A Biometric Authentication Protocol for 3G Mobile Systems: Modelled and Validated Using CSP and Rank Functions

Christos K. Dimitriadis[1] and Siraj A. Shaikh[2]
(Corresponding author: Christos K. Dimitriadis)

Department of Informatics, University of Piraeus[1]
80 A. Dimitriou, 18534 Piraeus, Greece (Email: cricodc@unipi.gr)
Department of Multimedia & Computing, University of Gloucestershire Business School[2]
LC 118, Park Campus, Cheltenham Spa, GL52 2RH, UK

(Received Nov. 14, 2005; revised and accepted Apr. 25, 2006)

Abstract

This paper describes a protocol, called BIO3G, for establishing secure and privacy friendly biometric authentication in 3G mobile environments. BIO3G provides real end-to-end strong user authentication to the mobile operator, requiring no storing or transferring of biometric data and eliminating the need for biometric enrolment and administration procedures, which are time-consuming for the user and expensive for the mobile operator. BIO3G was modelled and evaluated using the formal process algebra CSP.

Keywords: 3G, authentication, biometrics, CSP

1 Introduction

Third Generation (3G) mobile systems offer true broadband data transmission, opening the path for the provision of new and improved services. The two dominant standards, are the Universal Mobile Telecommunications System (UMTS) - developed by the 3rd Generation Partnership Project (3GPP), a joint initiative of telecommunication standardization organizations from the US, Europe, Japan and Korea - and the Code-Division Multiple Access 2000 (CDMA2000) - developed by a separate partnership of standardization organizations called 3GPP2 - are designed in order to support a wide range of multimedia services, with enhanced performance, security and cost effectiveness.

User authentication is a primary element of the 3G network access security mechanism, which is usually implemented by the use of a Personal Identification Number (PIN) [20]. The rest of the process relies on the authentication of pre-stored secrets, such as cryptographic keys, or identifiers such as the International Mobile Subscriber Identity (IMSI), but not actually the user. Furthermore, knowledge as well as the possession of an item, does not distinguish a person uniquely, revealing an inherent security weakness of password and token-based authentication mechanisms. Moreover, PIN stealing, guessing or cracking have become very popular, with software tools implementing relevant attacks and research papers describing sophisticated techniques for invading PIN security [4].

Modern biometric technologies provide enhanced security levels by introducing a new dimension in the authentication process called "proof by property". Biometrics is defined as the *automatic use of human physiological or behavioral characteristics to determine or verify an identity* [15]. However, the design and deployment of a security architecture incorporating biometric technologies hides many pitfalls, which when underestimated can lead to major security weaknesses and privacy threats [9].

This paper proposes a protocol, called BIO3G, for establishing secure and privacy friendly biometric authentication in 3G mobile environments. BIO3G provides real end-to-end strong user authentication to the mobile operator, requiring no storing or transferring of biometric data and eliminating the need for biometric enrolment and administration procedures, which are time-consuming for the user and expensive for the mobile operator. BIO3G substitutes the weak PIN mechanism upon which network access security relies and proposes an alternative to local biometric authentication, which is commonly deployed for gaining access to the mobile device. BIO3G was modelled and evaluated using the formal process algebra Communicating Sequential Processes (CSP) [13].

The paper is organized as follows: Section 2 presents the basics of 3G network access security. Section 3 discusses the results of a detailed security and privacy issues relevant to the incorporation of biometrics in 3G. Section 4 is a high level description of BIO3G's logic, which targets at facilitating its detailed modelling in later sec-

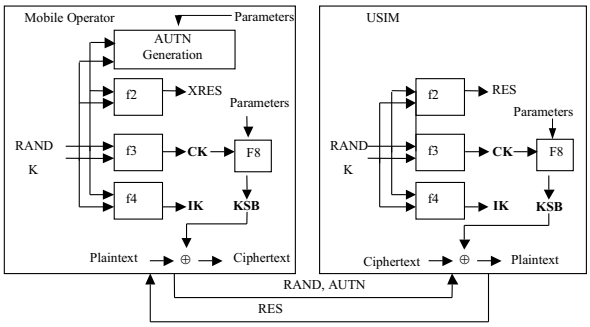

Figure 1: UMTS-AKA summary

tions using CSP. Section 5 presents the basic definitions of CSP and discusses trace semantics, which are later used to specify BIO3G's goals. Section 6 presents the CSP model of BIO3G. Section 7 presents the results of the security validation that was conducted using CSP. Section 8 concludes the paper.

