20 October 2020 Study · Digital Identity · Privacy & Crypto · 17 min read

IoT Devices in Service Compositions

Archived. This content is no longer updated. It remains available for reference.

Source https://technology2.a-sit.at/en/publications/iot-devices-in-service-compositions/

Online applications are often composed of services provided by multiple service providers. In some cases, a service consumer may want to use a service anonymously to preserve the privacy of his identity data. This becomes even more problematic, when IoT devices are included in a composite service, as these devices could give a link to the service consumer's identity. However, authentication of included IoT devices is necessary to prevent adversarial attacks.

In this work, we investigated different anonymous authentication schemes regarding their suitability for authentication of IoT devices to participate in a service composition system anonymously. We introduce a registration and authentication scheme based on hash-chain authentication, which provides anonymous authentication of IoT devices against a service composition system for a determined amount of requests.

1. Introduction

Service compositions are implemented through the interplay between actors of different organizations. Internet of Things [1] (IoT) technologies have advanced the way of how ubiquitous computing is held today. The continuous development of these technologies enable to integrate IoT devices in more and more IT systems, such as for instance service composition systems. These IoT devices are capable of transmitting and exchanging large amounts of classical and sensitive data with other devices over the network. Data generated and disseminated by IoT networks could contain sensitive temporal data. User anonymity is a highly desirable property in such a network. However, protecting the privacy of identity information about service consumers becomes an increasingly challenging part, especially for IoT systems.

To protect the privacy of identity data during service consumption, many related research has been conducted in the area of anonymous authentication ([2], [3], [4], [5], [6]). Anonymous authentication schemes are mostly based on cryptographic techniques, which usually require heavy cryptography processing, especially those the ones that rely on public key crypto. State-of-the-art anonymous credential schemes such as ([7], [8], [9], [10], [11], [12]) often invoke costly zero-knowledge proof of © A-SIT, Partner für Sicherheitsfragen knowledge, and as such, they are not satisfactory for weak IoT devices such as sensors, smart cards, e-passports, and vehicular devices. Specifically, the RSA-based schemes (e.g., [13,

14]) entail expensive zero-knowledge range proofs, which generate tens of kilobits in communication and tens of modular exponentiation operations. The majority of the IoT devices by nature are resource constrained in the sense that they have limited computation and storage capability. This creates a challenging task to design security protocols that should be lightweight in computation and communication costs, while keeping high security and maintaining the functionality attributes needed for designing the protocols in IoT environment.

In this work, we investigate the functionality of different schemes providing anonymous authentication with a specific focus on schemes that can be deployed to integrate IoT devices via a lightweight communication. As a result, we provide an anonymous authentication scheme that enables to include IoT devices in service composition systems.

2. Anonymous Authentication Schemes

In the past many research was done in the area of anonymous authentication. In this section, we list the most common cryptographic methods that are used. Derived from these methods, we determine the scheme that fits best for our purpose.

2.1 Group Signatures:

Group signatures make it possible to anonymously issue digital signatures on behalf of a group of users. In existing centralized approaches, some form of bootstrapping node, group authority or central organization (typically in the form of an on-line trusted third party) is assigned the responsibility of generating and distributing digitally signed group certificates [15]. It is not possible to make a connection between different signatures and a user during signature verification. Thus, a user can demonstrate group membership by, for example, proving knowledge of the certificate's associated private key. At the same time, group signatures offer the possibility of limiting possible misuse of this functionality, for example by withdrawing signature keys. Group signatures have the additional feature that the anonymity of a signer can be revoked (i.e., the signer can be traced) by a designated group manager.

A group signature attains unlinkability of signatures generated by the same signer. That is, a signature can only be recognized as valid, but cannot be attributed to any individual signer. While group signatures provide a mechanism for anonymous authentication, it may require cryptographic processing that cannot be easily conducted by IoT devices.

2.2 Ring Signatures:

A ring signature is a type of signature in which several possible senders are merged in the signature. They enable a user to sign a message so that a ring of possible signers (of which the user is a member) is identified, without revealing exactly which member of that ring actually generated the signature. A ring signature consists of several mixed signatures, which makes this signature unique and unmistakable and authorizes the transaction. In contrast to group signatures, ring signatures are completely "ad-hoc" and do not require any central authority or coordination among the various users. Ring signatures allow greater flexibility: no centralized group manager or coordination among the various users is required.

