An Identity Privacy Preserving IoT Data Protection Scheme for Cloud Based Analytics

Gehrmann, Christian; Gunnarsson, Martin

Published in:
Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019

DOI:
10.1109/BigData47090.2019.9006017

2019

An Identity Privacy Preserving IoT Data Protection Scheme for Cloud Based Analytics

Christian Gehrmann[*] and Martin Gunnarsson[†][*]
[*]Dept. of Electrical and Information Technology
Lund University
[†]RISE Research Institutes of Sweden, Lund, Sweden
christian.gehrmann@eit.lth.se, martin.gunnarsson@ri.se

***Abstract*—Efficient protection of huge amount of IoT produced data is key for wide scale data analytic services. The most efficient way is to use pure symmetric encryption as that allows both fast decryption at the analytic engine side as well as energy efficient encryption at the IoT side. However, symmetric encryption can only be performed if there is a way to directly map an encrypted object to the correct key. Typically, such mapping require a unique IoT identity, which constitute a privacy problem. In this paper, we present an IoT identity protection scheme for symmetric IoT data encryption. We give basic security definitions for this problem setting, present a new construction and give security proofs of security level achieved with the construction. Performance figures for a proof of concept implementation are also given. The new scheme gives a fair trade-off between identity privacy and complexity.**

***Index Terms*—identity privacy, IoT security, analytics**

I. Introduction

Internet-of-thing (IoT) is a network of physical objects or *things* embedded with electronics, software, and sensors, connected through the Internet to collect and exchange data with manufacturers, operators and other connected devices. IoT includes a variety of connected objects from tiny stuff (e.g. smart dust) to enormous stuff (e.g. an entire city). Most IoT devices are used in factories, businesses and healthcare systems. By 2025, there might be more than 75.4 billion connected devices [1] generating 175 trillion gigabytes of data[2], and total global worth of IoT technology would reach to USD 6.2 trillion by 2025[3]. This trend opens up to completely new possibilities with respect to data analysis services utilizing device data from a huge number of distributed devices [1]. The applications are very wide-ranging from healthcare and market analytics to industrial systems. In this paper, we consider big data IoT analytics from a privacy perspective. Even if the IoT units producing the data are not necessarily owned by an individual, the data they produce as well as communication patterns can reveal important business and industry secrets which should be avoided whenever possible [2] [3].

Huge scale analytic on powerful, third-party back-end cloud resources raises security and privacy concerns. One problem is that typically the cloud computing resources cannot be fully trusted. Another related problem is that if an adversary is able to observe the analytic operations or data storage read/write requests, sensitive information might be leaked. One way to tackle this problem is to use privacy-preserving cryptographic techniques [4] [5], [6]. However, so far these approaches have large overhead and thus severely limits the type of analytic operations that can be supported in the system. Especially the area of fully homomorphic encryption has achieved lots of attentions even if it is not yet fully practical [7]. An alternative approach is to operate on original data using analytic engines executed in Trusted Execution Environments (TEEs) such as Intel's SGX. This line of research has gained quite a lot of attention during the past years [8] [9] [10]. In these approaches, it is assumed that the data subject to data analysis is *already* encrypted with a suitable encryption key available to the trusted application running in the TEE. Hence, before these protection techniques can be applied, the database content must be properly encrypted with the expected keys. In a scenario, where a large amount of IoT devices regularly uploads new data items subject to analysis, the data items must also be protected prior to arriving at the cloud storage resources *and* they should be protected *end-to-end* without leaking any information about the source IoT unit. How to perform such encryption in an identity privacy-preserving and efficient way is the problem tackled in this paper.

One key difference that needs to be taken into account when designing an IoT system solution is that in many applications, the IoT devices are resource-constrained [11]. They can, for instance, be constrained both in terms of computation power, as well as, being energy-constrained since they can be powered by a battery. Besides being more resource-constrained, they are also often deployed in a decentralized manner. These constraints limits, to a varying degree, what kind of security mechanisms that can be put in place, as well as, what kind of algorithms that can be executed on the IoT units [12]. Hence by efficient, we here mean to avoid the trivial solution using public key encryption, which both is costly on the resource-

Work supported by framework grant RIT17-0032 from the Swedish Foundation for Strategic Research as well as the EU H2020 project CloudiFacturing under grant 768892

[1] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] https://www.seagate.com/gb/en/our-story/data-age-2025/

[3] https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html

constrained devices as well as when processing a large number of data items on the cloud resources.

Using a model of the availability of trusted computing engines in the cloud like the solutions in [9] and [10], we consider the *additional* and *orthogonal* security problem of privacy preserving data cloud upload of IoT subject to data analysis. Especially, we consider this problem in the context of *n*ot requiring any public key operation on the data collection, i.e. IoT, side but pure symmetric operations. Furthermore, we require IoT *individual* symmetric encryption keys as global encryption keys constitute a major security risk (a compromise of a single IoT unit will destroy the security for all or many devices). In this context, the main challenge is to design a symmetric data encryption scheme allowing fast encryption and decryption while preventing an attacker, observing the data in transfer or at cloud storage, deducing any information bout the data origin like the identity of the IoT unit producing the data. Still, it must be possible for the analytic engines running at the cloud resources to efficiently to decrypt the data. We address this challenge suggesting an data item identity preserving encryption scheme and corresponding key management scheme.

The main contributions of the paper are the following:

- We identify main security requirements for large scale, light-weight and identity privacy preserving, individual IoT data encryption and give formal security definitions.
- We present a novel encryption and corresponding key management scheme meeting the identified requirements.
- We evaluate the security properties of the proposed scheme and prove the security of the scheme for a couple of different attacker scenarios.
- We present a proof of concept implementation of the encryption scheme and make a performance evaluation.

We proceed as follows: we present the system scenario we are considering (§II), we introduce our adversary model and derive security requirements as well as make formal security definitions (§III), we give an overview of our novel IoT data encryption scheme and introduce notations (§IV), we describe our proposed key management solution (§V), we present the detailed data encryption and decryption procedures (§VI), We make a formal security analysis of the proposed solution (§VII) and present a proof of concept implementation, including performance figures (§VII). Lastly we discuss related work (§IX) and conclude (§X).

