Privacy Protected Subscriber Identification in 5G Network

Mohsin Khan and Valtteri Niemi

University of Helsinki, Department of Computer Science,
P.O. Box 68 (Gustaf Hällsträmin katu 2b)
FI-00014 University of Helsinki
Finland
{mohsin.khan,valtteri.niemi}@helsinki.fi

Abstract. The aspirations for the next generation mobile network (5G) are high. It has a vision of improved security and privacy over the existing LTE network. Subscription privacy of a user has been a historical concern with all the previous generation mobile networks, namely GSM, UMTS, and LTE. While a little improvement have been achieved in securing the privacy of long-term identity of a subscriber, the so called IMSI catchers are still in existence even in the LTE and advanced LTE networks. This report looks into this problem of concealing long-term identity of a subscriber and presents different techniques of using public-key cryptography to tackle it. One special case of public-key crytography is identity based crypto. A rigorous comparison among the pros and cons of the different techniques show that identity based cryptograpy is a potential solution for securing the long-term identity privacy of a user in the 5G network.

1 Introduction

NGMN Alliance has pointed out the privacy of a user as a requirement of the 5G network under the requirement category of enhanced services [1]. In 3GPP TR 33.899 [2], subscribers' privacy is captured as one of the high level security requirements of the 5G network. However, in the context of diversified devices and the complex business and service model of 5G, it is important to define who is a subscriber and what subscriber-privacy means. According to 3GPP TR 21.905 [3] a subscriber is an entity (associated with one or more users) that is engaged in a subscription with a service provider. A subscription describes the commercial relationship between the subscriber and the service provider, cf. 3GPP TR 21.905 [3]. A subscription identifier is the identifier that uniquely identifies a subscription in the 3GPP system. The identifier is used to access networks based on 3GPP specifications. Subscription Privacy refers to the right to the protection to any information that (a) can be used to identify a subscription to whom such information relates, or (b) is or might be directly or indirectly linked to a subscription. This definition of privacy suggests to protect any personally identifiable information (PII) from an active or passive attacker. While

it is important to classify the identifiers into PII and non-PII, the long-term identifier is surely a PII. In this report we will keep our discussion limited only to the case of identification of an UE using long-term identifier of the relevant subscriber. In the case of 2G (GSM), 3G (UMTS) and 4G (LTE) networks, this long-term identifier is known as international mobile subscriber identity (IMSI). Nevertheless, the same principles used in the solutions proposed in this report can be extended to conceal any PII.

One approach of protecting IMSI privacy is to use a temporary IMSI instead of the original IMSI and keep changing the temporary IMSI at a feasible frequency. Note that the temporary IMSI has to be assigned over a confidentiality protected channel and different entities of the network may assign different temporary IMSIs to the user equipment (UE). In the LTE network, the temporary IMSI assigned by serving network (SN) is called globally unique temporary identity (GUTI) and the home network (HN) does not assign any temporary IMSI to the UE. However, during the initial attachment of a UE to the SN, the UE has neither a GUTI nor a security context with the SN that can assign it with a GUTI. Besides, GUTI can be lost by either one or both of the UE and the SN. This forces the UE to reveal its IMSI to the SN to keep itself from permanently locked out of the network. This problem gives an opportunity to an active IMSI catcher (AICa) who impersonates a legitimate SN and forces the UE to run the initial attachment protocol. This also gives an opportunity to a passive IMSI catcher (PICa) to eavesdrop the IMSI sent in clear text. Solutions [5, 4] have been proposed by using temporary IMSI known as pseudonym assigned by the HN. While these solutions solve the cases of lost and unsynchronised GUTI, they still have the problem of lost or unsynchronised pseudonyms and also initial attachment. In this report we present how a security context can be set up in between the network (either with SN or HN) and the UE even before the identification of the UE so that the UE can use the security context to send its IMSI with confidentiality protection. Such a security context will mitigate the attack mounted by a PICa. Nevertheless, we show that an AICa would not be able to agree on a legitimate security context with the UE and consequently will not be able to reveal the IMSI.

