Privacy-preserving Quantified Self: Secure Sharing and Processing of Encrypted Small Data

Hossein Shafagh
Department of Computer Science
ETH Zurich, Switzerland
shafagh@inf.ethz.ch

Anwar Hithnawi
Department of Computer Science
ETH Zurich, Switzerland
hithnawi@inf.ethz.ch

ABSTRACT

The emergence of a plethora of wearables and sensing technologies has enabled non-intrusive digitization of our daily physical activities. Emerging applications utilize such data to make inferences about our physiological and health states, provide health diagnosis, and contribute to wellbeing improvements. The common approach for such applications is to collect data, either using mobile applications or special hardware, e.g., wearables, and store them on a third party storage provider. This results in many unconnected data silos of self-quantification data. Researchers and industry, advocate for a common personal storage space, to conquer the myriad of small chunks of data, deemed to be lost/forgotten in the long term. The benefits of such co-located personal data are tremendous, specifically with regards to personalized medicine, treatment, and health care. However, the centralized storage of data exacerbates the privacy and security concerns that the IoT ecosystem is facing today. In this position paper, we advocate the necessity of privacy and security guarantees for the paradigm of co-located storage of personal health data. We envision two core security functionalities: true end-to-end encryption, such that only encrypted data is stored in the cloud and secure sharing of encrypted data, without disclosing data owner's secret keys. We discuss the challenges in adopting such an end-to-end encryption paradigm while preserving the cloud's basic processing functionalities over encrypted data and how to cryptographically enforce access control.

CCS CONCEPTS

•Security, Privacy → Privacy-preserving protocols; •Human-centered computing → *Ubiquitous and mobile computing;*

KEYWORDS

IoT, Privacy, Encrypted Processing, Homomorphic Encryption, Secure Sharing

ACM Reference format:
Hossein Shafagh and Anwar Hithnawi. 2017. Privacy-preserving Quantified Self: Secure Sharing and Processing of Encrypted Small Data. In *Proceedings of MobiArch '17, Los Angeles, CA, USA, August 25, 2017,* 6 pages.
DOI: 10.1145/3097620.3097625

MobiArch '17, Los Angeles, CA, USA

978-1-4503-5059-4/17/08…$15.00
DOI: 10.1145/3097620.3097625

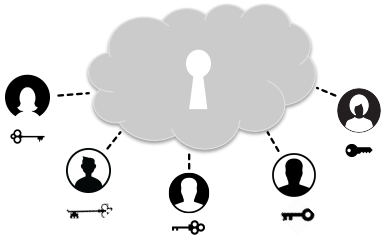

Figure 1: Sharing and processing of encrypted data

1 INTRODUCTION

The convergence of low-power embedded devices, wearables, and wireless networks has facilitated the emergence of novel applications and services that are changing the way we perceive and interact with the physical world. One promising application, yet with untapped potentials is self-quantification. There is a myriad of applications today collecting various types of data related to our activities and physical and mental health. Fitness and activity tracking applications, such as Fitbit, allow their user base to monitor their health and activity progress. In the long term, such longitudinal data can be utilized for a more accurate diagnosis and treatment of medical conditions. More specialized applications utilize various sensor readings to predict life-critical seizures [3], provide fertility-related information [1], or quantify personal well-being [3, 10].

The common technical ecosystem of such health-related IoT applications consists typically of a user-facing mobile app, potentially extended with specialized hardware, such as wearables, to collect special purpose sensor readings. The collected data is then stored on third-party storage services. The application logic, in the form of a back-end server, is as well mostly accommodated on third-party web service providers. The back-end server processes incoming queries from the front-end applications and sends back the results. Responsible apps and services promise high security and privacy standards, and vow not to sell user's data for revenue (or share only de-identified/anonymized data). However, currently employed security and privacy mechanisms do not provide the necessary degree of privacy. At best, transport layer security (TLS) is used which protects data in transit, but not in rest.

Research and industry alike advocate for a centralized storage of health-related data, though with different inducements. Mobile OS level solutions such as Google Fit or Apple's HealthKit provide a higher level API for application developers for data storage and a user-authorized cross-app data access. Open mHealth [17] is part of the research efforts to facilitate standardized central storage of personal health data from heterogeneous applications and services.

