New Privacy Issues in Mobile Telephony: Fix and Verification

Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan
University of Birmingham
School of Computer Science
Birmingham, UK
m.d.arapinis, l.mancini, e.ritter, m.d.ryan@cs.bham.ac.uk

Nico Golde, Kevin Redon, Ravishankar Borgaonkar
Technische Universität Berlin and Deutsche Telekom Laboratories
Berlin, DE
nico, kredon, ravii@sec.t-labs.tu-berlin.de

ABSTRACT

Mobile telephony equipment is daily carried by billions of subscribers everywhere they go. Avoiding linkability of subscribers by third parties, and protecting the privacy of those subscribers is one of the goals of mobile telecommunication protocols. We use formal methods to model and analyse the security properties of 3G protocols. We expose two novel threats to the user privacy in 3G telephony systems, which make it possible to trace and identify mobile telephony subscribers, and we demonstrate the feasibility of a low cost implementation of these attacks. We propose fixes to these privacy issues, which also take into account and solve other privacy attacks known from the literature. We successfully prove that our privacy-friendly fixes satisfy the desired unlinkability and anonymity properties using the automatic verification tool `ProVerif`.

Categories and Subject Descriptors

D.2.4 [**Software Program Verification**]: Formal Methods

General Terms

Verification, Security

Keywords

Unlinkability, Anonymity, ProVerif, Mobile Telephony

1. INTRODUCTION

While most mobile phone users accept that the network operator can track their geographical movements, few would be happy if an arbitrary third party could do so. Such a possibility would enable all kinds of undesirable behaviour, ranging from criminal stalking and harassment to more mundane monitoring of spouse or employee movements, as well as profiling for commercial and advertisement purposes. For this reason, 3G (Third Generation) mobile phone protocols have been designed to prevent third parties, eavesdropping on the radio link, from identifying wireless messages as coming from a particular mobile phone. Therefore, mobile phones identify themselves, whenever possible, by means of temporary identifiers (TMSIs) instead of using their long term unique identities (IMSIs). Temporary identities are periodically updated by the network. To avoid linkability, the assignment of a new temporary identity is encrypted using a session key established through the 3G Authentication and Key Agreement (AKA) protocol.

When 3G protocols were first introduced in 1999, active attack scenarios were a remote possibility because of the high cost of the equipment required, the closedness of the hardware design and the lack of open source implementations of the protocol stack. This scenario has recently changed. Cheap base stations [19] can be produced by programming USRP (Universal Software Radio Peripheral) boards [21]. These lower the cost of producing radio devices thanks to software emulation of specialized functions once executed by expensive hardware. The increasing popularity of USRPs led for example to a cheap implementation of fake base station attacks on GSM (Global System for Mobile Communication) [31], which were considered sufficiently costly to prevent wide-scale attacks. Shorter range base stations, available at affordable prices, have been targeted as well by open source developers (e.g. openBSC project [33]), security researchers [23]. Certain old mobile phones based on the Ti Calypso GSM baseband chips, can be reprogrammed by flashing an open source version of the protocol stack (developed by the osmocom-BB project [34]). These new developments open at the same time the way for the exploration of new uses of mobile telephony technology [1, 30] and for the exploitation of its weaknesses [20, 23, 31, 29], making active attack scenarios an increasingly likely reality.

Hence, we believe active attackers should now be considered when analysing mobile systems in order to obtain convincing and reliable results on their security. From this perspective we present a formal analysis of the 3G subscribers privacy. We expose two novel threats and we demonstrate that these threats can lead to real implementations which make use of cheap equipment. Furthermore, we propose privacy friendly fixes to thwart the detected privacy issues and we formally verify that our fixes achieve the desired privacy goals.

Our Contributions. Linkability of transactions has been

CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA.

identified and often reported by the media as an important threat for user privacy [14, 17, 24] though it has been overlooked so far by most of the existing studies of mobile telecommunication protocols which instead focus on confidentiality and authentication requirements (Section 2.2). In this paper, we present the first formal analysis of 3G protocols *w.r.t.* privacy of mobile phone users from third party attackers and in particular *w.r.t.* unlinkability and anonymity of 3G subscribers. For our analysis we use automated formal methods. The use of formal methods allows us to: *(i)* precisely and unambiguously define the desired privacy properties in terms of third-party strong anonymity and strong unlinkability; *(ii)* identify new vulnerabilities with respect to subscriber privacy thanks to a rigorous specification of the protocols and of the analysed properties.

However, the currently available automated tools are still quite limited and cannot straightforwardly be used to verify unlinkability and anonymity properties [11]. Here we develop ways to model the protocols and the desired properties as biprocesses in order to use the `ProVerif` tool on our 3G case study. The automatic verification with the `ProVerif` tool allows us to: *(i)* verify strong unlinkability and strong anonymity. To the best of our knowledge, it is the first time these definitions of privacy properties have been successfully used for verification using an automated tool; *(ii)* verify that the fixes we propose do preserve the privacy of the mobile phone users from third parties in terms of unlinkability and anonymity; *(iii)* automatically verify privacy properties expressed as equivalence relations between systems consisting of an unbounded number of agents executing an unbounded number of sessions;*(iv)* obtain a higher level of confidence in the resulting proofs than the ones provided by more error-prone manual techniques. With our method `ProVerif` successfully detects the privacy vulnerabilities (described in Section 3) and also successfully proves that the fixed protocols (presented in Section 5) satisfy both unlinkability and anonymity (Section 6).

Moreover, we demonstrate how these vulnerabilities can lead to practical attacks, by implementing them in real 3G networks in Germany (Vodafone, O2, T-Mobile) and in France (SFR) (Section 4).

2. BACKGROUND AND RELATED WORK

Third Generation telecommunications systems (3G) is a mobile telephony standard specified and maintained by the Third Generation Partnership Project (3GPP). It was introduced in 1999, with the birth of UMTS, to offer better support for mobile data applications, increased data rates and to lower costs of mobile data communications. Furthermore, 3G offers an improved security architecture with respect to previous mobile telecommunication systems such as GSM (Global System for Mobile Communication).

2.1 3G Security Requirements

3G aims to provide authentication, confidentiality of data and voice communication, as well as user privacy [6]. In particular, 3G privacy goals include the following [6]:

User identity confidentiality: the property that the permanent user identity (IMSI) of a user to whom a service is delivered cannot be eavesdropped on the radio access link;

User untraceability: the property that an intruder cannot deduce whether different services are delivered to the same user by eavesdropping on the radio access link.

In order to achieve these two privacy-related properties, 3G (and GSM) relies on the use of temporary identities TMSIs (Temporary Mobile Subscriber Identities) for identifying and paging mobile phones (more precisely mobile stations, MSs) instead of using their long-term identities IMSIs (International Mobile Subscriber Identities). Indeed, the eavesdropping of the IMSI in plaintext communications would allow the identification of mobile telephony users by third parties. Moreover, the 3G standard requires periodic updates of the temporary identity, to avoid the traceability of a mobile station by third parties. New temporary identities are periodically assigned by the network through the TMSI reallocation procedure. The newly assigned TMSI is encrypted using a session key which is established by executing the 3G Authentication and Key Agreement protocol (AKA). The 3G AKA protocol allows MS and network to achieve mutual authentication and establish a pair of shared session keys, namely a ciphering key and an integrity key. These keys are used to ensure the secrecy and integrity of the subsequent communications.

2.2 Related Work

Known 3G Vulnerabilities. Three categories of attacks on mobile telephony systems have been described in the past.

IMSI Catcher. The identification procedure, consisting in the request of the user identity by the network followed by a cleartext reply containing the user identity, is acknowledged in the 3G standard as a breach of the user identity confidentiality [6, p. 19, s. 6.2]. This procedure is exploited by the well-known "IMSI catcher" attack, which is the best known attack to mobile telephony users' privacy. It consists in forcing a mobile phone to reveal its identity (IMSI) [22, 32] by triggering the identification procedure from a fake operator base station (configured with the corresponding mobile network and country code settings). Until fairly recently, implementing an IMSI catcher required specialised software and equipment such as base stations. However, such devices have become more and more affordable thanks to software emulation [31]. To the best of our knowledge the only implementation of a 3G IMSI catcher is the one presented in [23] and is realised using a modified femtocell.

