translate()$ algorithm, where $att_k{}'$ corresponds to the new attribute (in our case, att_4). In general, determination of the new attribute is done obliviously based on the obfuscated original attribute's label and value; this determination is explained fully in Subsection 2.7.2.

When the user receives the translated record from the proxy, she combines it with her own attribute-layer components and data-layer components to create the final aggregated ciphertext. She uses the keys that she received from Aut_1 and Aut_2 to decrypt the aggregated ciphertext.[5] Decryption with this equation uses secret sharing and the unique structure of the translated attribute component received from the proxy, which includes both an obfuscated version of the original attribute att_3 and the new attribute att_4.

Finally, to enable hidden access policy, we do not attach the actual set of attributes S to the ciphertext. Instead, both the data owner and the proxy compute an obfuscated value of each attribute they add to the ciphertext, based on the PEKS construction given in [20]. Using trapdoors received from the TAs, u is able to perform a "blind intersection" of the obfuscated values received with the ciphertext and her own obfuscated access structure's attributes received from the TAs. Thus, u is able to determine which attributes are required for decryption without learning their values.

[5]Decryption of aggregated ciphertexts is done using Equation 2, which is given (along with the rest of the full construction) in Subsection 2.7.1).

2.6.2 Other components of PRShare

PRShare combines the MA-OTABE construction in Subsection 2.7.1 with the following building blocks:

- Pseudorandom functions: The data owner and each organization org_j agree on two random k-bit keys $K_{org(j)}$ and $K1_{org(j)}$ for the PRFs $F_p : \{0,1\}^k \times \mathcal{U} \rightarrow \mathcal{U}$ and $F : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^*$.

- Collision-resistant hash function: If the parties wish to use the hidden-access-policy feature, they agree on a collision-resistant hash function H.

- Searchable-encryption (SE) scheme, Λ: The input to the $Distribute()$ algorithm is a set I of ciphertexts' ids. I is the output of Search_Λ, an SE scheme's Search protocol, executed by the CSP and a data user u. I contains the ids of ciphertexts whose associated attributes satisfy the conjunctive query ψ sent by u to the CSP.

- Translation function: In the setup phase, each organization org_j provides to its proxy the translation function T_j and the encrypted auxiliary information $E_j(L)$ according to which it should perform attribute translation.

Section 2.7 provides detailed descriptions of our MA-OTABE scheme and the associated attribute-translation procedure. However, for ease of exposition, it does not present these contributions in their maximum generality or explain all of their features. We briefly discuss some natural generalizations and interesting features here.

One essential feature of PRShare is *oblivious translation* of attributes in S_m by a semi-trusted proxy. Oblivious translation is accomplished by applying a transformation to the attribute inside each attribute component; this allows translation without disclosing the attributes' values to the proxy. The version of the full construction given in Subsection 2.7.1 applies the same transformation to each attribute in the ciphertext, using two PRFs. This

version demonstrates a specific translation operation in which the proxy performs oblivious equality tests and set-membership tests to determine the new attribute. However, PRShare supports a more flexible approach in which different transformations are applied to different attributes in the ciphertext, based on the attributes' types and sensitivities. For example, if $att_k \in \mathcal{U}_{owner}$ is a numeric attribute, the proxy can translate it into a descriptive attribute $att'_k \in \mathcal{U}_{client}$ by comparing att_k with a threshold that was provided to it by the organization that it represents. It determines the value of the new, descriptive attribute according to the result of that comparison. In such a case, we would choose an order-preserving transformation instead of an equality-preserving transformation. Based on this modular approach and other PRF-based transformations, PRShare enables a broader set of translation operations that better suit organizations' translation logic. These operations include oblivious addition, keyword search, and numeric comparison [26]. Subsection 2.7.2 contains concrete examples of attribute translation.

The full construction in Subsection 2.7.1 involves just one data client. In fact, a data owner in PRShare can encrypt its data records once for use by multiple data clients, and it need not know who the data clients are at encryption time. What it does need to know is the universe T of TAs from which each data client chooses the set of TAs that it will use.

At encryption time, the owner uses the public keys of all $t \in T$ to create $C1$, which is the data layer. It creates the rest of the ciphertext's components exactly as they are created in Subsection 2.7.1. Now consider a client c that uses TAs $T' \subseteq T$. In the key-generation phase, data users associated with c will receive two types of keys: regular secret keys, issued by each TA in T' according to the $keygen()$ algorithm, and dummy secret keys, issued by TAs in $T \setminus T'$. Each dummy key represents a "decrypt all" policy and thus has no effect when combined with the actual decryption policies represented by key issued by TAs in T'.

Dummy keys are issued to each data user once during the setup phase, and the total

number of TAs in the system is small. Furthermore, the attribute-layer components, which constitute the longer part of the ciphertext, remain the same under this generalization. Therefore, the performance of this generalized construction will be reasonable.

Query and retrieval of encrypted records in PRShare use a searchable encryption (SE) scheme Λ. There is a CSP that stores ciphertext records that the data owner has created using the $Encrypt()$ algorithm and receives from data users requests that contain conjunctive queries on attributes in $\mathcal{U}$. In PRShare, storage and processing of the data records (aka "payloads") is decoupled from storage and processing of their metadata. The SE scheme can be chosen independently of the OTABE scheme, according to specific needs or privacy requirements of the client or owner. The only functionality that the SE scheme must provide is:

1. The data user can submit to the CSP a conjunctive query that contains attributes in $\mathcal{U}_{owner} \cap \mathcal{U}_{client}$.

2. The CSP is able to retrieve all the records that match the query, without learning the query's contents or the attributes associated with each record. Furthermore, the data user cannot learn the attributes that are associated with each record, except for those that appear in her query.

Upon receiving a query from a data user, the CSP searches for all the ciphertexts that satisfy this query; for each one, it performs the $Distribute()$ algorithm. Importantly, *the CSP need not perform any type of authentication or authorization of users*. Each payload and its associated attributes are stored in encrypted form according to the OTABE scheme, and *only users with suitable keys are able to decrypt the payload and the attributes*. If a user does not belong to a client organization that uses TAs in T, or if she *does* belong to such an organization but has not been issued the necessary decryption keys for the records that match her queries, she will learn nothing from the encrypted payloads and attributes that the CSP sends her.

Finally, note that the choice of SE scheme is highly flexible. One may choose a very simple scheme, in which tags are created using PRFs with keys that shared among the relevant entities (owner, CSP, and clients) or a more sophisticated schemes that provides stronger security and privacy guarantees.

2.7 Detailed construction and translation function

2.7.1 Construction

We denote by $org(k)$ the organization that governs the attribute att_k. We denote by $pub_{\Pi}(P_{org_j})$ the public key of a proxy P_{org_j}, created using a standard public key encryption scheme, Π.

Our MA-OTABE scheme consists of the following algorithms:

$\boldsymbol{GlobalSetup}(\lambda) \Rightarrow (PK)$: This algorithm takes as input a security parameter λ. It defines bilinear groups $G1$,$G2$ of prime order p and a bilinear map $e : G1 \times G1 \rightarrow G2$. The attribute universe is $\mathcal{U} = Z_p$. Finally, the algorithm selects θ, h, w randomly from $G1$. It returns the global public key PK as follows:

$$PK = (G1, G2, p, \theta, w, h, g, e)$$

$\boldsymbol{AuthoritySetup}(PK) \Rightarrow (PK_i, MSK_i)$: Each authority Aut_i chooses random numbers $\alpha_i, \beta_i \in Z_p$. It sets $PK_i = (e(g,g)^{\alpha_i}, g^{\beta_i})$ as its public key and $MSK_i = (\alpha_i, \beta_i)$ as its master secret key.

$\boldsymbol{Encrypt}(M, PK, S, \{PK_i\}_{i \in Aut}) \Rightarrow (CT)$: This algorithm takes as input a data record's payload M, the public keys for all authorities $\{PK_i\}_{i \in Aut}$, and a set of attributes S, $|S| = R$. It adds two attributes to S: $att_{DATE} =< DATE, ==, rand_1 >$, $att_{GID} =< GID, ==, rand_2 >$. Both are randomly initialized. It then chooses $2|S| + 2$ random

exponents $s, a, \{f_k\}_{k\in[R]}, \{l_k\}_{k\in[R]} \in Z_p$ and computes $\{d_k = f_k * l_k\}_{k\in[R]}$. The encryptor determines, according to the nature of attributes in S, the subset ORG_S of organization proxies that are able to perform translations of the ciphertext. The set of parties involved in decryption of C will include the set of proxies in ORG_S and the final decryptor, *i.e.*, the user. Hence $|DEC_S| = |ORG_S| + 1 = P$. The encryptor chooses another P random elements $s_j \in Z_p$, $\sum_{j\in DEC_S} s_j = s$. It then encrypts M under S. The resulting ciphertext is composed of four elements: $C0, C1, C2, C3$ and a set Tok of tokens:

$$W = g^a \qquad C0 = M \prod_{i\in Aut} e(g,g)^{s\alpha_i} \qquad C1 = g^s \quad C2_k = \{g^{d_k} | att_k \in S\}$$

$$C3_{k,j} = \bigcup_{\substack{att_k\in S,\\ j\in DEC_S}} c3_{k,j} \qquad c3_{k,j} = \begin{cases} D_{k,j} & \text{if } att_k^L \in S_{im} \\ E_{k,j} & \text{if } att_k^L \in S_m \end{cases}$$

where:

$$D_{k,j} = (\theta^{att_k} h)^{d_k} (w)^{-s_j} \qquad E_{k,j} = (\theta^{F_p(K_{org(k)}, att_k)} h)^{d_k} (w)^{-s_j}$$

$$C2 = \{C2_k | att_k \in S\} \qquad C3 = \{C3_{k,j} | att_k \in S, j \in DEC_S\}$$

$$Tok_j = \{Tok_{k,j} | att_k^L \in S_j\}$$

$$Tok_{k,j} = [Tok1_{k,j} || Tok1_{k,j} || Tok1_{k,j} || Tok1_{k,j}]_{pub_\Pi(P_{org(k)})}$$

$$Tok1_{k,j} = \theta^{l_k} \qquad Tok2_{k,j} = f_k$$

$$Tok3_{k,j} = F(K1_{org(k)}, att_k^L) \qquad Tok4_{k,j} = F_p(K_{org(k)}, att_k)$$

$$C = (W, C0, C1, C2, C3, Tok = \{Tok_j | j \in ORG_S\})$$

For each attribute att_k where $att_k^L \in S_{im}$, the encryptor computes an obfuscated value as follows:

$Y^k = e((g^{\beta_{aut(k)}})^a, H(att_k))$, where $aut(k)$ denotes the authority that may use the attribute att_k in its access structure. The encryptor computes its signature, $sig_u^{encryptor}$ on each element in C, as well as on the number of attributes that each proxy in ORG_S is allowed to translate. In addition, for each proxy, it computes a signature $sig_p^{encryptor}$ on each element in $\{C3_{k,p} | att_k^L \in S_p\}$, on each element in Tok_j, and on the size of both sets. The encryptor then uploads the following record to the cloud server:

$$CT = (C, UID, P, Y = \{Y^k \mid \forall att_k^L \in S_{im}\}, sig_u^{encryptor}, \{sig_p^{encryptor} | p \in ORG_S\})$$

$\boldsymbol{KeyGen}(PK, MSK_i, A_i, u, t) \Rightarrow (SK_{i,u,t})$: The key generation algorithm for Aut_i, user u and task t takes as input the master secret key MSK_i and access structure A_i, determined by the authority based on the combination of data-centric attributes that it considers to be a sufficient justification for decrypting a data record's payload in the context of task t and the role of user u. The authority determines a new or updated time limit for task t, tl_t, and creates a time-limit attribute: $att_{LIMIT} =< DATE, <, tl_t >$. Lastly, given the user's GID u the authority creates a GID attribute, $att_{GID} =< GID, ==, u >$. Aut_i then creates $A_i' = A_i \wedge att_{LIMIT} \wedge att_{GID}$, an updated version of A_i. To ensure the hidden-access-policy feature, the authority replaces each attribute att_x in the access structure, with a trapdoor $H(att_x)^{\beta_i}$ and transforms the resulting access structure into an LSSS access structure $(M_i; \rho)$ where M_i is an $ni \times mi$ matrix and ρ is a function which associates rows of M_i to attributes' trapdoors. The algorithm chooses random $y_2, \ldots, y_{mi} \in Z_p^n$ and creates a vector $vi = (\alpha_i; y_2, \ldots, y_{mi})$. For $c = 1, \ldots, ni$, it calculates: $\lambda_{i,c} = M_i(c) \cdot vi$,

where $M_i(c)$ is the vector corresponding to the c'th row of the matrix M_i. In addition, the algorithm chooses ni random exponents $r_1, \ldots, r_{ni} \in Z_p$. For each $x \in [ni]$, it sets the private key $SK_{i,u,t}$ as:

$$SK^1_{x,i,u,t} = g^{\lambda_{i,x}}(w)^{r_x} \qquad SK^2_{x,i,u,t} = (\theta^{\rho(x)}h)^{-r_x} \qquad SK^3_{x,i,u,t} = g^{r_x}$$

Each authority Aut_i then sends:

$$SK_{i,u,t} = \{SK^1_{x,i,u,t}, SK^2_{x,i,u,t}, SK^3_{x,i,u,t}\}_{x\in[ni]}$$

to u. The user's secret keys for task t are $\{SK_{i,u,t}\}_{i\in Aut}$.

$\boldsymbol{Distribute}(I) \Rightarrow (\{C^j | j \in DEC_S\})$: The input to the $Distribute()$ algorithm is a set of ciphertexts' ids I. The cloud first retrieves all the ciphertexts that are associated with ids in I. For a ciphertext CT, encrypted under a set of attributes S and retrieved by the CSP, the CSP sends to each proxy, P_{org_p}:

$$C^p = (\{C3_{k,p} | att^L_k \in S_p\}, P, sig^{encryptor}_p, Tok_p)$$

and sends to user u:

$$C^u = \{W, C0, C1, C2, C3_u, P, Y, sig^{encryptor}_u)$$

where:

$$C3_u = \{C3_{k,u} | att_k \in S\} \cup \{C3_{k,p} | att_k \in S, org(k) \neq p\}$$

$\boldsymbol{Translate}(PK, j = p, C^p, \{PK_i\}_{i\in Aut}) \Rightarrow C'^p$: the $Translate()$ algorithm for a proxy P_{org_p} and a data record's payload M encrypted under attribute S receives as input a partial ciphertext C^p. For each attribute component $C3_{k,p}$ that corresponds to an attribute

att_k to be translated, the proxy first verifies the encryptor's signature. It then decrypts its tokens using its private key and extracts each of them. It computes $T_j(Tok3_{k,j}, Tok4_{k,j}) = att_k{}'$, thus obliviously translating the attribute att_k into a new attribute, att'_k. The function T_j is determined separately by each organization; see Subsection 2.7.2. It then computes a new value for $E_{k,p}$, $E'_{k,p}$:

$$E'_{k,p} = E_{k,p} \cdot (Tok1_{k,p}^{-PTok4_{k,p}+Patt_k{}'})^{Tok2_{k,p}} = (\theta^{(Patt_k{}'-(P-1)F_p(K_{org(k)},att_k))}h)^{d_k}w^{-s_p} \tag{2.1}$$

Finally, the proxy chooses a random exponent, $c \in Z_p$, where $W_p = g^c$, and computes, for each new attribute $att_k{}'$ that it created, an obfuscated value as follows: $Y^{att_{k'}} = e((g^{\beta_{aut(k')}})^c, H(att_{k'}))$. We denote the set of obfuscated value corresponding to translated attributes by proxy p as Y_p. It then signs the new elements it added, as well as the number of attributes it translated. It sends those signatures sig_p and the translated partial ciphertext to the user u. The record that is sent to the user is:

$$C'^p = (C'3_p, sig_p, W_p, Y_p) \qquad C'3_p = \{C'3_{k,p} | att_k^L \in S_p\} = \{E'_{k,p} | att_k^L \in S_p\}$$

$\boldsymbol{Decrypt}(PK, \{SK_{i,u,t}\}, C^u, \{C'^j | j \in ORG_S\}) \Rightarrow M$: The decryption algorithm for data record's payload M, encrypted under a set of attributes S and a user u takes as input the global parameters, K secret keys $\{SK_{i,u,t}\}$, representing access structures $\{A'_i\}$, and two types of ciphertexts: a partial ciphertext C^u received directly from the CSP and $|ORG_S| = P - 1$ translated partial ciphertexts $\{C'^j | j \in ORG_S\}$, received from each one of the proxies in ORG_S, $C'^j = (C'3_j, sig_j, W_j, Y_j)$. After verifying both the encryptor's signatures and the proxies' signatures, the user aggregates all the translated partial ciphertexts she received from the proxies, extracts $C3_u$ from her partial ciphertext

C^u, and creates and updated version of $C3$, $C'3$:

$$C'3 = \{C3_{k,u} | att_k \in S\} \cup \{C3_{k,p} | att_k \in S, org(k) \neq p\} \cup \{C'3_j | j \in ORG_S\}$$

The user extracts $C0, C1, C2$ from C^u and merges those with $C'3$. The final ciphertext is:

$$C_f = (C0, C1, C2, C'3)$$

The user then determines the attributes that are needed for decryption, as well as their corresponding rows in the LSSS matrix of each authority. For a given access policy, represented by (M_i, ρ), the user uses W, received from the CSP and $\{W_j\}_{j \in ORG_S}$, received from each proxy and computes the following set:

$$S_i^* = \bigcup_{i \in [ni]} s_i^* \qquad s_i^* = \begin{cases} e(W, \rho(i)) & \text{if } att_k^L \in S_{im} \\ e(W_{org(k)}, \rho(i)) & \text{else} \end{cases}$$

The user collects both the original attributes of the ciphertext and the ciphertext's translated attributes, to create the final set of attributes $TR(S) = Y \cup \{Y_j\}_{j \in ORG_S}$. By performing $\hat{S}_i = S_i^* \cap TR(S)$, she receives the (obfuscated) set of attributes $\hat{S}_i$ that are needed for decryption, I_i. This process is performed for each access policy $(M_i, \rho)_{i \in Aut}$, resulting in K obfuscated attribute sets I_i and corresponding index-sets Ind_i such that:

- For all $c \in Ind_i, \rho(c) \in I_i$.
- Exist constants, $\{w_{c,i} \in Z_p\}_{c \in Ind_i}$, such that $\sum_{c \in Ind_i} w_{c,i} M_i(c) = (1, 0, \ldots, 0)$

The algorithm now recovers M by computing:

$$\frac{C0}{B}$$

where:

$$B = \prod_{i \in Aut} \prod_{c \in I_i} (e(C1, SK^1_{c,i,u,t})(e(C2_c, SK^2_{c,i,u,t}))^P \prod_{j \in [P]} e(C'3_{c,j}, SK^3_{c,i,u,t}))^{w_{c,i}} \quad (2.2)$$

2.7.2 Translation

The $Translate()$ algorithm given in our construction assumes the existence of a set of translation functions, $\{T_j | j \in ORG\}$. Each function is determined separately by each organization org_j and determines how to translate attributes in S_j.

The translation of an attribute can be done in two ways: either by changing both the label and the value of the attribute, or by keeping the attribute's label and only changing its value. A translation may require auxiliary information, provided to the proxy by its organization. In such a case, the translation is done by performing an oblivious operation on the attribute, that is encrypted using a certain transformation, and on another object (a number, a list *etc*), the "auxiliary information," that is encrypted using the same transformation. Such an oblivious operation can be a comparison, equality test, list membership test, keyword search *etc*. Since both the attribute inside the ciphertext and the organization-specific auxiliary information are encrypted using the same keyed transformation, with a key that is unknown to the proxy, the proxy can perform the translation without learning the attribute's value and without learning the contents of the private auxiliary information provided by the organization.

On a high level, the transformation applied by organization org_j to a data structure L that contains multiple auxiliary information items, $l \in L$, works by treating each item l as the value of the corresponding attribute's label in S_{owner}, mapping the resulting attribute to an element in $\mathcal{U}$, and using the transformation to encrypt that element. The result, the encryption of auxiliary information L that belongs to an organization org_j, is denoted by $E_j(L)$. A similar process is used for auxiliary information that includes only one element

l such as a threshold or a descriptive statistic.

Each organization prepares a lookup table, where entries represent obfuscated labels and values contain the translation logic and auxiliary information used for translation of attributes with that label. Using the same obfuscated label received from the owner, the proxy knows what logic and auxiliary information should it use for the translation of the attribute in its hands. It then uses the obfuscated value (in our construction, a PRF-encrypted value) of the attribute that the proxy obtained from the owner to compute the new attribute, using the translation logic and auxiliary information.

We now present three important examples. For simplicity, in the following examples we fix a specific translation function and refer to it as T. In addition, we use T as if it takes one argument, namely the original attribute. In practice (as shown in our construction), in order to support *oblivious* translation, a function T_j is a two-argument function, neither contains the actual original attribute, but instead, an obfuscated version of both the label and the value. In our construction, we use two PRFs for that purpose.

Dynamic translation between vocabularies: As discussed, translation of an attribute from org_{owner}'s vocabulary to org_{client}'s vocabulary is done according to the specific attribute being translated as well as the specific needs and work methodologies of the client organization.

One of the main reasons why attribute translation is essential for supporting multiple vocabularies, is that while the encryption of a data record's payload is done by the owner once, the relevance of the data record to the client changes over time. In ABE terms, that means that while the set of attributes under which a ciphertext is encrypted, taken from one vocabulary, does not change, the question whether or not this set satisfies a given access policy, taken from another vocabulary, does change over time. Furthermore, such decision, of whether or not a ciphertext is relevant to the client at a given point in time is made using external information, the "auxiliary information," that is related to one or more

of the owner's, client's, and intermediaries' professional domains. Because the auxiliary information changes over time, so does the decision whether or not the set of attributes of a given data record should satisfy a given access policy. Values of such attributes with respect to a data record cannot be fully determined at encryption time, but should rather be dynamically translated, only when a data user needs to access that data record. OTABE supports such dynamic attributes, as shown below.