Open mHealth defines data storage schemas which facilitate data standardization. The co-location of such personal data allows for a persistent data storage and the realization of innovative research and services. Big data-driven models can be used to make personal predictions, for instance, related to a healthier lifestyle, medical related alerts, to name a few examples.

The OS level approaches enforce security and privacy by only storing data locally. Despite giving users control over their data, this approach lacks the support of data sharing with external entities. Open mHealth foresees common security measures, such as transport layer security and sharing based on the OAuth protocol. OAuth [26] is currently the de-facto standard for shared access to online resources. However, it is based on access policies defined by web services and does not provide any cryptographic guarantees.

In this position paper, we advocate augmenting the application ecosystem with cryptographic mechanisms that can guarantee user's control over raw data, as streamed directly from sensors, yet allow sharing and processing. We propose to augment the centralized data storage schemes with transparent end-to-end encryption and sharing features, as illustrated in Figure 1. Application developers continue using the restful API, however, these calls are intercepted by our proposed framework, residing on users personal devices. Outgoing data is encrypted with user's keys and incoming data is decrypted before being exposed to the application layer. Hence, data remains strongly encrypted in transit and at rest. The cloud component of the ecosystem is augmented such that it facilitates sharing of encrypted data and basic processing of encrypted data. The encrypted sharing embodies a cryptographic access control with an efficient access revocation. The user can share her encrypted data with applications and services, for instance for analytical services or with individuals, such as a physician, friend, or partner. In the following sections, we discuss the design space for such a framework, the associated risks and challenges, and the opportunities arising from augmenting health data with encrypted sharing and data processing capabilities.

2 BACKGROUND AND MOTIVATION

The advances in the low-power built-in mobile sensors, such as gyroscope, accelerometer, and compass have made non-intrusive and energy efficient activity recognition and tracking possible. Basic automated activity tracking is currently supported by native mobile operating systems, such as Apple's HealthKit. For advanced activity and health tracking, wearables and external devices can be employed. For instance, Fitbit as the representative of fitness tracking wearables can be equipped with up to 8 built-in sensors. Most of these sensors are commonly available on today's smartphones, except the optical heart rate monitor. More specialized wearables can be equipped with skin temperature sensors, ultraviolet sensors, capacitive sensors, and galvanic skin response. These sensors can provide more insightful data about the user, such as emotional state, fertility-related conditions, and physiological signals.

With the continuous miniaturization of low-power sensors [21], we will soon be surrounded by smartphones and wearables collecting detailed vital signs of our body, capable of providing more insights about our health than what we would be capable of perceiving. The societal and personal gains of digitizing these facades of our lives are immense. However, this resembles a double-edged sword, in that the privacy and security risks, if not addressed properly and timely, could diminish the gains. Before continuing with measures on how to provide security and privacy at design, we review an important and sensitive class of tracking applications in more details.

2.1 Applications

The potentials of applying encrypted data sharing and processing for increased level of privacy without compromising efficiency are immense, particularly as the variety and volume of data collected about individuals increase exponentially. We now discuss the field of fertility tracking apps and discuss Ava's wristband fertility tracker.

Many women nowadays rely on mobile applications (e.g., Clue [2]) to track their menstrual cycle, as well as the accompanying symptoms like mood fluctuations or headaches. These applications assist women in many ways, such as tracking the infertile phase by integrating calendar-based contraceptive methods, detecting related health issues, and most importantly predicting the time of the next period or the fertile window (i.e., ovulation), based on previous data. Such applications are essential and support women in making informed decisions. Moreover, such a detailed medical history can enable a more accurate diagnosis and treatment. However, logging such delicate and intimate information raises serious privacy issues. With wearables such data can even be collected non-intrusively. For instance, body temperature is a means to infer information about the menstruation cycle and is used in specialized wearables (e.g., Ava [1], Femometer [4]).

