Academic year 2011/2012
Progress Report 6

The University of Birmingham
School of Computer Science

Research Student Monitoring Group

Progress Report 6

Working title:
Security Protocols Verification in Pervasive Systems

Date of report:
20th October 2012

Student:
Loretta Mancini

Supervisor:
Eike Ritter

Co-Supervisor:
Myrto Arapinis

Thesis Group Members:
Mark G. Lee, Mark D. Ryan
Contents

1 Introduction 3

2 Summary of work done so far 3
   2.1 3G Privacy Analysis ................................................. 4
   2.2 Further Analysis of Mobile Phones Protocols ...................... 5
      2.2.1 TMSI Reallocation Procedure ................................. 6
      2.2.2 Formal Analysis ............................................... 6

3 Changes to the Plan of Research 8

4 Draft chapter by chapter summary of planned thesis 8

5 Statement 9

6 Talks, Papers, etc. 9

7 Timetable 9
1 Introduction

The aim of our research is to formally and automatically verify security properties of protocols used in pervasive systems, with particular emphasis on privacy-related properties such as strong and weak unlinkability and strong and weak anonymity. The verification of privacy properties of pervasive systems’ protocols poses often new challenges to the current state of the art tools for the verification of security protocols because it involves an unbounded number of agents and sessions executed in parallel. Moreover, privacy is a multi-layered and multi-protocol problem and is often in conflict with other security goals as for example authentication. To model pervasive systems protocol and in particular protocols implementing privacy friendly mechanisms is often required to take into account the state of the agents involved in the analysed protocol, for example to keep track of a newly assigned pseudonym (or temporary identity). Although, steps have been made towards the verification of state-full processes [2], to date this problem remains a challenging one in particular when dealing with observational-equivalence-based properties and possibly infinite states, as in our case.

2 Summary of work done so far

We started our analysis from a set of formal definitions of privacy-related properties, namely strong/weak unlinkability and strong/weak anonymity given in [3] by Arapinis et al. aiming to translate/adapt them so to make it possible to verify these properties using automatic verification tools. Moreover, we identified real world case studies, serving the purpose of inspiring our analysis of privacy properties and demonstrating the efficacy and/or limitations of our formal techniques when applied to real world protocols.

We found out that, for stateless protocols which do not require an initialization phase, it is possible to translate strong unlinkability and strong anonymity, which are defined as observational equivalence properties, into diff-equivalence, that is, a stronger equivalence relation which is automatically verified by the automatic verification tool ProVerif [4]. The result is a quite simple process structure in form of a bi-process (a pair of processes differing by some messages). We used this biprocess structure to analyse a set of mobile telephony protocols not requiring the encoding of states and with no initialization phase.

On the contrary, the analysis of privacy properties for state-full processes cannot be handled by the currently available verification tools. For this reason, we were not able to model and automatically verify with the ProVerif tool the TMSI reallocation procedure. The TMSI reallocation procedure is used in mobile systems to assign new temporary identities to mobile phones,
i.e. it is the main mechanism mobile system rely on to ensure mobile phones unlikeliness and anonymity.

The analysis of this procedure would complete our study of the 3G protocols privacy at the mobility management layer. Moreover, similar verification issues arise in the verification of other protocols involving the use of pseudonyms like for example the ones proposed for the privacy in VANET’s (Vehicular Ad hoc Networks) Hence, we decided to analyse the TMSI reallocation procedure using the applied pi-calculus and manual proof techniques aiming to gain some insight on how to extend automatic verification tools for the analysis of privacy properties for state-full protocols more in general.

