International Journal of Information Security manuscript No.
(will be inserted by the editor)

Analysis of Privacy in Mobile Telephony Systems

Myrto Arapinis · Loretta Ilaria Mancini · Eike Ritter · Mark Dermot Ryan

Received: date / Accepted: date

Abstract We present a thorough experimental and formal analysis of users' privacy in mobile telephony systems. In particular, we experimentally analyse the use of pseudonyms and point out weak deployed policies leading to some critical scenarios which make it possible to violate a user's privacy. We also expose some protocol's vulnerabilities resulting in breaches of the anonymity and/or user unlinkability. We show these breaches translate in actual attacks which are feasible to implement on real networks and discuss our prototype implementation. In order to countermeasure these attacks we propose realistic solutions. Finally, we provide the theoretical framework for the automatic verification of the unlinkability and anonymity of the fixed 2G/3G procedures, and automatically verify them using the `ProVerif` tool.

Keywords Privacy · Automatic Verification · ProVerif · Mobile Telephony · Pseudonym

This work is supported by the EPSRC, under grant EP/F033540/1 Verifying Interoperability Requirements in Pervasive Systems, EP/H005501/1 Analysing Security and Privacy Properties

Myrto Arapinis
University of Edinburgh
E-mail: marapini@inf.ed.ac.uk

Loretta Ilaria Mancini
University of Birmingham
E-mail: lxm619@bham.ac.uk

Eike Ritter
University of Birmingham
E-mail: e.ritter@cs.bham.ac.uk

Mark Dermot Ryan
University of Birmingham
E-mail: m.d.ryan@cs.bham.ac.uk

1 Introduction

Mobile phones of the latest generation are relatively small but powerful devices that we keep always on and carry wherever we go. We rely more and more on services and apps running on them not only to make calls but to organize and enhance our daily life as well. For this reason, mobile phones are usually equipped with a variety of sensors and communication devices able to collect and/or communicate context-aware and personal data. This data collection and exchange activity take place even when the user is not aware of it. Hence, the user is often oblivious of potential security and privacy threats deriving from the use of these technologies, and has little or no control over the generation, sharing and use of the data itself. Therefore, along with the benefits, the ubiquitous presence of smart, context-aware sensing and computing devices brings with it various concerns, mainly related to the security and privacy properties of such systems.

It is difficult to ensure that a system satisfies the desired security properties. The properties have to be formally defined. Every possible interaction of the adversary with the system has to be considered. The information the attacker gains from the execution of the system and the knowledge he can infer from it should be taken into account. This involve both careful scrutiny of the cryptographic primitives and of the protocol message flow. Formal methods have proved to be effective in highlighting weaknesses of protocols and hence, enabling designers and programmers to patch them and strengthen their security guarantees, as well as in giving strong assurance on a protocol achieved properties by proving the absence of attacks undermining the stated security properties. For example, conference management [**?**], electronic voting [**?**,**?**,**?**], single-sign-on [**?**,**?**],

cloud storage [?], RFID [?], TPM [?,?] and mobile telephony protocols [?] have been scrutinized using manual and automatic verification techniques. Manual proof methods are lengthy and error prone, while automatic verification tools have to compromise in order to achieve decidability (at least for some subclass of processes) by bounding the number of agents and sessions and/or restricting the considered class of cryptographic functions.

Many of the security properties of mobile telephony systems can be modelled and analysed in terms of the classical concepts of confidentiality, authentication and integrity. However, new formal definitions of properties are required to reason about privacy and its different aspects and shades. Some of the most recurring privacy related properties can be described in terms of location privacy, anonymity and unlinkability.

Modelling and verifying these properties is a difficult task since the sources of possible attacks are often hidden in implementation details or even in the protocol logic rather than in the cryptographic guarantees of the employed encryption algorithms. Moreover the modelling of the privacy-related properties often relies on non standard definitions of equivalence and as a result some of the properties are not supported by the currently available automatic verification tools. Hence, modelling and verifying these novel properties is one of the challenges requiring further development of the currently available automatic tools and, of course, of the theory supporting them.

1.1 Contributions

In this work we present a thorough although not complete (i.e. we look only at some of the 2G/3G protocols) analysis of privacy in mobile telephony systems. We experimentally analyse these systems and point out flaws in the deployed policies. We formally analyse some of the protocols and identify vulnerabilities leading to user's privacy breaches which we show are feasible to implement on real networks. We also show that there are realistic solutions for ensuring privacy of mobile telephony subscribers and propose some privacy friendly fixes of the flawed procedures. Moreover, we contribute to the formal analysis of privacy-related properties. In particular, we show how some definitions of unlinkability and anonymity as given in [?] can be adapted to be used for the automatic verification using the `ProVerif` tool. We use this technique to formally analyse the fixes we propose to patch the flawed 2G/3G procedures. In Section 3, 4, and 5 we will investigate the privacy provided by 2G and 3G system. In particular, in Section 3 we recall a well known privacy weakness of mobile telephony which makes it possible to break user anonymity, and we reveal possible privacy leakages produced by signalling procedures of the 2G/3G protocol stack. Further, we conduct experiments to test the effectiveness of the changing pseudonym policies adopted by mobile telephony systems and present a replay attack affecting the TMSI reallocation procedure. In Section 4, we show an attack that affect only 3G systems and takes advantage of the information leakage resulting from the error messages produced by the execution of the AKA protocol. In Section 5 we demonstrate that it is possible to mount the attacks on real 3G networks. In Section 6 we provide privacy enhancing versions of the procedures analysed in Section 3 and 4. Finally, in Section 7 we use formal verification to give guarantees on the privacy properties (anonymity and unlinkability) of the fixed procedures.

1.2 Privacy

As D'Introna points out in [?], privacy is a very important characteristic for the definition of social relationships and more crucially a fundamental property to enable autonomy and individuality. Knowing to be observed influences a person's behaviour; indeed, we adopt different kinds of behaviours depending on who is the observer. We chose to share different things with different people according to for example whether the relationship is intimate, social or work-related. Hence, privacy seems to be fundamental for the definition of oneself.

Privacy is a relatively novel concept that appears for the first time in a 1890 analysis in the Harward Law Review in relation to a case of privacy invasion by the press [?] and there is no universally accepted definition of privacy. In the context of mobile telephony systems, privacy is a major concern [?,?,?,?] because, as previously mentioned, the quality, quantity and accuracy of collected personal data is of great magnitude. Furthermore, the user is often unaware of generating the data and has little or no control over its generation, exchange, transfer, and use.

As in the real world, in the electronic world as well, privacy is a difficult property to define and we rather speak of privacy-related properties instead, which can be general ones or application specific. Moreover, privacy is a complex property to enforce and verify since it is multi-level and multi-protocol, i.e. it crosses more than one layer, if not all layers of the protocol stack, and typically has to be enforced by all protocols in order to be satisfied [?]. While some aspects of privacy are more suited to be enforced by policies as for example the sale

of private information to third parties, others should be part of the verifiable properties embedded in the design of a system. However, privacy seems to be often in conflict with other system requirements as for example accountability, or in contrast with the nature of the system itself as in the case of location based services. For these reasons user's privacy is often overlooked or enforced by complex mechanisms which are difficult to verify by hand.

The most obvious privacy-related property is data confidentiality. However it is generally not sufficient to ensure confidentiality in order to achieve privacy. For example a user identifier could not be public and his identity could be disclosed by correlating his activity [**?**,**?**,**?**]. Moreover multiple accesses to a system from different locations by the same identifier could directly or indirectly reveal a user's position or his movement patterns. Hence, anonymity and unlinkability of a user's activity and access to a system are important aspects of a user's privacy. Location privacy is one of the privacy aspects which has been the subject of many studies [**?**,**?**,**?**]. Several techniques have been proposed to protect a user's location, for example obfuscation, pseudonymity and mix-zones [**?**]. Another interesting privacy-related property which can be for example desirable for private information stored on portable devices is forward privacy which requires that the secrecy of certain information holds even after the device has been corrupted. For example it may require that session keys are not retrievable even in the event that the master keys were disclosed after the device was corrupted. Formal definitions of anonymity and unlinkability are given in [**?**] definitions of untraceability and forward privacy are given in [**?**] and a definition of voter's privacy is given in [**?**]. These definitions are all equivalence-based and often pose challenges for the currently available automatic verification tools because of the definition of the property itself, because of the structure of the protocol or because of the algebraic properties of the involved cryptographic primitives.

In this work we are concerned about mobile telephony systems' privacy. We carry out both experimental and formal analysis in order to test the level of privacy provided to mobile telephony users over the radio link. In particular, we experimentally analyse the use of pseudonyms and expose some critical scenarios which makes it possible to violate a user's privacy. Moreover, we expose some protocol's flaws resulting in breaches of the anonymity and/or user unlinkability (as defined in [**?**]). We show that it is possible to implement the attack we found on a real network using an hacked femtocell. We propose some lightweight fixes of the flawed 2G/3G procedures. Finally, we provide the theoretical framework for the automatic verification of the unlinkability and anonymity of the fixed 2G/3G procedures.

1.3 Security Protocols and Formal Methods

Security protocols aim to protect sensitive data in particular when communicating over an unprotected connection, where there is little or no control over the flow of information and the attacker can intercept, manipulate, replay, inject, replace, substitute and compare messages. This model is sometimes called the Dolev-Yao attacker model [**?**]. This makes the design of security protocols a notoriously difficult task since it does not only involve the protection of data through some sort of smart cryptographic algorithm but also the ability of foreseeing how the flow of messages could be used to retrieve, deduce and get access to sensitive information. Details hidden in the protocol logic often allow to break a protocol security without even breaking the underlying cryptography. Examples are the attacks recently found on TLS [**?**], SAML [**?**,**?**] and OAuth [**?**]. These attacks do not take advantage of weak cryptography but rely on the logic and interaction between concurrent protocols at the same and/or different layer of the stack. Another example is the traceability attack on the French e-passport which allows one to trace an e-passport holder by performing a replay attack [**?**]. Hence, to declare a protocol secure one should check it against any possible adversary and any possible interaction. Moreover, one would need to state what being secure means for the protocol, i.e should specify the security properties the protocol aims to achieve, and this would depend on the purpose of the protocol and on the application requirements. Hence, to evaluate the security of a protocol it is very important to rigorously define both the protocol itself and the desired security properties.

Formal methods help establishing the security of protocols in three ways:

- rigorously modelling security protocols and the attacker model
- rigorously defining security properties
- evaluating the protocols against the desired properties

There are two different formal approaches to the problem of protocol security. The computational approach is closer to the actual protocol implementation, represents messages as bitstrings, cryptographic functions as polynomial time algorithms and the adversary as any probabilistic polynomial time algorithm and can give strong guarantees on the security of the cryptographic algorithms. However, proofs in this setting can

be very long, difficult, error prone and not very accessible to careful scrutiny. Moreover they are not very well suited for automation.

We will use symbolic methods instead. Symbolic methods abstract from the details of the cryptographic algorithms. Messages are represented by terms and cryptographic primitives are represented by function symbols which can be applied to terms. The properties of cryptographic primitives are abstracted by algebraic properties. In general, cryptographic primitives are assumed to be perfect. Hence, the adversary can have any interaction with the protocol, injecting, modifying, replaying, intercepting messages but cannot break cryptography.

As in the case of computational methods, in the symbolic world the security of protocols can be established by means of long, tedious and error prone manual proofs or by manually exhibiting a counter-example i.e. an attack. However, lots of work focus on automating the proofs and/or search for counter-examples. In this area there are still plenty of challenges and room for improvements and researchers continuously aim to expand automatic verification to larger classes of protocols and equational theories for both reachability and equivalence-based properties.

Some of the most famous formalisms in the field of symbolic methods are process calculi, constraint systems and strand spaces. In this work we will use the applied pi-calculus [**?**] which is a calculus for modelling cryptographic protocols offering flexibility in the range of cryptographic primitives represented by function symbols and equational theories and making the adversary knowledge explicit thanks to the frame construct. Many results obtained using different formalisms such as spi-calculus and constraint systems can be transposed into the applied pi-calculus. We describe some of the most relevant.

The general problem of security is shown to be undecidable even for a bounded number of sessions for both reachability-based properties [**?**] and for equivalence-based properties [**?**]. Restricting the calculus and or to the equational theory can yield decidability results.

One of the most common restrictions is to consider only a finite number of sessions. Automatic verifiers for bounded processes can be very useful to find protocol flaws, however when they do not find any attack, no guarantees of the absence of attack can be derived for a number of sessions greater than the one used to run the verifier on the protocol. To obtain general security results an unbounded number of sessions should be considered. Another common restriction to the calculus consists in forbidding else branching, and the equational theory is often required to be convergent.