II. System scenario

We consider a system consisting of distributed IoT units, a management domain and a third-party cloud back-end (cloud provider) responsible for IoT data storage as well as data analytic operations. Figure 1 depicts an overall system scenario which includes the following components:

- The Key Management System (KMS) deployed in a management domain is responsible for generating different credentials for IoT units and cloud execution containers, as well as the other entities running at a Cloud Service Provider (CSP) that need key material. The KMS may also collect analytic results.
- CSP Storage Resource (SR), which is a repository responsible for storing IoT data.
- Storage Manager (SM) which is the interface for collecting and accepting IoT data and storing it on the SR.
- IoT units or what we refer to as devices (u) producing data which is sent securely to the SM component. The architecture is agnostic with respect to how the devices are deployed and in what type of network. All devices are assumed to have global network connectivity.
- Database Manager (DM) responsible for sharing IoT data with analytic engines. The DM is deployed in a suitable execution container on the cloud resources in the form of a Virtual Machine (VM) or in a protected execution container like SGX.
- Analytical engines (A) perform data analytics on IoT data through the DM. The analytics engines in the system are deployed on suitable execution containers on the cloud resources in the form of Virtual Machines (VMs) or in protected execution containers like SGX.
- Aanalytics consumer (C) which is authorized to receive analytics results produced by an A.

The boundary of the CSP is the space that contains SM, SR, DM and A. The management domain might also be deployed in an cloud environment but must in the model we are considering be fully trusted. We discuss the adversary model and requirements in the next section.

III. Problem setting

Next, we discuss the details of the data protection problem we are considering. We start by defining our adversary model and use this model to identify privacy requirements. Although the system scenario and architecture we are considering implies several additional requirements, the focus here are on the privacy/security requirements under the assumption of resource constrained IoT units. Next, we identify security and functional requirements on the system we are considering. Finally, we give formal security definitions.

A. Adversary model

We consider a powerful adversary who may control the CSP network domain as well as having access to the SM and SR. We do not consider denial-of-service (DoS) types attacks on these nodes though and assume that SM and SR are being able to operate properly. The adversary might also try to get full access to the computing resources but we assume the A and DM to be deployed in secure containers using secure launch in combination with secure VMs or secure launch of SGX machines. Hence, the adversary has no possibility to directly modify or eavesdrop A or DM. This model is motivated, as we stated in the introduction, with reference to trusted computing techniques in combination with secure launch as reported in [13] and protected SGX analytics as described in [10]. Recent attacks like Metldown [14] and Spectre [15] have shown that one cannot even trust the fundamental hardware functions needed for secure isolation currently in use. Despite this fact, the security with respect to secure execution environment for virtualized systems is

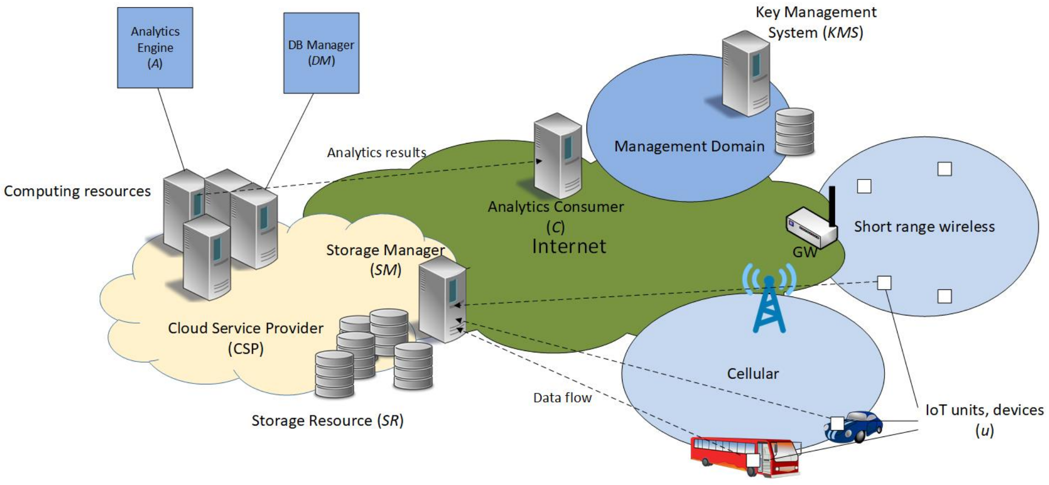

Fig. 1. System scenario.

steadily improving and we will in this paper disregard attacks on the isolation properties of the execution containers.

In line with many other works on IoT and cloud security, we assume that the adversary is acting according to the Dolev-Yao adversarial model [16]. This implies that an attacker is able to intercept, delete, change order or modify all communication messages sent over the communication links between the IoT units and the CSP domain. The adversary can also destroy messages but is not able to break any cryptographic mechanisms. The devices are assume to be semi-trusted. This means that as long as an external attacker has not compromised a unit, it will be trustworthy. However, we do not exclude the possibility of that a limited set of the IoT units in the system are completely taken over by the adversary.

The management domain including the KMS is assumed to secure not in the control of the adversary.

B. Requirements

Starting from the previously presented system architecture and the given adversarial model, we have identified the following security requirements:

R1. **Data items confidentiality:** All data items sent from a device u needs to be confidentiality protected, all the time, until they are processed by A in a secure execution environment.

R2. **Data items integrity:** All data items sent from a device u needs to be integrity protected, all the time, until they are processed by A in a secure execution environment.

R3. **Analytics results confidentiality and integrity:** All analytics results must be confidentiality and integrity protected before they are returned to the C.

R4. **Data items identity privacy:** It shall not be possible for an external adversary or a compromised IoT unit, u' to determine which data item that is produced by a, non compromised, specific IoT unit, u. This implies that it shall not be possible to trace data items from different IoT units through potential identities used in protected data items.

Many of the devices might be placed in internal network not accessible from the outside. Furthermore, a particular device, u, might for security reasons be restricted not to set up secure sessions with external entities. It might also be an advantage if several data items can be buffered at an intermediate node, before they are transferred to the SM for storage. Altogether, this gives us the following additional design requirement on the wanted solution:

R5. **IoT unit isolation:** A data transfer from u shall not require any direct interactions (session) between the IoT unit and the SM.

Among these, requirements, it is particular challenging to fulfill requirement R4 in combination with R1 and R2 for the scenario we are considering. This is due to the fact, that it shall be possible for analytic engines deployed in the cloud, to quickly decrypt data items uploaded to the cloud using symmetric encryption only. This on the other hand, requires that the symmetric key for the data items must be available which in turn typically means that the data item must be ”marked” with a key identity to allow symmetric key lookup. If a fixed identity is used, we do not fulfil R4 and this is the main security design challenge we address in this paper.

C. Formal security definitions

Next, we give formal security definitions. We here focus on formally defining R1, R2 and R4. The reason for this choice is that R3 can be fulfilled with standard secure channel and security association techniques and it is not the main problem we address here even this is a requirement for a complete system solution. Furthermore, R5 is not a security requirement as such, but a property on the solution we want to have in order to offer practical and broadly applicable solution. Hence, we here do not either give a formal definition for R5.