We modelled and we developed a proof by hand of the unlinkability of the TMSI reallocation procedure. This analysis allowed us to point out possible attack traces arising in case the protocol is not carefully implemented and hence to disambiguate the standard specification by proving that the desired unlinkability property is satisfied by one of the possible implementations of the protocol. We started our analysis proceeding by baby steps and adding complexity little by little, this process showed us how big is the impact of little changes on the structure of the bisimulation relation and made us understand how far we are from a fully general automation of the verification process. For this reason, we decided to approach the problem by bounding the number of states and adopting stronger definition for our privacy properties, this will allow us to work on an extension of the StatVerif[2] tool (an extension of the ProVerif tool for state-full processes) for the verification of observational equivalence and hence of a stronger version of unlinkability and anonymity.

2.1 3G Privacy Analysis

We used the strong unlinkability and anonymity definitions given by Arapinis et al. [3], to analyse a set of 3G (Third Generation telecommunication systems) protocols: Identification Procedure, Authentication and Key Agreement (AKA) Protocol and IMSI Paging Procedure. We found and implemented on real 3G networks two possible attacks to the user privacy. One, the IMSI paging procedure attack, assumes the adversary knows the long term identity IMSI (International Mobile Station Identity) of a mobile station and forces it to emit its temporary identity TMSI (Temporary Mobile Station Identity) and at the same time to reveal its presence in a monitored area. The second attack concerns the Authentication and Key Agreement protocol (AKA) and allows to link two or more session of the same mobile station and his presence in a monitored area, without any previous knowledge of its temporary or long term identifier. We proposed countermeasures to the attacks we found and to the identification procedure. The identification procedure is a well known breach to the user privacy as it reveals the long term user identity in plaintext. For this reason the standard requires
network operators to limit the use of this procedure. However, if limiting the use of this procedure was considered a good enough countermeasure at the time the 3G protocols were designed, it is not today, because the currently available technologies allow attackers to impersonate legitimate network operator’s base station and initiate all sort of attacks as the two we found by protocol inspection or for example simply triggering the identification procedure to reveal a user long term identity. For this reason, our analysis would have not been complete without a privacy friendly solution fixing this procedure. We automatically verified, using the ProVerif tool, that the proposed solutions satisfy unlinkability and anonymity while preserving the properties of the original procedures (namely, secrecy, authentication and integrity).

Moreover, we collaborated with Nico Golde, Kevin Redon and Ravi Bor- 

Moreover, we collaborated with Nico Golde, Kevin Redon and Ravi Bor-

gaonkar (Technical University of Berlin), whom hacked a short range 3G base station called femtocell [6], to obtain a real implementation and hence to confirm the attacks we found by inspection of the standards and formal analysis. As result of the above mentioned investigation of 3G protocols we produced a paper which was accepted at the ACM CCS 2012 conference [1].

2.2 Further Analysis of Mobile Phones Protocols

We are currently working on the formal analysis of another 3G procedure used to provide user anonymity and unlinkability: the TMSI (Temporary Mobile Subscriber Identity) reallocation procedure. The formalization of this procedure requires the modelling of state-full processes. The presence of (possibly infinitely) many processes and states makes the formal analysis of this procedure challenging and not suitable for the automatic verification using the ProVerif tool (or any other currently available tool). Moreover, when modelling the TMSI reallocation procedure, we found out that the 3G standard specification is not rigorous enough and allows implementations leading to attacks to the subscriber’s privacy. For this reasons we are both running some experiments to check real networks implementations and working on a manual proof of the TMSI reallocation procedure. Moreover, we are collecting some data about the behaviour of different UK mobile network operators, to enrich the formal analysis of the protocol with some statistical data concerning the way of using long term and temporary identity by network operator and handsets under different circumstances. The aim is to show that the measures taken to ensure user’s privacy are in general not effective since the temporary identities are not updated often enough and the long term identity is exposed too often.
2.2.1 TMSI Reallocation Procedure

Privacy properties of mobile telephony systems rely on the use of temporary identities to ensure subscribers anonymity. The mobile phone temporary identity is contained in the first message sent to the network after a physical layer connection has been established. Temporary identities are periodically assigned by the network to avoid long-term linkability of subscribers activity. The assignment of a new temporary identity is obtained through the execution of the TMSI reallocation procedure. The messages exchanged during the TMSI reallocation procedure (Figure 1) are encrypted so to avoid eavesdropping of the new identity and hence linkability between the old and the new identity. The key used for encryption is a temporary shared key established executing the authentication and key agreement protocol.

