electronics

MDPI

Article

5G-Compliant Authentication Protocol for RFID

Jorge Munilla [1,*,†], Adel Hassan [2,†] and Mike Burmester [3,†]

[1] E.T.S.I. de Telecomunicación, Departamento de Ingeniería de Comunicaciones, Universidad de Málaga, 29071 Málaga, Spain
[2] Computer Science and Information Technology Department, Al-Quds University, Main Campus, Abu Dis P.O. Box 89, Palestine; adel7377@gmail.com
[3] Computer Science Department, Florida State University, Tallahassee, FL 32306-4530, USA; burmester@cs.fsu.edu
* Correspondence: munilla@ic.uma.es; Tel.: +34-952-134-166
† These authors contributed equally to this work.

Received: 21 October 2020; Accepted: 15 November 2020; Published: 19 November 2020

Abstract: The term "Internet of Things" was originally coined when radio frequency identification (RFID) technology was being developed to refer to applications where RFID tagged objects and sensors enabled computers to achieve effective situational awareness without human intervention. Currently, this term encompasses a myriad of medium/small devices connected to the Internet. On the other hand, 5G is a key enabling technology that will support next generation wireless communications. Moreover, 5G aims to realize the "Internet of Everything". Surprisingly, despite the expected relationship between these two technologies, RFID tags have not been properly integrated into 4G and it is not clear if this will change in 5G. RFID is considered as a parallel technology where, at best, it has connection to the core network using back-end servers as gateways between the two technologies. With the aim of overcoming this problem, this paper proposes a 5G compliant RFID protocol that allows RFID tags to act as fully fledged 5G subscribers while taking into account the main characteristics of RFID systems. This proposal leverages the separation between USIM and mobile equipment within the user equipment to implement a 5G compliant protocol where tags accomplish the authentication part, as 5G subscribers, while readers assume the mobile equipment role, carrying out the 5G communication and most of the resource consuming tasks.

Keywords: 5G; 5G-AKA; security; USIM; primary authentication; RFID; EPC

1. Introduction

Cellular network technologies have evolved from 1G, that supported radio communications, through voice-only 2G technology and 3G networks, supporting Internet connections, to the faster and mostly used currently 4G. The standards of the two latter technologies, 3G and 4G, have been designed by the 3GPP consortium. This consortium has also been working to develop their successor 5G, which aims not just to be faster and more reliable, but also to become the key enabler to support the next generation wireless communications. Moreover, 5G aims to realize the "Internet of Everything" and thus, it is designed to support three main services: enhanced mobile broadband (eMBB), massive machine type communications (mMTC) and ultra-reliable and low latency communications (uRLLC). Nevertheless, some privacy issues detected in prior generations have increased the distrust in this new technology, and its security has become a crucial aspect that could derail, or at least delay, its imminent deployment.

On the other hand, radio frequency identification (RFID) is a widely deployed technology for supply chain management, inventory, retail operations and more generally automatic identification. A typical RFID deployment has three main components: tags or transponders, which are electronic

data storage devices attached to (or embedded in) objects to be identified; readers or interrogators, that manage tag population, read data from and write data to tags; and back-end servers, which exchange information with the readers and processes data according to specific task applications. Although the term "Internet of Things" (IoT) currently encompasses a myriad of medium/small devices connected to the Internet, it was originally coined in 1999, when RFID technology was being developed, to refer to applications where RFID-tagged objects and sensors enabled computers to achieve effective situational awareness without human intervention [1].

Surprisingly, despite the importance of RFID for the development of the IoT, and the fact that, as mentioned, 5G provides the mMTC service which improves the existing NB-IoT and LTE-M services introduced in 2015, RFID technology has not found its own space within mobile networks, and its integration is not as one would expect, considering RFID tags as fully fledged subscribers, but just as an independent system connected through back-end servers that act as gateways between RFID and 5G.

Apart from communication incompatibility, the main reason for this is that most RFID tags are not capable of implementing the cryptographic functions required for secure communication (ciphering and integrity check) and to compute the cryptographic results required to complete the 5G-AKA (authentication and key agreement) protocol, which the 3GPP has established for the authentication of subscribers in 5G. As a way to overcome these problems, this paper proposes an RFID protocol which is totally compatible with 5G-AKA. In this 5G-compliant RFID protocol, each tag acts as a 5G subscriber, keeping its own credentials and carrying out its authentication, but the most hardware demanding operations, including the public key computations, are transferred to the reader. It is proven that this transfer does not endanger the security of the system, since the most sensible cryptographic material is not revealed to the reader at any moment. Thus, as 5G subscribers can operate with different devices (e.g., different smart phones), the tags can likewise be interrogated by different readers with the guarantee that previous readers cannot access current exchanged data.

The outline of the rest of the paper is as follows. Section 2 describes the 5G-AKA protocol and the main 5G modules involved in the authentication procedure. Section 3 introduces the proposed protocol, detailing how the previous protocol can be adapted to be suitable for RFID systems. Then, Section 4 analyzes the security of this proposal and, finally, Section 5 concludes the paper.

2. Security Architecture and Procedures for 5G and Threat Model

Security architecture and procedures for 5G systems are mainly defined in the technical specification 3GPP TS 33.501 [2]. This section summarizes the main aspects extracted from this and related documents.

2.1. 5G Architecture

Two different sides can be identified in 5G: the subscriber or user equipment (UE) side and the network side. The UE contains the mobile equipment (ME) of the subscriber, typically a smartphone or an IoT device, that is equipped with a universal subscriber identity module (USIM), which keeps the security data and computes the cryptographic results required to complete the 5G-AKA. Plastic SIM cards have traditionally been used, but their cost, that includes a SIM card reader, and the difficulty associated with their substitution, make them unsuitable for IoT devices. In such cases, embedded SIMs (eSIMs), implemented on chips, where the credentials are provisioned remotely, are preferred. On the network side, we can further distinguish the service network (SN) and the home network (HN). The HN belongs to the subscriber's operator (that contains the subscriber's credentials in a database) and grants access to the different services within the core network, while the SN has base stations in the subscriber's vicinity and provides physical access to the network. The SN and the HN may both belong to the subscriber's operator (or PLMN—public land mobile network) or not, as happens, for example, when roaming. Figure 1 illustrates the 5G architecture with the most relevant components, from a security point of view. We note that these components do not always correspond to hardware

modules, but could be just software functions, since 5G uses NFV (network function virtualization) and SDN (software define networks) as implementation concepts.