A decision algorithm for trace equivalence of bounded processes with no else branching and a pre-defined signature consisting of pairing, symmetric and asymmetric encryption is given in [**?**] along with a prototype implementation for checking deducibility and static equivalence. However, for deciding trace equivalence is required to implement a further procedure that generates a pair of constraints for each possible interleaving. The AKiss tool implements two procedures for bounded processes with no else branches and for convergent rewrite systems with the finite variant property. One procedure under-approximate trace equivalence and can be used to prove protocols correct. The other over-approximate trace equivalence and can be used to discard incorrect protocols. The tool can be used to check observational equivalence of determinate processes since in this case observational equivalence coincides with the under approximation of trace equivalence.

The decidability of observational equivalence of "*simple*" processes with no replication and with no else branches for subterm convergent equational theories is shown in [**?**]. The decision procedure relies on the decidability of the equivalence of constraint systems given in [**?**]. This procedure has not been implemented so far. To the best of our knowledge the only tool able to automatically verify both reachability-based [**?**] and equivalence-based [**?**] properties for unbounded processes with else branches and convergent theories is the `ProVerif` tool [**?**]. Blanchet, Abadi and Fournet [**?**] introduce the concept of bi-process, which is a pair of processes that differ only in the choice of some term. `ProVerif` can prove observational equivalence of bi-processes. However the procedure is sound but not complete, meaning that the tool can prove that a property holds on the given protocol model but when it outputs an attack trace this could be a false attack. Moreover, the tool might not terminate.

2 Background: Mobile telephony systems

GSM (Global System for Mobile Communication) and UMTS (Universal Mobile Telecommunications System) are the most widely used mobile telephony standards with billions of users worldwide. GSM was developed by ETSI (European Telecommunications Standards Institute) in order to promote a common standard for the European cellular telephony replacing the multitude of first generation standards. UMTS is specified and maintained by the Third Generation Partnership Project (3GPP), it was introduced in 1999 to offer a better support for mobile data applications by increasing the data

rate and lowering the costs of mobile data communications. Furthermore, UMTS offers an improved security architecture with respect to previous mobile communication systems such as GSM. Both GSM and UMTS have been improved and extended several times. We will use the terms 2G and 3G to indicate the wider set of standards including GSM and UMTS respectively. Most of protocols and issues presented in the following Sections are common to both 2G and 3G systems, when this will not be the case, it will be explicitly pointed out. In the rest of this paper we will adopt a unified terminology to address mobile telephony systems components. However 2G and 3G standards may not use the same terminology even when addressing components having the same functions/purposes.

In the next Section, we will introduce the 2G/3G network architecture and we will describe in more details their security features. We will then summarize some of the well-known weaknesses of mobile telephony systems, along with relevant work on 2G/3G security and privacy.

2.1 GSM/UMTS Architecture

The 2G/3G network architecture, depicted in figure 1, integrates both GSM and UMTS components. The user side of the network consists of Mobile Stations (MS). This term is used in mobile telephony systems to indicate both the Mobile Equipments (ME) such as mobile phones, and the so-called SIM, or USIM card (Universal) Subscriber Identity Module in 2G, 3G systems respectively. The (U)SIM card identifies the user as a legitimate subscriber within a mobile telephony operator network. To access the services offered by a mobile operator, a MS connects through radio communication technology to the UTRAN (UMTS Terrestrial Radio Access Network) or GERAN (GSM/EDGE[1] Radio Access Network) network, that is a GSM access network. A mobile station directly communicates with a BTS (Base Transceiver Station) or Node B which covers the area the MS is located in. One or more Nodes B are connected to a Radio Network Controller (RNC), and one or more BTS are connected to a Base Station Controller (BSC) defining a cell. A group of cells forms a Location Area. RNCs and BSCs manage the radio resources and inter-cell handover. They are the interface between the mobile station and the core network. The core network offers circuit-switch and packet-switch services. The Mobile Switching Centre (MSC) and Gateway Mobile Switching Centre (GMSC) offer inter and intra-network circuit-switching domain services, respectively, and interface the 2G/3G systems with the traditional fixed telephony network. The Serving GPRS[2] Support Node (SGSN) and the Gateway Serving GPRS Support Node (GGSN) offer, respectively, inter and intra-network packet-switching domain services as well as connecting 2G/3G networks with the internet. Within the core network the Home Location Register (HLR) stores permanent sensitive information of subscribers such as identities, service profiles, and activity statuses. These informations are linked to the SIM and recorded when stipulating a contract with the Mobile Network Operator (MNO). 2G/3G systems offer roaming capabilities between different mobile network operators, and between the different technologies (provided that the mobile equipment supports them), meaning that a mobile station can be connected to a visited network, the Serving Network (SN), which might be different from the subscriber's Home Network (HN) and which could be using a different standard. Each subscriber has a long term identifier IMSI (International Mobile Subscriber Identity) that is stored in the (U)SIM and a temporary identifier TMSI (Temporary Mobile Subscriber Identity), allocated by the serving network to protect the subscriber's identity privacy. The Visitor Location Register (VLR) stores temporary informations about subscribers visiting a given location area of the serving network and maintains TMSI/IMSI associations. The network operator and each subscriber share a different unique long term secret key used for authentication purposes. This key is stored in the (U)SIM. The Authentication Centre (AuC) is a protected database storing associations between subscriber identities (IMSI) and long-term keys.

In the rest of this work, we only consider a simplified network architecture, since we are interested in third party attackers having access only to the radio path and not to the wired network infrastructure. This architecture involves simply the mobile stations and the network. The network models both the UTRAN/GERAN Base Station that the MS is directly communicating with and the complex structure of databases and servers connected with it and forming the UMTS/GSM control network. In particular we do not distinguish between serving network and home 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

[1] EDGE (Enhanced Data rates for Global Evolution) is a standard part of the 2G set aiming to provide faster bit rates for data application

[2] GPRS (General Packet Radio Service) adds packet switch functionalities to GSM

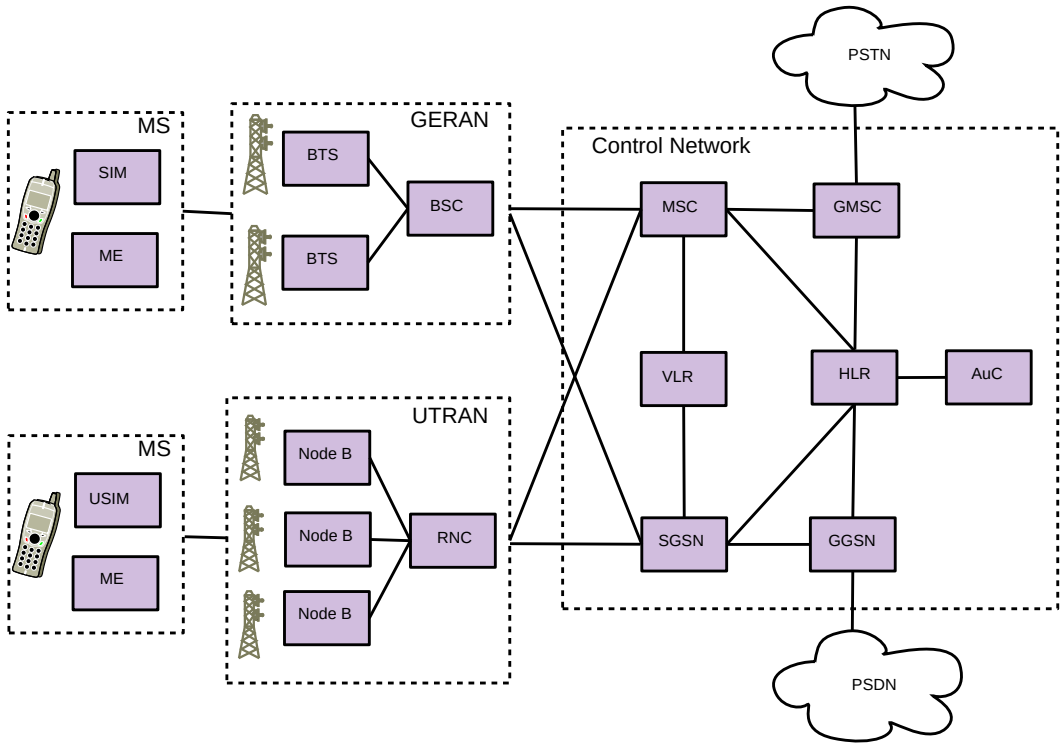

Fig. 1 2G/3G Architecture

small, but at the same time precisely model the interactions over-the-air between MS and network, which are the subject of our analysis.

2.2 GSM/UMTS Security Features

The intended security features of 2G and 3G systems are slightly different, as 3G systems aim to overcome the weaknesses of 2G systems such as the lack of mutual authentication and the use of weak ciphering algorithms.

2G systems aim to provide the following security features [**?**]:

- **Subscriber identity confidentiality**: the IMSI is not made available or disclosed to unauthorized individuals, entities or processes.
- **Subscriber identity authentication**: the corroboration by the mobile network communicating with a mobile station that the subscriber identity (IMSI or TMSI), sent by the mobile subscriber to identify itself, is indeed the one claimed.
- **User data confidentiality**: the user information exchanged on traffic channels is not made available or disclosed to unauthorized individuals, entities or processes.

The 3G communication system aims to improve the security features offered to mobile telephony subscribers, requiring for example mutual authentication between the user and the network and specifying more precisely the provided user's privacy properties in terms of identity confidentiality or anonymity, location confidentiality and user untraceability or unlinkability. The security properties stated by the 3G standard are the following [**?**]:

- **User Identity Confidentiality**:
  - *user identity confidentiality*: the permanent user identity (IMSI) of a user to whom a service is delivered cannot be eavesdropped on the radio access link;
  - *user location confidentiality*: the presence or the arrival of a user in a certain area cannot be determined by eavesdropping on the radio access link;
  - *user untraceability*: an intruder cannot deduce whether different services are delivered to the same user by eavesdropping on the radio access link.
- **Entity Authentication**:
  - *user authentication*: the serving network corroborates the user identity of the user;
  - *network authentication*: the user corroborates that he is connected to a serving network that is authorized by the user's HN to provide him services; this includes the guarantee that this authorization is recent.

- **Confidentiality**:
  - *cipher algorithm agreement*: the MS and the SN can securely negotiate the algorithm that they shall use subsequently;
  - *cipher key agreement*: the MS and the SN agree on a cipher key that they may use subsequently;
  - *confidentiality of user/signalling data*: user/signalling data cannot be overheard on the radio access interface;

Both 2G and 3G systems rely on an Authentication and Key Agreement (AKA) protocol, on the use of encryption of the confidential data transmitted on the radio channel and on the use of periodically changing temporary identities in order to achieve the above mentioned security properties. Moreover, 3G systems use integrity protection of the sensitive data. Unlike the 2G authentication protocol, the 3G AKA protocol allows MS and SN to establish not only a ciphering key but also an integrity key, and achieves mutual authentication of MS and HN.

2.3 Previous Work on 2G/3G Security and Privacy

Most of the work on security of mobile telephony systems concerns properties such as secrecy, integrity and authentication [**?**,**?**,**?**]. There are only few formal and experimental studies concerning the level of privacy provided to users by mobile telephony systems [**?**,**?**,**?**].

2G Vulnerabilities. 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 standard as a breach of the user identity confidentiality [**?**, 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) [**?**,**?**] by triggering the identification procedure from a fake operator base station (configured with the corresponding mobile network and country code settings). Moreover, the 2G fake base station can then force the attached mobile station to use no encryption for data and signalling communication and hence it can capture ingoing and outgoing plaintext traffic. 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 [**?**].

Foo Kune et al. [**?**] present a study on the use of the paging procedure to locate mobile telephony users. They perform a tracking attack relying on passive sniffing of paging response messages triggered by placing silent phone calls (obtained by hanging up before the receiving phone rings) to the victim phone. This technique allows to reveal the presence of the victim in an area monitored by the attacker. Munaut and Nohl [**?**] previously outlined a similar technique. They performed a GSM sniffing attack, which allows to eavesdrop a GSM phone call by using a modification of the osmocom-BB [**?**] opensource implementation of the GSM protocol stack and an old Motorola mobile phone. Differently from Foo Kune et al., they used a silent SMS to trigger the paging responses needed to locate the victim. Although these works take advantage of the fact that a TMSI is allocated for a long time window, they do not analyse the security and privacy provided by the TMSI reallocation procedure. Moreover, in order to perform the attack, the adversary needs to know the mobile number of the victim.

