VAULT: A Privacy Approach towards High-Utility Time Series Data

Christoph Stach

Institute for Parallel and Distributed Systems
University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
Email: Christoph.Stach@ipvs.uni-stuttgart.de

***Abstract*—While the *Internet of Things* (*IoT*) is a key driver for *Smart Services* that greatly facilitate our everyday life, it also poses a serious threat to privacy. Smart Services collect and analyze a vast amount of (partly private) data and thus gain valuable insights concerning their users. To prevent this, users have to balance service quality (i. e., reveal a lot of private data) and privacy (i. e., waive many features). Current IoT privacy approaches do not reflect this discrepancy properly and are often too restrictive as a consequence. For this reason, we introduce *VAULT*, a new approach for the protection of private data. VAULT is tailored to *time series data* as used by the IoT. It achieves a good tradeoff between service quality and privacy. For this purpose, VAULT applies five different privacy techniques. Our implementation of VAULT adopts a Privacy by Design approach.**

***Keywords*–Privacy; Time Series; Projection; Selection; Aggregation; Interpolation; Smoothing; Information Emphasization; Noise.**

I. Introduction

The ever-increasing popularity of the *Internet of Things* (*IoT*) is both, a blessing and a curse. On the one hand, sensors built into everyday objects enable to monitor entities (e. g., a machine or a person) permanently and very precisely. Since the gathered data are always tagged with a time stamp, the data of different sources can be combined to obtain a comprehensive chronological profile of the monitored entity. Subsequent analyses can provide even more profound knowledge about the entity. The IoT is therefore an enabler for *Smart Services* from a wide variety of domains, including *Smart Homes*, *Smart Cars*, and *Smart Health*. Such services are a great benefit for the users as they facilitate their daily life [1].

On the other hand, these great capabilities of such services pose a great danger at the same time. In particular, if the monitored entity is a natural person, his or her privacy is at risk. Users are often not even aware of the coherences between gathered data and insights derivable them. However, Smart Services not only have access to the data of a single user but to the data of a vast number of users. This even enables them to learn from the behavior of these users and to predict future behavior patterns of different users [2].

For this reason, the *General Data Protection Regulation* of the EU (*GDPR*, see [3]) tries to provide guidance to meet the interests of both, service providers (in terms of data quality) and users (in terms of privacy requirements) [4]. Nevertheless, the user is faced with the difficult task of balancing service quality and privacy. The more data a user shares with a service, the better is its service quality, as it is thereby able to perform more precise analyses and thus establish a more profound knowledge base. Its users, however, are fully exposed in the process. Whereas, if a user conceals all data that could reveal private information, his or her privacy is protected effectively—yet, the service is practically useless as a result [5].

Today's privacy approaches for the IoT contribute little to solve this dilemma, as they suffer from three critical flaws. *a)* Users are often overwhelmed by these approaches, as the coherences between gathered data and derivable knowledge are not comprehensible. That is, if the user grants a service access to two seemingly harmless data sources, the combination of these two sources might provide new insights. *b)* These privacy approaches completely ignore service quality. They focus solely on concealing certain, possibly private data, and as a result the service quality is often considerably, yet unnecessarily impaired. *c)* These privacy approaches are only applicable to certain application scenarios and analysis methods. As a result, users need a variety of different privacy solutions to make all of their Smart Services privacy-aware.

To this end, we make the following three contributions: **(1)** We introduce a pri**v**acy **a**pproach towards high-**u**ti**l**ity **t**ime series data, called *VAULT*. VAULT is a concept for the protection of personal data, which achieves a good compromise between service quality and privacy and optimizes both of these aspects. Furthermore, specifying privacy requirements is still very simple for the user. **(2)** We present five different privacy techniques that are applied in VAULT. These techniques are tailored to the analysis methods applied to time series data as Smart Services mainly handle such data. **(3)** We describe an implementation of VAULT based on *InfluxDB* [6]. Yet, VAULT is completely independent from its data source, i. e., InfluxDB can be replaced by any data source providing time series data.

The remainder of this paper is as follows: In Section II, we introduce a sample use case from the *Ambient Assisted Living* (*AAL*) domain. Using this example, we identify requirements a privacy system has to meet in order to be effective for Smart Services. Section III discusses whether the related work meets these requirements. We introduce our concept for VAULT and the applied privacy techniques in Section IV. An implementation of this concept is given in Section V. In Section VI, we assess VAULT according to our identified requirements. Finally, Section VII concludes this paper.

