Presence Patterns and Privacy Analysis

Ella Roubtsova[1], Serguei Roubtsov[2], and Greg Alpar[1]

[1] Open University of the Netherlands,
Ella.Roubtsova@ou.nl,Greg.Alpar@ou.nl
[2] Technical University Eindhoven, the Netherlands,
s.roubtsov@tue.nl

Abstract. Business applications often use such data structures as Presence Patterns for presentation of numbers of customers in service-oriented businesses including education, retailing or media. Presence Patterns contain open data derived from internal data of organizations. In this paper, we investigate different ways of defining Presence Patterns and possible privacy consequences dependent on the chosen definition. The first contribution of the paper is a definition of a family of Presence Patterns. The second contribution is a method for privacy analysis of Presence Patterns.

Keywords: Presence patterns, Privacy Requirements, Method for Privacy Analysis of Presence Patterns

1 Introduction

Businesses nowadays provide valuable Internet services to their customers or employees which require registration of customers or employees. This network data contain information about login and logout that can correspond to the 'presence' of registered users. In this paper, we assume that 'presence' online means actual or physical 'presence'. [3] This information about 'presence' can be used for different business analytics purposes. Business applications often use such data structures as Presence Patterns for presentation of numbers customers in service-oriented businesses, including education, retailing and media. However, businesses are not allowed to use personal data without limitations. The use of personal data is regulated by laws and, in particular, by the General Data Protection Regulation (GDPR).[4] The major privacy rule, coming from the GDPR, allows businesses to use only the minimum data needed for the given task and

[3] The login and logout do not always correspond to the physical presence of the logged persons. Nevertheless, the degree of correspondence can be always established by manual control of presence during a chosen time period by sampling. Usually, this degree of correspondence is high, as many people use their mobile phones or PC to connect to the organizational network because this connection provides useful information.

[4] General Data Protection Regulation (GDPR)(EU) 2016/679 GDPR official: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679

does not allow them to relate business indicators to personal data of clients. The way to meet this privacy requirement is to pay attention to new business concepts used by business analytics, carefully design the data structures for all new business concepts and avoid relations with personal data in their definitions.

Because of the GDPR, companies (and organisations in general) have to comply with the privacy-by-design principles [**?**,**?**]. Businesses have a hard time to figure out how to do this. They also experience pressure from the technology side. Ever more possibilities and new opportunities emerge every day. However, these new technologies are often coupled with extensive data usage making hard to assess how privacy invasive their deployment can be.

In this paper, we investigate different ways of defining so-called Presence Patterns and possible privacy violations caused by their coupling with the organisation's log files and also with, possibly relevant, publicly-available information. Section 2 presents the work related to protection of privacy. Section 3 defines a family of Presence Patterns. Section 4 defines a method for privacy analysis of Presence Patterns. The method is illustrated with examples of such patterns with and without possible privacy violation. Section 5 presents conclusions and future work.

2 Related Work

2.1 Privacy, Anonymity and *k*-anonymity

Privacy studies conclude that there are quite a few wrong beliefs about privacy protection. Sweeney [**?**] reports that it is believed that removing explicit identifiers, such as name, address, telephone number makes data anonymous and protect privacy. However, the studies show that combinations of a few publicly available characteristics "uniquely or nearly uniquely identify some individuals".

The Internet contains many independent data sources. They may contain the same personal information with different extensions and, as a result, may partially release this personal information. In this sense, the Internet is similar to a multi-level database. It is formally proven by Su and Ozsoyoglu [**?**], that in general, it is impossible to guarantee privacy in a multi-level database (due to functional dependencies and multi-level dependencies of data).

The public data can come from registers (phonebook-like databases), social networks (LinkedIn, Facebook, etc.), public ratings like on IMDb, Amazon or other websites and webshops. Moreover, we increasingly rely on infrastructures that collect, store and (possibly) publish data. For example, "smart" driver assisting services are aware of the current vehicle position and can potentially emit lots of data about your vehicle to the outside world. These data items may not be identifying in themselves, but when being aggregated, they can cause almost unforeseeable privacy violations.