3G/GSM-interoperability. Previously proposed attacks on 3G security exploit the vulnerabilities which are propagated from GSM to 3G when providing interoperability between the two systems. Most of the reported attacks of this kind take advantage of well-known weaknesses of the GSM authentication and key agreement protocol, such as the lack of mutual authentication and the use of weak encryption. These attacks allow an active attacker to violate the user identity confidentiality, to eavesdrop on outbound communications [28] and to masquerade as a legitimate subscriber obtaining services which will be billed on the victim's account [10]. However, these attacks cannot be carried out on pure 3G networks, which are the scope of our analysis, because they rely on the lack of mutual authentication in GSM and on the possibility of downgrading the communication from 3G to GSM.

3G specific. To the best of our knowledge, the only attack that does not rely on GSM/3G interoperability has been presented by Zhang and Fang in [36]. This attack is a variant of the false base station attack and takes advantage of the fact that the mobile station does not authenticate the serving network. It allows the redirection of the victim's

outgoing traffic to a different network, for example a network which uses a weaker encryption algorithm or one which charges higher rates than the victim's one. Zhang and Fang's attack concerns impersonation, service theft and data confidentiality, while our work exhibits privacy issues arising in 3G.

Our work is based on the formal analysis of pure 3G protocols. It relies on the study and modelling of the 3G standard and does not make assumptions about interoperability between GSM and 3G. We focus on subscriber privacy and discover further breaches other than the ones caused by the identification procedure and the propagation of GSM weaknesses to 3G.

Previous Formal Analysis The 3G AKA protocol in its pure form (i.e. with no GSM support) has been formally proved to meet some of the specified security requirements [3], such as authentication and confidentiality of data and voice communication. However, privacy related properties such as unlinkability and anonymity, which are the focus of our work, are not analysed in [3]. The framework applied in [3] cannot be used to specify unlinkability and anonymity properties, let alone reason about them. The formal framework used in our paper allows us to precisely define and verify privacy related properties. Hence, we can discover privacy attacks on the modelled protocols and propose solutions which are formally proved to satisfy the desired privacy properties.

Other Work on 3G Privacy Enhancement A new framework for authentication has been proposed to provide subscriber privacy with respect to the network [26]. In particular, the authors aim to achieve MSs anonymity with respect to the serving network, and location privacy with respect to the home network. To achieve this purpose, they propose a new mechanism for the location update and a three way handshake protocol, to be used for authentication instead of the currently used 3G AKA protocol. However, unlike our work, this work is not supported by a formal model of the AKA protocol, nor does it provide a formal verification of the properties of the proposed protocols. Moreover, their attacker model considers the network as not fully trusted, while we are only concerned about third party attackers controlling the radio link communications.

3. NOVEL PRIVACY THREATS

In this section we describe two breaches of privacy, which expose a subscriber's identity and allow an attacker capable of sending and receiving messages on the air to identify the presence of a target mobile phone (MS) in a monitored area, or even track its movements across a set of monitored areas. As we will see, the attacker does not need to know any keys, nor perform any cryptographic operation. This kind of vulnerabilities usually look trivial once uncovered but often remain unnoticed for long time, since they do not involve fancy cryptography but are caused by errors in the protocol logic.

As argued in Section 1 and as witnessed by the attacks implementation presented in Section 4, a convincing analysis of 3G privacy and security should consider active attackers instead of passive ones. For this reason, we assume that the attacker has unlimited access to the radio link between the mobile station and the base station. He can sniff, inject, replay, and modify messages. This attacker model is the same considered in most of the previous work on GSM/3G security [28, 10, 36].

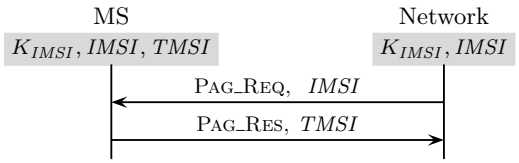

Figure 1: 3G IMSI Paging Procedure

In the rest of this paper, we consider a simplified network architecture. This architecture involves simply the mobile stations and the network. The network models both the Base Station (BS) which directly communicates with the MS on the radio link, and the complex structure of databases and servers connected with it and forming the 3G control network. Hence, we abstract away from any communication within the network and model only communication between mobile stations and the network. This abstraction allows us to hide details which are uninteresting for the purposes of our analysis and keep the models used for verification small, but at the same time precisely specify the interactions on the air between MS and network, which are the subject of our analysis.

3.1 IMSI Paging Attack

The paging procedure is used to locate a mobile station in order to deliver a service to it, for example an incoming call. Paging request messages are sent by the network in all the location areas most recently visited by the mobile station in order to locate it and deliver a service to it. The paging request message is sent on a Common Control Channel (CCCH) and contains the identity of one or more mobile stations. The paging procedure is typically run using the TMSI to identify a MS. However, the IMSI can be used when the TMSI is not known by the network. A mobile station receiving a paging request establishes a dedicated channel to allow the delivery of the service and sends a paging response containing the most recently assigned TMSI (see Figure 1).

The possibility of triggering a paging request for a specific IMSI allows an attacker to check a specific area for the presence of mobile stations of whom he knows the identity, and to correlate their IMSI and TMSI. As we will detail in Section 4.2, in a real setting, the link between the paged IMSI and the related TMSI would need to be confirmed by replaying the attack several times.

3.2 AKA Protocol Linkability Attack

The Authentication and Key Agreement (AKA) protocol achieves mutual authentication between a MS and the network, and establishes shared session keys to be used to secure the subsequent communications. The MS with identity *IMSI* and the network share a secret long-term key, K_{IMSI}, assigned to the subscriber by the mobile operator and stored in the USIM. The secret key allows the MS and the network to compute shared ciphering and integrity session keys to be used for encryption and integrity check of communications.

The 3G AKA protocol [6], shown in Figure 2, consists in the exchange of two messages: the authentication request and the authentication response. Before sending an authen-

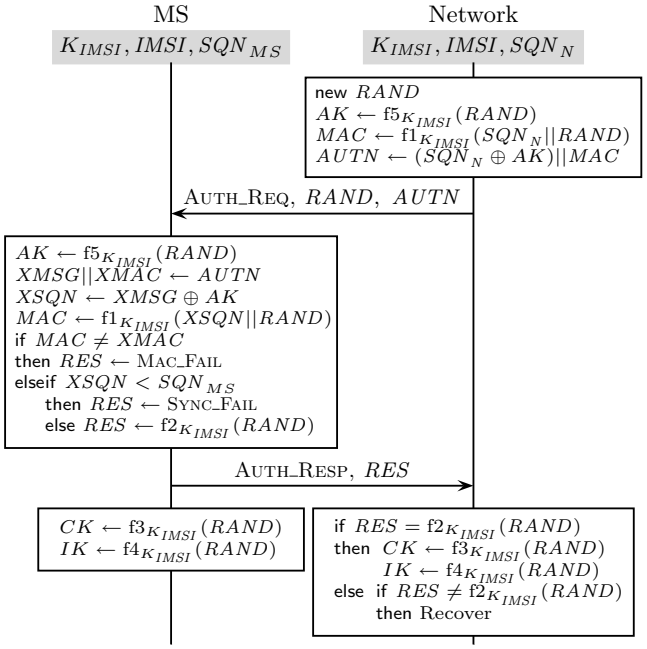

Figure 2: 3G Authentication and Key Agreement

tication request to the mobile station, the network computes the authentication data: a fresh random challenge $RAND$, the authentication token $AUTN$, the expected authentication response $\mathrm{f2}_K(RAND)$, the integrity key IK, and the encryption key CK (see Figure 2). The functions f1, f2, f3, f4 and f5, used to compute the authentication parameters, are keyed cryptographic functions computed using the shared key K_{IMSI} [8]. The authentication function f1 is used to calculate the message authentication code MAC; f2 is used to produce the authentication response parameter RES; the key generation functions, f3, f4 and f5 are used to generate the ciphering key CK, the integrity key IK and the anonymity key AK, respectively.

The network always initiates the protocol by sending the authentication challenge $RAND$ and the authentication token $AUTN$ to the mobile station. $AUTN$ contains a MAC of the concatenation of the random number with a sequence number SQN_N generated by the network using an individual counter for each subscriber. A new sequence number is generated either by increment of the counter or through time based algorithms as defined in [6]. The sequence number SQN_N allows the mobile station to verify the freshness of the authentication request to defend against replay attacks (see Figure 2).

