A Comparison of Payment Schemes for the IoT

Jens-Matthias Bohli,[1] Aljoscha Dietrich,[2] Ronald Petrlic,[3] Christoph Sorge[2]

Abstract: Technologies for the IoT have reached a high level of maturity, and a large-scale deployment will soon be possible. For the IoT to become an economic success, easy access to all kinds of real-world information must be enabled. Assuming that not all services will be available for free, an IoT infrastructure should support access control, accounting, and billing. We analyze available access control and payment schemes for their potential as payment schemes in the IoT. In addition to security and privacy, we discuss suitability for direct client to sensor communication and efficiency. We show shortcomings of existing protocols that need to be addressed by future research.

Keywords: Internet of Things, Tokens, Survey

1 Introduction

Despite its success, the Internet of Things (IoT) is still limited in scope. For the IoT to become an economic success, and to incentivize deployments of this technology [BSW09], it must be possible to offer paid services—either in the form of a micro-payment per access or based on a flat rate. The IoT infrastructure should therefore support accounting and billing in addition to access control. In this paper, we analyze available access control and payment schemes for their suitability for payment in the IoT. We consider objects that are publicly available to any paying user—for example, sensor-equipped parking lots that allow the user to find or reserve (and pay for) the parking space remotely.

System Model: The involved parties are the object **(O)**, an IoT platform provider **(P)**, where **O** is registered, and users **(U)** wishing to access **O**. We assume **O** and **U** to already be registered with **P**, and to have a shared key to establish a secure channel with **P**.

We consider a scenario with these three parties in which **U** is interested in using a service of **O**. We assume **O** is open to any **U** but demanding a payment. The platform provider is also the payment provider where **U** and object owner have account. A trivial solution would be to route every request and response between **U** and **O** over the platform as a proxy. In that case, **P** can make the according payments between the accounts. However, this solution requires high trust in **P** and will not work if multiple federated IoT platforms exist and **O** and **U** are registered at different providers. It also does not offer any privacy from the platform and direct local communication between **U** and **O** will not be possible. We therefore consider that **U** and **O** exchange payment information in the form of a token, that can afterwards be used to cause the necessary payment flows.

From **P**'s perspective **O** and **U** are untrusted in terms of payment. We assume **O** and **U** to be malicious and to collude in order to achieve a financial gain. This leads to strong

[1] Hochschule Mannheim, John-Deere-Straße 85, 68163 Mannheim, j.bohli@hs-mannheim.de
[2] CISPA, Saarland University, Campus E9 1, 66123 Saarbrücken, firstname.lastname@uni-saarland.de
[3] Commissioner for Data Protection Baden-Württemberg, Germany, petrlic@lfdi.bwl.de

accountability requirements: the object owner should be able to claim an access if and only if a paying **U** had access to **O**. On the other hand, **U** wants to hide when, where, and which services he consumes. We assume **P** and **O** to be a curious adversary. In the following, we will analyze existing protocols according to the following properties.

1. **Accountability** Accountability analyzes the provability of the object owner's payment claim. As the object owner and **O** are not trusted, the object owner should be able to prove cryptographically the reception of a payment token.
2. **Supported Payment Schemes** Here the protocols shall be analyzed regarding their support of pre-paid, post-paid or flat-rate tariff schemes. Secure transfer of tokens between **U** and **O** is insufficient if a flat-rate tariff is offered to users: **U** and **O** might collude to the disadvantage of **P** does not cost **U** anything while **P** will usually benefit from additional service usages. Additional users could copy their tokens and share them with others. A system suitable for the flat-rate payment model must mitigate these threats, e.g. by suitable accounting to detect this misbehavior. In addition, tokens generated for flat-rate payments need to either have a validity period, or be revocable.
3. **Privacy Support** In principle, access control and payment do not require identification of the principals involved. Pseudonymity is trivially achieved, so we investigate whether the scheme supports unlinkability. We distinguish
   - Unlinkability across objects/owners: Different owners cannot recognize from their received payment tokens, if they have been accessed by the same **U**.
   - Full Unlinkability: From the payment tokens received by a single **O**, it is impossible to link several access requests originate to the same **U**.
   - Unlinkability towards platform: **P** has no knowledge about which objects were used by a certain **U**, and cannot link different **O** usages (e.g., to a pseudonym).
4. **Online/Offline** discusses if the exchange of tokens can be direct and local between **U** and service, without a connection to **P**. Still, there are multiple possibilities for token generation. One can distinguish tokens (a) generated by the platform specifically for **O**, (b) generated by platform for universal use, or (c) directly generated by **U**.
5. **Efficiency** Efficiency discusses the suitability of the scheme for an IoT scenario where **U**'s device and **O** are resource limited.