Sweeney [**?**] proposed a model "for understanding, evaluating and constructing computational systems that control inferences" or private information. The model is based on the definition of a `quasi-identifier`.

- A table T is a finite number of tuples $(\mathtt{A}_1, ...\mathtt{A}_n)$, where $(\mathtt{A}_1, ...\mathtt{A}_n)$ are attributes.
- A quasi-identifier is defined as follows. "Given a population of entities (a universe)U, an entity-specific table $\mathtt{T}(\mathtt{A}_1, \dots, \mathtt{A}_n)$, $\mathtt{f}_c : \mathtt{U} \rightarrow \mathtt{T}$ and $\mathtt{f}_g : \mathtt{T} \rightarrow \mathtt{U'}$, where $\mathtt{U} \subseteq \mathtt{U'}$. A quasi-identifier of T, written $\mathtt{Q}_T$, is a subset of attributes $\{\mathtt{A}_i, \dots, \mathtt{A}_j\} \subseteq \{\mathtt{A}_1, \dots, \mathtt{A}_n\}$ where: $\exists \mathtt{p}_i \in \mathtt{U}$ such that $\mathtt{f}_g(\mathtt{f}_c(\mathtt{p}_i)[\mathtt{Q}_T]) = \mathtt{p}_i$."

In other words, a quasi-identifier $\mathtt{Q}_T$ of a table T is a subset of table's columns named by attributes $(\mathtt{A}_1, ...\mathtt{A}_j)$. A tuple of values of the attributes of the quasi-identifier, that appears in a row of table T, is a value or an occupance of the quasi-identifier.

Sweeney gives an example, of a voter-specific table (universe U) and a health patient-specific table (universe U'). In the voter-specific table, there are fields {Name, Address, ZIP, Birth date, Sex}. The medical data contain {Visit date, Diagnosis, ZIP, Birth date, Sex}. The quasi-identifier {ZIP, Birth date, Sex} can be used to re-identify the medical information of the voters. He concludes that "the attributes that appear in the private data and also appear in the public data are the candidates for linking: therefore, these attributes constitute the quasi-identifier and the disclosure of these attributes must be controlled". For such a control, he proposes an additional requirement of k-anonymity for any released data with associated quasi-identifier. By the released data, any publicly available table T is meant.

The k-anonymity of any table T means that each tuple of attribute-values of the table's quasi-identifier $\mathtt{Q}_T$ appears at least in k-rows of this table $\mathtt{T}[\mathtt{Q}_T]$.

Let it be two tables, a private table PT and a public (released or open) table T. It is proven that, "if the released data T satisfies k-anonymity with respect to the quasi-identifier $\mathtt{Q}_{PT}$ of a private table PT, then the combination of the released data T and the external sources on which $\mathtt{Q}_{PT}$ was based, cannot link on $\mathtt{Q}_{PT}$ or a subset of its attributes to match fewer than k individuals" [?].

2.2 Inference Attacks and Open Data

The k-anonymity of open (and released) data cannot protect from specific data mining techniques, called inference attacks, which use data queries, aggregation of data, sorting, etc. However, the attacks demand resources. The open data should not make the life of an attacker easier.

The most representative sources of open data are the social networks. The attacks on social networks are described in [?]. This work states that only a lack of resources can stop attackers from massive crawling via API or "screen-scraping" [?]. Gross and Acquisti [?] demonstrate that the attributes of the nodes (rows in a entity-specific data table), such as social security numbers and other profile attributes [?], can be predicted with higher accuracy than random guessing.

Besides the nodes (rows in a data table), the social network expose edges, being relations between nodes. Edges provide additional information about nodes (persons) and their behaviour patterns [?].