2.3 Threshold Signatures:

Threshold Signature Scheme (TSS) enables the definition of a flexible threshold policy. This technology replaces traditional central signature creation with a distributed computation, in which multiple actors jointly perform the signature creation. Every actor participating in the signature creation holds a share of a private signing key. Through the determination of a desired threshold, it is possible to define the number of required actors, which at least have to create a signature jointly [16]. In order to achieve such a goal, threshold signature schemes based on key sharing and agreement techniques are usually employed (Zhou and Haas, 1999). However, these schemes also have certain drawbacks. Most of them in relation to the need of having an honest dealer in charge of generating a signing key, dividing it into shares and secretly distributing them.

2.4 Zero Knowledge Proof:

ZKP is an interactive identification protocol, which enables a prover P to prove his identity polynomially many times to a verifier V without allowing V to misrepresent himself as P to someone else. The proof of identity is either accepted or rejected in real time, and as a result, the requested access is granted or rejected. The scheme provides lightweight identification and proves to be suitable for low-end systems with limited processing power, such as smart card technologies. This property makes ZKP suitable for IoT devices.

With carefully preselected parameters, the ZKP scheme is about two orders of magnitude faster than RSA-based identification schemes. The scheme assumes the existence of a trusted center who is involved in publishing a modulus m, which is the product of two large prime numbers of the form 4 r +3. Such moduli are used in a variety of cryptographic applications, and their most useful property is that –1 is quadratic non-residue whose Jacobi symbol is +1 (mod m). After publishing m, the center can be closed. The ZKP identification scheme relays that a prover P proves to a verifier V that he knows whether a certain number is a quadratic residue or quadratic non-residue (mod m) without revealing a single bit of information.

In practice, however, ZKP are rarely used for authentication because they usually require a high level of interaction, caused by the necessary exchange of multiple messages. Thus, they are prone to replay attacks.

2.5 Verifiable Common Secret Encoding:

Verifiable Common Secret Encoding (VCSE) construct incorporated with the formation of dummy group members that obfuscate the real identity of the device during authentication. Such an approach is based on public key cryptography. We recall the definition of VCSE of Rasheed et al.

[1]: For a group of z users, each with its public/private key pairs (Pub i, Pr i), the verifiable common secret encoding is constructed as (Pub 1 (x), Pub 2 (x), Pub 3 (x),..., Pub z (x)). Any member i of the group can decrypt its corresponding encoded message Pub i (x) with its private key Pr i, to obtain x ′ . It encrypts x ′ with the other z -1 public keys to obtain its own copy of the encrypted secrets (Pub 1 (x ′), Pub 2 (x ′), Pub 3 (x ′),...,Pub z (x ′)). It then verifies that these secrets match those that were received. If they all match up, it accepts the secret value x ′ as x and proceeds with the remaining protocol steps. The IoT device identity is hidden within a group of randomly chosen z members. Members of the group act as dummies to obscure the real identity of the device while performing authentication.

However, since VCSE is based on public key cryptography, this represents an issue for IoT devices with constrained processing power.

2.6 Hash-Chain Authentication

Hash-Chains (HC) represent a mechanism to produce various One-Time-Passwords (OTP) by applying a hash-function repeatedly on a secret. Since the sender transmits a hash-chain element based on the input of a secret, he delivers a proof without revealing the secret. Hence, Hash-Chain authentication could also be considered as a type of ZKP. They were first introduced by Lamport

[17] and aim to safeguard password schemes from eavesdrop and replay attacks.

Definition (Hash-Chain). Authentication via hash chains requires steps for its generation and verification. The following explanation recalls the syntactic definition of hash-chain suggested by Bicacki and Baykal [18].

```
Generate: Hash [N-i ] (secret) → (Pi)
```

To generate a hash-chain, a hash-function is applied N times, where N represents the number of allowed operations for authentication. Thus, the number of performed repetitions of a hash-function represents the number of generated OTP.

The first OTP is produced by applying the hash-function N times:

```
P0 =Hash [N] (secret)
```

The second OTP is generated by applying the hash-function N – 1 times:

```
P1 =Hash [N-1] (secret)
```

Hence, this results in the following general formula:

```
Pi =Hash [N-i] (secret)

Verify: Pi =Hash(Hash [N-i-1] (secret)) = Hash(Pi+1)
```

The authentication via hash-chain requires a sender to transmit the OTP P0 to the authentication server, which performs the verification algorithm to validate the OTP.