In order to present a formal discussion we need to know what are the entities involved in this identification process, what are the communication interfaces among those entities and how much the entities can be trusted with the IMSI. As the architecture of 5G security is yet to be finalized, we present an abstraction of the involved entities and assume that whatever the security architecture of 5G eventually be, it will contain these entities and interfaces. This abstraction is directly extracted from LTE security architecture. Figure 1 shows the entities. It involves the UE, serving radio network (SRN), serving core network (SCN), HN. We have two more entities: PICa and AICa. The interface UE-NB in between UE and SRN is initially unprotected. Nevertheless, SRN-SCN and SCN-HN are always protected and the security of these interfaces is out of the scope of this report. The PICas eavesdrop on the UE-SRN interface when it is unprotected to extract an IMSI. The AICas impersonate a legitimate SN and run a legitimate
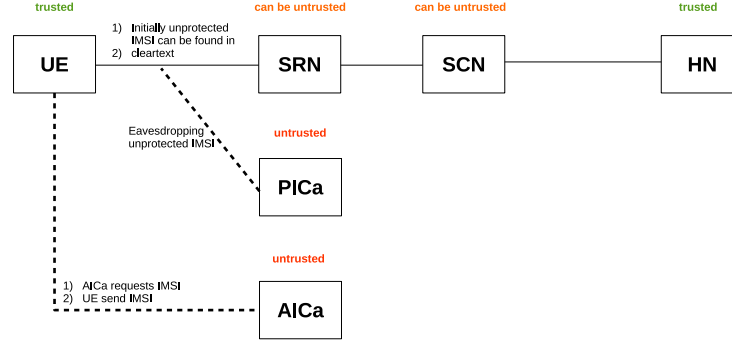
Fig. 1. High-level security architecture

protocol with the UE in order to reveal the IMSI. HN and UE both own the IMSI and they are trusted with it. Both of PICa and AICa are untrusted while it is technically possible to not trust SRN and SCN. However, by some other specification in 3GPP TS xx.xxx it is required to reveal IMSI to the SCN to enable lawful interception (LI) without involving HN.

We propose different solutions based on public-key cryptography which make the UE-SRN interface protected even during the initial attachment to fight against the PICa. Our solutions also stop the AICa from running a legitimate protocol successfully that could reveal the IMSI. However, there are other 5G requirements which our solutions should also meet. Such requirements are: reduced signalling overhead, improved control plane latency, concealing all the parts of IMSI (MCC, MNC and MSIN). To avoid the downgrade attack, the solutions need to be backward compatible with the legacy networks. Also, in the case of public-key, the complexity involved in setting a PKI and revocation of a publick-key need to be considered with high importance. Considering all these requirements, we evaluate our solutions based on the following criteria:

1. Concealed from PICa, AICa, SRN, SCN
2. Parts of the IMSI concealed
3. Signalling overhead
4. Latency

5. Backward compatibility
6. PKI complexity
7. Public-key revocation and re-provisioning

While the choice of the solution is dependent on how much want to achieve, hybrid solution using identity based public-key cryptography and pseudonyms appear to be a promising solution.

In Section 2 we present a quick intro to identity based cryptography (IBC). In Section 3 we present the solutions. In Section **??**, we present a rigorous solution based on the aforementioned evaluation criteria. Finally we conclude the paper in Section 4

2 IBC in the Jargon of Cryptography

Modern-day cryptography can be broadly categorized into two categories depending on how the keys are used to encrypt and decrypt the message. In symmetric key cryptography the sender and receiver share a secret key which is used for encryption and decryption both. In public-key cryptography the receiver has a pair of keys. One of the keys is public and the other is private. The public key is used by the sender to encrypt the message and the private key is used by the receiver to decrypt the message. While the challenge in symmetric key cryptography is to create a key known only by the sender and receiver but no one else, one major challenge in public-key cryptography is to authenticate the public key.