The MS receives the authentication request, retrieves the sequence number SQN_N and then verifies the MAC (condition $MAC = XMAC$ in Figure 2). This step ensures that the MAC was generated by the network using the shared key K_{IMSI}, and thus that the authentication request was intended for the mobile station with identity $IMSI$. The mobile station stores the greatest sequence number used for authentication, so far SQN_{MS}. This value is used to check the freshness of the authentication request (condition $XSQN < SQN_{MS}$ in Figure 2).

The mobile station computes the ciphering key CK, the integrity key IK and the authentication response RES and sends this response to the network.The network authenticates the mobile station by verifying whether the received response is equal to the expected one ($RES = \mathrm{f2}_K(RAND)$). The authentication procedure can fail on the MS side either because the MAC verification failed, or because the received sequence number $XSQN$, is not in the correct range with respect to the sequence number SQN_{MS} stored in the mobile station. In the former case, the mobile station sends an authentication failure message indicating MAC failure (Mac_Fail) as the failure cause. In the latter case, the authentication failure message indicates synchronisation failure (Sync_Fail) as the failure cause. When a MAC failure occurs the network may initiate the identification procedure. When a synchronisation failure occurs the network performs re-synchronisation.

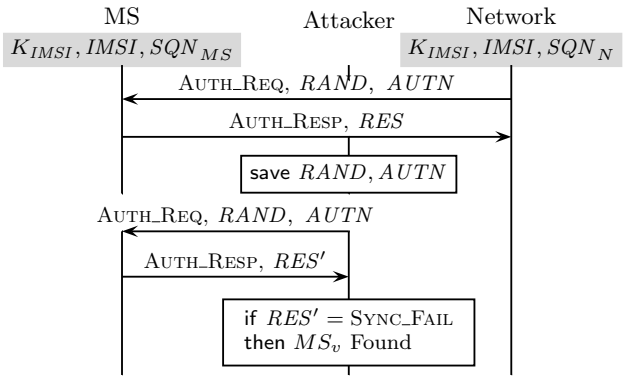

Figure 3: AKA Protocol Linkability Attack

To detect the presence of a victim mobile station MS_v, in one of his monitored areas, an active attacker just needs to have previously intercepted one legitimate authentication request message containing the pair $(RAND, AUTN)$ sent by the network to MS_v. The captured authentication request can now be replayed by the adversary each time he wants to check the presence of MS_v in a particular area. In fact, thanks to the error messages, the adversary can distinguish any mobile station from the one the authentication request was originally sent to. On reception of the replayed authentication challenge and authentication token $(RAND, AUTN)$, the victim mobile station MS_v successfully verifies the MAC and sends a synchronisation failure message. However, the MAC verification fails when executed by any other mobile station, and as a result a MAC failure message is sent. The implementation of few false BS would then allow an attacker to trace the movements of a victim mobile station, resulting in a breach of the subscriber's untraceability. The proposed attack is shown in Figure 3.

3.3 Formal Verification

While the paging procedure is obviously a breach of users' privacy, the traceability attack on the AKA protocol is much more subtle. Indeed, the messages exchanged through this procedure contain neither the IMSI nor the TMSI of the MS. So one could think that the AKA protocol provides untraceability by construction. But we just saw that this is not the case. Only careful analysis *w.r.t.* precisely defined privacy requirements could reveal this flaw.

We run the `ProVerif` tool on the IMSI paging procedure

and on the AKA protocol [1]. Details of how unlinkability and anonymity were defined using the `ProVerif` calculus are given in Section 6. `ProVerif` fails to prove the unlinkability and anonymity of the IMSI paging procedure and exhibits actual attack traces. In the case of the AKA protocol, the anonymity property is proved to hold, while the unlinkability property verification fails. Although, the trace provided by `ProVerif` is a false attack, it does give a hint of the real attack by highlighting the test of the MAC received from the network as the source of the problem. The adoption of formal verification tools during protocol design could have thus revealed design flaws.

4. THE ATTACKS IN PRACTICE

In order to test the attacks presented in Section 3 in a deployed telecommunication network, we use a commercially available femtocell. Although, the particular femtocell hardware is tied to the network operator SFR, the proposed attacks are not. Indeed, we tested the attacks using mobile phones registered to different operators, hence just using SFR as serving network. The authentication token $AUTN$ is still provided by the victim's Home network. So by testing our attacks on T-Mobile, O2, SFR, and Vodafone victim MSs, we establish that all these tested networks are vulnerable to the attacks described above. However, we want to stress here that our implementation has the only purpose of showing the feasibility of our attacks and confirm that real cellular networks follow the 3GPP standard specifications and thus are vulnerable to the proposed attacks. The same attacks could be mounted by appropriately programming a USRP [21], which is a hardware device able to emit and receive radio signals. In this case, one could obtain wider range attack devices in order to monitor larger areas.

4.1 Femtocell architecture

A femtocell is a device that acts as a small base station to enhance 3G coverage and connectivity, especially inside buildings with otherwise bad coverage. Its coverage radius ranges from 10 to 50 meters. It connects mobile phones to the network of the corresponding MNO (Mobile Network Operator) using an existing wired Internet connection provided by the femtocell user, not the operator. 3G femtocells, also called Home Node B (HNB) support most of the functionalities provided by a typical 3G base station (Node B), *e.g.* physical layer (radio signalling) functions. In addition, the HNB establishes an authenticated secure tunnel over the Internet with the network of the operator. Using this encrypted connection, the femtocell forwards all radio signalling and user-generated traffic to the GANC (GAN Controller), which is connected to the core network of the operator (refer to [7] for more details of the femtocell architecture).

The communication between the femtocell and the GANC is based on the Generic Access Network (GAN) protocol. The GAN protocol, was originally designed to allow mobile communication over Wi-Fi access points. The protocol was standardised by MNOs in 2004 [25] and led to the GAN specification [4, 5] in 2005. This specification has been adopted and extended to be used in femtocell environments [35] . The femtocell uses this protocol to forward communication from a mobile station via the GANC to the network or vice versa. The MS does not need any special GAN support, it just connects to the femtocell in the same way as it connects to a standard base station. The femtocell maps all Layer-3 radio signalling to TCP/IP based GAN messages and passes them to the GANC. Thus, it transparently encapsulates all traffic generated by the phone and the network.

[1]The `ProVerif` code is available online [2]

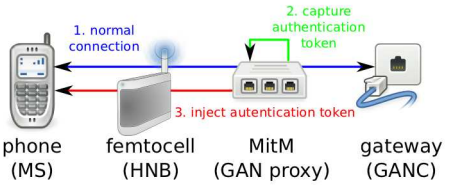

Figure 4: Experimental Attack Setup

4.2 Attack Procedure

For the purpose of implementing our attacks (Section 3), we use a compromised femtocell like the one described in [23]. More specifically, we reproduce the hacking performed in [23] to gain root access of our femtocell and redirect the traffic to a Man in the Middle (MitM) GAN proxy, positioned between the femtocell and the GANC. We use this MitM GAN proxy as entry point for message injection. In particular, using the MitM GAN proxy we can inject messages into the connection between the MNO and the femtocell. The femtocell forwards these messages to the mobile phone, making them appear as if legitimately delivered by the MNO. To perform the attacks, we intercept, modify and inject 3G Layer-3 messages into the communication from the base station to the mobile phone in both directions, GANC-to-femtocell and femtocell-to-GANC. We redirect all the traffic between the femtocell and the GANC to our GAN proxy. The GAN traffic is cleartext travelling over an IP Sec tunnel for which we own the key material, thanks to the initial rooting/hacking of the femtocell. Additionally, we developed a set of applications which allow us to intercept, manipulate or insert selected messages, and distinguish different types of GAN messages. This allows us, for example, to cache subscribers information used to perform the attacks. In particular, we store the random challenge $RAND$, the authentication token $AUTN$, the TMSI and the IMSI of our victim MS. This information is directly extracted from the traffic that is passed through the MitM GAN proxy.