Engel showed at the 25C3 conference [**?**] how network signalling messages, triggered when sending/receiving SMS messages, can be used to localize mobile telephony users. He suggests that network operators should use home routing, i.e. forwarding through the home network, as a countermeasure to this SMS tracking attack. This attack requires access to the intra-network communication infrastructure, which although possible may require subscription to a pay per query service. In Section 3, 4, we will analyse the privacy of the more exposed over-the-air communication which is available to any attacker with a radio enabled device. We will not assume to have control over the less easily accessible intra-network communication channel.

The *gsmmap* project [**?**,**?**] assesses and visually renders on a map the level of security and privacy provided by 2G and 3G network operators across the world. The data about the security of 2G systems is gathered using a variant of the open source GSM protocol stack developed within the osmocom-BB project, while data about the security of 3G networks can be collected using some specific model of mobile phones and downloading the gsmmap specific app. The project relies on volunteers to contribute the needed data, hence the map is not complete and some of the result are estimations based on the available data. In particular, gsmmap aims to check if network operators are protecting the users from well known attacks by adopting countermeasures such as the use of A5/3 encryption, padding randomization, and full authentication for outgoing calls and SMS to prevent impersonation and interception, and the use of regular TMSI updates, and home routing to prevent Engel's SMS tracking attack. However, the list of security protection mechanisms gsmmap is looking for in order to evaluate the security offered by mobile telephony systems is continuously updated according

to the most recent weaknesses found and the suggested countermeasures.

Golde, Redon and Seifert [?] showed that it is possible to hijack a mobile terminating service such as the reception of a phone call or an SMS message and even perform a targeted or large scale denial of service by answering paging requests faster than the intended recipient. They implemented their attacks on GSM networks, but speculate they would work on UMTS and LTE as well since the attacked procedure i.e. the paging procedure is common to the three systems.

3G Weaknesses Resulting from 3G/2G-interoperability. 3G mobile stations are as well vulnerable to 2G IMSI catcher, in fact 3G systems are fully interoperable with 2G systems, and 3G mobile stations can roam in 2G networks. A 3G mobile stations can be induced to attach to a 2G base station by broadcasting a stronger signal with respect to the 3G one. To avoid this, 3G mobile stations can be configured to use only 3G networks. To the best of our knowledge the only implementation of a pure 3G IMSI catcher is the one presented in [?] and is realised using a modified femtocell. 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 [?] and to masquerade as a legitimate subscriber obtaining services which will be billed on the victim's account [?]. However, these attacks cannot be carried out on pure 3G networks, 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 [?]. 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.

LTE Privacy. The LTE authentication and key agreement protocol suffers from a linkability attack [?] similar to the one we present in Section 4.2.

Previous Formal Analysis of 3G protocols. 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 [?], 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 [?]. The framework applied in [?] cannot be used to specify unlinkability and anonymity properties, let alone reason about them. The formal framework we used allows us to precisely define and verify privacy related properties. Hence, we can discover privacy attacks on the modelled protocols and propose solutions which we then formally prove 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 [?]. In particular, the authors aim to achieve MSs anonymity with respect to the serving network, and location privacy of mobile stations with respect to the home network. To achieve this purposes, they propose a new mechanism for the location update and a three way handshake protocol, to be used instead of the currently used 3G AKA protocol. However, this work is not supported by a formal model of the AKA protocol, nor does it provide a formal verification of the properties enforced by the proposed protocols. Moreover, their attacker model considers the network as not fully trusted, while in this work we are only concerned with third party attackers controlling the radio link communications.

The feasibility of implementing the solutions we propose to fix the identification procedure, IMSI paging procedure and AKA protocol on in the current 2G/3G infrastructure is discussed in [?].

3 Analysis of Mobile Systems' Privacy: 2G/3G Weaknesses

In this Section, we describe some of the mobile systems protocols that are dealing with the identity of mobile phone users and point out breaches of the user's privacy, which expose a subscriber's identity and allow an attacker 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. In Section 4 we illustrate a privacy breach that exploits a procedure specific of the 3G system. The attacker considered in this Section is capable of sending, receiving and sniff-

ing[3] messages over-the-air. Further, we assume that the attacker has unlimited access to the radio link between the mobile station and the base station but no access to the communication within the core network or between the base station and the core network. As we will see, the attacker does not need to know any keys, nor perform any cryptographic operation. In fact, the vulnerabilities exposed in this Section take advantage of the poor management of the user identities, due to the fact that at the time mobile telephony systems were designed, active attacks were considered too costly and hence unlikely to be performed. In fact, mounting an attack required very expensive equipment such as a base station. However as we briefly discussed in Section 2.3, nowadays the development of open source devices and software has lowered significantly these costs.

In the rest of this Section we present two procedures of the mobile telephony protocol stack that involve the mobile station identity, namely the identification procedure and the paging procedure and we show how these procedures can be used to violate a user's privacy. Furthermore, we introduce the TMSI reallocation procedure that is the procedure implementing the pseudonym changing mechanism adopted by mobile telephony systems in order to provide anonymity and unlinkability. We experimentally evaluate the effectiveness of the pseudonym changing mechanism on real networks. We finally show a replay attack on the TMSI reallocation procedure which is enabled by the fact that the reuse of previously established keys is allowed by the standard and, and is indeed a policy adopted by real network operators (see Section 3.8).

3.1 Identification Procedure

The identification procedure [**?**] allows the network to request the credentials of a mobile station. In particular, the network can request a mobile station's IMSI, that identifies the user's SIM card or better identifies the user as a legitimate one having a contract with the MNO, or a mobile station's IMEI, that identifies the specific handset. The identification procedure is always initiated by the network. It is typically used when a mobile station connects to a network for the first time. Once the network identifies the MS, it can initiate a procedure to establish a shared short term encryption key and then usually assigns a temporary identity, TMSI, to the MS. The temporary identity is transmitted to the MS in an encrypted fashion and is used instead of the IMSI in the following communications.

[3] With sniffing we here mean the ability of listening to the communication happening on a dedicated channel established between a target mobile station and the base station it is attached to

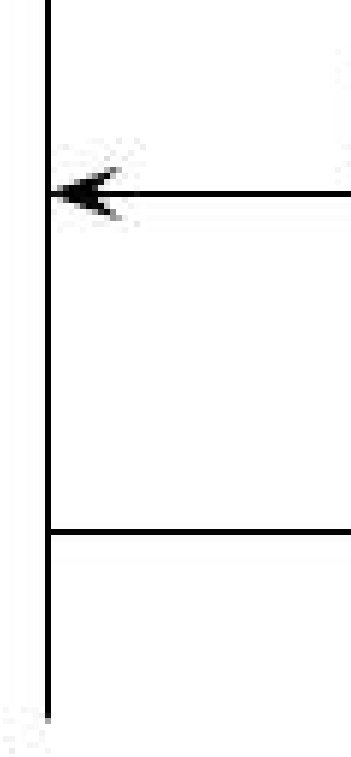

Fig. 2 Identification Procedure. The identity of the mobile phone is sent in clear on the radio path.

The identification procedure consists of two messages: the identity request message and the identity response message. The parameter IMSI_Req specifies that the identity requested by the network is the IMSI. The identity response message is sent in clear along with the requested identity. (see Figure 2).

3.2 Identification Procedure Attack

The identification procedure is acknowledged in the standard as breaching the user identity confidentiality. It allows a passive attacker to overhear an MS's IMSI. Moreover, any attacker controlling a tool able to act as a base station can trigger at any moment the identification procedure and capture the identity response, and thus can trace the presence of a particular subscriber in the range of his device. Moreover, he can track the movements of the subscriber by installing tracking devices in different areas. This attack to the subscribers' privacy is known as IMSI catcher attack [**?**,**?**,**?**]. Typically, the attacker would control a fake BS that just sends an identity request and waits for the response.

3.3 Paging Procedure

The paging procedure defined in [**?**] is used to locate a mobile station in order to deliver a Mobile Terminated (MT) service to it, for example an incoming call or SMS

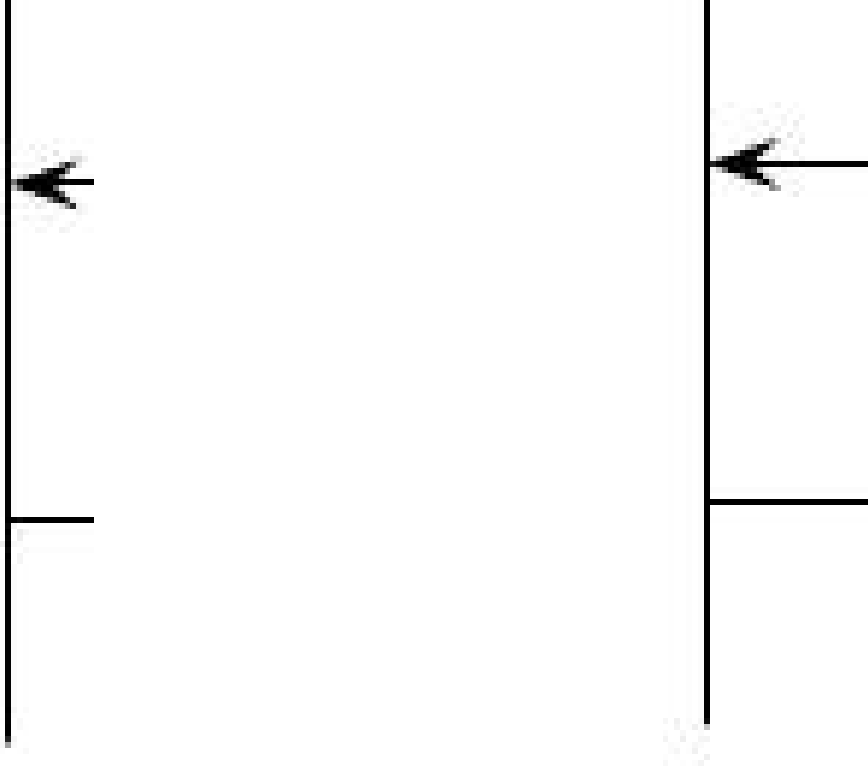

Fig. 3 IMSI Paging Procedure in GSM: the paging request message is sent on a common radio channel and contains the long term identity of the paged mobile station IMSI in cleartext. The paging response message also contains the IMSI and is sent in clear on a dedicated channel.

Fig. 4 IMSI Paging Procedure in UMTS: the paging request message is sent on a common radio channel and contains the identity of the paged mobile station (in this case the IMSI) in cleartext. The paging response message contains the mobile phone's temporary identity TMSI and is sent in clear on a dedicated channel.

message. The network sends paging requests in all the most recently visited location areas in order to locate the mobile station. In fact, most MS are idle most of the time to save battery so the network does not know which base station is currently providing the best reception level to the MS. 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 after a TMSI is assigned to the mobile station by the network, and hence the TMSI should be used to identify a MS in the paging request. However, the IMSI can be used when the TMSI is not known by the network.

A mobile station receiving a paging request message checks whether one of the identities included in the message is its own. If the check succeeds the MS establishes a dedicated channel to allow the delivery of the service. On this dedicated channel the MS sends a paging response message containing its IMSI in GSM networks (see Figure 3), the most recently assigned TMSI in UMTS networks (see Figure 4).

3.4 IMSI Paging Attack

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 whose identity is known to the attacker and in 3G networks to correlate IMSI and TMSI. The observation of the related paging response allows the correlation of the victim's IMSI with his current TMSI. In practice, an attacker would need to confirm the link between the paged IMSI and the related TMSI by replaying the attack several times.

3.5 TMSI Reallocation Procedure

A mobile station (MS) is uniquely identified by means of its IMSI. If a third party that eavesdrops on the radio link was able to identify wireless messages as coming from a particular mobile phone, he would be able to track the location of the mobile phone user in real-time. This could lead to stalking and other forms of harassment, as well as more mundane invasions of privacy. To avoid over-the-air attackers from identifying and linking a user's transactions, a temporary identity called TMSI is assigned by the network and is used to identify the mobile station in protocol messages sent in clear over-the-air. The mobile station identity (its TMSI, if available, or its IMSI) is always included in the first message sent from the MS to the network after the establishment of a dedicated channel. This allows the network to identify the MS before delivering a service to it. For example, the identity is carried in location update requests, CM (Call Management) requests, and paging responses. The use of TMSIs avoids the exposure of the long term unique identity (IMSI) and hence aims to provide third-party anonymity and unlinkabil-

ity to mobile telephony subscribers. Temporary identities are periodically updated by the network by means of the *TMSI reallocation procedure*. The 3GPP standard specifies that a new TMSI should be assigned at least at each change of location area. Besides this constraint, the choice of how often a new assignment is performed within a location area is left to the network operators [**?**]. In order to prevent an adversary linking the old TMSI with the new one, the assignment of a new TMSI is performed in ciphered mode. The session key used to encrypt the new TMSI is established by executing the AKA protocol.