We consider here two examples, representing common translation operations used for translating attributes from $\mathcal{U}_{owner}$.

The first operation, is determining the new attribute according to the original attribute's membership in a list provided by the client organization or an intermediary. Since both the attribute and the list-items are encrypted using the output of a PRF, such translation can be done obliviously.

To illustrate, we continue with the watchlist example given in Subsection 2.2.2. Two pieces of metadata that ESPs collect about their customers' email messages are the sender and receiver of the email. Such attributes, however, cannot be used in the secret keys issued by the LE agency to its employees: unless the investigation is targeted (and therefore the data subject's UID such as phone number or email address are known in advance), a raw email address will be meaningless in terms of justification for decryption, and therefore cannot be used for determining the relevance of a certain ciphertext to one of the LE agency's investigations. Furthermore, exposing raw sender's and receiver's email addresses to agents in the LE agency will violate the privacy of data subjects that do not appear on any watchlist. Hence, the translation of the attribute "sender" is a boolean attribute that indicates whether the sender of the email appears on an existing watchlist. Such an attribute better suits the daily activity of the LE agency as well as protects innocent citizens' privacy and thus can be included in the key in order to determine whether an access to an email address is justified. Clearly, such a list cannot be revealed to an external entity, including the ESP.

Note that the raw email address's relevance to a given investigation may vary over time. This is because the auxiliary information, *i.e.*, the watchlist, may change periodically and thus the membership of a data subject associated with a given email address in the watchlist may change over time as well. This is why such attributes can only be translated dynamically, when an agent submits an access request for that specific email record.

We now show how the data client, the LE agency, encrypts the watchlist. The watchlist L contains multiple items, $l \in L$, that represent data subjects' ids (for example, email addresses). In order for the watchlist to be compatible with the "sender" attribute under which email messages are encrypted, the LE agency performs the following preprocessing step on the watchlist:

$for\ l\ in\ watchlist$:

$E_{j=client}(watchlist).add(F_p(K_{org_{j=client}}, < label = SENDER, operator = \text{“}==''$ $, value = l >))$

Where $E_{client}(watchlist)$ represents the resulting, encrypted watchlist, containing multiple "sender" attributes, and thus compatible with the "sender" attribute used by the ESP.

Assuming $att_k =< SENDER, ==, c >$ represents a sender's email address, c, $att_k{}' =< ON - WATCHLIST, ==, b >$ is a boolean attribute that represents whether the sender appears on a watchlist, and $E_{org_j}(L)$ represents an encryption of the watchlist L as described above, the value of b is determined by the proxy as follows:

$Contains(E_{org_j}(L), Tok4_{k,j}) = b$

The second operation, is determining the new attribute by comparing it to one or more numeric auxiliary information pieces that usually represent either a certain threshold that is related to the attribute's value, or aggregated statistics about other data records that share the same attribute. In this case, instead of an equality-preserving transformation, we will

use an order-revealing transformation such as the ORE scheme presented in [26], denoted by Π_{ORE} (which also makes use of a PRF). Since both the attribute and the threshold or the descriptive statistic with which the attribute is to be compared are encrypted using Π_{ORE}, such translation can be done obliviously.

To illustrate, we consider the insurance-company example discussed in Subsection 2.2.2. We consider the attribute: "credit utilization ratio," used by the CRA to store credit reports. Such attribute, however, cannot be used in the secret keys issued by the insurance company to its employees, as a raw number will be meaningless in terms of determining whether a consumer is a good candidate for an insurance offer, and therefore cannot be used to determine the relevance of a certain credit report to an employee's task. Furthermore, exposing the exact utilization ratio to insurance company's employees will violate consumers' privacy. Hence, the translation of the numeric attribute "credit utilization ratio" is a boolean attribute that indicates whether that ratio is below the average ratio. Such attribute better suits the daily activity of the insurance company as well as protects consumers' privacy to the extent possible. Therefore, it can be included in employees' keys in order to determine whether the insurance company finds the data subject a good-enough candidate for an insurance offer (in other words, whether an access to the data subject's credit report is justified). In this case the translation will be made by the credit card company's proxy, acting as an intermediary, by obliviously comparing the number that represents the consumer's credit utilization ratio to the average utilization ratio of its customers. Clearly, the average utilization ratio that is calculated by each credit card company based on its own customers, constitutes proprietary information of the company and should not be revealed to other organizations.

As in the previous example, the auxiliary information (the utilization ratio's average) may change periodically. Thus, whether or not the utilization ratio of a data subject is above average may change over time. This is why such an attribute can only be translated dynamically, when an employee submits an access request for that specific credit report.

Assuming $att_k =< CREDIT - UTILIZATION - RATIO, ==, c >$ represents the credit utilization ratio, c,
$att_k' =< IS - CREDIT - RATIO - LESS - THAN - AVERAGE, ==, b >$ is a boolean attribute that represents whether the credit utilization ratio is above the current average, as calculated by the credit card company, and $E_{org_j}(l)$ represents an encryption of the average l using Π_{ORE}, the value of b is determined by the proxy as follows:

$\Pi_{ORE}.COMPARE(E_{org_j}(l), Tok4_{k,j}) = b$

Note that in both cases, both the label and the value of the attributes are being translated.

Key-level revocation: $P_{org_{client}}$ translates the value of the attribute $att_k =< DATE, ==, rand >$ from the c-bit random value (its default value given at encryption time by the data owner) to the current date (or the current time stamp, if a time limit is expressed using time instead of dates) $date_{cur}$ and so $T(att_k) = att_k' =< DATE, ==, date_{cur} >$. In this case, only the value of the attribute is being translated. Note that access structures in our system contain a time-limit attribute of the form: $< DATE, <, tl_t >$, where tl_t is the per-task time limit, assigned by the TAs.

User-level revocation: Given a data user's GID u who sent a data retrieval request, $P_{org_{client}}$ performs the following: it first checks whether u appears in org_{client}'s revocation list. If so, it aborts. Otherwise, the proxy translates the value of the attribute $att_k =< GID, ==, rand >$ from the c-bit random value (its default value given at encryption time by the data owner) to the data user's GID u and so $T(att_k) = att_k' =< GID, ==, u >$. Note, that if the revocation list contains the data user's GID, the partial ciphertext $C'^{P_{org_{client}}}$ will not be sent to the data user who initiated the retrieval request. Furthermore, in such a case att_{GID} will remain with its default random value assigned by org_{owner}.

In both revocation events, only the attribute's value is being translated, as the original attributes serve as placeholders. Thus, the proxy only needs to know the original attributes' obfuscated labels in order to perform the translation.

2.8 Results

We now give the formal statements and full proofs of the properties of the scheme presented in Section 2.7.

Lemma 1 If $n \geq 2$ and $t \leq n$, then (q, t, n)-DBTDH $\Rightarrow$ q-DBDH.

From Definition 3, it is enough to prove that (q, n, n)-DBTDH $\Rightarrow$ q-DBDH.

Given a distinguisher D_1 which is able to tell a (q, n, n)-DBTDH term from a random term with non-negligible probability, we want to show that there exists a polynomial distinguisher D_2 which is able to tell a q-DBDH term from a random term with non-negligible advantage. We are given the terms:

$$\Omega_1 = \{g, g^x, g^y\} \cup \{g^{b_l}, g^{y/b_l^2}, g^{y^2/b_l^2} | \forall l \in [q]\} \cup \{g^{yb_l/b_f^2} | \forall l, f \in [q], l \neq f\}$$

$$\Omega_2 = \{g^z, g^{(xz)^2}\} \cup \{g^{xzb_l}, g^{xz/b_l}, g^{x^2zb_l} | \forall l \in [q]\} \cup \{g^{xyzb_l/b_f^2}, g^{(xz)^2b_l/b_f}, g^{xzb_l/b_f} | \forall l, f \in [q], l \neq f\}$$

and R, where R is either a q-DBDH term $e(g, g)^{xyz}$ or a random term. We choose a set $A = \{a_i\}$, where each a_i is randomly selected from Z_p (note that for the (q, n, n)-DBTDH, V is uniquely determined, as $V = [n]$) and compute, for each element in Ω_2, a new term:

$$h1 = (g^z)^{\sum_{i=1}^n a_i} = g^{(\sum_{i=1}^n a_i z)}$$

$$h2 = (g^{(xz)^2})^{(\sum_{i=1}^n a_i)^2} = g^{(x(\sum_{i=1}^n a_i z))^2}$$

$$h3 = (g^{xzb_l})^{\sum_{i=1}^n a_i} = g^{x(\sum_{i=1}^n a_i z)b_l}$$

$$h4 = (g^{xz/b_l})^{\sum_{i=1}^{n} a_i} = g^{x(\sum_{i=1}^{n} a_i z)/b_l}$$

$$h5 = (g^{x^2 z b_l})^{\sum_{i=1}^{n} a_i} = g^{x^2(\sum_{i=1}^{n} a_i z) b_l}$$

$$h6 = (g^{xyzb_l/b_f^2})^{\sum_{i=1}^{n} a_i} = g^{xy(\sum_{i=1}^{n} a_i z) b_l/b_f^2}$$

$$h7 = (g^{(xz)^2 b_l/b_f})^{(\sum_{i=1}^{n} a_i)^2} = g^{(x(\sum_{i=1}^{n} a_i)z)^2 b_l/b_f},$$

$$\Psi'_{l,f} = \{(g^{xzb_l/b_f})^{a_i} | i \in [n]\} = \{g^{x(a_i z) b_l/b_f} | i \in [n]\}$$

We set Ω'_2 as:

$$\Omega'_2 = \{h1, h2\} \cup \{h3, h4, h5 || \forall l \in [q]\} \cup \{h6, h7, \Psi'_{l,f} | \forall l, f \in [q], l \neq f\}$$

Note, that if R is a q-DBDH term, then $R' = R^{(\sum_{i=1}^{n} a_i)} = e(g,g)^{xy(\sum_{i=1}^{n} a_i z)}$ is a (q,n,n)-DBTDH term, and if R is a random term then R' is a random term. We then view $(\Omega_1, \Omega'_2, R')$ as input to the oracle D_1 to obtain correct value $b \in \{0,1\}$ ($b = 0$ if the answer of D_1 is (q,n,n)-DBTDH term, and 1 otherwise). Therefore, we have a polynomial distinguisher D_2 which is able to tell q-DBDH term from a random term with same non-negligible advantage.

Theorem 1 If (q,n,n)-DBTDH holds, then our MA-OTABE scheme achieves selective security against all PPT adversaries with a challenge attribute set S of size W, where $W \leq q$, and a challenge decryption-parties set DEC_S^* of size P, where $P \leq n$.

To prove the theorem we will assume that there exists a PPT adversary $\mathcal{A}$ with a challenge attribute set S and a challenge decryption-parties set DEC_S^*, which has a non-negligible advantage in selectively breaking our MA-OTABE scheme. Using $\mathcal{A}$ we will build a PPT simulator $\mathcal{B}$ that attacks the (q,n,n)-DBTDH assumption with a non-negligible advantage.[6].

[6]For simplicity, we prove our attribute-secrecy related claims separately, in Theorem 3. We also omit the signatures that are attached to some of the messages in our construction

Init: The simulator receives the given terms from the assumption. The adversary chooses the challenge attribute set S, where $|S| = W$. Based on S, the adversary chooses the challenge decryption-parties set DEC_S^*, where $DEC_S^* \subseteq DEC_S$ and $|DEC_S^*| = P$. The adversary chooses a subset of corrupted authorities Aut_c. We assume all authorities but one are corrupted, and denote the honest authority by Aut_h. The adversary sends Aut_c, Aut_h, S and DEC_S^* to the simulator.

Setup: We denote S as $\{att_1, \ldots, att_W\}$ and the set of indexes of attributes in S as I_S.

The simulator chooses h^*, u^* randomly from Z_p. For each attribute att_l it chooses e_l randomly from Z_p. It then computes the global public parameters:

$$w = g^x$$

$$\theta = g^{u^*} \prod_{l \in I_S} (g^{y/b_l^2})$$

$$h = g^{h^*} \prod_{l \in I_S} (g^{xz/b_l e_l}) \prod_{l \in I_S} (g^{y/b_l^2})^{-att_l}$$

Based on the global public parameters, the simulator creates the parameters for authority Aut_i, as follows:

For every $Aut_i \in Aut_c$, the simulator chooses random $n_i \in Z_p$, and sets $MSK_i = -xn_i$. It computes $PK_i = e(g,g)^{MSK_i} = e(g^x, g^{-(n_i)})$. The simulator sends MSK_i and PK_i to the adversary. For Aut_h, the simulator sets $MSK_h = xy + x \sum_{i \in Aut_c} n_i$. It computes $PK_h = e(g^x, g^y) \prod_{i \in Aut_c} e(g^x, g^{n_i})$. The simulator sends only PK_h to the adversary.

Phase 1: The adversary chooses a revocation list RL and sends it to the simulator. Then it may issue any polynomial number of private key queries, for tuples of (access structure, GID, task identifier), and sends those to the simulator.

For a query: (access structure=AC, GID=u, task=t), the simulator does the following:

For queries issued for a corrupted authority $Aut_i \in Aut_c$, the adversary runs $SK_{iut} = KeyGen(PK, MSK_i, AC, u, t)$ itself, as it has MSK_i, given to it in the setup phase. For queries issued for the honest authority Aut_h, the simulator provides the answer. The simulator determines a time limit for task t, tl_t, and creates a time-limit attribute: $att_{LIMIT} =< DATE, <, tl_t >$. In addition, given the GID in the query u the simulator creates a GID attribute, $att_{GID} =< GID, ==, u >$. It then creates an updated version of AC, $AC' = AC \wedge att_{LIMIT} \wedge att_{GID}$ and performs the following:

- If $S \models AC'$ and $u \notin RL$, the simulator will abort.
- If $S \models AC'$ and $u \in RL$, S must contain $S_{GID} = u$. The simulator picks a GID u', $u' \neq u$, and generates the secret key using $SK_{hu't} = KeyGen(PK, MSK_h, AC, u', t)$
- If $S \not\models AC'$, the simulator generates the secret key using $SK_{hut} = KeyGen(PK, MSK_h, AC, u, t)$.

We will now show how the simulator produces the secret keys in the last two cases.

- In the second case, the simulator first creates $att'_{GID} =< GID, ==, u' >$. Then it needs to create a key for $AC^* = AC \wedge att'_{GID} \wedge att_{LIMIT}$.
- In the third case, the simulator needs to create a key for $AC^* = AC'$.

Those access policies are represented by an LSSS matrix M^{AC^*} with dimensions $l \times n$ and a row-mapping function ρ. Note, that in both cases S is not authorized for AC^*. Hence, we can split M^{AC^*}'s rows into two sets:

$$A = \{r | r \in [l], \rho(r) \in S\} \qquad B = \{r |\ r \in [l], \rho(r) \notin S\}$$

where $A, B \neq \emptyset$. Since S is not authorized for M^{AC^*}, from the properties of LSSS we can find a vector $\beta \in Z_p^n$ with $\beta_1 = 1$ fixed such that $\forall r \in A$, $M_r^{AC^*}\beta = 0$.

The simulator then chooses uniformly at random $n-1$ random elements in Z_p, v_i, and sets the shares of MSK_h as:

$$\lambda_r =< M_r^{AC*}, \Theta >$$

Where:

$$\Theta = MSK_h\beta + (0, v_2, \ldots, v_n)^{\perp}$$

Hence, row's r share is:

$$\lambda_r =< M_r^{AC^*}, (MSK_h\beta + (0, v_2, \ldots, v_n)^{\perp}) >=$$

$$xy < M_r^{AC^*}, \beta > +x \sum_{i \in Aut_c} n_i < M_r^{AC^*}, \beta > + < M_r^{AC^*}, (0, v_2, \ldots, v_n)^{\perp} >=$$

$$xy < M_r^{AC^*}, \beta > +x \sum_{i \in Aut_c} n_i < M_r^{AC^*}, \beta > +\lambda_r'$$

Now, let us see how the simulator computes the secret key for $r \in A$: From definition, $r \in A \rightarrow \rho(r) \in S$. From LSSS properties, $< M_r^{AC^*}, \beta >= 0$. Thus in this case,

$$\lambda_r = \lambda_r' =< M_r^{AC^*}, (0, v_2, \ldots, v_n)^{\perp} >$$

and hence, its value is known to the simulator. The simulator can then compute the key components SK^1, SK^2, SK^3 as in the $KeyGen$ algorithm:

$$SK^1 = g^{\lambda_r} w^{t_r} = g^{\lambda_r'} g^{x a_r}$$

$$SK^2 = (\theta^{\rho(r)} h)^{-t_r} = ((g^{u^*} \prod_{l \in I_S} (g^{y/b_l^2}))^{\rho(r)} (g^{h^*} \prod_{l \in I_S} (g^{xz/b_l e_l}) \cdot \prod_{l \in I_S} (g^{y/b_l^2})^{-att_l}))^{-a_r}$$

$$SK^3 = g^{t_r} = g^{a_r}$$

where $t_r = a_r$ are randomly selected from Z_p by the simulator and $\lambda_r = \lambda_r'$.

Finally, for $r \in B$, the simulator will compute the secret key in the following way: From definition, $r \in B \rightarrow \rho(r) \notin S$. In this case, the simulator will define

$$t_r = -\sum_{i \in Aut_c} (n_i) < M_r^{AC^*}, \beta > -y < M_r^{AC^*}, \beta > + \sum_{l \in I_S} \frac{xzb_l < M_r^{AC^*}, \beta >}{\rho(r) - att_l} + t'_r$$

where t'_r is randomly selected from Z_p. Hence the key components can be computed as:

$$SK^1 = g^{\lambda_r} w^{t_r} = g^{\lambda'_r} \prod_{l \in [n]} (g^{x^2 z b_l})^{<M_r^{AC^*}, \beta>/(\rho(r) - att_l)} \cdot g^{x t'_r}$$

$$SK^2 = (\theta^{\rho(r)} h)^{-t_r} = g^{\sum_{i \in Aut_c}(n_i) < M_r^{AC^*}, \beta > (\rho(r) u^* + h^*)} (g^y)^{<M_r^{AC^*}, \beta > (\rho(r) u^* + h^*)}$$

$$\cdot \prod_{l \in I_S} (g^{xzb_l})^{-(\rho(r) u^* + h^*) < M_r^{AC^*}, \beta > / (\rho(r) - att_l)}$$

$$\cdot \prod_{(l,f) \in I_S} (g^{(xz)^2 b_f / b_l e_l})^{-<M_r^{AC^*}, \beta>/(\rho(r) - att_f)} \cdot \prod_{l \in I_S} (g^{y^2 / b_l^2})^{<M_r^{AC^*}, \beta > (\rho(r) - att_l)}$$

$$\cdot \prod_{l \in I_S} (g^{xz / b_l e_l})^{\sum_{i \in Aut_c}(n_i) < M_r^{AC^*}, \beta>} \cdot \prod_{l \in I_S} (g^{y / b_l^2})^{\sum_{i \in Aut_c}(n_i) < M_r^{AC^*}, \beta > (\rho(r) - att_l)}$$

$$\cdot \prod_{\substack{(l,f) \in I_S \\ l \neq f}} (g^{xzy(b_f / b_l^2)})^{-<M_r^{AC^*}, \beta > (\rho(r) - att_l) / (\rho(r) - att_f)} \cdot (\theta^{\rho(r)} h)^{-t'_r}$$

$$SK^3 = g^{t_r} = (g)^{-\sum_{i \in Aut_c}(n_i) < M_r^{AC^*}, \beta >} (g^y)^{-<M_r^{AC^*}, \beta>} \cdot \prod_{l \in I_S} (g^{xzb_l})^{<M_r^{AC^*}, \beta>/(\rho(r) - att_l)} \cdot g^{t'_r}$$

Therefore, in both cases the simulator can reply to the adversary's query with the entire secret key. Note, that since $AC, AC' \subseteq \mathcal{U}_{client}$, and $S_{GID}, S_{LIMIT} \in \mathcal{U}_{client}$, $AC^* \subseteq \mathcal{U}_{client}$, and so does the secret key given to the adversary. In addition, all the secret key's terms, both for A and for B can be calculated by the simulator using terms from the assumption, the challenge set S (chosen by the adversary), and the access structure AC (chosen by the adversary).