IMSI-Paging Procedure Attack To perform the IMSI paging attack, our software crafts a paging message encoding the necessary paging headers and parameters and a mobile station identity, *i.e.* one of the previously stored victim IMSIs. The crafted paging request is then sent by the GAN proxy to the femtocell. When the victim mobile phone receives the IMSI paging request, it readily answers with a paging response containing the victim's TMSI. Thus, by injecting a paging request, we can check whether a phone belonging to a designated victim is in the area covered by our device. In case of success, the phone generates the paging response, while a failed attempt generates no message. In general, it is possible that more than one phone replies to a paging request during the same time slot. However, one can repeat this procedure multiple times and correlate the timing and TMSI usage from the multiple replies as in [20].

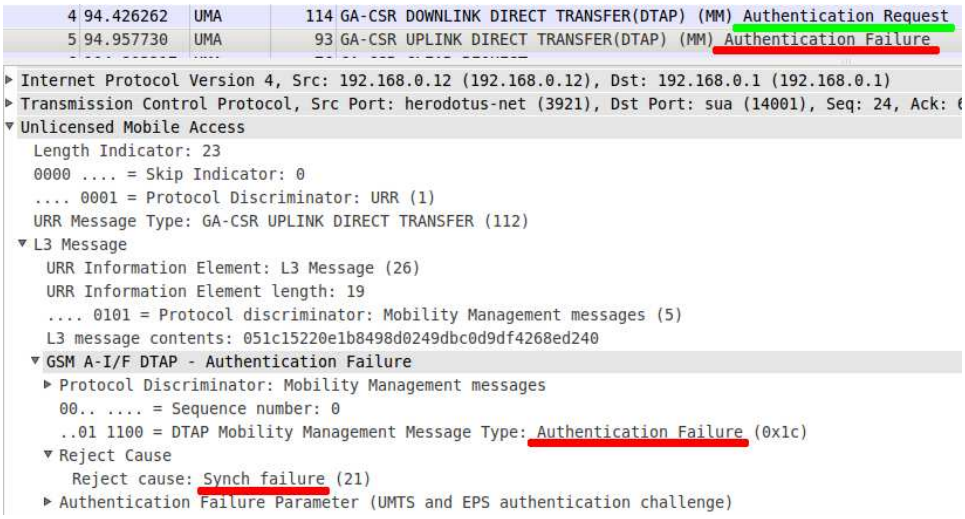

Figure 5: Successful Linkability-Attack

AKA Protocol attack To perform the AKA attack we replay a given authentication message for a specific target for which the GAN proxy cached the legitimate authentication data, *i.e.* $RAND$, $AUTN$. This data is sent unencrypted on the radio link and could be captured with any equipment capable of sniffing the radio link. As soon as a dedicated channel is allocated to the MS, *e.g.* after being paged or when initiating a phone call, our software crafts an authentication request AUTH_REQ using the previously cached $RAND$ and $AUTN$, *i.e.* replays a previous request. This request is encapsulated into a GAN message and sent to the femtocell. The femtocell takes care of delivering the authentication request message on the dedicated channel assigned to the MS, as illustrated in Figure 4. The phone performs a validation of the authentication request and answers with the authentication response. If the response to the replayed authentication is a Synchronisation Failure (Figure 5), then the MS on this dedicated channel is the victim's phone, and the victim is indeed in the femtocell area. Otherwise, the attacker needs to inject the same message to the other mobile stations in his area in order to find out if the victim MS is present or not.

The 3G AKA protocol is performed at each new session in the femtocell setting, this makes the caching of the authentication parameters very easy. Though, we do not have the tools to test if this applies when connecting to a typical Node B, we tested the 3G/GSM interoperability scenario by using the Osmocom-BB software and we observed that in this setting the execution of the AKA protocol can be triggered by calling for example the victim mobile phone a given number of times (by hanging up within a short time window this activity can be made non detectable by the victim [20]). For instance, our experiments showed that the execution of the AKA protocol on the UK Vodafone network can be triggered by calling six times the victim mobile phone, and hanging up before it even rings.

To illustrate the use of our attacks, consider an employer interested in tracking one of his employee's accesses to a building. He would first use the femtocell to sniff a valid authentication request. This could happen in a different area than the monitored one. Then the employer would position the device near the entrance of the building. Movements inside the building could be tracked as well by placing additional devices to cover different areas of the building. Similarly, these attacks could be used to collect large amount of data on users' movements in defined areas for profiling purposes, as an example of how mobile systems have already been exploited in this direction is available in [1] If devices with wider area coverage than a femtocell are used, the adversary should use triangulation to obtain finer position data.

5. PRIVACY PRESERVING FIXES

Despite the use of temporary identities to avoid linkability and to ensure anonymity of 3G subscribers, active attackers can rely on the paging procedure to break both anonymity and unlinkability. Moreover, the AKA protocol provides a way to trace 3G subscribers without the need to identify them in any way. As described in the previous section, these two attacks on privacy can be implemented using cheap devices which are widely available. This shows that the analysed procedures are a real threat for the users' privacy, and countermeasures should be promptly taken to provide an effectively privacy friendly mobile telephony system.

In this section we propose a set of countermeasures involving symmetric and public key-based cryptography. The public key infrastructure we propose is lightweight and easy to deploy because we only require one public/private key pair per mobile network operator, and none for the mobile stations. More generally, the solutions we present require only small changes to the current security architecture and to the cryptographic functions currently used in 3G. Hence we believe our solutions may be implemented in a cost-effective way, and thus could realistically be adopted by the telecommunication operators.

In addition to the solutions proposed to fix the IMSI paging and the AKA protocol, in this section we give a privacy friendly version of the identification procedure to fix the IMSI catcher attack. Indeed, the problem of privacy is a multilayer/multiprotocol problem [13] which requires all protocols at all layers to satisfy the desired properties. Even though, the analysis from the user privacy point of view of the entire set of 3G protocols cannot be tackled in a single

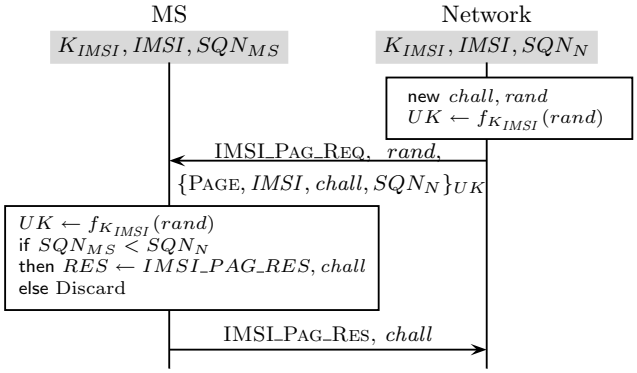

Figure 6: Paging Procedure Fix. The paging request is encrypted with the unlinkability key UK.

paper, we cannot ignore the best known privacy issue of mobile telecommunication systems. For this reason, we include a fixed version of the identification procedure in our privacy friendly solutions.

5.1 Lightweight Public Key Infrastructure

We propose the adoption of a lightweight public key infrastructure (PKI) providing each MNO with a private/public key pair. The public key of a network provider can be stored in the USIM. This public key makes it possible for a mobile station to encrypt privacy related information such as the IMSI, and deliver them to the network in a confidential manner. We do not require a public/private key pair to be assigned to the mobile stations. The adoption of such a lightweight PKI can also solve the problem exposed by Zhang and Fang in [36] concerning the lack of serving network authentication in the current infrastructure.

5.2 Protecting the IMSI Paging Procedure

To protect the paging procedure, we propose to encrypt the paging request using a shared session key UK, which we call unlinkability key. This key is generated by applying a new one-way keyed function f to the long-term key K_{IMSI}, and a random number *rand* contained in the paging request. This key should be used for privacy preserving purposes only. Furthermore, we require the encrypted request message to include a random challenge *chall* and a sequence number SQN. The network stores the random challenge and checks it against the one sent by the MS in the paging response (Figure 6). The aim of the SQN is to ensure freshness of the paging request and avoid replay attacks. The SQN should be handled in the same way as in the AKA protocol. A MS receiving a legitimate IMSI paging request should discard it if the SQN is not in the correct range. The use of this procedure should still be kept minimal (preferring the paging with TMSI whenever possible) to avoid burdening the signalling communication with cryptographic operations. In fact, each MS has to decrypt and check all the received IMSI paging to determine if it is the recipient.