II. Running Example

An application field, in which the IoT facilitates the users' daily routines by having access to highly sensitive data, is the healthcare domain. Sensors enable patients to monitor

© 2019 IARIA. This is the author's version of the work. It is posted at https://opencms.uni-stuttgart.de/fak5/ipvs/departments/as/publications/stachch/securware_19_vault.pdf by permission of IARIA for your personal use. Not for redistribution. The definitive version was published in *In: Rass, S. and Yee, G. (Eds.) Proceedings of the Thirteenth International Conference on Emerging Security Information, Systems and Technologies*, pp. 41–46, 2019.

themselves permanently, while their physicians and other parties involved obtain the processed data tailored to their requirements. In the following, we illustrate this using an AAL use case.

Due to an aging population, the *World Health Organization* has introduced the paradigm of *active ageing* to enable elderly people to remain involved in social life. A key aspect in this respect is, that they are not pulled from their familiar surroundings (e. g., by accommodating them in a care facility) and that there is no loss of autonomy. AAL achieves this via sensors acting as permanently present but invisible caregivers [7].

An AAL platform offers wide-ranging monitoring services. Special metering devices are capable to monitor medical data continuously (e. g., blood glucose or weight). Physicians are informed about them and are then able to adjust the medication remotely. For some of these health parameters, they require the chronological progression with high accuracy (e. g., blood glucose), while for others an approximate progression is sufficient and single values are negligible (e. g., weight). It is also possible to check remotely, whether the required medication has been taken. Yet, this information is not required to be transferred permanently. It is sufficient to inform physicians if the medicine is not taken several times in a row. Fall detection is realized via wearables. This enables to alert a caregiver immediately if a senior has fallen and needs help. For this purpose, the data from the gyroscope, the accelerometer, and the position sensor are analyzed. In addition, the location where the fall occurred has to be determined, e. g., if the "fall" occurred in bed, it may have been a false alarm and the senior just went to sleep. Although location data has to be analyzed for this purpose, the caregiver must not be allowed to access this data. However, relatives with guardianship should be informed of the senior's whereabouts (e. g., if s/he is suffering from dementia and wander around confused and disoriented) [8].

This example illustrates that Smart Services gather a variety of private data. The GDPR must thus be observed in such use cases [9]. For instance, it requires *data minimization* [Art. 5(1)(c)]. Caregivers only have to be informed when a senior has fallen, whereas permanent access to the his or her location is not required for them. Yet, relatives need access to this data, if they are the senior's guardian. This is regulated by the *purpose limitation* [Art. 5(1)(b)]. Service providers have to ensure the *accuracy* of the processed data [Art. 5(1)(d)]. To make this feasible, privacy measures must not arbitrarily manipulate sensor data. Especially when particularly sensitive data, such as health data, is involved, the data subject must give *explicit consent* to their processing [Art. 9(2)(a)]. A solution with respect to these legal obligations is given in Article 25: Technical measures are postulated to ensure privacy compliance, i. e., Smart Services monitor and regulate themselves by default (*Privacy by Design*). To be effective, such a technical privacy solution has to meet the following five requirements:

- **R_1 Individual Privacy Enhancement.** Each user has different privacy requirements. While some people have no concerns about sharing their location data, others consider this kind of data as highly sensitive. Thus, every user has to be able to decide individually what information s/he wants to reveal, i. e., make available to a service.
- **R_2 Utility Preservation.** However, not only privacy requirements need to be considered. Users also have to decide which services they want to use and what data the respective service requires in order to operate. Only if the service receives these data in a sufficient accuracy and quantity, the user receives the expected service quality.
- **R_3 Privacy and Data Quality Harmonization.** Privacy and service quality, however, are by no means independent objectives. Enhancing privacy significantly impairs service quality and vice versa. A privacy system therefore has to consider both aspects equally to achieve *Pareto optimality*.
- **R_4 Privacy Method Adaption.** To make this possible, a privacy system has to be able to adapt its privacy methods to the service quality requested by a user. That is, the privacy system has to select a method which matches a service's specific data quality and quantity requirements.
- **R_5 Dynamic Policy Application.** The application of the privacy requirements has to be dynamic, i. e., before a service gets access to data, its properties must be checked (e. g., a relative only gets access to a senior's location if s/he is his or her guardian at the time of the request).

III. Related Work

In the following, we review current privacy approaches for the IoT and assess them with regard to our running example.