Based on the authentication mechanism, public-key cryptography can be categorized into three more categories:

1. Certificate based
2. Root-key based
3. Identity based, which is known as IBC

Figure 2 shows this categorization with advantages and disadvantages of the respective categories. In the certificate based case, the public key is signed by a trusted third party. In the root-key based case, no runtime authentication of the public key is required because a very limited number of public key is used in the system and all the senders are pre-provisioned with all the existing public keys. In the IBC case, the public key of a receiver is computed from the identity of the receiver and the public key of a trusted third party. However, the private key of the receiver is computed from the identity of the receiver and the private key of the trusted third party. This private key has to be securely provisioned to the receiver by the trusted third party. In this case, even though an extra one-time burden of private key provisioning is required, the sender does not need to authenticate the public key of a receiver, because if the public key is not authentic, the receiver will not have the private key and any message encrypted by the public key will never be decrypted. In other words, the authenticity of the public key in IBC is guranteed by the trusted third party. Usually in IBC, the
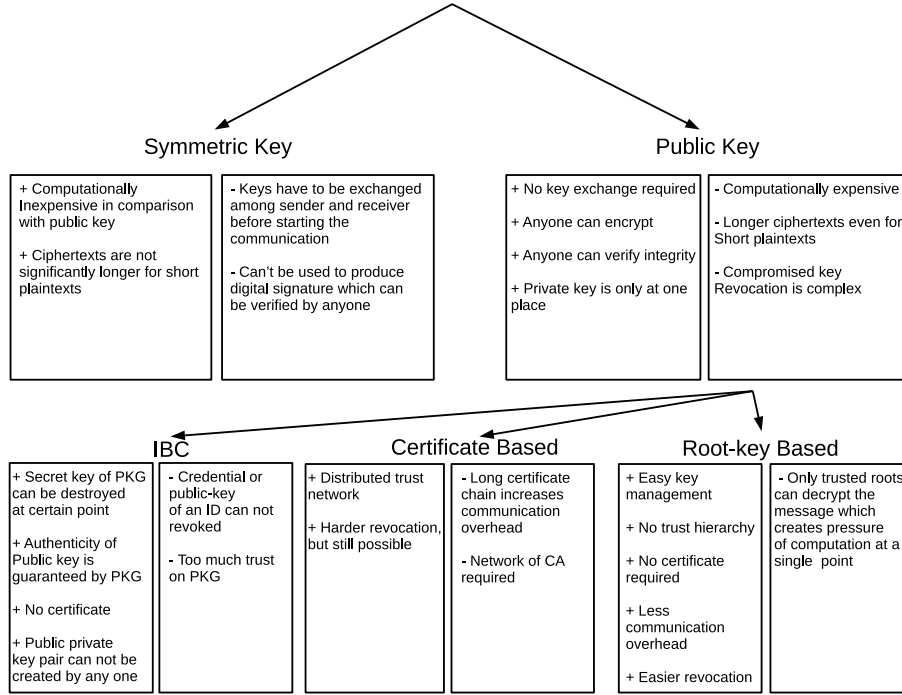
Fig. 2. IBC in the Jargon of Cryptography

trusted third party is known as the private key generator (PKG). While in the certificate based and root-key based case it is possible to revoke the public key of a receiver, it is impossible to revoke the public key in IBC unless the identity itself is revoked. In Figure 3 we show how IBC works pictorially.
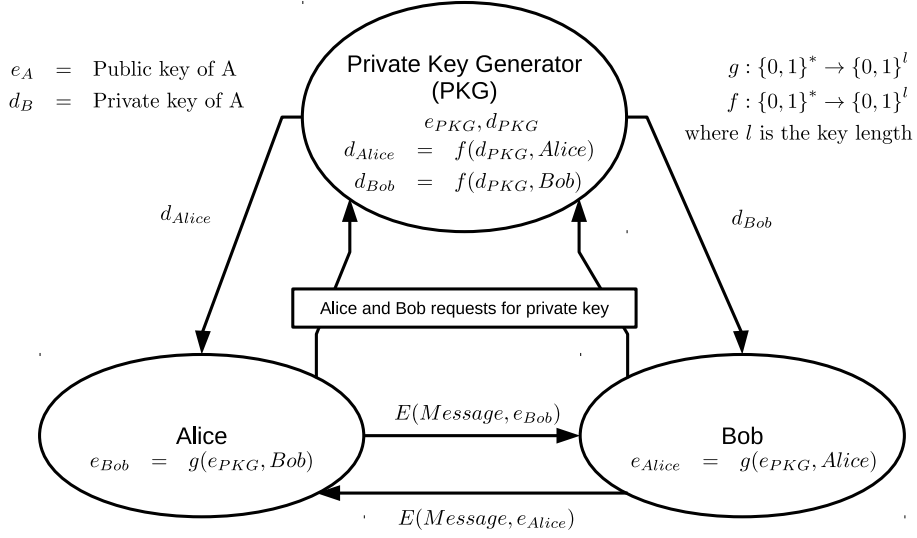
3 Solutions

It would be beneficial to mention some notation here before delving into the solutions.

