International Journal of Information Security (2022) 21:103–114
https://doi.org/10.1007/s10207-021-00543-w

REGULAR CONTRIBUTION

Public key versus symmetric key cryptography in client–server authentication protocols

An Braeken[1]

Published online: 8 March 2021

Abstract
Every month, several new protocols are popping up, comparing themselves with a few others and claiming to outperform the whole state of the art. The most popular domain of protocols is the one for authentication in a client–server architecture for which both symmetric key- and public key-based protocols are being proposed. The usage of public key-based mechanisms has several consequences, not only with respect to an increased computational and communication cost, but also with respect to increased possibilities to strengthen the protocol by making it resistant against a semi-trusted third party. On the other hand, we also recall that symmetric key-based protocols can already offer a nice set of security features. We see a trend in the current generation of papers published on public key-based client–server authentication protocols, showing that only a very limited amount of them really exploit the power that public key cryptography can offer with respect to this privacy towards a semi-trusted third party, and most of them do not even satisfy the same security features able to be also realised by a much more efficient symmetric key-based protocol. This paper serves as a warm wake-up call to all protocol designers to rethink the usage of more heavyweight constructions compared to symmetric key-based mechanisms in order to ensure that if they are used, they also fully exploit their inherent strength.

Keywords Elliptic curve cryptography · Symmetric key cryptography · Client–server authentication · Protocol design

1 Introduction

Client–server-based communication is one of the most common and basic architectures in network communication. The client can take the role of a powerful personal computer, but can also be more constrained like being a smartphone, a simple sensor or even a tag. The server should respond on the requests sent by the multiple clients and is considered to be more powerful. It is mostly represented by a cloud server, a gateway, a reader, etc. In all these settings, security should be established between client and reader. Depending on the type of application, the required security needs (like anonymity, unlinkability, perfect forward secrecy, etc.) vary and also the usable capabilities of the client determine to a large amount of the potential possibilities for security protection.

Hundreds of authentication and key establishment schemes have been proposed in the literature for such client–server-based communication settings. The most important difference between these protocols is the usage of the type of underlying cryptography, being either symmetric or public key-based cryptography. Public key-based cryptography requires much higher computation and communication costs and it is also known to offer higher security. This higher security basically refers to the possibility to obtain full user privacy as the owner (client in this context) is able to generate a private key only known by itself.

However, a large amount of papers on public key-based protocols are proposed in the literature, where no proper usage has been made of this specific security advantage that public key cryptography can offer, resulting in schemes which barely behave better with respect to presence of some important security features than the ones using solely symmetric key-based cryptography, ending up with a very expensive solution without added value. This is partly a consequence of the way the schemes have been invented, mostly as correction on a correction of another correction, resulting in a long generation of stepwise corrected protocols, which are in the end so complicated that the original ideas are not clear anymore.

✉ An Braeken
an.braeken@vub.be

[1] Vrije Universiteit Brussel, Pleinlaan 2, 1070 Brussel, Belgium

This paper has the goal to rethink the usage of public key-based mechanisms in protocols. We particularly focus on the most basic client–server architecture, but the reasoning is similar for other types of architectures. As a consequence, we start summarising the main shortcomings of using only symmetric key instead of public key-based mechanisms. Next, we do an analysis on ten recently (publication years: 2019–2020) published client–server authentication protocols in well-established journals and find out that only one is able to exploit the unique advantages public key cryptography can offer. Unfortunately, the scheme is still not capable to offer a broader set of security features, as already available in recently proposed symmetric key-based protocols.

The paper is organised as follows. In Sect. 2, some preliminaries on architecture, attack model, security features, symmetric and public key cryptography are given. Section 3 explains in depth the main differences between symmetric key- and public key-based mechanisms. In Sect. 4, we discuss the consequences of these differences on the general design principles of a protocol for a variety of scenarios. In Sect. 5, an analysis is provided on ten recently defined public key-based protocols in the literature, where different vulnerabilities of each scheme are discussed and where we can indeed see that public key-based mechanisms are not used in a proper way. We end the paper with some conclusions in Sect. 6.

2 Preliminaries

2.1 Architecture

The most basic set-up for a client–server mutual authentication protocol is the one existing of multiple clients trying to login at one particular predefined server with the aid of a trusted third party (TTP) for providing the registration and initialisation of the protocol, without active involvement during the actual key agreement process.

There exist different variants of this architecture. In order to make the classification later in Sect. 5, we distinguish three main categories of difference.

A1 In some cases, the server takes also the role of the TTP and thus there is no external TTP. This is against all fundamental rules of cryptography, where subscription and operation should be strictly distinguished. These schemes are analysed in Sect. 5 with the assumption that a TTP is involved besides the server, sharing only the symmetric key-related data with each entity and being not aware of the private keys of each entity.

A2 There is also the variant where the client can authenticate to multiple servers. In this case, we only limit our discussion to the schemes allowing this process, without active involvement of the TTP during the key agreement. Some examples of schemes with active involvement can be found in for instance [1–3] and have many disadvantages as they strongly rely on the availability of the TTP.

A3 Finally, the schemes also differ in the capabilities of offering multi-factor authentication by the client. In case of one-factor authentication, the client just represents a device like a sensor or a tag. For two-factor authentication, the client is often a user provided with a mobile phone or smartcard able to accept user input like for instance password. In three-factor authentication schemes, the client possesses a device where both password and biometric information of the user should be entered. This type of schemes always requires user interaction and only store a part of the secret key material to activate the protocol.

2.2 Attack model

We here consider four types of attackers.

I Yao Dolev Attacker [4]. This is a typical attack model, considering the presence of an active and passive attacker, who is able to eavesdrop on the channel and to change, delete, insert, replay messages or part of the messages.

II Leakage of session-specific temporary data information. In this attack, session-specific temporary data, like for instance random variables, timestamps and previous session keys, are leaked.

III Leakage of long-term private key material. This type of attack allows the retrieval of long-term key material, being the private key or symmetric key of the entity to derive previous session keys.

IV Inside attacker. An inside attacker for this architecture considers a malicious client trying to abuse its information to retrieve information on other users. It can also represent a malicious server in case of multi-server authentication schemes (cf. A2).

V Semi-trusted TTP. Such type of TTP is also called honest and curious TTP, who will perform all the required security-related operations in a correct way, but is interested to reveal the data for its own purposes like for instance for selling the retrieved data to other third parties.

2.3 Security features and attacks

When developing a security scheme, the following security features are the most important.