Access Control: The most basic approach to ensure privacy is access control. In *role-based access control*, each involved party is assigned to a specific role (e. g., physician). A party can be assigned to several roles at the same time. Access rights to certain data sources are granted to these roles instead of individual users. Although this approach sounds promising at first as there are few roles (compared to the number of parties), and thus the number of access rights which have to be specified is reduced, it is not flexible enough for the IoT due to its fixed pre-defined roles [10]. Assigning access rights to certain attributes is significantly more dynamic. *Attribute-based access control* validates any kind of attribute at runtime (e. g., attributes that describe the party requesting data access or that party's current context). Data access is only granted if these attributes meet the data subject's authorization requirements [11]. This way, it is possible to model that relatives only have access to a senior's location data if they currently have the guardianship.

Nevertheless, pure access control approaches are far too restrictive and thus severely limit service quality. The user can only make a binary decision—either s/he grants or denies access to a data source. A fine adjustment, however, is not possible (e. g., reduce accuracy of the data or add mock data).

Attribute-based Privacy: To address this problem, a filter can be integrated into a data source. So, particular attributes of the data provided by that source can be filtered out, if they reveal private information. This enables users to specify, e. g., that their medical metering device still provides access to their blood glucose level, but not the blood oxygen level. Each filter can optionally be linked to a *spatiotemporal context* to specify when it should be active [12]. Such a filter can also be tailored to the respective data source. Instead of fully filtering out certain attributes, they can be replaced by mocked but realistic data, in terms of, e. g., value range and distribution [13].

A fundamental problem of these approaches is that they do not take chronological aspects inherent in this kind of data into account. Often, isolated data values do not pose a privacy threat. Only a sequence of single values results in a privacy-relevant pattern (e. g., a sequence of singular gyroscope and acceleration data results in an activity pattern). Yet, users have to filter all data of the concerning attribute in these approaches

to ensure that such patterns are concealed. As a result, services depending on this type of data become non-functional.

Pattern-based Privacy: The intent of pattern-based privacy approaches is to conceal complex private information from a Smart Service without unnecessarily restricting its service quality. For this purpose, *Complex Event Processing* (*CEP*) is used. In CEP, no individual sensor values are considered, but higher-order events represented by a sequence of values within a given time window [14]. For instance, the event "senior leaves home" is a sequence of location data representing a motion vector heading away from the house. That way, users specify *private patterns* that must not be revealed and *public patterns* that are critical in terms of service quality. CEP is able to recognize these patterns and then private patterns are concealed by chronologically reordering some of the sensor values. A utility metric identifies the best permutation in terms of maximizing both, privacy and service quality [15].

Pattern-based privacy approaches are therefore particularly effective for maximizing service quality. They can also conceal patterns of any complexity consisting of sequences of individual values. However, such an approach is ineffective with respect to the principle of data minimization. By reordering, all individual values are still sent to the Smart Service. As it is known what kind of information is required by the service (via the public patterns), data could be pre-processed accordingly (e. g., by aggregating or tampering it) without affecting its service quality. For instance, to detect the pattern "senior leaves home", a Boolean statement whether this event occurred is sufficient—the whereabouts prior to this event are not required. Yet, this is not considered by pattern-based privacy approaches.

Statistical Privacy: Differential privacy is applicable to the IoT, e. g., in the context of *Smart Grids* [16]. There, data remains on each user's *Smart Meter*, while energy suppliers only receive aggregated data. It is ensured that no information about an individual user can be derived from the statistical analysis of this data. Yet, this kind of anonymization is only useful when information about a large group of users is required. It is not applicable to a use case like AAL, as in such a scenario sensor data must be evaluated for each user individually.

IV. VAULT Concept

Our review of related work shows that none of these approaches is by itself effective in ensuring both, privacy and service quality. So, we combine and extend these concepts to provide a privacy concept that is tailored to IoT time series data, called VAULT. Figure 1 shows its concept and workflow.

