Authenticated and Privacy-Preserving Consent Management in the Internet of Things

Maryline Laurent, Jean Leneutre, Sophie Chabridon, Imane Laaouane

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 151 (2019) 256–263

Procedia Computer Science

www.elsevier.com/locate/procedia

The 10th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 29 - May 2, 2019, Leuven, Belgium

Authenticated and Privacy-Preserving Consent Management in the Internet of Things

Maryline Laurent[a], Jean Leneutre[b], Sophie Chabridon[a,*], Imane Laaouane[a]

[a] *SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, France*
[b] *LTCI, Télécom ParisTech, Université Paris-Saclay, France*

Abstract

As the Internet of Things (IoT) starts providing meaningful solutions in multiple domains, users expect to take full advantage of the features and benefits of smart devices, but not at the cost of privacy loss. They want to keep control over their own data, e.g. through consent and authorization management. This paper proposes a lightweight privacy-preserving solution for managing user's consent relative to specific purposes (obligations). The originality of our proposal is manyfold. First, the consent is issued cryptographically by the user over some consented specific purposes, thus it protects both the user and the service provider against possible repudiations. Second, the users' privacy is preserved as the protocol supports untraceability over the channel, and pseudonymity with regard to the service provider. Pseudonyms are fully managed by the users themselves through suitable use of Hierarchical Identity-Based Signature (HIBS). Third, the solution is lightweight in terms of communication and computation, thus making it suitable for IoT resource constrained environments. Fourth, an illustrative car-sharing use case is presented where users are able to personalize their driving experience. Fifth, a formal validation of the protocol is provided with the AVISPA tool, along with an informal security and privacy analysis. Sixth, our approach addresses part of the European General Data Protection Regulation (GDPR), as it supports user consent management and helps providers with handling accountability.

Keywords:
Internet of Things (IoT); consent; authenticated consent ; privacy-preserving consent; privacy; security; General Data Protection Regulation (GDPR); HIBS; AVISPA;

1. Introduction

The Internet of Things (IoT) is transforming the interactions between companies and consumers in several domains. The underlying technology is evolving quickly, and Gartner [1] estimates that more than 20.4 billions IoT devices will

* Corresponding author. Use email
E-mail address: Sophie.Chabridon@mines-telecom.fr

1877-0509
10.1016/j.procs.2019.04.037

be in use by 2020. It is now becoming possible to collect a high amount of personal data in many different ways without the user being clearly aware of that. As a consequence, the development of the IoT raises concerns in terms of security and privacy [16, 8], especially with regard to user information and consent [9].

The informed consent of a data subject (e.g., a citizen) is indeed necessary to support the legitimate processing of personal data by a third-party application. The General Data Protection Regulation (GDPR) puts emphasis on digital citizen awareness requiring consent for data processing and usage [2]. However, as discussed in [9], the GDPR application along with associated recommendations is still an open issue in the IoT environment.

This paper addresses the issue of a privacy-preserving consent management in the IoT. Our proposal supports undeniable consent, untraceability over the channel, and pseudonymity with regard to the service provider with user-centric pseudonyms management. The user is requested to sign an informed consent, i.e. the user is informed through the obligations sent by the service provider for which purposes their personal data are collected. This signature is related to a pseudonym the user is issuing on her own per service provider, thus mitigating some possible cross sharing of user data among providers. Signature generation and pseudonym management are lightweight processings based on Hierarchical Identity-Based Signature (HIBS). The signed consent can later prove authenticity of the consent, which is helpful to prove that data handling is compliant to the user's consent in case of a provider being audited.

The rest of this paper is organized as follows. Section 2 discusses related works with regard to consent management in the IoT. Section 3 describes our original and illustrative car sharing scenario. Section 4 presents how personal data are managed through the app with HIBS. Section 5 details our proposed protocol and Section 6 validates it using the AVISPA tool. The protocol analysis is done in Section 7 and Section 8 concludes the paper.