Challenge: $\mathcal{A}$ submits two messages, m_0 and m_1 to the simulator. In addition, for every proxy in DEC_S^*, j, it sends a bit a_j to the simulator. The simulator then flips a random coin b and encrypts m_b under S: $CT = Encrypt(m_b, PK, S, \{PK_i\}_{i \in Aut})$, by implicitly setting $s = z$, $\{l_k = b_k | \forall k \in I_S\}$, $\{f_k = e_k | \forall k \in I_S\}$, $\{d_k = b_k e_k | \forall k \in I_S\}$ and $\{s_j = z_j | \forall j \in DEC_S^*\}$. For each proxy $j \in DEC_S^*$, the simulator creates a partial ciphertext $C^j = (\{C3_{k,j} | att_k^L \in S_j\}, P, Tok_j)$ using the $Distribute$ algorithm, and, if $a_j = 1$, performs $C'^j = Translate(PK, j, C^j, \{PK_i\}_{i \in Aut})$. Note, that for every proxy j such that $a_j = 1$, if an attribute $att_k^L \in S_j$, the simulator has two attributes in its hands: the original attribute att_k and the translated attribute, $att_{k'}$. Finally, the simulator extracts $C0, C1, C2, C3 = \{C3_{k,j} | att_k \in S, j \in DEC_S^*\}, Tok$ from CT, and extracts $C'3_j = \{C'3_{k,j} | att_k^L \in S_j\}$ from each translated partial ciphertext, C'^j. The simulator then sends the translated ciphertext C^* to $\mathcal{A}$. Note, that each element in C^* can be computed using terms from the assumption:

$$C^* = \{C0, C1, C2, C^*3, Tok\}$$

where:

$$C0 = m_b \cdot e(g,g)^{xys} = m_b \cdot R \qquad C1 = g^s = g^z \qquad C2 = \{g^{d_k} | att_k \in S\} = \{g^{b_k e_k} | att_k \in S\}$$

$$C^*3 = \bigcup_{\substack{att_k \in S, \\ j \in DEC_S^*}} c^*3_{k,j} \qquad c^*3_{k,j} = \begin{cases} C'3_{k,j} & \text{if } att_k^L \in S_j \wedge a_j = 1 \\ C3_{k,j} & \text{otherwise} \end{cases}$$

From the construction,

$$c^*3_{k,j} = \begin{cases} D_{k,j} & \text{if } att_k^L \in S_{im} \\ E_{k,j} & \text{if } (att_k^L \in S_m \wedge att_k^L \notin S_j) \vee (att_k^L \in S_j \wedge a_j = 0) \\ E'_{k,j} & \text{if } att_k^L \in S_j \ \wedge \ a_j = 1 \end{cases}$$

Now, the simulator can compute the following terms using terms from the assumption:

$$D_{k,j} = ((g^{u^*} \prod_{l \in I_S} (g^{y/b_l^2}))^{att_k} (g^{h^*} \prod_{l \in I_S} (g^{xz/b_l e_l}) \cdot \prod_{l \in I_S} (g^{y/b_l^2})^{-att_l}))^{b_k e_k} g^{-xz_j} =$$

$$g^{b_k e_k (u^* att_k + h^*)} \prod_{l \in I_S} g^{xz b_k e_k / b_l e_l} \prod_{l \in I_S} g^{y b_k e_k (att_k - att_l)/b_l^2}) g^{-xz_j} =$$

$$g^{b_k e_k (u^* att_k + h^*)} \prod_{l \in I_S} \prod_{c \in [P]} g^{xz_c b_k e_k / b_l e_l} \prod_{l \in I_S} g^{y b_k e_k (att_k - att_l)/b_l^2} \cdot g^{-xz_j} =$$

$$g^{b_k e_k (u^* att_k + h^*)} \prod_{l \in I_S} \prod_{\substack{c \in [P] \\ (l,c) \neq (k,j)}} g^{xz_c b_k e_k / b_l e_l} \cdot \prod_{l \in I_S} (g^{y b_k e_k / b_l^2})^{att_k - att_l}$$

$$E_{k,j} = ((g^{u^*} \prod_{l \in I_S} (g^{y/b_l^2}))^{F_p(K_{org(k)}, att_k)} (g^{h^*} \prod_{l \in I_S} (g^{xz/b_l e_l}) \cdot \prod_{l \in I_S} (g^{y/b_l^2})^{-att_l}))^{b_k e_k} g^{-xz_j} =$$

$$g^{b_k e_k (u^* F_p(K_{org(k)}, att_k) + h^*)} \prod_{l \in I_S} g^{xz b_k e_k / b_l e_l} \cdot \prod_{l \in I_S} g^{y b_k e_k (F_p(K_{org(k)}, att_k) - att_l)/b_l^2}) g^{-xz_j} =$$

$$g^{b_k e_k (u^* F_p(K_{org(k)}, att_k) + h^*)} \prod_{l \in I_S} \prod_{c \in [P]} g^{xz_c b_k e_k / b_l e_l} \cdot \prod_{l \in I_S} g^{y b_k e_k (F_p(K_{org(k)}, att_k) - att_l)/b_l^2} \cdot g^{-xz_j} =$$

$$g^{b_k e_k (u^* F_p(K_{org(k)}, att_k) + h^*)} \prod_{l \in I_S} \prod_{\substack{c \in [P] \\ (l,c) \neq (k,j)}} g^{xz_c b_k e_k / b_l e_l} \cdot \prod_{l \in I_S} (g^{y b_k e_k / b_l^2})^{F_p(K_{org(k)}, att_k) - att_l}$$

$$E'_{k,j} = \left(\theta^{(Patt_k{}' - (P-1)F_p(K_{org(k)}, att_k))} h\right)^{d_k} w^{-s_p} =$$

$$g^{b_k e_k (u^*(Patt_k{}' - (P-1)F_p(K_{org(k)}, att_k)) + h^*)} \prod_{l \in I_S} g^{xzb_k e_k / b_l e_l}$$

$$\cdot \prod_{l \in I_S} g^{yb_k e_k ((Patt_k{}' - (P-1)F_p(K_{org(k)}, att_k)) - att_l)/b_l^2} g^{-xz_j} =$$

$$g^{b_k e_k (u^*(Patt_k{}' - (P-1)F_p(K_{org(k)}, att_k)) + h^*)} \cdot \prod_{l \in I_S} \prod_{c \in [P]} g^{xz_c b_k e_k / b_l e_l}$$

$$\cdot \prod_{l \in I_S} g^{yb_k e_k ((Patt_k{}' - (P-1)F_p(K_{org(k)}, att_k)) - att_l)/b_l^2} g^{-xz_j} =$$

$$g^{b_k e_k (u^*(Patt_k{}' - (P-1)F_p(K_{org(k)}, att_k)) + h^*)} \cdot \prod_{l \in I_S} \prod_{\substack{c \in [P] \\ (l,c) \neq (k,j)}} g^{xz_c b_k e_k / b_l e_l}$$

$$\cdot \prod_{l \in I_S} (g^{yb_k e_k / b_l^2})^{Patt_k{}' - (P-1)F_p(K_{org(k)}, att_k) - att_l}$$

$$Tok = \bigcup_{\substack{att_k \in S, \\ att_k^L \in S_j}} Tok_{k,j} = \bigcup_{\substack{att_k \in S, \\ att_k^L \in S_j}} (Tok1_{k,j}, Tok2_{k,j}, Tok3_{k,j}, Tok4_{k,j})$$

$$Tok1_{k,j} = (g^{u^*} \prod_{l \in I_S} (g^{y/b_l^2}))^{b_k} = (g^y)^{u^*} \prod_{l \in I_S, l \neq k} (g^{yb_k/b_l^2}) \qquad Tok2_{k,j} = e_k$$

$$Tok3_{k,j} = F(K1_{org(k)}, att_k^L) \qquad Tok4_{k,j} = F_p(K_{org(k)}, att_k)$$

Phase 2: Phase 1 is repeated.

Guess: The adversary outputs a guess b' of b. If $b = b'$ the challenger outputs 0, *i.e.* it claims that the challenge term is $R = e(g,g)^{xyz}$. Otherwise, it outputs 1 to indicate that it believes R is a random group element.

If $R = e(g,g)^{xyz}$ then $\mathcal{A}$ played the proper security game, because $C = m_b \cdot R = m_b \cdot e(g,g)^{xys}$. On the other hand, if R is a random term then all information about the

message m_b is lost in the challenge ciphertext. Therefore the advantage of $\mathcal{A}$ is exactly 0. As a result if $\mathcal{A}$ breaks the proper security game with a non negligible advantage, then $\mathcal{B}$ has a non negligible advantage in breaking the (q, n, n)-DBTDH assumption.

Theorem 2 Let $C = (M)_S$ be a MA-OTABE ciphertext. No coalition of at most $|DEC_S| - 1$ parties can learn anything about M.

Let M be a message to be encrypted under an MA-OTABE scheme and a set of attributes S. Let $|DEC_S| = n$. Additionally, let Φ be a colluding set, $|\Phi| = n - 1$.

From our definition, the colluding set might contain either all the proxies in ORG_S or a subset of $|ORG_S| - 1$ proxies from ORG_S and the data user. Since the information that the colluding set has in the former case is a subset of the information that it has in the latter case, it is enough to prove the lemma for the latter case.[7]

We claim that no information about the data layer, represented by data-layer components $C0$ and $C1$, can be inferred from translation tokens, as well as any combination of attribute-layer components held by members of Φ.

Attribute-layer components include $C2 = \{C2_k | att_k \in S\}$ and $C3 = \{C3_{k,j} | att_k \in S, j \in DEC_S\}$. Each attribute component $C3_{k,j}$ includes two parts: a "local randomness" part, either $(\theta^{att_k} h)^{d_k}$ or $(\theta^{F_p(K_{org(k)}, att_k)} h)^{d_k}$ and a binder-term part $(w)^{-s_j}$, while $C2 = \{g^{d_k} | att_k \in S\}$ contains only a "local randomness" part.

Since each f_k, l_k are chosen uniformly at random, so does d_k. In addition, each d_k is local to the attribute-layer components corresponding to att_k, $\{C2_k\}, \{C3_{k,j}\}$ in which it appears, and does not appear in attribute-layer components that correspond to other attributes, and, more importantly, in the data layer. Therefore, the local-randomness parts of any combination of attribute-layer components $C2$ and $C3$ are independent of the data

[7] Theoretically, the colluding set may include proxies that do not belong to ORG_S; because they do not participate in the translation of C, they are not given any partial ciphertext related to C. Since each ciphertext's elements are individually randomized, even if such proxies are part of the colluding set, they will not be able to provide any information that may contribute to the collusion attempt.

layer. Furthermore, each token, set of tokens or combination of sets of tokens in $Tok = \{Tok_j | j \in \Phi\}$ are also independent of the data layer. This means that, any information about the data layer, and therefore, the plaintext, can only be obtained using a combination of binder term parts taken from various attribute components $C3_{k,j}$ that parties $j \in \Phi$ hold. We now show that any such combination provides no information about the the binder term, and hence, no information on the data layer.

Let us look at any $n-1$ combination of binder term parts held by members of Φ. We define the following random variables:

$$\{S_i\}_{1 \leq i \leq n} = w^{-s_i}$$

Consider a set of $n-1$ binder-term parts, T. If T does not include S_n, it must contain $S_1, \ldots, S_{n-1}$. Since $s_1, \ldots, s_{n-1}$ are chosen uniformly at random from Z_p, $S_1, \ldots, S_{n-1}$ are independent random variables. If T does include S_n, assume without loss of generality that $T = \{S_1, S_2, \ldots, S_{n-2}, S_n\}$. Clearly, the first $n-2$ variables are independent. $S_n = w^{-s_n} = w^{-s+\sum_{i=1}^{n-1} s_i} = w^{-(s+\sum_{i=1}^{n-1} -s_i)} = (w^s S_{n-1} \prod_{i=1}^{n-2} S_i)^{-1}$ depends on $S_1, S_2, \ldots, S_{n-2}$, but also on the missing S_{n-1}. S_{n-1} is independent of $S_1, \ldots, S_{n-2}$. Each distinct value of S_{n-1} produces a distinct value of S_n. Therefore S_n is independent of the first $n-2$ S_i's. In both cases, all $n-1$ elements in T appear to the adversary as independent random numbers, and give no information about the binder term.

Thus, for any colluding set of size at most $n-1 = |DEC_S| - 1$, any combination of tokens and attribute-layer components which members of Φ are given as part of the protocol provide no information about the data layer, and hence, about the plaintext M.

Lemma 2 Let $C = (M)_S$ be a MA-OTABE ciphertext. The proxies in an MA-OTABE scheme cannot learn anything about M, even if they *all* collude.

The proof follows from Theorem 2 since every colluding set of at most $|DEC_S| - 1$

parties cannot learn any information about M and, by definition, only $|DEC_S| - 1$ proxies participate in each ciphertext's translation.

Theorem 3 Let F and F_p be two PRFs used in the construction of our MA-OTABE scheme. If F and F_p are secure, then the scheme achieves attribute secrecy.

Consider a message M, encrypted under a set of attributes S resulting in a ciphertext C.

Hidden access policy: We consider both the servers, and the data users:

CSP, proxies: The set of attributes Y that is stored with the ciphertext on the CSP includes only the obfuscated values of immutable attributes from S. In addition, neither the CSP nor the proxies are given any trapdoors for attributes in S. Thus, Y is hidden from the servers (for more details, the reader is referred to [20]).

Apart from Y, an attribute $att_k \in S$ may appear in the ciphertext only within the attribute components $\{C3_{k,j}\}$ to which the attribute corresponds. Immutable attributes can only appear within $D_{k,j}$, as the exponent of θ inside the local randomness part. Since each local randomness part in which the attribute itself appears is blinded by a local, uniformly random chosen element, known only to the owner, the attribute remains hidden. Mutable attributes in S can appear within $E_{k,j}$ or $E'_{k,j}$, as the exponent of θ inside the local randomness part, or in $Tok3_{k,j}, Tok4_{k,j}$. In both $E_{k,j}$ and $E'_{k,j}$, each local randomness part in which the attribute itself appears is blinded by a local, uniformly random chosen element, known only to the owner. Furthermore, In both $E_{k,j}$, $Tok3_{k,j}$ and $Tok4_{k,j}$, either att_k or att_k^L only appear in their encrypted form, using a keyed PRF with a key that is unknown to the CSP, or to any proxy. Lastly, each PRF-encrypted term inside $Tok3_{k,j}$ and $Tok4_{k,j}$ is encrypted using the public key of the proxy who is allowed to translate att_k ("the translator"). Hence, mutable attributes inside the ciphertext remain hidden as well.

Data users: We start by defining the term "terminal attributes." Terminal attributes include either immutable attributes, or attributes which are the result of an attribute trans-

lation performed by one of the proxies. Intuitively, those are the attributes that the data user will eventually receive with the ciphertext, and thus must be kept hidden from the user, in such a manner that enables her to know which attributes should she use for decryption.

Each immutable attribute in S is replaced by the owner at encryption time by an obfuscated value of that attribute,
$e((g^{\beta_{aut(k)}})^c, H(att_k))$, derived from the PEKS construction in [20] where c is a random number, creating the set Y.

When a proxy P_{org_j} performs a translation of an attribute, it computes an obfuscated value of the new attribute that it created, and only that obfuscated value is attached to the translated partial ciphertext that it sends to the user, as Y_j.

The data user never receives the actual S. Instead, it receives $TR(S) = Y \cup \{Y_j\}_{j \in ORG_S}$ where Y represents immutable attributes in S and $\{Y_j\}$ represent the set of mutable attributes in S. Hence, all the *terminal attributes* are obfuscated, and therefore remain hidden from the user (for more details, the reader is referred to [20]). Note, however, that unlike the servers, data users do hold trapdoors for attributes that appear in their access policies, $H(att_k)^{\beta_{aut(k)}}$, and those trapdoors do leak some information about the attributes in S. Such leakage to the data user is limited to those attributes in S that also appear in the user's access policy; that is, the user learns nothing about attributes in S that are not in her access policy. Such leakage includes, for instance, the ability of the user to know whether an attribute in S, that also appears in the user's access policy, appeared in previous ciphertexts that the user has retrieved from the CSP (we note that the source of such leakage is the transitivity of the equality operation, not the attributes' actual values. The user is not able to learn any of the attributes in S, even for those attributes that appear in her access policy).

Oblivious translation: A proxy P_{org_j} uses its partial ciphertext, its tokens, and auxiliary information in order to perform a translation of an attribute att_k. We claim that neither of

the items above reveals the attribute att_k.

Within the partial ciphertext, an attribute att_k such that $att_k^L \in S_j$ can only appear in the attribute components $\{C3_{k,j}\}$ to which the attribute att_k corresponds, within $E_{k,j}$, as the exponent of θ inside the local randomness part. However, each local randomness part within an element $E_{k,j}$ in which the attribute appears is blinded by a local, uniformly random chosen element, known only to the owner.

Tokens include f_k, θ^{l_k}, which cannot provide any information about att_k. Tokens also include the label and the value of att_k, each encrypted using a different keyed PRF (F and F_p). The keys of both F and F_p are shared between the owner and org_j, and are unknown to the proxy. Auxiliary information pieces are also encrypted using the same keyed PRFs, with the same key used for encryption of the attribute to be translated by the proxy. Hence, if F and F_p are secure, att_k remains hidden from the proxy. As discussed in Subsection 2.6.2, the PRF can be replaced with other transformations that better suit the translation logic of each organization, *e.g.*, order-preserving transformations. Often such transformations also use PRFs to some extent. In this case, because both the original attribute and the auxiliary information will be encrypted using the same transformation, using a key that is unknown to the proxy, the original attribute will remain hidden from the proxy as well.

Lastly, we would like to note that though the translation is done obliviously and the proxy does not learn the original attribute, it does leak some information about the original attribute, as well as the auxiliary information, to the proxy. For instance, the proxy is able to know, because of the deterministic encryption, which attributes in S_j are used in different ciphertexts, as equality can be determined based on the PRF-encrypted value. However, at least some sort of leakage appears to be inherent, as this is exactly what enables the proxy to perform the functionality required from it in our scheme. Also note, that such leakage is limited to the translator. This is because each attribute component that is meant to undergo translation by a proxy has two encryption layers: In the outer layer,

we use strong encryption, based on traits of our proposed scheme as discussed in Theorem 1 or on traits of Π; all system entities except the translator will be unable to decrypt this layer. Only the translator is able to decrypt the outer layer and access the inner layer, which contains the actual attribute encrypted in a "weaker" encryption that enables it to perform the translation.

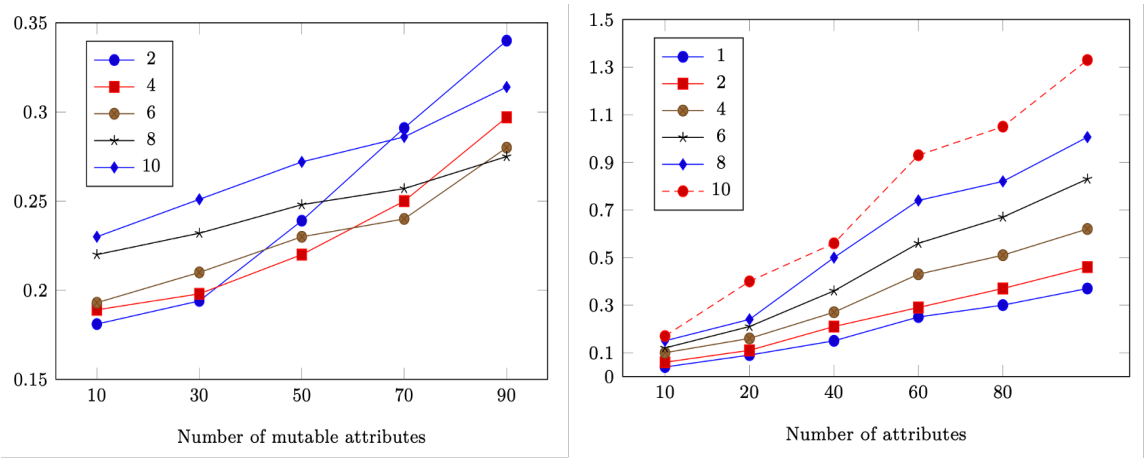
Attribute privacy: We consider the data owner and the data client:

Data client: Given a translated attribute $v \in \mathcal{U}_{client}$, such that $MAP^{-1}(M, v)$ is a mutable attribute, $MAP^{-1}(M, v)$ may appear either within the attribute components $\{C3_{k,j}\}$ to which the attribute v corresponds, inside an element $E_{k,j}$, or within $Tok3_{k,j}, Tok4_{k,j}$. In both the ciphertext and the tokens, $MAP^{-1}(M, v)$ only appears in its encrypted form, using a keyed PRF with a key that is unknown to any member of org_{client}. Furthermore, each local randomness part inside each element $E_{k,j}$ in which $MAP^{-1}(M, v)$ appears, is blinded by a local, uniformly random chosen element, known only to the owner. Lastly, PRF-encrypted terms inside $Tok3_{k,j}, Tok4_{k,j}$ are encrypted using the translator's public key, and can only be decrypted by the translator.

Hence, for every attribute $v \in \mathcal{U}_{client}$ such that $MAP^{-1}(M, v)$ is a mutable attribute, org_{client} does not learn $MAP^{-1}(M, v)$.

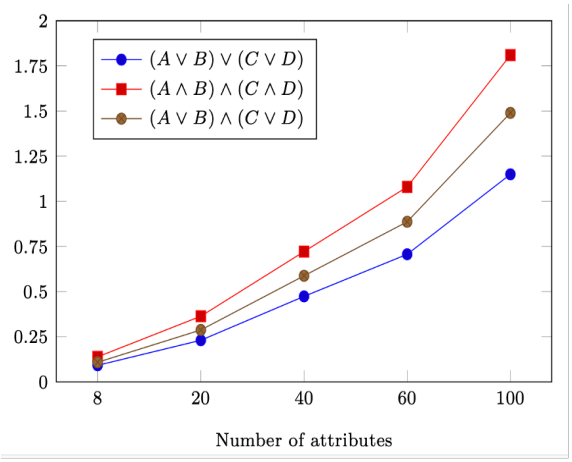
Data owner: For every mutable attribute $s \in S$, the $Encrypt()$ algorithm given in our construction does not require any knowledge about $MAP(M, s)$. Furthermore, for each $s \in S$, the resulting ciphertext, C (including both ciphertext's elements and translation tokens), does not contain $MAP(M, s)$. Lastly, for every mutable attribute $s \in S$, data owners participating in PRShare receive neither terms that include $MAP((M, s)$, nor terms that can provide any information on the value of $MAP(M, s)$.

Hence, for every attribute $s \in \mathcal{U}_{owner}$ such that s is a mutable attribute, org_{owner} does not learn $MAP(M, s)$.