The TMSI reallocation assigns a new pseudonym (TMSI) to a mobile station. The new TMSI is sent to the mobile station in an encrypted fashion. Figure 5 depicts the TMSI reallocation procedure as defined in the 3GPP standard [**?**,**?**]:

- The mobile station sends a first message on a dedicated channel. This message contains the current MS's temporary identity $oTMSI$;
- upon receipt of this message, the network can identify the MS and establish means for ciphering of the subsequent communication on the dedicated channel;
- the rest of the communication is then encrypted and consists of a TMSI reallocation command message containing a new pseudonym $nTMSI$ randomly chosen by the network and the current location area $nLAI$ (the area within which $nTMSI$ is meaningful);
- this message is followed by a TMSI reallocation complete message which is sent by the MS to acknowledge the completion of the reallocation procedure.

If the network does not receive the expected acknowledgement from the MS, it maintains both $oTMSI$ and $nTMSI$ as valid pseudonyms for the IMSI. The network can perform a TMSI reallocation at any time whilst a dedicated channel is established. The standard does not fully specify how often this procedure should be performed. However, it mandates that it should at least be performed at each change of location [**?**]. The standard defines two options for the management of the means for ciphering (i.e. to establish the ciphering key CK):

(1) either a fresh ciphering key is established by executing the authentication procedure;
(2) or a previously established ciphering key can be restored by means of the security mode set-up procedure, which allows the MS and the network to agree on a ciphering algorithm.

Fig. 6 Experimental Tools

3.6 Subscriber Privacy Experimental Analysis

In this Subsection, we will present the results of our experimental analysis of the use of TMSIs in mobile networks and highlight critical scenarios from a privacy point of view. Our aim is to analyse whether the changing pseudonyms policy adopted by mobile telephony systems guarantees user privacy as intended. In particular, two aspects appear to be important:

1. TMSI reallocation will protect user privacy only if TMSIs are re-allocated often enough, and at the right times (e.g., when users move between locations). The 3GPP standard does not rigorously define the conditions under which TMSI reallocation takes place, leaving the choice to the network operators. As our experiments show this leaves users open to privacy abuses.
2. The success of TMSI reallocation requires that a wireless eavesdropper cannot link the new TMSI to the old one. Encrypting the TMSI in the allocation message is necessary (but may not be sufficient) to ensure that. It turns out that other factors, in particular the use of fresh session keys for each TMSI reallocation, are also necessary to guarantee unlinkability of old and new TMSIs. The 3GPP standard does not mandate this, again leaving user privacy subject to choices made by network operators.

We monitored over-the-air communications of idle and active MSs in order to understand how real networks implement user identity confidentiality through the use of TMSIs, both in terms of frequency of reallocation, and ciphering keys used. Our experiments confirm that the reuse of previously established keys is a commonly adopted policy. However, we show that in case the reuse of encryption keys is adopted for the execution of the TMSI reallocation procedure, this enables

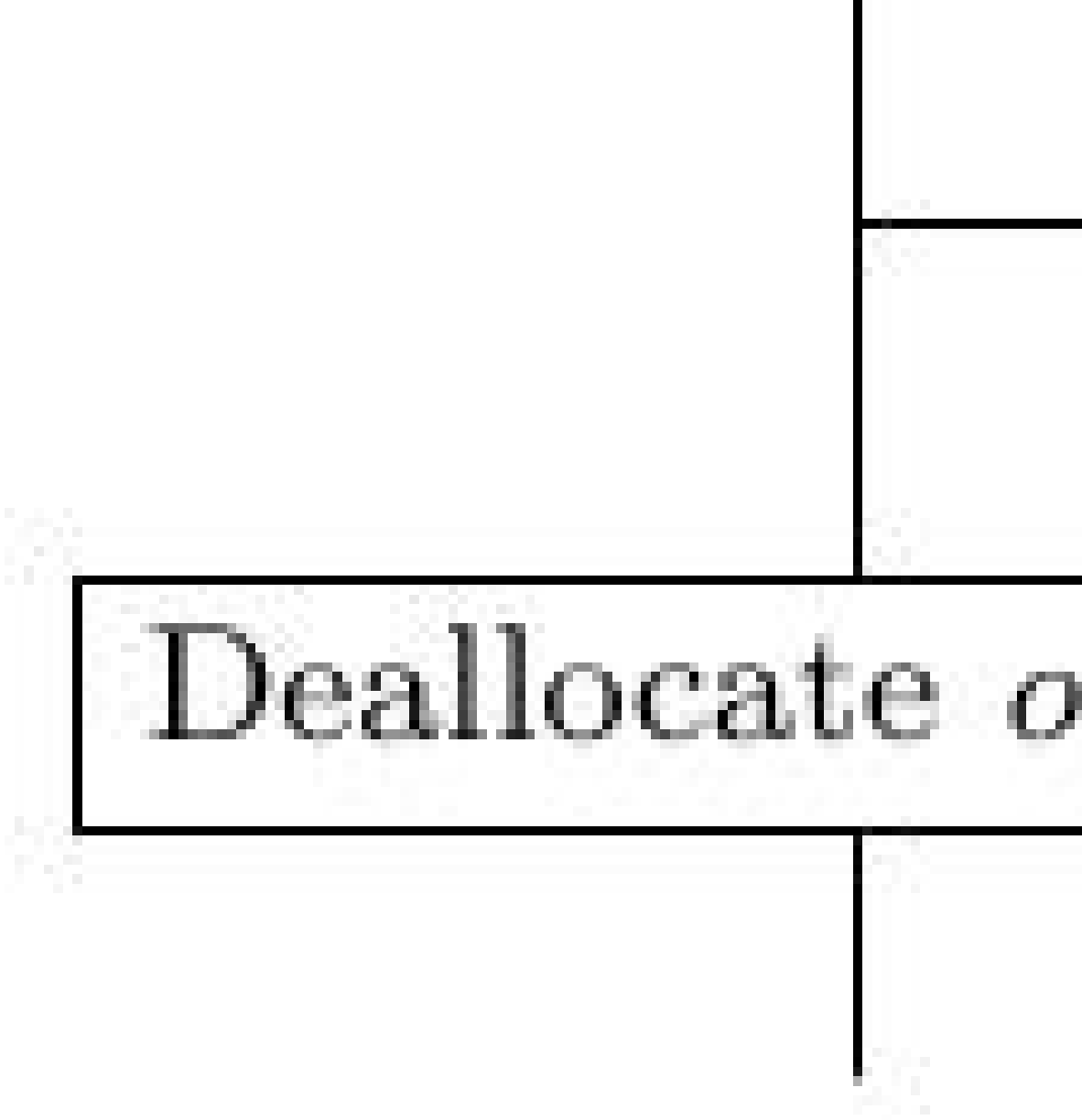

Fig. 5 TMSI Reallocation Procedure: the new TMSI along with the current Location Area Identifier (LAI) is sent in an encrypted message to the mobile station in order to avoid users' linkability

a linkability attack which makes it possible to link old and new TMSIs.

Our experiments were carried out using an old GSM Motorola C115 mobile phone in France, UK, Greece, and Italy and using SIM cards from all the major UK, Greek, and Italian network operators.[4]

3.7 Experimental Settings and Scenarios

The Motorola C115 has a TI Calypso baseband chipset which is supported by the Osmocom-BB project [**?**]. The Osmocom-BB project includes an open source implementation of the GSM baseband and various other applications aiming to implement a GSM mobile station. The radio communication functions are implemented in the firmware which is flashed from a laptop into the mobile phone through the Osmocon software, by means of a T191 unlock cable (Figure 6). The firmware implements layer 1 of the GSM protocol stack, while layers 2 and 3 are implemented in specialised applications running on the laptop and communicating with the mobile phone through the T191 cable (Figure 8). In particular, we used the 'mobile' application which implements layer 2 and 3 of the GSM protocol stack to provide all the basic functions of a mobile phone (network registration, location update, making and receiving calls, and sending and receiving SMSs). The mobile phone activities are logged on a shell terminal and the radio communication is encapsulated in UDP packets sent to a configurable IP address. This traffic can be captured through the Wireshark network traffic analyser [**?**]. Interactions with the mobile phone are enabled

[4] More specifically, we used O2, T-Mobile, Vodafone, and Orange in the UK; Vodafone and Wind in Greece; Bouygues and Orange in France; and Wind, Vodafone and TIM in Italy.

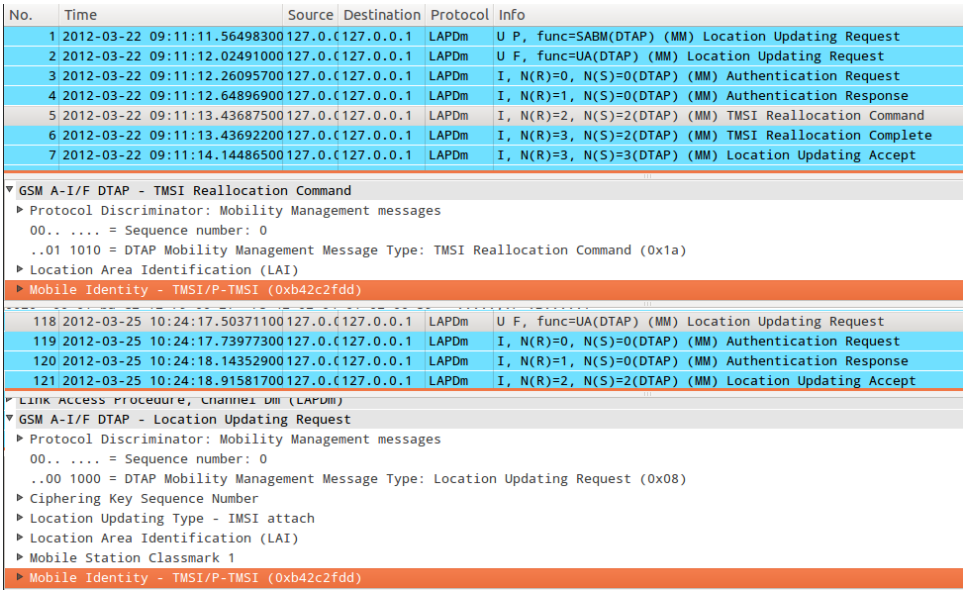

Fig. 7 Trace of a UK Vodafone SIM card obtaining a new TMSI (0xb42c2fdd) on 22/03/12. The same TMSI is still in use on 25/03/12 after 3 days from its allocation.

by a telnet command interface. This allows one to manually select a network, start phone calls, send SMSs and service requests, etc. We captured over-the-air messages using the 'mobile' application in different settings:

1. mobile station in idle state and not moving;
2. mobile station in idle state and moving across two urban areas;
3. mobile station involved in activities such as receiving or starting phone calls, receiving or sending SMSs, and requesting services as for example call diversions.

Since the 3GPP standard merely gives guidelines, real networks differ in the implementation details of the TMSI reallocation. To understand if the different implementations achieve the privacy guarantees they were intended for, we analysed the traffic captured with the mobile application. In particular, we are interested in finding out if the frequency of TMSI reallocation execution is high enough to defeat passive and active tracking attacks, if the policy of changing TMSI at least at each change of location is actually implemented so to obtain at least location dependent privacy, and if the frequency of execution of the TMSI reallocation procedure is related to the amount of activity of the MS (i.e., to how often the TMSI is exposed to overhearing).

3.8 Findings/Results

The 3GPP standard relies on the use and frequent reallocation of TMSIs in order to provide user's untraceability. In particular, it mandates that the TMSI reallocation should be performed whenever the MS moves between "location areas" (identified by location area identifiers, LAIs). However, it is known that location areas often span over several square kilometres, and a subscriber's movements are typically confined within one or two location areas [?,?]. So location areas may be too large to trigger TMSI reallocations in practice. We report on three different scenarios showing that some of the actual implementations of the strategy for changing pseudonyms to avoid tracking are not offering enough privacy guarantees to the mobile telephony subscribers. Our observation and their consequences on users' privacy are discussed in this Section[5].