2. Related Works

We discuss in this section some research works related to consent management in the IoT. The GDPR and WP29 recommendations [3] are discussed in [9] with regard to transparency and consent management in the IoT. This paper highlights the urgent need for new solutions able to reduce the imbalance of powers between data controllers and data subjects. It is suggested to use registries in order to communicate data subjects privacy preferences. Also, for the data subjects to express their consent, it is proposed to use Wi-Fi information elements to advertise information. However, these suggestions are restricted to the way consent is transmitted.

The paper [11] discusses the concept of privacy-by-design, informed consent and universal usability in the smart health domain. The article highlights the practical steps to be taken in the enhancement of the IoT regarding the privacy-by-design principles but no technical solution is described.

An agent-based, policy-based, and privacy-enabling framework for informed consent in IoT is proposed in [10]. This framework includes three components: 1) Privacy preserving authentication, relying on IBM's Identity Mixer 2) Informed Consent Policy Rules, following Event-Condition-Action (ECA) structure 3) User-Centric Policy Manager, controling the flow of data collected and transmitted by the IoT devices to service providers. Even though this solution manipulates only a subset of the identity attributes of a user, there is still a risk of re-identification by combining them with external data that requires further assessment.

A recent work [5] proposes authenticated key establishment in the IoT, relying on OAuth 1.0a and COSE (Concise Object Signing and Encryption) objects. Although this work is not specifically targeted at consent management, it presents similarities with our approach by not relying on a trusted third party, the authorization server.

As presented above, the current literature dealing with consent management in the IoT mainly provides a general overview of research challenges and only very preliminary candidate solutions have been proposed.

3. Scenario

To better illustrate the current issues of privacy and control of personal data, we present a specific use case of a car sharing system that allows to personalize the driving experience. This car sharing system manages a fleet of vehicles, with vehicles available at several rental stations, to be used by a group of users pre-registered into the system. When entering a vehicle, a user may provide their driving preferences and configure the vehicle.

We consider a mobile application allowing the user to manage their personal data to be accessed by several services, one of them is the car sharing service. The context is as follows: the user of the vehicle consents to provide some

personal data, in exchange for a personalized driving experience. We identify two types of personal data used in the car:

1. Ergonomics of the car: seating position, steering wheel position, neck restraints.
2. Comfort: music, radio, temperature.

At each driving session, the user may consent to the collection of the personal data generated via the sensors of the vehicle, in order to optimize the personalization process. Without consent, the car may still be used but with a standard configuration.

Our proposed scenario includes four communicating parties:

1. **The user (User)**: the driver who wishes to personalize their driving experience.
2. **The Mobile application (App)**: the application managing the user's personal data and issuing electronic consent to the service provider. The consent is issued for some data processing, that the user affords to the service provider. This application is located on the personal device of the user (e.g. smartphone), it stores data locally and might interact with many different service providers.
3. **The car (Car)**: the connected object that we illustrate in this use case.
4. **The car server (Server)**: the remote server that lays behind the connected object and interacts with it.
5. **The App manager**: the authority delivering the App software embedding some HIBS cryptographic material.

Following the analysis conducted in [15], this use case is representative of personalization services exploiting a user's profile to provide a relevant service tailored to the needs of this user. Such user's profile data collection may lead to inferences about the user that could reveal sensitive information or indirectly help in identifying the user. In the scenario, the seating position combined with the steering wheel position could be an indicator of the weight of the user and might be used to infer if this user is overweight. Music preferences, if very unusual and infrequent, may lead to identify the person when fused with some unsecured online download music platform data. We therefore consider profile data and IoT generated data that refer to a user as personal data that require to be managed under data protection law as proposed in [12, 13].

4. Personal Data Management through HIBS at App Side

To give the App application full autonomy in self-managing its security with regard to several service providers and to secure consent proof generation, we selected the hierarchical identity-based cryptography (HIBC) [7]. First, as an identity-based cryptography scheme, it enables public and private key management by a Private Key Generator (PKG) in a certificateless manner, with no need for some cumbersome Public Key Infrastructure deployment. The public key is directly derived from the identity of App [14], and App has only to send its ID for the communicating entity to know its public key. Second, it is hierarchical and it enables any entity, e.g. App, to generate on its own as many identities as needed with the associated private keys. We consider a three level ID hierarchy where the App manager is the root ID entity in charge of generating HIBC public parameters for the whole ID tree and its master key (which must remain private). App is assigned a secret by the App manager and is then able to generate one pseudo with its private keys for each of its interactive services, e.g. the car sharing service. To issue a secure consent proof for Server approval, App has to sign the obligations consented by the user using a Hierarchical Identity-Based Signature (HIBS) scheme. The Server checks the validity of the signature, based on the preregistered public parameters and pseudo. Preregistration of the App pseudo is part of the critical enrolment phase presented in Section 5.4.