Denote by $u \in U$ an arbitrary IoT unit and by $m \in M_n$, where M_n is a plain text space index by n (message of length n bits), a data item produced by such unit. Furthermore, let $K_e \in \mathcal{K}_e$ and $K_{mac} \in \mathcal{K}_{mac}$, be a symmetric encryption and integrity keys respectively, known to the u and the

DM. We then denote by $c = E_{K_e}(r, m)$ the encryption of m with Initialization Vector (IV) $= r \in R$ and using a suitable symmetric encryption algorithm, E. Similar, we denote by $x = \text{MAC}_{K_{mac}}(m)$, the message tag calculation for a message, m, using a suitable MAC function , MAC[4].

Let K be an arbitrary key space and $v = f_K(u, m), K \in \mathcal{K}$ be the random packaging of message m for unit u. Here the function f_K denotes the combination of one or several encryption and/or MAC functions for a particular unit. v is then the actual message "observed" by an external entity when the message is transferred to SM.

We use the classical security by indistinguishability definition to define the expected confidentiality property of the scheme [17].

Definition III.1. An IoT protection schemes provides *confidentiality protection* if for all (non-uniform) polynomial time limited adversaries, AT, there exist a negligible function $\epsilon(n)$, such for all $\forall m_0, m_1 \in M_n, \forall r \in R$:

$$\begin{aligned}&|Pr[AT(E_{K_e}(r, m_0)) = 1] - Pr[AT(E_{K_e}(r, m_1)) = 1]|\\ &< \epsilon(n), \qquad (1)\end{aligned}$$

where the probability is taken over all choices of K_e and coin tosses by AT.

Let the adversary, AT having access to $\text{MAC}_{K_{mac}}$. We then consider the following security game (unforgeability under chosen message attack):

Game UF-CMA

- Setup: $K_{mac} \leftarrow_R \mathcal{K}_{mac}$
- Query phase: AT makes a set of quires, by selection of message $m \in M$ to get $x = \text{MAC}_{k_{mac}}$
- Guess phase: $AT \rightarrow (m', x')$
- Verify: If $m' \notin M$ and $x' = \text{MAC}_{K_{mac}}(m')$, AT wins, else AT loose.

We then use the classical unforgeability MAC security definition for message integrity security.

Definition III.2. A family of functions, MAC, is said to be (q, l, ϵ) unforgeable under chosen message attack if for all adversaries, AT who makes q queries with total size of the queries bits maximum equal l:

$$Pr[AT \text{ win game UF-CMA}] \leq \epsilon \qquad (2)$$

Definition III.3. An IoT data protection scheme which protects messages by a (q, l, ϵ) unforgeable MAC is said to provide *integrity protection* if q is greater than the maximum number of MAC values that the attacker can observe from a single IoT unit, ϵ i negligible and l is greater than the maximum number of bits in d, i.e, the maximum number of bits produced by any IoT unit for a single message.

Next, we give our identity privacy definitions. Now, let the adversary, AT having access to the output of f_k. We consider the following security game (Identity attack):

Game IDA

- Setup: $K \leftarrow_R \mathcal{K}$
- Query phase: AT makes a set of queries to get $v = f_K(u, m)$ together with u for random $u \in U$ and chosen message $m \in M$.
- Observe phase: For random $\hat{u}$ and chosen $m \in M$, AT observes $v' = f_K(\hat{u}, m)$
- Guess: $AT \rightarrow u'$
- Verify: If $u' = \hat{u}$, AT wins, else AT loose.

Definition III.4. A data and identity protection scheme, f, is said to be (q, p) unforgeable if for all adversaries, AT who makes q queries:

$$Pr[AT \text{ win game IDA}] \leq p \qquad (3)$$

Furthermore, we say that a (q, p) unforgeable protection scheme with $p \leq 1/k$, for an integer k, provides *k-anonymity*.

IV. Design overview and notations

Our goal is to provide confidentiality, integrity and identity privacy of cloud uploaded data items. The goal with the design has been to use, due to resource consumption reasons, pure symmetric key algorithms and without any requirement on session handling at the IoT side and with individual encryption keys on the IoT side avoiding that a single or few compromised IoT units will destroy the security of the complete system.

Our solution is based on the following assumption:

- Referring to solutions like the one presented in [10] [13], a trusted analytics provider is able to securely launch analytics applications (A) as well as a database manager (DM) on secure/isolated VM/containers on the CSP computing resources. The DB server is working on encrypted data stored at general available storage resources (SR) in the provider cloud.

We suggest a solution where the DB server is pre-configured (prior to secure launch) with IoT data item symmetric key material that will allow it to read encrypted data items stored on the provider storage resources. Similarly, all IoT units are pre-configured with matching (but not the same) symmetric key material allowing them to upload or release (for instance through a third entity in the local network) encrypted data items to the provider storage resources.

Data items are directly or indirectly uploaded to the storage resources (SR) through the SM in the provider network. The solution is agnostic with respect to how the data items are uploaded to the SM. The encryption of the data items are done so that an attacker who only observes stored or sent data items neither can obtain the clear text of the individual data items nor being able to know which particular IoT node that uploaded the protected data item.

Once a set of new data items are uploaded to the provider storage resource, the DM is able to immediately fetch any new items, and with low computational overhead (only symmetric encryption), decrypting these items. When the items have been decrypted, the DM updates the internal database index such that efficient search of the data items are possible. The DM

[4] In the scheme we consider the encryption and message authentication message scopes are not always the same. However, for simplicity, we here just use the notion of m for the message input both to a encryption function and a MAC function

server keeps the index in internal protected memory and/or in protected external non-volatile memory.

The data analytic application, or applications, can contact the DM through a protected channel to issue database quires on the encrypted data items. The DM server then efficiently fetches encrypted data items using the internal index and the clear text of the data items are obtained using the symmetric encryption scheme together with the shared (with the IoT units) key management scheme.

Table I summarizes the notations we use throughout the rest of the paper.

TABLE I
NOTATIONS.