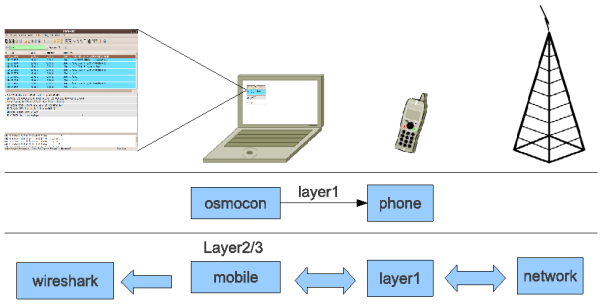

Fig. 8 Osmocom-BB architecture

The TMSI reallocation procedure is rarely executed. Although in the standard the privacy offered to mobile phone bearers is based on frequent up-

[5] The traces that allowed us to draw the conclusions presented are made available for inspection [?]

dates of TMSIs, our experiments show that the same TMSI can be allocated for several hours and even days. Moreover, turning on and off the MS does not usually result in a new TMSI being allocated. As an example Figure 7 shows that a TMSI allocated on 22/03/2012 had not been updated by 25/03/2012, making the phone trackable for a period of 3 days. This behaviour can be observed for the major UK, Greek, French and Italian network operators. An attacker could take advantage of the long life of a TMSI and monitor a few sub-areas using short range devices in order to obtain a fine grained tracking of his victim within a same LAI.

We observed that the major UK network operators and the Vodafone and TIM Italian operators rarely execute the TMSI reallocation even in presence of MS activity, but the first message sent by a MS when requesting or receiving a service contains its TMSI, hence exposes it to eavesdropping third parties. As mentioned in Section 2.3, TMSI liveness makes it possible to locate mobile telephony users without alerting them. The attack consists in paging the victim and hence provoking a paging response. To reduce the set of answering TMSIs to the victim's one, the attacker must repeat the process several times because more than one MS could be sending a paging response at the same time and it is possible only if the TMSI is not reallocated even in case of activity exposing the TMSI (*e.g.*receiving calls). The attack in [**?**] thus relies on the low frequency of TMSI reallocations and demonstrates that changing pseudonyms, as mechanism to provide location privacy, is not effective without a policy for changing of pseudonyms which takes into account the actual exposure of the pseudonym caused by the mobile station activity.

A change of location area does not imply a change of TMSI although such a change is mandated by the 3GPP standard. We observed this behaviour when capturing the signalling messages of a mobile station moving by coach between different cities in the UK, using the Orange and the O2 networks where we observed the same pseudonym being accepted in different location areas with no further execution of the TMSI reallocation procedure. Assuming an average speed of 70Km/h we observed that a new TMSI was assigned after about 45 min (about 53km) and a second one after about 60 min (about 70km) while we observed a change of LAI every 5 min on average and hence a new TMSI should have been allocated, on average, about every 3km. Figure 10 shows an example trace where a TMSI used at location 234/33/1381 (packet no. 668) is accepted at a different location 234/33/29 (packet no.678). The fact that a TMSI was accepted in two neighbouring LAIs contradicts the specification that a TMSI reallocation should be performed at least at each change of location. However, changing pseudonym when changing location area would provide location-dependent privacy to the user since it would prevent passive tracking across different LAIs. The combination of the two behaviours reported so far (i.e. keeping the same TMSI for a long period of time and not changing it when changing location area) enables the attacker to both track his victim within an area and follow him across different areas without doing any extra effort other than passively sniffing.

Previously established keys are restored and used to encrypt the TMSI reallocation procedure. Our captures confirm that the reuse of previously established keys is a policy adopted by real networks. In particular, we show that this policy is adopted for the execution of the TMSI reallocation procedure. The experiments we performed show that major UK and Italian network operators[6] reuse previously established keys instead of performing the authentication procedure before each execution of the TMSI reallocation procedure. Figure 9 shows a trace from a UK Lebara SIM card attached to the Vodafone network performing a location update (packet no. 4063). Then the execution of the authentication procedure establishes a new ciphering key (packets 4065, 4068) and consecutively the TMSI reallocation procedure (packets 4079, 4081) is executed. The subsequent TMSI reallocations (packets 9691, 9693, 71695, 71697, 92653, 92655) are executed without previously performing the authentication procedure and hence reusing the previously established ciphering key.

The reuse of a previously established ciphering key enables a replay attack. We describe the TMSI reallocation replay attack in Section 3.9.

3.9 TMSI Reallocation Replay Attack

So far in Section 3 we analysed the TMSI reallocation procedure from an *experimental* point of view. In this Section we demonstrate a replay attack on this procedure which allows a third party to violate a user's privacy in spite of the reallocation protocol. This attack is enabled by the 3GPP standard policy allowing to restore previously established keys. The experiments reported in Section 3 confirm that this policy is commonly adopted across real mobile telephony operators. We now show that a linkability attack is enabled by the reuse of session keys making it possible to link old and new TMSIs on real networks. In particular, a replay attacks such as the one depicted in Figure 11 could be mounted.

[6] UK: Vodafone and T-mobile; Italy: Vodafone.

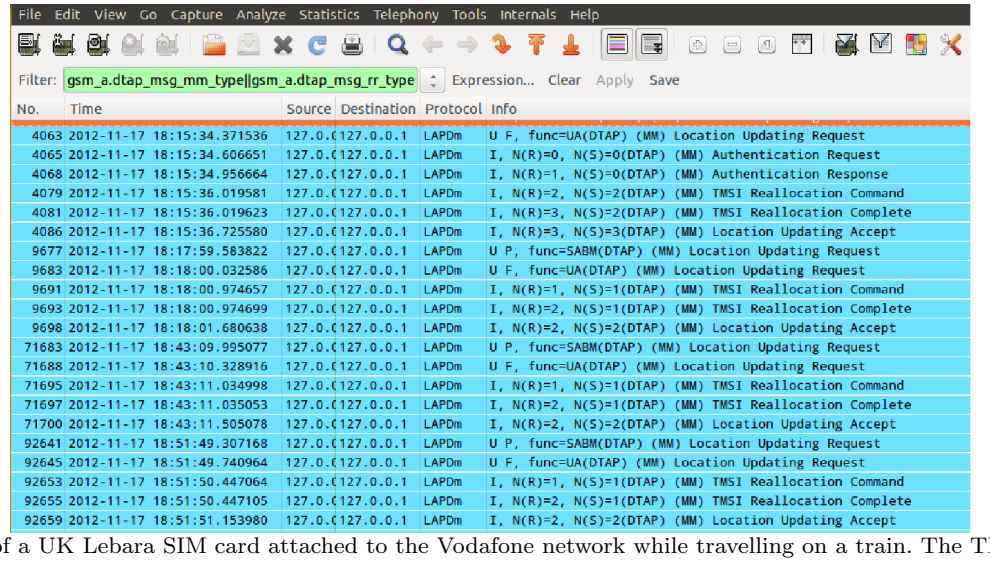

Fig. 9 Trace of a UK Lebara SIM card attached to the Vodafone network while travelling on a train. The TMSI reallocation procedure is executed by reusing a previously established key. The MS first performs a location update (packet no. 4063), then the authentication procedure to establish a ciphering key (packets 4065, 4068), followed by the TMSI reallocation procedure (packets 4079, 4081). The following three TMSI reallocations (packets 9691, 9693, 71695, 71697, 92653, 92655) are executed without first performing the authentication procedure and hence reusing the previously established ciphering key.

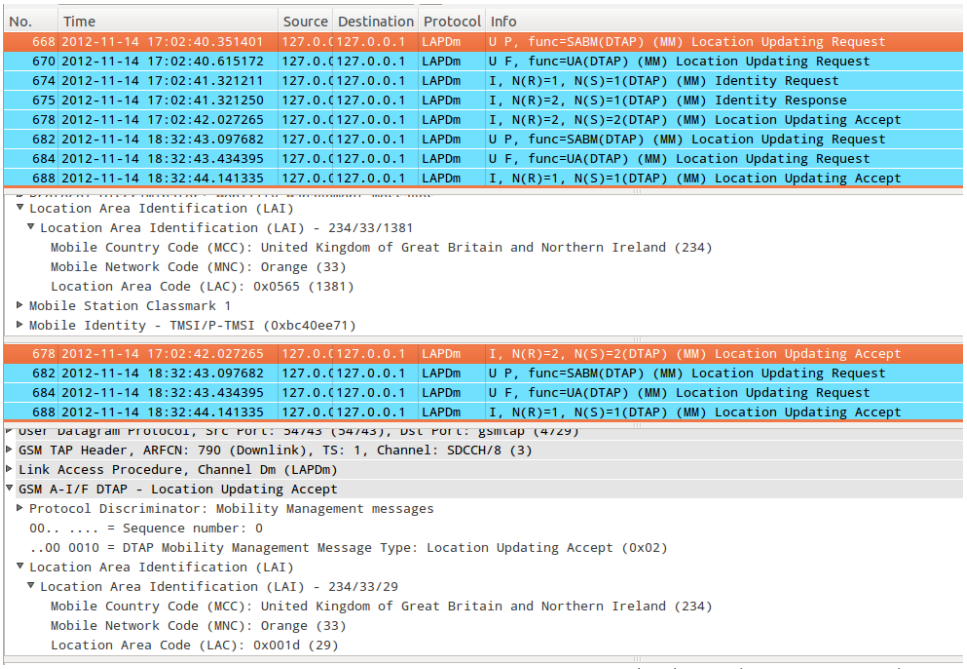

Fig. 10 Trace of a UK Orange SIM card. The TMSI used at location 234/33/1381 (packet no. 668) is accepted at location 234/33/29 (packet no.678), while the 3GPP standard mandates a TMSI reallocation at each change of location.

An attacker, controlling a radio device able to sniff and inject messages over-the-air, first captures a TMSI reallocation command (the second message in Figure 11). Later on, when the MS has possibly already changed its pseudonym but not yet established new keys, the attacker can replay the captured TMSI reallocation command (one message before last in Figure 11). Since reuse of the session key CK is allowed, the victim's MS successfully decrypts the reallocation message and sends the TMSI reallocation complete message. This allows the attacker to distinguish the victim's MS from any other that would not successfully decrypt the message and thus would not send any reply, even though in the meantime a different TMSI ($nTMSI_k$ in Figure 11) was assigned to the victim's MS.

Note that although the TMSI reallocation suffers from a replay attack, from a theoretical point of view, we have not investigated the feasibility of the attack in practice. In particular, it could be difficult to capture a legitimate TMSI reallocation command message to be replayed during the attack, since this message is encrypted. Moreover, the actual replay of the message is not trivial because the correct frame number should be in use by the MS when decoding it, in order to obtain the correct decryption. Further work would be needed to assess the practicality of this attack on both 2G and 3G networks.

3.10 Discussion

The experiments we conducted show how the adoption of pseudonyms is not a sufficient condition to ensure the privacy of mobile telephony users and that real network implementations leave plenty of room to tracking attacks. We suggest network operators should adopt activity related policies in order to prevent active tracking attacks. In general, the execution of the TMSI reallocation procedure should be more frequent even when the MS is in idle state, so to prevent mere passive tracking. Moreover, we suggest that a MS should be able to trigger the TMSI reallocation procedure in order to tackle active attacks exposing the TMSI without involving the network as for example directly injecting paging messages on the paging channel.

In particular, mobile network operators should enforce the policy of changing pseudonyms at each change of location area. This would reduce the probability of tracking across different LAIs. Furthermore, considering that Foo Kune et al. [**?**] are able to determine the presence of a target MS in an area by triggering the paging procedure on average 8 times, we suggest that this threshold should be used as maximum number of TMSI exposing messages a MS can send in clear over the air before a reallocation procedure is initiated by the network. This would lower the probability of successful tracking attacks. Moreover, as a countermeasure to simple passive tracking a reallocation procedure should periodically be triggered even in case the TMSI exposing activity has been under the suggested threshold. This would avoid the possibility of determining for example that a MS has not moved in a given time frame. We suggest to keep the time threshold fairly low. A sensible time threshold could be devised by estimating the time window for 8 TMSI exposing activity to occur when the MS is subject to average or low usage load.

Using pseudonyms is a good mechanism to ensure the user's privacy, provided that there is enough possibility of mixing within the network, which is usually the case in mobile telecommunication networks. However, the efficiency of the pseudonym change strategy depends on many factors which the 3GPP standard leaves as implementation choices.

We showed that the implementation choices made by real network operators do not provide a satisfactory level of privacy and leave space for different kinds of tracking attacks. Moreover, we showed that the loose standard specification can produce implementations of the TMSI reallocation procedure which are subject to a linkability attack.

The TMSI reallocation procedure is adopted in 3G+ networks as well. However, our experiments were conducted on the GSM network, and further work and specialized equipment would be needed so to investigate the usage scenario of TMSIs and TMSI reallocation in real 3G+ networks.