F1 Mutual authentication and confidentiality between client and server should be established in order to prevent impersonation and man-in-the middle attacks. Both enti-

ties should contribute in the construction of the shared session key to avoid known key attacks. Typically, this feature is defined for an attacker of types I and IV.

F2 Integrity of the messages sent between client and server is required to ensure that the actual content is not changed. In particular, it avoids replay attacks and is considered for type I and IV.

F3 Anonymity guarantees that no other outsider is able to derive the real identities of client and server. This feature should hold for an attacker of types I and IV.

F4 Unlinkability is a stronger version of anonymity, where the attacker is not able to link two requests coming from the same client. Again, protection against the attack types I and IV should be considered.

F4* Strong unlinkability offers unlinkability even if the identity or public key of the device is leaked at a certain moment in time, taking into account an attacker of types I and IV.

F5 Protection against session-specific temporary data attack requires to still obtain mutual authentication, integrity, anonymity and unlinkability in case of an attacker of type II,

F6 Perfect forward security offers protection of the previous derived session keys and involved identities, thus security features F1–F4, in case of an attacker of type III. We distinguish between F6-C and F6-S, referring to either client or server, whose key material is leaked.

F7 Protection against TTP involvement avoids the TTP to break the security features F1–F4, in case of an attacker of type V.

F8 Protection against impersonation of TTP considers an attacker of type V, which should not be able to send requests or responses in the name of one of the other entities participating in the protocol, without being noticed by the server.

There are two additional features, which have no direct security impact, but are very important with respect to the performance of the scheme.

P1 Scalability. The reader should be able to directly look-up the corresponding identity of the client based on information in the request, instead of going exhaustively through the whole list of registered clients and to verify for each of them if it satisfies a certain equality provided in the request.

P2 No individual storage. The reader does not need to store security material for each individual client, but only a global secret master key and a list of revoked clients.

2.4 Notations

In symmetric key-based protocols, both client and server possess secret key material, which can be used to construct a common shared key relying on solely symmetric key-based operations. Typical operations in symmetric key-based protocols are the block ciphers and stream ciphers, which are symmetric en/decryption schemes $C = E_K(M), M = D_K(C)$, able to encrypt a message M to a ciphertext C or to decrypt C to M using a common shared key K. The most simple example, often used in protocols, is the Vernam scheme or one-time pad $C = M \oplus K$ offering perfect security [5]. Another important primitive, often used in protocols and having typically smaller or similar computational costs compared to encryption operations, is the hash function $H(M)$, applied on a message M of arbitrary length to result in an output message of fixed length l. In order to be resistant against collision attacks, pre-image and second pre-image attacks, this length l should be at least 256, due to the birthday attack.

Public key cryptography or also called asymmetric key cryptography has been invented by Diffie and Hellman [6]. Here, both entities possess a key pair, consisting of a private and public key. The security of public key cryptography relies on the hardness of a mathematical problem like for instance the factorisation of large prime numbers (RSA) or the solution of discrete logarithms (DL). Elliptic curve cryptography (ECC) currently offers the most lightweight public key-based cryptographic solution [7,8], both with respect to computation and communication costs. To give an idea, for a 128-bit security, field sizes of at least 3072 bits are required in RSA, while a field size of only 256 bits is needed with ECC. ECC is based on the algebraic structure of elliptic curves (ECs) over finite fields F_p. A generator G of order q is chosen for each curve. There are two main operations defined in EC, addition $P_1 + P_2$ of two points and multiplication rP with $r \in F_q$. EC multiplication is built up of additions and doubling operations and requires the highest computation cost, and thus the number of EC multiplications should be limited as much as possible for constrained clients.

The security of ECC relies on two well-known computational hard problems: the Elliptic Curve Discrete Logarithm Problem (ECDLP) and the Elliptic Curve Diffie Hellman Problem (ECDHP). Due to the ECDLP, it is computationally very hard to find x given xG and due to the ECDHP, it is computationally very hard to find xyG given xG, yG, G.

3 Differences between symmetric and public key cryptography

We now describe the main differences relevant for the discussion in our analysis.

3.1 Security-related differences

As mentioned in the previous section, both symmetric key and public key cryptography rely on two completely different underlying principles and thus result in different consequences with respect to security.

The first main difference is the key sharing of material by the TTP. In symmetric key-based protocols, the TTP decides on the key material and shares the material among the clients and the servers. During this process, the TTP should possess an out-of-bound channel (like for instance pre-installation via physical presence) to realise it in a secure way. On the other hand, for public key-based protocols, the entity can decide itself on a key pair and require the TTP to make a certificate on the key pair. The only difficulty in here is to prove that the identity requesting for a certificate corresponds with the real identity claiming it, which can be done by external channels like for instance usage of smartphone, reader with identity card, proving of knowledge publicly available on the blockchain, etc. In practice, often the first asked first served principle is applied for devices, such that devices are linked to the first identity requesting a certificate. In constrained devices, the Elliptic Curve Qu Vanstone (ECQV) [9] mechanism offers a lightweight solution to deal with certificates as based on the identity and the certificate (being a point on the EC), the public key can be derived by anybody. A lot of lightweight protocols using ECQV have been proposed in the literature [10,11].

The structure of the key material has two main consequences for the security strength against a type V attacker, also referred as security features F7 and F8 in Sect. 2.

- F7: First, since the TTP is aware of all the key material in a symmetric key-based protocol, it is able to follow the communications between the clients and the server and to retrieve the identities of the involved entities and the derived session keys in order to obtain the secret data sent between both.
- F8: Second, the TTP can impersonate an entity and send requests in the name of that entity, who is not able to deny the request afterwards.

Consequently, the usage of symmetric key cryptography in a secure way always requires a complete trust of the TTP, since F7 and F8 can never be satisfied due to the inherent structure of symmetric key-based cryptography.

In order to completely satisfy the security features F7 and F8, both client and server should possess a private key, which is only known by the entity itself, e.g. construction by means of for instance the ECQV mechanism. In addition, the usage of the private key should be involved in the request and response message of each entity, such that impersonation becomes impossible by the TTP and such that the used key material can be hidden from the TTP.