2.7 Conclusion of anonymous authentication schemes:

Table 1 summarizes the advantages and drawbacks of all anonymous authentication schemes mentioned above. Derived from these results it becomes apparent that ZKP and the hash-chain authentication scheme is suitable to implement an anonymous authentication for IoT devices.

Authentication Scheme	Advantages	Drawbacks	Suitable for anonymous IoT authentication
Group Signatures	Unlinkability, anonymity, revokeable anonymity by trusted group manager	Relatively heavy cryptographic processing	x
Ring Signatures	Unlinkability, anonymity, no need for any trusted third party	Anonymity of a signer cannot be revealed, relatively heavy cryptographic processing, requires involving third party signers	x
Threshold Signatures	Unlinkability, anonymity	Requires a specific amount of signers to reach the threshold, relatively heavy cryptographic processing	x
Zero Knowledge Proof	Anonymity, lightweight communication, no need for involving any third party	High amount of exchanged messages, causing weakness against replay attacks	≣
Verifiable Common Secret Encoding	Unlinkability, anonymity	Relatively heavy cryptographic processing, requires involving third party	x
Hash-Chain Authentication	Anonymity, lightweight communication, no need for any trusted third party	Requires initial exchange of hash-chain (One-Time- Passwords)	≣

Table 1: Authentication Schemes including their advantages/drawbacks and suitability for anonymous IoT authentication

3. Anonymous Registration at Service Composition Systems

In our concept, we use the hash-chain authentication scheme, which turns out as most suitable for the implementation on IoT devices. Since we focus on the anonymous integration of IoT devices into service composition systems, we provide a brief explanation of the typical architecture of such systems in the following paragraph.

Service consumers have to register at a service registry in order to be authorized to use a composite service. For privacy protection reasons, no identity data related to the service consumer must be transmitted to the service registry. For this purpose, the client also referred to as service consumer initially has to authenticate at a Trusted Third Party (TTP), which provides an authentication mechanism based on hash-chain authentication. In order to authenticate, the client (smartphone, laptop etc.) transmits the required hash-chain element accompanied by his public key, to the TTP.

After a successful authentication, the TTP sends a signed token to the client that confirms a sufficient level of security regarding the authentication without containing identity data about the service consumer. Next, the client relays this token to the service registry to consume a composite service at the service composition system. The service registry can verify the authenticity of the TTP that has issued the token, which ensures trust in the authentication.

To verify a valid ownership of this token by the client, the service registry sends a data-to-be-signed (DTBS) referred to as challenge to the client, which in turn signs this data using its private key. After validating the challenge, the service registry authorizes the client to use a composite service.

3.1 Anonymous Registration of IoT Devices

To include IoT devices into service compositions, a client has to register them via an initial registration. The client device can be a smartphone/tablet or laptop/PC, which provides a service for IoT registration and authentication. The registration requires the client to send a registration request via the registration service to a TTP. In the course of this registration, the TTP may require an authentication of the client device based on desired authentication types.

When the TTP receives the registration request and optionally successfully authenticates the client, it sends a request to the hash-chain generator (HCG) to trigger the generation of a hash-chain with the length of N, where N represents the number of one-time-passwords (OTP) issued to the client. The generation of a hash-chain is illustrated by Figure 1. The number can vary depending on the security provided by the performed authentication. This means, the TTP determines the number of allowed authentications through a hash-chain, until the authentication at the TTP has to be performed again, to request a new hash-chain for authentication at the service composition system.