Ava's bracelets are worn at night during sleep and collect data about resting pulse rate, skin temperature, heart rate variability, sleep, breathing rate, perfusion, and bio-impedance. Heart rate variability is the variation in the time interval of consecutive heart rates. It is a strong indicator of physiological stress, where higher variations indicate a lower physiological stress level. Perfusion is the process of blood flow through the capillaries to tissues of the body. Besides providing information about body fat, perfusion reveals information about hydration and sweating patterns. The collected data by the various built-in sensors is stored in the cloud once the bracelet is in the vicinity of an already paired personal smartphone with an activated BLE. Ava leverages the collected data to track the menstrual cycle, predict the fertile window (i.e., most suitable time to conceive), and provides information about the quality of sleep and stress level. Ava foresees sharing of data with individuals, such as the partner, or medical experts for consultancy.

At the clinical level, wristbands can be equipped with even more sophisticated sensors. For instance, Empatica [3, 18], a novel wristband for nervous system monitoring and alerting in case of life-critical seizures integrates PPG and EDA sensors. The photoplethysmography (PPG) sensor measures blood volume pulse. This is then used to derive heart rate, heart rate variability, and other cardiovascular features. The electrodermal activity (EDA) sensor is utilized to measure sympathetic nervous system arousal, which can derive features related to stress, engagement, and excitement.

2.2 Challenges and Risks

The advancements in the cloud domain contribute to a faster pace of developing IoT applications. Specifically, ready-to-use cloud services, such as pre-image dockers, VM web servers, query and response image recognition services, have reduced the entry barrier for application developers and increased the quality of services and convenience for users. At times, where we store our data mainly on cloud service providers infrastructures, we face the emerging security and privacy risks of collecting such health data. Financial incentives fuel internal (e.g., admins) and external adversaries towards unauthorized trade with personal data. This is not only a threat towards individuals, whose data get compromised, but to the success of this new emerging ecosystem of health and activity-based services, and to users trust in them. Privacy and human rights advocates' major concerns with regards to self-quantification data are due to the higher risk of: *profiling* (exclusion or discrimination against certain types of people), *embarrassment and extortion*, and *corporate misuse* [9]. Current studies show the extent of trade with health data [43] and that the majority of smartphone health applications systematically trade user's data, with or without user consent [16]. This is not considering applications with unintentional data leakage [9]. More conscientious apps apply anonymization of data, which is mainly replacing personally identifiable information, such as social security numbers, names, and detailed addresses with random identifiers. In addition to being capable of finding the original identity with access to the random identifier's mapping, ubiquitous data-mining technologies can easily learn a previously anonymous individual's identity [43]. Hence, it is necessary to adopt cryptographic measures that give users control over their data with strong guarantees. Data owners should have full control of their data and decide with whom, what and at which granularity to share their data.

3 DESIGN SPACE AND EXPLORATION

In this section, we briefly explore main cryptographic components that can facilitate the realization of encrypted sharing with processing capabilities.

Fully Homomorphic Encryption (FHE): With FHE one can compute any arbitrary mathematic computations on encrypted data which can be reduced to a composition of addition and multiplication gates. The concept of fully homomorphic encryption was already introduced in the 70s. However, the first implementable schemes only appeared in the last decade [19]. Since then many research efforts have contributed to improving the sluggish performance of FHE. Despite considerable performance improvements in the recent years, the performance of FHE schemes is considered presently too slow for practical systems.

Partially Homomorphic Encryption (PHE): A more performant, however, less powerful approach for end-to-end encryption can be realized based on partially homomorphic encryption [30]. PHE only allows one type of computation on encrypted data, for instance, either additions or multiplications. Though this approach lacks the functionally of FHE, it is several orders of magnitude faster than FHE. Moreover, additions play an important role in the query processing domain. Combined with the computational partitioning where computation is divided between the cloud and the client, PHE can become a powerful scheme. Property-preserving encryption (PPE) schemes are computationally often as efficient as symmetric-key based encryption schemes. PPE allows the computation of comparison-related operations [32], such as ordering, min/max, and equality check over encrypted data. Since PPE schemes inherently leak information [28], these schemes are less secure and should be utilized carefully. The combination of PPE and PHE schemes has resulted into efficient encrypted query processing systems [33, 38].