1. $hnid = MCC||MNC$ identifies the HN
2. $scnid = MCC||MNC$ identifies the SCN
3. e_A is the public key of entity A
4. d_A is the private key of entity A
5. $\mathcal{X}_{A,B}$ is the certificate of the public key of A issued and signed by B
6. E, D are the encryption and decryption functions respectively

3.1 Certificate Based Public-key Cryptography

In certificate based public-key cryptography, certificates digitally signed by a trusted third party is used to authenticate the ownership of a public key. The

e_A = Public key of A

d_B = Private key of A

Private Key Generator (PKG)

e_{PKG}, d_{PKG}

d_{Alice} = $f(d_{PKG}, Alice)$

d_{Bob} = $f(d_{PKG}, Bob)$

$g : \{0,1\}^* \to \{0,1\}^l$

$f : \{0,1\}^* \to \{0,1\}^l$

where l is the key length

d_{Alice}

d_{Bob}

Alice and Bob requests for private key

$E(Message, e_{Bob})$

Alice

e_{Bob} = $g(e_{PKG}, Bob)$

Bob

e_{Alice} = $g(e_{PKG}, Alice)$

$E(Message, e_{Alice})$

Fig. 3. IBC mechanism

trusted third party who can sign the certificate is called a certicate authority (CA). It is possible to build a chain of trust in certificate based public-key cryptography by allowing an entity to become a CA who owns a public key signed by a CA. A certificate is valid if it is signed by a CA and the CA has a valid certificate. There has to exist at least one entity who has a self signed certificate. The entity having a self signed certificate is considered as the root CA. To use certificate based public-key cryptography to secure IMSI privacy, we need to figure out few things first: who is the root CA and who else can be a CA, who are the entities that own a public key, how a certificate can be revoked and how the root CA can change its public key.

We choose the HN of a subscriber is the root CA for the subscriber. And every UE is provisioned by the public key of its HN. A SCN owns public keys signed signed by HNs. The SRN broadcasts its identity. The UEs interested to attach with the network sends its *hnid* to the SRN and SRN relays it to SCN. SCN sends its public key certificate along with the signature signed by the corresponding HN to SRN. SRN relays it to the UE. The UE validate the certifiate and the signature. If it is a valid signature, the UE concatenates some random bits called *salt* at the end of the IMSI, encrypts the result with the public key and sends the ciphertext to the SRN. The SRN relays this to the SCN. SCN decrypts the ciphertext, discard the salt from the tail and extracts the IMSI. This completes the identification of a UE to SCN. Figure shows the protocol in detail. Note that, in the roaming agreement phase, the SCN does not need to get the certificate exactly from the HN but can get it from any CA who is trusted in by the HN in the chain of trust.
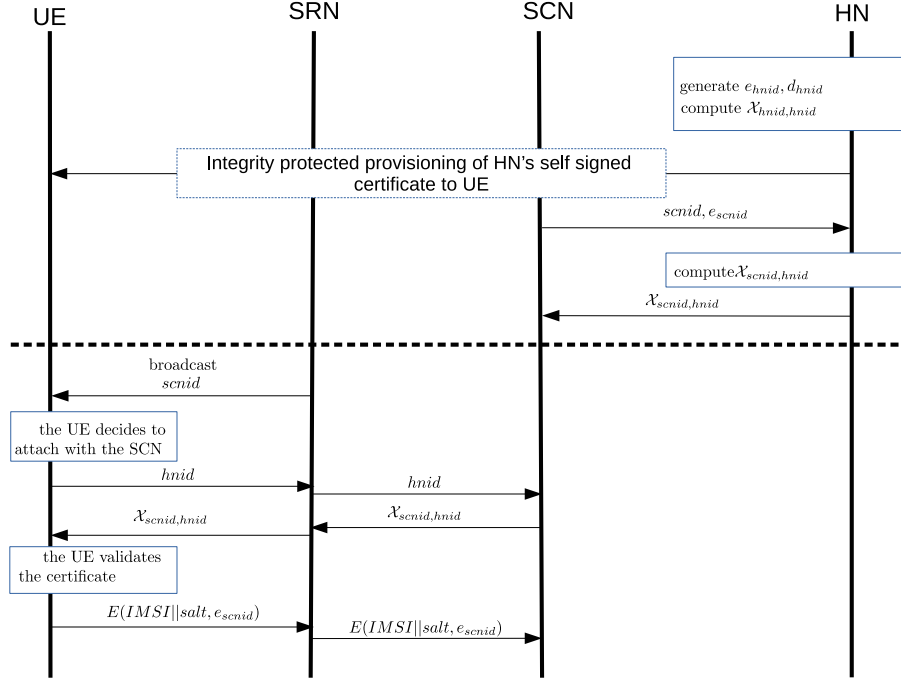
Fig. 4. Privacy protected UE identification using certificate based public-key cryptography