3.2 Performance-related differences

In [12], the performance of EC multiplication, EC point addition, symmetric key encryption AES_128_CCM_8 symmetric encryption/decryption and SHA256 hash function have been determined on different platforms. The most constrained platform is represented by the Zolertia RE-mote ARM Cortex-M3 running at 32 MHz, with 32 KB RAM and using the Hardware Acceleration Engine to optimise the EC and AES operations. The so-called fog device is represented by a Raspberry PI 3B Quad Core running at 1.2 GHz with 1 GB RAM and by using the BouncyCastle library for the implementation of the cryptographic operations. Finally, for the server, a personal computer Intel Core i7-8750H CPU running at 2.2 GHz with 16 GB RAM is considered, using also the BouncyCastle library. Table 1 summarises the measurements of [12] (cf. Table 2 in [12]).

The difference in efficiency of an EC multiplication versus a symmetric key encryption or hash function is significant. In particular, for the constrained Zolertia RE-mote, an EC multiplication takes more than 2298 times and 5068 times compared to a symmetric key encryption and hash function, respectively. For more powerful devices, these differences decrease, but are still remarkable. For the Raspberry PI, the differences are 632 times and 2530 times, while for the personal computer, these differences are approximately 383 times and 1148 times.

4 Consequences of differences for protocol design

We have summarised the main advantages of public key-based cryptography above symmetric key-based cryptography in Sect. 3 for client–server authentication schemes. Besides the initialisation, which is mostly a one-time process, the main strength of public key cryptography in these authentication protocols is thus the possibility to keep the security strength in the own hands of the client, meaning the avoidance of a TTP that can play the role of big brother in the system.

As a general guideline, when developing or selecting a proper security scheme, one should first think of the application in which the scheme needs to be applied and the corresponding security guarantees which should be offered to the client. A lot depends on the type of client, mainly being a device or a user. We now describe these two scenarios and the reasoning why to choose for symmetric versus public key cryptography.

Table 1 The average time of EC point multiplication, EC point addition, AES_128_CCM_8 symmetric encryption/decryption and SHA256 hash function for the Zolertia RE-mote, Raspberry PI 3B and personal computer

Device	EC_mult (ms)	EC_add (ms)	AES (ms)	SHA256 (ms)
Zolertia RE-mote	344.659	5.080	0.150	0.068
Raspberry PI 3B	37.943	0.182	0.060	0.015
Personal computer	1.148	0.005	0.003	0.001

4.1 Type of client

4.1.1 User as client

In times where privacy of the user takes a more and more important role, which can be demonstrated by the development of the General Data Privacy Regulation (GDPR) [13] in 2018 by the European Commission, it should be compulsory to include the necessary measurements to allow the user strict control and anonymity on the usage of applications. In particular, when critical data is involved like in the domains of healthcare, insurance, etc., or when the application performs critical actions like for instance payments, there should be no doubt. This protection is translated in security features F7 and F8 and as said before, can only be realised by means of public key-based mechanisms.

Note that this scenario is obtained via architecture A3 (multi-factor). Consequently, any scheme in the literature with architecture A3 should not combine it with architecture A1 (server=TTP) and should provide security features F7 and F8.

We will see in the next section (e.g. Table 3) that in most of the recently published schemes in the literature with architecture A3 both F7 and F8 are not supported, and that also often it is combined with architecture A1. Moreover, we can also see that many schemes just barely offer the same security features, which can be obtained using symmetric key-based protocols.

4.1.2 Device as client

The client can also represent a device. Here, we make distinction between two main situations. First, the device can be one used in daily non-critical and non-privacy intrusive domains, like for instance sensors applied in farming, supply chain, climate, environmental monitoring, etc., and often consists of constrained resources due to being battery powered or having limited available bandwidth. In this case, protection in general against a semi-trusted TTP is not a major concern. Instead, it is of paramount importance not to use the more computational and communication demanding public key-based mechanisms. All the other essential security features (F1–F6) can be also perfectly obtained using symmetric key-based protocols.

Second, the device can also be used in applications related to smart grid and vehicular networks, where protection against an overall controlling unit, being TTP, should be included since they are inherently linked to the behaviour and the privacy of the related user. Consequently, protection against F7–F8 is required and thus public key-based mechanisms are needed.

To conclude, the need for protection against a semi-trusted TTP in this scenario should be well thought and mainly depends of the type of application as it has important consequences regarding computation and communication efficiency. Note that this scenario corresponds to the architecture not belonging to A3.

4.2 Revocation

Revocation typically belongs to a public key infrastructure (PKI) in which the certificate includes the lifetime and eventually other attributes. When using the constrained ECQV certificates, the expiration date and or other attributes can be easily included. However, it also requires to send this extra information into the message, resulting in additional communication overhead. In addition, at the server side, access to a certification revocation list or a service managing the revocation is needed, demanding additional communication and computation overhead.

As a consequence, taking the fact that revocation brings substantial overhead to the whole process, it is important to clearly motivate the need for it. There are different situations in which revocation can be useless. First, devices with limited material and financial implications when they got lost, do not have any advantage of including such service. Second, if there are no means installed to report a revocation, like for instance a user or management application, it does not make sense to include revocation mechanisms. Third, also renewal processes can be omitted if the lifetime of the device is limited and no changes in identity or ownership are expected.

If we refer to the two scenarios of client, it is clear that both the user as client and security critical device as client both have advantage of including a certificate process, while a non-critical device as client can perfectly operate without.

Table 2 Comparison of lengths of private and public keys for different postquantum algorithms

Algorithm	Type	Public key size (B)	Private key size (B)
NTRU encrypt	Lattice	6130	6743
Rainbow	Multivariate	124,000	95,000
New hope	Ring LWE	2000	2000
Goppa-based McEliece	Code based	1,000,000	11,500
Random linear code-based encryption	Code based	115,000	3000
Quasi-cyclic MDPC-based McEliece	Code based	1232	2464
SIDH	Isogeny	751	48
SIDH (Compressed keys)	Isogeny	564	48

4.3 Postquantum security

It is also important to be mentioned that in case the quantum computer [14] becomes reality, the most efficient public key-based mechanism ECC will be completely broken [15]. The current public key-based alternatives resistant in the postquantum era, like lattice-based cryptography, multivariate-based cryptography, hash-based cryptography, code-based cryptography and supersingular EC isogeny cryptography, require much higher computation and communication cost. On the other hand, the security of symmetric key-based cryptography will still be valid, taken into account that the security level should be doubled. To give an idea about the difference in complexity, Table 2 from [16] provides the length of the private and public keys of the most well-known postquantum public key-based algorithms offering 128-bit security. These numbers give a first impression as often public keys need to be sent during the protocol. As can be seen, the best one (SIDH compressed) still has a size which is more than 17 times larger than AES-256.