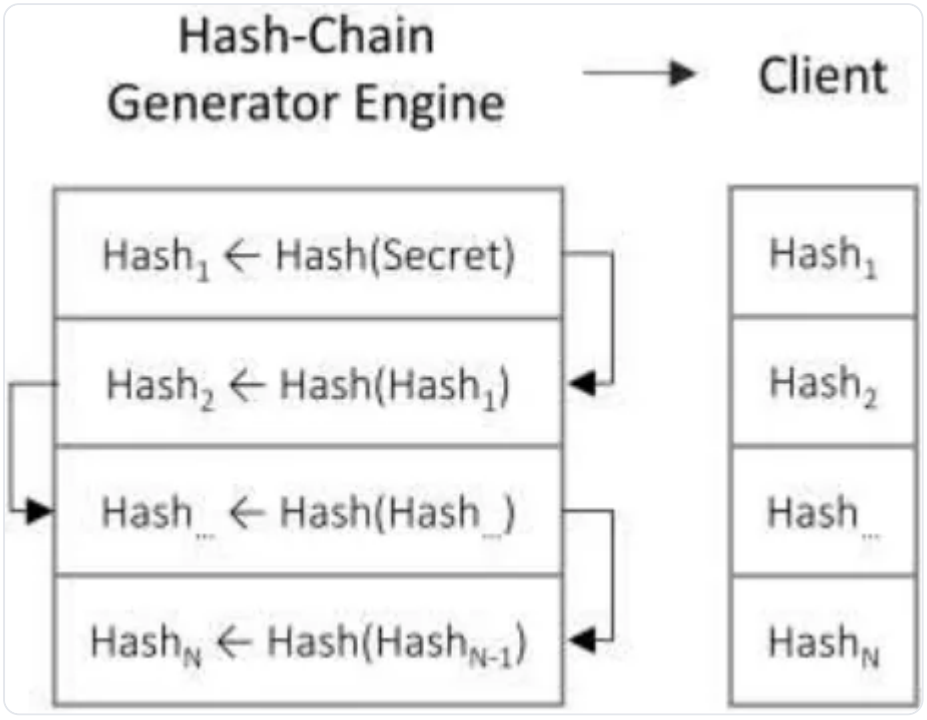


Figure 1: Generation of a hash-chain consisting of N elements

The TTP stores the hash-chain together with a public key of the client and transfers the encrypted hash-chain to the service consumer to finish the registration. The service consumer relays this hash-chain to any IoT devices that he wants to register. This can be done by various connection types, which are typically offered by IoT devices (Bluetooth, NFC, etc.). The whole process is illustrated by Figure 2.

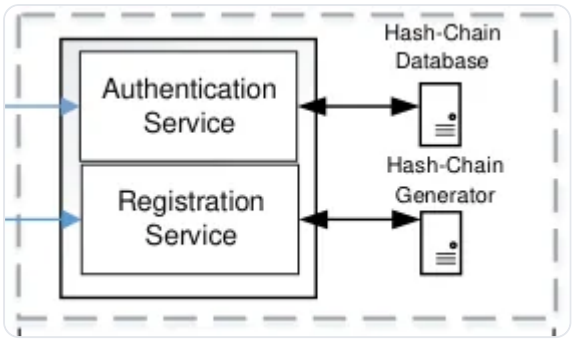
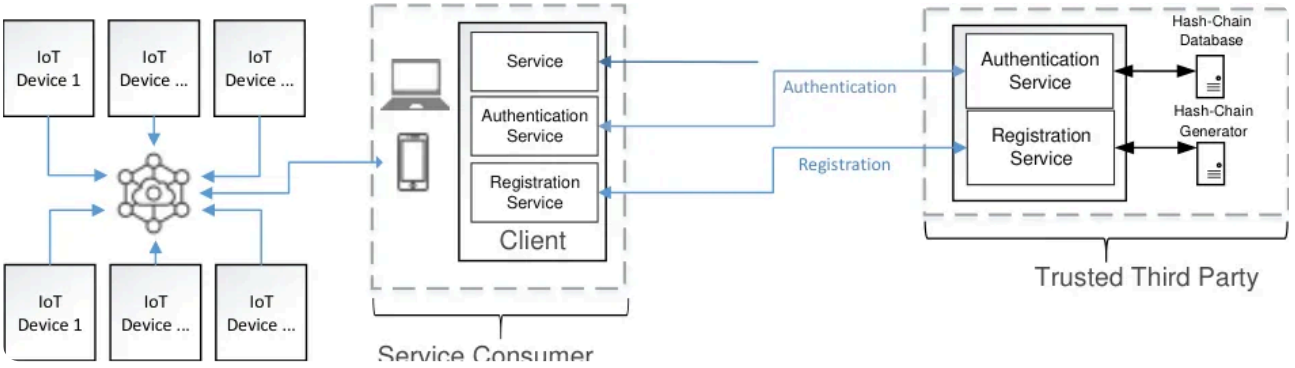
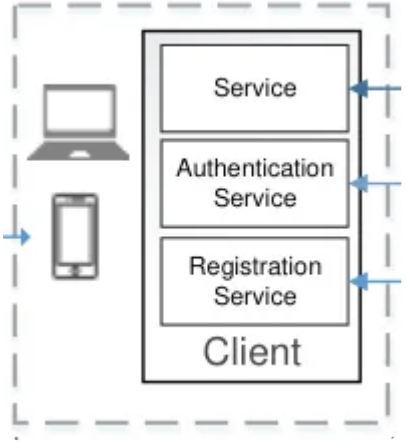


Figure 2: Registration of IoT devices via a client device





4. Anonymous Authentication of IoT Devices

Registered service consumers have to authenticate at the service registry to be authorized to use a composite service. For this purpose, the client sends a service request including its public key to the TTP, which uses this key to identify the related hash-chain stored in its database.