This approach can not conceal MCC and MNC at all. It coneals MSIN from AICa, PICa and SRN. The IMSI is revealed entirely at SNC. It requires a full round trip signalling in betwenn UE and SCN. It carries *hnid* from UE to SCN and in return carries the public-key certificate back from SCN to UE. Public keys are quite large and consequently it adds signficant signalling overhead. Also, public-key cryptography is compuntationally heavier than that of symmetric-key cryptography. Surely latency will increase significantly in this identification protocol. Nevertheless, once an UE is identified by the network, the network assings a GUTI to the UE. It is a question for further research to evaluate the effect of this extra signalling overhead. This solution is not backward compatible with LTE, UMTS and GSM. It doesn't require to esbalish any new trusted authority, so PKI complexity is very little. However, revocation of the public-key of an SCN and HN is not trivial and need to be investigated closely.

Every CA in the system maintains a list of revoked public keys which were certified and signed (at the first place) by it or by any other CA trusted by it. If there is a revocation, the CA who signed the certificate at the first place updates its list and the CA updates its CA with the information. The UE periodcally run a protocol with HN to download the latest revoked public keys. If a dishonest SCN presents a UE with the certificate which is already revoked but the UE

has not yet been able to downloaded it, then the MSIN will be revealed to the SCN. However, such a dishonest SCN will fail to run a successful authentication protocol. As soon as the UE come in touch with a legitimate SCN, it identifies itself sucessfully, runs a successful authentication protocol, and download the latest revocation list from the HN. Thereafter the dishonest HN will not be able to reveal the MSIN anymore. This implies that an AICa can mount an attack if it was once trusted by the HN as a legitimate SCN but is no more trusted by the HN and the UE is not yet updated with the withdrawal of the trust. The existence of such an AICa is apparently extremely rare. The time window this attack allows to track a subscriber is also very small. Which altogether make this attack very expensive. However, if the private key of an SCN is stolen, both PICa and AICa can mount attacks untill the public key is revoked.