Consequently, in the future, the trade-off between complete user privacy and efficiency will become even larger and it is very likely that sacrifices will be made on the first one if no other postquantum efficient alternatives will be found in short term.

5 Analysis on recently published public key-based protocols

5.1 Discussion on published public key-based protocols in the literature

Based on a search in google scholar on the concepts, "public key/ECC-based client–server authentication protocols", "public key/ECC-based multi-server authentication protocols", starting from the year 2019, we selected ten most relevant security schemes and investigated their behaviour with respect to the previously defined security features F1–F8 and performance features P1–P2. For completeness in the classification, we also included the type of architecture A1-A3. We now shortly discuss each of the schemes.

- Sowanjanya et al. [17] propose a client–server authentication scheme as a reaction on the security flaws found in [18]. No perfect forward secrecy at both tag and server side with respect to retrieval of identity is provided. If also session state information is retrieved, the session keys are derived. On the other hand, the scheme utilises the public key mechanisms in a proper way, such that resistance is offered by a semi-trusted TTP (cf. F7,F8).
  The main problem in this scheme is that it does not offer scalability since the server needs to exhaustively compute the key $K'_{UA} = S_{AS} P K_U$, with S_{AS} the private key of the server and PK_U the different stored public keys of the users.
  Their scheme is said to outperform the schemes of [18–20] with respect to security strength and computational costs. The communication and storage costs are slightly worse than [20], but better than the others.
- Dinarvand et al. propose in [21] a client–server protocol in the context of RFID, where the server takes the role of TTP (cf. A1). As mentioned before, we analyse the scheme as having an additional TTP, responsible for the division of the key material. This key material consists of shared data between client and reader. Only the reader possesses a private key, whose public key is also prestored at client side. In such scenario, the scheme does not offer protection against impersonation of the TTP since the random value R_2 can be easily replaced by a new value. On the other hand, a semi-trusted TTP is not able to derive the identity of the client or to retrieve the session key without knowledge of the private key of the server. Perfect forward secrecy, both at client and server side, is not provided.
  The scheme [21] is claimed to outperform [19,22–24] with respect to security strength and computational costs, but is slightly worse for communication costs.

– Merabet et al. [25] have proposed two client–server schemes. The first scheme is based on an improvement of [24] and gives the server the role of TTP. When again assuming the existence of an external TTP, the scheme becomes vulnerable for a semi-trusted TTP, both for impersonation and identity retrieval. Impersonation follows from the fact that the client does not possess its own private key and identity retrieval is due to the definition of the parameter $Auth_s$, which only requires to verify the different stored identifiers of the tag ID_{Ti} in its database. As a consequence, the system also does not offer strong unlinkability. Furthermore, perfect forward secrecy at both server and client side, and protection against session-specific information attack are not satisfied.
  The second scheme is another client–server authentication scheme, also called the hash-based M2C authentication protocol. Similar as the previous scheme, it does not satisfy F7 and F8 with respect to the semi-trusted TTP. It does not satisfy unlinkability as the first submitted random value R_S is not authenticated and based on the response identity-related information is retrieved. This scheme also does not offer perfect forward secrecy at both server and client side, and protection against session-specific information attacks.
  Both schemes have been compared with [19,21–24] and outperform with respect to computation, communication and storage costs.
– In [26], a two-factor client–server authentication scheme is proposed, where both client and server possess a private key. However, these keys are not exploited in the correct way since the scheme does not offer protection against a semi-trusted TTP, both for identity retrieval and impersonation. Moreover, the scheme does not offer unlinkability as in each request the client sends the static parameter El_i. Another major issue in the scheme is the lack of scalability, which requires to look up in its database the different values of PV_i in order to retrieve the corresponding I_i. The system does not offer security for session-specific information attack.
  The scheme of [26] claims to outperform the schemes of [22,27–32] with respect to security strength. Regarding computational costs, it is a bit higher than the schemes of [22,28,31,32] and for communication costs, only the schemes [22,27] behave better.
– Ying et al. describe in [33] a two-factor multi-server authentication protocol, to be applied in the Fifth Generation (5G) networks. Unfortunately, it does not satisfy both security features F7 and F8 in case of a semi-trusted TTP due to the construction of the key material by means of self-certificates generated by the TTP in which the TTP determines the private keys of the entities.
  Since $A^*_{U_i}$ is static in each request, no unlinkability is obtained in the scheme. In fact, none of the servers got to know the real identity of the user. Furthermore, no security is offered for a session-specific information attack and also perfect forward secrecy at both client and server side is not obtained. The scheme is secure for malicious insider servers due to the usage of self certificates.
  The scheme of [33] has been compared with [20,22,34–36] and claims to outperform all of them with respect to security and traffic load. Regarding computational costs, it behaves better compared to only schemes [20,34].
– Shafiq et al. present in [37] a two-factor client server authentication protocol without external TTP. Considering an additional TTP, it does not offer protection against a semi-trusted TTP with respect to security features F7 and F8 since the user does not possess a private key and given the parameters PID_r, τ submitted over the public channel, the semi-trusted TTP is able to retrieve the identity of the sender and derive the session key. The scheme is also vulnerable for session-specific information attack and does not offer protection to perfect forward secrecy at both client and server side.
  Comparisons have been made with [38–40] and it has been shown that the proposed scheme of [37] outperforms all of them with respect to security strength, computation, communication and storage costs.
– Kumari et al. [41] present the ESEAP scheme, which is another two-factor client–server authentication scheme without external TTP. When considering an external TTP in the scheme, again features F7 and F8 with respect to a semi-trusted TTP are not satisfied as all identity- and security-related information in the first step is encrypted with a session key constructed by means of secret key material known by the TTP. The scheme does not satisfy unlinkability as the parameter P_3 is static and sent in each request.
  An extensive comparison has been made between the proposed scheme of [41] and related work [36,42–58] regarding 20 different security features and attacks. The scheme [41] claims to be the only one to possess these 20 security features. Moreover, it also behaves best with respect to communication costs. When comparing computational costs, the schemes of [42,43,51,54,57,58] still outperform [41].
– Wan et al. [59] propose a smartcard-based authentication scheme for a multi-server architecture, based on the cryptanalysis of the scheme proposed by Wei et al. [60]. The client needs to store the public key of each server individually and all servers possess the same shared key. It is not secure against a malicious server who previously received already the identifier information of the client via an earlier registration and exhaustively verifies the correctness of M_1 (cf. Attacker type IV). For the same reason, it is not secure against a curious TTP and also

impersonation is possible as the client does not possess its own private key. There is also no protection against perfect forward secrecy of the server and session-specific information attack.
Both performance and security strength of the scheme [59] have been compared with the schemes [60–63] and it turned out that [59] outperforms for all of them.