U	Set of devices in the system
$\lvert U\rvert$	Cardinality of set U
$\{U_0, U_1, \cdots, U_{q-1}\} = U$	Set of q distinct subsets of U
$u \in U$	A device in the system
i	Device index
t	Group index
u_i	Device with index i
d	Data item produced by a device
IK	System wide integrity protection key
$KM1$	First symmetric master key
$KM2$	Second symmetric master key
$KM3$	Third symmetric master key
IK_i	Device unique integrity protection key
$K1_i$	First device unique encryption key
$K2_i = K2_t$	Group unique, second device encryption key
$IV1$	First Initialization Vector (IV)
$IV2$	Second IV
$c1$	First ciphertext
$c2$	Second ciphertext
h_o	Outer message authentication tag
h_{in}	Inner message authentication tag
$r, n1, n2$	Random numbers
$\lvert\lvert a\rvert\rvert$	Size of parameter a
$a\lvert\lvert b$	Concatenation of value a and b
$L_i = \{l_{i0}, l_{i1}, \cdots, l_{iw-1}\}$	Set of indices given to unit u_i
$\mathrm{E}_K(a, m)$	Symmetric encryption of message m with key K and IV = a.
$\mathrm{D}_K(a, c)$	Decryption of ciphertext c using key K and IV = a
$\mathrm{MAC}_K(m)$	Message authentication code for key K and message m
PRF(K,a)	A Pseudo Random Function taking a key K and additional data, a, as input

V. Key generation and distribution

Next, we describe the principles for key generation and distribution in the system. According to our design, the KMS is responsible for generating keys and to distribute them to the IoT units as well as the DB manager (DM), analytic engine (A) and storage manager (SM) in the system.

The design is based on the usage of four different master keys: $IK, KM1, KM2$ and $KM3$. The IK is a system global integrity protection key and the other keys different encryption master keys. Before system deployment, the KMS uses a good random source to generate these four different keys. The key IK is securely transferred and stored at SM while $KM1, KM2$ and $KM3$ are all securely transferred to the DB.

To give a k-anonymity on visible device index, the set of IoT units, U, is divided IoT subsets of at least size k:

$$U = \{U_0, U_1, \ldots, U_{s-1}\}, \forall t, 0 \leq t \leq s-1, |U_t| \geq k. \quad (4)$$

Each IoT unit is associated with a random index, i selected by the KMS. i is configured into the DB together with the rest of the key material but is *not* given to the device (u) itself. Instead, each device $u_i \in U$ is given a device unique index set:

$$L_i = \{l_{i0}, l_{i1}, \ldots, l_{iw-1}\}, l_{ip} = r||\mathrm{E}_{KM3}(r, i), \quad (5)$$

where r is chosen uniformly and at random by the KMS and E_{KM3} is suitable symmetric encryption function. The device uses the index to ”mark” data items produced by the item (see Section VI for the detailed data protection procedure). In addition, u_i is configured with *three* different symmetric keys:

- IK: the global integrity protection key.
- IK_i = PRF($KM1$, ”MAC”$||i$): an individual integrity protection key.
- $K1_i$ = PRF($KM1$, ”Enc”$||i$): a symmetric *inner* encryption key
- $K2_i = K2_t$ = PRF($KM2, t$): a symmetric *outer* encryption key

Figure 2 gives an overview of the different key configurations done during system deployment.

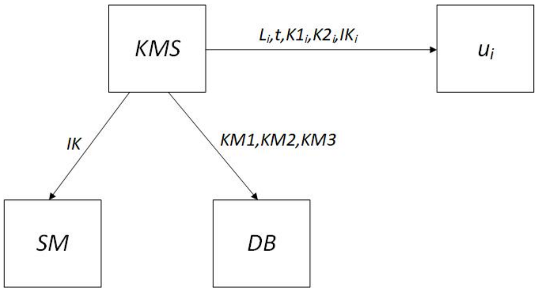

Fig. 2. Deployment key configurations.

VI. Data protection

We are considering a model where a huge number of IoT devices regularly uploads new data items to the storage server SM. According requirement R5, this shall be possible to do without the need for any security sessions. A straight forward way to handle this is to use an object security model. Object security for the IETF session protocol for constraint devices, CoAP [18] is standardized in the Object Security for Constrained RESTful Environment (OSCORE) standard [19]. While this is a very resource efficient protocol, it gives not identity anonymity of the sending party. Furthermore, it is closely aligned to the CoAP protocol. In our scenario, we do *not* want to just protect the data from the sending device to the storage manager end-to-end as offered by OSCORE, but actually *also* data storage at the SR as we considering a model where the attacker might have access to both the SM and SR. Hence, we have defined a new privacy preserving object security format. The format is completely independent of the actual bearer protocol but can for instance be transferred over CoAP as standard non-protected payload. Below, we describe the encryption procedure (at the device side) and format as well as decryption procedures (database side of the system).

A. Data encryption procedure

Each devices regularly uploads data to the SM in protected format. We suggest the following encryption procedure:

1) u_i uses a good random source to generate two random values: n_1, n_2.
2) u_i selects uniformly and at random an index, l_{ip}, from the set L_i.
3) u_i selects a first encryption IV, $IV1 = l_{ip}||n_1$.
4) u_i selects a second encryption IV, $IV2 = t||n_2$.
5) u_i encrypt the data item, d to obtain a first ciphertext: $c1 = \text{E}_{K1_i}(IV1, d)$.
6) u_i encrypt $IV1$ to obtain a second ciphertext: $c2 = \text{E}_{K2_i}(IV2, IV1)$.
7) u_i calculates a inner message authentication cod: $h_{in} = \text{MAC}_{IK_i}(IV2||c2||c1)$
8) u_i calculates a message authentication code over $IV2||c2||c1||h_{in}$: $h_o = \text{MAC}_{IK}(IV2||c2||c1||h_{in})$.

Finally, u_i sends the protected message, $IV2||c2||c1||h_{in}||h_o$, using an arbitrary communication channel to SM, which verifies the message authentication tag, h_o, and if the verification is successful, stores $IV2||c2||c1||h_{in}$ for future processing at SR. The protected message format is illustrated in Figure 3.

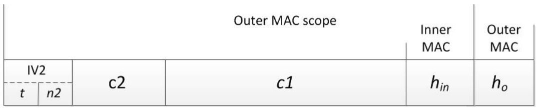

Fig. 3. Protection format.

B. Data decryption procedure

According to the system scenario we are considering, the DB is responsible for decryption protected IoT data items on SR and to index them for future processing. However, there is no need for the DB to re-encrypt the data items but they can be kept in protected form on the SR as the decryption process is quick as we show below. The decryption procedure is as follows:

1) DB fetches a protected data item from the SR: $IV2||c2||c1|h_{in}$
2) DB extracts t from $IV2$.
3) DB calculates: $K2_t = \text{PRF}(KM2, t)$.
4) DB decrypts $c2$ to obtain: $IV1 = D_{KM2}(IV2, c2)$.
5) DB extracts $l = r||c$ from $IV1$.
6) DB obtains the true device index i through decryption: $i = \text{D}_{KM3}(r, c)$ (corresponding to the index encryption in (5)).
7) DB calculates $KI_i = \text{PRF}(KM1, ”MAC”||i)$.
8) DB calculates $h'_{in} = \text{MAC}_{IK_i}(IV2||c2||c1)$. If h'_{in} equals h_{in}, the data, the item is accepted, otherwise it is rejected.
9) DB calculates $K1_i = \text{PRF}(KM1, ”Enc”||i)$.
10) DB uses $K1_i$ and $IV1$ to obtain the clear text data item $d' = \text{D}_{K1_i}(IV1, c1)$.