5.3 Fixing the AKA Protocol

The AKA protocol is a threat for the unlinkability of 3G subscribers because the error messages sent in case of authentication failure leak information about the identity of

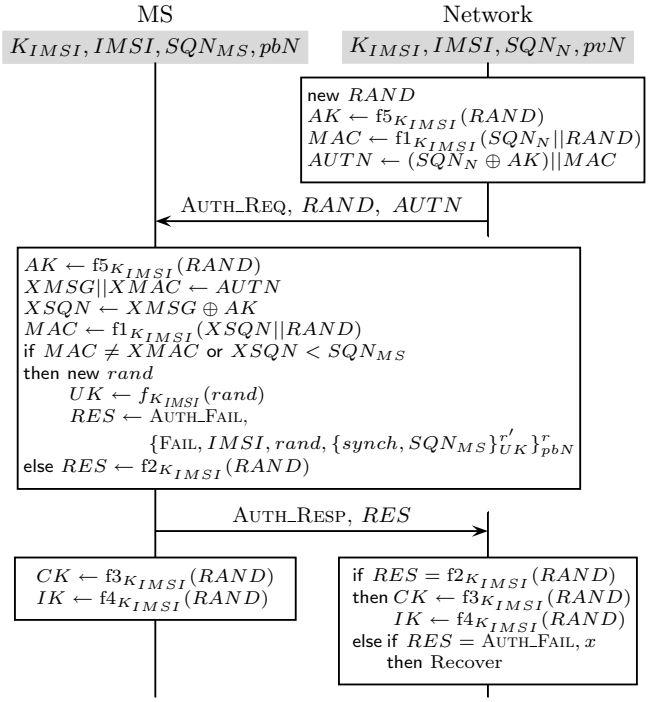

Figure 7: The fixed AKA protocol. The error messages are encrypted using the network public key.

the subscriber. To avoid this information leakage, the error messages sent in case of any type of failure should look indistinguishable from an attacker's point of view.

Moreover, the 3G standard stipulates [6] different procedures to recover from each of the two kinds of failure, but this is a source of additional information flow that can be used to launch our privacy attack. In the solution we propose we solve this problem since error recovery can be performed within the network without the need to trigger further procedures on the air. Indeed, all the parameters needed for error recovery are sent in the error message allowing the recovery procedure to be carried within the network.

The fixed version of the AKA protocol (Figure 7) carries on as specified by the standard, the network sends $RAND$, $AUTN$ and waits for a response. The response is $RES = \text{f2}_{K_{IMSI}}(RAND)$, as in the standard, in case the checks of MAC and sequence number are successful. If either of these checks fails, an error message is sent to the network. The failure message is now encrypted with the public key of the network pbN, and contains a constant FAIL, the IMSI, and the current sequence number SQN_{MS} of the MS. The IMSI sent encrypted in the error message allows the network to check the identity of the MS without triggering the identification procedure. The current sequence number of the mobile station enables the network to perform resynchronisation with the Authentication Centre (AuC, the server storing subscribers authentication data) of the operator of the mobile station, if needed. SQN_{MS} is sent encrypted with the unlinkability key (as defined in the fixed paging procedure) in order to authenticate the error message to the Network as coming from the MS with permanent identity *IMSI*. The Network can deduce the cause of the failure from the *IMSI* and SQN_{MS} contained in the error message. Upon receipt of

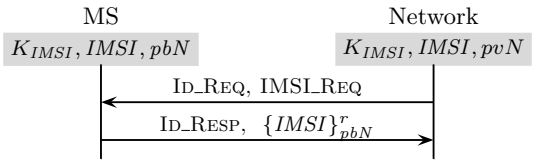

Figure 8: Identification Procedure Fix. The identity response is encrypted with the public key of the network. The r denotes randomised encryption.

this authentication failure message the action performed for error recovery purposes should be the same regardless of the type of failure occurred. Indeed any difference in behaviour would be a source of additional information flows.

5.4 Protecting the Identification Procedure

The identification procedure exposes the IMSI of a MS (the IMSI is sent in cleartext upon request by the network). Hence, it breaches both anonymity and unlinkability. According to the standard, the use of the identification procedure should be limited as much as possible, to avoid a passive attacker overhearing the IMSI. However, the cost of devices allowing active attacks is constantly decreasing. As a consequence, enhancing the protocol to protect the IMSI is vital to ensure privacy.

The fixed version of the identification procedure (Figure 8) involves two messages: the first is sent by the network to ask for the IMSI, the second, the identity response, is the randomised encryption of the IMSI of the mobile station using the public counterpart (pbN) of the private key of the network operator (pvN).

5.5 Discussion of the Proposed Fixes

While the fix we propose for the identification procedure is intuitive and straightforward, this is not the case for the other two procedures. In particular, we take care of maintaining the style of mobile telecommunication protocols and at the same time ensuring privacy. We introduce the unlinkability key, a new session key generated for privacy purposes, instead of using the long term key K_{IMSI} (as in the 3G AKA), and make use of the sequence number SQN for freshness purposes (this is needed to avoid user linkability caused by replay attacks); We maintain the authentication flow of the AKA and modify only the way error messages are dealt with by including error recovery information inside the error message (this avoids the triggering by the network of diversified procedures in order to perform error recovery).

Our proposed fixes use public-key cryptography; intuitively, there is no way to avoid that, since if a mobile station's TMSI is unknown to the serving network (hence the need to perform the identification procedure) then there is no shared key by which they can communicate privately. The additional costs associated with deploying and using public-key cryptography are in fact small for the two following reasons.

Firstly, only mobile telephony *operators* are required to have a public/private key pair. Neither subscribers, nor mobile phone equipments nor USIMs need to have their own public/private key pair. The operator's public key could be stored in the USIM of the mobile station, as it is already the case for the IMSI and the long-term key K_{IMSI}. The Home Network can act as a certifying authority for the public key of the different Serving Networks (see below). Thus, the public key infrastructure is similar to that used on the web, where corporations (not users) have certified keys.

Secondly, the computationally expensive public-key encryption and decryption are required only for the identification protocol and when the AKA-protocol fails. The execution of the identification and the IMSI paging procedures should anyway be kept minimal according to the currently deployed standard. Moreover, failures during the execution of the AKA-protocol rarely occur according to our experiments. Hence, the computational overhead of the public-key cryptography is not significant. Moreover, it is possible to delegate the encryption and decryption to the mobile equipment, instead of executing them on the USIM. This would not weaken the security properties of the 3G procedure, since the mobile equipment in the current architecture has already access to the IMSI, while the network public key is publicly available information.

For roaming purposes, each Home Network (HN) can act as certifying authority of the Serving Network (SN) for its own subscribers. The public key $pbHN$ of the HN could be stored in the USIM. At registration time with a SN, the MS would declare its HN, and the SN would provide the MS with its public key $pbSN$, together with a certificate from the mobile station's HN ($\mathsf{sign}_{skHN}(pbSN)$). Hence, a mobile station would only need to obtain a certified version of the SN's public key, and verify it using its own network provider public key. This would provide, in a efficient way, the MS with the necessary public keys to execute our fixed versions of the protocols.

The introduction of cryptographic operations on the mobile equipment side could be a source of Denial of Service (DoS) attacks aiming to consume the battery load of victim mobile phones. To mitigate the effect of such attacks, the mobile phone's software could rate limit the phone's willingness to respond to authentication, IMSI paging and identity request messages, so to guarantee a minimum battery lifetime even in case of attempted DoS attacks. We have calculated that responding to such requests on average once per minute would consume an additional one tenth of battery life.

6. VERIFICATION

Many deployed protocols have subsequently been found to be flawed [27, 18, 12, 16]. In this perspective and in order to increase the confidence one can have in the solutions proposed at the previous section, we formally analyse our proposed fixes *w.r.t.* privacy. We present the results of the automatic verification of the privacy-friendly enhancement discussed in Section 5. Table 1 summarises these results which apply for the protocols running both in parallel and in isolation. We use the `ProVerif` tool [15] to verify the unlinkability and anonymity properties of our fixes for the 3G procedures exposing the IMSI (identification and paging) and the 3G AKA protocol. We use the formalisation of privacy-related properties as given by Arapinis et al. in [11], namely strong unlinkability and strong anonymity.

Note that for verification purposes we use randomised symmetric encryption to conceal the sequence number SQN instead of using the exclusive-or. Indeed, even if the theory allows to write a set of reduction rules to model the `xor` function, the `ProVerif` tool cannot deal with its algebraic

properties. The use of randomised encryption anyway would achieve stronger properties with respect to the secrecy of the sequence number, we hence recommend the adoption of this modification in the standard protocol.