2.2.2 Formal Analysis

To complement our previous analysis we started to model the TMSI reallocation procedure with the purpose of studying its privacy properties. This procedure requires to model a memory for the storage of the temporary identities and requires an initialization phase to initialize the memory. We tried to verify different variation of ProVerif models of this procedure but none of our attempts to automatically verify this procedure were successful because of the limitation of the ProVerif tool. We hence modelled the procedure in applied pi calculus so to produce a manual proof of the unlinkability property.

We soon realized that if not carefully implemented the TMSI reallocation procedure would allow linkability attacks to be performed. We found a
possible attack trace. An attacker could capture and then replay a TMSI reallocation command previously sent to a victim phone, and wait for a reply. The reply indicates that the mobile phone accepted the reallocation command and hence the re-assigned TMSI, which in turn means that it was able to decrypt the replayed message using its session key \( CK_{IMSI} \). Thus the attacker can deduce that the replying phone is the victim one (Figure 2).

The described attack is possible only if the network and the victim phone have not established a new shared session key \( CK_{IMSI} \) (by executing the AKA protocol).

Currently, we are polishing the proof of unlinkability of the TMSI reallocation procedure when executed using fresh session keys at each session and we are running some experiments to monitor how the procedure is implemented in different real networks. For this purpose we use Motorola C115 phones and the Osmocom-BB [5] software, which allows us to capture mobile phones communications with the network and visualize the exchanged messages using the "Wireshark" [?] network protocol analyser. This allows us to record how many times TMSI reallocation and AKA protocol are executed and if the AKA protocol is always executed at least once between two TMSI reallocation updates. Moreover, we will extract statistics on the usage of long term and temporary identities under a variety of mobile phone usage scenarios.
3 Changes to the Plan of Research

Aiming to prove the unlinkability of the TMSI reallocation procedure, we produced several proofs of unlinkability of toy examples. In particular, unlinkability is expressed as observational equivalence but the definition of observational equivalence is not easy to handle when working on manual proofs for this reason we instead proved labelled bisimilarity. The process of going step by step towards the proof of a good approximation of the TMSI reallocation procedure showed us that little changes in the process can result in dramatic changes in the structure of the bisimulation relation, making it difficult to automate the process. For this reason it seems we are still far from understanding how to full automate unlinkability and anonymity of state-full processes in the general case of infinite states and unbounded processes. Hence, we decided to start by bounding the number of state since this will allow us to work on an extension of the automated tool for the verification of protocol with a bounded number of states: StatVerif. The following step of our analysis of privacy related properties will focus on the extension of StatVerif to the verification of observational equivalence.

4 Draft chapter by chapter summary of planned thesis

1. Introduction
   - The research problem
   - Motivation
   - Contribution

2. Background
   - Applied pi-Calculus
   - ProVerif and Automatic Verification
   - Formalization of Privacy Properties

3. Verification of Strong Privacy Properties
   - Formalization of Strong Privacy Properties in ProVerif
   - 3G Protocol Case Study
   - Related Work

4. Verification of Strong Privacy Properties for State-full Protocols
   - A Case for Manual Proof
   - TMSI reallocation Case Study
• Related Work

5. Automatic Verification of Observational Equivalence for State-full Protocols
   • StatVerif Extension
   • Case Study
   • Related Work

6. Open Problems

7. Conclusion

5 Statement

I have read and bookmarked the School’s current documents giving advice on writing up, presenting and submitting theses, as well as the University’s on submitting theses.