2 UMTS Network Access Security Summary

BIO3G utilizes the existing network access authentication mechanism of UMTS, as specified by the latest 3GPP standards for 3G security [2]. These standards describe the UMTS-AKA mechanism as their core element for entity authentication, user identity management, confidentiality and integrity. The UMTS-AKA mechanism is based on a 128-bit secret key (K), which is pre-shared between the mobile operator and the USIM. The USIM is a cryptography-enabled smart card identified by a 15 digit number called IMSI. The USIM authenticates the user, by the use of a PIN. Figure 1 presents a summary of the elements of UMTS-AKA.

Mutual authentication between the USIM and the mobile operator is realized by a challenge and response mechanism. A random number ($RAND$) is calculated by the mobile operator and submitted to the USIM, along with a value ($AUTN$) derived by the combination of $RAND$ and K with a number of parameters, including a sequence number. The USIM authenticates the mobile operator by analysing and verifying $AUTN$. The USIM computes a value (RES), by applying $RAND$ and K to a function (f2), and submits it to the mobile operator for verification (comparison with similarly computed $XRES$), realizing the authentication of the USIM. Figure 1 presents a summary of UMTS-AKA.

Regarding confidentiality, the USIM is using the UMTS ciphering algorithm (f8), which produces a keystream block (KSB) using a 128-bit Cipher Key (CK) and a number of parameters. Integrity protection is implemented by the deployment of a 128-bit Integrity Key (IK), which is used for the calculation of Message Authentication Codes (MAC). The CK and IK are computed by the USIM and the mobile operator, by applying the pre-shared K and $RAND$ to key generation functions (f3 and f4 respectively). The CK, IK, $AUTN$, $RAND$ and $XRES$ compose a group of UMTS-AKA authentication elements, called Authentication Vector (AV).

To summarize, according to the specifications of 3GPP, the user is authenticated only locally in the USIM, by the provision of a PIN. The USIM utilizes two pre-stored values, IMSI for identifying and K for authenticating the user to the mobile operator, residing the whole security infrastructure to a simple PIN mechanism.

3 Biometrics in 3G - Protocol Specifications

Biometrics enhance security and privacy, by implementing strong authentication mechanisms towards the protection of private data that may be exchanged over a 3G application. On the other hand, biometrics may threaten the overall security of the system, since immethodically designed and developed implementations may lead to even greater security weaknesses [16]. Privacy may be ventured by the unintended use of private information that could be derived from biometric measurements, such as genetic or medical data that may become criteria for discriminating human population into segments [21]. Privacy may also be invaded, in terms of identity disclosure and position or services tracking, because of the strong binding between a user and a user identity. Furthermore, according to the relevant legislation [3], biometric data are considered as private and should be stored only for justified purposes, after ensuring the free and informed consent of the user.

A security and privacy analysis was conducted in order to derive the protocol specifications. The analysis studied the sensitivity of biometric data in combination with the vulnerabilities of the 3G environment, including the communication links and the user equipment and its biometric component, by deploying a specialized methodology (called BK) for risk analysis of biometrics [8]. BK takes into account the specifications of the Biometric Evaluation Methodology (BEM) [7], which is a supplement to the Common Evaluation Methodology (CEM) of the Common Criteria (CC) [14] specialized for biometric systems. Furthermore, BK also considers the biometric protection profiles [5, 6] of the CC. In that sense, improving BIO3G through BK, contributed to its compatibility with the CC towards security certification.

