A Framework for Handling Internet of Things Data with Confidentiality and Blockchain Support

Manoharan Ramachandran, Niaz Chowdhury, Allan Third, Zeeshan Jan, Chris Valentine, and John Domingue

Knowledge Media Institute
The Open University
United Kingdom
{manoharan.ramachandran, niaz.chowdhury, allan.third, zeeshan.jan, chirs.valentine, john.domingue}@open.ac.uk

Abstract. The Internet of Things (IoT) has emerged as a disruptive technology in recent time. Due to a sharp rise in the deployment of IoT devices across the globe, the total amount of generated data reaching record highs every day. There is no denying that IoT devices with the data that they produce enhancing our day to day lives by bringing in innovative solutions. However, there is a darker side of this technology as it enables device manufacturers and service providers to retain this data that they can later use for several purposes, including targeted marketing. Unfortunately, not many people are aware of this data misuse by service providers. To address this issue, we propose a framework for the confidential handling of IoT data with Blockchain-based verification. We used a combination of Solid and Blockchain to accomplish this goal. Our experimental results show that the proposed approach is promising and has the potential to bring a considerable change in the IoT industry.

Keywords: Internet of Things · Data Confidentiality · Blockchain · Solid · Decentralisation.

1 Introduction

With the recent advancements in the Internet of Things (IoT) industry, IoT devices started to play an essential role in our day to day lives and their uses are becoming inevitable. According to Statista, there are more than 20 billion IoT devices in use around the world which is nearly double the number from 2018 [1]. FutureIoT forecasted more than 40 billion IoT devices by 2025, generating approximately 80 Zettabytes of data [2]. Due to this rapid growth, the data generated by the IoT devices are also growing exponentially. Not many people

are aware that most IoT device manufacturers can use the data generated by devices for analytical purposes. This information usually remains buried in the terms and conditions which most people agree without reading it [3]. Because of this negligence by the people, IoT device manufacturers take control of the data with users' consent. As devices produce a considerable amount of data, it seems as good as a gold mine for IoT device manufacturers to capture the data and use it for different purposes, including targeted marketing.

This problem mostly exists because the manufacturers provide the software that interfaces with IoT devices and collects data from the devices to store them in their cloud. Users can view these data using a panel, where the analytics provides them with charts and aggregated information. Even though the analysis carried out by IoT interface software provides useful and insightful information to the users, it paves the path for manufacturers to use the data for their benefits. There is also a monopoly in the IoT industry where most of the devices are manufactured and controlled by top companies including Google, Microsoft, Amazon, Cisco and ARM [4]. As they provide sophisticated tools and analytical solutions, the demand for the IoT devices from top manufacturers is always high, giving them access to a massive amount of data every year.

To solve this problem, we propose a framework for handling IoT data with confidentiality and Blockchain support. Generally, IoT devices send their data where service providers want them to go. This practice helps the providers controlling the data by hosting them at locations of their choice. Our framework aims to break this convention by separating data from the providers. We make this separation possible using Social Linked Data (Solid). It is a Linked Data-based storage service that can be hosted and maintained by the users [5]. Solid is created by the inventor of the World Wide Web Sir Tim-Berners Lee to give control and ownership of data back to the users [6]. We also provide Blockchain-based verification to prove the integrity of the data stored in Solid Pods.

The remaining of the paper are structured as follow: Section 2 presents the literature review on IoT data handling techniques and problems; Section 3 explains the proposed framework; Section 4 describes its implementations, and finally, Section 5 demonstrates the evaluation of the framework with existing works before the paper concludes in Section 6 with a summary.

2 Literature Review

Data confidentiality and privacy of IoT data are areas of research which received a lot of attention from the research community in the last decade. Various surveys have been carried out focusing on the methods for improving security, privacy and confidentiality of such data. Most of the existing research, however, concentrate on the threats that can come from external sources to compromise the data. Recent reviews about IoT data confidentiality show that there is only a limited number of studies that evaluated IoT devices, protocols and applications

from a security point of view where specific characteristics like manufacturers and deployment contexts are discussed [7]. Instead, the foci are primarily on the security issues between the IoT layers [8], and there exists no framework or model to secure data from manufacturers [9]. On that note, it is worth mentioning that IoT systems generally do not have a standardised layered architecture, but most manufacturers follow a convention of dividing the functionalities into four laters: *Sensing Layer*, *Network Layer*, *Middleware Layer* and *Application Layer* [7]. As our work focuses on data confidentiality, we concentrate on the Middleware Layer.