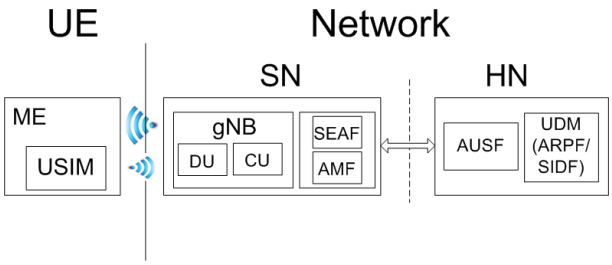

Figure 1. Here: 5G architecture and the main elements involved in the security procedures.

The subscription credentials shared by the USIM and HN consist of, at least, the $SUPI$ (subscription permanent identifier) and a long-term key K, used to uniquely identify the subscriber and mutually authenticate the UE and the 5G core network. However, to implement 5G authentication, it is also required that USIM stores the public key pk_{HN} of HN, used to encrypt the $SUPI$ (with an asymmetric key cryptosystem based on elliptic curves) and obtain the $SUCI$ (subscription concealed identifier), and a sequence number SQ_{UE}, used to prevent replay attacks. The associated HN, on its side, stores the corresponding private key sk_{HN}, and another sequence number SQ_{HN}, loosely synchronized with SQ_{UE}. Although, as explained, USIM carries out the authentication, based on the home operator's decision as indicated by the USIM, some operations can be performed by the ME, such as the public key encryption of its identity (to obtain $SUCI$). Once authentication is granted, ME is given information to compute an anchor key, K_{SEAF}. K_{SEAF} is used for the derivation of security keys for the secure communication between UE and SN. The same K_{SEAF} can be used in subsequent procedures, preventing the need of new authentication runs: for example, when keys for more security context are required.

The SN provides physical access via its base stations or gNBs (next generation Node B), which consist of distributed and central units (DUs and CUs). DUs have the antennas and do not implement security functions as they may be deployed in unsupervised sites, while CUs control one or more DUs and perform confidentiality and security functions, using keys derived from the anchor key K_{SEAF}, provided by HN and stored in the security anchor function (SEAF). Finally, the SN also contains or implements the AMF (access and mobility management function), which offers functionalities such as registration, reachability, mobility and connection management services (equivalent to MME in 4G).

In the HN, we can highlight the unified data management (UDM), which manages subscriber information, including what is required for authentication; the authentication server function (AUSF), which handles authentication requests, computes and provides K_{SEAF} to the SEAF of the SN, and informs the UDM that a successful or unsuccessful authentication of a subscriber has occurred; the authentication credential repository and processing function (ARPF) which keeps the authentication credentials and provides them to AUSF; and the subscriber identity de-concealing function (SIDF) that is responsible for de-concealing the $SUPI$ from the $SUCI$.

2.2. The 5G Authentication and Key Agreement Protocol, 5G-AKA

A primary authentication is compulsory for all the devices, regardless of the service or network access these require (agnostic access network). The purpose of this primary authentication and key agreement procedure is to enable mutual authentication between the UE and the network and provide keying material (K_{SEAF}) that can be used between UE and SN. A secondary authentication is also possible in 5G, but is optional. This is intended for authenticated access to services provided over 5G; e.g., access to corporate data, and is beyond the scope of this paper (for more information consult [2]).

Moreover, 3GPP proposes two AKA protocols in the standard to be used for primary authentication: EAP-AKA and 5G-AKA. This paper focuses on the most recent one: 5G-AKA. EAP is

very similar but uses slightly different flows and key derivation. For more details, we refer the reader to [3] and Subclause 6.1.3.1 of [2].

Before describing the specific messages exchanged in 5G-AKA, we discuss its main aspects:

A. Shared cryptographic material. HN and UE (USIM) share a long-term symmetric key K. All the UEs belonging to that HN also store the public key pk_{HN} of HN, while HN stores the corresponding secret key sk_{HN}.
B. Sequence numbers. UE and HN share a loosely-synchronized sequence number SQ: UE stores SQ_{UE}, while HN stores SQ_{HN}. These are checked during the authentication procedure to prevent replay attacks.
C. Cryptographic functions. The 3GPP architecture defines a key derivation function KDF, and seven cryptographic functions, $f1$, $f1^*$, $f2$, $f3$, $f4$, $f5$ and $f5^*$ [4]: the first three are message authentication functions, while the last four are key derivation functions. These are one-way keyed functions that should be practically indistinguishable from independent random functions. In particular: (i) knowledge of the values of one function on a fairly large number of given inputs should not enable its values to be predicted on other inputs; (ii) outputs from any one function should not be predictable from those of the other functions (on the same or other inputs); (iii) it should be infeasible to determine any part of the secret key, by manipulation of the inputs and examination of the outputs of the algorithm. These requirements are broadly captured by pseudo-random functions (PRF) and pairwise independence (Section 19.3.2.2, p. 19 [5]), and in particular jointly PRFs [6]. Informally, the PRFs $hi : \{0,1\}^n \times \{0,1\}^{s_i} \rightarrow \{0,1\}^{t_i}$, $i = 1, \ldots, m$, are jointly PRFs if for any key $k \in \{0,1\}^n$, the output of $(h1(k,\cdot), \ldots, hm(k,\cdot))$ is practically indistinguishable from the output of $(g_1(\cdot), \ldots, g_m(\cdot))$, on the same input, where $g_i : \{0,1\}^{s_i} \rightarrow \{0,1\}^{t_i}$, $i = 1, \ldots, m$, are functions drawn uniformly from the set of all functions from $\{0,1\}^{s_i}$ to $\{0,1\}^{t_i}$. Finally, the KDF is used to derive the keys for secure communication. It is specified in [7], as a keyed hash function based on SHA256 [8].
D. The MILENAGE algorithms [5]. Although the implementation of $f1$–$f5$ and $f1^*$, $f5^*$ is proprietary and not standardized, the 3GPP security working group has developed an example algorithm set for these authentication and key generation functions. This uses a 128-bit blockcipher (AES-128) as a kernel function. The seven functions $f1$–$f5$ and $f1^*$,$f5^*$ are constructed by invoking the kernel function and using circular shifts $r_1, \ldots, r_5$ and xor-ing constants $c_1, \ldots, c_5$, selected appropriately to ensure separation (independence). Furthermore, $f1$ and $f1^*$ use a CBC MAC structure, while the others use double encryption in counter mode.
E. Identifiers. The $SUCI$ is, as explained, the encryption of the $SUPI$ with the public key of HN using a random number ne : $SUCI = \{SUPI\}^{ne}_{pk}$. The $SUCI$ is used to identify the UE, while the $SUPI$ is never sent clearly. This prevents a security flaw in 4G that allows an attacker to get the permanent identifier of an UE ($IMSI$) by impersonating a serving network ($IMSI$ catcher attack [9]). After a successful execution of 5G-AKA, the SN can issue a pseudonym called $GUTI$ (globally unique temporary UE identity) that the UE can be used to identify itself for a certain period of time [10]. This avoids computing the $SUCI$ that requires generating a random number ne and computing an assymmetric encryption. The $GUTI$ is updated by AMF upon receiving a registration request message from a UE of type "initial registration", "mobility registration update", "periodic registration update" or in response to a paging message, but it is left to the operator to re-assign it more frequently.
F. Assumptions on Channels. The channel between UE and SN, on the radio physical layer, is subject to attacks by passive and active adversaries (Section 2.3). By contrast, the channel between the SN and HN is supposed to be secure (Clause 5.9.3 [2]).