Afterwards, the TTP returns the hash-chain element with the index N, where N represents the length of the hash-chain. Next, the client relays this element to the IoT device, for which an authentication is needed. This IoT device forwards the hash-chain element with the index N-1 to the authentication service provided

by the client device, which in turn relays this element to the TTP.

Finally, the TTP performs the verification by applying the hash-function initially used to generate the hash-chain on the received hash-value. If the result matches the hash-chain element N, the authentication of the IoT device is successful. Both, the TTP as well as the IoT device (optionally) delete the hash-chain element N from the hash-chain. A client can repeat this authentication until no hash-chain element is remaining. In this case, he has to re-authenticate at the TTP and the service registry to obtain a new hash-chain. The whole process flow regarding the registration and authentication is illustrated by Figure 3.

IoT Device	Col2	Col3	Col4	Trusted Third Party	Col6
	Regist Auth HCElement[n-1] HCElement[n] Auth. Token	Auth. Token ration successfully Service Request + PKClient HashChain ←Identify(PKClient) HCElement[n] Verify(HCElement[n-1]) entication successfully HCElement[n-1]	Auth. Token ration successfully Service Request + PKClient HashChain ←Identify(PKClient) HCElement[n] Verify(HCElement[n-1]) entication successfully HCElement[n-1]	Auth. Token ration successfully Service Request + PKClient HashChain ←Identify(PKClient) HCElement[n] Verify(HCElement[n-1]) entication successfully HCElement[n-1]	Auth. Token ratic successfully Ser Request + PKClie HashChain ←Identify(PKClie HCElement[n] Verify(HCElemer 1]) entication successfully HCElement[n-1]

Figure 3: Registration and authentication flow

5. System Architecture

This section provides the whole architecture including the components needed to provide an anonymous authentication against a service composition system. Figure 4 illustrated the system architecture. We briefly summarize the tasks performed by each actor in the following paragraphs.

Service Consumer : This represents any actor that wants to use a composite service provided by the service composition system. Initially, they have to authenticate at a Trusted Third Party, which provides reliable means of authentication. Afterwards, the service consumer proves a successful authentication to the service registry to register at the service composition system.

IoT Device 1-n: These components represent any IoT devices that are included in a service composition system. A service consumer has to register these devices by obtaining a hash-chain for each device, which enables an anonymous authentication.