(a) Data query

(b) Encryption



(c) Key generation

Figure 2.1: Typical running times in seconds

2.9 Implementation and evaluation

To assess the feasibility of our framework, we implemented the full version of our OTABE scheme using Charm, a framework developed for rapidly prototyping advanced cryptosystems [4]. Charm was used to develop multiple, prominent existing ABE schemes, including that of Rouselakis and Waters [96]. We instantiated our implementation using a 256-bit Barreto-Naehrig (BN) curve. Note that in our implementation, we translated our scheme to the asymmetric setting, as charm uses formally asymmetric groups. The assumptions and the security proofs can be translated to the asymmetric setting in a generic way.

We consider a setting with three authorities and policies of size ten, where the decryption is always successful, and use oblivious list membership as our translation operation. We present benchmarks for three operations. The first is the overall turnaround time of a data query, *i.e.*, the total time between a user's initiation of a query and her receiving the *plaintext* records that satisfy it. We also provide benchmarks for the encryption algorithm and the key-generation algorithm, despite the fact that encryptions are done offline, and key requests are significantly less frequent than data queries. Note that the hidden-access-policy feature is turned off in our experiments.

Recall that each data query entails the following steps. A query is sent to the CSP. The CSP searches for all of the records that satisfy the query. For each ciphertext returned by the search, the CSP sends its partial ciphertexts to the relevant proxies. Each proxy obliviously translates the partial ciphertext it received. The user aggregates all partial ciphertexts and decrypts the result to obtain the plaintext.

To enable adequate comparison of our OTABE scheme and other ABE schemes, results are given for a single-record data query. Indeed, our running times are similar to other multi-authority ABE schemes, such as [97]. When generalizing our results to the multi-record case, it is important to note that our scheme is highly parallelizable. No TA or proxy

needs to coordinate its computation with any other TA or proxy; thus they can all proceed in parallel. In order to decrypt, a data user must perform a separate computation for each TA, and all of these computations can be done in parallel. Finally, partial ciphertexts that correspond to different attributes can be translated in parallel.

Figure 4.2(e) compares the average time of a data query that contains 100 attributes, for different numbers of mutable attributes and various sizes of ORG_S. The runtimes are relatively small: it takes only 314ms to perform a 90-translation data query when $|ORG_S| = 10$. Although there is an increase in runtime as the number of mutable attributes increases, this increase is significantly more noticeable when ORG_S contains fewer proxies. Figure 4.2(e) also demonstrates an inherent trade-off between the translation and decryption algorithms: A larger number of proxies results in better load balancing of translation operations, but it also results in more expensive decryption.

Figure 2.1(b) shows the average time taken by the encryption algorithm for different numbers of attributes in the ciphertext and various sizes of DEC_S. As expected, encryption time increases as the number of attributes in the ciphertext increases, and as the number of organizations that participate in the decryption increases. Yet, as can be seen, all times are very reasonable compared to other ABE schemes: it only takes 0.46s to encrypt a ciphertext that contains 100 attributes if the number of decrypting entities is 2, and 0.81s if the number of decrypting entities is 6. Bear in mind, that encryption is done once per record, and offline.

Finally, Figure 2.1(c) shows the average time taken by the key generation algorithm for various policies. The times are all under 1.81s. This means that, within less than two seconds from a data user's request for a task-related key, she will receive, from each authority, a key that supports a policy of size 100. Bear in mind that key requests are significantly less frequent than data queries and only occur once per time-limited task.

2.10 The q-DBDH assumption

In Section 2.5, we define a q-type assumption, (q, t, n)-DBDTH, and use it to prove that our OTABE scheme is secure. This assumption is based on another q-type assumption (used in [96], as well as in other existing ABE works) that we referred to as the q-DBDH assumption.

The q-DBDH assumption is parameterized by a security parameter λ, a suitably large prime p, two prime-order bilinear groups $G1$ and $G2$, a bilinear map $e : G1 \rightarrow G2$, and an integer q. It is defined by a game between a challenger and an attacker. The challenger picks a group element g uniformly at random from $G1$ and $q + 3$ exponents $x, y, z, b_1, b_2, \ldots, b_q$ independently and uniformly at random from Z_p. Then it sends $(p, G1, G2, e)$ and the following terms to the attacker:

$$g, g^x, g^y, g^z, g^{(xz)^2}$$

$$\forall l \in [q] : g^{b_l}, g^{xzb_l}, g^{xz/b_l}, g^{x^2zb_l}, g^{y/b_l^2}, g^{y^2/b_l^2}$$

$$\forall l, f \in [q], l \neq f : g^{yb_l/b_f^2}, g^{xyzb_l/b_f^2}, g^{(xz)^2b_l/b_f}, g^{xz(b_l/b_f)}$$

The challenger flips a fair coin b. If $b = 0$, it gives the term $e(g, g)^{xyz}$ to the attacker. Otherwise, it gives the attacker a term R chosen uniformly at random from $G2$. Finally, the attacker outputs its guess b' for the value of b.

We say that ***the q-DBDH assumption holds*** if all PPT attackers have at most a negligible advantage in λ in the above security game, where the advantage is defined as $\Pr[b' = b] - 1/2$.

Chapter 3

Social-network mining to infer voters' intentions

3.1 Introduction

Political choices and voting decisions are considered by many people to be highly sensitive and private information that they are reluctant to reveal. Political campaigns, on the other hand, are investing heavily in voter targeting, focusing intently on social network platforms because of the micro-targeted advertising capabilities they provide. In this work, we explore the following question: Can we predict the voting behavior of Facebook users from their public Facebook profiles?

We present a novel approach for predicting the voting behavior of Facebook users using a Bayesian-network model that combines demographic, behavioral and social features. Although we focus on the 2016 U.S. Presidential election, our approach can be extended to any two-part system.

This work is the first to use a Bayesian network model for political attribute inference. Furthermore, it is the first to not only address, but also offer concrete solutions to the selection bias problem, combining a representative and heterogeneous datasets of both politically active and passive users, a semi-supervised training methodology and a diverse

set of demographic, behavioral and social features. Finally, our model is trained to predict not only to *whom* the user will vote, but also *whether* she will vote at all, a task that has not been performed by any existing work on vote prediction.

3.1.1 Related work

There are several streams of research that investigate the predictive power of social media for political purposes.

One stream of research concentrates on predicting the political orientation of an individual from various components of her social network profile: Tweets' content [95]; retweet graph [29]; following behavior [14]; degree of tweets and retweets [22, 114]; and "liking" politically oriented pages [19].

A more general approach, taken by [90, 111, 118, 120] aims at building a generic framework for latent attribute inference of social media users. Those works do not focus on political orientation as a stand-alone trait, but rather use it to demonstrate the functionality of their generic classification system.

Another stream of research focuses on election prediction from social network data. Such works usually take either a volume based approach or a sentiment analysis based approach and use it to predict the outcomes of various election systems in both two-party and multiparty settings [24, 74, 106, 108].

Although closely related to the lines of research discussed above, individual voting behavior differs in several respects. Individual voting behavior prediction aims at predicting an individual choice (unlike election prediction, that aims at capturing an aggregated measure) for a specific event. This event has a well-defined end date and a well-defined set of choices (labels) that represent the possible voting choices the individual has in a given election. In contrast, political orientation reflects a generic and subjective measure that does not have well defined time boundaries; furthermore, the multitude of scales used for

measuring political orientation, combined with the fact that any point on such scale may have different meanings to different individuals results in the lack of a well-defined set of labels.

In contrast to the work on political orientation and election prediction, relatively little work has focused on individual voting behavior prediction: Gayo-Avello [43] tried to infer the votes of Twitter users in the 2008 U.S. elections by applying multiple sentiment analysis methods. Bachhuber *et al.* [13] examined several approaches aimed at finding distinct clusters of Twitter users based on linguistic properties of their tweets, creating language profiles for supporter groups in the 2016 US elections. Kristensen *et al.* [65] used Facebook to predict voter intentions in the 2015 Danish election, and examined users' political like history to predict which party will they vote for.

3.1.2 Shortcomings of prior research

A significant shortcoming of prior research is the use of biased datasets. Those datasets are composed of politically active social-network users, a minority which does not represent the ordinary user population. Classifiers trained on such datasets will thus experience limited predictive accuracy when applied to ordinary users [28], who are less politically engaged yet constitute the majority of social media users [85].

We recognize three potential sources of bias:

Platform: Twitter is the social network most commonly used for politically oriented data-mining research. However, Twitter is one of the least representative social networks [3, 80]: First, the usage statistics among American adults are quite low. Second, Twitter users are not demographically representative of the population; the resulting demographic bias is often ignored in research concerning political orientation inference and voting behavior prediction. Third, Twitter is considered the most "political" social network, attracting a user population with unusually high political awareness. Twitter users' datasets

are therefore highly likely to be politically and demographically unrepresentative of the general population.

Features: The vast majority of prior work relies solely on the analysis of user-generated content or politically oriented activities in social networks. This approach introduces substantial selection bias because the only social network users who appear in such datasets are those who engage in politically overt activities, and only a minority of all users do so [28], [65]. In other words, those approaches yield high-accuracy results, but those results are limited to a small fraction of social media users. Indeed, less than 27% of the users in our dataset performed a public activity related to Trump or Clinton before the 2016 election date; that is, more than 73% of the users for whom solely relying on user-generated content would yield results which are essentially no better than those we would have received using a random classifier.

Labels: Previous works have used several methodologies for extracting labels from social-network accounts, relying on online behaviors such as explicitly stating political orientation online [43, 90, 118]; including politically related content in tweets [13, 29, 114]; supporting partisan causes [95] or following candidates [110]. Subsequently, only users for whom such label exists are included in the datasets, leading to the creation of biased datasets, composed entirely of individuals who voluntarily disclose their political preference online. This methodology introduces self-selection bias into the final results, as users who choose to disclose their political affiliations constitute a minority of social media users[92].

3.1.3 Our contribution

The goal of the models presented in this work is the following: given a Facebook user, predict the individual voting behavior of that user based on the public portions of the user's Facebook profile. Our main contributions are the following:

Addressing sources of potential bias: we address platform-based bias by using Facebook as our social network platform. Facebook is known to be much more representative of the general population than Twitter and supports a richer profile representation; we address features-based bias by going beyond active content analysis and combining multiple types of information about the user: static (demographic attributes of the user), dynamic (activities performed by the user and their frequency patterns) and social (information we can learn about the user from her social network links). Finally, we address label-based bias by applying a semi-supervised approach that uses a combination of labeled data, where labels are obtained from surveys, and unlabeled data, composed of users who didn't participate in the survey though were offered to.

Incorporating non-voters: Existing works implicitly assume that all users indeed vote. That is, the only users that are included in the datasets are users who voted for one of the candidates in a given election. This assumption is not only an unrealistic oversimplification of election systems but also prevents us from identifying important subpopulations, whose specific vote is inconsistent with their general partisanship such as strong partisans who decide not to vote in a given election. Our key assumption is that a voting decision is influenced by two types of components: a "static" component that is determined by the general party identification of the individual, and "dynamic" components that are determined by specific characteristics of the candidates and include both factual policy-related material and subjective perception of the candidate. Each component can be learnt from different elements in a social-network profile, and its influence on the individual's voting behavior is combined with the rest of the components using a Bayesian network model.

Using a novel Bayesian network model: Bayesian-network (BN) classifiers offer important advantages that specifically fit the nature of our problem as well as the nature of our data.

One advantage of BNs is their ability to support the combination of data and prior knowledge about the problem's domain. This allows us to use existing research and statis-

tics for encoding some of the model's parameters; these include interactions between demographic attributes, and interactions between demographic attributes and party identification. Another advantage of BNs is that they handle missing data very well within both training and evidence data. This is particularly important when dealing with social network datasets, which are often incomplete. Indeed, datasets used in this work contain a large number of missing values, which correspond to attributes that users have chosen not to include in their public Facebook profile. Due to our use of BN, missing values in the evidence data need not be imputed but rather can be fed directly to the model. Furthermore, the probabilistic representation combines naturally with the Expectation-Maximization (EM) algorithm [31], enabling our training data to include both missing values and latent variables. It is this trait of BNs that facilitates the use of both labeled and unlabeled training data.

3.2 Methodology and datasets

We designed and distributed a comprehensive survey[1] which adopted purposeful sampling of eligible voters in the 2016 U.S. elections that are also Facebook users. We used the Qualtrics survey platform to create and host the survey.

Survey construction: The survey included questions about the user's demographics, her Facebook activity, her political opinions including her party identification and her vote in the 2016 presidential election. All survey data was anonymized after collection. We informed participants that their responses would be used for academic research.

We implemented several methods for identifying and excluding data from participants who answered unreliably. First, we eliminated responses from participants who took the survey more than once. We took a conservative approach and discarded responses that came from the same IP address. Second, to ensure the eligibility of participants, they had

[1]The full survey can be found at https://lihiid.files.wordpress.com/2021/09/survey voting behavior.pdf

to complete a screening questionnaire before taking the survey. The questionnaire was carefully designed, in order to prevent respondents from inferring the qualifications we were looking for and taking the survey without being an eligible participant. We avoided yes/no questions regarding the qualification needed and used multiway questions instead, as yes/no questions tend to insinuate the "correct" answer in order to pass the screening questionnaire. In addition, we disguised the real screening questions among other dummy questions. For example: instead of directly asking: "do you have a Facebook account"? we asked a series of identical multiway questions about news consumption habits from each of the media platforms. For example, the question "how often do you consume news via TV?" had the following five answers: More than 5 times a day; 3 times a day; once daily; Never; I do not have a TV. In the same question that dealt with Facebook, the last answer was replaced by "I do not have a Facebook account". Third, we included control questions to ensure that the respondents were providing reliable data; those were fairly straightforward questions which asked the same question multiple times, in different parts of the survey, and using a slightly different terminology. We excluded participants who failed in one or more of the control questions. Lastly, the survey's name did not include words such as 'politics' or 'social media' so as to not expose the qualifications needed, as well as not to oversample from the more politically engaged population.

Datasets: In this work, we make use of three datasets:

$\mathcal{D}1$ consists of 1638 survey responses collected via Amazon's Mechanical Turk (MTurk). Labels for this dataset are obtained through the survey.

$\mathcal{D}2$ consists of 841 Facebook profiles and corresponding survey responses collected via both Facebook and Qualtrics. Labels for this dataset are obtained through the survey.

$\mathcal{D}3$ consists of 500 unlabeled Facebook profiles, corresponding to individuals who did not participate in our survey. The only information that was collected on this dataset is *public* — details that the users have published in their public Facebook profile. For that reason, $\mathcal{D}3$ is unlabeled and contains a large number of missing features.

Table 3.1: Descriptive statistics for $\mathcal{D}1$ and $\mathcal{D}2$

Attribute	Metric	$\mathcal{D}2$	$\mathcal{D}1$	Census
Gender	Female	48.6%	50.6%	50.8%
	Male	51.4%	49.4%	49.2%
Educational attainment	Post-graduate degree	13%	14.3%	11%
	College degree	35%	31.2%	26.2%
	No college degree	52%	54.5%	62.6%
Marital Status	Married	48.9%	49.5%	48.3%
	Unmarried	51.1%	50.5%	51.7%
Ethnicity	Caucasian	66.6%	71.4%	62.8%
	African American	10.7%	9.77%	12.2%
	Hispanic	15.6%	12.1%	16.9%
	Other	7.13%	6.72%	8.1%
State Of Residence	Solid republican	28.3%	23.6%	24%
	Lean republican	8.56%	10.5%	11.7%
	Competitive	38%	39.2%	40.5%
	Lean democratic	13.1%	14.6%	14.8%
	Solid democratic	12%	11.9%	9%
Age	20-33	38.9%	53.7%	19.7%
	34-49	30.2%	26%	20.5%
	50+	30.9%	20.3%	44%
Party Identification	Republican	33.9%	33.3%	
	Democrat	40%	33.3%	
	Independent	26.2%	33.3%	
Vote	Trump	30.8%	32.1%	
	Clinton	36.7%	37.3%	
	Other	32.5%	30.6%	

The use of $\mathcal{D}1$, $\mathcal{D}2$ and $\mathcal{D}3$ was motivated by our attempt to address three types of selection bias:

Visibility bias: Datasets used in prior research were often created by systematically excluding users who did not publish a certain attribute — one that is crucial to the model's performance, such as politically oriented activity or self-reported label — in their public social network profile. Acknowledging that such approach yield datasets that are not representative of the Facebook user population, our datasets were constructed so that the absence of a feature or a label from a user's public profile does not affect the user's chance of being selected. As is evident from Table 3.2, neither feature was publicly disclosed by more than 46% of the users in $\mathcal{D}2$, and only 4.4% of them published their vote in the 2016 presidential election.

Self-selection bias: $\mathcal{D}1$ and $\mathcal{D}2$ represent a population of individuals who have *selected* to participate in our survey. In order to address this selection bias, we created a third dataset

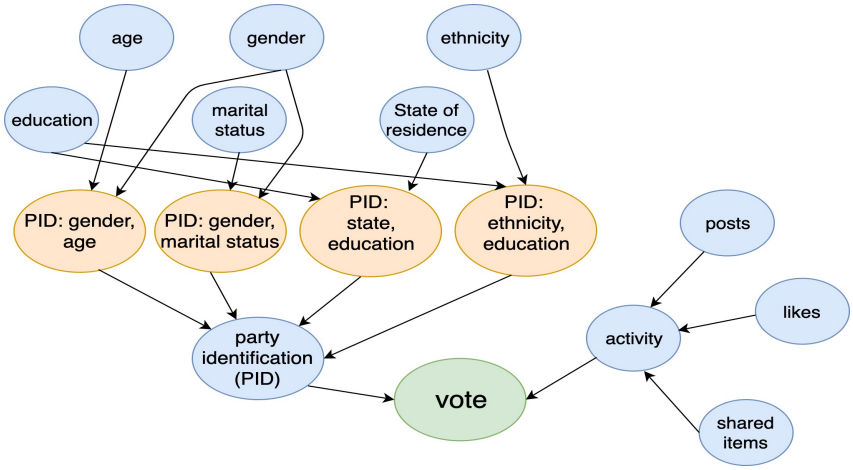


Figure 3.1: The BASIC model

$\mathcal{D}3$ composed of Facebook users who, though were asked to, did not participate in our survey. While we could not use $\mathcal{D}3$ for testing, as it is unlabeled, we did combine it in our training set in order to further generalize our model.

Demographic bias: one advantage of the use of MTurk for data collection is the ability to obtain data from demographically diverse groups; although MTurk's population is not fully representative of the US population, the ease of data collection via MTurk enabled us to reach crowds that are diverse across the primary demographic dimensions used in this work. Table 3.1 presents descriptive statistics of $\mathcal{D}1$ and $\mathcal{D}2$ and compares a subset of their demographics to the 2014 US census. As can be seen, each group is well represented in the datasets. Note that $\mathcal{D}1$ was stratified by Party Identification, resulting in a balanced representation of Republicans, Democrats, and Independents within the training set.

External software: Throughout this work we refer to specific subtasks that were performed using existing, readily available software. Examples include Genderize.io[2] and ethnicolr[3] for gender and race inference based on the user's name; Python NLTK for sentiment analysis tasks; GeNle and pysmile for our BNs' creation, parameter learning, and inference.

[2]https://genderize.io/
[3]https://ethnicolr.readthedocs.io/

3.3 The Bayesian network

The target node in our BN, "vote", represents the voting behavior of an individual user. It may take one of three values: vote for candidate 0 ("c0"); vote for candidate 1 ("c1"); not vote at all or vote for a third party candidate ("Other"). In our setting, "candidate 0" represents a vote for Donald Trump; "candidate 1" represents a vote for Hillary Clinton; "Other" represents non voters or a vote for a third party candidate in the 2016 U.S. elections. The basic idea behind our BN models is to treat the vote node as both a cause and an effect. As such, it is influenced by a set of causes and causes a set of effects: Causes include the party identification (PID) of the user and the voting behavior of the user's network neighborhood (NET). Effects include the user's politically oriented activities on Facebook (ACT). Each of the causes and effects belongs to a subnetwork that includes a different type of observable variables: demographic attributes of the user ("static subnetwork"); types and patterns of Facebook activity performed by the user ("dynamic subnetwork"); and activities performed by the user's network neighborhood ("social subnetwork"). Due to the BN's structure, we are able to elicit different priors with varying levels of confidence to the different subnetworks.

Formally, given a target Facebook user u_t, Let $D = \{D_t^1, D_t^2, \ldots, D_t^n\}$ be n demographic attributes of u_t, $A = \{A_t^1, A_t^2, \ldots, A_t^k\}$ be k attributes describing Facebook activities performed by u_t and $N = \{N_t^1, N_t^2, \ldots, N_t^j\}$ be j attributes describing u_t's network neighborhood. Given our target node, vote, we are interested in the label c that maximizes the following posterior probability:

$$c = \underset{v \in \{c0, c1, Other\}}{\operatorname{argmax}} P(vote = v \mid PID, ACT, NET)$$

Assuming that $\{D_t^1, D_t^2, \ldots, D_t^n\}$, $\{A_t^1, A_t^2, \ldots, A_t^k\}$ and $\{N_t^1, N_t^2, \ldots, N_t^j\}$ are observable attributes, and using a bayesian approach $P(ACT \mid A_t^1, A_t^2, \ldots, A_t^k)$,

$P(NET \mid N_t^1, N_t^2, \ldots, N_t^j)$ are obtained from our datasets and a uniform prior, and $P(PID \mid D_t^1, D_t^2, \ldots, D_t^n)$ is obtained from our datasets and survey-based, or census-based priors.

Using the EM algorithm, we employed a two-stage learning methodology: first, we trained a BN using the available labeled data ($\mathcal{D}1$) and probabilistically labeled the unlabeled data; then, we trained a second BN using both the labeled data and a subset of the unlabeled data ($\mathcal{D}3$) about which the first BN was the most confident. This allowed us to both learn the parameters of the model's latent variables and diversify our training set with a more representative, but unlabeled, data. Inference was done using a Junction tree based algorithm.