4 Analysis of Mobile Systems' Privacy: Pure 3G Linkability Attack

In the previous Section we showed how the paging procedure can be exploited to perform a simple linkability attack and we presented an experimental analysis of the identity management thanks to which we exposed the presence, in real mobile networks, of critical behaviour from a privacy point of view. Most of these privacy critical scenarios lead straightforwardly to breaches of the mobile telephony users' privacy. In this Section, we consider a 'Dolev Yao' attacker [**?**] who has full access only to the radio link, where he can sniff, inject, replay, and modify messages, but cannot break the cryptography involved in the protocol, *i.e.* cannot encrypt/decrypt without knowing the required encryption/decryption keys. In this Section we show a replay attack that allows to track the presence of a user in a monitored area or across a set of monitored areas. This attack concerns the 3G authentication and key agreement protocol.

4.1 3G Authentication and Key Agreement Protocol

The Authentication and Key Agreement (AKA) protocol achieves mutual authentication between a MS and the network, and establishes shared keys to be used to secure the subsequent communications.

The keys are not exchanged during the protocol but computed locally by the MS and the SN. According to [**?**], the authentication procedure is always initiated by the SN for the purpose of:

- Checking whether the identity provided by the MS is acceptable or not.

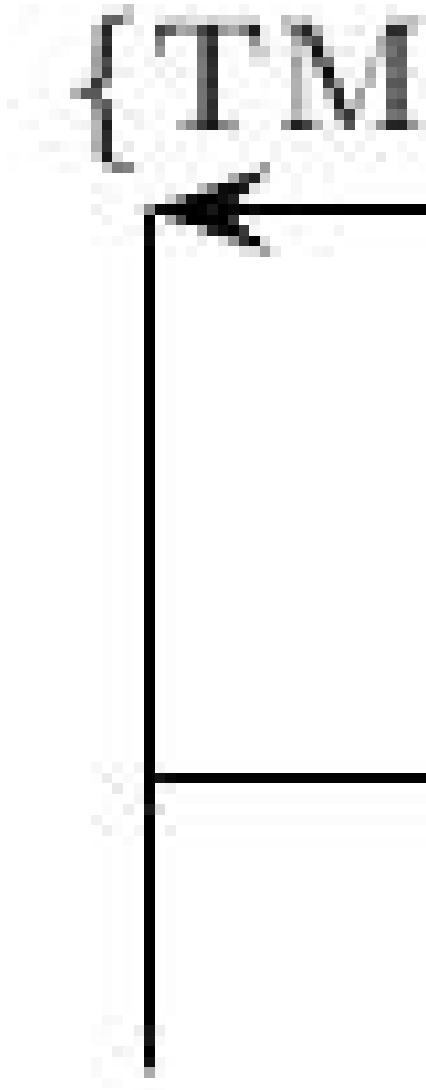

Fig. 11 TMSI Reallocation Procedure Attack

– Providing parameters enabling the MS to calculate a new UMTS ciphering key.
– Providing parameters enabling the MS to calculate a new UMTS integrity key.
– Allowing the MS to authenticate the network.

Each MS with identity *IMSI* and the network share a different 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.

A successful authentication procedure establishes a so-called *security context* which identifies the set of keys to be used for secrecy and integrity purposes; a Key Set Identifier (KSI) is used to retrieve the established security context through different sessions. Once a security context is established it is considered valid until the authentication procedure is next executed or the deletion of KSI is requested by the network.

The 3G AKA protocol [**?**], depicted in Figure 12, consists in the exchange of two messages: the authentication request and the authentication response. Before sending an authentication request to the MS, the network computes the authentication data: a fresh random challenge $RAND$, the authentication token $AUTN$, the expected authentication response $\mathrm{f2}_{K_{IMSI}}(RAND)$, the integrity key IK, and the encryption key CK (see Figure 12). 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}, see [**?**] for more details. 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

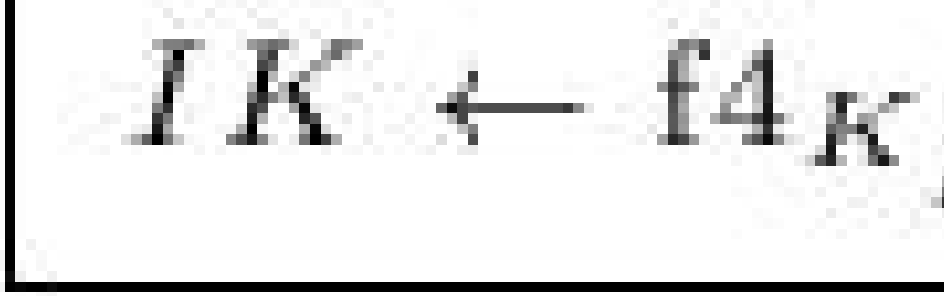

Fig. 12 3G Authentication and Key Agreement

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 [**?**]. The sequence number SQN_N allows the mobile station to verify the freshness of the authentication request to defend against replay attacks (see Figure 12).

The MS receives the authentication request, retrieves the sequence number SQN_N and then verifies the MAC (condition $MAC = XMAC$ in Figure 12). 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 12) to avoid replay attacks.

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 MS 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 authentica-

tion 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.

After successful authentication, the SN sends a security mode command message to the MS, indicating which one of the allowed algorithms to use for ciphering and integrity checking of the following communications.

4.2 3G 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 BSs 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 13. Note that this attack affects only 3G mobile systems, in fact 2G systems adopt a different authentication protocol which does not provide mutual authentication (i.e. the mobile station does not authenticate the network) and does not involve error and recovery procedures in case the authentication of the network fails. Thus, the distinguishing attack on the error messages cannot be performed in 2G networks.

5 Implementation of some 3G Protocols Attacks

In order to test the attacks presented in Section 3.4 and Section 4.2 in a deployed telecommunication network, we used 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 [**?**], 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.

5.1 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 [**?**] 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 [**?**] and led to the GAN specification [**?**,**?**] in 2005. This specification has been adopted and extended to be used in femtocell environments [**?**] . 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.

Fig. 13 AKA Protocol Linkability Attack

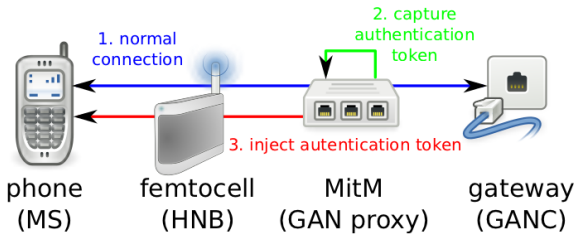

Fig. 14 Experimental Attack Setup

5.2 Attack Procedure

For the purpose of implementing our attacks (Section 3 and Section 4.2), we use a compromised femtocell like the one described in [**?**]. More specifically, we reproduce the hacking performed in [**?**] 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 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 informa-

tion 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 [**?**].

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 14. 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 15), then the MS on this dedicated channel is the victim's phone, and the victim is indeed in the femtocell area. Otherwise, if the response to the replayed authentication is a MAC failure (Figure 16), 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 [**?**]). 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 [**?**]. If devices with wider area coverage than a femtocell are used, the adversary should use triangulation to obtain finer position data.

6 Privacy Friendly Fixes

Despite the fact that mobile telephony systems adopting temporary identities to avoid linkability and to ensure anonymity of mobile telephony subscribers, active attackers can exploit the identification and the paging procedure to break both anonymity and unlinkability. Moreover, the TMSI reallocation procedure and the AKA protocol provide a way to trace mobile telephony users without the need to identify them in any way. These attacks on user privacy are not only theoretical but can be implemented in practice, as described in Section 3 and 5. Hence, the analysed procedures are a real threat for users' privacy, and countermeasures should be taken to provide an effectively privacy-friendly mobile telephony system. In this Section we propose some privacy-friendly solutions tackling the issues exposed in Sections 3, and 4. In Section 7 we show how to model and verify anonymity and unlinkability of the proposed fixed procedures using an automatic protocol verifier called `ProVerif` .

We propose a set of countermeasures involving symmetric and public key-based cryptography. The public key infrastructure we propose is easy to deploy because

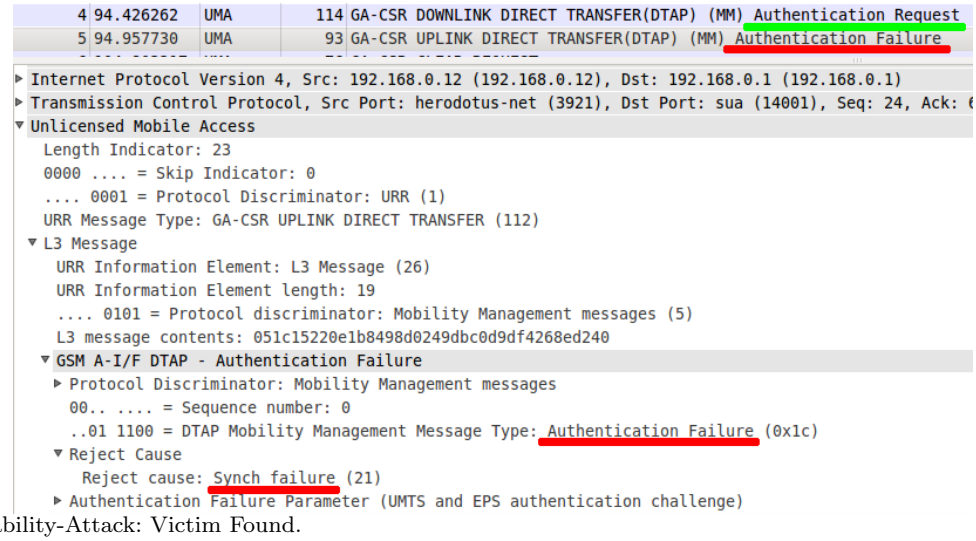

Fig. 15 Linkability-Attack: Victim Found.

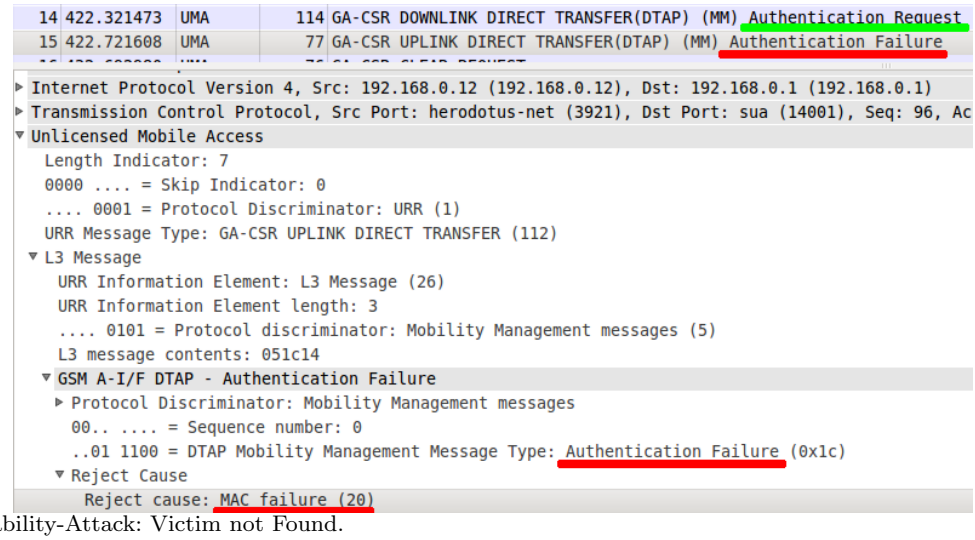

Fig. 16 Linkability-Attack: Victim not Found.

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 future generations mobile telephony systems. However, our solutions do not aim to be full protocol specifications. Their role is to show that realistic privacy-friendly solutions are possible without dramatic changes to the current infrastructure.

In this Section, we propose solutions to fix the identification procedure, the IMSI paging procedure, the TMSI reallocation procedure and the AKA protocol. Indeed, the problem of privacy is a multilayer/multi-protocol problem [**?**] which requires all protocols at all layers to satisfy the desired properties.

6.1 Public Key Infrastructure

We propose the adoption of a public key infrastructure (PKI) providing each MNO with a private/public key pair. The public key of a user's 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 PKI can also solve the problem exposed by Zhang and Fang in [**?**] concerning the lack of serving network authentication in the current 3G infrastructure.