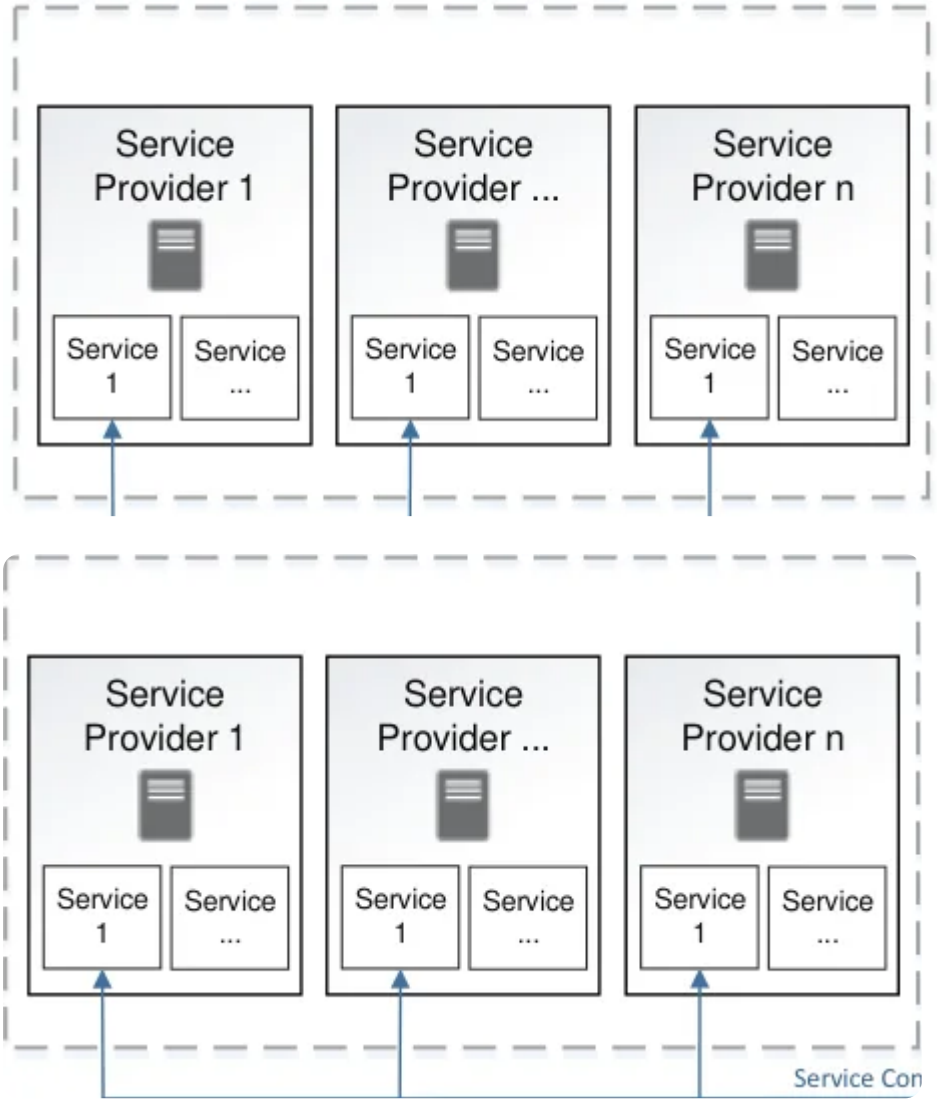
Trusted Third Party: The Trusted Third Party (TTP) represents a trustworthy authority that provides means for authentication, which can be based on legal standards. A service consumer has to register

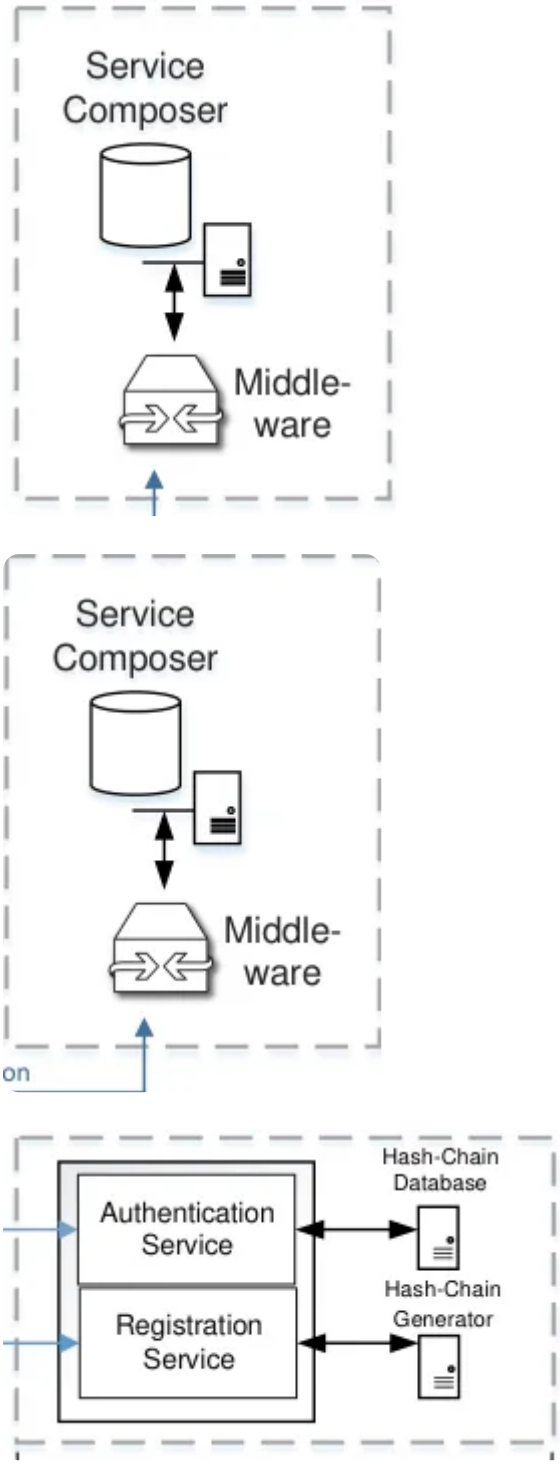
his client device as well as desired IoT devices at the TTP before he is able to consume a composite service at the service registry.