– Naeem et al. [64] propose a client–server authentication protocol in the use case of RFID and IoT as an improvement of the scheme of Alamr et al. [23], which has been shown to possess several weaknesses. We also consider here an external TTP for the analysis. The main problem in the scheme is that there is no mutual authentication in the first response of the tag. As a consequence, any attacker can send a random value to the tag, who will respond with a message revealing its identity. Consequently, the scheme fails with respect to anonymity and unlinkability. Furthermore, it is vulnerable for the session-specific information attack and impersonation of the TTP.
The scheme has been compared with [22,23,65] and claims to outperform all of them with respect to security strength, computation and communication costs.

In our analysis, also two recently proposed symmetric key-based schemes of [66] have been included to serve as reference. It is often forgotten in the literature that even symmetric key-based protocols are able to offer a significant amount of security features.

For instance, the papers [66–70] include both anonymity and unlinkability features into the protocols. In particular, in [66], two new symmetric key-based protocols are proposed for this type of client–server architecture with a focus on efficiency at the side of the client, such that even a very small device is able to complete the protocol. Both protocols rely on a dynamic update of the identity- and key-related information of the client after each protocol run to enable unlinkability and anonymity. In the first protocol, the server stores a copy of the dynamic identity and key of each client, while in the second protocol only the storage of the revoked identities is required. It has been proven in [66] that both protocols outperform related work [67–70] on symmetric key-based client–server protocols both with respect to security and performance. In particular, the security features (F1–F6) are satisfied for any type of attacker in scheme 2 of [66], except type III with respect to the server and type V with respect to a semi-trusted TTP. Since in scheme 1 of [66], the previous values of the key material are stored, a full protection against perfect forward secrecy can never be obtained. However, the attacker can only go back until the last used key. The perfect forward secrecy with respect to the server has only been recently solved in [71] for symmetric key-based protocols. This solution focuses on the perfect forward secrecy feature, did not satisfy anonymity and unlinkability and consists of five phases with multiple hash functions included. Theoretically, it should be possible to include this technique in the second protocol of [66], however, resulting in additional computation and communication costs. The vulnerability to the TTP (cf. F7 and F8) is due to the inherent symmetric key properties, as mentioned before.

5.2 Comparison of published public key-based protocols in the literature

Table 3 summarises the characteristics of the different schemes discussed above with respect to architectures A1–A3, security features F1–F8, and performance aspects P1–P2. The following conclusions can now be drawn out of this table.

5.2.1 Regarding F7, F8

In order to make the difference between the more efficient symmetric key-based protocols (cf. [66]), these additional security features with respect to a semi-trusted TTP should be included in the public key-based protocols. However, from Table 3, it can be concluded that only a limited amount of schemes are able to offer these features, which public key-based cryptography can inherently offer (cf. F7, F8). It is remarkable that a lot of schemes [21,25,26,37,41,64] are still proposed in Architecture type 1 (A1), where the server takes the role of TTP and thus in which F7 and F8 can be naturally not satisfied. However, even if we consider in our analysis different roles for both, only the schemes of Dinarvand et al. [21] and Nayeem et al. [64] satisfy protection against a TTP following the communications (cf. F7), but still suffer from impersonation (cf. F8) as the client does not possess its own private key in their schemes. All the other schemes do not offer both F7 and F8. Even if the client possesses its own private key, it is not a guarantee that it is used in a correct way (cf. [26]) to enable F7 and F8. From the complete analysis, we found that only the scheme of Sowanja et al. [17] satisfies both features.

5.2.2 Regarding F6-S

The feature perfect forward secrecy at server side, which is possible to be obtained by symmetric key cryptography but with huge additional communication costs (see [71]), is only rarely addressed by the public key-based authentication protocols and thus seems to be not seen as top priority in the proposal of new schemes. It is mostly assumed that the TTP possesses sufficient resources to protect its secret key material. However, in a strong cryptographic scheme, it is better to also consider this feature as potential failures can always appear. Only the schemes [26,41,64] satisfy this feature and at the same time perfect forward secrecy at the client side (cf.

Table 3 Comparison of security strength F1–F8 and performance aspects P1–P2 for the previously discussed security schemes with architectures A1–A3; [1]: Only leakage of last session key is possible, [2]: It is possible with additional cost

Scheme	A1	A2	A3	F1	F2	F3	F4	F4*	F5	F6-S	F6-C	F7	F8	P1	P2
Braeken [66]-1	0	0	0	x	x	x	x	x	x	x[1]	x	0	0	x	0
Braeken [66]-2	0	0	0	x	x	x	x	x	x	0[2]	x	0	0	x	x
Sowanja [17]	0	0	0	x	x	x	x	x	0	0	x	x	x	0	0
Dinarvand [21]	x	0	0	x	x	x	x	x	x	0	0	x	0	x	0
Merabet [25]-1	x	0	0	x	x	x	x	0	0	0	0	0	0	x	0
Merabet [25]-2	x	0	0	x	x	x	x	0	0	0	0	0	0	x	0
Panda [26]	x	0	x	x	x	x	0	0	0	x	x	0	0	0	0
Ying [33]	0	x	x	x	x	x	0	0	0	0	0	0	0	x	x
Shafiq [37]	x	0	x	x	x	x	x	x	0	0	x	0	0	x	0
Kumari [41]	x	0	x	x	x	x	0	0	x	x	x	0	0	x	0
Wan [59]	0	x	x	x	x	x	x	x	0	0	x	0	0	x	x
Nayeem [64]	x	0	0	0	0	0	0	0	0	x	x	x	0	x	0

F6-C). Note that in the first symmetric key-based scheme of [66], this feature is partly addressed as still the last session key can be revealed since the server stores both last and second last key material in the database in order to overcome potential desynchronisation problems.

5.2.3 Regarding F6-C

There are clearly more schemes addressing F6-C than F6-S, which is logic as it is more more difficult to be organised in a cryptographic correct way. In most of the multi-factor authentication schemes [26,33,37,41,59], perfect forward secrecy of the client is satisfied since it corresponds with the protection against a stolen smartcard. However, there is still a two-factor authentication scheme [33] for which it does not hold. Most of the other single-factor authentication schemes (e.g. [21] and the two schemes of [25]) do not satisfy this feature too. The only non-multi-factor public key-based authentication schemes satisfying F6-C are the schemes of [17,64].