Encrypted Data Sharing: Sharing of encrypted data can be achieved simply by sharing the secret key. This approach is efficient. However, it puts the shared secret at the risk of disclosure. An alternative approach for encrypted sharing, is public-key-based re-encryption (RE). With RE a user Alice can issue the cloud a cryptographic token that allows her data to be re-encrypted from under her key to encrypted under Bob's key. The cloud does not learn any keys nor the content of the encrypted data. Bob can then decrypt the shared data with his private key. It is important that the re-encryption scheme exhibit the following properties: *(i) key-private*: non of the involved parties learn the other parties private key. Thus, the re-encryption token is computed based on the public key of the other party. *(ii) uni-directional and single-use*: not permitting multiple re-encryptions to avoid uncontrolled ciphertext transformations towards an unauthorized user. With any sharing scheme, access revocation plays a vital role, as users may wish to discontinue sharing data.

4 PITFALLS AND RISKS

While building an encrypted data processing and sharing system, it is vital to be aware of the remaining weaknesses and how they impact the overall security of the system. Otherwise, we pay the price for the involved overheads, such as higher bandwidth, CPU, and memory, without significantly gaining in security. It is as well important to bear in mind that each security measure aims to create a cost-increasing barrier for the adversary. Despite the increased barriers, extraordinary or niche adversaries might still be capable of overcoming them. Hence, the security model should clearly state which type of adversaries it is addressing and which types not. In the following, we discuss typical pitfalls of practical encrypted data processing systems.

Leakage. Property-preserving encryption schemes are accompanied with data leakage. For instance order-preserving encryption, per definition reveals the ordering information among the ciphertexts. While for certain data types, the traded leakage for performance is acceptable for others it might pose the risk of complete disclosure. This is especially the case for low-entropy values. For instance, assume the heart-rate which has a range between 20 to 300 beats per minute. Encrypting heart-rate with order-preserving encryption would allow ordering of the ciphertexts and hence encrypted min/max queries. However, a histogram of encrypted heart-rate values provides the same information as of the histogram over the plaintext values. The histogram reveals certain activities using sophisticated models.

Metadata. Communication-related metadata refers to with whom and when communication took place. Researchers have shown how the communication metadata can be utilized to learn a significant

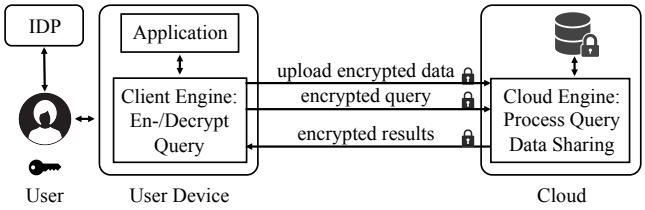

Figure 2: Design Overview. Encryption and decryption take place only on user's personal devices.

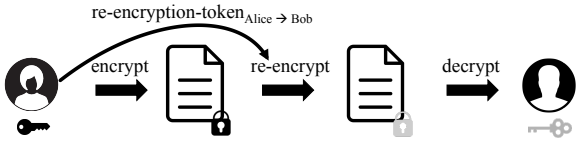

Figure 3: Sharing using re-encryption of the ciphertext without access to encryption key nor plaintext data.

amount of information [24]. This is even possible with a secure end-to-end communication, such as TLS. Metadata plays as well a major role in surveillance. For instance, evaluation of distinct packet size patterns and timings can be sufficient to identify visited web pages, watched videos, and used applications [20]. Regarding data storage, metadata refers to the storage type, database schema, and any information related to storage facilities. Storage metadata is used to launch a targeted attack or as well to support inference attacks on encrypted data processing systems with a considerable amount of leakage. For instance, Naveed et al. [28] show that with metadata information about hospital databases, they can disclose a large amount of property-preserved encrypted data.

Access Pattern. The frequency and pattern of how the database is accessed and which tables or rows are read from might intuitively appear not relevant. This might even appear to be more the case for lower system level access patterns, for instance regarding memory addresses. However, similar to the communication metadata, the access pattern information can be used to learn sensitive information [22]. For instance, in private keyword search over encrypted data, auxiliary information about the frequency of most searched words could compromise the encrypted search results.