Social networks provide some protection from attackers. Attackers need to create many dummy nodes helping to re-identify social network members. The online social networks "check the uniqueness of e-mail addresses, and deploy other methods for verifying accuracy of supplied information making creating of dummy nodes relatively difficult" [?].

Nevertheless, on small scale, for defined targets found in the released (open) data, the attacks can be feasible. So, the awareness about quasi-identifiers in any released data should be raised. In this paper, we talk about the released data in form of Presence Patterns.

3 Presence Pattern

In this section we define a family of Presence Patterns: simple presence pattern and several extensions of it for real-world applications. In the next section we use these definitions to propose a method for privacy analysis of such patterns.

3.1 Simple Presence Pattern

Presence of someone or something is the state of existing of someone or something in a given place in a given time interval. The given place can be described as a class room, a shop, but may also be seen as the state "online" in a media. Presence Patterns are often released for public. For example, a polyclinic may release a presence pattern as patients may choose a less visited day for their visit.

A Simple Presence Pattern in business usually concerns with numbers of present customers. As businesses are interested in performance and revenues, they are interested in the number of customers who are in the state "In" during a given time interval. They can estimate how many human and other resources are needed for service of customers. We define a Simple Presence Pattern as a mapping of numbers of present customers onto natural time intervals defined during a week.

```
Simple Presence Pattern = (Days, Hours, NumberOfPresent, function-times,
                                  function-presence)
```

- `Days={Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}`.
- `Hours={1,2,3,....24}`.
- `NumberOfPresent` is a finite natural number (positive integer number or zero).
- `function-times: Hours` → `Days` divides days of the week into time intervals.
- `function-presence:NumberOfPresent`→ `(Hours` → `Days)` assigns the numbers of customers to the time intervals for each day of the week.

3.2 Presence Pattern extended with a Schedule

Any real Presence Pattern used for a given business purpose is usually an extension of the Simple Presence Pattern. The most widely used sources of extension are schedules.

```
Schedule = (Days, Times, OpeningStatus, function-times, function-opening)
```

- Times= (Morning work hours 9:00-12:00, Evening work hours 13:00-16:00, Closing times 16:00-9:00);
- OpeningStatus ={Open,Closed}.
- function-times: Times → Days;
- function-opening = OpeningStatus → (Times → Days).

A schedule is easily included into the Presence Pattern.

```
Presence Pattern = (Days, Times, NumberOfPresent, OpeningStatus,
     function-times, function-opening, function-presence).
```

3.3 Presence Pattern and Schedule with Business Information

In practice, schedules contain also sets and relations [?] with business related information. For instance, in an educational institution, an address with the room number (say, "R206"), and a course name (for example, "Physics"), name of a teacher usually correspond to some time interval.

```
Presence Pattern=(Days, Times, NumberOfPresent, Locations, Courses,
   Teachers, function-times, function-location, function-course,
            function-teacher-name, function-presence).
```

- function-location = Address→ (Times → Days)
- function-course = Courses → (Times → Days)
- function-teacher = Teachers→ (Times → Days)

This Presence Pattern contains private information about the service provider (addresses of an educational institute and the names of its teachers).

3.4 Presence Pattern, Event Log and User Profile

Another data structure that is closely related to Presence Patterns is an event log. Industrial event logs are emitted by operating systems and applications to record, among other information, online events. The record of any occurrence of a login or logout of a network device always contain a time stamp and may contain private information like the user IP address or the MAC address of a mobile device or a PC, activity (associated or disassociated) with the time stamp. An event record often contains additional information such as the profile (or the access right) of the user (guest, employee, etc), protocol and so on.

A time stamp, an activity (state) and a User Identifier are often considered as the main fields of any industrial log.

```
Event log=(Time stamp, User, Activity, function-activity-time,
                      function-user-time
```

- Time stamp is a set of time stamps, for example, 12.07.2017.10:25.
- Activity = {In, Out}. For example, In means login or check-in; Out means Logout or check-out.