Figure 2 describes the flows of the 5G-AKA protocol, where $\oplus$ and $||$ stand for xor and concatenation, respectively. For simplicity, we just identify three parties: UE, SN and HN, integrating the different sub-entities (SEAF, AUSF, UDM...), as we do not require this level of granularity. Nevertheless, within UE,

we distinguish between computations in the USIM and in the ME, since this is relevant for the new proposal described in the next section. The protocol consists of three phases:

- The first phase is the initiation of the authentication procedure and the selection of the authentication method. The SEAF of SN may initiate an authentication with the UE during any procedure establishing a signalling connection with the UE. If UE does not have a valid $GUTI$, then it computes and sends the $SUCI$ to SN, who relays it to HN. Otherwise, UE sends the $GUTI$ to SN. If the SN (AMF) is able to obtain the corresponding $SUPI$, by looking it up in its database, then SN forwards it to HN. If not, it requests ("Identifier Request Message") the $SUCI$ to UE and relays it to HN. SN includes its identifier (SN_{name}) in the message to HN. The HN (AUSF) then checks whether the SN is authorized and if so, obtains the $SUPI$: directly if it was sent or, otherwise, deconcealing it from the $SUCI$ (at SIDF). Based on $SUPI$, the HN (UDM/ARPF) shall choose the authentication method.
- The second phase is the actual authentication procedure, which is based on a typical challenge–response mechanism. Thus, upon receiving a request for authentication, HN generates a random number R, which is used as a challenge, and an authentication value $AUTN$, which consists of a message authentication code MAC (using $f1$) and a value, $CONC$ (using $f5$), which masks the sequence number SQ_{HN}, required to compute the MAC and included to prevent replay attacks. It additionally computes the response to this challenge ($XRES$) and sends its hashed value ($HXRES$) to SN, so that this can detect incorrect responses, even when it is not able to compute the correct ones. The use of $HXRES$ reduces the vulnerability to denial of service (DoS) attacks, since junk messages can be detected earlier. The response involves the long-term key K, the challenge R and the identifier of SN, to guarantee that both parties are communicating with the same SN. UE receives the challenge and the authentication value: $R, AUTN$. Then, USIM retrieves SQ_{HN}, computes the MAC and makes a twofold check: (i.) that the MAC is correct and therefore the authenticity of the message, and (ii.) that SQ_{HN} has not been replayed ($SQ_{HN} > SQ_{UE}$) and that it is within a certain range ($SQ_{HN} < SQ_{UE} + \Delta$), to prevent de-synchronization attacks by forcing the counter to wrap around. If (i.) fails, then a "MAC failure message" is sent. If (i.) is correct but (ii.) fails, then a "Synchronization failure message" is replied along with a re-sync token to inform to HN of the value SQ_{UE} in a hidden way; using $f1^*$ for authentication and $f5^*$ to mask and protect it from eavesdroppers. Otherwise, if (i.) and (ii.) are correct, USIM computes RES (using $f2$), CK (using $f3$) and IK (using $f4$) and forwards it to ME, who computes and sends $XRES$ and derives the anchor key K_{SEAF}. SN verifies that the hashed values of this response is correct and if so, forwards it to HN. Finally, HN verifies the response and if correct, sends K_{SEAF} to SN.
- Final phase. A successful 5G-AKA ends up with the derivation of the anchor key K_{SEAF} by both HN and UE, from which further keys can be obtained. The authentication is implicit [11], since the authentication confirmation occurs only when the parties succeed in exchange messages correctly using the derived keys. The standard does not specify any additional key confirmation query for K_{SEAF}. If the exchanged messages are not correct, then the parties assume that either the derived keys or K_{SEAF} are not correct, and as a result, that the authentication process was not successful.