Complexity-accuracy trade-off: the models presented in this work were carefully designed so as to create compact models without damaging performance; if a certain edge did not substantially contribute to the model's predictive performance, we did not include it in the network. For example, the income node in our SELF INFER model is, theoretically, influenced by many demographic attributes. However, we have found that the only edges that significantly contribute to the overall accuracy were those connecting income with occupation, race and gender, and hence included only those edges in the network.

BASIC: Two nodes, "party identification" and "activity" are directly linked to the "vote" node. "Party identification" covers the influence of an individual's general party identification on her actual vote. It is considered an unobservable attribute, and thus inferred from the user's demographic attributes. The attributes which are included in the model met two criteria: attributes which are highly indicative of party identification; observable attributes, in the sense that they are either offered by Facebook as an optional field in a profile or can be inferred from other public information, such as name. The following attributes were included in our model: gender, age, race, state of residence, education level, marital status.

We enhance the static subnetwork using prior information about the magnitude of the interactions between each demographic attribute and the PID. Priors are based on multiple

surveys conducted by Pew research center[4] and on statistics provided by the U.S. Census Bureau; priors refer to data that was collected at least six months before the elections and were incorporated into the model's parameters using a Bayesian approach and a Dirichlet prior with varying confidence factors based on the available prior information on a given parameter.

As can be seen in Figure 3.1, the static subnetwork is built in a conceptually "layered" fashion: The first level represents ascribed attributes; the second level represents acquired attributes; the final level contains the PID node. In order to avoid a large CPT for the PID node, we used a parent-divorcing technique, introducing intermediate nodes (colored orange in Figure 3.1, Figure 3.2 and Figure 3.3) that serve as the "accumulators" of the PID given the subset of demographic attributes to which they are linked.

The second node which is directly linked to the target node is the "activity" node, which addresses the more dynamic indicators of the user's voting behavior and is influenced by the public Facebook activities that the user performs.

In BASIC, we included three types of activities: writing a post; sharing an item; and "liking" a page that is positively or negatively associated with one of the candidates. Each of those activities is represented in the model using a separate observable node. The "activity" node aggregates the different combinations of activities into three states that represent the user's overall activity: activity associated with supporting candidate 0, activity associated with supporting candidate 1 and activity that is not associated with supporting either candidate (such as writing a positive post about both candidates).

REVISED: Our second model, the REVISED model, refines the dynamic subnetwork (Figure 3.2) in order to reflect several insights we obtained from reviewing the explanations respondents gave to different questions in the survey:

Positive activity towards a candidate and negative activity towards her opponent are not equivalent: Although positive activities were almost always associated with a

[4]https://www.pewresearch.org/

Table 3.2: Fraction of users ($\mathcal{D}2$) whose public profile contains various attributes

Attribute	Fraction revealed
Age	18.2%
Educational attainment	34.8%
Marital status	36%
Occupation	32.7%
State of residence	45.5%
Wall content (full)	57.7%
Wall content (partial)	72.5%
Friends list	53.6%
Activity (positive or negative) associated with either Clinton or Trump	26.6%
Vote in the 2016 Presidential election	4.4%

vote for the candidate who was the subject of the activity, negative activities towards a candidate were associated either with a vote for the candidate's opponent, or with non voters.

Passive users versus *politically* passive users: The respondents who answered that they did not perform any political activities on Facebook were asked to explain why they didn't. A common explanation was an unwillingness to reveal political opinions on Facebook. It is important to note that those answers differed from the answers of respondents who reported that they rarely use Facebook, use Facebook without performing activities, or are simply not interested in politics. An interesting observation was that most users who said they did not want to expose their political opinions on Facebook identified themselves as either voted for Trump or did not vote at all.

Considering other Facebook activities: Respondents pointed out that, although they have not performed an activity associated with one of the main candidates, they did perform an activity that was associated with a third-party candidate.

Based on our observations, we created the REVISED model: First, the "activity" node was replaced with two nodes: "positive activity" and "negative activity". In addition, the observed "post", "share", and "like" nodes are each replaced by two nodes, one for the positive form of the activity and one for the negative form of the activity. Note, that unlike

the BASIC model in which there was no distinction between a positive activity towards candidate x and a negative activity towards her opponent (as both were mapped to the same "pro candidate x" state), here we distinguish between positive and negative activities, each separately influences the vote node.

Second, we added a new binary node, labeled "Other activity", that represents various online activities that may indicate a vote for a third-party candidate. In the REVISED model, it is set to one only if there is a positive activity associated with a third party candidate in the user's profile.

Third, we added a binary node, "activity level", that represents the general activity level of the user on Facebook. We wanted to distinguish those users who are not active on Facebook from those who are not *politically* active on Facebook; our second observation suggested that while the first group is not particularly important for the inference process, the second group is strongly associated with users who voted for Trump (more generally, with the candidate who is considered less "socially acceptable"). We added four questions to the survey: one asked for the subjective estimation, on a ten-point scale, of the respondent's activity volume on Facebook. The remaining three asked about the number of posts, shares, and likes that the user has recently performed. We aggregated those four measurements into one node, "activity level", by discretizing each variable with a median split and setting the "activity-level" value to the four variables' logical AND.

3.4 Handling incomplete data

The models discussed thus far has considered a simplified setting, in which no evidence data is missing. While social-networking services can potentially obtain such complete information about their users (by accessing profile items in all visibility levels or collecting demographic data when the user creates an account), third-party services can not, because they are limited to the public portion of social-network profiles that often lacks some of

the attributes that comprise our models' features. Thus, it is essential to understand how well does the model perform in the presence of missing data.

MISSING: In order to understand to what extent our model's performance is influenced by missing data, we tested the REVISED model on another dataset $\mathcal{D}'2$ which is identical to $\mathcal{D}2$ except for the following case: if the value of an attribute was not publicly shared in a user's Facebook profile, we deleted the attribute's value from the user's record in $\mathcal{D}'2$ and treated it as "missing evidence". In other words, $\mathcal{D}'2$ includes only those attributes that the users in $\mathcal{D}2$ have chosen to publish under a "visible to everyone" setting. For attributes that do not have their own field in a Facebook profile, such as race, we used external software and inferred them from other public information such as name; if the software misclassified the attribute (compared to the real label that was obtained from the survey), we deleted it from $\mathcal{D}'2$.

Results for the MISSING model suggested a significant decrease in the overall accuracy, resulting in the need to design models that are specifically suited for missing evidence. The following models demonstrate several strategies that were used to handle the challenge of incomplete datasets.

SELF INFER: This model enriches REVISED by exploring various correlations between the input variables for the purpose of reducing the influence of missing data on the overall results.

Interactions among demographic attributes: Instead of treating demographic attributes uniformly, we partition them into two groups: ascribed attributes and acquired attributes. We can then make use of the possible dependency relations between the two groups. This idea is incorporated in the model by creating new edges between the two "acquired" attributes, education level and marital status, and the "ascribed" attributes which are known to be highly indicative of them: race, gender for the education level, and age, gender for the marital status. An important advantage of this approach is the ability to combine priors in the CPD of the demographic child nodes, which represent the known

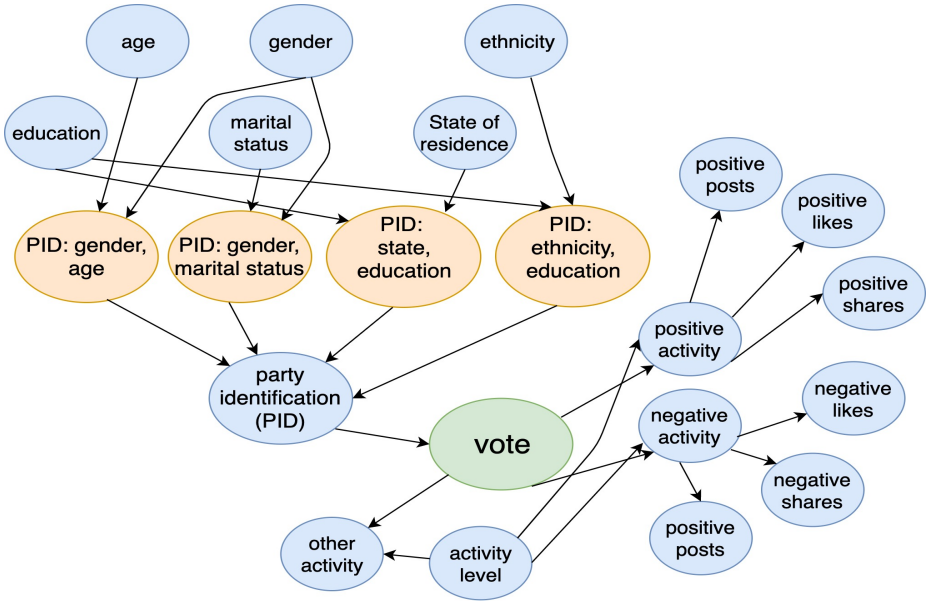


Figure 3.2: The REVISED model

"influence" of the node's parents on their descendant; the priors for this purpose were taken from census statistics and were incorporated into the model's parameters using a Bayesian approach with a Dirichlet prior.

Hidden demographic attributes: Demographic attributes that were included in the former models were both influential on the PID and observable, *i.e.*, can be directly extracted from a Facebook profile. A question that may arise is whether both of those traits must exist in a single attribute. The key idea is that while some observable attributes are not highly influential on the PID, they are influential on other unobservable attributes which do have a high influence on the PID. Using the chain of observable attribute $\rightarrow$ unobservable attribute $\rightarrow$ PID we can combine such unobservable nodes in our model as well.

We consider one such pair: occupation and personal income. While income is not a "potentially observable" attribute on Facebook, it is highly indicative of the individual's PID. Occupation, on the other hand, is not considered very indicative of PID but can be considered as an observable attribute. In the model shown in Figure 3.2, we incorporate

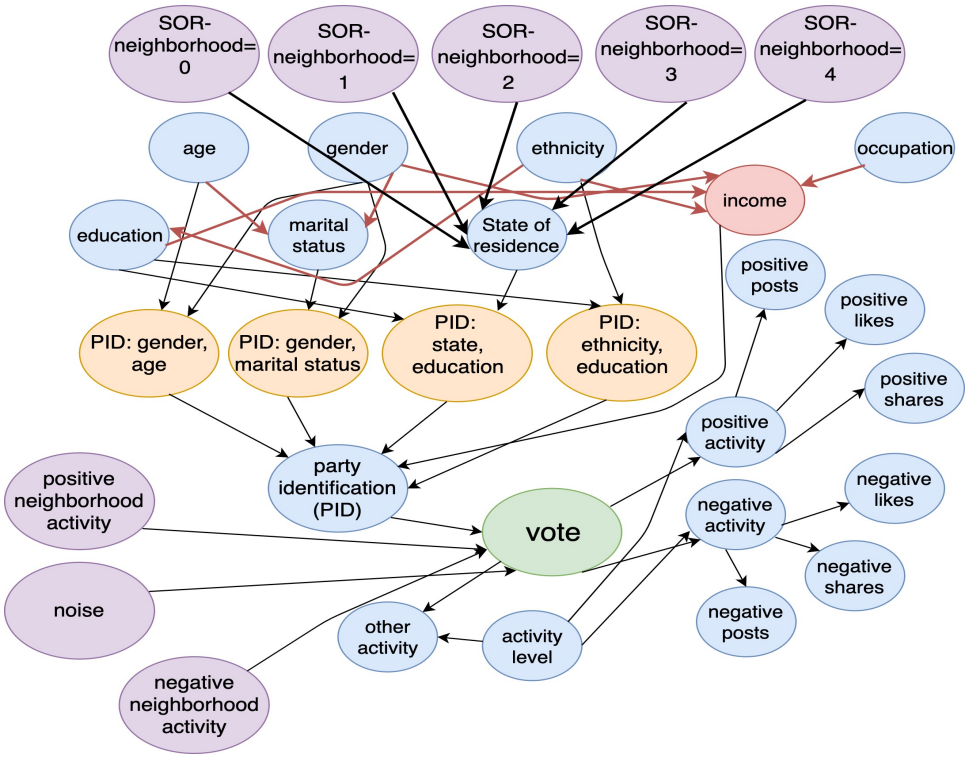


Figure 3.3: The FULL model

this idea, introducing two new nodes: an observed occupation node and a hidden income node. The latter is fed by three observed nodes: occupation, race and gender, and feeding the PID node. The occupation node's states represents the twenty main occupation categories according to the standard SOC (Standard Occupational Classification) system, to which we added two additional states: "student" and "retired".

3.5 Combining link information: The FULL model

Homophily is the tendency of individuals to link to others who are similar to them. In a social network setting, this principle implies that the network neighborhood of an individual may reveal a significant amount of information about the individual. In the FULL model, we adopt a fine-grained interpretation of the homophily principle in order to both enrich the static subnetwork and create the social subnetwork.

A common BN representation of social network ties [53, 61] relies on a user-based granularity where each node corresponds to a user within the target user's (u_t) neighborhood; in such representation, dependency relations (and, consequently, similarity relations) can only be established between *users*. Thus, such representation implicitly assumes that if two users share a tie, all their traits are similar. As demonstrated by [79], this assumption is false: It is known that some attributes experience a more homophilous nature than others. For some, similarity does not induce homophily at all.

In order to facilitate a more realistic representation of social network ties, our FULL model relies on an attribute-based granularity, thus reflecting a more "fine-grained" homophily which may vary across different attributes. To achieve that, we decompose a user into the different attributes that constitute her social network profile by associating nodes with pairs of <user, attribute>, and dependency relations with links between *attributes* of users. Using this fine-grained representation, dependency relations can be established between a single attribute of u_t and the same attribute of each of her neighbors only if the attribute is known to be highly homophilous.

The above idea is implemented in the model using a small, fixed number of "aggregator" nodes (colored purple in Figure 3.3), thus avoiding the overhead of dynamically allocating a separate node for each neighbor of u_t. For a given attribute, those nodes aggregate the magnitude of each of the attribute's states within u_t's neighborhood: let a_t be an attribute of u_t that we want to infer from u_t's neighbors, and has $q(a_t)$ states, $a_t^1..a_t^{q(a_t)}$. In the BN, we allocate $q(a_t)$ aggregator nodes, parents of a_t, where the *i*th node represent the magnitude of the subset of u_t's neighbors for whom the value of a_t equals the state a_t^i.

3.5.1 Enriching the static subnetwork

We use the homophily principle to infer missing demographic traits of u_t from the Facebook profile of u_t's neighbors.

Table 3.3: Comparison of the overall classification accuracy using different training configurations.

TEST →	$\mathcal{D}2$			$\mathcal{D}'2$		
TRAIN ↓	BASELINE	BASIC	REVISED	MISSING	SELF INFER	FULL
10-fold CV	.382	.68	.745	.661	.718	.825
$\mathcal{D}1$	.382	.726	.76	.67	.712	—
$\mathcal{D}1$+$\mathcal{D}3$	.382	.738	.782	.695	.755	—

FULL: In the FULL model (Figure 3.3), we focused on a particular subset of ties within u_t's network neighborhood. The subset is composed of neighbors who are "close friends" of u_t, where a close friend is defined as a friend who has reacted to a recent post on u_t's wall. Using this approach, we only consider ties that are both intense and recent, as the stronger the tie connecting two individuals, the more similar they are[48, 79]. Furthermore, such activity based metric is more predictive of tie strength than metrics based on network structure or social distance, incorporating both intimacy and intensity factors[44, 48]. The major advantage of this approach is its simplicity and practicality, as it does not require access to the full list of friends, but only to u_t's wall. Furthermore, such ties can be identified and accessed directly from u_t's profile.

We examined two attributes that are known to be highly homophilous: state of residence (SOR) and age. We elaborate on the CPD design of the state of residence node (V_{SOR}). The CPD of the age node was built in a similar process.

The influence of u_t's neighbors' SOR on u_t's own SOR is modeled using five aggregator nodes that represent the five states that the SOR variable can take in the BN (see Table 3.1), $\{SOR^i\}_{1\leq i\leq 5}$. The *j*th aggregator node quantifies the portion of u_t's close friends for whom V_{SOR}=SOR^j. Each of the aggregator nodes was discretized to reflect the rank of the state it represents among V_{SOR}'s states and was linked to V_{SOR} as its parent using a memory-efficient multinomial logistic CPD.

3.5.2 Creating the social subnetwork

The influence of the target user's neighborhood on her voting behavior has been considered thus far only indirectly, through the static subnetwork. A natural extension is to treat the vote node as a stand-alone trait; then, the homophily principle can be used directly to infer the voting behavior of u_t from the voting behavior of her neighborhood. However, because the voting behavior cannot be directly extracted from a Facebook profile field we cannot use exact counts as we did for demographic attributes. Instead, we use an approximation of the overall voting behavior of u_t's neighborhood, obtained via the following procedure: First, we mark the subset of u_t's neighborhood whose voting behavior can be estimated with high confidence; these are users who have performed a social network activity that is associated with one of the candidates. Second, we determine the influence of u_t's neighborhood on her own voting behavior using *only* this marked subset, as well as the magnitude of this subset relative to u_t's neighborhood.

FULL: A social subnetwork, built under the same principles used to enhance the static subnetwork, is added to the FULL model. The subnetwork contains four aggregator nodes, representing the portion of u_t's close friends who performed either a positive or negative Facebook activity associated with candidate 0 or 1, and a "noise" node, representing the portion of u_t's close friends who did not perform any public Facebook activity associated with a candidate. In order to avoid continuous-valued nodes, the portions represented by each node were discretized into five intervals using equal-frequency discretization. All four nodes were linked into one of two intermediate nodes ('Positive neighborhood activity' and 'Negative neighborhood activity'), while the 'noise' node was linked to both. Those intermediate nodes aggregate the influence of the user's neighborhood on her voting behavior and feed directly to the vote node.

3.6 Experimental results

The BASELINE, BASIC and REVISED models were evaluated on $\mathcal{D}2$, which includes attributes in all visibility levels. The REVISED model was also evaluated on $\mathcal{D}'2$, which only includes attributes from $\mathcal{D}2$ published under a "visible to everyone" setting. SELF INFER and FULL, designed to handle missing evidence, were evaluated on $\mathcal{D}'2$.

Table 3.4 provides a detailed summary of results. We report overall accuracy, Precision, Recall, and AUC for each class. For FULL, we report the 10-fold cross-validation results on $\mathcal{D}'2$, because $\mathcal{D}1$ does not include social features. However, in order to take advantage of the information in $\mathcal{D}1$, we use the CPDs obtained for the static and dynamic subnetwork in previous models as high-confidence priors for FULL.

Inspired by [90], we employ a simple baseline system (BASELINE) that classifies all the users explicitly mentioning their vote in the 2016 election within one of their public posts. All other users are considered misses for the given class.

Table 3.3 compares the overall accuracy of each model using different training configurations. The cross-validation results allow adequate comparison between the results of FULL in Table 3.4 and the rest of the results. As can be seen, augmenting the labeled dataset $\mathcal{D}1$ with unlabeled data $\mathcal{D}3$ indeed improves the classification accuracy. Furthermore, this increase becomes more significant as the number of missing values within the test set increases.

BASELINE's low overall accuracy probably results from the fact that very few users have publicly disclosed their voting intention. Results for the REVISED model demonstrate an improvement over the BASIC model, especially for the "other" class. This improvement suggests that the finer-grained modeling of the dynamic subnetwork allows the classifier to capture additional information which uniquely characterizes the "other" class, thus better separating it from the other two classes. Results also show that the MISSING

model underperforms the REVISED model. However, the differences are not as sharp as we expected considering the large number of missing values.

FULL was the best model, as evidenced by its high accuracy score (82.5%). Its use of social features boosts both precision and recall, but the SELF INFER model, which only includes demographic and behavioral features, achieves a decent score (75.5%) as well. This demonstrates the fact that using carefully designed models combined with complementary sources of information about the user yields solid predictions even when both training and testing datasets are incomplete.

Most determinant features: We performed a sensitivity analysis to assess the impact of the various models' features on our target variable, vote. We elaborate here on the two most determinant features for each state of the vote node. The most influential feature for vote=Trump was race=African American, which decreased the posterior probability of vote=Trump by 19%, followed by positive_neighborhood_activity_Trump=very high, which increased the posterior probability of vote=Trump by 13%. For vote=Clinton, the two most influential features were positive_post=Clinton and education_level=post graduate degree; they increased the posterior probability of vote=Clinton by 22% and 15%, respectively. The two most influential features on vote=Other were negative_activity=both candidates and noise=very high. They increased the posterior probability of vote=Other by 15% and 9%, respectively.

Comparison with Existing Results: Unlike political orientation, prediction of individual voting behavior has not been intensively studied. Exceptions include [13], [43], and [65], achieving an overall accuracy of 63%, 78.8%, 70.8% respectively. However, datasets used in all three works were artificially limited to include only politically active users: users who performed politically oriented tweets [13], hashtags [43] or likes [65]. The last is highlighted because the accuracy of political-orientation classifiers is heavily dependent on users' political-engagement level. For example, [28] showed that methods for inferring the political orientation of social-network users previously claimed to achieve greater

Table 3.4: Detailed results for the Bayesian network models presented in this chapter

Model	Class				Acc.
		Prec.	Rec.	AUC	
BASIC	Trump	.77	.76	.91	.738
	Clinton	.72	.89	.91	
	Other	.72	.53	.78	
REVISED	Trump	.73	.85	.94	.782
	Clinton	.79	.83	.92	
	Other	.83	.64	.84	
MISSING	Trump	.63	.84	.88	.695
	Clinton	.75	.7	.85	
	Other	.72	.53	.78	
SELF INFER	Trump	.69	.87	.91	.755
	Clinton	.81	.73	.88	
	Other	.76	.65	.82	
FULL	Trump	.78	.88	.94	**.825**
	Clinton	.83	.86	.93	
	Other	.85	.72	.91	

than 90% accuracy on politically active users, actually achieve barely 65% accuracy when applied to "politically modest" users.