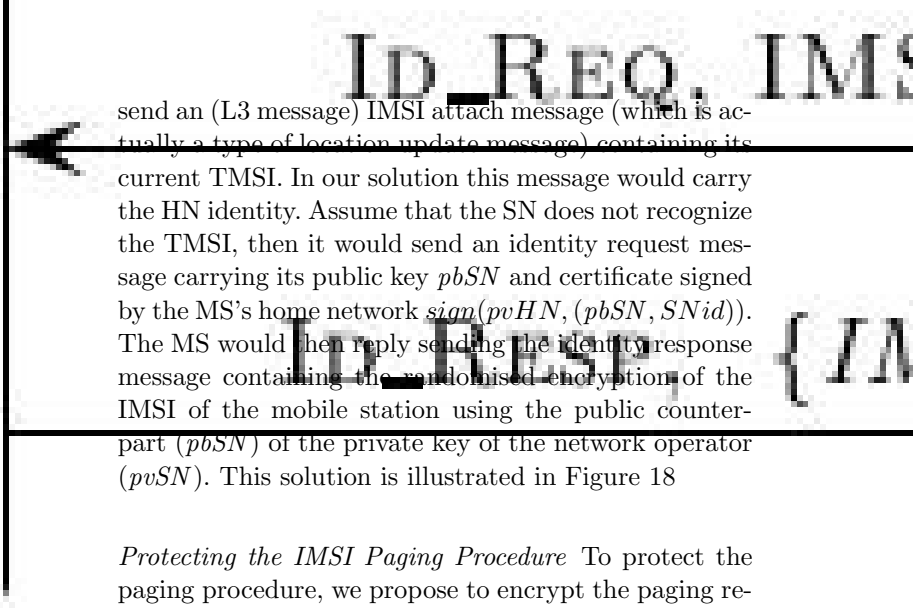

Fig. 17 Identification Procedure Fix. The identity response is encrypted with the public key of the network. The r denotes randomised encryption.

6.2 Fixes of the Analysed Procedures

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, as discussed in Section 2 the cost of devices allowing active attacks is constantly decreasing. An active attacker can simply trigger the identification procedure to retrieve the identity of all users in a monitored area [**?**]. We propose a solution to tackle this well-known attack (IMSI catcher attack). The fixed version of the identification procedure (Figure 17) 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). The identification procedure always happens on a dedicated channel and after the MS sent a first Layer 3 message containing its temporary identity TMSI. We require an MS attached to a network different from its HN to include the HN identity in the first L3 message sent when attaching to the SN. The SN can then send its public key and certificate signed by the mobile station's HN along with the identity request message. For example, an MS that has just been switched on would attach, by asking for a dedicated channel, to the (allowed) network offering the best signal and would then send an (L3 message) IMSI attach message (which is actually a type of location update message) containing its current TMSI. In our solution this message would carry the HN identity. Assume that the SN does not recognize the TMSI, then it would send an identity request message carrying its public key $pbSN$ and certificate signed by the MS's home network $sign(pvHN, (pbSN, SNid))$. The MS would then reply sending the identity response message containing the randomised encryption of the IMSI of the mobile station using the public counterpart ($pbSN$) of the private key of the network operator ($pvSN$). This solution is illustrated in Figure 18

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 19). 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 encrypted IMSI paging to determine if it is the recipient (Note that TMSI paging are still sent in cleartext). To enable the IMSI paging by a serving network, the encrypted paging request may be precomputed by the home network and IMSI paging vectors could be sent in bulk to the serving network the MS is attached to. An IMSI paging vector ($chall$, $rand$, $\{\text{PAGE}, IMSI, chall, SQN_N\}_{UK}$) would contain the challenge, the random, and the encrypted paging, in this way the unlinkability key UK and the sequence number SQN_N only have to be shared between the MS and the home network.

As an alternative solution, one could slightly relax the privacy requirements and allow a passive attacker to discover that the victim recently visited a certain location area. In this case, the IMSI paging request could be sent in clear over-the-air as it already is in the current systems. However, the paging response would be encrypted using the network's public key. This would

Fig. 18 Identification Procedure Fix when Roaming. The identity response is encrypted with the public key of the serving network. The r denotes randomised encryption.

eliminate the burden for the MS of decrypting all the received IMSI paging requests.

Fixing the TMSI reallocation procedure The solution we propose to fix the TMSI reallocation procedure does not require any change in the security architecture of mobile telephony systems. We only require the standard to specify that the reuse of the encryption key, CK is not permitted when the key is used to execute the TMSI reallocation procedure, i.e. the establishment of a new symmetric encryption key before the execution of the TMSI reallocation procedure should be mandatory. This would avoid the possibility of replay attacks to be mounted. However, frequent executions of the authentication procedure could burden the radio communication and slow down the delivery of mobile telephony services. Alternative solutions are possible, as for example the introduction of a sequence number in the TMSI reallocation command, similarly to the one used to avoid replay attacks against the Authentication and Key Agreement protocol [**?**]. We illustrate this solution in Figure 20. The network sends a sequence number SQN_{SN} along with the TMSI reallocation command. The MS checks if the received sequence number is in the expected range ($SQN_{MS} \leq SQN_{SN}$). If so it carries on with the reallocation of the TMSI. Otherwise the MS aborts the TMSI reallocation execution, hence avoiding replay attacks.

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 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.

Fig. 19 Paging Procedure Fix. The paging request is encrypted with the unlinkability key UK.

Moreover, the 3G standard stipulates [**?**] 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 over-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 21) carries on as specified by the standard. The network sends $RAND$, $AUTN$ and waits for a response. The response is $RES = \mathrm{f2}_{K_{IMSI}}(RAND)$, 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 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. Note that, as suggested in [**?**]

Fig. 20 TMSI Reallocation Procedure Fix: this fix uses the SQN to ensure the freshness of the reallocation command.

simpler solutions such as not sending any error message or sending a constant error message are also possible. However, these solutions do not allow the network to perform neither resynchronization nor any other sort of error recovery procedure since no information on the cause of the failure is given to it.

6.3 Discussion of the Proposed Fixes

While the fix we propose for the TMSI reallocation procedure and for the identification procedure are 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 specific procedures in order to perform error recovery depending on the occurred error).

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

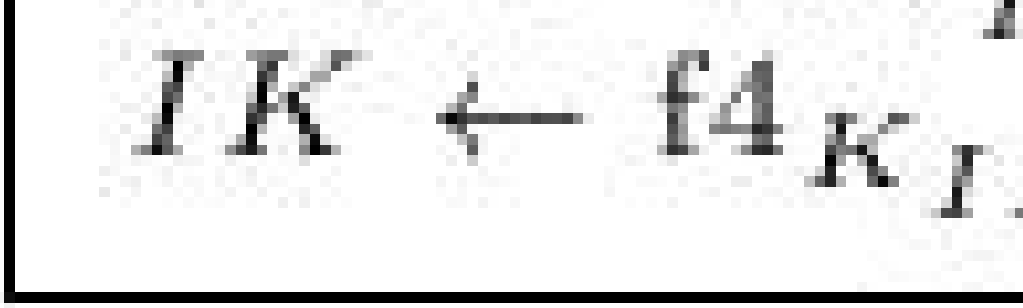

Fig. 21 The fixed AKA protocol. The error messages are encrypted using the network public key.

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 MSs. 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 life-time 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.

7 Automatic Verification of the Fixed Protocols

Many deployed protocols have subsequently been found to be flawed [?,?,?,?]. In this perspective and in order to increase the confidence one can have in the solutions proposed in Section 6, we show how to formally analyse our proposed fixes *w.r.t.* privacy. We use the formalisation of privacy-related properties as given by Arapinis et al. in [?], namely strong unlinkability and strong anonymity and adapt them to obtain definitions suitable for automatic verification using the `ProVerif` tool [?].

7.1 `ProVerif` Encoding

We use the automatic verification tool `ProVerif` [?] which takes in input processes written in a syntax similar to the applied pi-calculus [?] one. In the rest of this Section we use the `ProVerif` syntax (see Figure 22) to describe how we modelled the fixed protocols. However, we omit the full formal details for which we refer the reader to [?,?].

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 $f_1 \ldots, f_n$. 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.

```
fun senc/3, pub/1, aenc/3,
    f/2, f1/2, f2/2, f3/2, f4/2, f5/2.
reduc sdec(k, senc(k, m, r))= m.
reduc adec(k, aenc(\texttt{pub}(k),m,r))= m.
```

The functions `senc`, and `aenc` model randomised symmetric and asymmetric encryption respectively. In particular, the reduction rules allow to retrieve the plaintext `m` from its encryption given the knowledge of the decryption key `k`. The functions `f/2, f1/2, f2/2, f3/2, f4/2, f5/2` model the one way functions used in mobile telephony protocols.

`P, Q, R ::=`	plain processes
`0`	null process
`P | 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

Fig. 22 ProVerif syntax

We briefly describe the informal semantics of the calculus (formal details can be found in [?,?]. The null process does nothing. `P | 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 A system consisting of 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:

```
S = new pvN; let pbN = pub(pvN) in
      out(c, pbN);
      !new sk; new imsi; !new sqn; (SN|MS).
```

The privacy properties we verify are expressed in terms of *observational equivalence*. Intuitively, two processes `P` and `Q` are observationally equivalent denoted by $\mathtt{P} \approx \mathtt{Q}$, 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 bi-processes, which implies observational equivalence. Bi-processes 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(c, choice[a, b]). Diff-equivalence is stronger than observational equivalence this means that when ProVerif terminates producing an attack trace, the attack could be a false attack. However, when ProVerif terminates with a proof this means that, under the abstractions/assumptions made when modelling the protocols, there are no attacks to the verified property.

The encoding of the equivalence representing unlinkability is challenging since the processes to be tested do not have the same structure and because of this it is not straightforward to see how to build the biprocess representing them. In the next two Sections we show how we built the biprocess to test strong unlikability and the biprocess to test strong anonymity respectively.

7.2 Strong Unlinkability

In 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 process $\mathtt{S}_{\mathtt{UNLINK}}$ defined as follows:

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

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

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

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 fixed identification, paging, TMSI reallocation and AKA protocols as described in Section 6 (See table 1).

7.3 Strong Anonymity

In our mobile phone scenario, strong anonymity requires a system in which a mobile station $\mathtt{MS}_V$ with publicly known identity $\mathtt{imsi}_\mathtt{V}$ executes the protocol to be indistinguishable from a system in which the $\mathtt{MS}_V$ is not present at all. Such a system obviously preserves $\mathtt{imsi}_\mathtt{V}$'s anonymity. Formally, we want the system S, defined as in Example 2 to be observationally equivalent to the system $\mathtt{S}_V$ defined as follows:

$$\begin{aligned}S_V = \;& \mathtt{new\ pvN}; \mathtt{letpbN} = \mathtt{pub(pvN)in}\\ & \mathtt{out(c, pbN)};\\ & !\mathtt{new\ sk}; \mathtt{new\ imsi}; (!\mathtt{new\ sqn}; (\mathtt{SN}|\mathtt{MS}))\\ & |\mathtt{new\ sk}; !\mathtt{new\ sqn}; (\mathtt{SN}|\mathtt{MS}_\mathtt{V}).\end{aligned}$$

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

```
free imsi_v.

let PV_ANON =  new pvN; let pbN = pub(pvN) in
    out(c, pbN);
    (!new sk; new imsi;
        (!new sqn; (MS | SN)))
    |(new sk; new imsi_ms;
        let  imsi=choice[imsi_V, imsi_ms ] in
        !new sqn; (SN | MS)).
```

The left side of the choice represents a system where the mobile station with public identity $\mathtt{imsi}_\mathtt{V}$ can run the protocol. Our fixes of the identification procedure, paging procedure, TMSI reallocation and 3G AKA protocol as described in Section 6 are proved by ProVerif to satisfy anonymity (See table 1).

7.4 Automatic Verification Results and Remarks

We run the ProVerif tool on the 2G/3G identification procedure, on the 2G/3G IMSI paging procedure, on the 2G/3G TMSI reallocation procedure and on the 3G AKA protocol, in order to confirm that the tool would have detected the breaches of the privacy properties present in the 3GPP standard procedures. Even though the coding of the protocols in ProVerif is

Properties	Identification	IMSI Paging	TMSI reallocation	3G AKA
Unlinkability	√	√	√	√
Anonymity	√	√	√	√

NA Not Applicable √ Proved to hold × Attack found

Table 1 ProVerif results of the on Fixed Procedures

Properties	Identification	IMSI Paging	TMSI reallocation	3G AKA
Unlinkability	×	×	×	×
Anonymity	×	×	FA	√

NA Not Applicable √ Proved to hold × Attack found FA False Attack

Table 2 ProVerif results on original 3GPP Procedures

straightforward, the coding of the observational equivalences defining the privacy properties in term of biprocesses is not. In fact the biprocess structure is symmetrical while the definitions of anonymity and unlinkability are not. We showed in the previous Section how we obtained a symmetrical definition in terms of biprocesses. Moreover, we had to take 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 ($\approx_{\texttt{diff}}$). In particular, diff-equivalence can distinguish between the execution of different branches of a conditional statement even in the following case:

```
if a = a then P else P
        ≈diff
if a = b then P else P
```

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 1-2, 10, and 15-16 of the code of the original AKA protocol in Table 7.4. We check the MAC and the sequence number (line 10) 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 1-2) to determine the error message (lines 15-16) and avoid the use of an if statement.