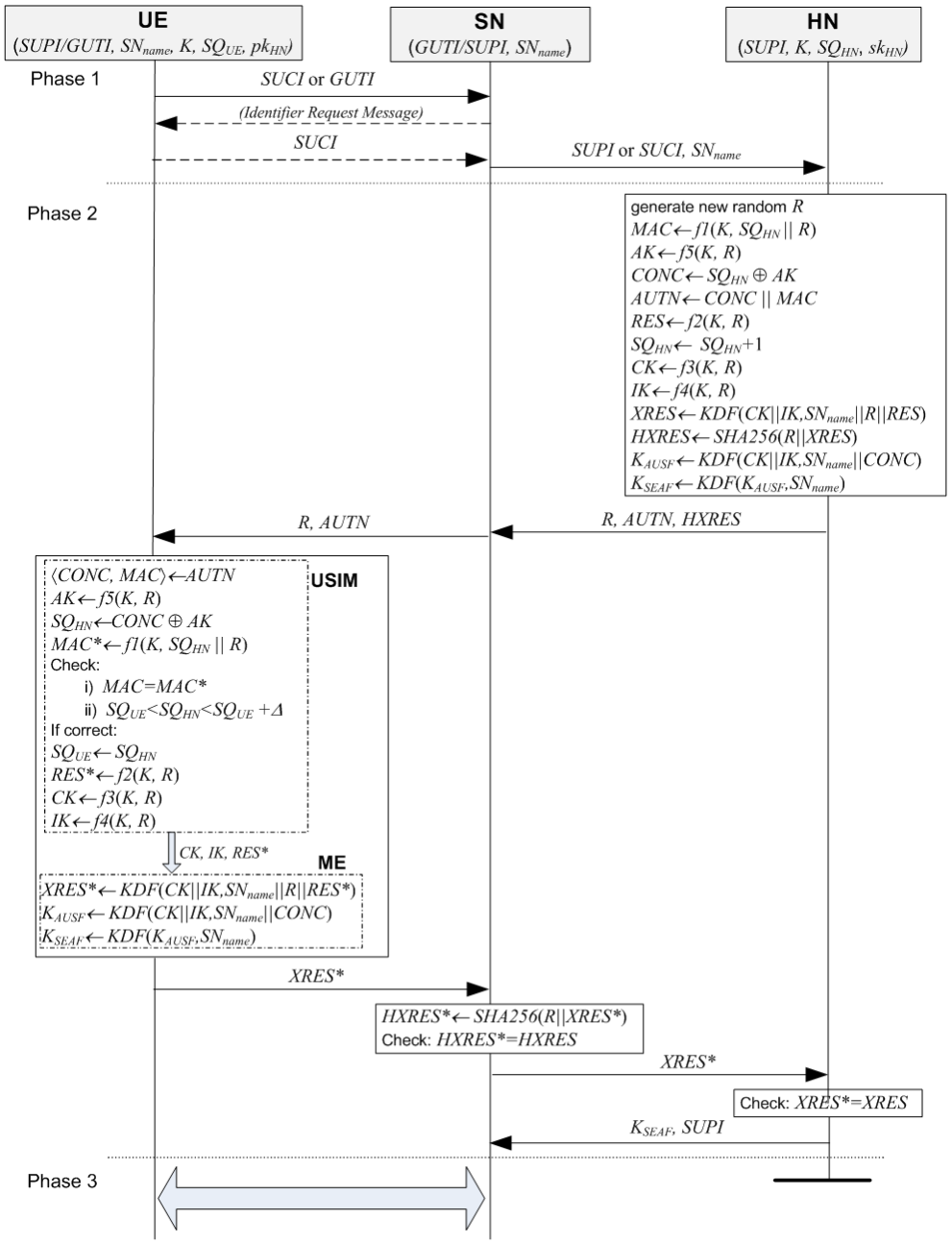

Figure 2. 5G-authentication and key agreement (AKA).

2.3. Threat Model

We use the standard model in which parties (including adversaries) are modeled by probabilistic polynomial time (PPT) Turing machines. We use the Dolev–Yao (DY) model [12] to model the ability (control) of adversaries. In this model, an adversary controls the network, and can eavesdrop on the communication (passive adversary), as well as intercept, inject, manipulate or drop messages (active adversary). In our proofs, we model adversaries by PPT algorithms. Security is based on indistinguishability.

3. Proposed Protocol

This section describes a 5G-compliant protocol for RFID systems. Although several authors have proposed changes to the 5G-AKA protocol [6,11,13], mainly focused on improving its privacy, our protocol follows the current version of 5G-AKA (Release 16), as described earlier. It is out of the

scope of this paper to analyze these proposals, and the security claims and assumptions of the standard are adopted here.

In the 5G-compliant RFID protocol, the RFID tags are the 5G subscribers, while the readers and the back-end servers physically connect the tags to 5G. For simplicity, the readers and back-end servers are identified as a single entity from a security point of view (called just the reader). These high-level entities could be implemented in practice on the same device but, even if this is not the case, they are able to perform complex cryptographic operations, such as asymmetric encryption/decryption and digital signatures/verification, so that the channel between them can be considered secure. Although inductive RFID tags that operate in ranges of centimeters (near field communication—NFC), can be used in secure applications implementing complex cryptographic algorithms; most of the RFID tags currently used are UHF passive tags. These tags have no power of their own and use backscatter coupling, as they operate in the far field [14]. As a result, they work at much higher distances, of up to ten meters, but the delivered power is low. This, along with the fact that low cost is also a common requirement, means that lightweight or not too complex cryptographic tools must be used [15]. Our proposal focuses on such tags and it shall assume that these are able to perform one-way symmetric cryptographic operations, select pseudo-random numbers, and use functions that integrate strong security services based on the MILENAGE algorithms (Section 2.2, Assumption D). The case of NFC tags, by contrast, where tags work very close to the reader and therefore attacks in the air channel become less relevant, and are able to compute asymmetric cryptographic primitives, may be better analyzed as a particular case of SIM card implementation.

The main design principle of the protocol is to move the most complex (asymmetric encryption) and computationally intensive operations (ciphered and integrity verification of the communication) from the tag to the reader. For this purpose, the protocol leverages the task division between the USIM and the ME within the UE, associating these with the tag and the reader, respectively, and assuming a similar degree of trust between them (see Figure 3) . This trust is temporal as in 5G: that is, as the subscribers can use their USIM in another ME without the risk of being impersonated, the tags likewise can be assigned to another reader without the previous reader being able to impersonate them. The protocol uses six additional cryptographic functions $h1, \ldots, h6$, with the first four used for privacy protection, while the last two are used for integrity protection. These are keyed PRFs that can use, likewise to the functions in 5G-AKA, the same kernel based on AES-128, with different circular shifts $r'_1, \ldots, r'_6$ and/or constants $c'_1, \ldots, c'_6$, for strong one-wayness and pairwise independence (Section 2.2). There are several AES-128 implementations designed for RFID applications and optimized for low resource requirements [16,17], and the cost of implementing right circular shifts and xor's is almost negligible.

Figure 3. Equivalent Architecture for the 5G-compliant radio frequency identification (RFID) protocol.

3.1. 5G-Compliant RFID Initialization Protocol

A RFID setup is assumed in which the tag's identity, id_t, and a symmetric key, k_t, are securely provisioned to the tag and the reader. New values for these parameters will be used if the tag is assigned later to another reader. The required information to access the 5G network, that is: $SUCI$ and pk_{HN}, is transferred from the tag to the reader using an initialization protocol. The RFID initialization protocol is sketched in Figure 4.

In the initialization RFID protocol, the tag generates a nonce r_i and computes three masking values, $AK1i$, $AK2i$, $AK3i$, using $h1, h2, h3$, to protect $SUPI$ and pk_{HN}. Then, a message authentication code $AUTNi$ is computed, using $h5$, for both authenticity and integrity verification. Timers, although not

included in the descriptions, are used by the parties to close sessions if no response is received after a certain time. The reader first looks in its database (DB) for any key k_n that matches with $AUTNi$. If no key is found, the process is aborted. Otherwise, the tag is identified as id_t, and the extracted values $SUPI$ and pk_{HN} are assigned to it. A confirmation message, $CONFi$, using $h6$, is then computed and sent to the tag. If this message is not received, because either it was intercepted or the reader did not send it (something was wrong or the initial message was not received), the tag, according to the system policies (to prevent DoS attacks), tries the initialization after a certain time.