The prior work most relevant to ours is that of Kristensen *et al.* [65], which considers five models. We could not use all five as baselines for our work, because some use features that can not be directly extracted from a Facebook profile (such as the user's opinion towards politically dividing issues, information which [65] obtains from surveys). In contrast, our main principle in this work is to only use features that can be extracted from a Facebook profile directly, without requiring the user to participate actively in the data-collection process (our survey is used for validation purposes only). We were able to apply model 2, which uses the user's single most recent like that is associated with a party or politician's page; model 3, which uses the number of such likes over the past two years; and their final model, which combines the features from model 3 with all the features from the survey that served as their baseline model. We selected from their final model only those features that can be extracted from a Facebook profile directly: gender, age, geog-

raphy and education. All models were implemented using Python's scikit-learn library.

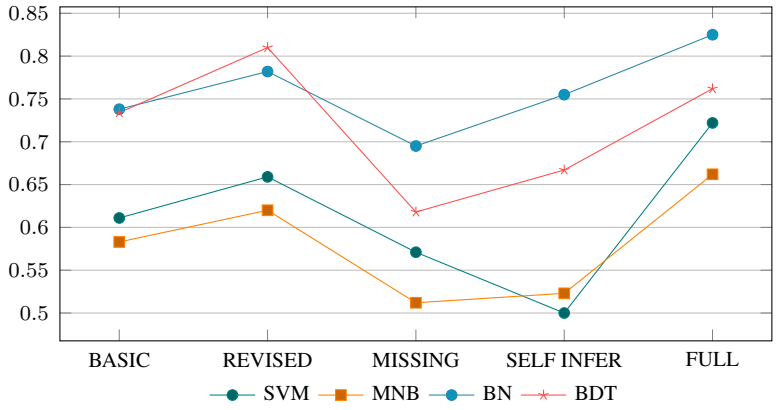


Figure 3.4: Overall classification accuracy among different classifiers

Applying model 2 to our dataset (Using $\mathcal{D}1$ and $\mathcal{D}2$ as training and test sets) achieved an overall accuracy of 38.6%, a low score both compared to the result achieved in [65] (43.9%) and to our BN models' results. Applying model 3 to our dataset achieved an overall accuracy of 40.4%. It is interesting to see that there is not much improvement in the accuracy compared to model 2, while the difference between the two models in [65] is more significant, with model 3 achieving 60.9%. Finally, applying their final model to our dataset achieved an overall accuracy of 60%; we can see a clear increase in the accuracy compared to models 2 and 3; however, it still underperforms [65]'s final results (70.8%). This results from the fact that [65] both artificially limit their datasets to users who performed a "political like" (only 19% from our dataset) and use features that can not be directly extracted from a user's profile such as opinion about politically dividing issues. As evidenced by Table 3.4, [65]'s final model significantly underperforms all our BN models when applied to our datasets. This highlights the added value that BNs have for this specific task compared to a regression model, and the importance of the other features that are not included in [65]'s final model but are included in our BN models.

Apart from [65], no other existing work could serve as a proper baseline for our work,

primarily because their objective is different from ours; Unlike works such as [74, 106], we are *not* trying to forecast the general outcome of an election. Unlike works such as [28, 29, 90, 118], we are *not* trying to predict a general political orientation of an individual. Therefore, a direct comparison of our results and theirs is meaningless. For election forecasting, such comparison is impossible since the evaluation metrics are different: while we use an overall accuracy score, election prediction papers use MSE and compare their forecasting results to national surveys. Furthermore, the vast majority of political orientation inference papers have dealt with Twitter; thus, their models heavily rely on Twitter-specific features, making it impossible to apply those models directly to our datasets. However, in order to gain further insights about how our BN compares to other models used in previous work, we chose two other classifiers commonly used in previous work on political orientation inference: Support Vector Machines (SVM) and Boosted Decision Trees (BDT), as well as simplified Multinomial Naive Bayes (MNB) classifier, and tested the performance of each classifier when applied to our datasets and fed with the features of each of our BN models.

Hyperparameters (cost and γ for the RBF SVM; number of trees and learning rate for BDT) were chosen using a randomized grid search with 5-fold cross-validation. All three models were implemented using Python's scikit-learn library. As seen in Figure 3.4, both MNB and SVM considerably underperform the BN on all five configurations. On the other hand, the BDT classifier performs on par with the BN; it slightly outperforms the BN on the REVISED model but underperforms the BN on the MISSING, SELF-INFER and FULL models; that is, when the test set contains missing evidence.

Chapter 4

Inferring behavioral intentions of social-network users

4.1 Introduction

Knowledge of SN users' decisions and intentions has immense potential to improve the design of recommendation systems, ad-targeting mechanisms, public-health campaigns, and other social and commercial endeavors. At the same time, such knowledge can have a detrimental effect on users' privacy. In this work, we are interested in inferring intentions of SN users using public data extracted from their SN accounts.

Problem description: Let u be an SN user and S_u be the set of SNs on which u has accounts. We use $\xi_{(u,s)}$ to denote user u's account on network s. Each account has a private portion $\xi^{pr}_{(u,s)}$ and a public portion $\xi^{pu}_{(u,s)}$. The private portion contains data that only u's ties and the SN provider can see, while the public portion contains data that can be seen by everyone. In addition to data that u publishes, $\xi^{pu}_{(u,s)}$ contains metadata information about $\xi_{(u,s)}$ such as the mere existence of $\xi_{(u,s)}$ and the visibility levels of different attributes in $\xi_{(u,s)}$. The goal of this work is to infer an SN user u's *offline* intentions using only the public portions, $\{\xi^{pu}_{(u,s)}\}_{s \in S_u}$, of her *online* SN accounts. We focus on present or near-future behavioral intentions, *i.e.*, on decisions to perform certain actions within short periods of

time after the decisions are made. Examples include weight-loss intentions, vaccination intentions and purchase intentions.

As intention inference is a complex task, combining external, internal and temporal factors, we tackle it in two stages: First, we propose a novel methodology for inferring behavioral attributes such as decisions and intentions. We go beyond the inference of single attributes and design a modular Bayesian-Network model that, using public SN data, aims at inferring different types of intentions in different domains. We then build on this general-purpose model and use it to create intention-specific Dynamic Bayesian Networks. Because of their temporal nature, DBNs can capture the evolving nature of the decision-making process. The work makes the following contributions:

Intention inference: Intention inference differs from the inference of other attributes in multiple respects. Unlike persisting attributes such as personality traits or attitudes, intentions are dynamic in the sense that their values with respect to a given user may constantly change. Unlike attributes such as mental state or demographic attributes, intentions are "self-controlled" attributes, in the sense that the user is able to control the values of those attributes. The intentions that we explore in this work can be seen as "behavioral" attributes, as they aim at predicting a time-varying *behavior* rather than a time-invariant tendency, opinion or preference. Intentions are therefore significantly harder to infer than other extensively researched attributes.

A new methodology for intention inference: We present a modular model that draws on behavioral-psychology literature and can be used to infer different different types of intentions in different domains. Our Bayesian-Network-based models are built using a sophisticated, hybrid feature-selection method and can handle common challenges in SN research such as incomplete datasets and bidirectional influence between features and the target variable.

A novel DBN representation of SN users: We offer a new conceptual model of SN users that is particularly suited for inference of dynamic attributes. Each user u is modeled

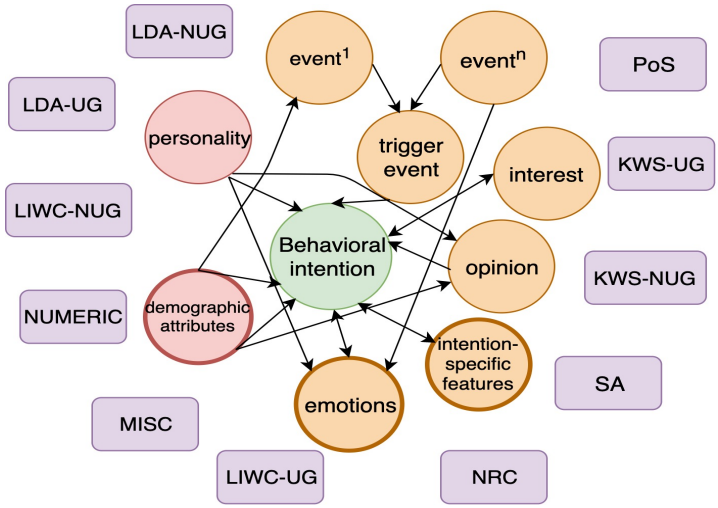


Figure 4.1: Our general intention-inference model.

using a set of DBNs, $\{D_u^d\}$. Each DBN $D_u^{d=k}$ aims at inferring or predicting a different dynamic attribute of u, k. Users' SN profiles are sampled at regular intervals; the resulting data is fed into each attribute-specific DBN D_u^k and is used, along with data sampled in prior time slots, for the inference of k's current value. We demonstrate our approach when applied to the inference of various intentions using temporal SN data collected in multiple waves. This work is the first to take a DBN-based approach to the SN-attribute-inference problem.

Finally, We evaluate our new DBN-based SN-users representation using five intentions: vaccination intentions, weight-loss intentions, purchase intentions, borrowing intentions and job-searching intentions, achieving varying, yet promising results despite the use of highly imbalanced, incomplete datasets.

4.2 Methodology

4.2.1 Key ideas

We start by clarifying the terminology used in this chapter. The terms "decision" and "intention" are sometimes mistakenly used to describe a past *behavior*, rather than a cognitive state. This work does not aim at predicting past behaviors, but rather the decision to perform those behaviors. Identifying the specific cognitive stage that lies between planning an action and performing that action is of utmost importance for many applications, such as recommendation systems and targeted advertising. Therefore, in this work we will use the term "behavioral intention" to describe a decision to perform a present or near-future behavior.

"Intentions are people's decisions to perform particular actions" [103]. In this work, we aim at understanding to what extent we can infer behavioral intentions of SN users. In order to do that, we build a Bayesian-Network model that is based on intentions' most influential factors as shown in behavioral psychology literature [39, 52, 103]. We split those factors into two groups: static factors, such as personality, demographic attributes and self-efficacy, and dynamic factors such as emotions, situational factors, interest and opinions. While those factors are known to be excellent determinants of behavioral intentions, the values of some of them (personality, for instance) can not be directly obtained from the user's SN profile ("latent variables"). Therefore, we enrich the model with various observed or partially observed network features which may assist in both inferring the target intention and inferring each intention's latent determinants.

Though different intentions are assumed to be influenced by the same high-level factors, their associated BNs still differ in their qualitative, quantitative and temporal specifications. To reflect those differences, we build on our general intention-inference BN and create, for each behavioral intention, an intention-specific DBN. This is achieved using a

multistage process: first, intention-specific predictors are obtained from relevant literature and added to the set of general features as described above. Second, the final feature set of each intention is determined using priors, feature-selection methods, or both. Third, the set of intention-specific features is mapped into network nodes; this includes aggregation, state-elicitation and discretization strategy. Fourth, the structure of each intention-specific DBN is specified using priors, structure-learning methods, or both. Lastly, The DBN's parameters are quantified using a combination of prior information and data.

In order to simplify the models' description, they are specified in a gradual manner: in Sections 4.3 and 4.4 we show how to implement the above methodology using static BNs and in Section 4.5 we elaborate on the process of extending the resulting BNs and creating intention-specific DBNs. In Section 4.6, we present our models' results when used for the inference of various intentions: weight-loss intentions (WI), vaccination intentions (VI), travel-purchase intentions (PI), borrowing intentions (BI) and job-searching intentions (JI).

4.2.2 Data collection

We designed and distributed a comprehensive survey[1] created and hosted using Qualtrics survey platform. The first part of our survey contained questions about the participants' personal attributes, as discussed in Section 4.3. The second part contained the following statements, which users were asked to rank (as well as dummy statements about unrelated intentions): "I am planning to start a weight-loss regime within the next 1-4 weeks" and "I am currently trying to lose weight" (weight-loss intentions); "I am planning to look for a new job within the next 1-4 weeks" and "I am currently looking for a new job" (job-searching intentions); "I am planning to apply for a loan within the next 1-4 weeks" (borrowing intentions); "I received a flu vaccine this season" — depending on the answer

[1]The full survey can be found at https://lihiid.files.wordpress.com/2021/09/survey behavioral intentions.pdf

to that question the following question was asked for either the upcoming (2020-2021) flu season or the next season (2021-2022): “I am planning to get vaccinated against influenza this upcoming fall-winter/next year” (vaccination intentions); “I am planning to make a travel-related purchase within the next 1-4 weeks” (travel-purchase intentions). Ambiguous words and phrases such as ”travel-related purchase” and ”weight-loss regime” were explained to the participants.

We implemented several methods for identifying and excluding data from participants who answered unreliably, as extensively discussed in Chapter 3. All survey data was anonymized after collection. We informed participants that their responses would be used for academic research.

Table 4.1: Datasets’ statistics

	Size	%VI	%WI	%BI	%PI	%JI
1st-wave dataset	1300	58	38	17	24	19
2nd-wave dataset	803	66	40	13	19	23

Datasets: Survey data was collected in two waves with a three-month lag. Training and test datasets include data obtained from Amazon Mechanical Turk (MTurk), Facebook (profile attributes and activities), Instagram (profile attributes and activities) and Linkedin (only profile attributes). MTurk participants were presented with two options: providing links to their SN profiles (which they reported having in the screening step), or answering a series of questions about their SN profiles; providing SN links was optional and completely voluntary. We took a wide and shallow approach: while data was gathered from multiple SN accounts of each target user, we did not collect “deep” network information. Therefore, our datasets do not include information about the target user’s SN ties, except for information that can be directly extracted from the target user’s SN accounts such as the number of user’s SN ties (see Section 4.3).

As in [57], our datasets include both labeled and unlabeled data; unlabeled data is

specifically important when using multi-wave data, as a considerable number of participants dropped out after the first wave: from 1300 respondents who participated in our first-wave survey, only 803 participated in our second-wave survey (0.617 response rate). In order to both reduce non-response bias and create a bigger training dataset, we chose a subset of our partially-labeled data records which belong to participants who dropped out (missing attributes were treated as missing values) and added it to our training set (see Section 4.5). Our training datasets, $\mathcal{D}_j^1$ (first-wave data for intention j) and $\mathcal{D}_j^2$ (second-wave data for intention j) consist of 780 and 592 labeled and unlabeled data records, respectively. Our test sets, $\mathcal{D}_j^3$ (first wave-data for intention j) and $\mathcal{D}_j^4$ (second-wave data for intention j) consist of 520 and 361 labeled data records, respectively. As mentioned in Section 4.1, in order to simulate a real-world inference task we only consider the public portion of each user's online SN profiles. Therefore, each dataset contains a large number of missing values which corresponds to attributes that the user has not publicly revealed on one of her SN accounts. The size of our datasets is similar to datasets used in prior work in which labels are obtained using surveys or others means that require the users' active participation (unlike self-reported labels). Examples include [60] (523), [29] (956), [65] (1216), [118] (400), [30] (1583), [46] (279), [105] (53).

4.3 Features

A template of our general intention-inference model is shown in Figure 4.1. Second layer-third-layer edges and first-layer-third-layer edges are not presented, because they are intention-specific and must be determined separately for each behavioral intention. Likewise, double-sided edges represent scenarios in which influence may flow in different directions, depending on the inferred intention. Bold nodes are translated into multiple nodes in each intention-specific model.

To reduce noise when combining high dimensional features, and to avoid a large con-

ditional probability table (cpt), we used a layering-divorcing technique. Our basic idea was to create a "layered" network. The first (inner) layer comprises the target intention node that we aim at inferring (colored green in Figure 4.1). The second layer comprises either latent or partially observed nodes, obtained using prior research (colored orange in Figure 4.1). Those nodes represent external and internal factors that are known to have a strong relation to the formation of behavioral intentions. The third layer consists of observable network features (colored purple in Figure 4.1). The purpose that they serve is twofold: both to assist in inferring the behavioral intention, and serve as observed predictors for second layer's latent variables. While some prior research exists on relations between network features and behavioral intentions, it is both relatively sparse, and not general enough. Therefore the decision of which network features to include in each BN was made primarily on the basis of feature-selection results. Some of the network features were found to directly influence the behavioral intention (first layer), and were therefore included in the second layer as well.

4.3.1 Second-layer features

Training-set values for second-layer variables were obtained from our survey. We elaborate on our survey questions for non-trivial attributes. Test-set values were either obtained or inferred using network features as discussed in Subsection 4.3.2.

Personality: This variable represents five broad dimensions of personality obtained from the "Big Five" model of personality dimensions. The big five model distills personality to five traits: neuroticism, extraversion, agreeableness, conscientiousness, and 'openness to experience'. To measure the Big Five personality traits among survey participants we used a short version of the Big Five Inventory based on BFI-10 [94].

Demographic attributes: We considered the following demographic attributes: age, gender, ethnicity, marital status, occupation group, income (latent variable). Only a subset of

those attributes were used in each model.

Situational variables: Events that might trigger a certain behavioral intention. Those events include personal-life transitions, professional-life transitions, external events (such as a holiday or an election), *etc*. Priors were obtained for some events-intentions relations. For instance, life-transitions are shown to have an important impact on weight-loss intentions [23].

Emotions: Our model should represent the fact that different emotions may serve as either the cause of a behavioral intention or as its effect. Therefore, we went beyond the binary emotion-representation (positive-negative) and also considered fine-grained emotions. The most studied model of discrete emotions is the Ekman model [34] which posits the existence of six basic emotions: anger, disgust, fear, joy, sadness and surprise. Since momentary emotion ratings are not particularly indicative of the behavioral intentions explored in this work, survey participants were presented with eight emotion categories (six basic emotions and two positive-negative emotion categories) and were asked to rate their feelings over their past week/month/three months *in general*.

Interest, Opinion: Those variables represent the user's level of interest and opinion regarding topics related to a given behavioral intention.

Intention-specific features: Apart from the set of general features as discussed above, we also consider several intention-specific predictors that are known to be strongly linked to different intentions such as body image (weight-loss intentions) [78], impulsivity (borrowing intentions) [89], employment status ("status" in Figure 4.2), *etc*.

4.3.2 Network features

We considered a diverse set of network features. The values of a given network feature were included in our datasets if and only if this network feature was part of the public portion of one of the user's SN profiles.

Numeric features (NUMERIC): We considered statistics about the user's activity (number of posts, status updates, number of uploaded photos, *etc*), reactions to the user's content (number of tagged photos, for instance) and the user's reactions to other users' content. The latter measure was sparse, as both Facebook and Instagram limit the visibility of such reactions. We also considered basic statistics about the users' network, but we limit ourselves to statistics that are both publicly visible and can be directly extracted from the user's own SN profile/s (number of friends, followers-following ratio, *etc*).

Raw Textual features (TEXT): Textual features were classified as either user-generated (UG) features (including textual content that was written by the user), or non-user-generated (NUG) features (textual features that were not written by the user such as likes (Facebook) or hashtags (Instagram)). We limit ourselves to textual content that is both publicly visible and was either produced by the target user, or can be directly extracted from the user's own SN profile/s.

Miscellaneous features (MISC): Miscellaneous features include features that are neither numeric nor textual, such as the mere existence of various SN accounts, visibility level/s that the user has chosen to apply to her SN accounts, profile attributes from which demographic attributes can be extracted, *etc*. MISC features can be seen as SN accounts' metadata rather than the data itself (NUMERIC, TEXT).

4.3.3 Linguistic features

Keyword-search (KWS-UG, KWS-NUG): For a given intention, or an event, A, we manually identified the most prominent keywords related to A. We then performed a keyword search on our textual features. For some textual features, items that were found to contain a relevant keyword were further processed using a second method, depending on the nature of A and the nature of the textual feature. This resulted in two groups of features, KWS-UG (keyword search applied to user-generated content) and KWC-NUG (keyword

search applied to non-user-generated content).

LIWC (LIWC-UG, LIWC-NUG): LIWC is a text analysis tool that is widely used in psychological studies [107]. Each list of words is associated with a semantic or syntactic category, such as positive emotion, adverbs or tone. LIWC analysis was applied to both UG and NUG textual features. However, with regards to NUG features, we only considered a relatively small subset of LIWC categories that mostly deal with features' topics, such as leisure, work, money or sentiment.

Topic modeling (LDA-UG, LDA-NUG): Topics were extracted using Latent Dirichlet Allocation (LDA). Shorter features (such as likes) and longer features (such as posts) were considered separately using different parameters. As many likes only contain names (*e.g* brand names), we considered a like to be both the like's title as well as the category to which it belongs.

Sentiment analysis, part-of-speech tagging (SA, PoS): These were only applied to KWS-UG (SA and PoS) and KWS-NUG (SA), *i.e.*, textual features that were found to contain at least one relevant keyword. SA was applied to items that were found to contain keywords that relate to the behavioral intention to be inferred, in order to assess the user's opinion on topics related to the behavioral intention. The use of PoS tagging was more implicit, and was applied to items that were found to contain keywords that relate to events.

Emotions (NRC, LIWC): We automatically quantify emotions from our UG textual features using LIWC and NRC. NRC is a publicly available lexicon of words associated with any of the six emotions, as well as general positive and negative sentiment [82]. We assign a predicted emotion to each UG textual feature and then average across all users' features.