Signature
Date

6 Talks, Papers, etc.


7 Timetable

T1 CCS12 Conference attendance 15-20 Oct. 12
T2 RSMG Report 6 by 20 Oct. 12
T3 RSMG Meeting to be hold by 7th Nov. 12
T4 TMSI Reallocation procedure paper submission to IEEE Security and Privacy 13, 14th Nov. 12
T5 Extension of StatVerif middle Feb. 13
T6 Use StatVerif extension for the verification of a case study and write a paper, end Mar 13

T7 Report 7, 17th Apr 13

T8 Writing up end o June 13

<table>
<thead>
<tr>
<th></th>
<th>Oct 12</th>
<th>Nov 12</th>
<th>Dec 12</th>
<th>Jan 13</th>
<th>Feb 13</th>
<th>Mar 13</th>
<th>Apr 13</th>
<th>May 13</th>
<th>June 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
References


New Privacy Issues in Mobile Telephony: Fix and Verification

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

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

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

Categories and Subject Descriptors
D.2.4 [Software Program Verification]: Formal Methods

General Terms
Security, Verification

Keywords
Unlinkability, Anonymity, ProVerif, Mobile Telephony

1. INTRODUCTION

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

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

Hence, we believe active attackers should now be considered when analysing mobile systems in order to obtain convincing and reliable results on their security. From this perspective we present a formal analysis of the 3G subscribers privacy. We expose two novel threats and we demonstrate that these threats can lead to real implementations which make use of cheap equipment. Furthermore, we propose privacy friendly fixes to thwart the detected privacy issues and we formally verify that our fixes achieve the desired privacy goals.
Our Contributions. Linkability of transactions has been identified and often reported by the media as an important threat for user privacy [14, 17, 24] though it has been overlooked so far by most of the existing studies of mobile telecommunication protocols which instead focus on confidentiality and authentication requirements (Section 2.2). In this paper, we present the first formal analysis of 3G protocols w.r.t. privacy of mobile phone users from third party attackers and in particular w.r.t. unlinkability and anonymity of 3G subscribers. For our analysis we use automated formal methods. The use of formal methods allows us to: (i) precisely and unambiguously define the desired privacy properties in terms of third-party strong anonymity and strongunlinkability; (ii) identify new vulnerabilities with respect to subscriber privacy thanks to a rigorous specification of the protocols and of the analysed properties.

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

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

2. BACKGROUND AND RELATED WORK

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

2.1 3G Security Requirements

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

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

User confidentiality: the property that the permanent user identity (IMSI) of a user to whom a service is delivered cannot be eavesdropped on the radio access link;
the false base station attack and takes advantage of the fact that the mobile station does not authenticate the serving network. It allows the redirection of the victim’s outgoing traffic to a different network, for example a network which uses a weaker encryption algorithm or one which charges higher rates than the victim’s one. Zhang and Fang’s attack concerns impersonation, service theft and data confidentiality, while our work exhibits privacy issues arising in 3G.

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

**Previous Formal Analysis** The 3G AKA protocol in its pure form (i.e. with no GSM support) has been formally proved to meet some of the specified security requirements [5], such as authentication and confidentiality of data and voice communication. However, privacy related properties such as unlinkability and anonymity, which are the focus of our work, are not analysed in [3]. The framework applied in [3] cannot be used to specify unlinkability and anonymity properties, let alone reason about them. The formal framework used in our paper allows us to precisely define and verify privacy related properties. Hence, we can discover privacy attacks on the modelled protocols and propose solutions which are formally proved to satisfy the desired privacy properties.

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

3. **NOVEL PRIVACY THREATS**

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

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

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

3.1 **IMSI Paging Attack**

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

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

3.2 **AKA Protocol Linkability Attack**

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

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

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, generated by the network using an individual counter for each subscriber. A new sequence number is generated either by increment of the counter or through time based algorithms as defined in [6]. The sequence number SQN, allows the mobile station to verify the freshness of the authentication request to defend against replay attacks (see Figure 2).