- `User` is a set of user unique IDs such as names or addresses of personal devices.

Event logs do not belong to any Presence Pattern, but they are used to derive the calculated information for Presence Patterns. For this, the event logs can be automatically processed in order to produce the `NumberOfPresent. NumberOfPresent` has a `one-to-many` relation with the users in the log. For every time interval in the schedule, the automated query derives the users logged in within this time interval and logged out within or after this time interval. If for a user there is a time stamp `In` before or within the given `Time Interval i` and there is a stamp `Out` within or after it, then this user is present in the given time interval and the counter `NumberOfPresent` is increased by `1`. By counting, the private information is transformed into the statistical information, i.e. the numbers of customers per a time interval of the schedule.

Many companies have internal user (customer, employee) profiles that can be used to extend Presence Patterns for particular business purposes. For example, if a user profile contain fields `User, Gender, Address, Date of birth`, then the event log and the customer profile can be used to derive the `NumberOfPresent` people leaving in a given city, or of a given gender, or of a given age. The example below extends a Presence Pattern in order to indicate the presence of women:

```
Presence Pattern = (Days, Times, NumberOfPresent, Gender, Locations,
Courses, Teachers, function-times, function-location, function-course,
function-teacher, function-presence).
```

- `Gender = {Woman};`
- `NumberOfPresentGender` is a finite natural number (positive integer number or zero).
- `function-presence: NumberOfPresentGender→ (Times → Days)` assigns the numbers of present woman to the time intervals for each day of the week.

In the extended form, a Privacy Pattern contains information about a subset of entities from a private universe (women, students following a given education, cars, citizens of a chosen city), that are present at a given time at a given place. Figure 1 shows a Presence Pattern for a company extended with the information about present entities for a chosen home-city. If the number of entitles from a City is less than a predefined k number, the entities can be re-identified. Ideally, a released Presence Pattern should not contain attributes of the quasi-identifiers that are related to less than `k` entities in the chosen profile.

```
Presence Pattern = (Days, Times, City, NumberOfPresentFromCity,
function-times, function-presence-from-city)
```

- `City = {Uden}`.
- `NumberOfPresentFromCity` is a finite natural number (positive integer number or zero).
- `function-presence-from-city: NumberOfPresentFromCity→(Times→Days)` assigns the numbers of customers to the time intervals for each day of the week.

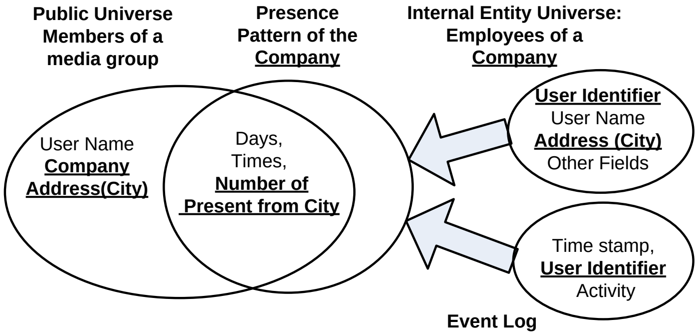
Public Universe Members of a media group
Presence Pattern of the Company
Internal Entity Universe: Employees of a Company
User Name
Company
Address(City)
Days,
Times,
Number of Present from City
User Identifier
User Name
Address (City)
Other Fields
Time stamp,
User Identifier
Activity
Event Log

Fig. 1. Quasi-identifier in a Presence Pattern

4 Method for Privacy Analysis of Presence Patterns

Now we use the definition of privacy property via k-anonymity discussed in section 2 and our definition of a family of Presence Patterns given in section 3 to propose a method for privacy analysis of Presence Patterns.

The value of `k` for `k`-anonymity is defined in advance, depending on the importance of privacy. A general 'rule of thumb' should be that an individual person could be determined from this k-subset with the effort more significant to the perpetrator compared to the privacy violation benefit.