Network representation: The features described in this section were first fed into our feature selection algorithm (Section 4.4). Only features that were found to be relevant for the inference of a given intention were included as nodes in the intention's BN. Nodes in our BNs represent either discretized values of numeric features; categorical features

such as demographic attributes; or discretized frequencies of various categories of a certain linguistic feature; such nodes are essentially aggregators as their value represents the prevalence of a specific linguistic category among the entire set of a user's raw textual features. For instance, a node may represent "the aggregated portion of textual features that contain keywords related to fitness". Finally, situational variables are connected to the target node through an intermediate node, "trigger event".

4.4 Feature selection and model selection

Feature selection: We designed a two-level, hybrid feature-selection (FS) method. Due to the high number of correlations between different features, we opted for a BN-based-FS method that utilizes a BN-learning algorithm rather than a univariate filter-based approach. However, performing FS using only a BN may lead to overfitting. In addition, the vast majority of BN-learning algorithms require complete datasets. To combine the best of both worlds, we employed a hybrid FS approach. First, a simple, univariate FS method was applied to a subset of the features on which we didn't have strong prior information. For that purpose, we used a mutual information-based FS method and removed all the features that received a score below a certain threshold. Mutual information measures the dependency between variables (specifically, between each feature and the target variable) and is thus highly suited for BN-based models. The resulting features, as well as the set of latent/high-prior features were the input for the second phase which used the Greedy Thick Thinning learning algorithm (GTT) [27]. This phase aimed at identifying the best features using Markov Blankets.

A Markov Blanket of a variable t is a minimal variable subset conditioned on which all other variables are probabilistically independent of t. The Markov Blanket of a BN node, $MB(t)$ is the set of its parents, $P(t)$; children, $C(t)$; and spouses, $U(t)$ (*i.e.*, parents of common children) as encoded by the graph structure of the BN. As shown in [64], the

Markov Blanket of a given target variable is the theoretically optimal set of variables to predict its value. However, simply considering all the features in the Markov Blanket of the behavioral intention node is unsatisfactory in our case, due to the existence of latent variables. Thus, a better strategy would be to first find an "approximated" Markov Blanket of the target node, $MB'(t)$ which includes the variables in the sets $P(t)$, $C(t)$ and $U(t)$ as discussed above. Then, identify the Markov Blanket of *each* latent variable that is also a member of the target's approximated Markov Blanket and include the features in the union of those blankets in our feature set (in addition, of course, to features in $MB'(t)$). That is, our feature set is:

$$\{MB'(t)\} \cup \{MB(I) \mid I \in S \cap MB'(t)\}$$

Where S represents the set of latent variables in our model. The above strategy would have probably been sufficient if our datasets were complete. However, our datasets contain missing values which had to be imputed before running the SL algorithm. We note that although one of the main strengths of BNs is the ability to perform parameter learning and inference in the presence of missing values, most of the BN-structure-learning algorithms require complete datasets. Hence, for some variables we consider an "extended" notion of a Markov Blanket which also includes certain variables that belong to the variable's second-degree Markov Blanket. Specifically, if a given variable v represents an observed attribute with more than 50% missing values in our test sets ($m()$) and for which we do not have a strong prior ($p()$), we consider a restricted notion of v's second degree Markov Blanket, and add both its direct parents $P(v)$ and its direct children $C(v)$ to our feature set. Let F be that entire variable-set before applying feature selection, and O the set $F \setminus S$; that is, the set of observed variables (many of which are only partially observed). Our final feature set includes the following features:

$$\{MB'(t)\} \cup \{MB(I) \mid I \in S \cap MB'(t)\} \cup$$

$$\{P(I) \mid I \in O \cap MB'(t) \wedge m(I) > 50\% \wedge p(I) = false\} \cup$$

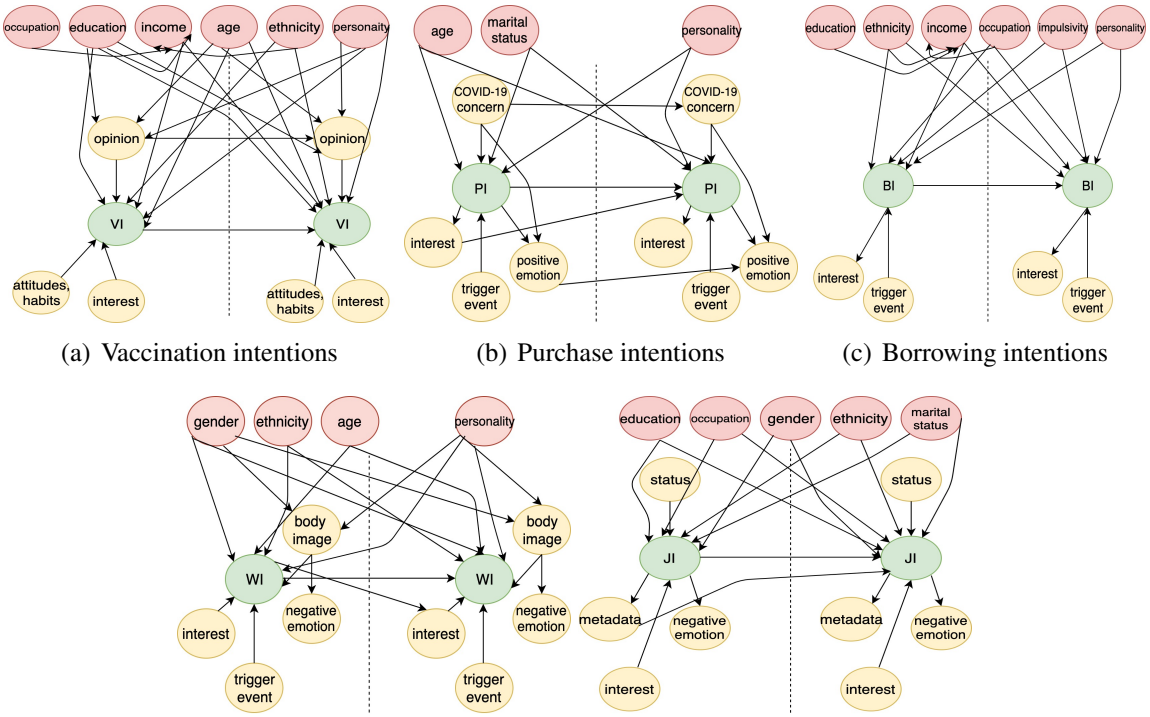
$$\{C(I) \mid \ I \in O \cap MB'(t) \wedge m(I) > 50\% \wedge p(I) = false\}$$

Structure learning (SL): Using the GTT learning algorithm, the approach described above not only yields a feature set but also a BN structure (by considering all the nodes in the feature set and all the edges connecting features in the feature set). A small number of edges were corrected in order to reflect strong prior information.

4.5 Moving from static to dynamic models

So far, we have treated intentions as time-invariant attributes using a static BN. Decision-making, however, is a dynamic process. Thus, a temporal model should be used; one that can capture temporal relations between different variables and our target behavioral intention. Using a DBN, not only can we model temporal relations between different variables and each behavioral intention, but also model both static and temporal relations among different variables.

A DBN is a sequence of BNs. Each BN represents a time slice of the DBN, $i \in T$, corresponding to a particular instance of time. A DBN adds three components to a static BN: temporal variables, temporal edges and temporal evidence. For instance, if a static BN contains the variables $\{X^j\}_{j \in D}$, a DBN may also contain variables that can take different values in different time slices, *e.g.* $\{X_i^j\}_{j \in D, i \in T}$. In order to build a DBN we must specify both its intra-slice structure and corresponding cpts (as in BNs) and its inter-slice cpts. Intra-slice cpts represent dependency relations within a specific time slice. Inter-slice cpts represent temporal dependency relations; that is, relations between variables from different



(a) Vaccination intentions (b) Purchase intentions (c) Borrowing intentions

(d) Weight-loss intentions (e) Job-searching intentions

Figure 4.2: A DBN representation of various intentions.

time slices.

Formally, a DBN is defined as a pair (B_0, B_t) where B_0 defines the prior $P(X_1)$ and B_t is a two-slice temporal BN that defines $P(X_i|X_{i-1})$ by means of a directed acyclic graph:

$$P(X_i|X_{i-1}) = \prod_{j \in D} P(X_i^j \mid PA(X_i^j))$$

If $PA(X_i^j)$ (X_i^j's parents in the network) only contains variables from either the same time slice or an immediate previous time slice, the DBN is considered a first-order DBN.

The temporal nature of DBNs, combined with other unique features such as their ability to work on incomplete datasets make DBNs highly suited for representing various SN components. In this work we focus on SN users, and apply a user-centric approach in which each SN user u is represented as a set of DBNs, $\{D_u^d\}$. Each DBN $D_u^{d=k}$ corresponds to a different dynamic attribute, k; after it is built and trained, D_u^k can be used to

infer the value of k, using data that is sampled at regular intervals from u's SN profiles.

DBNs are extremely versatile; once a DBN D_u^k is built and trained, it can be used to conduct various types of analysis. This includes inferring k's current values, predicting its future values, identifying key determinants of k and performing backward reasoning and retroactive analysis of historical values. A particularly useful feature of DBNs is the ability to assess how a given dynamic attribute changes over time. This ability can be used, for instance, by advertisers in order to assess the impact of a given targeted campaign on the users' intentions.

We demonstrate our approach using the five intentions considered in this work. For each intention, we build its own DBN which builds on the static model developed in previous sections. Note that in this work data is sampled twice, at the same sampling rate for all the intentions. However, if time allows and enough resources are available, better results can be achieved when sampling each SN multiple times using sampling rates that are uniquely tailored to each intention.

In order to define the inter-slice structure of each intention-specific DBN, the following components must be specified:

Temporal/static nodes: The set of static and temporal nodes. Some nodes, such as emotions, are inherently temporal (colored yellow/green in Figure 4.2). Other nodes, such as gender, are static (colored pink in Figure 4.2). For some nodes, their static/temporal definition depends on the sampling rate of the network with regards to a given attribute. Note that some variables that were considered static in our BN are latent attributes, and thus represent the *approximated* value of those attributes. This approximation may be dynamically updated based on temporal evidence that is added in each time slot.

Inter-slice structure: Dependency relations between a subset of each DBN's temporal nodes that belong to different time slices. As we only considered a first-order model, it can be assumed that temporal relations may only include edges between two consecutive time slices. However, in higher-order models inter-slice relations may include other types

of edges as well. In fact, including historical values for some variables can be very useful for the inference of certain intentions. The inter-slice structure of each DBN was obtained using a combination of priors and data using a SL algorithm similar to the one described in Section 4.4.

Figure 4.2 presents our intention-specific DBNs. For simplicity, we omit network-level features.

As can be seen, a temporal link is created between variables that represent our target intentions in consecutive time slices. $P(intention_{i+1} \mid intention_i, U)$ represents the intention's evolution over time, given changes in other temporal variables in the network (U).

Interest-intention is an interesting relation. First, we see that interest may serve as either a cause or an effect of different intentions. Second, interest seems to be a cyclic process, to some extent, as can be concluded from $P(WI_i \mid interest_i, U)$ and $P(interest_{i+1} \mid WI_i)$, for example. Such a temporal relation might be attributed to the fact that interest in a certain topic assists in forming a behavioral intention related to that topic. After the intention has been formed, a new level of interest is formed, aimed at understanding how to fulfill that intention. In addition, $P(PI_{i+1} \mid interest_i, U)$ and $P(interest_{i+1} \mid PI_{i+1})$ show that both prior interest-level and current interest-level are important determinants of some intentions. In higher-order models, such a dependency relation might also include older interest levels. Such historical data can assist in identifying a sudden increase in the user's interest level.

"Opinion" is another interesting variable. An opinion might be influenced by multiple factors such as personality traits and demographic attributes as demonstrated by VI's $P(opinion_{i+1} \mid opinion_i, education, age, personality)$. Note that this cpt also contains $opinion_i$. This represents the fact that oftentimes, opinion is a self-propelling process: opinion at a given point in time, in addition to other factors, influences opinion at future points in time. A similar cpt structure is seen in "COVID-19 concern".

Fine-grained emotions were not used in either of the models. Furthermore, we weren't able to extract from the data meaningful inter-slice relations between different fine-grained emotions and the target intentions. We attribute that difficulty to the fact that unlike other features, emotions change quickly. Thus understanding emotions' temporal evolvement mechanism for each intention requires the use of finer-grained sampling rates.

Parameter learning: parameter learning was performed using the Expectation-Maximization algorithm [31]. This allowed us to use the *original*, incomplete training dataset. Our belief was that since our test datasets include a large number of missing values, training the DBN on incomplete datasets will allow the BN to take into consideration relations between missing and observed values of different features in different records, thus allowing the network to generalize better. Parameter learning was done in two stages: first, we trained a DBN using our labeled training data and probabilistically labeled the unlabeled training data; then, we trained a second DBN using both the labeled data and a subset of the unlabeled data on which the first DBN was the most confident. This semi-supervised approach allowed us to incorporate unlabeled data in the training process as well to reduce selection and visibility bias.

After each DBN is built and trained, it can be used to perform different kinds of inference and prediction by periodically sampling users' SN profiles, using the sampled data as temporal values for the temporal nodes in the DBN and running an inference algorithm. In this work, we used the Lauritzen-Spiegelhalter's junction tree algorithm [69], applied to *incomplete* test sets. In the next section, we show our inference results when using a two-slice DBN and SN data sampled twice.

4.6 Results

For a given intention j we tested its DBN DBN_j using our datasets as follows: in the first stage, (1), we trained DBN_j using $\mathcal{D}_j^1$ and tested it on $\mathcal{D}_j^3$. Only the first DBN's slice

Table 4.2: Results of the DBN models presented in this chapter

Intention	VI	WI	BI	PI	JI
Micro F1, (1)	.732	.832	.663	.763	.704
Macro F1, (1)	.73	.815	.54	.691	.623
Micro F1, (2)	.75	.831	.662	.812	.747
Macro F1, (2)	.741	.82	.526	.732	.699

was affected in this stage. In the second stage, (2), we trained DBN_j using $\mathcal{D}_j^2$ (implicitly using $\mathcal{D}_j^1$ as well due to the use of priors) and tested it on $\mathcal{D}_j^4$, using evidence data from $\mathcal{D}_j^3$ as well. Hence, inference results in (2) were obtained based on data and parameters from two slices of the DBN. The idea behind our approach was to simulate a real-world scenario in which SN data is sampled multiple times at different points. In such cases, all the data collected up to time t can be used as evidence to infer the target intention at time t (and predict it at time t', $t' > t$). Note that our test sets are *highly imbalanced*, in order to accurately represent the original distribution of each intention within our collected datasets (Table 4.1). Moreover, as we only consider the public portion of SN profiles, our datasets contain a large number of missing values. Those facts make the inference task highly challenging, both as a stand-alone task and compared to inference tasks in prior attribute-inference works.

Table 4.2 provides a detailed summary of our results. We report Micro F1 and Macro F1 scores for each stage ((1) and (2)) and for each intention-specific DBN. Note that the "intending" class is, almost always, the minority class. We also compare our average ROC AUC scores to those achieved by a discriminative classifier (SVM, see details below).

As can be seen, different intentions achieved significantly different Micro F1 and Macro F1 scores. BI's score is the lowest, whereas WI's score is the highest. A possible explanation for BI's performance is that applying for a loan is an intention that is often-times not publicly shared in SNs. However, other non-publicly shared intentions such as

JI scored significantly better than BI. This can be attributed to the fact that we were able to find other strong predictors for JI which don't depend on user-generated content, whereas for BI we failed to do so.

Figure 4.3 compares our average ROC AUC scores to those of a Support Vector Machine classifier (SVM, RBF kernel). Imputation of missing values was done using scikit-learn's IterativeImputer, a multivariate imputation method; it is currently considered the best imputation method offered by scikit-learn that can work with mixed datasets (*i.e.* numeric and categorical features). Note that because BNs are able to work directly with incomplete datasets, we did not need to perform any imputation on our datasets when fed to our BN models. As seen in Figure 4.3, our models outperform a SVM classifier on all five intentions, though the differences between the two classifiers vary between intentions. A possible explanation for those differences is the varying number of dependencies among each model's variables, or the existence of dependencies that are of specific importance for each inference task. Another possible explanation is the varying number of missing values within the features used for each intention's inference task.

When comparing Micro F1 and Macro F1 scores achieved in different stages ((1) and (2)) using the same DBN, we can see that the differences are more pronounced for PI and JI. This can be attributed to the underlying differences between different intentions. As evidenced by our data, intentions such as WI and VI can be seen as "continuous intentions" in the sense that the period of time between intention formation and completion of the associated behavior is longer than for other intentions; the persistence rate (participants who report the same intention in both the first and the second wave) of such intentions is significantly higher than rates reported for PI or JI. Another explanation for the varying differences is the different set of determinants of each intention. While the importance of some of those determinants stems from their intra-slice values (that is, their values at a given point in time), the importance of others is derived from a combination of intra-slice values *and* inter-slice change patterns between slices. For instance, various features

related to *non* user-generated content serve as excellent predictors of PI in (2), but only as solid predictors in (1). In a similar manner, the change in different MISC features such as attributes' visibility levels between (1) and (2) serve as an important predictor of JI in (2) (and for obvious reasons, can not be used for JI's inference in (1)).

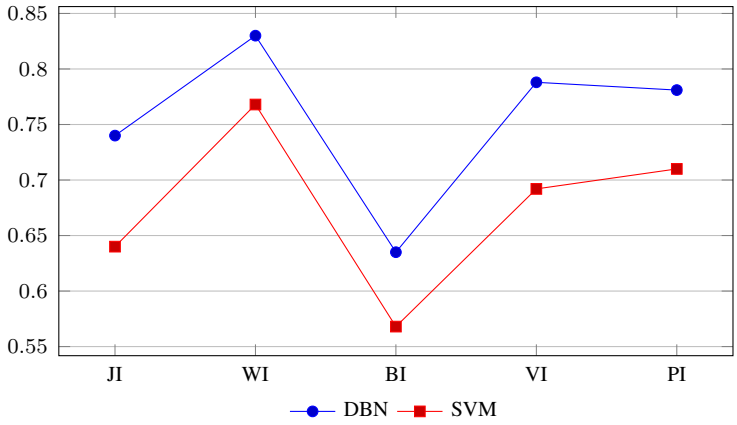


Figure 4.3: Average ROC AUC scores

4.7 Related work

Inference of personal attributes using SN data has been extensively researched. Inferring users' personality type was investigated in [45, 46] using regression models and Twitter/Facebook data, respectively. Youyou *et al.* [111] showed that automatic inference methods that rely on Facebook likes achieve better prediction accuracy than those achieved by asking the users' friends. Staiano *et al.* [105] used data gathered through smartphones such as calls and texts; their results significantly vary across different personality dimensions.

Demographic attributes' inference is another well-studied topic, with age and gender being the most researched attributes [100, 118].

A related stream of research focuses on psychological and mental conditions. Depression is the most researched condition, followed by anxiety and stress [30, 38].

The common denominator of all the above works is that they focus on attributes that are either static (their values rarely change), non-self controlled, or both.

Inference of self-controlled attributes has also been extensively studied. However, such works focus on the inference of opinions and attitudes [29, 90] rather than behavioral attributes. While a substantial amount of work does study different types of behavioral attributes, their goals are different than ours. Such works study general correlations between network or linguistic features and a given behavior, identify the prevalence of a certain behavior among the general population, or classify SN textual objects such as tweets or posts. For example, while there exists a considerable amount of work about the use of SNs for monitoring public health, none of those works aims at inferring vaccination intent of a given SN user at a given point in time. Rather, existing works analyze collective sentiment towards vaccinations [81], track the spread of infectious diseases [68], or perform classification of stand-alone SN objects according to vaccination attitudes or intentions of the object's creator [8, 55].

Inferring time-varying, behavioral attributes using public SN data has therefore been hardly researched, with two exceptions: voting intentions and *online* purchase intentions. There are several key differences between this work and prior ML work on PI. First, the majority of existing works examine general buying preferences rather than time-varying PIs [119]. Other works try to infer PI of stand-alone SN objects (content-centric) rather than PI of SN users (user-centric), an approach which is inherently biased [10, 51]. Note that [10]'s test set is *perfectly balanced*; such a test-set composition is far from being representative of any real-world tweet set as the vast majority of tweets do not reflect a PI. The remaining works that do try to infer a user-centric, time-varying PI use data derived solely from E-commerce platforms. Such data is both platform-specific, and oftentimes considered private (session logs, for example), unlike our use of public SN data [83]. The closest work to ours is [77] which infers PI of Pinterest users using static and temporal

features. However, they only consider Pinterest users (31% of Americans [2]); furthermore, they only consider online purchases and do not differentiate between different product-categories.

[2]https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/

Chapter 5

Conclusions and open problems

In this final chapter, we briefly review our conclusions and offer some directions for future research.

5.1 Privacy-preserving data sharing

In Chapter 2, we presented PRShare, an interorganizational data-sharing framework that protects the privacy of data owners, data clients, and data subjects. In designing PRShare, we introduced the novel concept of *Attribute-Based Encryption With Oblivious Attribute Translation* (OTABE), which may be of independent interest. Our experimental results indicate that the performance of our OTABE-based data-sharing framework is competitive with that of earlier MA-ABE schemes that provide less sophisticated privacy guarantees.

One natural open question is whether it is possible to relax one or more assumptions that PRShare relies on. For example, can the proxies in PRShare be malicious?

We plan to investigate connections between the techniques developed for PRShare and OTABE and the areas of blockchain systems, smart contracts, and cryptocurrencies. One such connection may enable the use of malicious proxies: We conjecture that existing work on smart contracts [12, 113] can be combined with fair-exchange protocols [6, 7, 42]

to remove the assumption that proxies in OTABE are honest but curious. Conversely, we plan to use OTABE to improve existing blockchain access-control mechanisms.

Another natural goal for future work is to demonstrate the power and applicability of OTABE in concrete applications. We plan to provide one such demonstrations by implementing a credit-report-management system that makes essential use of blockchains and OTABE.