The MS receives the authentication request, retrieves the sequence number SQN, and then verifies the MAC (condition MAC = XMAC in Figure 2). This step ensures that the MAC was generated by the network using the shared key KIMSI, 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. This value is used to check the freshness of the authentication request (condition XSQN < SQN in Figure 2) 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 mobile station by verifying whether the received response is equal to the expected one (RES = fK(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, stored in the mobile station. In the former case, the mobile station sends an authentication failure message indicating MAC failure (MAC_FAIL) as the failure cause. In the latter case, the authentication failure message indicates synchronisation failure (SYNC_FAIL) as the failure cause. When a synchronisation failure occurs the network may initiate the identification procedure. When a synchronisation failure occurs the network performs re-synchronisation.

To detect the presence of a victim mobile station MS, 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. The captured authentication request can now be replayed by the adversary each time he wants to check the presence of MS, 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, successfully verifies the MAC and sends a synchronisation failure message. However, the MAC verification fails when executed by any other mobile station, and as a result a MAC failure message is sent. The implementation of few false BS would then allow an attacker to trace the movements of a victim mobile station, resulting in a breach of the subscriber’s untraceability. The proposed attack is shown in Figure 3.

3.3 Formal Verification

While the paging procedure is obviously a breach of users’ privacy, the traceability attack on the AKA protocol is much more subtle. Indeed, the messages exchanged through this procedure contain neither the IMSI nor the TMSI of the MS. So one could think that the AKA protocol provides untraceability by construction. But we just saw that this is not the case. Only careful analysis w.r.t. precisely defined privacy requirements could reveal this flaw.
We run the ProVerif tool on the IMSI paging procedure and on the AKA protocol\textsuperscript{1}. Details of how unlinkability and anonymity were defined using the ProVerif calculus are given in Section 6. ProVerif fails to prove the unlinkability and anonymity of the IMSI paging procedure and exhibits actual attack traces. In the case of the AKA protocol, the anonymity property is proved to hold, while the unlinkability property verification fails. Although, the trace provided by ProVerif is a false attack, it does give a hint of the real attack by highlighting the test of the MAC received from the network as the source of the problem. The adoption of formal verification tools during protocol design could have thus revealed design flaws.

4. THE ATTACKS IN PRACTICE

In order to test the attacks presented in Section 3 in a deployed telecommunication network, we use a commercially available femtocell. Although, the particular femtocell hardware is tied to the network operator SFR, the proposed attacks are not. Indeed, we tested the attacks using mobile phones registered to different operators, hence just using SFR as serving network. The authentication token AUTN is still provided by the victim’s Home network. So by testing our attacks on T-Mobile, O2, SFR, and Vodafone victim MSs, we establish that all these tested networks are vulnerable to the attacks described above. However, we want to stress here that our implementation has the only purpose of showing the feasibility of our attacks and confirm that real cellular networks follow the 3GPP standard specifications and thus are vulnerable to the proposed attacks. The same attacks could be mounted by appropriately programming a USRP\textsuperscript{2} [21], which is a hardware device able to emit and receive radio signals. In this case, one could obtain wider range attack devices in order to monitor larger areas.

4.1 Femtocell architecture

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

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

4.2 Attack Procedure

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

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

\textsuperscript{1}The ProVerif code is available online [2]

\textsuperscript{2}The femtocell architecture

![](image-url)
AKA Protocol attack To perform the AKA attack we replay a given authentication message for a specific target for which the GN 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 AuthReq using the previously cached RAND and AUTN, i.e. replays a previous request. This request is encapsulated into a GAN message and sent to the femtocell. The femtocell takes care of delivering the authentication request message on the dedicated channel assigned to the MS, as illustrated in Figure 4. The phone performs a validation of the authentication request and answers with the authentication response. If the response to the replayed authentication is a Synchronisation Failure (Figure 5), then the MS on this dedicated channel is the victim’s phone, and the victim is indeed in the femtocell area. Otherwise, the attacker needs to inject the same message to the other mobile stations in his area in order to find out if the victim MS is present or not.