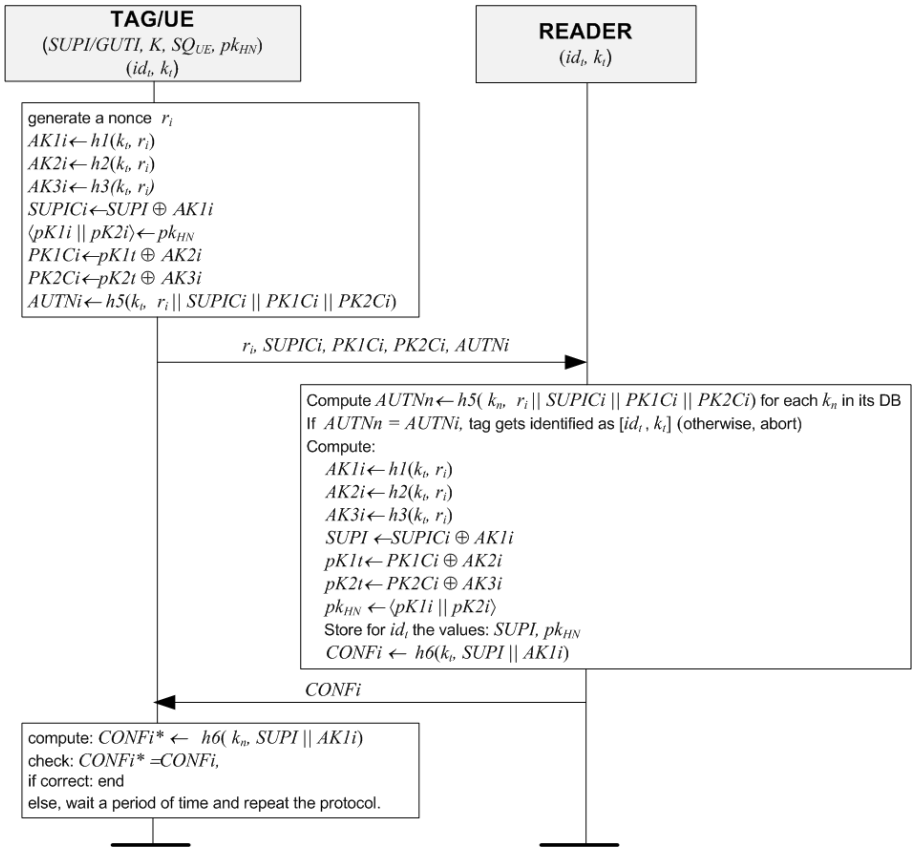

Figure 4. 5G-compliant RFID Initialization Protocol.

3.2. 5G-Compliant RFID Authentication Protocol

Once the initialization process is completed, the reader is capable of accessing a service network in its proximity under the tag's subscriber identity. Thus, when the tag wants to authenticate itself to the 5G network, the authentication protocol, described in Figure 5, is performed. Interaction between SN and HN are exactly the same as described in the previous section and therefore have not been included in the figure. The protocol starts with the tag generating a session nonce and sending the reader an authentication request. This authentication request is masked with the value $AK1t$ (computed using $h1$), and sent along with the message authentication code $AUTNt$ (computed using $h6$), to guarantee its authenticity and integrity. Likewise in the initialization protocol, the reader looks up in its DB for a key k_t that matches with $AUTNt$; if it is not found, the process is aborted, otherwise the tag gets identified as id_t and the reader initiates the communication with the SN in its vicinity, whose identifier is SN_{name}. If the reader has a previous valid $GUTI$ for that id_t, then it sends it to SN ($IDENT = GUTI$), otherwise it computes and sends the $SUCI$ ($IDENT = SUCI$). If $GUTI$ was sent but an "Identifier Request Message was received", then the Reader computes and sends the $SUCI$ ($IDENT^* = SUCI$). After the tag is identified as a 5G subscriber by HN, the reader receives the challenge and forwards it to the tag. The tag acts then as in the 5G-AKA protocol; it checks the

authenticity of the challenge, using $AUTN$, and the freshness, verifying SQ_{UE}. If they are correct, then the tag computes RES^* and the keys CK and IK, and masks them computing the values $AK2_t$, $AK3t$ and $AK4t$ (computed using $h2$, $h3$ and $h4$), to obtain $RESt$, CKt and IKt. Next, the tag sends these to the reader along with a message authentication code $INTRESt$ (computed using $h5$). Upon receiving these values, the reader checks that $INTRESt$ is correct and if so, computes the masking values and retrieves RES^*, and the keys CK and IK. Now, the reader, acting as the ME of the UE, computes and forwards $XRES^*$ to SN, and derives the anchor key K_{SEAF}. If HN accepts the response, the tag is authenticated as a 5G subscriber. From that point on, the communication between the tag and the reader, which is expected to consist of not very frequent small-data, is protected using k_t, while the 5G communication (implementing the different layers of the communication protocol) is carried out by the reader.