Advantages of HIBS against legacy PKI are manyfold when App interacts with several service providers. First, App is required to host only one main identity (the second level identity of HIBC hierarchy) whatever the number of service providers. App is then responsible for generating one pseudonym per service provider. This feature is memory saving with limited cryptographic material needed to be stored at the mobile App. Second, untraceability across service providers is the second key feature of our HIBS approach, as Apps are sharing the same public parameters and

have each one pseudo per Server which is unlinkable to their other pseudos. Untraceability support with a PKI would be at the cost of defining one identity (and certificate) per service.

For further explanations relative to HIBS, please refer to [14] which presents the basic IBE (identity-based Encryption) principles, [6] that elaborates the transformation from IBE to IBS (identity-based Signature) and [7] which extends IBS to HIBS.

5. Protocol Supporting Mutual Authentication and Consent Proof

After presenting the required security properties, threat models and assumptions of our protocol, we explain how the enrolment of the user in the system works and then our 3-way mutual authentication protocol with consent management.

5.1. Security Properties Required in the Scenario

The specific security properties of our protocol can be specified as follows:

- **Authentication**, which can be subclassified into:
  - **User authentication to App**: App must verify the user is a legitimate user (e.g. through a pin code or fingerprint capture) before giving him/her access to any services, including the free sharing car service.
  - **App authentication to Server**: Server must check the communicating App is the claimed one, so that a service is next charged to the right service consumer. This mitigates fraud experience.
  - **Car authentication to App**: App needs to check that a car is belonging to the official car sharing service, to guarantee user's safety.
  - **Consent origin authentication by Server**: For the consent to be registered by Server as a secure proof of consent issued by User over some personal data processing (obligations), the consent has to be proven to be authentic.
- **Anonymity of App to Car**: Car must not identify App (and so User) during the authentication session. As a consequence, the protocol must not leak any identifying values to Car.
- **Untraceability of App to Car**: Car must not track App from one session to another, based on the authentication protocol elements.
- **Untraceability of App across multiple Servers**: Cross identification of App clients by the Servers is not possible.
- **Freshness**: Messages exchanged during our mutual authentication session must be fresh to counteract possible replays and masquerading attacks. Our protocol uses random values to enforce freshness.

5.2. Privacy and Security Threat Models

The following privacy and security games are defined to model possible threats against our 3-way protocol depicted in Figure 1.

Game 1: Privacy as an Indistinguishability Problem

This game questions the anonymity, untraceability and freshness properties described in Section 5.1.

Phase 1: A probabilistic polynomial time (PPT) adversary A has access to any App_i, $i \in I$ secret material, public parameters and ID and establishes as many sessions as it wants to Server(s).

Phase 2: A plays the role of Car and the challenger is acting as an App. In subphase 2.A, the challenger is App_j, $j \notin I$ and answers to multiple A's requests. In subphase 2.B, the challenger flips a coin and acts either as App_j or App_k, $k \notin I \cup \{j\}$, and answers to A. A wins the game if it is able to guess whether $j = k$.

We say that the protocol supports indistinguishability if the probability for A to win the game is close to 0.5.

Game 2: Security - User's Account Masquerading

This game questions the authentication (User authentication to App, App authentication to Server) and freshness properties described in Section 5.1.

Phase 1: In subphase 1.A, a PPT adversary A has access to any App_i, $i \in I$ secret material, public parameters and ID, and it establishes as many sessions as it wants to Server. In subphase 1.B, A observes messages between the Server and a chosen App_j, $j \notin I$.
Phase 2: The challenger acting as the Server challenges A for getting valid message $M2$ (e.g. in Figure 1) for any App_i, $i \notin I$. A wins the game if it is able to generate a valid signed consent.
We say that the protocol is resistant against user's account masquerading if the probability for A to win the game is negligible.