VII. Security analysis

Next, we analysis the security properties of the proposed protection scheme. The focus of the analysis is the security requirements R1, R2 and R4 (see Section III-B). R3 is here omitted as this is a pure back-end system property that can be achieved by state-of-the-art protection mechanisms.

Proposition 1. Given that the symmetric encryption algorithm, E, provides confidentiality protection and for a non-compromised IoT unit encryption key, $K1_i$, the proposed scheme provides confidentiality protection.

Proof. The worst case attack scenario given the prerequisites in the proposition, is when the attacker has full knowledge of $IV1$ but no knowledge of the key $K1_i$. In this case, for all different data items, d_0, d_1 and corresponding encrypted cipher texts, $c_0 = E_{K1_i}(IV1, d_0), c_1 = E_{K1_i}(IV1, d_1)$, the distinguish probability (1) equals the very same probability for the used symmetric encryption algorithm. This proofs the Proposition. □

According to our attacker model, adversary knowledge of $K1_i$ only happens when the IoT unit u_i is compromised. However, if this IoT input is compromised, the attacker will have access to all data protected by this particular unit anyway, and our any protection scheme is not useful. Hence, we conclude that the proposed scheme give good protection for the data for the *majority* of the IoT units. This is true as we assume it will only be feasible for an attacker to compromise a limited number of the IoT units in the system.

Proposition 2. For a non-compromised IoT unit u_i, given that the chosen function MAC is (q, l, ϵ) unforgeable, the proposed scheme provides data item integrity if: l is greater than the maximum number of bits in d, q is larger than the maximum number of messages produced by u_i and ϵ is negligible.

Proof. A data item produced by u_i is first encrypted into c_1, which in turn is protected by (q, l, ϵ) unforgeable MAC using key IK_i. Hence, if IK_i is not compromised. d is (indirectly) protected by a (q, l, ϵ) unforgeable MAC. Furthermore, it follows from the assumptions in the Proposition that l is greater than the maximum number of bits in d, that ϵ is negligible and that $q >$ maximum number of messages produced by u_i. Hence, the Proposition follows directly from Definition III.3 □

The same reasoning around IoT encryption key compromise, $K1_i$ for unit u_i applies also to integrity key compromise, IK_i of the unit. Hence the scheme also give good integrity data protection for the *majority* of the IoT units.

Proposition 3. Given that the symmetric encryption algorithm, E, provides confidentiality protection and for a non-compromised group encryption key, $K2_t$, the proposed protection scheme provides k-anonymity.

Proof. In the IDA game, the attacker, for chosen messages $m \in M$ observes q different evaluations of $f_{K2_i}(u_i, m) = c1, c2, h_{in}$ together with u_i. Here we have, $c2 = \mathrm{E}_{K2_t}(IV2, IV1)$, where $IV1$ is an encrypted index, l_{ip} randomly mapped from random selections of u_i. As the input to the calculation of $c1, c2$, and consequently, also the input to the calculation of h_{in} are depending on random numbers $n1, n2$, these values are randomly distributed on the encryption and MAC spaces independently of the chosen message, m. Next, the attacker can choose any previously observed value t (part if $IV1$) and corresponding previously observed message m and get the corresponding, $v' = c1', c2', h'_{in}$. If $c1', c2'$ equals a previous observed $c1, c2$, the attacker wins with probability one as he/she can choose, in this case, a previously observed, h_{in}. This probability is less than the probability that just $c2'$ equals a previously observed crypto text $c2$. Due to the random selections of $n1$, $n2$, l_{ip} by the IoT unis, observation of a previously observed $c2$ happens with probability less than $2^{\log_2(q) - \log_2(w) - \log_2(||n1||) - \log_2(||n2||)} = \epsilon$. If, $c2'$ does *not* equal a previously observed crypto text, an attacker that tries to decrypt $c2'$ to obtain l', can in the worst case map l' to a particular user u'. However, since, the group key, $K2_2$, not is known to the attacker, and the encryption algorithm gives confidentiality protection, it follows from (1) that this attack game succeeds with a probability of at most $\epsilon(n)$. Hence, in case of that $c2'$ does not equal a previous observed crypto text, a random selection of $u' \in U_t$ (as t is known to the attacker) gives the best chance of success. As $|U_t| \leq k, Pr(u' = \hat{u}) \leq 1/k$, this gives an overall probability of success $\epsilon + (1 - \epsilon) \cdot (1/k) \approx 1/k$. □

This proposition shows that as long as the group unique key not is leaked, the scheme provided k-anonymity. However, as the size of a group can be rather large (equal to t), compromise of this key cannot be excluded in same cases. However, even in this situation, the proposed scheme gives some anonymity guarantees as showed by the following proposition.

Now, denote by $\mathrm{Bin}(q; k/|U|)$ the binomial distribution, i.e. with the density function:

$$P(X = j) = \binom{q}{j} (k/|U|)^j (1 - (k/|U)^{q-j}.$$

Then let:

$$\mathrm{BSum}[(k, w), \mathrm{Bin}(q; k/|U|)] =$$

$$\begin{cases} \sum\limits_{j=0}^{q} \frac{1}{k - \frac{j}{w}} P(X = j), & \text{if } q \leq w(k-1) \\ \sum\limits_{j=0}^{w(k-1)} \frac{1}{k - \frac{j}{w}} P(X = j) + \sum\limits_{w(k-1)+1}^{q} P(X = j) & \text{otherwise} \end{cases}$$

Proposition 4. Assume, $q < w|U|$, then the proposed protection scheme is $(q, q/(w|U| + (1 - q/w|U|)\mathrm{BSum}[(k, w), \mathrm{Bin}(q; k/|U|)])$ unforgeable.