5 DESIGN

In this section, we introduce an architectural design to accommodate for an efficient encrypted data sharing with limited processing power. Our goal is to keep user's data encrypted at all times outside user's personal devices. The computational resources available at the edge of the Internet and on our personal devices support demanding computations at fast rates. This strengthens the paradigm of computational locality, where sensitive computations can be performed on user's personal devices. This, however, necessitates private scope search and sharing capabilities on encrypted data to gain access to a specific segment of the remote data. In our design, we leverage novel cryptographic schemes to realize the desired capabilities.

We distinguish between three main players, as depicted in Figure 2: The **client** device is the personal mobile system that is equipped with several built-in sensors and capable of collecting valuable data. The client device typically hosts the application front-ends. It can as well serve as a gateway for less powerful devices, such as wearables. The **cloud** hosts the application logic and provides data storage features. The identity provider (**IDP**) plays an important role in any multi-user scenario. It verifies the identity-to-key mappings. The role of IDP can be outsourced to recent online-network-based approaches, such as Keybase [5]. They provide a provable mapping of an online identity to its public key, by utilizing prevalent social media channels and in general online accounts. To this end, a prospective user who claims an identity is asked to provide a proof of identity by publishing a unique individual token over the claimed identity channel (e.g., Twitter, Facebook, Instagram, Github, Coinbase, etc.). Hence, one can search for the public key of another user based on their known online identity.

Encrypted Data Sharing. We leverage an elliptic curve pairing-based re-encryption system (referred to as AFGH [7]) to realize the encrypted sharing, as depicted in Figure 3. A bilinear paring-based cryptosystem [12] defines two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q with the following property for $a, b \in \mathbb{Z}q$, $g \in \mathbb{G}_1$, and $h \in \mathbb{G}_2$: $e(g^a, h^b) = e(g, h)^{ab}$. The user Alice computes her public key as $pk_a = g^a$ and her private key as $sk_a = a$. Alice can issue Bob the re-encryption token based on his public key pk_b as the $Token_{a \to b} = pk_b^{1/a} = g^{b/a} \in \mathbb{G}$.

Our revocation procedure discontinues sharing of data with an individual by renewing all encryption keys. This has the consequence that previously issued re-encryption tokens become obsolete. However, old data can still be re-encrypted with obsolete tokens. This is acceptable since already shared data is considered to be in the wild (e.g., already cached). However, to prevent additional leakage of old data, we introduce an in-situ re-keying mechanism. This enables the cloud to change the encryption key of encrypted data without the need of decryption. Hence, all data is encrypted at all times with the latest keys, preventing any undesired access by revoked sharing parties.

Limited Encrypted Processing and Scope Search. Our utilized re-encryption scheme is partially homomorphic with respect to additions and multiplications with constants. We leverage this feature to perform limited computations on encrypted data (e.g., average, mean, etc.). More complex computations are executed locally, which requires the capability of searching the desired encrypted data segments and download them. To realize the scope search over structural data, it is important to be able to search over encrypted indices or time stamps. We construct an encrypted range query scheme with limited leakage based on a recent order-revealing encryption scheme [25]. The ciphertexts in this scheme are divided into two segments, referred to as left and right parts. A left part compared to any right part reveals ordering information. Hence, only the right parts are stored ordered in the cloud. Right parts are semantically secure and can not be compared with each other. Hence, offline attacks based on ordering information are not feasible anymore. While making a search query,

the left parts of the boundaries are provided to the cloud. The cloud uses the left parts to retrieve the data within the defined boundaries.

Challenges. Efficient computations on user's personal devices are vital to render our approach practical. Our sharing scheme is realized by means of a pairing-based cryptosystem, known to be computationally expensive. Our initial results show that expensive operations are mostly computed on the cloud (re-encryption) or can be amortized over time (token generation). Despite featuring an efficient encryption, which is important with regards to higher number of encryptions compared to decryptions (since more data points are generated), the decryption requires further optimizations to be competitive with symmetric-key based counterparts.