Game 3: Security - Fake Car
This game questions the authentication (Car authentication to App) and freshness properties.
Phase 1: A PPT adversary A has access to App_i, $i \in I$ secret material, public parameters and ID, and it establishes as many sessions as it wants to Server.
Phase 2: The challenger acting as any App_i, $i \in I$ challenges A for getting a valid message $M3$. A wins the game if it is able to generate a valid $M3$.
We say that the protocol is resistant against the fake car attack if the probability for A to win the game is negligible.

5.3. Security and Communication Assumptions

Our approach is designed under the following assumptions:

1. The communication between Car and Server is secure and reliable, thus providing mutual authentication, data confidentiality, data integrity and service availability;
2. The communication between Car and App is reliable (i.e. cannot be cut);
3. The pseudorandom number generator (PRNG), hash function and HIBS scheme are robust;
4. App securely hosts its HIBS secret key which cannot leak to third parties;
5. App manager is trusted, and securely hosts its HIBS master key which cannot leak to third parties;
6. Car and Server are provided with some private keys that cannot leak to third parties.

5.4. Enrolment Phase to Services

Before App is able to interact with Server, it is necessary that User first installs the appropriate App software for self-managing his/her personal data (cf. Section 4). To be the only one authorized to access the application, he/she needs to securely register the unlocking parameters such as some pin code, fingerprint... Then User has to enrol to the service (e.g. car sharing company) to become a member. Beyond the classical form to be filled in with personal information, credit card and driving license details, User has to securely register its User ID (pseudo-AS) along with his App manager ID for the Server to retrieve the appropriate HIBS public parameters. Users can subscribe as many services they want. The only condition is for the service to accept personal data management of customers through the application. Note that the same public parameters will apply for as many App as being managed by the App manager, and across the companies.

5.5. Mutual Authentication Phase with Consent Proof Generation

The protocol is depicted in Figure 1 based on the notations given in Table 1. In the first three exchanges, Car is a simple relay that permits connectivity between App and Server to support mutual authentication and consent transfer. First Server challenges App with a random $r1$ in $M1$. Then App generates a random $r2$, it encrypts $r1, r2$ and *pseudo-AS* with the public key of Server, it sends in cleartext the obligations and it sends a signature over the obligations, $r1$ and $r2$, to Server. After receiving message $M2$, Server checks that obligations *oblig* are compliant for the service to run legitimately, it deciphers $M2(c)$ with its private key and obtains $r1, r2$ and $pseudo\text{-}AS$, it checks that $r1$ matches $M1$ content, it goes through its database and identifies App/User based on the $pseudo\text{-}AS$ value. It then checks the HIBS signature in $M2$ thanks to $pseudo\text{-}AS$ and its related (App manager) public paramaters. In case of success (proving App is the claimed one), it registers both the obligations and the consent in the form of a signature so a to prove User's consent afterwards. Then Server sends back $r2$ value to Car, which is transferring the corresponding hash value to

App. App has only to check that $r2$ is matching the value in message $M2$. In case of success (proving Server is the claimed one), App is deriving a session key $Kr1r2$ based on $r1$ and $r2$ values for privately communicating personal data to Car. Car is able to derive the same key $Kr1r2$ as it also knows $r1$ and $r2$.

Table 1. Protocol notations

Parameter name	symbol
Pseudo of App to Server	pseudo-AS
Personal data	data
Obligation	oblig
Hash function	hash
Signing function	sign
Server public key	Ks
Application public key	Ka
Intruder public key	Ki
Random generated by the server	r1
Random generated by the app	r2
Hash of random r1 and r2	Kr1r2