IoT Data Handling with Confidentiality: Wang et al. [10] proposed a secure cloud-assisted IoT data managing method. It confidentially keeps the collected IoT data and shares with entities upon users' consent. A proxy re-encryption scheme enables the technique to secure the data by resisting all external and internal attacks. Confidential data collection, storage and access are three main components of this work. It makes IoT devices encrypt data and send to the cloud server where data resides in an encrypted format and can only be decrypted by the user during the confidential data access. Although this method looks reliable for providing data confidentiality, only Intra-IoT devices can use it because of the encrypted nature of the data. Furthermore, there is an encryption overload at the IoT device side which limits the amount of data allowed to handle at a given time.

The time latency and performance degradation in the previous method might not be suitable for the IoT setup that handles real-time data because time and performance play a key role in real-time IoT systems. To solve this problem, AI-Turjman et al. [11] proposed an agile framework that uses elliptic curve cryptography to enable confidentiality, authentication and integrity while collecting the data from the IoT devices. This method sends raw data from the sensor nodes to a traffic monitoring firewall where the data is checked for its source and encrypted. An online intrusion detection service checks the incoming encrypted data and sends it to a private cloud service where the encrypted data is stored. Based on the request, the data is decrypted and read by the user. Though this technique solves the latency and performance problems by removing the load from IoT sensors and employing separate service for encryption/decryption tasks, the costs of maintaining independent services are expensive. Besides, there is no mechanism authorise applications to access the data in the cloud server.

Kavin et al. [12] proposed a cloud storage mechanism based on the Chinese Remainder Theorem (CRT) in which the IoT data is encrypted and stored in a cloud server for preserving the confidentiality of the data. The method collects data from the source and encrypts using a novel encryption technique proposed by the authors and decrypts based on user request. The problem with this approach is that the algorithms are quite resource-intensive and not suitable for IoT applications as they can place a significant toll on the IoT devices in terms of processing power. Authors stated that a lighter version of the algorithm is in

the works. This method does not support user-controlled data sharing, and the IoT data resides in a central server potentially controlled by the service provider.

IoT and Blockchain: Recent advancements in Blockchain technology attracted researchers from various domains, including IoT. Compared to encrypted data stored on a cloud server, encrypted data coupled with Blockchain-based verification provide robust data integrity. Liu et al. [13] proposed a Blockchain-based data integrity service framework to eliminate third-party auditors validating IoT data between owners and consumers by storing them on the Blockchain in the form of smart contracts (a piece of self-executable codes that executes when specific criteria fulfil [16]). The main drawback of this work is that data stays on third-party machines with a lack of appropriate users' authorisation for sharing. Javaid et al. [14] proposed a Blockchain-based data integrity and provenance framework for IoT environments. It utilises a combination of physically unclonable functions and an Ethereum Blockchain with smart contract features for providing key functionalities. There are two notable shortcomings in this work: Storing raw data directly on an Ethereum Blockchain introduces scalability issues while frequent access to a public distributed ledger by the IoT devices would generate expensive bills.

More recently, Baqa et al. [15] proposed a new framework called *semantic smart contracts* for Blockchain-based IoT services, which enables the indexing and invoking smart contracts on Ethereum Blockchain via URIs. It converts IoT data to Linked Data using a relevant ontology and stores triples on the Blockchain. The framework enables data seekers using domain-specific semantic queries on Blockchain to retrieve the IoT data. Nevertheless, despite this approach introducing innovations to the IoT ecosystem, confidentiality issues of the data are overlooked. As can be seen from the literature, most of the existing works focused on the authentication and privacy perspective of the IoT data; there is a clear need for a solution to address the growing data exploitation problem.

3 Proposed Framework

This section describes our proposed framework in detail. The basic idea of the framework is to store the data generated from an IoT device in a Solid pod for increased confidentiality and a hash of the data on a Blockchain for validation purposes. Through the Solid pod authentication mechanism (Solid auth client), any third-party application can be granted with access to the IoT data in the Solid pod and can also verify the authenticity of the data by cross-checking the hash of the data on the Blockchain.