1.1 Kerberos

Kerberos is a AAA protocol based on symmetric cryptography. It uses tickets which, in our terminology, confirm **U**'s and **O**'s identity. **P** plays the role of the authentication server (AS) and the ticket granting server (TGS). The AS verifies **U**'s credentials and issues a ticket granting ticket (TGT) which **U** can use at the TGS to obtain a ticket for a specific **O**. A Kerberos ticket is a single-use token authenticator, only **P** can issue tickets.

- **Accountability** The original protocol does not provide accountability: To prove receipt of the ticket *to **P***, **O** could present the session key from the ticket. The original design of Kerberos does not require the TGS to keep that session key after generating it. This aspect could be changed even locally on the TGS (**P**). Moreover, in Kerberos v5, the ticket contained in the KRB_TGS_REP message is only encrypted with the shared key between the TGS and the service (object). Even a service acting as a passive adversary can obtain the ticket by eavesdropping. The attack could be prevented by encrypting the whole KRB_TGS_REP message. Thus, in that case a certain accountability is given. If the service provider knows the session key from the ticket, this proves that the client did request a service from the service provider. As Kerberos is based on symmetric cryptographic primitives, neither **O** nor **P** can prove the access to a third party.

- **Supported Payment Schemes** Kerberos can support pre-paid schemes by charging for issuing tickets from the TGS, and postpaid by the provided accountability. The protocol is, in principle, suitable for flat-rate payments. This assumes that tickets are intended (and accepted) for one-time use only. Should the object owner collaborate with **U**, this can easily be recognized as the tickets are specifically issued for a given service.
- **Privacy Support** Kerberos does not support anonymity towards **P**, which must know the identities of the services (objects) it issues tickets for. Tickets contain **U**'s identity, so different tickets can be linked to the same person by different **O** owners. However, Kerberos could be altered to use new identities for each service usage, thus achieving full unlinkability on the object owners' side.
- **Online/Offline** The protocol allows the actual service request to be offline. However, **U** must obtain a ticket to the respective **O** beforehand, by communicating with **P**.
- **Efficiency** Kerberos is purely based on symmetric cryptography and therefore suitable for IoT devices. A drawback is the need for time syncronization. Ladon [AJHH12] is a variant of Kerberos—taking into account the characteristics of resource-restricted devices; especially, removing the clock synchronization requirement of Kerberos.

1.2 Bitcoin

Bitcoin is essentially a system that allows transfers between arbitrary, self-generated accounts. A transfer is confirmed by a digital signature. To prevent double spending, all transactions are broadcast in a peer-to-peer network. They are collected in transaction blocks, which are added to the complete (public) transaction history, the blockchain. Cryptographic proofs of work, which depend on the existing blockchain and the newly added transactions, are generated by participants. Bitcoin is suitable only as a payment scheme.

Bitcoin does not provide authorization tokens per se. However, **P** with a known Bitcoin address could transfer a small amount of Bitcoin to its users, who could then spend a part of that amount when interacting with **O**. The latter transactions could be interpreted as tokens. Alternatively, if access to **O** is open to all paying users, payment could be done directly with Bitcoin.

- **Accountability** Given the public nature of the blockchain, any Bitcoin amount paid to **O** can be proven later on.
- **Supported Payment Schemes** The above-mentioned construction does not allow an actual flatrate. However, one satoshi (the smallest Bitcoin unit) is worth 10^{-8} Bitcoin, i.e. less than 0.001 US cent. If **U** has to spend this amount to prove authorization for service usage, a flatrate scheme is achieved in practice.
- **Privacy Support** Bitcoin is not designed to achieve unlinkability. While new identities can be easily created, any transactions create an (undesired) links.
- **Online/Offline** Payments need to be announced to the blockchain, thus **U** and **O** need to be connected to the platform during the payment.