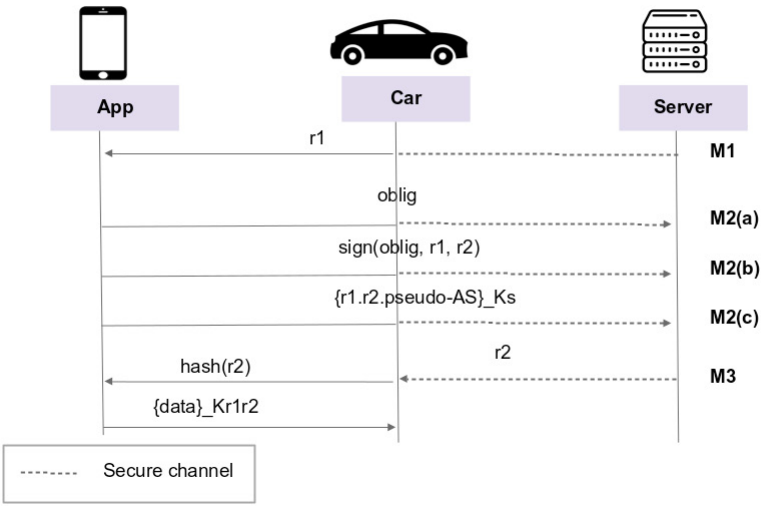

Fig. 1. Protocol interactions

6. Protocol Validation

In order to validate our protocol (i.e. resistance against Game 2), we use Automated Validation of Internet Security Protocols and Applications (AVISPA) [4] tool. AVISPA enables to validate the security properties of some security-sensitive protocols based on a specification expressed in the formal language High Level Protocol Specification Language (HLPSL). AVISPA requires modeling our challenge-response protocol relying on the following considerations:

1. AVISPA considers the Dolev-Yao threat model where the intruder is assumed to have full control over the network. The intruder is located over the unsecure link between App and Car, as a channel established as secure is

available between Car and Server. The objective is to validate the mutual authentication protocol between App and Server, in particular against game 2.
2. AVISPA does not support unclassical cryptographic schemes, like HIBS. For this reason, we model the HIBS signature as a classical asymmetric signature (e.g. RSA).

The full AVISPA code is available at http://www-public.tem-tsp.eu/~lauren_m/DOCUMENTS/Appendix-IoTConsentAVISPA.pdf.

The security properties declared in our protocol are presented in Algorithm 1, and includes the secrecy of $Kr1r2$ and the mutual authentication between App and Server.

Algorithm 1: Security properties to be validated by AVISPA

```
goal
    secrecy_of Kr1r2
    authentication_on app_service_r2
    authentication_on service_app_r1
end goal
```

In AVISPA, the intruder is assumed to have the knowledge listed in Figure 2, i.e. the names of agents App and Server, all the public keys Ka, Ks, Ki and its own private key ($inv(Ki)$), and the hash function type.

intruder_knowledge = a,s,ka,ks,ki,h,inv(ki)

Fig. 2. AVISPA intruder's knowledge

AVISPA outputs SAFE from On-the-fly Model-Checker (OFMC) back-end and Constraint-Logic-based Attack Searcher (CL-AtSe) back-end. This implies that AVISPA could not re-produce any attack on our proposed protocol.

7. Security and Privacy Analysis

Untraceability of App across multiple Servers (cf. section 5.1) is directly inherited from the HIBS properties and assumed robustness.

Additionally to the AVISPA results (validating resistance against Game 2), we provide below an analysis of our protocol against part of the attack scenarios put forth in Section 5.2.

Game 1: Privacy as an Indistinguishability Problem

Let us consider that the same $r2$ value is sent by the adversary A in messages $M2$ and even that the challenger is answering with the same obligations content (oblig in $M2(a)$). Even under these favorable conditions, the adversary is not able to guess whether two messages $M2$ are issued by the same App_j. Indeed, portions of $M2$, i.e. $M2(b)$ and $M2(c)$, behave like independent random numbers as both are applying different cryptographic functions (either encryption or signature) over some fields including $r2$ values that only App knows. As such, whatever the number of App_i learnt by A from message $M1$, A cannot extract any kind of information, and the game cannot succeed.
However, this reasoning holds only if the type or volume of personal data does not favor re-identification, implying to manipulate coarse grain and not fine grain data for tuning the car.