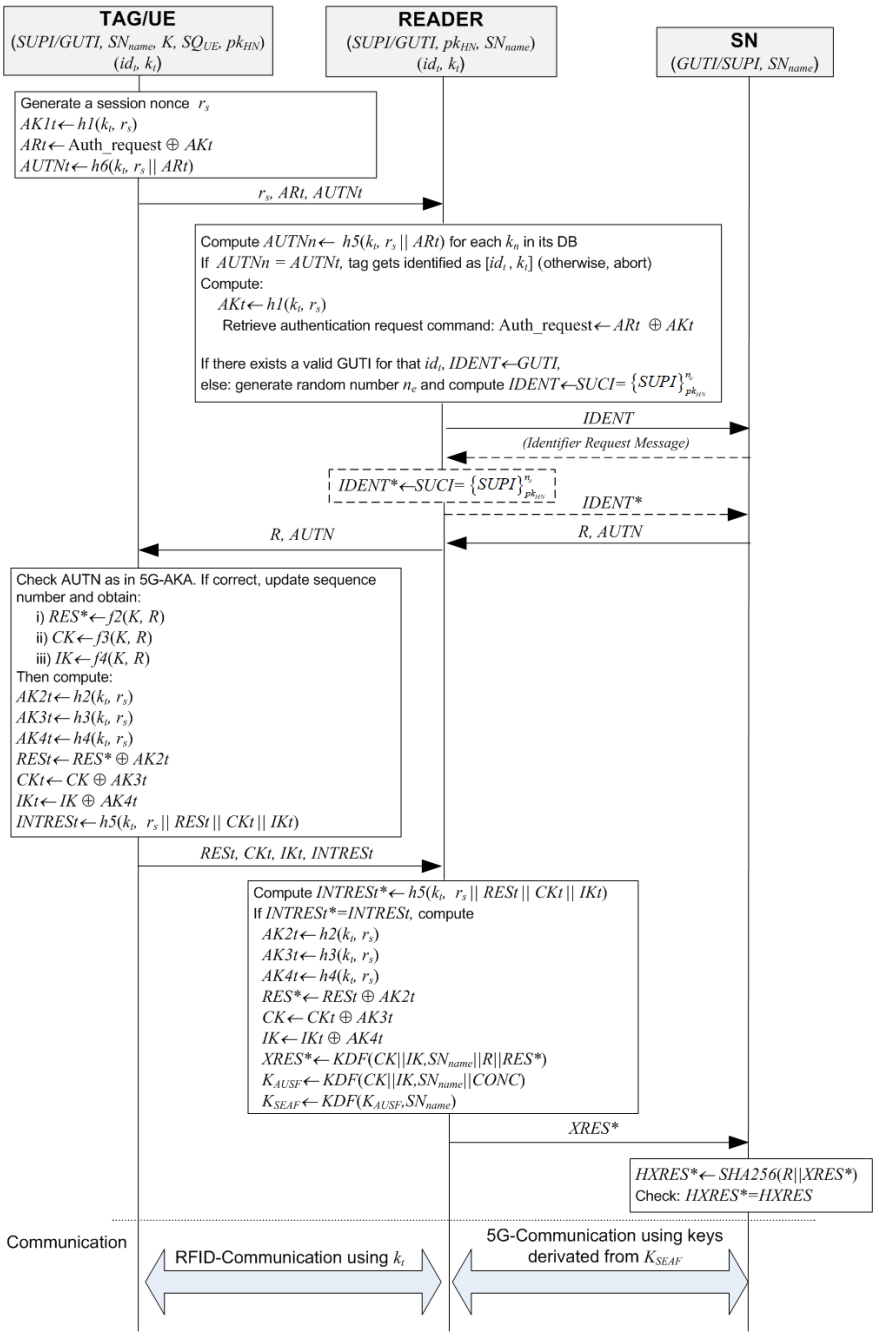

Figure 5. 5G-compliant RFID Authentication Protocol.

4. Analysis

As explained, the goal of the proposed protocol is to match the 5G-security level taking into account the hardware and communication constraints of RFID systems. We analyze then the protocol considering three main adversaries: passive adversaries, active adversaries and (previous) readers.

4.1. Passive Adversaries

According to 5G, the $SUPI$ and sequence number (that also leaks information about the frequency of connections) must be protected from eavesdroppers to guarantee the privacy of the subscribers. In the 5G-compliant RFID protocol the sequence number is not involved in the RFID part and the $SUCI$ is protected during the initialization procedure by using the masking value $AK1$.

Theorem 1. *A passive adversary cannot distinguish the messages exchanged during the 5G-compliant RFID protocol from random messages with probability better than negligible (in terms of the length n of the key k).*

Proof. We first prove the result for the initialization procedure (Lemma 1) and then for the main protocol (Lemma 2). We note that one cannot extract any specific message from a pseudo-random message with probability better than negligible. □

Lemma 1. *A passive adversary cannot distinguish the exchanged messages from random messages with probability better than negligible during the initialization procedure.*

Proof. (Sketch) The messages exchanged during the initialization procedure are:

$$\langle r_i, SUPICi, PKC1Ci, PKC2Ci, AUTNi, CONFi \rangle.$$

We are assuming that $h1(k,\cdot), h2(k,\cdot), h3(k,\cdot), h5(k,\cdot), h6(k,\cdot)$ are jointly PRF. Note that if we XOR any constant values to one or more of these functions, then we do not loose pseudo-randomness. In particular, for the constants $SUPI$, $pk1$, $pk2$: the functions,

$$\begin{array}{rcl} h1^*(k,\cdot) & = & h1(k,\cdot) \oplus SUPI \\ h2^*(k,\cdot) & = & h2(k,\cdot) \oplus pk1 \\ h3^*(k,\cdot) & = & h3(k,\cdot) \oplus pk2 \\ h5^*(k,\cdot) & = & h5(k,\cdot), \text{ and} \\ h6^*(k,\cdot) & = & h6(k,\cdot), \end{array}$$

are jointly PRF. However, these functions for joint input:

$$\langle r_i, r_i, r_i, r_i \| SUPICi \| PK1Ci \| PK2Ci, SUPIi \| AK1i \rangle,$$

will output the exchanged messages $\langle SUPICi, PKC1Ci, PKC2Ci, AUTNi, CONFi \rangle$. It follows that they must be pseudo-random. □

Lemma 2. *A passive adversary cannot distinguish the exchanged messages from random messages with probability better than negligible during the authentication process.*

Proof. (Sketch) We use a similar approach. In this case, the messages exchanged are:

$$\langle r_s, ARt, AUTNt, RESt, CKt, IKt, AUTN, INTRESt, XRES^* \rangle.$$

The first five (excluding r_s) can be defined in terms of the functions: $h1, h6, h2, h3, h4$ that are jointly PRF. These functions, on joint input:

$$\langle r_s, r_s \| ARt, r_s, r_s, r_s \rangle$$

will output the five exchanged messages (excluding r_s). $AUTN$, $INTRESt$ and $XRES^*$ are 5G-AKA messages generated by $(f1, f5)$, $h5$ and a KDF that are also pseudo-random and independent. The $SUPI$ is not involved in the communication between the tag and the reader, and only its randomized asymmetric encryption $SUCI$ is transmitted by the reader if SN requests it. This uses an independent random number n_e. It follows that the exchanged messages are jointly pseudo-random. □

4.2. Active Adversaries

An adversary, according to the assumed threat model, can intercept, replay and/or modify messages exchanged by the tag and the reader. Replaying messages from previous sessions is prevented by (a) using random numbers r_i and r_s, that randomise all the computed values for that session, and (b) using MACs for integrity checks, which prevent messages being altered in transit. A more formal proof is given below.

Theorem 2. *The 5G-compliant RFID protocol is secure against the alteration of the flows by an active adversary.*

Proof. The proof is divided into two parts. We first prove that the flows exchanged during the initialization: $\langle r_i, SUPICi, PK1Ci, PK2Ci, AUTNi \rangle$ and $\langle CONFi \rangle$ form sets where none of whose elements can be altered and get accepted (Lemma 3). Then, we prove that this also happens for the flows $\langle r_s, ARt, AUTNt \rangle$ and $\langle RESt, CKt, IKt, INTRESt \rangle$ of the tag during the authentication (Lemma 4).

The flow, in which the reader forwards $R, AUTN$ to the tag during the authentication, is not included in the proof because it is identical to the flow in 5G-AKA. Note also that "altered messages" refers to the case where some parts are modified while others remain the same. The case where the entire set is replaced by another set is analyzed later, where two situations can be further distinguished, depending on whether the used set of messages are forged by an adversary, or generated by a genuine party and later replayed (replay attack). □

Lemma 3. *The messages $\langle r_i, SUPICi, PK1Ci, PK2Ci, AUTNi \rangle$ and $\langle CONFi \rangle$ form a set that cannot be altered and get accepted.*

Proof. (Sketch) By contradiction, assume that there exists a probabilistic polynomial time (PPT) adversary (algorithm) that is able to alter this set and get accepted. We first consider $AUTNi$, and distinguish three cases:

(a) $AUTNi$ is not changed and some or all of the other values are changed. This is not possible because it implies that the adversary is able to compute a collision (same output, different input) for the MAC $AUTNi$.
(b) $AUNTi$ is changed but some or all of the other values remain unchanged. Again, this is not possible because it implies that the adversary is able to compute a valid $AUTNi$ for a different input that will get accepted with non-negligible probability, without knowing k_t.
(c) $AUNTi$ and the input is changed. Again the adversary cannot compute with non-negligible probability a valid $AUTNi$ for a given input without knowing k_t.