5.2.4 Regarding F5

Offering protection against session-specific temporary data does not seem to be a very important feature to be included in the latest generation of protocols proposed in the literature. However, it is a required feature to offer security in more advanced security models like the Canetti–Krawczyk (CK) adversary model [72], in which the attacker has access to either temporary session information or long-term private key material at both client and server at the same time, except the combination of long-term key material at both client–server is not allowed. In fact, it corresponds with the attacker having capabilities of types II and III combined. Especially in the domains of smart grid security, several schemes [73–76] have been proposed in the literature satisfying CK security. Besides the symmetric key-based schemes of [66], only two public key-based schemes [21,41] from our analysis took this feature into account and offered protection for it. None of the public key-based authentication schemes can offer resistance in the CK security model as none of them satisfy at the same time features F5 and F6, except the scheme of Kumari [41].

5.2.5 Regarding F4 and F4*

Still, a significant amount of schemes are not able to offer unlinkability [26,33,41,64] or strong unlinkability (schemes of [25]) since they are using a mechanism of static pseudonyms to establish anonymity. In fact, all of the multi-factor authentication schemes, except the scheme of Shafiq [37] were not able to offer the unlinkability feature. This is strange as the combination of multi-factor authentication and unlinkability does not cause particular problems. It is important to mention that the scheme of Kumari [41], which was the only public key-based scheme able to satisfy both F5 and F6, cannot offer protection against tracing attacks.

5.2.6 Regarding F1, F2, F3

Offering protection against F1, F2, F3 is the minimum goal to be obtained in an authentication scheme and all currently proposed schemes in the literature aim to reach these features. However, we still found one scheme [64], which suffered from an attack due to a lack of mutual authentication in the first phase and thus is not able to offer F1, F2, F3. Although they claimed to have proven the security of the scheme with ProVerif and Burrows–Abadi–Needham (BAN) logic, they were not able to detect the vulnerability present in the first response by the tag.

5.2.7 Regarding P1

Most of the schemes consider to address scalability. However, we still found two schemes [17,26] that require a highly inefficient exhaustive search by the server, who has to go through the list of all stored identities in order to respond on the request of the client. In the case of [17], it was required to obtain strong unlinkability, while in [26], it only leads to anonymity.

5.2.8 Regarding P2

It is remarkable to note that almost all schemes require individual storage of the key material at the server side, which is indeed in particular important if frequent revocation is required. However, none of the schemes have referred to revocation as a regular step in their scheme. If no individual information is stored, revocation becomes more difficult, but can still be added by working with temporary approved key material or revocation lists. Only the multi-server-based schemes [33,59] do not need to store key material for all the approved clients at server side, but are also the schemes that rely on an active involvement of the TTP (cf. A2). In contrast, it has been shown that this interesting performance feature can also be realised with symmetric key-based cryptography, cf. scheme 2 of [66]. In fact, this is the main underlying difference between scheme 1 and scheme 2 of [66].

5.3 Overall conclusion on the comparison analysis

From Table 3, it can be concluded that in this list of recently published protocols, only one scheme [17] fully exploits the advantage of public key cryptography. Unfortunately, the scheme of [17] is not able to offer other interesting security features like perfect forward secrecy from server side and protection against session-specific temporary data. In particular, it also possesses nefast performance characteristics (cf. P1, P2). In all of the other investigated public key-based protocols, always one or more essential security features is leaking, compared to the two symmetric key-based protocols of [66].

We further want to note that the construction of a public key-based protocol, satisfying all of the above security features is not an open problem in the literature. In [73], a key agreement protocol in the context of smart metering has been proposed, which even satisfied protection against the CK adversary model. In [73], previously constructed schemes [74–76] claiming to possess CK security resistance have been shown to be vulnerable, resulting in a fundamental new scheme in [73] relying on the ECQV certificates.

6 Conclusion

We see in the literature that many authors are using public key-based mechanisms without real exploitation of the full functionality of it, resulting in schemes which offer no more security (with respect to other security features like anonymity, unlinkability, perfect forward secrecy, etc.) than the much more low cost symmetric key-based schemes.

Therefore, this paper starts with a clear analysis of the advantages of public key-based cryptography compared to symmetric key-based cryptography. In fact, the advantages all come down on the protection against a semi-trusted TTP or the avoidance of one big brother in the system. In order to fully exploit this advantage, both client and server should possess their own private key, only known by themselves and not shared with the TTP.

We hope this paper can be a wake-up call to all designers of authentication and key management protocol designers to first think about the application and context for which a security protocol is developed. Only for security critical use cases, requiring protection against a semi-trusted TTP and the inclusion of a revocation process, the usage of public key-based cryptography is allowed. ECC-based mechanisms offer a viable solution for this, able to work on constrained devices. However, it should be taken into account that ECC will not offer very long-lasting security as it is not quantum secure and the best other currently known alternative is not sufficiently efficient to run on a constrained device. Currently, only symmetric key-based algorithms offer a quantum secure solution, able to run efficiently on constrained devices.

To conclude, a protocol should not be designed with the idea to have a public key (mostly ECC)-based protocol, but with the idea to use it in a smart way such that all additional overhead (compared to more lightweight symmetric key-based alternatives) is fully motivated.

Funding The research was funded by the Tetra Vlaio project Velcro, Award Number HBC.2020.2073.

Declarations

Conflict of interest There are no conflicts of interest or competing interests.

Human and animal rights The research did not involve human participants and/or animals.

Informed consent No informed consent is to be reported.