- **Efficiency** If used as an authorization check as sketched above, one signature verification is required only. Verification of an actual payment requires an additional check for double spending. Most importantly, a payment should only be considered as confirmed once it has been incorporated into the blockchain, and additional blocks have been added afterwards. On average, addition of a new block takes about 10 minutes. Thus it is not suited for the scenario at hand.

1.3 Distributed Privacy-Preserving Access Control (DP^{2}AC)

DP^{2}AC [ZZR12] is designed for a sensor network owned by a *single* party, but with *multiple* users. **U** buys (anonymous) tokens beforehand and can spend those tokens to access services from the nodes *directly*. Access control is thus enforced by the network owner itself, who is assumed to be semi-honest: it abides by the protocol, but is interested in **U**'s access patterns. Users, on the other hand, are assumed to compromise a few sensor nodes if it helps them to reuse their tokens. The scheme is straight-forward, making use of CHAUM'S blind signature protocol. The sensor nodes only need to perform an efficient RSA signature verification to check the authorization of a request. The main challenge is token reuse detection. The authors come up with five approaches, where some nodes are "witnesses" of already spent tokens. This might be impractical in terms of storage and communication overhead. The authors suggest using a Bloom filter in order to overcome the issue of sensor nodes' restricted buffer size.

DP^{2}AC has been designed for *single-owner* sensor networks. However, an adaptation to the scenario at hand—with **P** responsible for handling the payment—is easily possible, with **P** assuming the role of the owner. However, it requires communication between the objects of different owners for token reuse detection.

- **Accountability** To claim the use of **O**, its owner shows the received token. Since the tokens can only be generated by **P** and there exist methods for reuse detection, accountability is achieved. As tokens are neither signed nor encrypted by **U**, an attacker may be able to intercept and reuse them. However, the required modification is straightforward.
- **Supported payment schemes** This approach's design follows a pre-paid scheme. The users need to buy their tokens from **P** before their usage. Thus a postpaid tariff is not applicable. A flat-rate model could be imaginable by introducing some tokens which can be reused for a specified period or could be revoked. However, such an approach entails three problems: privacy would no longer be guaranteed as access requests would be linkable to each other based on the token, the token could easily be shared by different users, and a malicious cooperation between an object owner and **U** it not detectable.
- **Privacy Support** Unlinkability across objects and owners, and for service usage at the same object owner, is achieved. The owners cannot establish a link between **U** and the spent tokens; from their perspective, tokens are essentially random numbers signed by the platform owner. Due to the properties of blind signatures, as applied in this protocol, the platform owner cannot link the token to a certain owner either. If DP^{2}AC) is applied in an individual network with only few users in the system, though, such a linking could be done by the respective **O**, as the tokens are bought from the same party where they are spent afterwards. This issue, however, does not play a role in the scenario at hand.

- **Online/Offline** Offline usage is possible because tokens are generated in the beginning. Also the tokens allow universal use and are not restricted here. However, a more fine-grained authorization is not supported, as each anonymous **U** has same privileges.
- **Efficiency** The main problem of this approach is the reuse detection for which the authors developed 5 schemes. Still it is uncertain how these schemes would work in a large IoT environment, as they cause both communication and storage overhead.

1.4 Priccess

Priccess [HBZ^{+}11] is designed for a similar setting as DP^{2}AC. The users should here communicate within a WSN without revealing their identity. This is achieved by grouping users with similar access privileges. **U** gets access to **O** by signing a request with a ring signature scheme, unlinkable to **U**, but only to a set of group members' public keys. Hence the query verifies as valid without revealing the identity of **U**. The scheme is designed for *single-owner* sensor networks. Its applicable to the scenario at hand, though.

- **Accountability O** can prove his payment claim by showing the properly signed query. Thus he can prove that he received a valid request from a group.
- **Supported Payment schemes U** gets assigned to a group with his privileges by **P**. The group membership could be used to model the access to the paid services. But since group members are undistinguishable, pre-paid and post-paid schemes are not applicable. This applies also to flat-rate tariffs because collusion between an object owner and **U** to **P**'s disadvantage is possible. However, with sufficiently small groups, anomalous behavior is detectable. Unfortunately, this comes at the cost of privacy.
- **Privacy Support** This protocol meets all privacy degrees mentioned. The ring signature leads to unlinkability towards individual objects, collaborating object owners, and **P**.
- **Online/Offline** A set up phase including group assignment is needed. Later the signed requests can be generated by **U** alone but are limited to the group's privileges.
- **Efficiency** For efficiency, cryptographic protocols based on elliptic curves are chosen considering limited energy, processing and storage. Scalability is problematic, though: When users join, this needs to be communicated to all objects.