6.1 ProVerif Calculus

We use the `ProVerif` calculus, which is similar to the applied pi-calculus [9], to precisely model the privacy enhancing solutions proposed in Section 5. It makes it possible to automatically verify protocol models written in the language, using the `ProVerif` tool [15]. We introduce the `ProVerif` calculus aiming to give a flavour of the verification process. The description of the calculus that we give here is not comprehensive (refer to [15] for a detailed presentation).

Cryptographic primitives are modelled as functions and messages are represented by *terms* built over an infinite set of names $a, b, c, \ldots$, an infinite set of variables $x, y, z, \ldots$ and a finite set of function symbols $f_1 \ldots, f_n$. Function symbols represent cryptographic primitives that can be applied to messages. The effect of applying function symbols to terms is described by a set of reduction rules.

Example 1. Using functions and reduction rules we can define cryptographic functions, for example, let Σ=`{senc/3, pub/1, aenc/3, f/2, f1/2, f2/2, f3/2, f4/2, f5/2}`, and consider the reductions: `reduc sdec`$(k, \texttt{senc}(k, m, r)) = m$ and `reduc adec`$(k, \texttt{aenc}(\texttt{pub}(k), m, r)) = m$. Where, `senc` and `aenc` model, respectively, randomised symmetric and asymmetric encryption and model the property that the plaintext, m, can be retrieved from the cyphertext given the knowledge of the key k.

The syntax of `ProVerif` calculus *processes* is given by the following grammar:

$P, Q, R ::=$	plain processes
0	null process
$P \mid Q$	parallel composition
$!P$	replication
`new` $n; P$	name restriction
`if` $M = N$ `then` P `else` Q	conditional
`let` $M = D$ `in` P `else` Q	destructor application
`in`$(M, x); P$	message input
`out`$(M, N); P$	message output

We give here only the informal semantics of the calculus. The null process does nothing. $P \mid Q$ represents the parallel execution of P and Q. The replication $!P$ of a process P acts like the parallel execution of an unbounded number of copies of P. The name restriction `new` $n; P$ creates a new name n whose scope is restricted to the process P and then runs P. The message input `in`$(M, x); P$ represents a process ready to input from the channel M. The message output `out`$(M, N); P$ describes a process that sends a term N on the channel M and then behaves like P. The let construct tries to rewrite D and matches the result with M; if this succeeds, then the variables in M are instantiated accordingly and P is executed; otherwise Q is executed. The conditional checks the equality of two terms M and N and then behaves as P or Q accordingly. We will omit the else branch of a let or a conditional when the process Q is 0.

Example 2. Multiple mobile stations MS, with identity $imsi$, and long-term private key sk running along with the serving network, SN, can be modelled by the process:

$$\texttt{S} = \texttt{new}\ pvN;\ \texttt{let}\ pbN = \texttt{pub}(pvN)\ \texttt{in}\ \texttt{out}(c, pbN);\ \texttt{!new}\ sk; \texttt{new}\ imsi; \texttt{!new}\ sqn; \texttt{(SN | MS)}.$$

The privacy related properties we verify are expressed in terms of *observational equivalence*. Intuitively, two processes P and Q are observationally equivalent denoted by $P \approx Q$, if any interaction of P with the adversary, can be matched with an interaction of Q (and vice versa, *i.e.* all interactions of Q can be matched by P) and the same input/output behaviour is observed.

The `ProVerif` tool can prove diff-equivalence of biprocesses, which implies observational equivalence. Biprocesses are pairs of processes which differ by some choice of terms, this choice is written `choice`$[M, M']$. For example, to test if the processes `out(c, a)` and `out(c,b)` are equivalent, one would check the following biprocess using `ProVerif`: `out`$(\texttt{c}, \texttt{choice}[\texttt{a}, \texttt{b}])$.

6.2 Strong Unlinkability

Strong unlinkability is defined in [11] as follows. Let $P = \texttt{new}\ \tilde{n}.(!R_1 \mid \cdots \mid !R_p)$ be a p-party protocol where $\forall i \in \{1, \ldots, p\},\ R_i = \texttt{new}\ id.\texttt{new}\ \tilde{m}.init_i.!(\texttt{new}\ s.main_i)$. For all $i \in \{1, \ldots, p\}$, we build the protocol P^{Ri} as follows:

$$\begin{aligned} P^{Ri} &= \texttt{new}\ \tilde{n}.(!R_1 \mid \cdots \mid !R_{i-1} \mid !R_i'' \mid !R_{i+1} \mid \cdots \mid !Rp) \\ R_i'' &= \texttt{new}\ id.\texttt{new}\ \tilde{m}.init_i.\texttt{new}\ s.main_i. \end{aligned}$$

P is said to preserve strong unlinkability of R_i if $P \approx P^{Ri}$. Informally, this means that the adversary cannot distinguish a situation where the role R_i was executed many times from one in which it was executed at most once, i.e. he cannot link two executions of the role R_i. Going back to our mobile phone scenario, the strong unlinkability property holds when the situation where mobile stations access services multiple times looks the same as the ideal situation where each mobile station accesses the services at most once, *i.e.* where by construction unlinkability holds. Formally, we want the process `S`, defined in Example 2, to be observationally equivalent to the system $\texttt{S}_{UNLINK}$ defined as follows:

$$\begin{aligned} \texttt{S}_{UNLINK} = \ & \texttt{new}\ pvN;\ \texttt{let}\ pbN = \texttt{pub}(pvN)\ \texttt{in} \\ & \texttt{out}(c, pbN); \\ & \texttt{!new}\ sk; \texttt{new}\ imsi; \texttt{new}\ sqn; \texttt{(SN | MS)}. \end{aligned}$$

The absence of the replication before the `new sqn` construct means that in $\texttt{S}_{UNLINK}$ each MS executes the protocol at most once. The above mentioned observational equivalence can be verified with `ProVerif`, defining `S` and $\texttt{S}_{UNLINK}$ as the following biprocess $\texttt{PV}_{UNLINK}$, where $sk1$, $sk2$ are long term keys and $imsi1, imsi2$ are long term identities:

$$\begin{aligned} \texttt{PV}_{UNLINK} = \ & \texttt{new}\ pvN;\ \texttt{let}\ pbN = \texttt{pub}(pvN)\ \texttt{in} \\ & \texttt{out}(c, pbN); \\ & \texttt{!new}\ sk1; \texttt{new}\ imsi1; \\ & \texttt{!new}\ sk2; \texttt{new}\ imsi2; \texttt{new}\ sqn; \\ & \texttt{let}\ (sk, imsi) = \texttt{choice}[(sk1, imsi1), (sk2, imsi2)] \\ & \texttt{in (SN | MS)}. \end{aligned}$$

We have that the left side of the choice represents a system where a mobile station (with identity $imsi1$ and key $sk1$) may execute the protocol many times, while the right side represents a system where mobile stations execute the

protocol at most once (the identity $imsi2$ and the key $sk2$ are always different and can be used at most once for the execution of the protocol). Hence, we reduce the problem of testing strong unlinkability to the diff-equivalence of a biprocess. ProVerif proves that the strong unlinkability property is satisfied by our models of the fixes identification, paging and AKA protocols as described in Section 5.