Proof. A worst case scenario is an attacker with full knowledge of $K2_t$ for all possible choices of t. Under these circumstance, the attacker can ask for q number of different values $f_{L_i}(u_i) = IV1 = (l_{ip}, n_1)$ together with u_i (outer encryption and message selection can be disregarded in this case). Next, in the game, the attacker observes $v' = f_{L_i}(\hat{u})$. If v' has been previously observed, the attacker wins the game with probability one. For each data protection occasion at most q different (l_i, u_i) pairs have been recorded by the attacker. Furthermore, as $\hat{u}$ is selected at random *and* the index l_i is chosen at random among the w different available indices, the probability that v' has previously been observed is then less than $q/(w|U|)$. This follows from the fact that the total number of (l_i, u_i) pairs equals $w|U|$ and that maximum q unique different pairs have been observed by the attacker. If v has not previously been observed, an optimal game strategy for the attacker is to choose u' as the identity of the least number of previously observed identities in $\{v'\}$ belonging to set U_t. Denote this number by z. Furthermore, assume, the number of observed elements U_t equals j. Then the probability of successful attack for this strategy is less than $(w - z)/(wk - j) \leq w/(wk - j) = 1/(k - (j/w))$, if $j \leq w(k-1)$. While if $j > w(k-1)$, the probability is less than 1. The probability of having j elements in the previous observation belong to set U_t is due to the random selections, equal to the binomial density function. Hence, by taking the expected value of $1/(k - (j/w))$ for the binomial distribution and summing up to the number of observations, q, we end up with the an expected probability which is less than $\mathrm{BSum}[(k, w), \mathrm{Bin}(q; k/|U|)]$. □

This proposition is proved under the scenario that all keys $K2_t$ are leaked which is typically not possible to achieve for a limited attacker. Even under this circumstance, as the proposition shows, the scheme still provides a level of anonymity. The strength of the anonymity can be tuned using the size of the parameter w. However, a larger w comes with higher IoT non-volatile memory cost. It is important to also notice though, that unforgeability is made under the worst chosen message attack scenario and in many practical situations it will not be possible for an attacker to gather enough number of clear text (l_i, u_i) pairs for protected data items. Especially, it is hard for an attacker to get knowledge of the real identity behind an observed index value, l_i.

VIII. Performance figures

A. Proof of Concept Implementation

To evaluate the feasibility of our suggested privacy protection scheme, we have implemented a proof of concept system. We have developed an application for IoT devices that generate data items that are encrypted according to our proposed scheme. These encrypted data items are then sent to a SM where h_0 is verified and then to DM where they are decrypted. The KMS is left out of scope. Our IoT application for data encryption is written in C and is running on Contiki-NG[5], an open source operating system for constrained IoT devices. The IoT devices that we have run our tests on are Zolertia Firefly[6] development boards based on the Texas Instruments cc2538 [20] system on chip. The cc2538 features an ARM Cortex-M3 clocked at 32MHz, with 32KB of RAM

[5] https://github.com/contiki-ng/contiki-ng/

[6] https://zolertia.io/product/firefly/

and 512KB of ROM. The back-end system that consists of the SM and DM is written in Java and running on a Linux host, specifically a Lenovo T460 laptop with an Intel Core i7-6600U CPU clocked at 2.60 GHz. We have chosen the following algorithms in our implementation:

- $E_K(m)$ & $D_K(c)$ AES128-CTR
- $PRF(K,a)$ HKDF-SHA256
- $MAC_K(m)$ HMAC-SHA256

The AES128-CTR algorithm and the SHA265 algorithm used was implemented in software on the IoT devices. The encrypted device indexes l_{ip} was selected to be 8 Bytes long, the IoT device was provisioned with $|L_i| = 10$. The encrypted data items were transferred from the IoT device to the back-end using CoAP [18]. The transfer of data is left out-of-scope for these performance measurements since our proposed scheme is independent of underlying protocols.

B. IoT Device Performance

As discussed in Section I, energy is a major concern for IoT devices, especially those that rely on battery power. CPU-time is also limited on constrained systems. Both these metrics are important when considering solutions aimed at IoT devices. We have measured the time taken to encrypt data items. To investigate how the performance depends of the size of time data item d we have tested the following sizes of d; 1,8,16,32 and 64 bytes. For each size of data items d we did the encryption 500 times. The times were measured and the energy consumption was calculated from the times and the stated power consumption in the cc2538 data-sheet. The results can be seen in Figure 4.

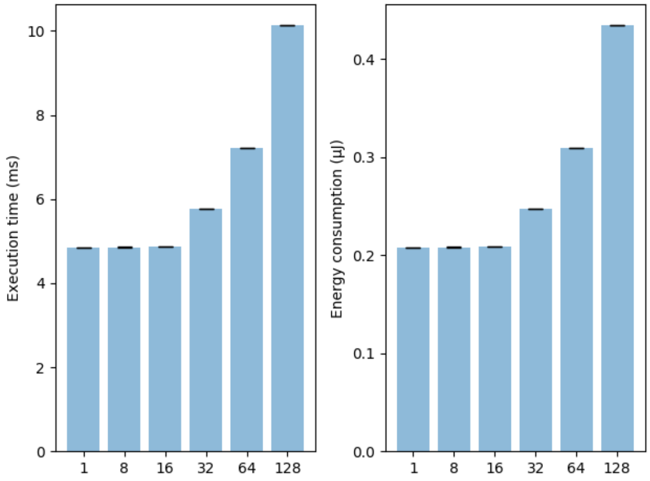

Fig. 4. Execution time and energy consumption for encrypting data. The graphs show the mean of the of the execution times and derived energy consumption with a 95% confidence interval.

C. Back End Server Performance

To evaluate how the throughput of a back-end server would be affected by the privacy protection scheme we have measured the time taken to verify h_0, furthermore we have measured the time taken to decrypt the data item d including verifying h_{in}. The performance was measured running in a single thread. We have measured the for different encrypted payloads d; 1,8,16,32 and 64 bytes. The times for a single payload size d varies considerably, we have made 2000 measurements for each payload size. The times can be seen in Figure 5.

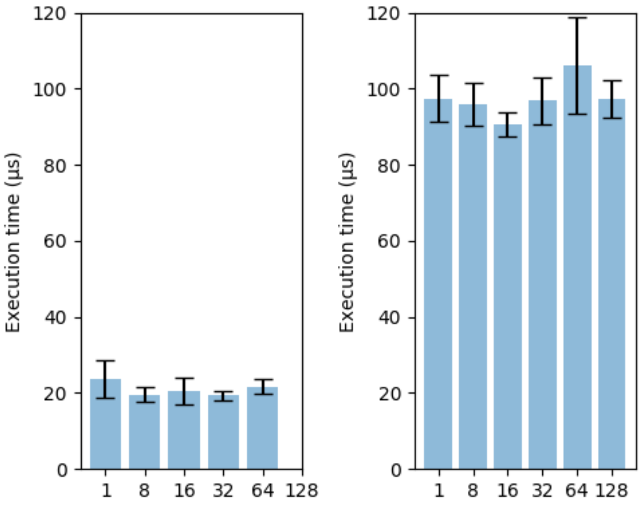

Fig. 5. Execution time, left graph shows verifying h_0, right graph decryption of encrypted d and verifying h_{in}. The graphs show the means of the of the execution times with a 95% confidence interval.