Guarantees. Our primary goals are to defy passive attacks targeted at data on the cloud as well as to prevent access of unauthorized users while enabling an efficient sharing of encrypted data. Our design fulfills these goals such that data on the cloud remains strongly encrypted (i.e., semantic security) at all times. The cloud never gains access to any decryption keys. To protect the data from unauthorized access, we cryptographically restrict data access to users with decryption keys associated with issued re-encryption tokens. With the re-encryption token, the cloud can only re-encrypt the stored data towards the authorized service. To prevent a malicious cloud from creating fake users, we rely on an IDP to verify the identity of the users. Moreover, we prevent a malicious cloud from unauthorized re-encryptions towards a malicious user. This is achieved with the one-hop re-encryption property of the underlying re-encryption scheme.

6 RELATED WORK

In the following, we discuss important directions of research related to this position paper.

Encrypted Search. Recent advancements of fully homomorphic encryption [19] have resulted into implementable schemes [15],which are however presently too slow for real world applications. Searchable encryption schemes support only a limited set of operations, but can be efficiently used in specialized domains. Song et al. [42] introduced the first encrypted search scheme for text files, where the metadata is encrypted deterministically and hence searchable. Their idea is based on deterministically encrypting the meta information of files, and hence being able to search over them. Follow-up schemes address other problems such as encrypted data de-duplication [23], deep packet inspection [39], and private network function virtualization [6]. More capable search schemes [13, 31, 33, 36, 38, 40], targeted for structured databases employ additional techniques such as partially homomorphic and order-preserving encryptions. Monomi [44] improves the performance and extends supported queries of CryptDB. In CryptDB, the application server has access to keys and carries out en-/decryptions. Hence, it can leak information if compromised. Talos [37, 38] tailors CryptDB for IoT devices and eliminates the risk of compromise due to application server. Mylar [34] introduces encrypted search over text files encrypted under multiple keys. Mylar leverages a pairing-based cryptosystem, such that given corresponding re-encryption tokens, the server re-encrypts the provided search token of a user for all group documents she is a member of (each group document is encrypted with a different key). Shi et al. [41] propose private aggregation of time series data, which blends secret sharing with homomorphic encryption. Seabed [31] introduces an additively symmetric homomorphic encryption scheme to perform large-scale aggregations efficiently. Access patterns to encrypted data still leak sensitive information about the plaintext data. Oblivious RAM approaches [35] prevent leakage of sensitive information through access patterns.

MPC. In traditional secure Multi-Party Computation (MPC) [46] private functions are computed among a set of users without the need of a trusted party. Hereby individual values from participating users are kept confidential, while the outcome can be public. This requires high interactions between users which would strain the limited resources of mobile platforms. With the rise of cloud computing, server-aided or outsourced MPC approaches have emerged. However, these schemes are either only of theoretical interest [27] or require at least two non-colluding servers, where for instance one server has only access to encrypted data and the other server has access to the keys only [29].

Trusted Computing. An orthogonal approach to encrypted computing assumes a trusted computing module on an untrusted cloud environment [8]. The data remains encrypted at rest and is only decrypted in the trusted module for computations. This approach is appealing to data center operators, due to control over hardware. However, it implies that users consider the trusted computing module trustworthy.

Re-Encryption. The idea of Re-Encryption (RE) has been initially proposed for email and cloud-sharing applications. The initial schemes [11] have the bi-directional property and are not resistant against collusion. More importantly, the parties need to exchange their private keys. The later schemes [7] address these weaknesses and are uni-directional and non-interactive. The symmetric-key RE based on Boneh et al.'s key homomorphic PRF scheme [14] lacks our required homomorphic property on ciphertexts and master secret security. Sieve [45] utilizes this scheme to provide cryptographically enforced access control for cloud data. Hence, Sieve's security model assumes no collusion between the cloud and users.

7 CONCLUSION

In this position paper, we advocate giving users control over their cloud data. We discuss the necessity of such a control that can only be achieved with true end-to-end encryption, and the need of cryptographically enforced data sharing features. We first explore the design space for composing such a system and the accompanying risks. We then lay down the design of our scheme to fulfill the encrypted sharing goal. We are currently in the process of finalizing our design and developing a reference implementation.