6.3 Strong Anonymity

Strong Anonymity is defined in [11] as follows. Let $P = \texttt{new}\ \tilde{n}.(!R_1 \mid \cdots \mid !R_p)$ be a p-party protocol where $\forall\ i \in \{1,\ldots,p\},\ R_i = \texttt{new}\ id.\texttt{new}\ \tilde{m}.init_i.!(\texttt{new}\ s.main_i)$. For all $i \in \{1,\ldots,p\}$, we build the protocol P^{Ri} as follows:

$$\begin{array}{rcl} P^{Ri} & = & \texttt{new}\ \tilde{n}.(!R_1 \mid \cdots !R_p \mid R_V) \\ R_V & = & \texttt{new}\ \tilde{m}.init_i\{^{id_V}/_{id}\}.!(\texttt{new}\ s.main_i\{^{id_V}/_{id}\}). \end{array}$$

Where the identity id_V of the agent playing the role R_V is a public name not occurring in P. P is said to preserve strong unlinkability of R_i if $P \approx P^{Ri}$. Informally, this means that the adversary cannot distinguish a situation where the role R_V with known identity id_V was executed from one in which it was not executed at all, i.e. he cannot breach the anonymity of the agent with role R_V. Going back to our mobile phone scenario, strong anonymity requires a system in which a mobile station $\texttt{MS}_V$ with publicly known identity $IMSI_V$ executes the protocol to be indistinguishable from a system in which the $\texttt{MS}_V$ is not present at all. Such a system obviously preserves $IMSI_V$'s anonymity. Formally, we want the system S, defined as in Example 2 to be observationally equivalent to the system $\texttt{S}_V$ defined as follows:

$$\begin{array}{rcl} S_V & = & \texttt{new}\ pvN;\ \texttt{let}\ pbN = \texttt{pub}(pvN)\ \texttt{in} \\ & & \texttt{out}(c, pbN); \\ & & \quad !\texttt{new}\ sk; \texttt{new}\ imsi; (!\texttt{new}\ sqn; \texttt{(SN | MS)}) \\ & & \texttt{|}\ \texttt{new}\ sk; !\texttt{new}\ sqn;\ \texttt{(SN | MS}_V\texttt{)}. \end{array}$$

In the system $\texttt{S}_V$ the mobile station $\texttt{MS}_V$ with publicly known identity $imsi_V$ can run the protocol. The mentioned observational equivalence can be translated in the following ProVerif biprocess $\texttt{PV}_{ANON}$, where $imsi_V, imsi_{ms}$ are permanent mobile station identities:

$$\begin{array}{rcl} \multicolumn{3}{l}{\texttt{free}\ \ imsi_V.} \\ \texttt{PV}_{ANON} & = & \texttt{new}\ pvN;\ \texttt{let}\ pbN = \texttt{pub}(pvN)\ \texttt{in} \\ & & \texttt{out}(c, pbN); \\ & & \quad (!\texttt{new}\ sk;\ \texttt{new}\ imsi; \\ & & \quad (!\texttt{new}\ sqn;\ \texttt{(MS | SN)})) \\ & & \texttt{|}\ (\texttt{new}\ sk;\ \texttt{new}\ imsi_{ms}; \\ & & \quad \texttt{let}\ imsi\texttt{=choice[}imsi_V, imsi_{ms}\texttt{]}\ \texttt{in} \\ & & \quad !\texttt{new}\ sqn;\ \texttt{(SN | MS)}). \end{array}$$

The left side of the choice represents a system where the mobile station with public identity $imsi_V$ can run the protocol. Our fixes of the identification procedure, paging procedure and AKA protocol as described in Section 5 are proved by ProVerif to satisfy anonymity.

We took particular care in avoiding false attacks that could be reported by the tool due to its abstractions. Indeed, we formally define privacy properties through observational equivalence, however, ProVerif adopts a stronger equivalence relation called diff-equivalence. In particular, diff-equivalence can distinguish between the execution of different branches of a conditional statement even in the following case: $\texttt{if}\ a = a\ \texttt{then}\ P\ \texttt{else}\ P \not\approx_{diff} \texttt{if}\ a = b\ \texttt{then}\ P\ \texttt{else}\ P$

Properties	Identification	Paging	AKA
Unlinkability	√	√	√
Anonymity	√	√	√

NA Not Applicable √ Proved to hold × Attack found

Table 1: ProVerif Results on the Fixed Procedures

Properties	Paging	AKA
Unlinkability	×	×
Anonymity	×	√

NA Not Applicable √ Proved to hold × Attack found

Table 2: ProVerif Results on the current 3G Procedures

and hence, although the above processes are observationally equivalent (P is executed regardless the result of the if statement evaluation), they do not satisfy diff-equivalence. We are dealing with this issue in our code for the verification at lines 4-5, 36, 73-74, 81, and 86-87 of the code in the Appendix. As expected, the verification with the ProVerif tool fails to prove the anonymity of the 3G IMSI paging procedure and the unlinkability of both 3G IMSI paging and AKA protocols (see Table 2) and finds counterexamples showing that the two systems are distinguishable by the adversary. The modelling of unlinkability and anonymity into diff-equivalences we showed in this Section can in general be adopted for protocols which do not require an initialization phase preceding the main protocol procedure. Hence, our method is not specific for the analysed protocols, and shows how to automatically verify unlinkability and anonymity on a wide class of protocols[2].

7. CONCLUSION

The widely-deployed 3GPP 3G protocols aim to prevent unauthorised parties (such as private organisations and individuals) from tracking the physical location of users by monitoring the signals from their mobile phone. Specifically, the protocols use temporary identifiers and cryptography to achieve this aim.

We have shown that the protocols are vulnerable to new privacy threats and that these threats lead to attacks that can be mounted in practice at low cost. Currently, our demonstration relies on particular hardware/software using closed source implementation of the 3G protocol stack and radio signalling functions. We tested several networks of major operators (T-Mobile, O2, SFR, and Vodafone) and demonstrated that these are vulnerable to our attacks.

We used formal methods to show that the exposed privacy vulnerabilities could have been detected at design time, We developed and verified lightweight solutions to avoid the privacy vulnerabilities. The solutions we propose show that privacy friendly measures could be adopted by the next generation of mobile telephony standards while keeping low the computational and economical cost of implementing them.

Acknowledgement We are very grateful to Steve Babbage (Vodafone) for insightful comments, and are thankful to EPSRC for supporting this work through the projects *Verifying Interoperability Requirements in Pervasive Systems* (EP/F033540/1) and *Analysing Security and Privacy Properties* (EP/H005501/1).

[2]The ProVerif code is available online [2]

8. REFERENCES

[1] http://www.pathintelligence.com. Path Intelligence Ltd. (2010) FootPath.

[2] http://www.markryan.eu/research/UMTS/.

[3] 3GPP. Technical specification group services and system aspects; 3G security; formal analysis of the 3G authentication protocol (release 4). Technical Report TR 33.902, V4.0.0, 3rd Generation Partnership Project, 2001.

[4] 3GPP. Generic Access Network (GAN); Mobile GAN interface layer 3 specification. Technical Specification TS 44.318 v9.2.0, 3rd Generation Partnership Project, 2010.

[5] 3GPP. Generic Access Network (GAN); Stage 2. Technical Specification TS 43.318 v9.0.0, 3rd Generation Partnership Project, 2010.

[6] 3GPP. Technical specification group services and system aspects; 3G security; security architecture (release 9). Technical Report TS 33.102 V9.3.0, 3rd Generation Partnership Project, 2010.

[7] 3GPP. Security of Home Node B (HNB) / Home evolved Node B (HeNB). Technical Specification TS 33.302 v11.2.0, 3rd Generation Partnership Project, 2011.

[8] 3GPP. Technical specification group services and system aspects; 3G security; cryptographic algorithm requirements (release 10). Technical Report TS 33.105 V10.0.0, 3rd Generation Partnership Project, 2011.

[9] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL, 2001.

[10] Z. Ahmadian, S. Salimi, and A. Salahi. New attacks on UMTS network access. In *Conference on Wireless Telecommunications Symposium*, WTS'09, 2009.

[11] M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *IEEE Computer Security Foundations Symposium*, CSF, 2010.

[12] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, and M. L. Tobarra. Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps. In *ACM Workshop on Formal Methods in Security Engineering*, FMSE, 2008.

[13] G. Avoine and P. Oechslin. RFID Traceability: A Multilayer Problem. In *Financial Cryptography*, FC, 2005.

[14] M. Barbaro and T. Zeller Jr. A face is exposed for AOL searcher no. 4417749. The New York Times, August 9, 2006.

[15] B. Blanchet. Proverif: Cryptographic protocol verifier in the formal model. http://www.proverif.ens.fr/.

[16] M. Bortolozzo, M. Centenaro, R. Focardi, and G. Steel. Attacking and fixing PKCS#11 security tokens. In *ACM Conference on Computer and Communications Security*, CCS, 2010.

[17] C. Caldwell. A pass on privacy? The New York Times, July 17, 2005.

[18] I. Cervesato, A. D. Jaggard, A. Scedrov, J.-K. Tsay, and C. Walstad. Breaking and fixing public-key kerberos. *Inf. Comput.*, 206:402–424, February 2008.

[19] D. Burgess et al. OpenBTS. http://openbts.sourceforge.net/.