To give an estimation of the throughput of our solution we use conservative numbers of $50\mu s$ for verifying h_0 and $150\mu s$ for decrypting and verifying d, this gives us a total time of $200\mu s$ for one data item. The total throughput for one core would then be 5000 data items per second.

D. Memory usage on IoT devices

IoT devices have limited resources in terms of memory, any scheme developed for such a device must keep this in mind. Here we present numbers for the memory utilization of our implementation. The total utilization of ROM was 2344 Bytes and 426 Bytes of RAM. This was used by SHA256 836 Bytes, HMAC-SHA256 114 Bytes, AES128-CTR 1026 Bytes, Encryption Function 368 Bytes. The RAM was divided between 144 Bytes of keys in RAM and buffers for the encryption process of 282 Bytes. This is manageable amounts of memory needed for such a scheme. If the cipher and hash algorithms would be hardware-accelerated the memory usage would be even lower.

IX. Related work

Privacy is a major concern in the IoT paradigm [21]. People and devices are surrounded by billions of IoT devices gathering zettabytes of data, device manufacturers still do not pay enough attention to privacy while IoT devices are not capable of handling costly solutions to preserve privacy.

When discussing privacy it is worthwhile to note that there are several types of privacy [22]. Data privacy aims at preserving privacy by not revealing data created or owned by an entity, while identity privacy aims at protecting the identity of a user or entity. There are also the notion of spatial or location privacy, here the goal is to hide or obfuscate the location of the user or entity. This is mostly relevant in the domain of mobile devices [23] but can also have an impact for V2X networks and IoT networks. Location privacy is not directly related to the work we present in this paper.

The principle of k-anonymity was first introduced 1998 by Pierangela and Sweeney and has been extensively use as an annonymization measure in different privacy settings [24]. An overview of different k-anonymity approaches is given in [25]. In our paper we adapt the k-anonymity principle in an IoT identity privacy setting.

General privacy-preserving solutions include differential privacy [26], homomorphic encryption [27] [28], and secure multi-party computation [29]. Another general line of research which is relevant to the IoT paradigm is privacy-preserving aggregation of time-series data [30] [31] [32] [33]. Many sensors periodically generate data on e.g. temperature and sends it to a server for analysis. A recent summary of these more general problems can be found in [34]. Bista et. al. provides a survey of privacy-preserving data aggregation protocols for wireless sensor networks (WSN). In [35], a scheme for anonymous data transfer using only symmetric key operations is presented. The paper introduces the notion of twin-keys, keys negotiated between two nodes where the nodes does not know the identity of the other node in the pair. This provides anonymity of individual devices when doing data aggregation.

However, all these approaches are so far elusive for the IoT paradigm: they are too computationally costly for resource-constrained IoT devices.

Going into solutions aimed specifically at IoT it is worth to note that IoT includes a wide spectrum of devices and technology. While different solutions have been proposed for IoT, the work has primarily aimed at data privacy. One application of IoT is smart electricity meters (SM), a device measures the electricity consumption at a customer. The measurements needs to be forwarded to the utility-company for billing, but the customers privacy needs to be considered. Learning when the customer utilizes electricity can reveal the users habits. In [36] Silva et. al. presents a scheme for data aggregation in smart electricity meters using an Intel SGX enclave to perform the data aggregation while providing end user privacy.

In [37] Zhang et. al. survey the entire fields of security and privacy in edge computing. They give an overview of edge computing, list issues regarding security and privacy, list requirements. They also provide descriptions of key technologies: Identity-based encryption, Attribute-based encryption, proxy re-encryption, homomorphic encryption and searchable encryption. They give an overview of state-of-the-art solutions for data confidentiality, data integrity, privacy preserving and more. They have a section on both data-privacy and identity-privacy, they list some proposed schemes for identity privacy.

Identity privacy has been researched primarily in the fields of Mobile Communications and Vehicular Networks. in the field of Mobile Communications Khan et. al. presents their scheme for dynamic credential generation in [38], they also provide an extensive overview of work in the field. Their proposed scheme uses public-key encryption, which makes their proposed solution too computationally intensive for our use-case.

In the field of Vehicular Networks Identity privacy is important [39] since a vehicle broadcasting the the identity of the vehicle or driver would enable location tracking of the vehicle or driver. Most of the identity privacy issues of Vehicular networks are solved by pseudonyms, the vehicle is issued with a public-key pair that is periodically changed. Much research has been done on how to improve these schemes [40], [41]. However, since the basis of these systems are based on public-key cryptography they are to computationally complex for constrained IoT devices.

X. Conclusion and future work

In this paper, we presented a novel principle for IoT identity protection when using pure symmetric key based data confidentiality and integrity protection. The symmetric key approach has big advantages compare to a public key-based approach as it allows fast analytic processing directly on the protected data items on cloud resources. Identity privacy in this context has not been treated in the literature before and we provided basic security definitions. Using these definition, we presented a novel *combined* identity protection, encryption and integrity protection scheme for IoT data objects. The suggested protection scheme gives not full privacy in all adversary scenarios but, as we view it, gives a fair trade-off between identity protection and complexity. In particular, the proposed schemes uses a two layered protection approach where an ”outer” protection schemes gives k-anonymity based on symmetric keys shared by several IoT units. If such key would be compromised, an ”inner” identity protection schemes based on random encryption gives a second level of privacy defense. The security analysis we presented, shows that a reasonable level of identity privacy is achieved with this approach, as long as the adversary not has access to a large number of compromised IoT units or a large number of mappings between specific protected data items and the IoT identity behind the data items. Furthermore, by tuning the protection parameters, increased privacy can be achieved thought the price of more memory usage at the IoT device side. Our proof of concept implementation verifies that the proposed principle indeed offers both low energy consumption encryption at the IoT side as well as fast decryption at the analytic engine side. In future work, we intend to make a full-scale implementation of the approach on IoT data from an industrial control system. In this extended system trial, we will also integrate the solution with a selected set of state-of-the art analytic engines. It is also left for future work to investigate if even more efficient identity privacy preserving schemes for symmetric encryption can be constructed.

References

[1] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, “Big iot data analytics: Architecture, opportunities, and open research challenges,” *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[2] A. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.

[3] C. Maple, "Security and privacy in the internet of things," *Journal of Cyber Policy*, vol. 2, no. 2, pp. 155–184, 2017. [Online]. Available: https://doi.org/10.1080/23738871.2017.1366536

[4] A. Papadimitriou, R. Bhagwan, N. Chandran, R. Ramjee, A. Haeberlen, H. Singh, A. Modi, and S. Badrinarayanan, "Big data analytics over encrypted datasets with seabed," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 2016, pp. 587–602. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/papadimitriou

[5] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: Protecting confidentiality with encrypted query processing," in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, ser. SOSP '11. New York, NY, USA: ACM, 2011, pp. 85–100. [Online]. Available: http://doi.acm.org/10.1145/2043556.2043566

[6] D. Wang, B. Guo, Y. Shen, S. Cheng, and Y. Lin, "A faster fully homomorphic encryption scheme in big data," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, March 2017, pp. 345–349.