References

1. Tomar, A., Dhar, J.: An ECC based secure authentication and key exchange scheme in multi-server environment. Wireless Pers. Commun. **107**, 351–372 (2019)
2. Haq, I.U., Wang, J., Zhu, Y.: Secure two-factor authentication protocol using self-certified public key cryptography for multi-server 5G networks. J. Netw. Comput. Appl. **161**, 102660 (2020)
3. Yao, H., Fu, X., Wang, C., Meng, C., Hai, B., Zhu, S.: Cryptanalysis and improvement of a remote anonymous authentication protocol for mobile multi-server environments. In: IEEE Fourth International Conference on Data Science in Cyberspace (DSC) (2019)
4. Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)
5. Shannon, C.: Communication theory of secrecy systems. Bell Syst. Tech. J. **28**(4), 656–715 (1949)
6. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
7. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **48**(177), 203–209 (1987)
8. Miller, V.: Use of elliptic curves in cryptography. Crypto Lect. Notes Comput. Sci. **85**, 417–426 (1985)
9. Certicom Research. SEC4: elliptic curve Qu-Vanstone implicit certificate scheme. In: Standards for Efficient Cryptography Group. Version 1.0. Retrieved May 15, 2020 from http://www.secg.org/sec4-1.0.pdf (2013)
10. Porambage, P., Schmitt, C., Kumar, P., Gurtov, A., Ylianttila, M.: Two-phase authentication protocol for wireless sensor networks in distributed IoT applications. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 2728–2733. Istanbul (2014)
11. Ha, D.A., Nguyen, K.T., Zao, J.K.: Efficient authentication of resource-constrained IoT devices based on ECQV 505 implicit certificates and datagram transport layer security protocol. In: Proceedings of the Seventh Symposium on Information and Communication Technology, pp. 173–179 (2016)
12. Shabisha, P., Braeken, A., Kumar, P., Steenhaut, K.: Fog-orchestrated and server-controlled anonymous group authentication and key agreement. IEEE Access **7**, 150247–150261 (2019)
13. Complete guide to GDPR, Retrieved May 15, 2020 from https://gdpr.eu
14. Bernstein, D.J.: Introduction to post-quantum cryptography. In: Post-Quantum Cryptography (2009)
15. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
16. https://en.wikipedia.org/wiki/Post-quantum_cryptography
17. Sowjanya, K., Dasgupta, M., Ray, S.: An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems. Int. J. Inf. Secur. **19**, 129–146 (2020)
18. Li, X., Peng, J., Kumari, S., Wu, F., Karuppiah, M., Choo, K.K.R.: An enhanced 1-round authentication protocol for wireless body area networks with user anonymity. Comput. Electr. Eng. **61**(C), 238–249 (2017)
19. Zhao, Z.: An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem. J. Med. Syst. **38**(2), 1–7 (2014)
20. He, D., Zeadally, S., Kumar, N., Lee, J.H.: Anonymous authentication for wireless body area networks with provable security. IEEE Syst. J. **11**(4), 2590–2601 (2017)
21. Dinarvand, N., Barati, H.: An efficient and secure RFID authentication protocol using elliptic curve cryptography. Wireless Netw. **25**, 415–428 (2019)
22. Liao, Y.P., Hsiao, C.M.: A secure ECC-based RFID authentication scheme using hybrid protocols. Adv. Intell. Syst. Appl. **2**, 1–13 (2013)
23. Alamr, A.A., Kausar, F., Kim, J.S.: Secure mutual authentication protocol for RFID based on elliptic curve cryptography. In: Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), pp. 1–7. IEEE (2016)
24. Jin, C., Xu, C., Zhang, X., Li, F.: A secure ECC-based RFID mutual authentication protocol to enhance patient medication safety. J. Med. Syst. **40**(1), 6 (2016)
25. Merabet, F., Cherif, A., Belkadi, M., Blazy, O., Conchon, E., Sauveron, D.: New efficient M2C and M2M mutual authentication protocols for IoT-based healthcare applications. In: Peer-to-Peer Networking and Applications. Springer (2019)
26. Panda, P.K., Chattopadhyay, S.: A secure mutual authentication protocol for IoT environment. J Reliab. Intell. Environ. **6**, 79–94 (2020)
27. Islam, S.K.H., Biswas, G.P.: A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. J. Syst. Softw. **84**, 1892–1898 (2011)
28. Kalra, S.: Secure authentication scheme for IoT and cloud servers. Pervasive Mob. Comput. **24**, 210–223 (2015)
29. Chang, C.C., Wu, H.L., Sun, C.Y.: Notes on secure authentication scheme for IoT and cloud servers. Pervasive Mob. Comput. **38**, 275–278 (2016)
30. Wang, F., Chen, C.M., Fang, W., Wu, T.Y.: A secure authentication scheme for Internet of Things. Pervasive Mob. Comput. **42**, 15–26 (2017)
31. Kumari, S., Karuppiah, M., Das, A.K., Kumar, N.: A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. J. Supercomput. **74**, 6428–6453 (2017)
32. Bhuvaneshwari, S., Narayanan, A.V.: Enhanced mutual authentication scheme for cloud of things. Int. J. Pure Appl. Math. **119**(15), 1571–1583 (2018)
33. Ying, B., Nayak, A.: Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography. J. Netw. Comput. Appl. **131**, 66–74 (2019)
34. Hsieh, W., Leu, J.: An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures. J. Supercomput. **70**(1), 133–148 (2014)
35. Wang, D.H.D.: Robust biometrics based authentication scheme for multi server environment. IEEE Syst. J **9**(3), 816–823 (2015)
36. Odelu, V., Das, A.K., Goswami, A.: A secure biometrics based multi server authentication protocol using smart cards. IEEE Trans. Inf. Forensics Secur. **10**(9), 1953–1966 (2015)
37. Shafiq, A., Altaf, I., Mahmood, K., Kumari, S., Chen, C.M.: An ECC based remote user authentication protocol. J. Internet Technol. **21**, 285–294 (2020)
38. Qu, J., Tan, X.L.: Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem. J. Electr. Comput. Eng. **2014**, 1–6 (2014)
39. Huang, B., Khan, M.L., Wu, L., Muhaya, F.T.B., He, D.: An efficient remote user authentication with key agreement scheme using Elliptic Curve Cryptography. Wireless Pers. Commun. **85**(1), 225–240 (2015)
40. Chaudhry, S.A., Naqvi, H., Mahood, K., Ahmad, H.F., Khan, M.K.: An improved remote user authentication scheme using Elliptic Curve Cryptography. Wireless Pers. Commun. **96**(4), 5335–5373 (2017)
41. Kumari, A., Jangirala, S., Abbasi, M.Y., Kumar, V., Alam, M.: ESEAP: ECC based secure and efficient mutual authentication protocol using smart card. J. Inf. Secur. Appl. **51**, 102443 (2020)
42. Kumari, S., Khan, K., Li, X.: An improved remote user authentication scheme with key agreement. Comput. Electr. Eng. **40**(6), 1997–2012 (2014)