Game 3: Security - Fake Car

To masquerade as a valid Car, a fake Car has to get or guess the valid $r2$ value to provide the challenger with a valid hash value of it. On one hand, it is not possible for the fake car to guess the value as it is assumed that the PRNG function used for random generation at App side is robust. On the other hand, for getting value $r2$ from Server, it is necessary that the fake Car previously establishes an authenticated channel with Server. This requires the fake

Car to know a legitimate Car's private key, which is in contradiction with our assumptions. Thanks to the transitive authentication procedure, where App trusts Server to establish a secure channel only to legitimate Cars, the game cannot succeed.

8. Conclusion

This paper presents a technical approach for consent management, which goes in the same direction as the GDPR European regulation, i.e. to help supporting the accountability principle by providing the service providers with an undeniable cryptographic consent proof issued by users. Combined with a mutual authentication protocol, the resulting consent is strongly secure and should be stored at the service provider side for later proving its willingness to comply with the user's consent, in case of an auditing procedure or of disputes with users. The protocol was formally validated with AVISPA. The protocol was also analysed as privacy preserving with the support of pseudonymity and untraceability properties, and as secure against realistic scenarios illustrated in a car sharing use case.

References

[1] https://www.gartner.com/newsroom/id/3598917.
[2] http://www.privacy-regulation.eu/en/.
[3] Article 29 Working Party Guidelines on consent under Regulation 2016/679, Adopted on 28 Nov. 2017, Revised on 10 April 2018. https://ec.europa.eu/newsroom/article29/item-detail.cfm?item_id=623051.
[4] Alessandro Armando et al. The AVISPA tool for the automated validation of Internet security protocols and applications. *Computer Aided Verification*, 26(1):281285, 2005.
[5] Timothy Claeys, Franck Rousseau, and Bernard Tourancheau. Securing Complex IoT Platforms with Token Based Access Control and Authenticated Key Establishment. In *Int. Workshop on Secure Internet of Things (SIOT)*, Oslo, Norway, September 2017.
[6] Yang Cui, Eiichiro Fujisaki, Goichiro Hanaoka, Hideki Imai, and Rui Zhang. Formal Security Treatments for IBE-to-Signature Transformation: Relations among Security Notions. *IACR Eprint archive*, 26(1):23–30, 2007.
[7] Craig Gentry and Alice Silverberg. Hierarchical ID-Based Cryptography. *Int. Conf. on the Theory and Application of Cryptology and Information Security*, 4(120):1802–1831, 2002.
[8] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of Things: Vision, Applications and Research Challenges. *Ad Hoc Networks*, 7(120):1497–1516, 2012.
[9] Victor Morel, Daniel Le Métayer, Mathieu Cunche, and Claude Castelluccia. Enhancing Transparency and Consent in the IoT. *IWPE 2018 - International Workshop on Privacy Engineering*, 4(120):1802 – 1831, 2018.
[10] Ricardo Neisse, Gianmarco Baldini, Gary Steri, Yutaka Miyake, Shinsaku Kiyomoto, and Abdur Rahim Biswas. An Agent-based Framework for Informed Consent in the Internet of Things. In *2nd IEEE World Forum on Internet of Things (WF-IoT)*, pages 789–794, 2015.
[11] Yvonne O'Connor, Wendy Rowan, Laura Lynch, and Ciara Heavin. Privacy by Design: Informed Consent and Internet of Things for Smart Health. *Procedia Computer Science*, 113(120):653–658, 2017.
[12] Paul Ohm. Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization. *UCLA Law Review*, 57:1701, 2010.
[13] Scott R. Peppet. Regulating the Internet of Things: First Steps Toward Managing Discrimination, Privacy, Security and Consent. *Texas Law Review*, 93:83–176, 2014.
[14] A. Shamir. Identity-Based Cryptosystems and Signature Scheme. *Advances in Cryptology*, 196(2):47–53, 1984.
[15] Sandra Wachter. Normative Challenges of Identification in the Internet of Things: Privacy, Profiling, Discrimination, and the GDPR. *Elsevier Computer Law and Security Review*, 34(3):436–449, 2018.
[16] Rolf H. Weber. Internet of Things New security and privacy challenges. *Computer Law and Security Review*, 26(1):23–30, 2010.