When the private key of an HN is compromised, the HN update itself with a new public-private key pair. All the relevant certificates in the system get updated with the required change according to standard PKI procedure. However, The new public key certificate of the HN needs to be re-provisioned to all the UE that has a subscription with the HN. Because, if an SCN is updated with a new certificate, but the UE does not have the new root certificate, then the UE will never be able to validate the certificate presented by a legitimate SCN. One way to circumvent this problem is to store certificates concerned with all the root certificate used by the HN in every SCN . If an UE can not validate a certificate, the SCN sends the immediate older certificate. However, the UE will not validate a certificate associated with a root certificate which is older than the current one it has. Consequently, an SCN which is not updated with the latest root certificate of an HN will not be able to identify a UE that belongs to the HN. Once the UE is identified by the SCN and authentication becomes successful, the HN knows the location of the UE. At this point the HN can send the new certificate and signature, integrity protected by the master symmetric key shared by the HN and the USIM. Another way is to publish the new root certificate in the public media with integirity protection so that anyone can download the root certificate using some other internet connection. The need for reprovisioning the root certificate is an extremely rare event. In case the root private key is stolen by an attacker, it allows the attacker can act as both AICa or PICa. Once the need for existing root certificate revoking and provisioning of a new one is identified, it does not allow the attacker a long time window to track the UEs. Protecting the privacy of a private key and detecting the compromisation of a private key is a whole different security question and we do not delve into that in this report.
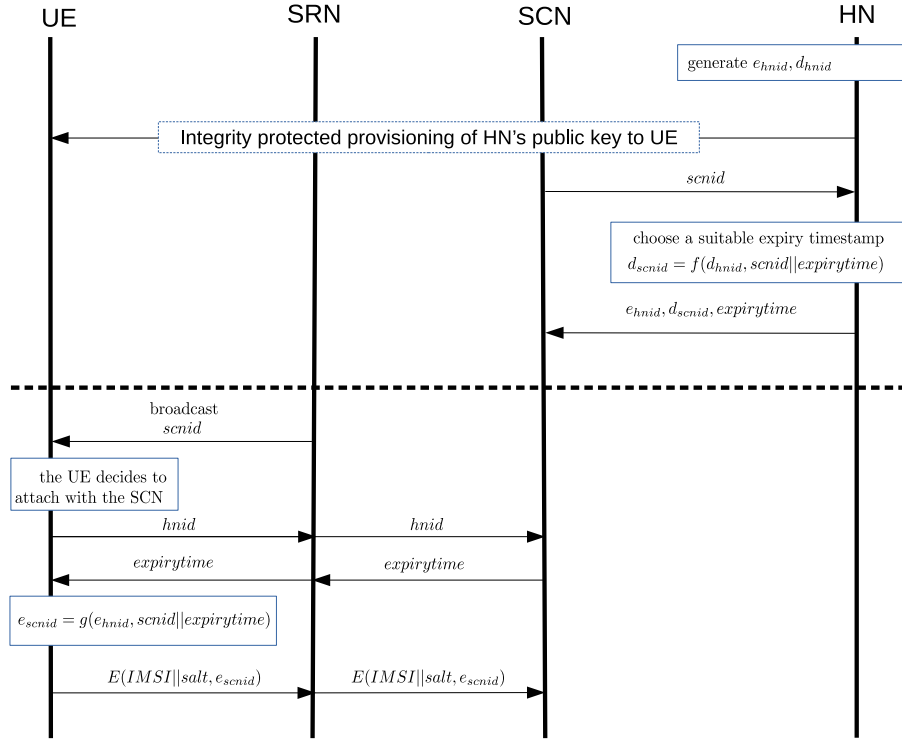
UE	SRN	SCN	HN

generate e_{hnid}, d_{hnid}

Integrity protected provisioning of HN's public key to UE

$scnid$

choose a suitable expiry timestamp
$d_{scnid} = f(d_{hnid}, scnid||expirytime)$

$e_{hnid}, d_{scnid}, expirytime$

broadcast
$scnid$

the UE decides to
attach with the SCN

$hnid$ $hnid$

$expirytime$ $expirytime$

$e_{scnid} = g(e_{hnid}, scnid||expirytime)$

$E(IMSI||salt, e_{scnid})$ $E(IMSI||salt, e_{scnid})$

Fig. 5. Privacy protected UE identification using IBC

3.2 Solution based on IBC

4 Conclusion

5 Acknowledgement

References

[1] NGMN 5G White Paper V1.0 [cited Jan, 2017]. Available at: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf
[2] 3GPP TR 33.899 V0.6.0 [cited Jan, 2017]. Available at: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3045
[3] 3GPP TR 21.905 [cited Jan, 2017]. Available at: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=558
[4] Karl Norrman, Mats Nslund, Elena Dubrova: Protecting IMSI and User Privacy in 5G Networks. 2nd International Workshop on 5G Security

[5] Philip Ginzboorg, Valtteri Niemi: Privacy of the long-term identities in cellular networks. Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications Pages 167-175