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

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

5. PRIVACY PRESERVING FIXES

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

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

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

### 5.1 Lightweight Public Key Infrastructure

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

### 5.2 Protecting the IMSI Paging Procedure

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

### 5.3 Fixing the AKA Protocol

The AKA protocol is a threat for the unlinkability of 3G subscribers because the error messages sent in case of authentication failure leak information about the identity of the subscriber. To avoid this information leakage, the error messages sent in case of any type of failure should look indistinguishable from an attacker’s point of view.

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

The fixed version of the AKA protocol (Figure 7) carries on as specified by the standard, the network sends RAND, AUTN and waits for a response. The response is $RES = f_2(K_{IMSI}, RAND)$, 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.

5.4 Protecting the Identification Procedure

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

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

5.5 Discussion of the Proposed Fixes

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

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

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

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

For roaming purposes, each Home Network (HN) can act as certifying authority of the Serving Network (SN) for its own subscribers. The public key $pbHN$ of the HN could be stored in the USIM. At registration time with a SN, the MS would declare its HN, and the SN would provide the MS with its public key $pbSN$, together with a certificate from the mobile station’s HN ($\text{sign}_{peHN}(pbSN)$). 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.

6. VERIFICATION

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

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

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{identification.png}
\caption{Identification Procedure Fix. The identity response is encrypted with the public key of the network. The $r$ denotes randomised encryption.}
\end{figure}
properties. The use of randomised encryption anyway would achieve stronger properties with respect to the secrecy of the sequence number, we hence recommend the adoption of this modification in the standard protocol.

### 6.1 ProVerif Calculus

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

Cryptographic primitives are modelled as functions and messages are represented by terms built over an infinite set of names a, b, c, . . ., an infinite set of variables x, y, z, . . . and a finite set of function symbols f1 . . . fn. Function symbols represent cryptographic primitives that can be applied to messages. The effect of applying function symbols to terms is described by a set of reduction rules.

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

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

\[
P, Q, R ::= \text{null processes} \\
P | Q \text{ parallel composition} \\
!P \text{ replication} \\
\textbf{new} n; P \text{ name restriction} \\
\textbf{if} M = N \text{ then } P \text{ else } Q \text{ conditional} \\
\textbf{let} M = D \text{ in } P \text{ else } Q \text{ destructor application} \\
\textbf{in}(M, x); P \text{ message input} \\
\textbf{out}(M, N); P \text{ message output}
\]

We give here only the informal semantics of the calculus. 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 \( \textbf{new} n; P \) creates a new name \( n \) whose scope is restricted to the process \( P \) and then runs \( P \). The message input \( \textbf{in}(M, x); P \) reads a process ready to input from the channel \( M \). The message output \( \textbf{out}(M, N); P \) describes a process that sends a term \( N \) on the channel \( M \) and then behaves like \( P \). The let construct tries to rewrite \( D \) and matches the result with \( M \); if this succeeds, then the variables in \( M \) are instantiated accordingly and \( P \) is executed; otherwise \( Q \) is executed. The conditional checks the equality of two terms \( M \) and \( N \) and then behaves as \( P \) or \( Q \) accordingly. We will omit the else branch of a let or a conditional when the process \( Q \) is 0.

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

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

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

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

\[
\text{out}(c, \text{choice}(a, b)).
\]

### 6.2 Strong Unlinkability

Strong unlinkability is defined in [11] as follows. Let \( P = \text{new} n.(\textbf{let} R_1 = \text{new id.new} \text{ m.init.};!(\text{new s.main.}) \text{ for all } i \in \{1, \ldots, p\}, \text{ we build the protocol } P_R^i \text{ as follows:}

\[
P_R^i = \text{new} n.(\textbf{let} R_1 = \text{new id.new} \text{ m.init.};!\text{new s.main.})
\]