To ensure service quality, a service has to define its quality requirements **(1)**. These include, e. g., which data a service requires and with what accuracy these data are required. Thus, the quality requirements correspond to the basic idea of the public pattern. In addition, a service description is mandatory that identifies the service, e. g., the service name, its execution environment, or the service owner **(1)**. This description is used to authenticate to VAULT. Like attribute-based access control, permissions in VAULT are not linked to a specific service, but to a set of its attributes. For instance, different permissions may apply to the same service depending on the country where it is hosted. The data subjects specify which permissions are assigned **(2)**. To this end, s/he provides a high-level description of his or her privacy requirements in natural language. Similar to the privacy patterns, s/he only has to describe which knowledge

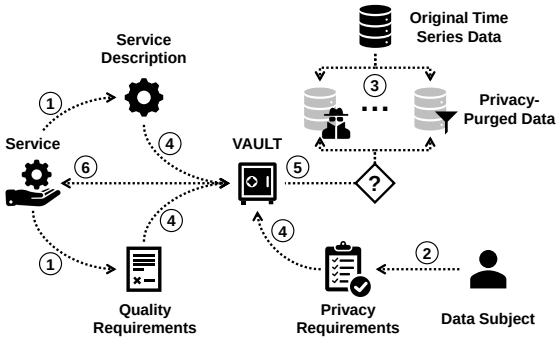

Figure 1. Concept of and Workflow for Data Access via VAULT.

must not be disclosed. A model in VAULT indicates from which data this knowledge can be derived (e. g., *ACCESSORS* [17] can be used to model these correlations). Based on this model, machine learning can automatically derive permissions from these privacy requirements [18]. As VAULT provides different privacy techniques depending on the respective service (i. e., in accordance with its quality and privacy requirements), the time series data has to be initially prepared accordingly **(3)**. **(1)** to **(3)** are independent tasks and can be carried out in any order.

If a service requests data access, VAULT first checks its service description (i. e., attributes of the service) and which permissions (i. e., privacy requirements) are linked to it. They are then consolidated with its quality requirements **(4)**. Based on these two requirement specifications, an appropriate VAULT privacy technique is selected **(5)**. Subsequently, the request is executed, and the results are sent back to the service **(6)**.

VAULT relies on existing techniques, which are already used for processing and analyzing time series data, to ensure privacy. As a result, the impact on service quality should be negligible. We discuss the following five privacy techniques:

Projection, Selection, and Aggregation: The most basic privacy technique used in VAULT is the application of relational algebra operators. A *projection* constrains the number of attributes whereas a *selection* filters out certain tuples of a data source entirely. As the data sources we consider in VAULT provide time series data, a selection operator is therefore synonymous with specifying a specific time frame. An *aggregation* can be used to consolidate the analyzed data (e. g., via set operators such as *average* or *sum*). Smart Services use these operators anyway to select the data that is relevant to them and thus reduce the huge amount of available data. VAULT is therefore able to restrict the available data according to the quality requirements of a service via theses operators in order to ensure privacy. For instance, a service gets only access to certain sensor values, certain days, or summarized data.

Data Interpolation: When dealing with sensor data, one has to reckon that sensors occasionally deliver no or incorrect values due to technical problems. To ensure that the data are still processed correctly, strategies must be implemented to deal with these missing and incorrect readings. For this purpose, these incorrect readings have to be substituted with artificial, yet realistic data. On the one hand, *interpolation techniques* can be used to smooth the temporal progression of the values, assuming that the sensor signal describes a continuous function [19]. On the other hand, it is possible to use machine learning to make

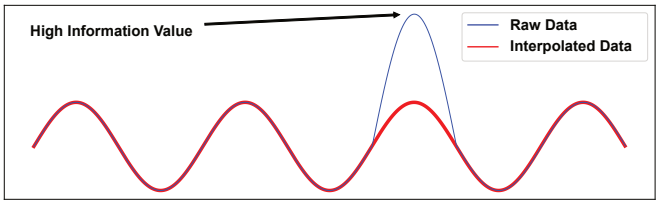

Figure 2. Application of a Spline Interpolation to Time Series Data.

predictions regarding the progression of the values. Missing values or outliers (in terms of values exceeding or falling below a threshold) can then be substituted with these predictions. We use these data cleansing techniques in VAULT to ensure privacy. In certain situations, outliers have a particularly high information value and are therefore considered as particularly sensitive data. Figure 2 shows the time course of a senior's whereabouts indicated as the distance to his or her home (blue line). S/he walks the same distance every day. One day, however, s/he changes this routine, which is a decisive information. For instance, if a service only needs to monitor that a senior takes a walk every day, VAULT first uses outlier detection to identify data points with high information value, deletes them, and then fills the resulting gap via *spline interpolation* (red line).