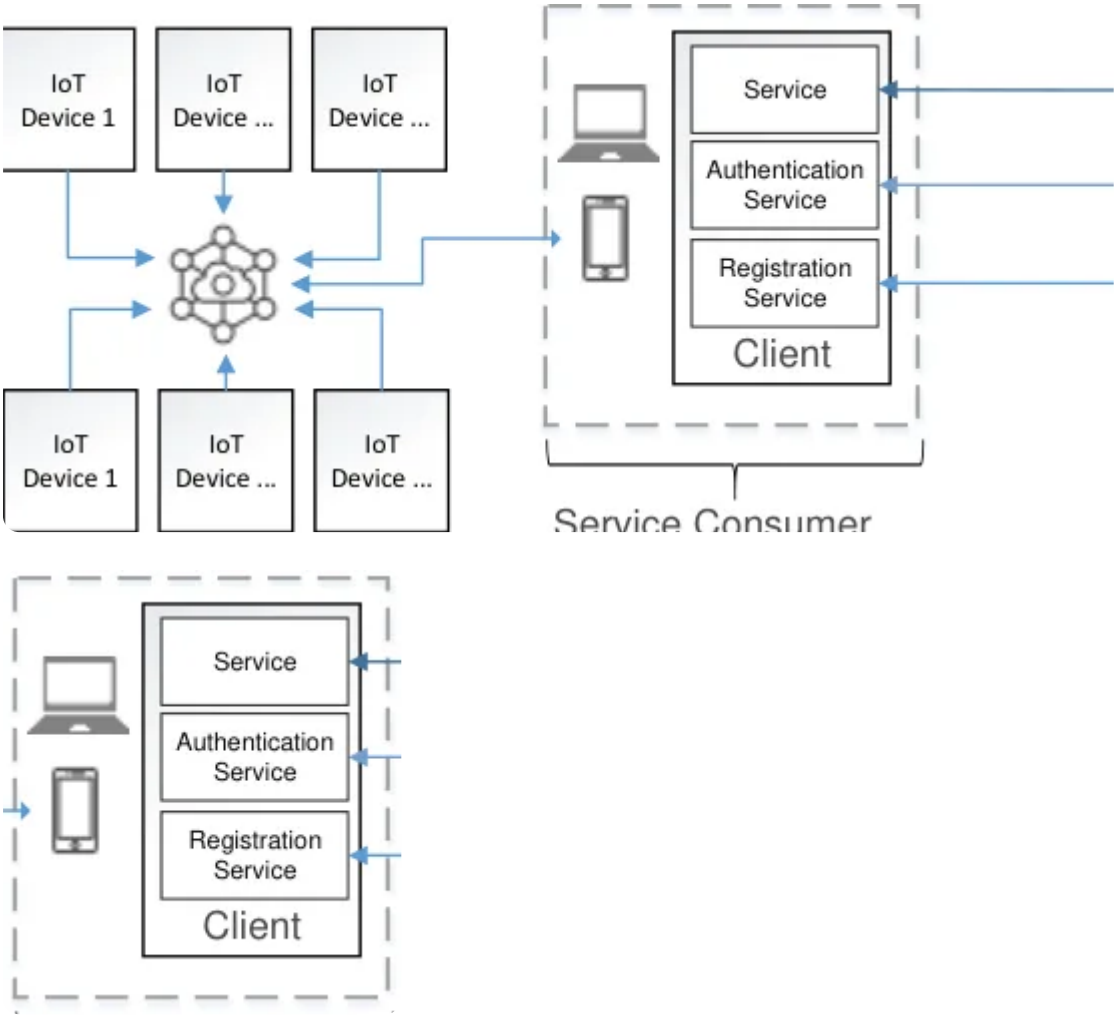
Service Registry: The service registry consists of several components that provide necessary functionality needed to perform a service composition. A service consumer has to authenticate against this component to be authorized to use a composite service.

Service Providers: Service providers are actors in a service composition system, who provide individual services for composition. Each service provider has to perform an initial registration and authentication against the service registry to participate in the service composition network.