8 ACKNOWLEDGMENTS

We thank Friedemann Mattern for the comments on earlier versions of this paper. We are thankful to the anonymous reviewers for their valuable feedback. Moreover, we thank Lukas Burkhalter for the support with the development.

REFERENCES

[1] 2016. Ava: Fertility Tracking Bracelet. avawomen.com. (2016).
[2] 2016. Clue: Period/Ovulation Tracker. helloclue.com. (2016).
[3] 2016. Empatica. empatica.com. (2016).
[4] 2016. Femometer: Fertility Tracker. femometer.com. (2016).
[5] 2016. Keybase. keybase.io. (2016).
[6] Hassan Jameel Asghar, Luca Melis, Cyril Soldani, Emiliano De Cristofaro, Mohamed Ali Kaafar, and Laurent Mathy. 2016. SplitBox: Toward Efficient Private Network Function Virtualization. In *Workshop on HotMiddlebox*.
[7] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. 2005. Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. In *NDSS*.
[8] Summet Bajaj and Radu Sion. 2011. TrustedDB: A Trusted Hardware-Based Database with Privacy and Data Confidentiality. In *ACM SIGMOD*.
[9] Mario Ballano Barcena, Candid Wueest, and Hon Lau. 2014. *How safe is your quantified self?* Technical Report. Symantec.
[10] Liliana Barrios and Wilhelm Kleiminger. 2017. The Comfstat ? Automatically Sensing Thermal Comfort for Smart Thermostats. In *PerCom*.
[11] Matt Blaze, Gerrit Bleumer, and Martin Strauss. 1998. Divertible Protocols and Atomic Proxy Cryptography. In *EUROCRYPT*.
[12] Dan Boneh and Matthew K. Franklin. 2001. Identity-Based Encryption from the Weil Pairing. In *CRYPTO*.
[13] Dan Boneh, Craig Gentry, Shai Halevi, Frank Wang, and David J. Wu. 2013. Private Database Queries Using Somewhat Homomorphic Encryption. In *Applied Cryptography and Network Security (ACNS)*.
[14] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. 2013. Key Homomorphic PRFs and Their Applications. In *CRYPTO*.
[15] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2012. (Leveled) Fully Homomorphic Encryption Without Bootstrapping. In *Innovations in Theoretical CS Conference*.
[16] Stuart Dredge. 2013. Yes, those Free Health Apps are Sharing your Data with other Companies. Guardian, Online: theguardian.com/technology/appsblog/2013/sep/03/fitness-health-apps-sharing-data-insurance. (2013).
[17] Deborah Estrin and Ida Sim. 2010. Open mHealth Architecture: an Engine for Health Care Innovation. *Science* 330, 6005 (2010), 759–760.
[18] Maurizio Garbarino, Matteo Lai, Dan Bender, Rosalind W Picard, and Simone Tognetti. 2014. Empatica E3 - A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. In *Mobihealth*.
[19] Craig Gentry. 2009. Fully Homomorphic Encryption Using Ideal Lattices. In *ACM Symposium on Theory of Computing (STOC)*.
[20] Ben Greenstein, Damon McCoy, Jeffrey Pang, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. 2008. Improving Wireless Privacy with an Identifier-free Link Layer Protocol. In *MobiSys*.
[21] Anwar Hithnawi, Hossein Shafagh, and Simon Duquennoy. 2015. TIIM: Technology-Independent Interference Mitigation for Low-power Wireless Networks. In *ACM Conference on Information Processing in Sensor Networks (IPSN)*.
[22] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. 2012. Access Pattern Disclosure on Searchable Encryption: Ramification, Attack and Mitigation. In *NDSS*.
[23] Sriram Keelveedhi, Mihir Bellare, and Thomas Ristenpart. 2013. DupLESS: Server-Aided Encryption for Deduplicated Storage. In *USENIX Security*.
[24] David Lazar and Nickolai Zeldovich. 2016. Alpenhorn: Bootstrapping Secure Communication Without Leaking Metadata *(USENIX OSDI)*.