Data Smoothing: While data interpolation is well-suited for eliminating a few isolated outliers, sensor data can also be noisy as a total. Analyzing noisy data is often difficult and leads to poor results. So, the noise component is removed from the data by means of *filters*. Especially if the examined data contains some periodicity, which is often the case with AAL data due to regular daily routines, *Fourier transforms* are well-suited for noise reduction. This creates a band filter effect, i. e., certain interference frequencies can be attenuated [20]. Figure 3 shows the effect of a *Discrete Cosine Transform* on a noisy signal (blue line). The output is a smoothed signal (red line). However, this data cleansing method can also be used to protect private data. The transform removes details from the time series data and less information is shared with requesting services. Nevertheless, the actual data progression is still available to them with great accuracy.

Information Emphasization: Using wavelet transform, noise can even be filtered out to such an extent that only data with a high information value remains in the signal (e. g., peaks or turning points). For this purpose, the data progression is compared with a basic function, the so-called *wavelet*. This *window function* defines the weighting of each signal value in subsequent analyses. The *Continuous Wavelet Transform* constantly varies the parameters of this *mother wavelet* to obtain a band of *daughter wavelets*. This facilitates a particularly selective filtering and compression of the data [20]. In Figure 4,

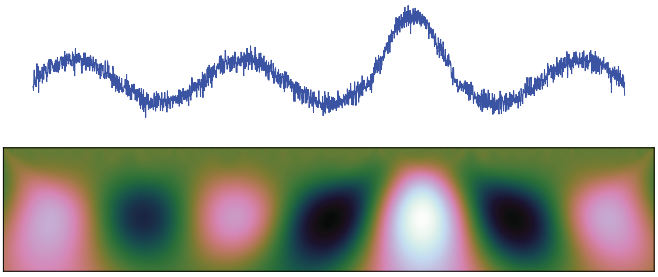

Figure 4. Time-Frequency Representation of Noisy Time Series Data.

the noisy sensor signal (upper half of the figure) is converted into a *time-frequency representation* (lower half of the figure) using the *Mexican Hat Wavelet* as mother wavelet. Relevant data segments are exposed in this representation (light and dark zones). For instance, if the signal represents blood glucose levels, these zones indicate hypoglycemia or hyperglycemia, respectively. The information about the occurrence of these events is sufficient to generate appropriate recommendations concerning medication and treatment schedule. The exact glucose values need not be disclosed to a caregiver for this purpose. This increases privacy as no details in the data are available to third parties.

Adding Noise: A completely different privacy approach is adding noise to a signal on purpose. In Figure 5, *Gaussian noise* is added to formerly noise-free sensor data (blue line). That is, the noise in the resulting data is *Gaussian-distributed* (red line). So, actual values are concealed in a set of corrupted values. Although the general data progression is still noticeable, details and characteristics of the data are hidden by the noise. For instance, activity patterns are thus still recognizable despite the noise, whereas characteristics on how a senior performs that activity are concealed. While this initially sounds like a deterioration in data quality, it can even have a positive effect on certain data analyses. For instance, noise can cause *chaotic dynamics* within data. Therefore, if *deterministic chaos* is to be expected in a data set (e. g., data on the course of a disease), but it is not noticeable as too little data are available, adding noise can be useful in this regard to improve analysis results [21].

V. VAULT Implementation

There are three implementation strategies for the realization of the VAULT concept, which are shown in Figure 6.

Query pre-processing rewrites queries before execution and adds further constraints to eliminate private information from the result set. This is well-suited for simple privacy techniques such as projection or selection. Yet, these query adaptations become complex for more advanced privacy techniques. Then,

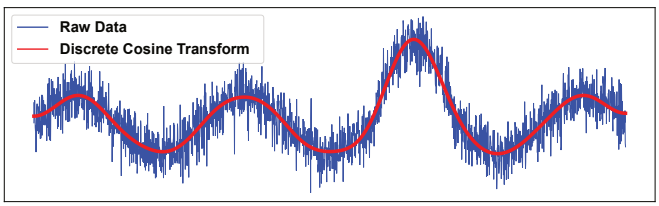

Figure 3. Application of a Fourier Transform to Time Series Data.

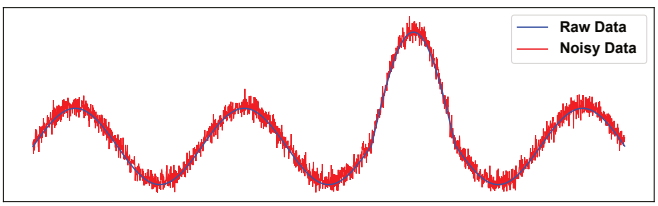

Figure 5. Adding Gaussian Noise to Time Series Data.