Next consider $CONFi$. Again, if the adversary only changes values of the input, then this will not be accepted because $CONFi$ is an MAC. Similarly, $CONFi$ cannot be changed and get accepted, because it cannot be computed without knowing k_t. □

Lemma 4. *The messages* $\langle r_s, ARt, AUTNt\rangle$ *and* $\langle RESt, CKt, IKt, INTRESt\rangle$ *form a set that cannot be altered and get accepted.*

Proof. Again, we consider the cases of $AUTNt$ and $INTERSt$ separately. The first case is identical to Lemma 3, by replacing: $AUTNi$ and the input $\langle r_i, SUPI, PK1Ci, PK2Ci\rangle$, by $AUTNt$ and the input $\langle r_s, ARt\rangle$. The same argument applies to $INTERSt$. In this case, we replace $AUTNi$ and input $\langle r_i, SUPI, PK1Ci, PK2Ci\rangle$, by $INTERESt$ and input $\langle r_s, RESt, CKt, IKt\rangle$. □

We have shown that the flows of the 5G-compliant RFID protocol cannot be distinguished from random flows by a PPT adversary $\mathcal{A}$. This implies that $\mathcal{A}$ cannot compute the private key k_t from such flows.

Corollary 1. *An active adversary cannot obtain the private key* k_t *used to generate the flows of the 5G-compliant RFID protocol.*

Proof. We only consider efficient (PPT) adversaries $\mathcal{A}$. If $\mathcal{A}$ can compute k_t, then it can also distinguish the flows of the 5G-compliant RFID protocol from random flows. □

Corollary 2. *An active adversary cannot forge new valid flows of the 5G-compliant RFID protocol without knowing the private key* k_t.

Proof. We have shown that an adversary $\mathcal{A}$ cannot modify the flows of the 5G-compliant RFID protocol so that they remain valid. To compute new valid flows, $\mathcal{A}$ must know the private key used to generate them (the *hi* are one-way keyed cryptographic functions). □

As observed in the previous Section, the authentication in 5G-AKA is implicit. This means that although the parties may accept a message as valid, the authentication is not completed until a valid K_{SEAF} is computed and the derived keys are used. The computation of K_{SEAF} is not possible without the private key, so replay attacks mainly focus on privacy and DoS (de-synchronizing the parties) attacks.

Theorem 3. *The 5G-compliant RFID protocol is secure against replay attacks.*

Proof. We first prove that the replay of the first flow of each protocol: $\langle r_i, SUPICi, PK1Ci, PK2Ci, AUTNi\rangle$, respectively $\langle r_S, ARt, AUTNt\rangle$, do not have any significant effect (Lemmas 5 and 6). Then, we prove that the replay of the second flow of each protocol: $\langle CONFi\rangle$ and $\langle RESt, CKt, IKt, INTRESt\rangle$, respectively, will not be accepted (Lemmas 7 and 8). □

Lemma 5. *The replay of a previously sent flow* $\langle r_i, SUPICi, PK1Ci, PK2Ci, AUTNi\rangle$ *does not have any effect on the initialization procedure of 5G-compliant RFID protocol.*

Proof. Suppose that an adversary intercepts the first flow $\langle r_i, SUPICi, PK1Ci, PK2Ci, AUTNi\rangle$, which in a normal execution would cause the reader to store the values $SUPI$ and pk_{HN}. Later, the adversary, even after the execution of other initialization procedures, replays this flow, impersonating the tag to the Reader. The Reader will check the authenticity of the flow and store $SUPI$ and pk_{HN}. The replay of flows will not have any effect on this protocol. □

Lemma 6. *The replay of a previously sent flow* $\langle r_S, ARt, AUTNt\rangle$ *does not have a negative effect on the authentication procedure of 5G-compliant RFID protocol.*

Proof. Suppose that an adversary intercepts the first flow $\langle r_S, ARt, AUTNt\rangle$, which in a normal execution would cause that reader to initiate an authentication request by sending the tag's subscriber

identity. Later, the adversary, even after the execution of other authentication procedures, replays this flow to the reader, impersonating the tag. The reader will check the authenticity of the flow and initialize the authentication procedure. However, because the genuine tag is not involved, the adversary cannot respond correctly to the challenge and the authentication will not be successful. This is exactly what happens in 5G-AKA when the adversary replays a previously intercepted $GUTI$ or $SUCI$. Therefore, this replay attack does not have any significant effect on 5G-compliant RFID protocol. □

Lemma 7. *The replay of a CONFi message will not be accepted by the tag in a different session.*

Proof. Three options are possible, depending on when the adversary interrupts a previous legitimate communication.

(a) The adversary intercepts the first flow: $\langle r_i, SUPICi, PK1Ci, PK2Ci, AUTNi \rangle$, and later replays it, impersonating the tag to the reader. The reader then computes a valid $CONFi$ that is stored by the adversary.
(b) In the second flow, the adversary intercepts the valid $CONFi$ computed by the Reader.
(c) The adversary does not intercept any flow, just eavesdrops and stores $CONFi$ during a legitimate communication.

Later, after the legitimate tag sends a new first flow $\langle r_i', SUPICi', PK1Ci', PK2Ci', AUTNi' \rangle$, the adversary, impersonating the reader, replays the stored $CONFi$. As this $CONFi$ is different from the unique value $CONFi'$, corresponding to the new set of values singularized by r_i', the message will not be accepted as valid by the tag. □

Lemma 8. *The replay of $\langle RESt, CKt, IKt, INTRESt \rangle$ will not be accepted by either the reader or SN.*

Proof. Suppose an adversary has stored the valid flows from a previous session, singularized by r_s and R: $\langle r_s, ARt, AUTNt \rangle$, $\langle R, AUTN \rangle$, $\langle RESt, CKt, IKt, INTRESt \rangle$ and $\langle XRES^* \rangle$. Then, there are two possible options:

(a) After a new first flow from a legitimate tag $\langle r_S', ARt', AUTNt' \rangle$ and the corresponding challenge $\langle R', AUTN' \rangle$, the adversary impersonates the tag and replays the stored flow $\langle RESt, CKt, IKt, INTRESt \rangle$. This will not be accepted by the reader because $INTRESt$ is different from the unique value $INTRESt'$, that corresponds to the new set of values singularized by r_s'.
(b) The adversary, impersonating the tag, replays the first flow $\langle r_S, ARt, AUTNt \rangle$, receives a new challenge $\langle R', AUTN' \rangle$ from the Reader, and replays the second flow $\langle RESt, CKt, IKt, INTRESt \rangle$. In this case, the reader will check that $INTRESt$ is correct, according to the received flows, and then compute and send $XRES^*$. However, because $XRES^*$ is different from $XRES^{*\prime}$, that corresponds to the challenge R', the hashed value $HXRES^*$ will not be accepted by SN.