[25] Kevin Lewi and David J Wu. 2016. Order-Revealing Encryption: New Constructions, Applications, and Lower Bounds. In *ACM CCS*.
[26] Torsten Lodderstedt, Mark McGloin, and Phil Hunt. 2013. OAuth 2.0 Threat Model and Security Considerations. *IETF, RFC 6819* (January 2013).
[27] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. 2012. On-the-fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. In *ACM STOC*.
[28] Muhammad Naveed, Seny Kamara, and Charles V. Wright. 2015. Inference Attacks on Property-Preserving Encrypted Databases. In *CCS*.
[29] Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft. 2013. Privacy-Preserving Ridge Regression on Hundreds of Millions of Records. In *IEEE Symposium on Security and Privacy*.
[30] Pascal Paillier. 1999. Public-key Cryptosystems Based on Composite Degree Residuosity Classes.. In *EUROCRYPT*.
[31] Antonis Papadimitriou, Ranjita Bhagwan, Nishanth Chandran, Ramachandran Ramjee, Andreas Haeberlen, Harmeet Singh, Abhishek Modi, and Saikrishna Badrinarayanan. 2016. Big Data Analytics over Encrypted Datasets with Seabed. In *USENIX OSDI*.
[32] Raluca Ada Popa, Frank H. Li, and Nickolai Zeldovich. 2013. An Ideal-Security Protocol for Order-Preserving Encoding. In *IEEE Symposium on Security and Privacy*.
[33] Raluca Ada Popa, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan. 2011. CryptDB: Protecting Confidentiality with Encrypted Query Processing. In *ACM SOSP*.
[34] Raluca Ada Popa, Emily Stark, Jonas Helfer, Steven Valdez, Nickolai Zeldovich, M. Frans Kaashoek, and Hari Balakrishnan. 2014. Building Web Applications on Top of Encrypted Data Using Mylar. In *USENIX NSDI*.
[35] Ling Ren, Christopher Fletcher, Albert Kwon, Emil Stefanov, Elaine Shi, Marten van Dijk, and Srinivas Devadas. 2015. Constants Count: Practical Improvements to Oblivious RAM. In *USENIX Security*.
[36] Tahmineh Sanamrad, Lucas Braun, Donald Kossmann, and Ramarathnam Venkatesan. 2014. Randomly Partitioned Encryption for Cloud Databases. In *DBSec*.
[37] Hossein Shafagh, Lukas Burkhalter, and Anwar Hithnawi. 2016. Demo Abstract: Talos a Platform for Processing Encrypted IoT Data. In *ACM SenSys*.
[38] Hossein Shafagh, Anwar Hithnawi, Andreas Dröscher, Simon Duquennoy, and Wen Hu. 2015. Talos: Encrypted Query Processing for the Internet of Things. In *ACM SenSys*.
[39] Justine Sherry, Chang Lan, Raluca Ada Popa, and Sylvia Ratnasamy. 2015. Blind-Box: Deep Packet Inspection over Encrypted Traffic. In *ACM SIGCOMM*.
[40] E. Shi, J. Bethencourt, T.-H.H. Chan, D. Song, and A. Perrig. 2007. Multi-Dimensional Range Query over Encrypted Data. In *IEEE Symposium on Security and Privacy*.
[41] Elaine Shi, Richard Chow, T-H. Hubert Chan, Dawn Song, and Eleanor Rieffel. 2011. Privacy-preserving Aggregation of Time-series Data. In *NDSS*.
[42] D. X. Song, D. Wagner, and A. Perrig. 2000. Practical Techniques for Searches on Encrypted Data. In *IEEE Security and Privacy*.
[43] Adam Tanne. 2016. For Sale: Your Medical Records. In *Nature*. 26–27.
[44] Stephen Tu, M. Frans Kaashoek, Samuel Madden, and Nickolai Zeldovich. 2013. Processing Analytical Queries Over Encrypted Data. In *VLDB*.
[45] Frank Wang, James Mickens, Nickolai Zeldovich, and Vinod Vaikuntanathan. 2016. Sieve: Cryptographically Enforced Access Control for User Data in Untrusted Clouds. In *USENIX NSDI*.
[46] Andrew C. Yao. 1982. Protocols for Secure Computations. In *Symposium on Foundations of Computer Science*. 160 – 164.