43. Kumari, S., Li, X., Wu, F., Das, A.K., Odelu, V., Khan, M.K.: A user anonymous mutual authentication protocol. KSII Trans. Internet Inf. Syst. **10**(9), 4508–4528 (2016)
44. Jiang, Q., Ma, J., Li, G., Li, X.: Improvement of robust smart-card-based password authentication scheme. Int. J. Commun. Syst. **28**(2), 383–393 (2015)
45. Islam, S.K.H.: Design and analysis of an improved smartcard-based remote user password authentication scheme. Int. J. Commun. Syst. **29**, 1708–1719 (2014)
46. Karuppiah, M., Ramakrishnan, S.: A secure remote user mutual authentication scheme using smart cards. J. Inf. Secur. Appl. **19**(4–5), 282–294 (2014)
47. Maitra, T., Obaidat, M.S., Amin, R., Islam, S., Chaudhry, S.A., Giri, D.: A robust ElGamal-based password-authentication protocol using smart card for client-server communication. Int. J. Commun. Syst. **30**(11), e3242 (2016)
48. Xie, Q., Wong, D.S., Wang, G., Tan, X., Chen, K., Fang, L.: Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Trans. Inf. Forensics Secur. **12**, 1382–1392 (2017)
49. Wang, C., Wang, D., Xu, G., Guo, Y.: A lightweight password-based authentication protocol using smart card. Int. J. Commun. Syst. **30**, e3336 (2017)
50. Jangirala, S., Das, A.K., Kumar, N., Rodrigues, J.: Cloud centric authentication for wearable healthcare monitoring system. IEEE Trans. Dependable Secure Comput. **17**, 942–956 (2018)
51. Wang, D., Wang, P.: Two birds with one stone: two-factor authentication with security beyond conventional bound. IEEE Trans. Dependable Secure Comput. **15**, 708–722 (2016)
52. Muhaya, F.T.B.: Cryptanalysis and security enhancement of Zhu's authentication scheme for Telecare medicine information system. Secur. Commun. Netw. **8**(2), 149–158 (2015)
53. Amin, A.R., Islam, S.K.H., Gope, P., Choo, K.K.R., Tapas, N.: Anonymity preserving and lightweight multi-medical server authentication protocol for telecare medical information system. IEEE J. Biomed. Health Inform. **23**, 1749–1759 (2018)
54. Wang, D., Wang, P.: Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity. Inf. Sci. **321**, 162–178 (2015)
55. Wu, F., Xu, L., Kumari, S., Li, X.: A new and secure authentication scheme for wireless sensor networks with formal proof. Peer-to-Peer Netw. Appl. **10**(1), 16–30 (2015)
56. Ali, R., Pal, A.K., Kumari, S., Karuppiah, M., Conti, M.: A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. Future Gener. Comput. Syst. **84**, 200–215 (2017)
57. Luo, H., Wen, G.J., Su, J.: Lightweight three factor scheme for real-time data access in wireless sensor networks. Wireless Netw. **26**(11), 955–970 (2018)
58. Roy, S., Das, A.K., Chatterjee, S., Chattopadhyay, S., Rodrigues, J.J.: Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. IEEE Trans. Ind. Inf. **15**, 457–468 (2018)
59. Wan, T., Liu, X., Liao, W., Jiang, N.: Cryptanalysis and improvement of a smart card based authentication scheme for multi-server architecture using ECC. Int. J. Netw. Secur. **21**(6), 993–1002 (2019)
60. Wei, J.H., Liu, W.F., Hu, X.X.: Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture. Wireless Pers. Commun. **77**(3), 2255–2269 (2014)
61. Wang, B., Ma, M.D.: A smart card based efficient and secured multi-server authentication scheme. Wireless Pers. Commun. **68**(2), 361–378 (2013)
62. He, D.B., Wu, S.H.: Security flaws in a smart card based authentication scheme for multi-server environment. Wireless Pers. Commun. **70**(1), 323–329 (2013)
63. Pippal, R.S., Jaidhar, C.D., Tapaswi, S.: Robust smart card authentication scheme for multi-server architecture. Wireless Pers. Commun. **72**(1), 729–745 (2013)
64. Naeem, M., Chaudhry, S.A., Mahmood, K., Karuppiah, M., Kumari, S.: A scalable and secure RFID mutual authentication protocol using ECC for Internet of Things. Int. J. Commun. Syst. **33**(13), 3906 (2019)
65. Tewari, A., Gupta, B.B.: Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. J. Supercomput. **73**(3), 1085–1102 (2017)
66. Braeken, A.: Highly efficient symmetric key based authentication and key agreement protocol using Keccak. Sensors **20**(8), 2160 (2020)
67. Kumar, P., Braeken, A., Gurtov, A., Iinatti, J., Ha, P.H.: Anonymous secure framework in connected smart home environments. IEEE Trans. Inf. Forensics Secur. **12**(4), 968–979 (2017)
68. Lara, E., Aguilar, L., Sanchez, M.A., Garcia, J.A.: Lightweight authentication protocol for M2M communications of resource-constrained devices in industrial internet of things. Sensors **20**(2), 501 (2020)
69. Chen, J., Gui, Z., Ji, S., Shen, J., Tan, H., Tang, Y.: Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. J. Netw. Comput. Appl. **106**, 117–123 (2018)
70. Mansoor, K., Ghani, A., Chaudhry, S.A., Shamshirband, S., Ghayyur, S.A.K., Mosavi, A.: Securing IoT-based RFID systems: a robust authentication protocol using symmetric cryptography. Sensors **19**(21), 4752 (2019)
71. Avoine, G., Canard, S., Ferreira, L.: Symmetric-key authenticated key exchange (SAKE) with perfect forward secrecy. In: Topics in Cryptology-CT-RSA 2020. Lecture Notes Computer Science, vol. 12006, pp. 199–224 (2020)
72. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Advances in Cryptology EUROCRYPT 2001, pp. 453–474. Springer (2001)
73. Braeken, A., Kumar, P., Martin, A.: Efficient and provably secure key agreement for modern smart metering communications. Energies **11**(10), 2662 (2018)
74. Odelu, V., Kumar, A., Wazid, M., Conti, M.: Provably secure authenticated key agreement scheme for smart grid. IEEE Trans. Smart Grid **9**, 1900–1910 (2018)
75. Chen, Y., Martinez, J.G., Catellejo, P., Lopez, L.: An anonymous authentication and key establish scheme for smart grid: FAuth. Energies **10**, 1345 (2018)
76. Abbasinezhad-Mood, D., Nikoohgadam, M.: Anonymous ECC-based self-certified key distribution scheme for smart grid. IEEE Trans. Ind. Electron. **65**(8), 7996–8004 (2018)