Figure 1 shows an overview of our framework. Data from any IoT devices can be stored in a Solid pod and depending on the type of an IoT device, how the IoT data stored on Solid pod might differ. For example, if an IoT device can run a node service (e.g. Raspberry Pi 4), it can directly communicate with

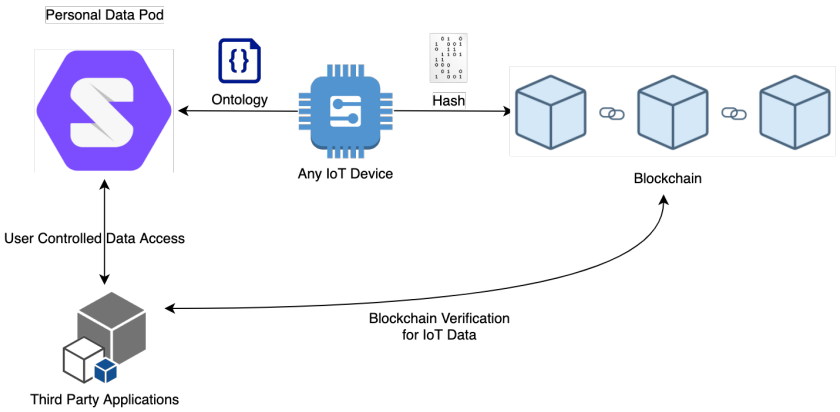

Fig. 1. Proposed framework for handling IoT data with confidentiality

the Solid pod and Blockchain. Else, it requires an external service where the data can be passed, and then the service communicates with the pod and the Blockchain. Also, the IoT data stored in the Solid pod can be converted into Resource Description Format (RDF) based on IoT ontologies so that the third-party applications can query the data directly using SPARQL when given access. At the same time, the data from the IoT device can also be hashed and stored on the Blockchain for verification purposes.

For a better understanding of the framework, we have mathematically modelled it using equations 1 – 5. In Equation 1, If_{1-n} represents the IoT data files $IoTFile_n$ from 1 to n received from an IoT device where n represents the total number of files.

$$If_{1-n} = IoTFile_1|IoTFile_2|....|IoTFile_n \tag{1}$$

In Equation 2, $IONf_{1-n}$ represents the IoT data files converted to RDF format based on a particular Ontology (ONT) related to the type of IoT data. Equation 3 and 4 represent hashing of IoT data files, while Equation 5 represents verification of the hash files.

$$IONf_{1-n} = ONT(if_1)|ONT(if_2)|....|ONT(if_n) \tag{2}$$

In Equation 3, $Bhash_{1-n}$ represents the hash pairs $\{al(If_n), al(IONf_n)\}$ of IoT data files stored in a Solid pod ranging from 1 to n that are then put on the Blockchain. Here, $al(If_n)$ represents the hash of the source IoT data file and $al(IONf_n)$ represents the hash of the same file converted to RDF format using respective ontology where n is the total number of files, and al represents the hashing algorithm used to hash the files.

$$Bhash_{1-n=\{al(If_1),al(IONf_1)\}|\{al(If_2),al(IONf_2)\}|...|\{al(If_n),al(IONf_n)\}} \quad (3)$$

Similar to Equation 3, in Equation 4, $Phash_{1-n}$ represents the hash pairs of the IoT data files stored in the Solid pod that requires verification. The pod-stored files are hashed by any user authorised third-party applications and compare them with the Blockchain-stored hash pairs to verify its authenticity. $Verify_{1-n}$ in Equation 5 is a boolean variable representing the verification of the pod-stored files in which the hash pairs of the pod-stored files $Phash_{1-n}$ are compared with the Blockchain-stored hash pairs $Bhash_{1-n}$.

$$Phash_{1-n=\{al(If_1),al(IONf_1)\}|\{al(If_2),al(IONf_2)\}|...|\{al(If_n),al(IONf_n)\}} \quad (4)$$

$$Verify_{1-n} = comp_{1-n}(Bhash_{1-n}, Phash_{1-n}) \quad (5)$$

4 Implementation and Testing

To implement and test our framework, we chose the fitness domain as it has various areas in which the manufacturers can easily use sensitive data. For example, a basic fitness tracker which can track footsteps and heart rate can be a huge source of data for a tracker manufacturer as the steps can be used to determine the physical activities of a particular person and heart rate data can aid in the identification of abnormal heart condition. These data may not appear significant, but once put into context, they become valuable. For example, the footsteps activity is a good source of data for sports shoe companies to do targeted marketing, while heart rate can be useful data for pharmaceutical companies. People who use fitness trackers should not be surprised to see advertisements related to their interest when they surf online as their fitness device can be revealing their personal information to different companies who sell relevant products.