Our method uses the following assumptions about public data:
(1) Publicly available data structures may contain data fields with private information with the same data types as the company Entity Profiles used for design of extended Presence Patterns;
(2) Publicly available data structures do not contain event logs of any other company-related activities of any person.

Simple Presence Pattern does not have any privacy issues. A simple Presence Pattern `SPP` consists of the time schedule fields and one field, `NumberOfPresent`. The values of this one field are derived from the `event log that is not public`. This field, `NumberOfPresent`, does not identify any group from the Entity Profile of the counted Entities.

Privacy Analysis of Extended Presence Patterns. Let an Analysed Presence Pattern `APP` consist of the same schedule fields as the Simple Presence Pattern `SPP`, field `NumberOfPresent` and a set `E` of some other fields.

1. Initially, the set of fields for privacy analysis contain field `NumberOfPresent`: `SF=`$\{$`NumberOfPresent`$\}$. The set of subsets causing privacy violation is empty, `SV=`$\emptyset$.
2. Add a set `E` of fields in `APP` to `SF`, `SF=SF`$\cup$`E`.

3. For each subset `sf` of `SF`, `sf`$\subseteq$ `SF`:
   - analyse if `sf` is a quasi-identifier with less than `k`-anonymity for the Internal profiles used by the company to create `APP`; [5]
   - Optional: analyse if `sf` is a quasi-identifier with `k`-anonymity for the public profile, relevant for the the Presence Pattern.
   - If YES, then add quasi-identifier `sf` to `SV: SV=SV`$\cup$`sf`.
4. If `SV`$\neq \emptyset$, then the release of this Presence Pattern can cost privacy violation, otherwise the Presence Pattern respects privacy and can be released.
5. End of privacy analysis.

This procedure gives a systematic way for relating the attributes of a Presence Pattern with the attributes of internal and public profiles.

4.1 Examples of Privacy analysis

Let us consider a case study. A company would like to optimize the use of the parking space in its premises. The entrance to the parking places has an automatic licence plate recognition system. The following internal (not public) data is available:

- an internal log of all cars that arrive to and leave from the parking. The log records include a time stamp and a licence plate number.
- an internal database of cars of employees. Any record in this database presents an employee's name and his(her) car licence plate number.
- the database of all employee's profiles. Among other fields, it contains the address of each employee (Figure 1).

The possibly relevant to this use case public information includes

- the LinkedIn (or other professional social network) profiles of employees of the company;
- The telephone guide, each record of which includes a person name, his(er) phone number(where the city can be identified);
- The public registry of cars containing the licence plate numbers with the corresponding brand and the model of the the car, engine type, etc.[6]

In the investigation of the use of the parking space, the company creates new public data in the form of a series of Presence Patterns.

1. First, the company derives and makes public the Simple Presence Pattern with the `Number Of Present Cars at the Parking Place` during the working hours and outside working hours. This pattern does not have any privacy issues as the `Number Of Present Cars at the Parking Place` is not related to any restricted set of personal data. The persons using the parking place cannot be identified.

[5] Remember, that we assume that public data structures available on the Internet may contain the data fields with private information with the same (or convertible) data types as the User (Entity) Profiles used for design of extended Presence Patterns.

[6] This kind of data is publicly available in many countries (for example the RDW registry in the Netherlands).

2. Next, the company desires to analyze how many electric cars are parked daily parking in order to arrange the necessary amount of charging sockets. The internal database of coming cars has to be extended with the information about the type of the engine in order to be able to identify cars with electric engines. The presence pattern is extended with the indicator (or data type) `ElectricOrNot`. This information can be derived from the public car registry using the license plate number. The `Number Of Present Electric Cars at the Parking Place` during and outside the working hours is presented by the new presence pattern.

   This new presence pattern also does not identify persons as publicly available car registries normally does not contain the information related to the car owner.
3. Third, the company decides to stimulate its employees to use other means of transportation instead of cars, in order to reduce the parking space and air pollution.