The analysis indicated that biometric data should not be stored, neither by the mobile operator, or the UMTS Subscriber Identity Module (USIM), or user equipment. The biometric data that are captured by a sensor during a sampling procedure, before being processed by another component of the biometric device are called raw biometric data [15]. After their processing from the feature extractor, the biometric data are encoded to non-invertible *biometric templates* [15]. Raw biometric data are very sensitive and should not be stored permanently at any

form in the 3G device or the mobile operator. Moreover, it must be ensured that temporary stores are securely erased. The biometric templates should be stored in secure mediums. Server based architectures, where the biometric templates are stored centrally, should be avoided due to the introduction of increased risk in the system [9]. Template storage in smart cards is considered as more secure [17]. Smart cards however, do not lack of vulnerabilities. Capturing the power consumption of a chip can reveal the software code running on the chip, even the actual command. The application of Simple Power Analysis and Differential Power Analysis [12] techniques is possible to break the matching mechanism of the biometric system or reveal the biometric template. Timing Analysis attacks are similar, measuring the processing time instead of the power consumption.

The analysis also indicated that the biometric data should be protected, while being processed by the various components of the biometric module and any transmitted data should be protected in terms of confidentiality and integrity. We distinguish two categories of communication channels: The 3G network and the communication channels within the 3G user equipment and between the user equipment and the UMTS Subscriber Identity Module (USIM). Regarding the first category, the sensitivity of biometric data, as explained in the previous paragraph, imposes significant security and privacy needs for their submission over a 3G network. The transfer of raw biometric data over communication networks should be avoided. The transfer of biometric templates also introduces high risk and if realized, strong security measures should be deployed. Forward to the above, the most secure solution would be to avoid any submission of any form of biometric data. Regarding the second category, data could be captured in order to be replayed at a future time for gaining access to the system, realizing replay and man-in-the-middle attacks. These types of attacks should be addressed for preserving the security of the biometric component of the system. Confidentiality and integrity should be implemented for all transmitted data in both categories towards communication security.

According to the analysis, the biometric module should embed vitality detection features, implement mutual authentication between its components and reduce the local biometric functions such as template matching. Several attacks can be realized to operations of the biometric component of the system. A possible attack can be realized with a Trojan Horse on the feature extractor, the matching algorithm or the decision algorithm of the biometric system, acting as a manipulator of each component's output [9]. Spoofing attacks, where human artifacts or mimic techniques are deployed are also very effective [19]. Brute force attacks are also applicable and are implemented by attempting continuously to enter the system, by sending incrementally increased matching data to the matching function until a successful matching score is accomplished [17]. These attacks are addressed by several countermeasures including vitality detection (an extra measurement of properties such as skin elasticity, the relative dielectric constant, the conductivity, eye movement), mutual authentication between the components of the system, reduction of the local functions (for example template matching).

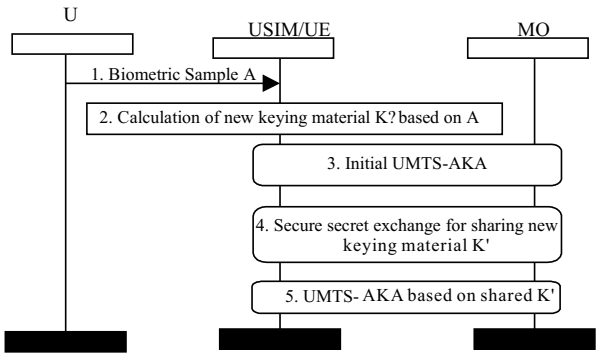

Figure 2: BIO3G overview

Finally, the need for enrolment and biometric data administration procedures should be - if possible - eliminated, in order to reduce the corresponding risk. Poor enrolment and biometric data administration procedures expose system to serious threats. During the enrolment phase, raw biometric data and biometric templates can be compromised and databases can be altered or filled with imprecise user data. Moreover, the enrolment and administration overload for preserving security and privacy is very demanding for the entity that manages the relevant services [16], which is the mobile operator in the current application. The most secure and cost-effective solution would be to minimize and if possible eliminate these procedures.

4 Protocol Logic Overview

This section presents the logic of BIO3G, in order to facilitate the detailed model of the protocol that is presented in the following sections. BIO3G implements end-to-end strong authentication of the user to the mobile operator, by introducing a biometric process to the core of UMTS-AKA. Figure 2 presents a sequence chart of BIO3G, where there are three entities present: the User (U), the USIM connected to the User Equipment (USIM/UE) and the Mobile Operator (MO).