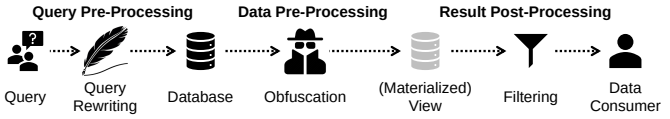

Figure 6. Implementation Strategies for the Privacy Techniques in VAULT.

errors are likely to occur when automatically rewriting queries. These errors compromise privacy as well as service quality.

Result post-processing enables a thorough control of a query's result set. That way, it can be filtered before forwarding it to the data consumer. However, a query can add hidden information to its result set. For instance, if the weight must not be revealed, a data consumer could query all data entries where the weight is x kg (without including the weight itself in the result set). Then, s/he repeats the query and increases x successively. Thus, s/he knows the weight for each entry implicitly, although it never explicitly appeared in the result set. Result post-processing is not able to detect and prevent this.

Due to the shortcomings of those strategies, we use *data pre-processing* in VAULT. This strategy pre-processes all data by removing or obscuring private data. Queries are not executed on the original data, but on this purged data. However, this data pre-processing increases the runtime. Yet, as Smart Services often use recurring queries, which are known due to their service descriptions, the runtime can be improved by using materialized views to persist the pre-processed data in advance.

Figure 7 shows how we realized the VAULT concept following the data pre-processing strategy. VAULT introduces a database abstraction layer to strictly isolate services from data sources. From a service's perspective, it therefore seems that it directly interacts with a data source and it is not aware of the privacy techniques applied to the data [22], [23].

Before using a service for the first time, it must define its quality requirements and the user must specify the privacy requirements. As this needs to be done only once (unless requirements change), these steps are not shown in Figure 7.

A registered service authenticates to VAULT with its attributes **(a)**. To prevent a service from getting too many permissions by falsifying its attributes, Gritti, Önen, and Molva [24] introduce a process for verifying these attributes. This approach takes into account that the privacy of the service has to be ensured as well, as the attributes might contain private information about the service provider. This approach is therefore a valuable supplement to the authentication process of a data provisioning platform, such as VAULT [25]. If a service is authorized to use VAULT, its queries are temporarily stored in a query buffer **(b)**. VAULT checks in the access policy which quality requirements this service has, and which permissions are granted to its attributes **(c)**. Then, a utility metric is used to search for privacy techniques that maximize both, privacy and service quality **(d)**. Basically, it compares how much information relevant to the service is concealed and how much private data is disclosed when a particular privacy technique is applied. Additionally, the user can determine via a weight, whether his or her focus is more on privacy or service quality [26]. We implemented each of the privacy techniques presented in Section IV as Python scripts. These scripts are

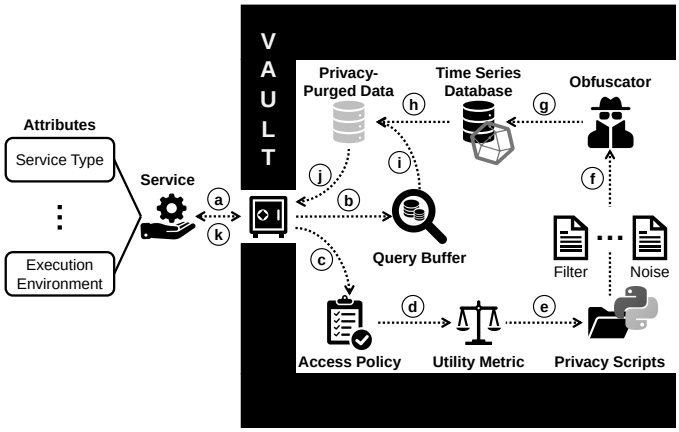

Figure 7. Implementation of and Query Processing in VAULT.

made available to VAULT in an archive. Further scripts and thus privacy techniques can be added to the archive to extend the functionality of VAULT. The utility metric selects the most suitable scripts and forwards them to the *Obfuscator* **(e)**. The Obfuscator merges the scripts and adjusts them according to the service **(f)**. It then applies the resulting script to the affected time series data **(g)**. In our prototype, we use InfluxDB. However, due to the database abstraction any other time series database can be used as well. The privacy-purged data are made available in materialized views **(h)** and the queries stored in the query buffer are executed on them **(i)**. Then, the database abstraction layer—which, in analogy to the result post-processing strategy, performs a final audit **(j)**—returns the results to the service **(k)**.