1.5 Accountable and Privacy-Enhanced Access Control (APAC)

APAC [HCG15] preserves the users privacy by hiding the data access from the sensor network owner. At the same time, a fine-grained access control is enforced. Cheating users can be identified, however, only by a collaborating network owner and a "law authority". Users are parts of groups in the proposed protocol, with groups managed by different sensor network providers. Agreements between different providers allow **U** from one group to access data from nodes belonging to another group. A modified group signature variant based on [CG05] is employed in order to perform user authentication and authorization.

- **Accountability** The scheme offers the possibility for the authority, i.e. **P**, to open the group signatures of the requests, therefore to link a request to **U**.

- **Support of Payment Schemes** The provided accountability allows for post-paid schemes. It's less suitable for pre-paid tariffs as the access tokens, i.e. signed requests, are directly computed by **U**. Even though not mentioned in the paper, the provision of a flat-rate model should be possible by means of the introduction of a flat-rate group, i.e. a group that is attached with the privilege to access data on a flat-rate basis. The given accountability allows for identifying fraudulent users afterwards.
- **Privacy Support U** cannot be tracked individually by **O**/service providers. For any data request, **O**/in the service only learns that some **U** from a certain group requests access. This holds even for collusion of multiple **P**. **P** is however capable of opening the signatures on the request and can identify the users.
- **Online/Offline U** can compute own access tokens. An ideal offline capability.
- **Efficiency** The used cryptographic primitives such as the group signatures are on the expensive side and require the verification to be done at a gateway node, rather than **O**.

2 Conclusion

	Kerberos	Bitcoin	DP^{2}AC	Priccess	APAC
Accountability	–$^{1.1}$	+	+	+	+
Payment schemes (Pre-/Postpaid/Flat)	+/+/+	+/+/+	+/–/+	–/–/–	-/+/+
Token Generation	P	P	P+U	U	U
Privacy Support (acr. objects/full/platform)	+/+/–	–/–/–	+/+/+	+/+/+	+/+/–
Offline Support	–	–	+	+	+
Efficiency	+	–	–	–	–

Tab. 1: Comparison of the analysed protocols

We summarize the results in Table 1. The paper has identified some requirements for access tokens for smart objects and presented initial ideas for possible solutions. An access control suite for smart objects will have to be flexible enough to provide access tokens for multiple purposes and support a wide class of the objects from unprotected, computationally restricted to protected and unrestricted objects. While appropriate solutions for all individual requirements exist, their combination is still an open research issue.

References

[AJHH12] J. Astorga; E. Jacob; M. Huarte; M. Higuero. Ladon: end-to-end authorisation support for resource-deprived environments. Information Security, IET, 6(2):93–101, June 2012.

[BSW09] J. Bohli; C. Sorge; D. Westhoff. Initial observations on Economics, Pricing, and Penetration of the Internet of Things Market. ACM CCR, 39(2):50–55, 2009.

[CG05] J. Camenisch; J. Groth. Group Signatures: Better Efficiency and New Theoretical Aspects. In SCN, Jgg. 3352 of LNCS, Seiten 120–133. Springer, 2005.

[HBZ$^+$11] D. He; J. Bu; S. Zhu; S. Chan; C. Chen. Distributed Access Control with Privacy Support in Wireless Sensor Networks. IEEE TWC, 10(10):3472–3481, October 2011.

[HCG15] Daojing He; S. Chan; M. Guizani. Accountable and Privacy-Enhanced Access Control in Wireless Sensor Networks. IEEE Trans. on Wirel. Comm., 14(1):389–398, Jan 2015.

[ZZR12] Yanchao Z.; Yanchao Z.; Kui R. Distributed Privacy-Preserving Access Control in Sensor Networks. Par. and Dist. Syst., IEEE TPDS, 23(8):1427–1438, Aug 2012.