5.2 Social-network mining to predict voting behavior

In Chapter 3, we presented a novel approach to predicting the voting behavior of Facebook users based on a BN model that combines diverse yet complementary types of information about the user. In contrast to previous works, we made use of data about ordinary Facebook users, thus avoiding the bias entailed in cherry-picked datasets that are limited to politically active users such as those who are most active on Twitter. Using a semi-supervised method, we applied our model to the case of the 2016 U.S. elections, achieving promising results despite large amounts of missing data.

Interesting avenues for future research include augmenting the model with additional behavioral and interest-based traits, combining more complex measures of the network neighborhood, and adding temporal features that capture how various attributes change over time. Temporal features may be particularly useful in election-oriented applications, examples of which include identifying swing voters by examining whether users' political opinions are consistent over time and measuring the extent to which a specific event influences a user's voting intentions.

5.3 Inferring behavioral intentions of social-network users

In Chapter 4, we presented a new, BN-based methodology for inferring the intentions of SN users. We reconceived SN users using DBNs and built intention-specific DBN models that can capture the temporal nature of the human decision-making process. Our DBN models also handle common challenges in SN-based inference, including incomplete datasets, unlabeled data, and bidirectional influence. We evaluated our methodology on multiple real-world inference tasks and multi-wave SN data, achieving promising results despite the use of highly imbalanced, incomplete datasets.

Interesting directions for further related work include the use of higher-order DBNs and decision-specific sampling rates. Specifically, we plan to explore the combination of advanced SN features such as photos and videos.

Bibliography

[1] Health insurance portability and accountability act. https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html, 1996.

[2] Cambridge analytica and facebook: The scandal and the fallout so far. https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html, 2018.

[3] Social media use in 2018. https://www.pewinternet.org/2018/03/01/social-media-use-in-2018/, 2018.

[4] Joseph A. Akinyele, Christina Garman, Ian Miers, Matthew W. Pagano, Michael Rushanan, Matthew Green, and Aviel D. Rubin. Charm: a framework for rapidly prototyping cryptosystems. *Journal of Cryptographic Engineering*, 3(2):111–128, 2013.

[5] Joseph A. Akinyele, Matthew W. Pagano, Matthew D. Green, Christoph U. Lehmann, Zachary N. J. Peterson, and Aviel D. Rubin. Securing electronic medical records using attribute-based encryption on mobile devices. In *1st Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 75–86. ACM, 2011.

[6] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. Fair two-party computations via bitcoin deposits. In *18th Interna-*

tional Conference on Financial Cryptography and Data Security, pages 105–121. Springer, 2014.

[7] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. Secure multiparty computations on bitcoin. In *38th Symposium on Security and Privacy*, pages 443–458. IEEE, 2016.

[8] Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. Twitter catches the flu: detecting influenza epidemics using twitter. In *12th Conference on Empirical Methods in Natural Language Processing*, pages 1568–1576. ACM, 2011.

[9] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *12th Network and Distributed System Security Symposium*, pages 29–43. IEEE, 2005.

[10] Samed Atouati, Xiao Lu, and Mauro Sozio. Negative purchase intent identification in twitter. In *29th International Conference on World Wide Web*, pages 2796–2802. ACM, 2020.

[11] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *14th International Conference on Practice and Theory in Public-Key Cryptography*, pages 90–108. Springer, 2011.

[12] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *2nd International Conference on Open and Big Data*, pages 25–30. ACM, 2016.

[13] Johannes Bachhuber, Christian Koppeel, Jeronim Morina, Kim Rejström, and David Steinschulte. Us election prediction: A linguistic analysis of US twitter users.

In *Designing Networks for Innovation and Improvisation*, pages 55–63. Springer, 2016.

[14] Pablo Barberá. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. *Political Analysis*, 23(1):76–91, 2015.

[15] Sana Belguith, Nesrine Kaaniche, Maryline Laurent, Abderrazak Jemai, and Rabah Attia. PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT. *Computer Networks*, 133:141–156, 2018.

[16] Tara Siegel Bernard, Tiffany Hsu, Nicole Perlroth, and Ron Lieber. Equifax says cyberattack may have affected 143 million in the U.S. *The New York Times*, September 7, 2017.

[17] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *28th Symposium on Security and Privacy*, pages 321–334. IEEE, 2007.

[18] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *17th EUROCRYPT*, pages 127–144. Springer, 1998.

[19] Robert Bond and Solomon Messing. Quantifying social media's political space: Estimating ideology from public revealed preferences on facebook. *American Political Science Review*, 109:62–78, 2015.

[20] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public-key encryption with keyword search. In *23rd EUROCRYPT*, pages 506–522. Springer, 2004.

[21] Dan Boneh, Xuhua Ding, and Gene Tsudik. Fine-grained control of security capabilities. *ACM Transactions on Internet Technology*, 4(1):60–82, 2004.

[22] Antoine Boutet, Hyoungshick Kim, and Eiko Yoneki. What's in your tweets? I know who you supported in the UK 2010 general election. In *6th International Conference on Web and Social Media*. AAAI, 2012.

[23] Pamela Brink and Kristi Ferguson. The decision to lose weight. *Western Journal of Nursing Research*, 20(1):84–102, 1998.

[24] Matthew C. MacWilliams. Forecasting congressional elections using facebook data. *Political Science & Politics*, 48:579–583, 2015.

[25] Melissa Chase. Multi-authority attribute based encryption. In *4th Theory of Cryptography Conference*, pages 515–534. Springer, 2007.

[26] Nathan Chenette, Kevin Lewi, Stephen A. Weiss, and David J. Wu. Practical order-revealing encryption with limited leakage. In *23rd International Conference on Fast Sofware Encryption*, pages 474–493. Springer, 2016.

[27] Jie Cheng, David A. Bell, and Weiru Liu. Learning belief networks from data: An information theory based approach. In *6th International Conference on Information and Knowledge Management*, pages 325–331. ACM, 1997.

[28] Raviv Cohen and Derek Ruths. Classifying political orientation on twitter: It's not easy! In *7th International Conference on Web and Social Media*. AAAI, 2013.

[29] Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. Predicting the political alignment of twitter users. In *3rd International Conference on Social Computing*, pages 192–199. IEEE, 2011.

[30] Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. Predicting depression via social media. In *7th International Conference on Web and Social Media*. AAAI, 2013.

[31] AP Dempster and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–22, 1977.

[32] Xin Dong, Jiadi Yu, Yuan Luo, Yingying Chen, Guangtao Xue, and Minglu Li. Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. *Computers and Security*, 42:151–164, 2013.

[33] Electronic Communications Privacy Act, Public law 99-508. `https://it.ojp.gov/PrivacyLiberty/authorities/statutes/1285`, 1986.

[34] Paul Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200, 1992.

[35] Benjamin Fabian, Tatiana Ermakova, and Philipp Junghanns. Collaborative and secure sharing of healthcare data in multi-clouds. *Information Systems*, 48:132–150, 2015.

[36] Fair Credit Reporting Act, Public law 91-508. `https://www.consumer.ftc.gov/articles/pdf-0111-fair-credit-reporting-act.pdf`, October 26, 1970.

[37] Federal Trade Commission. Equifax data breach settlement. `https://www.ftc.gov/enforcement/cases-proceedings/refunds/equifax-data-breach-settlement`, 2017.

[38] Katya C Fernandez, Cheri A Levinson, and Thomas L Rodebaugh. Profiling: Predicting social anxiety from facebook profiles. *Social Psychological and Personality Science*, 3(6):706–713, 2012.

[39] Martin Fishbein. The role of theory in HIV prevention. *AIDS Care*, 12(3):273–278, 2000.

[40] Jonathan Frankle, Sunoo Park, Daniel Shaar, Shafi Goldwasser, and Daniel Weitzner. Practical accountability of secret processes. In *27th USENIX Security Symposium*, pages 657–674, 2018.

[41] David Froelicher, Patricia Egger, Joao Sa Sousa, Jean Louis Raisaro, Zhicong Huang, Christian Mouchet, Bryan Ford, and Jean-Pierre Hubaux. UnLynx: A decentralized system for privacy-conscious data sharing. *Proceedings on Privacy Enhancing Technologies Symposium*, 2017(4):232–250, 2017.

[42] V. Lekakis C. Papamanthou E. Paraskevas G. Ateniese, M. T. Goodrich and R. Tamassia. Accountable storage. In *15th International Conference on Applied Cryptography and Network Security*, pages 623–644. Springer, 2017.

[43] Daniel Gayo-Avello. Don't turn social media into another 'literary digest' poll. *Communications of the ACM*, 54(10):121–128, 2011.

[44] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *1st SIGCHI Conference on Human Factors in Computing Systems*, pages 211–220. ACM, 2009.

[45] Jennifer Golbeck, Cristina Robles, Michon Edmondson, and Karen Turner. Predicting personality from twitter. In *3rd International Conference on Social Computing*, pages 149–156. IEEE, 2011.

[46] Jennifer Golbeck, Cristina Robles, and Karen Turner. Predicting personality with social media. In *12th SIGCHI Conference on Human Factors in Computing Systems*, pages 253–262. ACM, 2011.

[47] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *13th Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.

[48] Mark Granovetter. The strength of weak ties: A network theory revisited. *Sociological Theory*, 1:201–233, 1983.

[49] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In *5th International Conference on Applied Cryptography and Network Security*, pages 288–306. Springer, 2007.

[50] Matthew Green, Susan Hohenberger, and Brent Waters. Outsourcing the decryption of ABE ciphertexts. In *20th USENIX Security Symposium*, pages 523–538, 2011.

[51] Vineet Gupta, Devesh Varshney, Harsh Jhamtani, Deepam Kedia, and Shweta Karwa. Identifying purchase intent from social posts. In *8th International Conference on Web and Social Media*. AAAI, 2014.

[52] Stuart Hampshire and Herbert LA Hart. Decision, intention and certainty. *Mind*, 67(265):1–12, 1958.

[53] Jianming He, Wesley W. Chu, and Zhenyu Liu. Inferring privacy information from social networks. In *4th International Conference on Intelligence and Security Informatics*, pages 154–165. IEEE, 2006.

[54] David Heckerman. A tutorial on learning with bayesian networks. In *Innovations in Bayesian Networks: Theory and Applications*, pages 33–82. Springer, 2008.

[55] Xiaolei Huang, Michael C Smith, Michael J Paul, Dmytro Ryzhkov, Sandra C Quinn, David A Broniatowski, and Mark Dredze. Examining patterns of influenza vaccination in social media. In *31st Conference on Artificial Intelligence*. AAAI, 2017.

[56] Luan Ibraimi, Milan Petkovic, Svetla Nikova, Pieter H. Hartel, and Willem Jonker. Mediated ciphertext-policy attribute-based encryption and its application. In *10th*

International Conference on Information Security Applications, pages 309–323. Springer, 2009.

[57] Lihi Idan and Joan Feigenbaum. Show me your friends, and I will tell you whom you vote for: predicting voting behavior in social networks. In *11th International Conference on Advances in Social Network Analysis and Mining*, pages 816–824. IEEE/ACM, 2019.

[58] Lihi Idan and Joan Feigenbaum. PRShare: A framework for privacy-preserving, interorganizational data sharing. In *19th Workshop on Privacy in the Electronic Society*, pages 137–149. ACM, 2020.

[59] Lihi Idan and Joan Feigenbaum. PRShare: A framework for privacy-preserving interorganizational data sharing, Yale University Technical Report YALEU/DCS/TR-1554. `https://cpsc.yale.edu/sites/default/files/files/tr1554.pdf`, 2020.

[60] Kokil Jaidka, Sharath Guntuku, and Lyle Ungar. Facebook vs. twitter: Cross-platform differences in self-disclosure and trait prediction. In *12th International Conference on Web and Social Media*. AAAI, 2018.

[61] Jinyuan Jia, Binghui Wang, and Le Zhang. Attriinfer: Inferring user attributes in online social networks using markov random fields. In *26th International Conference on World Wide Web*, pages 1561–1569. ACM, 2017.

[62] Seny Kamara. Restructuring the NSA metadata program. In *2nd Financial Cryptography Workshop on Applied Homomorphic Cryptography and Encrypted Computing*, pages 235–247. Springer, 2014.

[63] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[64] Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *13th International Conference on Machine Learning*, pages 284–292. ACM, 1996.

[65] Jakob Bæk Kristensen, Thomas Albrechtsen, Emil Dahl-Nielsen, Michael Jensen, Magnus Skovrind, and Tobias Bornakke. Parsimonious data: How a single facebook like predicts voting behavior in multiparty systems. *PloS one*, 12(9), 2017.

[66] Joshua A. Kroll, Edward W. Felten, and Dan Boneh. Secure protocols for accountable warrant execution. https://www.cs.princeton.edu/ felten/warrant-paper.pdf, 2014.

[67] Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng. Attribute-based encryption with verifiable outsourced decryption. *IEEE Transactions on Information Forensics and Security*, 8(8):1343–1354, 2013.

[68] Alex Lamb, Michael J. Paul, and Mark Dredze. Separating fact from fear: Tracking flu infections on twitter. In *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 789–795. ACL, 2016.

[69] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 50(2):157–194, 1988.

[70] Jiguo Li, Wei Yao, Yichen Zhang, Huiling Qian, and Jinguang Han. Flexible and fine-grained attribute-based data storage in cloud computing. *IEEE Transactions on Services Computing*, 10(5):785–796, 2017.

[71] Ming Li, Shucheng Yu, Kui Ren, and Wenjing Lou. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In *6th International Conference on Security and Privacy in Communication Networks*, pages 89–106. Springer, 2010.

[72] Kaitai Liang, Liming Fang, Willy Susilo, and Duncan S. Wong. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security. In *5th International Conference on Intelligent Networking and Collaborative Systems*, pages 552–559. IEEE, 2013.

[73] Xiaohui Liang, Zhenfu Cao, Huang Lin, and Jun Shao. Attribute based proxy re-encryption with delegating capabilities. In *4th Symposium on Information, Computer, and Communications Security*, pages 276–286. ACM, 2009.

[74] Drew A Linzer. Dynamic bayesian forecasting of presidential elections in the states. *Journal of the American Statistical Association*, 108:124–134, 2013.

[75] Chang Liu, Xiao Shaun Wang, Kartik Nayak, Yang Huang, and Elaine Shi. ObliVM: A programming framework for secure computation. In *36th Symposium on Security and Privacy*, pages 359–376. IEEE, 2015.

[76] Xuefeng Liu, Yuqing Zhang, Boyang Wang, and Jingbo Yan. Mona: Secure multi-owner data sharing for dynamic groups in the cloud. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1182–1191, 2013.

[77] Caroline Lo, Dan Frankowski, and Jure Leskovec. Understanding behaviors that lead to purchasing: A case study of pinterest. In *22nd International Conference on Knowledge Discovery and Data Mining*, pages 531–540. ACM, 2016.

[78] Charlotte N Markey and Patrick M Markey. Relations between body image and dieting behaviors: An examination of gender differences. *Sex Roles*, 53(7-8):519–530, 2005.

[79] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[80] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J. Niels Rosenquist. Understanding the demographics of twitter users. In *5th International Conference on Web and Social Media*. AAAI, 2011.

[81] Tanushree Mitra, Scott Counts, and James W. Pennebaker. Understanding anti-vaccination attitudes in social media. In *10th International Conference on Web and Social Media*. AAAI, 2016.

[82] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[83] Osnat Mokryn, Veronika Bogina, and Tsvi Kuflik. Will this session end with a purchase? inferring current purchase intent of anonymous visitors. *Electronic Commerce Research and Applications*, 34, 2019.

[84] Kevin Patrick Murphy. *Dynamic bayesian networks: representation, inference and learning*. University of California, Berkeley, 2002.

[85] Eni Mustafaraj, Samantha Finn, Carolyn Whitlock, and Panagiotis Takis Metaxas. Vocal minority versus silent majority: Discovering the opionions of the long tail. In *3rd International Conference on Social Computing*, pages 103–110. IEEE, 2011.

[86] Kartik Nayak, Xiao Shaun Wang, Stratis Ioannidis, Udi Weinsberg, Nina Taft, and Elaine Shi. GraphSC: Parallel secure computation made easy. In *36th Symposium on Security and Privacy*, pages 377–394. IEEE, 2015.

[87] Takashi Nishide, Kazuki Yoneyama, and Kazuo Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. In *6th International Conference on Applied Cryptography and Network Security*, pages 111–129. Springer, 2008.

[88] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *14th Conference on Computer and Communications Security*, pages 195–203. ACM, 2007.

[89] Cristina Ottaviani and Daniela Vandone. Impulsivity and household indebtedness: Evidence from real life. *Journal of Economic Psychology*, 32(5):754–761, 2011.

[90] Marco Pennacchiotti and Ana-Maria Popescu. A machine learning approach to twitter user classification. In *5th International Conference on Web and Social Media*. AAAI, 2011.

[91] Raluca Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. CryptDB: Protecting confidentiality with encrypted query processing. In *23rd Symposium on Operating Systems Principles*, pages 85–100. ACM, 2011.

[92] Anna Priante, Djoerd Hiemstra, Tijs van den Broek, Aaqib Saeed, and Ariana Need. # whoami in 160 characters? classifying social identities based on twitter profile descriptions. In *1st Workshop on NLP and Computational Social Science*, pages 55–65. ACL, 2016.

[93] Yogachandran Rahulamathavan, Raphael C.-W. Phan, Muttukrishnan Rajarajan, Sudip Misra, and Ahmet Kondoz. Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. In *11th International Conference on Advanced Networks and Telecommunications Systems*. IEEE, 2017.

[94] B. Rammstedt and O.P. John. Measuring personality in one minute or less: A 10-item short version of the big five inventory. *Journal of Research in Personality*, 41:203–212, 2007.

[95] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classify-

ing latent user attributes in twitter. In *2nd International Workshop on Search and Mining User-Generated Contents*, pages 37–44. ACM, 2010.

[96] Yannis Rouselakis and Brent Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In *20th Conference on Computer and Communications Security*, pages 463–474. ACM, 2013.

[97] Yannis Rouselakis and Brent Waters. Efficient statically-secure large-universe multi-authority attribute-based encryption. In *19th International Conference on Financial Cryptography and Data Security*, pages 315–332. Springer, 2015.

[98] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *32nd CRYPTO*, pages 199–217. Springer, 2012.

[99] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *24th EUROCRYPT*, pages 457–473. Springer, 2005.

[100] H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, and Martin EP Seligman. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9), 2013.

[101] Aaron Segal, Joan Feigenbaum, and Bryan Ford. Open, privacy-preserving protocols for lawful surveillance. *CoRR*, abs/1607.03659, 2016.

[102] Aaron Segal, Joan Feigenbaum, and Bryan Ford. Privacy-preserving lawful contact chaining [preliminary report]. In *15th Workshop on Privacy in the Electronic Society*, pages 185–188. ACM, 2016.

[103] Paschal Sheeran. Intention—behavior relations: a conceptual and empirical review. *European Review of Social Psychology*, 12(1):1–36, 2002.

[104] Yanfeng Shi, Qingji Zheng, Jiqiang Liu, and Zhen Han. Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation. *Information Sciences*, 295:221–231, 2015.

[105] Jacopo Staiano, Bruno Lepri, Nadav Aharony, Fabio Pianesi, Nicu Sebe, and Alex Pentland. Friends don't lie: inferring personality traits from social network structure. In *14th Conference on Ubiquitous Computing*, pages 321–330. ACM, 2012.

[106] Emil Aas Stoltenberg. Bayesian forecasting of election results in multiparty systems. Master's thesis, Department of Political Science, University of Oslo, 2013.

[107] Yla Tausczik and James W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.

[108] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G. Sandner, and Isabell M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *4th International Conference on Web and Social Media*. AAAI, 2010.

[109] Dhinakaran Vinayagamurthy, Alexey Gribov, and Sergey Gorbunov. StealthDB: a scalable encrypted database with full SQL query support. *Proceedings on Privacy Enhancing Technologies*, 2019(3):370–388, 2019.

[110] Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. Inferring user political preferences from streaming communications. In *52nd Annual Meeting of the Association for Computational Linguistics*, pages 186–196. ACL, 2014.

[111] M. Kosinski W. Youyou and D. Stillwell. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, 112(4):1036–1040, 2015.

[112] Guojun Wang, Qin Liu, and Jie Wu. Hierarchical attribute-based encryption for fine-grained access control in cloud-storage services. In *17th Conference on Computer and Communications Security*, pages 735–737. ACM, 2010.

[113] Hao Wang and Yujiao Song. Secure cloud-based ehr system using attribute-based cryptosystem and blockchain. *Journal of Medical Systems*, 42(8):1–9, 2018.

[114] Felix Ming Fai Wong, Chee Wei Tan, Soumya Sen, and Mung Chiang. Quantifying political leaning from tweets and retweets. In *7th International Conference on Web and Social Media*. AAAI, 2013.

[115] Xuanxia Yao, Zhi Chen, and Ye Tian. A lightweight attribute-based encryption scheme for the Internet of Things. *Future Generation Computer Systems*, 49:104–112, 2015.

[116] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *29th Conference on Computer Communications*, pages 534–542. IEEE, 2010.

[117] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Attribute-based data sharing with attribute revocation. In *5th Symposium on Information, Computer, and Communications Security*, pages 261–270. ACM, 2010.

[118] F. Zamal, W. Liu, and D. Ruths. Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors,. In *6th International Conference on Web and Social Media*. AAAI, 2012.

[119] Yongzheng Zhang and Marco Pennacchiotti. Predicting purchase behaviors from social media. In *22nd International Conference on World Wide Web*, pages 1521–1532. ACM, 2013.

[120] Elena Zheleva and Lise Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *18th International Conference on World Wide Web*, pages 531–540. ACM, 2009.