[7] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 79:1–79:35, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3214303

[8] M. Xu, A. Papadimitriou, A. Feldman, and A. Haeberlen, "Using differential privacy to efficiently mitigate side channels in distributed analytics," in *Proceedings of the 11th European Workshop on Systems Security*, ser. EuroSec'18. New York, NY, USA: ACM, 2018, pp. 4:1–4:6. [Online]. Available: http://doi.acm.org/10.1145/3193111.3193115

[9] W. Zheng, A. Dave, J. G. Beekman, R. A. Popa, J. E. Gonzalez, and I. Stoica, "Opaque: An oblivious and encrypted distributed analytics platform," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. Boston, MA: USENIX Association, 2017, pp. 283–298. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/zheng

[10] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, "VC3: Trustworthy data analytics in the cloud using SGX," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 38–54.

[11] D. Mun, M. L. Dinh, and Y. Kwon, "An assessment of internet of things protocols for resource-constrained applications," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, June 2016, pp. 555–560.

[12] R. Roman, P. Najera, and J. Lopez, "Securing the internet of things," *Computer*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[13] N. Paladi, C. Gehrmann, and A. Michalas, "Providing user security guarantees in public infrastructure clouds," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 405–419, July 2017.

[14] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin, Y. Yarom, and M. Hamburg, "Meltdown: Reading kernel memory from user space," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 2018, pp. 973–990. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/lipp

[15] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, "Spectre attacks: Exploiting speculative execution," in *2019 2019 IEEE Symposium on Security and Privacy (SP)*, vol. 00, 2019, pp. 19–37. [Online]. Available: doi.ieeecomputersociety.org/10.1109/SP.2019.00002

[16] D. Dolev and A. C. Yao, "On the security of public key protocols," in *Proceedings of the 22Nd Annual Symposium on Foundations of Computer Science*, ser. SFCS '81. Washington, DC, USA: IEEE Computer Society, 1981, pp. 350–357. [Online]. Available: https://doi.org/10.1109/SFCS.1981.32

[17] O. Goldreich, "On the foundations of modern cryptography," in *Advances in Cryptology — CRYPTO '97*, B. S. Kaliski, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 46–74.

[18] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252, Jun. 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7252.txt

[19] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, "Object security for constrained restful environments (oscore)," Internet Requests for Comments, RFC Editor, RFC 8613, July 2019.

[20] T. Instruments, "Cc2538 powerful wireless microcontroller system-on-chip for 2.4-ghz ieee 802.15. 4, 6lowpan, and zigbee applications," *CC2538 datasheet (April 2015)*, 2015.

[21] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Big data privacy in the internet of things era," *IT Professional*, vol. 17, no. 3, pp. 32–39, 2015.

[22] P. Porambage, M. Ylianttila, C. Schmitt, P. Kumar, A. Gurtov, and A. V. Vasilakos, "The quest for privacy in the internet of things," *IEEE Cloud Computing*, vol. 3, no. 2, pp. 36–45, 2016.

[23] L. Chen, S. Thombre, K. Järvinen, E. S. Lohan, A. Alén-Savikko, H. Leppäkoski, M. Z. H. Bhuiyan, S. Bu-Pasha, G. N. Ferrara, S. Honkala *et al.*, "Robustness, security and privacy in location-based services for future iot: A survey," *IEEE Access*, vol. 5, pp. 8956–8977, 2017.

[24] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1142/S0218488502001648

[25] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, and L. Murphy, "A systematic comparison and evaluation of k-anonymization algorithms for practitioners," *Trans. Data Privacy*, vol. 7, no. 3, pp. 337–370, Dec. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2870614.2870620

[26] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, 2006, pp. 265–284. [Online]. Available: https://doi.org/10.1007/11681878_14

[27] C. Gentry *et al.*, "Fully homomorphic encryption using ideal lattices." in *Stoc*, vol. 9, no. 2009, 2009, pp. 169–178.

[28] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. ACM, 2011, pp. 113–124.

[29] J. Bringer, H. Chabanne, and A. Patey, "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 42–52, 2013.

[30] M. Joye and B. Libert, "A scalable scheme for privacy-preserving aggregation of time-series data," in *Financial Cryptography and Data Security*, A.-R. Sadeghi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 111–125.

[31] F. Benhamouda, M. Joye, and B. Libert, "A new framework for privacy-preserving aggregation of time-series data," *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 3, pp. 10:1–10:21, Mar. 2016. [Online]. Available: http://doi.acm.org/10.1145/2873069

[32] K. Emura, H. Kimura, T. Ohigashi, and T. Suzuki, "Privacy-Preserving Aggregation of Time-Series Data with Public Verifiability from Simple Assumptions and Its Implementations," *The Computer Journal*, vol. 62, no. 4, pp. 614–630, 12 2018. [Online]. Available: https://doi.org/10.1093/comjnl/bxy135

[33] E. Shi, T. H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Annual Network & Distributed System Security Symposium (NDSS)*, 2011.

[34] Z. Shan, K. Ren, M. Blanton, and C. Wang, "Practical secure computation outsourcing: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 31, 2018.

[35] M. Conti, L. Zhang, S. Roy, R. Di Pietro, S. Jajodia, and L. V. Mancini, "Privacy-preserving robust data aggregation in wireless sensor networks," *Security and Communication Networks*, vol. 2, no. 2, pp. 195–213, 2009.

[36] L. V. Silva, R. Marinho, J. L. Vivas, and A. Brito, "Security and privacy preserving data aggregation in cloud computing," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. New York, NY, USA: ACM, 2017, pp. 1732–1738. [Online]. Available: http://doi.acm.org/10.1145/3019612.3019795

[37] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18 209–18 237, 2018.

[38] A. N. Khan, M. M. Kiah, S. A. Madani, M. Ali *et al.*, "Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing," *The Journal of Supercomputing*, vol. 66, no. 3, pp. 1687–1706, 2013.

[39] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn, "A security credential management system for v2v communications," in *2013 IEEE Vehicular Networking Conference*. IEEE, 2013, pp. 1–8.

[40] X. Lin and X. Li, “Achieving efficient cooperative message authentication in vehicular ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3339–3348, Sep. 2013.

[41] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen, “Tsvc: Timed efficient and secure vehicular communications with privacy preserving,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4987–4998, 2008.