Without any loss of generality, a time series database is used in VAULT. Yet, VAULT can also be applied to a stream processing system for time series data, such as *Kapacitor* [6].

VI. Assessment

Having presented VAULT's concept and implementation, we now need to evaluate whether it meets the requirements towards a privacy system for Smart Services (see Section II).

In VAULT, each user is able to specify his or her individual privacy requirements. Since this is done in natural language and the mapping to actual data sources can be realized automatically, the configuration is also user-friendly. That way, users are enabled to specify their privacy requirements very precisely and VAULT fulfills these requirements as good as possible ($\mathbf{R_1}$).

VAULT also preserves the utility of a service when it is compatible with the privacy requirements. This is made possible by the specification of the service's quality requirements. This ensures that the service receives usable data in terms of quantity and quality. That is not the case with approaches working only with data suppression or mock data, which have a sustainably negative impact on these two parameters ($\mathbf{R_2}$).

The utility metric applied in VAULT balances privacy and quality requirements against each other and determines the best configuration. It aims to maximize both, the amount of concealed private data as well as the amount of revealed information, which is relevant to the service. As it might not be possible to maximize both of these values at the same time, at least Pareto optimality is achieved. The user can also weight, which of these objectives should be preferred by VAULT ($\mathbf{R_3}$).

To this end, VAULT provides five different privacy techniques that are tailored to IoT time series data. Each of these techniques deals with different privacy aspects. Furthermore, these techniques can be extended and combined so that a suitable technique can be found for every use case ($\mathbf{R_4}$).

In VAULT, permissions (and thus the applied privacy techniques) are not assigned to a service, but to a specific combination of its attributes. This enables a considerably more dynamic permission assignment ($\mathbf{R_5}$).

Thus, VAULT fulfills all requirements towards a privacy system for time series data as processed by Smart Services.

VII. Conclusion

The tremendous progress that IoT-enabled devices have made in recent years in terms of computing power, transmission speed, and sensor technology provides the technical foundation for a wide range of IoT applications. Such Smart Services affect all aspects of our daily lives (e. g., Smart Homes, Smart Cars, and Smart Health). In order to enjoy the benefits of these services, however, users have to disclose a lot of data, some of which revealing highly sensitive information. However, current privacy approaches are not adapted to the specific characteristics of time series data as processed by Smart Services, making them unnecessarily restrictive. As a result, users have to disclose too much private information in order to prevent that the service quality deteriorates too much.

In this paper, we therefore introduce VAULT, a new privacy concept for time series data. If data are queried by a service, VAULT considers besides privacy requirements also quality requirements of this service towards the data. This includes, among other things, what data is required, what accuracy this data must have, and how the data is pre-processed by the service. VAULT then selects a privacy technology fitting to this pre-processing. For instance, projection, selection, and information emphasization are suitable for data reduction, whereas data interpolation and data smoothing can be used as noise filters or for outlier suppression. Thus, VAULT can find a good ratio between privacy and service quality. In our prototype, five privacy techniques are implemented as Python scripts. However, these scripts can be combined, and more scripts can be added if needed. As a result, the service quality can be increased for any type of service and the privacy can be enhanced. VAULT can be applied to time series databases (e. g., InfluxDB) as well as stream processing systems for time series data (e. g., Kapacitor). That is, VAULT meets the request of the GDPR for a manageable Privacy by Design solution for the IoT.

As part of future work, the performance of the VAULT prototype has to be evaluated thoroughly in terms of processing time and data throughput.

Acknowledgment

This paper is part of the PATRON research project, which is financed by the Baden-Württemberg Stiftung gGmbH.

References

[1] M. S. Jalali, J. P. Kaiser, M. Siegel, and S. Madnick, "The Internet of Things Promises New Benefits and Risks: A Systematic Analysis of Adoption Dynamics of IoT Products," *IEEE Security Privacy*, vol. 17, no. 2, pp. 39–48, 2019.

[2] Q. Pan, "Privacy in the New Age of IoT," in *Women Securing the Future with TIPPSS for IoT*, F. D. Hudson, Ed. Springer, 2019, pp. 37–52.

[3] European Parliament and Council of the European Union, "Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC," Legislative acts L119, 2016.

[4] S. Wachter, "Normative Challenges of Identification in the Internet of Things: Privacy, Profiling, Discrimination, and the GDPR," *Computer Law & Security Review*, vol. 34, no. 3, pp. 436–449, 2018.