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

\[
S_{\text{SNLINK}} = \text{new} pvN; \textbf{let} pbN = pub(pvN) \textbf{in} \\
\text{out}(c, pbN); \textbf{!new sk; new imsi; new sqn}; (SN | MS)
\]

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

\[
PV_{\text{SNLINK}} = \text{new} pvN; \textbf{let} pbN = pub(pvN) \textbf{in} \\
\text{out}(c, pbN); \textbf{!new sk1; new imsi1; new sk2; new imsi2; new sqn}; \\
\textbf{let} (sk, imsi) = \text{choice}((sk1, imsi1), (sk2, imsi2)) \\
\textbf{in} (SN | MS)
\]

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 bi-process. ProVerif proves that the strong unlinkability property is satisfied by our models of the fixed identification, paging and AKA protocols as described in Section 5.

6.3 Strong Anonymity

Strong Anonymity is defined in [11] as follows. Let \( P = \text{new } \pi(R_1 | \cdots | R_p) \) be a \( p \)-party protocol where \( \forall i \in \{1, \ldots, p\}, R_i = \text{new } id.N.\text{new } m.i.n.t.i.(\text{new } s.m.a.in.). \) For all \( i \in \{1, \ldots, p\}, \) we build the protocol \( P^{R_i} \) as follows:

\[
P^{R_i} = \text{new } \pi(R_1 | \cdots | R_p | R_V)
\]

\[
R_V = \text{new } \pi.m.i.t.n.(\text{id}_{i/V} / \text{id}).(\text{new } s.m.a.in.(\text{id}_{i/V} / \text{id})).
\]

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

\[
S_V = \text{new } pm.N; \text{let } pb.N = \text{pub}(pv.N) \text{ in out(c, pb.N)};
\]

\[
\text{![new sk; new imsi; !(new sqn; (SN | MS))]} \mid \text{new sk; ![new sqn; (SN | MS)]}.
\]

Where the mobile station MS, with publicly known identity IMSI, can run the protocol. The mentioned observational equivalence can be translated in the following ProVerif birecursive PVANON, where \( \text{imsi}_{V}, \text{imsi}_{MS} \) are permanent mobile station identities:

\[
\text{free } \text{imsi}_{V}, \text{imsi}_{MS}.
\]

\[
PVANON = \text{new } pv.N; \text{let } pb.N = \text{pub}(pv.N) \text{ in out(c, pb.N)};
\]

\[
(\text{![new sk; new imsi; }}) (\text{![new sqn; (MS | SN)]}) \mid (\text{new sk; new imsi}_{MS}; \text{let } \text{imsi} = \text{choice}((\text{imsi}_{V}, \text{imsi}_{MS}) \text{ in !new sqn; (SN | MS)})).
\]

The left side of the choice represents a system where the mobile station with public identity \( \text{imsi}_{V} \) can run the protocol. Our fixes of the identification procedure, paging procedure and AKA protocol as described in Section 5 are proved by ProVerif to satisfy anonymity.

We took particular care in avoiding false attacks that could be reported by the tool due to its abstractions. Indeed, we formally define privacy properties through observational equivalence, however, ProVerif adopts a stronger equivalence relation called diff-equivalence. In particular, diff-equivalence can distinguish between the execution of different branches of a conditional statement even in the following case: if \( a = a \) then \( P \) else \( P' \) \( \neq_{lg} \) if \( a = b \) then \( P' \) and hence, although the above processes are observationally equivalent \((P \text{ is executed regardless the result of the if statement evaluation}), \) they do not satisfy diff-equivalence.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Identification</th>
<th>Paging</th>
<th>AKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unlinkability</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Anonymity</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
</tbody>
</table>