As expected, the verification with the `ProVerif` tool fails to prove the anonymity of the IMSI paging procedure and gives a false attack when verifying the anonymity of the TMSI reallocation procedure. Unlinkability is not satisfied by the original IMSI paging, TMSI reallocation and 3G AKA protocols (see Table 2). In case of the IMSI paging procedure `ProVerif` exhibits actual attack traces. In the case of the original 3G AKA pro-

Fig. 23 code of the MS side of the original AKA protocol

```
1 reduc geterr(err(x,z,y,y))=macFail;
2       geterr(err(x,x,y,z))=synchFail.
3
4 let AKA_MS = new r_ms; in(c, x);
5   let (xrand, xautn) = x in (
6   let (msg, xmac) = xautn in (
7   let ak = f5(k, xrand) in (
8   let xsqn = sdec(ak, msg) in (
9   let mac = f1(k, (xrand, xsqn)) in (
10   if (xmac, xsqn) = (mac,sqn) then (
11      let res = f2(k, xrand) in (
12      let ck = f3(k, xrand) in (
13      let ik = f4(k, xrand) in (
14      out(c, res); in(c, xmsg)))))
15   else (let err_msg =
16         geterr(err(mac, xmac, sqn, xsqn)) in
17         out(c, err_msg))))))).
```

tocol, the anonymity property is proved to hold, while the verification of the unlinkability property 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 modelling of unlinkability and anonymity into diff-equivalences we showed in the previous 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. The `ProVerif` code used for the automatic verification is available online [**?**] and in part in Appendix.

Note that for verification purposes in our models of MS and SN 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.

The model of the TMSI reallocation procedure requires the use of a state (memory cell) where to store the newly assigned TMSI to be used in the following session. The state is initialized during an initialization phase. We use a private channel to model the state. An auxiliary process (`MEM`) makes the content of the state available to read and write from the channel. In particular, the process used for the verification of the unlinkability property of the fixed procedures in presence of the memory cell is coded as follows.

```
let MEM = in(mem,x);out(mem,x).

process new pvN;
  let pbN = pub(pvN) in out(c, pbN);
  (! (new sk1; new imsi1;new otmsi1; new mem;

 (* memory initialisation *)
     out(mem,otmsi1);

 (! (new sk2; new imsi2; new osqn;
     new otmsi2; new sqn_p;
     let imsi = choice[imsi1, imsi2] in (
     let k = choice[sk1, sk2] in (
     let otmsi = choice[otmsi1,otmsi2] in (
        (MS) | (SN) | (MEM))))))))
```

This abstraction may produce false attacks and non termination problems (see [**?**]). For this reason the encoding of the TMSI reallocation procedure is not the one you would obtain with a straightforward encoding of the description of the protocol in applied pi-calculus. In particular, the TMSI of a single session mobile stations is created during the i-th session of the multi-session mobile station and used in the $i+1$-th session (see lines 105, 108, 156 of the code in Appendix. Intuitively, this gives the TMSI used by the single session mobile station and the one used by the multi-session one the same amount of freshness in the `ProVerif` model. Hence, the position of the new construct at lines 105 and 156 in the code in Appendix is critical to the success of the verification process.

Authentication, Secrecy, Integrity. The main purpose of the 3G 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 3G AKA protocol are preserved by our fixes; the verification results are shown in Table 3.

The full code used for the verification of the security and privacy of the 3GPP procedures analysed in this paper when run in isolation and when run in parallel is available on-line [**?**]. In the Appendix we report the code used to verify the unlinkability of all the fixed procedures when running in parallel.

8 Conclusions

In this work, we presented a thorough (although not complete) experimental and formal analysis of users' privacy in mobile telephony systems. In particular, we experimentally analysed the use of pseudonyms and pointed out weaknesses in the deployed policies. We showed how these weak policies make it possible to violate a user's privacy and proposed some solutions to strengthen the pseudonyms management in mobile telephony systems. We also exposed some protocol's vulnerabilities resulting in breaches of the anonymity and/or user unlinkability. We showed with a prototype implementation that the breaches we found translate into actual attacks on real networks. To countermeasure these attacks, we proposed solutions which are realistic and lightweight but require some changes to the existing infrastructure. The fixes we proposed show that privacy-friendly solutions could be adopted by future generation of mobile telephony systems. However, these are not meant to be full protocol specifications and implementation details are left for the mobile telephony operators to specify. Finally, we provided a theoretical framework for the automatic verification of the unlinkability and anonymity using the `ProVerif` tool. We used this framework to automatically verify the fixed 2G/3G procedures. Further investigation of mobile telephony users' privacy should be undertaken in order to obtain a complete picture of the privacy offered by the interactions of the many protocols at different level of the stack. Currently, our demonstration of the attacks presented in Section 5 relies on particular hardware/software using closed source implementation of the 3G protocol stack and radio signalling functions. It would be interesting and beneficial for further research in the area of mobile telephony systems to investigate the possibility of implementing open source testing equipment, such as a 3G base station and mobile phone, using low cost hardware, e.g. USRP, and the GNU radio software. Further work is needed in order to confirm experimentally the replay attack presented in Section 3.9. This would allow to check if there are

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

NA Not Applicable √ Proved to hold × Attack found

Table 3 Results of the Automatic Verification of the Fixed Procedures

or not mechanisms in place (not stated in the standard) to thwart this attack by preventing replayed messages from being accepted by the Mobile Station. Also, a thorough and methodical analysis of the level of privacy achieved by different privacy policies would be of great interest. However, this would possibly require collecting further data about user mobility, aggregation areas, population density, network coverage and user base per geographical area. The impact of the adoption of the proposed TMSI reallocation policies on the network performances should be studied and related to the level of achieved user's privacy in order to carefully balance these equally important aspects of mobile telephony systems. The overall privacy of mobile telephony systems and at each layer of the protocol stack requires further investigation and would possibly offer interesting challenges for the development of formal methods as well. In particular, the proliferation of location based services makes the analysis of application layer privacy highly desirable from a user's point of view.

Acknowledgements We would like to thank Kevin Redon, Nico Golde and Ravi Borgaonkar for the proof of concept implementation of the attacks presented in this paper. We are also grateful to Chris J. Mitchell for lively discussion about our ideas on mobile telephony protocols.

APPENDIX: ProVerif code

`ProVerif` code used to verify the unlinkability of the fixed procedures when running in parallel. Note that the fixed TMSI reallocation procedure, executes the AKA protocol prior to the reallocation so to obtain a fresh encryption key.

```
1  (* public communication channel *)
2  free c.
3
4  (* constant values *)
5  fun Fail/0.
6  fun reject/0.
7  fun page/0.
8  fun pagingReq/0.
9  fun pagingResp/0.
10 fun imsiReq/0.
11
12 (* UMTS AKA protocol specific mac and
      key generation functions *)
13 fun f1/2.
14 fun f2/2.
15 fun f3/2.
16 fun f4/2.
17 fun f5/2.
18
19 (* New key generation function *)
20 fun f/2.
21
22 (* symmetric key encryption function *)
23 fun senc/3.
24 fun sdec/2.
25 equation sdec(k, senc(k, r, m)) = m.
26
27 (* public key generation function *)
28 fun pub/1.
29
30 (* public key encryption function *)
31 fun aenc/3.
32 reduc adec(k, aenc(pub(k), r, m))= m.
33
34 let AKA_MS =
35    new r_ms;
36    in(c, x);
37    let (xrand, xautn) = x in (
38    let (msg, xmac) = xautn in (
39    let ak = f5(k, xrand) in (
40    let xsqn = sdec(ak, msg) in (
41    let mac = f1(k, (xrand, xsqn)) in (
42    if (xmac, xsqn) = (mac, osqn) then (
43       let res = f2(k, xrand) in (
44       let ck = f3(k, xrand) in (
45       let ik = f4(k, xrand) in (
46       out(c, res);
47       in(c, xmsg)))))
```

```
48    else (
49       out(c, aenc(pbN, r_ms, (Fail, imsi, osqn)
      ))))))).
50
51 let AKA_SN =
52    new rand;
53    new r_sn;
54    new s;
55    new r;
56    let mac = f1(k, (rand, osqn)) in (
57    let res = f2(k, rand) in (
58    let ck = f3(k, rand) in (
59    let ik = f4(k, rand) in (
60    let ak = f5(k, rand) in (
61    let autn = (senc(ak, r_sn, osqn), mac) in (
62    let av = (rand, res, ck, ik, ak) in (
63    out(c, (rand, autn));
64    in(c, xres);
65    if xres = res then (
66       out(c, senc(ck, r, s))
67    )
68    else (
69       out(c, reject)))))))))).
70
71 let ID_MS =
72    new r;
73    in(c, req);
74    if req = imsiReq then (
75       out(c, aenc(pbN, r, imsi))).
76
77 let ID_SN =
78    out(c, imsiReq);
79    in(c, ximsi).
80
81 let PAGING_MS =
82    in(c, x);
83    let (msgtype, xrand, xblob) = x in (
84    if msgtype = pagingReq then (
85      let (xpage, ximsi, =sqn_p, xchall) =
            sdec(f(k, xrand), xblob) in (
86      if xpage = page then (
87          if imsi = ximsi then (
88      out(c, (pagingResp, xchall)))))))).
89
90 let PAGING_SN =
91    new rand;
92    new chall;
93    new r_sn1;
94    let UK = f(k, rand) in (
95    out(c, (pagingReq, rand, senc(UK,
         r_sn1, (page, imsi, sqn_p, chall))));
96    in(c, pres)).
97
98 let TMSI_MS =
99   in(mem, xmem);
100   new mr;
101   out(c, xmem);
102   in(c, y);
103   let (=TMSI_REALL, yid) = sdec(ck, y) in
104  (out(c, senc(ck, mr, COMPLETE));
105 out(mem,choice[yid, otmsi2])).
106
107 let TMSI_SN =
108    new nid;
109    new sr;
110    in(c, z);
111    out(c, senc(ck, sr,
                 (TMSI_REALL, nid)));
112    in(c, w).
113
114 let FixedTMSI_MS =
115    new r_ms;
116    in(c, x);
117    let (xrand, xautn) = x in (
118    let (msg, xmac) = xautn in (
119    let ak = f5(k, xrand) in (
120    let xsqn = sdec(ak, msg) in (
121    let mac = f1(k, (xrand, xsqn)) in (
122    if (xmac, xsqn) = (mac, osqn) then (
123       let res = f2(k, xrand) in (
124       let ck = f3(k, xrand) in (
125       let ik = f4(k, xrand) in (
126       out(c, res);
127       TMSI_MS))))
128    else (
129       out(c, aenc(pbN, r_ms, (Fail, imsi,
                osqn)))))))))).
130
131 let FixedTMSI_SN =
132    new rand;
133    new r_sn;
134    new s;
135    new r;
136    let mac = f1(k, (rand, osqn)) in (
137    let res = f2(k, rand) in (
138    let ck = f3(k, rand) in (
139    let ik = f4(k, rand) in (
140    let ak = f5(k, rand) in (
141    let autn = (senc(ak, r_sn, osqn),
                          mac) in (
142    let av = (rand, res, ck, ik, ak) in (
143    out(c, (rand, autn));
144    in(c, xres);
145    if xres = res then (
146       TMSI_SN)
147    else (
```

```
148       out(c, reject))))))))).
149
150 let MS = (AKA_MS|ID_MS|PAGING_MS|FixedTMSI_MS).
151 let SN = (AKA_SN|ID_SN|PAGING_SN|FixedTMSI_SN).
152 let MEM = in(mem,x);out(mem,x).
153
154 process new pvN; let pbN = pub(pvN) in
                    out(c, pbN);
155 (! (new sk1; new imsi1;new otmsi1; new mem;
     out(mem,otmsi1);
156 (! (new sk2; new imsi2; new osqn;
    new otmsi2; new sqn_p;
157 let imsi = choice[imsi1, imsi2] in (
158 let k = choice[sk1, sk2] in (
159 let otmsi = choice[otmsi1,otmsi2] in (
160 (MS) | (SN) | (MEM))))))))
```