Service Providers Service Registry




Service
Composer
Middle-
ware
Service
Composer
Middle-
ware
on
Hash-Chain
Database
Authentication
Service
Hash-Chain
Generator
Registration
Service



Authentication Registration

IoT Device 1	Col2

Authentication Service Registration Service	Col2
Authentication Service Registration Service	

Trusted Third Party Figure 4: Service Composition Architecture

6. Conclusion

In this work, we presented a mechanism to include IoT devices into a service composition system anonymously. We have listed some of the most common schemes used for anonymous authentication and have identified a suitable scheme to be used for the authentication of IoT devices. The hash-chain authentication scheme represents an approach that is suitable for implementation on IoT devices. Furthermore, the hash-chain issuing authority can determine the time until a new registration of a device has to be performed, which enables a flexible reaction on different security levels offered by the performed authentication in the course of the registration.

References

[1] [A. Rasheed, R. R. Hashemi, A. Bagabas, J. Young, C. Badri and K. Patel, “Configurable Anonymous Authentication Schemes] For The Internet of Things (IoT),” 2019 IEEE International Conference on RFID (RFID), Phoenix, AZ, USA, 2019, pp. 1-8, doi: 10.1109/RFID.2019.8719256

[2] [S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: The road ahead,”] Computer Networks, vol. 76, pp. 146–164, 2015

[3] [T. Borgohain, U. Kumar, and S. Sanyal, “Survey of security and privacy issues of internet of things,” arXiv preprint] arXiv:1501.02211, 2015

[4] [H. Eun, H. Lee, and H. Oh, “Conditional privacy preserving security protocol for nfc applications,” Consumer Electronics, IEEE] Transactions on, vol. 59, pp. 153–160, February 2013

[5] [A.-R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and privacy challenges in industrial internet of things,” in] Proceedings of the 52nd Annual Design Automation Conference, p. 54, ACM, 2015

[6] [Fongen, “Identity management and integrity protection in the internet of things,” in 2012 Third International Conference on] Emerging Security Technologies, pp. 111–114, IEEE, 2012.

[7] [Au M., Susilo W., Mu Y.: Constant-Size Dynamic k-TAA. In: Security and Cryptography for Networks, SCN’06. LNCS, vol. 4116,] pp. 111–125. Springer (2006)

[8] [Boneh D., Boyen X.: Short Signatures Without Random Oracles. In: Advances in Cryptology, Enrocrypt’04. LNCS, vol. 3027,] pp. 56-73. Springer (2004)

[9] [Boneh D., Boyen X., Shacham H.: Short Group Signatures. In: Advances in Cryptology, Crypto’04. LNCS, vol. 3152, pp. 41–55.] Springer (2004)

[10] [Camenisch J., Herreweghen E.: Design and Implementation of the Idemix Anonymous Credential System. In: ACM] Conference on Computer and Communication Security, CCS’02. ACM (2002)

[11] [Camenisch J., Kohlweiss M., Soriente C.: An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous] Credentials. In: Public Key Cryptography, PKC’09. LNCS, vol. 5443, pp. 481–500. Springer (2009).

[12] [Camenisch J., Lysyanskaya A.: An Efficient Sysem for Non-Transferable Anonymous Credentials with Optional Anonymity] Revocation. In: Advances in Cryptology, Eurocrypt’01. pp. 93–118. Springer (2001)

[13] [Camenisch J., Lysyanskaya A.: An Efficient Sysem for Non-Transferable Anonymous Credentials with Optional Anonymity] Revocation. In: Advances in Cryptology, Eurocrypt’01. pp. 93–118. Springer (2001)

[14] [Camenisch J., Lysyanskaya A.: A Signature Scheme with Efficient Protocols. In: Security and Cryptography for Networks,] SCN’02. LNCS, vol. 2576, pp. 268–289. Springer (2002)

[15] [Wang, Guilin. (2004). On the Security of a Group Signature Scheme with Forward Security. 2971. 27-39. 10.1007/978-3-540-] 24691-6_3

[16] [Wiener F.: Threshold Signatures: Security for the Libra Digital Asset Era. In: Whitepaper] (2019)

[17] [L. Lamport, “Password Authentication with Insecure Communication”, Communications of the ACM 24.11 (November 1981),] pp 770-772

[18] [K. Bicakci, N. Baykal, Infinite Length Hash Chains and Their Applications. In: Proceedings. Eleventh IEEE International] Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Pittsburgh, PA, USA, pp. 57–61, (2002)