Interestingly, most people do not read the terms and conditions specified in the fitness tracker where top manufacturers specify the information related to data capture, storage and sharing [3]. They take advantage of this oversight of the users and state that generated data will be manufacturers property, or they can share it with third parties [17]. Our proposed framework has the potential to give a perfect solution to this data exploitation problem. We have used exported dataset from a Xiaomi Mi Smart Band 4 (MiBand4) fitness tracker [21] and an open-source android application to demonstrate how users can keep the fitness data collected from a tracker under their control with blockchain-based verification [18].

Figure 2 shows the entire implementation of the proposed framework in the fitness domain. All the processes involved in this framework and shown in the mentioned figure are then explained in the following subsections.

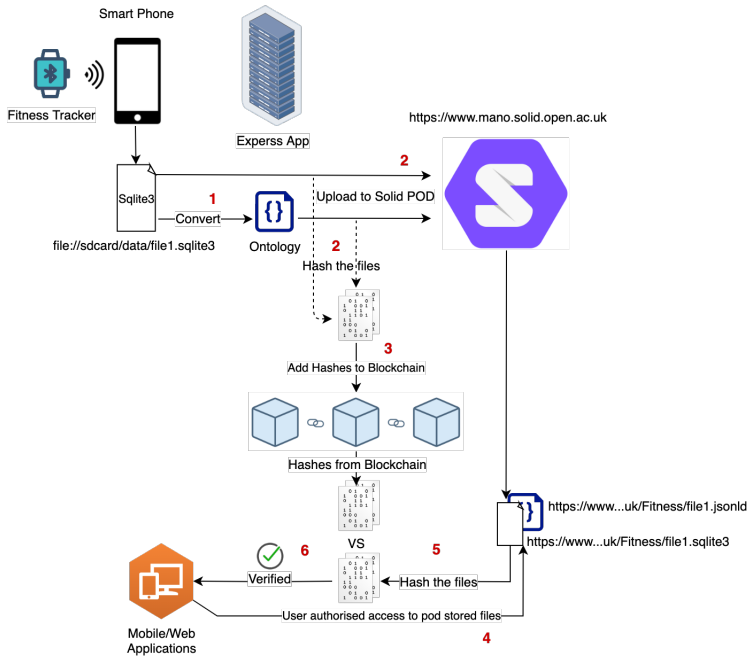

Fig. 2. Implementation of the proposed framework in fitness domain

Data Collection: Initially, the data needs to be collected from the fitness tracker using a mobile application. Most fitness trackers can only communicate through a medium such as a mobile application service that pairs with the fitness tracker and synchronizes the data from the tracker to a local mobile database. Our implementation exported the data set from MiBand4 in the form of CSV file and converted it to sqlite3 database file used with an open-source mobile application called GadgetBridge. We then again converted the data into a suitable format that mobile applications can use. To be precise, the sqlite3 file is imported into the GadgetBridge mobile application via a backup-restore option. We have created our own set of APIs using a *Node.js* express service through which the mobile application can post the sqlite3 file to the service. We have also modified the backup option in the GadgetBridge mobile application in a way that it can talk to our APIs. When the backup option is selected in the mobile application, the fitness data in sqlite3 format sent to the server in the form of a multipart POST request using retrofit where the data can be further processed [19]. The sqlite3 file has four fields, *timestamp*, *user_id*, *steps* and *heart_rate*. While some applications may contain JSON databases which do not require conversion, we used one that requires processing to demonstrate the necessary steps for con-

verting the sqlite3 to JSON before annotating it with a suitable ontology and making it a JSON-LD file.