Table 1: ProVerif Results on Fixed Procedures

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

7. CONCLUSION

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

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

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

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

\(^2\)The ProVerif code is available online [2]
8. REFERENCES

9. APPENDIX

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

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

Fixed IMSI paging procedure in ProVerif.

```
1 let PAGING_MS = in(c, x);  
2 let (msgtype, xrand, xblob) = x in (  
3 if msgtype = pagingReq then (  
4 let (xpage, ximsi, =sqn, xchall) =  
5 sdec(f(k, xrand), xblob) in (  
6 if xpage = page then (  
7 if ximsi = ximsi then (  
8 let imsi = choice[ximsi, imsi_V] in (  
9 let otmsi = new rand; new chall;  
10 new r_sn1; new r_sn2;  
11 let UK = f(k, rand) in (  
12 out(c, (pagingReq, rand, senc(uk, r_sn2,  
13 (page, imsi, sqn, chall))));  
14 in(c, pres)).

Biprocess for unlinkability of IMSI paging.

15 process new pvN; let pbN = pub(pvN) in (  
16 out(c, pbN);  
17 (! (new sk1; new imsi1; new otmsi1;  
18 (! (new sk2; new imsi2; new otmsi2; new sqn;  
19 let imsi = choice[imsi1, imsi2] in (  
20 let k = choice[sk1, sk2] in (  
21 let otmsi = choice[otmsi1, otmsi2] in (  
22 !( (PAGING_MS) | (PAGING_SN))))))).

Biprocess for anonymity of IMSI paging.

23 process new pvN; let pbN = pub(pvN) in (  
24 out(c, pbN);  
25 (! (new k; new imsi; new otmsi;  
26 !( (new k; new imsi_W) | (PAGING_SN))))).  
27 ! (new k; new imsi; new otmsi)
28 let imsi = choice[id, imsi_W] in (  
29 !( (PAGING_MS) | (PAGING_SN)))).
```

Fixed AKA procedure in ProVerif.

```
30 let AKA_MS = new r_ms; in(c, x);  
31 let (xrand, xautn) = x in (  
32 let (msg, xmac) = xautn in (  
33 let ak = f5(k, xrand) in (  
34 let sqn = sdec(ak, msg) in (  
35 let mac = f1(k, (xrand, xautn)) in (  
36 if (xmac, sqn) = (mac, osqn) then (  
37 let res = f2(k, xrand) in (  
38 let ck = f3(k, rand) in (  
39 let ik = f4(k, xrand) in (  
40 out(c, res);
41 in(c, xmsg));  
42 else (out(c, senc(pbN, r_ms,  
43 Fail, imsi, osqn))))))).

Biprocess for unlinkability of AKA.

44 let AKA_SN =  
45 new rand; new r_sn; new s; new r;  
46 let mac = f1(k, (rand, osqn)) in (  
47 let res = f2(k, rand) in (  
48 let ck = f3(k, rand) in (  
49 let ik = f4(k, rand) in (  
50 let autn = (senc(ak, r, r_sn, osqn), mac) in (  
51 let av = (rand, res, ck, ik, ak) in (  
52 out(c, (rand, autn));  
53 in(c, xres);  
54 if xres = res then (  
55 out(c, senc(ck, r, s)));  
56 else (out(c, reject))))))).

Biprocess for anonymity of AKA.

57 process new pvN; let pbN = pub(pvN) in (  
58 out(c, pbN);  
59 ! (new sk1; new imsi1; new otmsi1;  
60 (! (new sk2; new imsi2; new osqn; new otmsi2;  
61 let imsi = choice[imsi1, imsi2] in (  
62 let k = choice[sk1, sk2] in (  
63 let otmsi = choice[otmsi1, otmsi2] in (  
64 !( (AKA_MS) | (AKA_SN))))))).
```

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

<table>
<thead>
<tr>
<th>Properties</th>
<th>Identification</th>
<th>Paging</th>
<th>AKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secrecy</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K&lt;sub&gt;IMSI&lt;/sub&gt;</td>
<td>NA</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CK, IK</td>
<td>NA</td>
<td>NA</td>
<td>✓</td>
</tr>
<tr>
<td>Confidential</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NA Not Applicable ✓ Proved to hold × Attack found