[5] K. M. Ramokapane, A. C. Mazeli, and A. Rashid, "Skip, Skip, Skip, Accept!!!: A Study on the Usability of Smartphone Manufacturer Provided Default Features and User Privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 2, pp. 209–227, 2019.

[6] InfluxData Inc. (2019). "InfluxDB: Purpose-Built Open Source Time Series Database," [Online]. Available: https://www.influxdata.com.

[7] A. Dohr, R. Modre-Osprian, M. Drobics, D. Hayn, and G. Schreier, "The Internet of Things for Ambient Assisted Living," in *ITNG '10*, 2010, pp. 804–809.

[8] E. Borelli *et al.*, "HABITAT: An IoT Solution for Independent Elderly," *Sensors*, vol. 19, no. 5, pp. 1–23, 2019.

[9] E. Thorstensen, "Privacy and Future Consent in Smart Homes as Assisted Living Technologies," in *ITAP '18*, 2018, pp. 415–433.

[10] Y. Ning, Y. Zhu, R.-c. Wand, R. Malekian, and L. Qiao-min, "An Efficient Authentication and Access Control Scheme for Perception Layer of Internet of Things," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, pp. 1617–1624, 2014.

[11] M. Hüffmeyer and U. Schreier, "Analysis of an Access Control System for RESTful Services," in *ICWE '16*, 2016, pp. 373–380.

[12] K. Olejnik *et al.*, "SmarPer: Context-Aware and Automatic Runtime-Permissions for Mobile Devices," in *SP '17*, 2017, pp. 1058–1076.

[13] S. Alpers *et al.*, "PRIVACY-AVARE: An approach to manage and distribute privacy settings," in *ICCC '17*, 2017, pp. 1460–1468.

[14] G. Cugola and A. Margara, "Processing Flows of Information: From Data Stream to Complex Event Processing," *ACM Computing Surveys*, vol. 44, no. 3, 15:1–15:62, 2012.

[15] S. M. Palanisamy, F. Dürr, M. A. Tariq, and K. Rothermel, "Preserving Privacy and Quality of Service in Complex Event Processing Through Event Reordering," in *DEBS '18*, 2018, pp. 40–51.

[16] K. Birman, M. Jelasity, R. Kleinberg, and E. Tremel, "Building a Secure and Privacy-Preserving Smart Grid," *ACM SIGOPS Operating Systems Review*, vol. 49, no. 1, pp. 131–136, 2015.

[17] C. Stach and B. Mitschang, "ACCESSORS: A Data-Centric Permission Model for the Internet of Things," in *ICISSP '18*, 2018, pp. 30–40.

[18] C. Stach and F. Steimle, "Recommender-based Privacy Requirements Elicitation – EPICUREAN: An Approach to Simplify Privacy Settings in IoT Applications with Respect to the GDPR," in *SAC '19*, 2019, pp. 1500–1507.

[19] M. Pourahmadi, "Estimation and Interpolation of Missing Values of a Stationary Time Series," *Journal of Time Series Analysis*, vol. 10, no. 2, pp. 149–169, 1989.

[20] T. Sakamoto *et al.*, "A crop phenology detection method using time-series MODIS data," *Remote Sensing of Environment*, vol. 96, no. 3, pp. 366–374, 2005.

[21] L. Billings and I. B. Schwartz, "Exciting chaos with noise: Unexpected dynamics in epidemic outbreaks," *Journal of Mathematical Biology*, vol. 44, no. 1, pp. 31–48, 2002.

[22] C. Stach and B. Mitschang, "The Secure Data Container: An Approach to Harmonize Data Sharing with Information Security," in *MDM '16*, 2016, pp. 292–297.

[23] ——, "CURATOR—A Secure Shared Object Store: Design, Implementation, and Evaluation of a Manageable, Secure, and Performant Data Exchange Mechanism for Smart Devices," in *SAC '18*, 2018, pp. 533–540.

[24] C. Gritti, M. Önen, and R. Molva, "Privacy-preserving delegatable authentication in the Internet of Things," in *SAC '19*, 2019, pp. 861–869.

[25] C. Stach, F. Steimle, C. Gritti, and B. Mitschang, "PSSST! The Privacy System for Smart Service Platforms: An Enabler for Confidable Smart Environments," in *IoTBDS '19*, 2019, pp. 57–68.

[26] C. Stach, F. Dürr, K. Mindermann, S. M. Palanisamy, and S. Wagner, "How a Pattern-based Privacy System Contributes to Improve Context Recognition," in *CoMoRea '18*, 2018, pp. 238–243.