Processing: Once the express service receives the sqlite3 file, we generate a JSON version with ontology headers. This JSON file contains three sets representing three rows from the source file in the form of key-value pairs. We also convert the data and time into ISO 8601 standard for maximal compatibility [20]. The JSON file is then annotated with the "Fitness Tracking Vocabulary for Semantic Web-based IoT devices" which is an IoT fitness tracking ontology described in [22]. The JSON-LD files are created based on a 1:1 ratio concerning the source files. As we have one source file, one equivalent JSON-LD file gets created as shown in Figure 3.


```
{
  "@context":{
    "TIMESTAMP":"http://purl.org/ifo#dateTimeStamp",
    "USER_ID":"https://schema.org/identifier",
    "STEPS":"http://purl.org/ifo#Steps",
    "HEART_RATE":"http://purl.org/ifo#HeartRate",
    "readings":" http://purl.org/ifo"
  },
  "readings":[
    {
      "TIMESTAMP":20200306T131412Z,
      "USER_ID":5678765,
      "STEPS":2454,
      "HEART_RATE":98
    },
    {
      "TIMESTAMP":20200306T131644Z,
      "USER_ID":5678765,
      "STEPS":2512,
      "HEART_RATE":90
    },
    {
      "TIMESTAMP":20200306T131721Z,
      "USER_ID":5678765,
      "STEPS":2524,
      "HEART_RATE":92
    }
  ]
}
```


Fig. 3. JSON-LD file annotated with 'IoT Fitness Tracking Ontology'

Verification Both the source file (sqlite3) and its JSON-LD file are hashed using the MD5 hashing algorithm [20], thereby creating two separate hashes for these two files. By applying the hash values in Equation 3, we obtain the $Bhash_1$. The source file and the JSON-LD file are then uploaded to a user's solid pod, and at the same time, both the hashes are inserted into the Blockchain. We used OpenBlockchain, which is private Ethereum Blockchain with proof of authority consensus algorithm. The verification process begins with applying the pod stored file hashes in Equation 4, giving us the $Phash_1$. We then apply the $Bhash_1$ and the $Phash_1$ values in Equation 5. As the process ends, $Verify_1$

returns a value of 1 if both the Blockchain stored hashes and pod stored file hashes match, otherwise 0.

If the user wishes to use the data directly from the Solid pod through URIs, then the user can wipe the local data from the mobile phone, and the mobile application can directly run the SPARQL queries on the Solid pod to view the analysis of the user's fitness data. As the fitness data resides in the Solid pod of the user, the user can have full control over the data with Blockchain support and can grant access to any third-party applications on-demand basis to access the data and revoke the access at any point of time.

5 Evaluation

Our proposed framework aims to combine the best data confidentiality practices and new technologies to provide a comprehensive solution for the fast-growing data exploitation by the IoT device manufacturers and service providers. There are several existing works which tried to address the confidentiality, privacy and integrity of IoT data. We have listed the most recent and appropriate ones in Table 1, along with the evaluation. The existing works and our proposed framework are evaluated based on the criteria: Data Confidentiality (**Confidentiality**), Authorised data sharing (**Sharing**), User-controlled storage space (**Storage**), Decentralised Trust (**Trust**) and Data-driven Reasoning (**Reasoning**).

Table 1. Evaluation of the proposed framework with existing works

Frameworks	Ref.	Year	Confidentiality	Sharing	Storage	Trust	Reasoning
Wang et al.	[10]	2018	Yes	Yes	No	No	No
AI-Turjman et al.	[11]	2018	Yes	No	No	No	No
Kavin et al.	[12]	2019	Yes	No	No	No	No
Liu et al.	[13]	2017	Yes	No	No	Yes	No
Javaid et al.	[14]	2018	Yes	No	No	Yes	No
Baqa et al.	[15]	2019	Limited	No	No	Yes	Yes
Proposed	–	2020	Yes	Yes	Yes	Yes	Yes

Data confidentiality is the measures taken into account to ensure concealment of the user's IoT data. This criterion ensures that third-party applications or hackers have no access to the stored data. Authorised data sharing refers to the process in which a framework allows its users to securely permit third parties and service providers to access their IoT data either partially or in full. The third criterion, user-controlled storage, checks if a user is using their own storage space or using the storage space provided by the service providers. Besides,

decentralised trust makes sure the data is not altered or manipulated at the time of being stored. Introducing a central authority for establishing trust could potentially jeopardise the goal of data confidentiality and accessibility; hence, the decentralised approach. Blockchain is a suitable technology that assures trust in a decentralised environment. Finally, the last criterion, data-driven reasoning, allows both the humans and computer to interpret a piece of data and understand the meaning better. Linked Data can potentially aid data-driven reasonings in the digital world.