[20] N. H. Denis Foo Kune, John Koelndorfer and Y. Kim. Location leaks over the gsm air interface. In *Annual Network & Distributed System Security Symposium*, NDSS, 2012.

[21] Ettus. USRP. http://www.ettus.com/products, 2009.

[22] D. Fox. IMSI-Catcher. *Datenschutz und Datensicherheit (DuD)*, 21:539–539, 1997.

[23] N. Golde, K. Redon, and R. Borgaonkar. Weaponizing femtocells: The effect of rogue devices on mobile telecommunications. In *Annual Network & Distributed System Security Symposium*, NDSS, 2012.

[24] D. Goodin. Defects in e-passports allow real-time tracking. The Register, 26th January 2010.

[25] Kineto Wireless Inc. official Unlicensed Mobile Access presentation webiste. http://www.smart-wi-fi.com/, June 2010.

[26] G. Koien and V. Oleshchuk. Location privacy for cellular systems; analysis and solution. In *Privacy Enhancing Technologies Symposium*, volume 3856, 2006.

[27] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using fdr. In *Tools and Algorithms for the Construction and Analysis of Systems*, TACAS, 1996.

[28] U. Meyer and S. Wetzel. A man-in-the-middle attack on UMTS. In *ACM Workshop on Wireless Security*, WiSe, 2004.

[29] K. Nohl and S. Munaut. Wideband gsm sniffing. http://events.ccc.de/congress/2010/Fahrplan/attachments/1783_101228.27C3.GSM-Sniffing.Nohl_Munaut.pdf.

[30] openBSC Project. GSM Network at 28C3. http://events.ccc.de/congress/2011/wiki/GSM#GSM_Network_at_28C3, December 2011.

[31] C. Paget. Practical cellphone spying. Def Con 18 Hacking Conference, 2010.

[32] D. Strobel. IMSI Catcher, 2007. Seminar Work, Ruhr-Universitat Bochum.

[33] H. Welte, H. Freyther, D. Spaar, S. Schmidt, D. Willmann, J. Luebbe, T. Seiler, and A. Eversberg. OpenBSC. http://openbsc.osmocom.org.

[34] H. Welte, S. Munaut, A. Eversberg, and other contributors. OsmocomBB. http://bb.osmocom.org.

[35] J. Zhang and G. de la Roche. *Femtocells: Technologies and Deployment*. John Wiley & Sons, Ltd, 2009.

[36] M. Zhang and Y. Fang. Security analysis and enhancements of 3GPP authentication and key agreement protocol. *IEEE Transactions on Wireless Communications*, 4(2):734–742, 2005.

9. APPENDIX

Authentication, Secrecy, Integrity. The main purpose of the AKA protocol is to provide mutual authentication and establish session keys to be used for integrity protection and secrecy. Hence, our analysis would not be complete without ensuring that our privacy preserving version of the 3G AKA protocol still achieves the goals it was originally designed for. We verify mutual authentication and integrity properties as injective correspondence properties. We prove using `ProVerif` that the original properties of the AKA protocol are preserved by our fixes; the verification results are shown in Table 3.

Properties	Identification	Paging	AKA
Secrecy			
IMSI	√	√	√
K_{IMSI}	NA	√	√
CK, IK	NA	NA	√
confidential information	NA	NA	√
Authentication	NA	NA	√
Integrity	NA	NA	√

NA Not Applicable √ Proved to hold × Attack found

Table 3: Results of the Automatic Verification of the Fixed Procedures

ProVerif code We report the most relevant parts of the `ProVerif` scripts used for the verification of the fixed protocols. We omit the declaration of constants, any name which is not under the scope of a new statement as public name and hence as part of the adversary knowledge. Note that the identity of the victim mobile for the anonymity property is public.

Fixed IMSI paging procedure in ProVerif.

```
let PAGING_MS = in(c, x);
 let (msgtype, xrand, xblob) = x in (
 if msgtype = pagingReq then (
    let (xpage, ximsi, =sqn, xchall) =
     sdec(f(k, xrand), xblob) in (
     if xpage = page then (
        if imsi = ximsi then (
           out(c, (pagingResp, xchall))))))).
let PAGING_SN = new rand; new chall;
  new r_sn1; new r_sn2;
  let UK = f(k, rand) in (
  out(c, (pagingReq, rand, senc(UK, r_sn2,
                 (page, imsi, sqn, chall))));
  in(c, pres)).
```

Biprocess for unlinkability of IMSI paging.

```
process new pvN; let pbN = pub(pvN) in (
  out(c, pbN);
  (! (new sk1; new imsi1; new otmsi1;
  (! (new sk2; new imsi2; new otmsi2; new sqn;
      let imsi = choice[imsi1, imsi2] in (
      let k = choice[sk1, sk2] in (
      let otmsi = choice[otmsi1, otmsi2] in (
      (PAGING_MS) | (PAGING_SN)))))))))
```

Biprocess for anonymity of IMSI paging.

```
process new pvN; let pbN = pub(pvN) in (
  out(c, pbN);
  ((! (new k; new imsi; new otmsi;
             (! ((PAGING_MS) | (PAGING_SN)))))
| (new k; new id; new otmsi;
   let imsi = choice[id, imsi_V] in
   (! ((PAGING_MS) | (PAGING_SN)))))
```

Fixed AKA procedure in ProVerif.

```
let AKA_MS = new r_ms; in(c, x);
  let (xrand, xautn) = x in (
  let (msg, xmac) = xautn in (
  let ak = f5(k, xrand) in (
  let xsqn = sdec(ak, msg) in (
  let mac = f1(k, (xrand, xsqn)) in (
  if (xmac, xsqn) = (mac, osqn) then (
      let res = f2(k, xrand) in (
      let ck = f3(k, xrand) in (
      let ik = f4(k, xrand) in (
      out(c, res);
      in(c, xmsg)))))
  else (out(c, aenc(pbN, r_ms,
            (Fail, imsi, osqn)))))))).
let AKA_SN =
  new rand; new r_sn; new s; new r;
  let mac = f1(k, (rand, osqn)) in (
  let res = f2(k, rand) in (
  let ck = f3(k, rand) in (
  let ik = f4(k, rand) in (
  let ak = f5(k, rand) in (
  let autn = (senc(ak, r_sn, osqn), mac) in (
  let av = (rand, res, ck, ik, ak) in (
  out(c, (rand, autn));
  in(c, xres);
  if xres = res then (
     out(c, senc(ck, r, s)))
  else (out(c, reject))))))))).
```

Biprocess for unlinkability of AKA.

```
process new pvN; let pbN = pub(pvN) in (
  out(c, pbN);
  (! (new sk1; new imsi1;new otmsi1;
  (! (new sk2; new imsi2; new osqn; new otmsi2;
      let imsi = choice[imsi1, imsi2] in (
      let k = choice[sk1, sk2] in (
      let otmsi = choice[otmsi1,otmsi2] in (
      (AKA_MS) | (AKA_SN)))))))))
```

Biprocess for anonymity of AKA.

```
process new pvN; let pbN = pub(pvN) in (
  out(c, pbN);
  ((! (new k; new imsi; new otmsi;
       (!new osqn;  ((AKA_MS) | (AKA_SN)))))
 | (new k; new id; new otmsi;
    let imsi = choice[id, imsi_V] in (
    !new osqn;  ((AKA_MS) | (AKA_SN))))))
```

Original AKA procedure in ProVerif. We check the MAC and the sequence number (line 81) in the same conditional statement, so to avoid false attacks due to the evaluation of the conditional. For the same reason we introduce the functions `err` and `geterr` (lines 73-74) to determine the error message (lines 86-87) and avoid the use of an if statement.

```
reduc geterr(err(x,z,y,y))=macFail;
      geterr(err(x,x,y,z))=synchFail.
let AKA_MS = new r_ms; in(c, x);
  let (xrand, xautn) = x in (
  let (msg, xmac) = xautn in (
  let ak = f5(k, xrand) in (
  let xsqn = sdec(ak, msg) in (
  let mac = f1(k, (xrand, xsqn)) in (
  if (xmac, xsqn) = (mac,sqn) then (
     let res = f2(k, xrand) in (
     let ck = f3(k, xrand) in (
     let ik = f4(k, xrand) in (
     out(c, res); in(c, xmsg)))))
  else (let err_msg =
        geterr(err(mac, xmac, sqn, xsqn)) in
        out(c, err_msg))))))).
```