□

4.3. Readers

Finally, given the special characteristics of the proposed protocol, we analyze the sensitive information given to, or computed by, the readers and the expected trusted level of these parties. In 5G-AKA the following sensitive information is available to ME: CK, IK, RES^*, and $SUCI$ and pk_{HN}, if the operator decides that it computes $SUCI$, are given by the USIM to ME. Additionally, ME computes $XRES^*$, K_{AUSF} and the anchor key K_{SEAF}. Finally, $GUTI$ is provisioned by SN to ME. On the other hand, in 5G-AKA after a successful authentication, the following information is given to SN by HN: $SUPI$ (this is not technically needed, but is done for legal reasons to be able to respond to lawful Interception requests [6]), $GUTI$ and K_{SEAF}. It must be noted that, according to 5G security requisites, the key K_{SEAF} that is established in a given session must remain confidential even when the attacker learns the K_{SEAF} keys established in other sessions [2], so that future K_{SEAF} keys will remain

secure from previous ME and/or SN. As a result, only sensitive information related to the privacy, namely the $SUPI$, is leaked to previous ME and SN, which could then use this information to continue tracing the tag.

In the proposed 5G-compliant RFID protocol, readers can be assimilated into ME. Readers receive exactly the same information as ME (when they compute the SUCI). In particular, they do not have access to K, SQ_{UE}, nor do they need to know the proprietary message authentication functions. Thus, if the tag moves from one reader to another (with another id_t and k_t), the change is similar to when a user puts the USIM into different MEs. In particular, as 5G subscribers can use the USIM in different devices, likewise the tags can be interrogated by different readers with the guarantee that previous readers cannot impersonate them.

5. Conclusions

We have presented a 5G-compliant protocol for RFID systems. In this protocol, RFID tags act as 5G subscribers, while the readers and back end servers physically connect the tags to 5G networks. This protocol is designed so that the complex and computationally intensive operations, like asymmetric encryption and implementation of the communication protocol stack, including privacy and integrity verification, are transferred from the tag to the reader (that is equated with the ME). RFID tags only need to perform one-way symmetric cryptographic operations (based on AES-128) and select pseudo-random numbers. In this protocol, the trust in readers is temporal, similar to the trust in MEs with the 5G-AKA protocol. In particular, tags can be interrogated by different readers, without previous readers being able to impersonate them. We prove that this protocol is secure against passive and active adversaries, including replay attacks, in the standard model.

Author Contributions: Writing—original draft, J.M. and M.B.; Writing—review and editing, A.H., J.M. and M.B. All authors have read and agreed to the published version of the manuscript

Funding: This work was supported in part by FEDER funds (Junta de Andalucía-University of Málaga) under Project UMA18- FEDERJA-172, and in part by NSF under Grant DUE 1241525.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ashton, K. That 'Internet of Things' Thing. *RFID J.* **2009**, *22*, 97–114.
2. 3GPP. Security Architecture and Procedures for 5G System, (3GPP), TS 33.501. Available online: https://www.3gpp.org/ftp/Specs/archive/33_series/33.501/ (accessed on 21 September 2020).
3. Arkko, J.; Lehtovirta V.; Eronen, P. RFC 5448:Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). RFC Editor. 2009. Available online: https://www.rfc-editor.org/info/rfc5448 (accessed on 17 November 2020).
4. 3GPP. 3G Security; Security Architecture, (3GPP), TS 33.102. Available online: https://www.3gpp.org/ftp/Specs/archive/33_series/33.102/ (accessed on 21 September 2020).
5. 3GPP. 3G Security; Specification of the MILENAGE Algorithm Set: An Example Algorithm Set for the 3GPP Authentication and Key Generation Functions f1, f1*, f2, f3, f4, f5 and f5*; Document 5: Summary and Results of Design and Evaluation, (3GPP), TR 35.909. Available online: https://www.3gpp.org/ftp/Specs/archive/35_series/35.909/ (accessed on 23 September 2020).
6. Koutsos, A. The 5G-AKA Authentication Protocol Privacy. *arXiv* **2018**, arXiv:1811.06922.
7. 3GPP. Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA), (3GPP), TS 33.220. Available online: https://www.3gpp.org/ftp/Specs/archive/33_series/33.220/ (accessed on 19 September 2020).
8. Krawczyk, H.; Bellare, M.; Canetti, R. RFC 2104: "HMAC: Keyed-Hashing for Message Authentication". Available online: https://tools.ietf.org/html/rfc2104 (accessed on 17 November 2020).
9. Shaik, A.; Borgaonkar, R.; Seifert, J.P.; Asokan, N.; Niemi, V. Practical Attacks Against Privacy and Availability in 4G/LTE. In Proceedings of the 23nd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, CA, USA, 21–24 February 2016.

10. 3GPP. Numbering, Addressing and Identification, (3GPP), TS 23.003. Available online: https://www.3gpp.org/ftp/Specs/archive/23_series/23.003/ (accessed on 21 September 2020).
11. Basin, D.; Radomirovic, S.; Dreier, J.; Sasse, R.; Hirschi, L.; Stettler, V. A formal analysis of 5g authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1383–1396. [CrossRef]
12. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208.
13. Khan, H.; Martin, K.M. A Survey of Subscription Privacy on the 5G Radio Interface—The Past, Present and Future. *IACR Cryptol. ePrint Arch.* **2020**, *2020*, 101.
14. Paret, D. *RFID and Contactless Smart Card Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
15. ISO/IEC 29192-1:2012. In *Information Technology—Security Techniques—Lightweight Cryptography—Part 1: General*; International Organization for Standardization: Geneva, Switzerland, 2012.
16. Feldhofer, M.; Wolkerstorfer, J.; Rijmen, V. AES implementation on a grain of sand. *IEEE Proc. Inf. Secur.* **2005**, *152*, 13–20.
17. Dao, V.L.; Hoang, V.P.; Nguyen, A.T.; Le, Q.M. A compact, low power AES core on 180 nm CMOS process. In Proceedings of the 2016 International Conference on IC Design and Technology (ICICDT), Ho Chi Minh, Vietnam, 27–29 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5.