6 Conclusion and Future Work

There is no denying that IoT devices have the potential to improve our living. However, the data that they generate are often sensitive and can pose a threat should the device manufacturers and service providers decide to exploit it. With the growing demand for these devices, the amount of generated data will be enormous in volume in the future requiring immediate attention. Most of the existing works focused on the threats arising from external sources, but the literature on the internal threats are minimal. This paper presented a comprehensive framework for the confidential handling of IoT data with Blockchain support. Implementation and testing of this framework in the Fitness domain showed us the proposal works as intended. Also, it proves that the proposed framework has the potential to bring a considerable change in the IoT industry. As our future work, we are going to implement our framework in different IoT domains to test its robustness and prove the adaptability to a broader range of products.

References

1. S. O'Dea, "IoT device installed base worldwide 2009-2020", Statista, Feb 2020
2. E. Estopace, "IDC forecasts connected IoT devices to generate 79.4 ZB of data in 2025", FutureIoT, June 2019
3. D. Berreby, "Click to agree with what? No one reads terms of service, studies confirm", The Guardian, March 2017
4. K. L. Lueth, "IoT Platform Companies Landscape 2019/2020: 620 IoT Platforms globally", IoT Analytics, Dec 2019
5. A. Sambra et al. "Solid: A Platform for Decentralized Social Applications Based on Linked Data", Technical Report, MIT CSAIL & Qatar CRI
6. E. Mansour et al., "A Demonstration of the Solid Platform for Social Web Applications", in proceedings of the 25th International Conference Companion on World Wide Web, pp. 223–226, Montreal, Canada, April 2016.
7. V. Hassija et al., "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures", IEEE Access. 7, 82721–82743, 2019

8. N. Neshenko et al., "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations", IEEE Communications Surveys & Tutorials. 21, pp. 2702–2733, 2019
9. H. HaddadPajouh et al., "A survey on internet of things security: Requirements, challenges, and solutions", Internet of Things, Elsevier, 2019
10. W. Wang, P. Xu and L.T. Yang, "Secure data collection, storage, and access in cloud-assisted IoT", IEEE Cloud Computing 5, pp. 77–88, 2018
11. F. Al-Turjman and S. Alturjman, "Confidential smart-sensing framework in the IoT era", Journal of Supercomputing, Springer, 74, pp. 5187–5198, 2018
12. B. P. Kavin and S. Ganapathy, "A secured storage and privacy-preserving model using CRT for providing security on cloud and IoT-based applications", Computer Networks. 151, pp. 181–190, 2019
13. B. Liu et al., "Blockchain Based Data Integrity Service Framework for IoT Data", in Proc. of 24th IEEE International Conference on Web Services (ICWS), pp. 468–475, Honolulu, HI, USA, June 2017
14. U. Javaid and M. N. Aman, "BlockPro: Blockchain based Data Provenance and Integrity for Secure IoT Environments", in Proc of the 1st Workshop on Blockchain-enabled Networked Sensor Systems, pp. 13–18, Shenzhen, China, Nov 2018
15. H. Baqa et al., "Semantic Smart Contracts for Blockchain-based Services in the Internet of Things", IEEE 18th International Symposium on Network Computing and Applications, 1–5, Cambridge, MA, USA, 2019
16. N. Chowdhury, "Inside Blockchain, Bitcoin, and Cryptocurrencies", ed 1, Taylor and Francis, 2019
17. Xiaomi User Agreement, https://www.mi.com/global/about/agreement, last accessed 25 April 2020
18. FreeYourGadget, Gadgetbridge, https://github.com/Freeyourgadget/Gadgetbridge, last accessed 25 April 2020
19. Square Inc: Retrofit - A tpe-safe HTTP client for Android and Java, https://square.github.io/retrofit/, last accessed 25 April 2020
20. ISO 8601, https://www.iso.org/iso-8601-date-and-time-format.html, last accessed 25 April 2020
21. Mi Smart Band 4, https://www.mi.com/uk/mi-smart-band-4, last accessed 25 April 2020
22. R. Reda, "Fitness Tracking Vocabulary for Semantic Web-based IoT Devices", https://raw.githubusercontent.com/roberto-reda/IFO/master/ifo.ttl, last accessed 25 April 2020
23. R. Rivest, "The MD5 Message-Digest Algorithm", RFC1321, April 1992, https://tools.ietf.org/html/rfc1321, last accessed 25 April 2020