   The employees who live nearby the company premises, unlike the ones who live far away, may be convinced to avoid using cars. To be able to monitor its anti-pollution effort, the company needs to know how many distantly leaving employees are present during and outside the working hours.

   The pattern of presence is extended accordingly with the `City` taken from the addresses employees. So, the new pattern of presence shows the `Number Of Present Electric Cars at the Parking Place` from each `City`.

   This pattern has a possible privacy violation if just one or a small number of persons (less than a predefined `k`) lives in a particular city.

   The quasi-identifier is (company, city) (Figure 1). From this pattern, one can derive new information, namely, a pattern of presence of persons working in the given company and living in the given city. This can be done using the combination of the public LinkedIn-like profiles of employees of the company and the public telephone guide.

These examples show the need of privacy analysis. The check of privacy may be done on the data structures used for creating Presence Patterns, e.g, in our example, these data structures were employees' profiles and the public car registry.

5 Conclusion and Future work

Privacy by design has been proposed [**?**] as seven fundamental principles: privacy as proactive, privacy by default, privacy embedded into design, full functionality (respecting other aspects of the system), end-to-end life cycle protection, respect for user privacy.

Respecting the principle 'privacy as proactive', we have proposed a proactive method for analysis of Presence Patterns released by data analytics. Our method is build on (1)our inductive definition of a family of Presence Patterns that reveals relations with open data and (2) the refinement of privacy by `k`-anonymity.

Presence Patterns are widely used by the television and radio companies, team sports [**?**], trends in tourism [**?**], voting sites and education institutions [**?**,**?**].

In the future work, we plan to provide evaluation of our method for pattern definition and privacy analysis for publicly available released Presence Patterns and other released structures derived from internal data of different organizations.

References

1. Aziz, N.L.A., Aizam, N.A.H.: A survey on the requirements of university course timetabling. pp. 236–241. World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering (2016)
2. Cavoukian, A.: Privacy by design: the definitive workshop. A foreword by Ann Cavoukian, Ph.D. pp. 247–251. Springer (2010)
3. Chew, M., Balfanz, D., Laurie, B.: (Under) mining privacy in social networks. Citeseer (2008)
4. Ganji, F., Budzisz, Ł., Debele, F.G., Li, N., Meo, M., Ricca, M., Zhang, Y., Wolisz, A.: Greening campus wlans: Energy-relevant usage and mobility patterns. pp. 164–181. Elsevier (2015)
5. Griffin, P.S.: Girls' participation patterns in a middle school team sports unit. pp. 30–38. Journal of Teaching in Physical Education (1985)
6. Gross, R., Acquisti, A.: Information revelation and privacy in online social networks. In: Proceedings of the 2005 ACM workshop on Privacy in the electronic society, pp. 71–80 (2005)
7. Hoepman, J.H.: Privacy design strategies. In: IFIP International Information Security Conference, pp. 446–459. Springer (2014)
8. Kim, H., Cheng, C.K., O'Leary, J.T.: Understanding participation patterns and trends in tourism cultural attractions. pp. 1366–1371. Elsevier (2007)
9. Korolova, A., Motwani, R., Nabar, S.U., Xu, Y.: Link privacy in social networks. In: Proceedings of the 17th ACM conference on Information and knowledge management, pp. 289–298 (2008)
10. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 29–42 (2007)
11. Mulhanga, M., Lima, S., Carvalho, P.: Characterising university wlans within eduroam context. pp. 382–394. Springer (2011)
12. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: Security and Privacy, 2009 30th IEEE Symposium on, pp. 173–187 (2009)
13. Su, T.A., Ozsoyoglu, G.: Controlling FD and MVD inferences in multilevel relational database systems. pp. 474–485. IEEE (1991)
14. Sweeney, L.: k-anonymity: A model for protecting privacy. pp. 557–570. World Scientific (2002)