In Step 1, the user provides a biometric sample to the USIM/UE, which in turn calculates new 128-bit keying material in Step 2. This calculation is possible by the deployment of specific techniques [10, 18], which utilize error correction codes in order to address the non-uniformity of biometric measurements. This non-uniformity is a result of reasons such as the user interaction with the biometric sensor and the slight changes of user characteristics through time. In Step 3 UMTS-AKA is deployed for user authentication to the MO. In Step 4 BIO3G deploys a cus-

tom and privacy friendly key exchange mechanism. According to this mechanism, transformed and encrypted secrets are passed to the MO, using elements of UMTS-AKA, in order to achieve the sharing of K' with the MO. This finalizes the initial handshake of the protocol, which is realized only during the first interaction of the user with the mobile operator. Whenever the user tries to authenticate to the MO, he/she should provide a biometric sample (Step 1) to the USIM/UE in order to produce K' on the fly (Step 2). UMTS-AKA is then executed between the USIM/UE and MO (Step 5), using the new key K'.

5 Introduction to CSP

In this section we introduce CSP along with its trace semantics. We then present Schneider's CSP approach [26] to analyse and verify security protocols. We describe the idea of *rank functions* along with Schneider's central *rank function theorem* [26]. While we discuss this CSP notation in detail relevant to our usage in this paper, we take for granted the reader's basic knowledge of CSP; in-depth treatments of CSP are also provided by Roscoe [22], Schneider [24] and Ryan et al. [23].

The rest of this section is organized as follows. In Section 5.1 we introduce CSP and its trace semantics in Section 5.2. In Section 5.3 we present Schneider's model of the network. In Section 5.4 we introduce *rank functions* and Schneider's *rank function theorem* to formally verify protocols.

5.1 CSP Events and Processes

A CSP system is modelled in terms of processes and events that these processes can perform, which are essentially instances of communication, usually involving a channel and some data value. Events may be atomic in structure or may consist of distinct components. The CSP expression $a \rightarrow P$ describes a process P with event a in the interface of P. The process is initially able to perform a and then behaves as P. The process *Stop* is the simplest CSP process that can be described; it has no event transitions and does not engage in any events. The *parallel* operator $\|_A$ is used to allow P and Q to run in parallel and synchronize on events in a set of events A. This would be written as $P \|_A Q$. If P or Q were to perform any events that are not in A then they can do so independently without the need for any synchronization. A process P could be restricted on certain events A, expressed as $P \|_A STOP$ which means P is not able to perform any events in A. The *interleaving* operator $|||$ is used to allow P and Q to run in parallel but with no interaction with each other. This is written as $P ||| Q$. For the purpose of communication, a process may have channels on which it accepts inputs or produces output. The expression $c!v \rightarrow P$ describes a process that will output the value of v on the channel c and then behave as P. A process P accepting an input x on the channel c is described as $c?x \rightarrow P(x)$ where the behavior of P after the input is described as $P(x)$, determined by the input.

5.2 Trace Semantics

The trace semantics in CSP allows us to capture the sequence of events performed by a communicating process as a trace and then use the trace to model the behavior of the process. A trace is a sequence of events tr. A sequence tr is a trace of a process P if some execution of P performs exactly that sequence of events. This is denoted as $tr \in traces(P)$, where $traces(P)$ is the set of all possible traces of P. An example of a trace could be $\langle a, b \rangle$ where event a is performed followed by event b, whereas $\langle\rangle$ is an empty trace.

A concatenation of two traces tr_1 and tr_2 is written as $tr_1^\frown tr_2$, which is the sequence of events in tr_1 followed by the sequence of events in tr_2. A trace tr of the form $\langle a \rangle^\frown tr'$ expresses event a followed by tr', the remainder of the trace. A prefix tr' of tr is denoted $tr' \leq tr$. The length $\#tr$ of a trace is the number of elements that it contains so that for example, $\#\langle a, b, d \rangle = 3$, whereas the set of events appearing in a trace tr is denoted as $\sigma(tr)$. The projection operation, $tr \upharpoonright A$, is the maximal subsequence of tr, all of whose events are drawn from a set of events A. Trace semantics can be used to specify security properties for protocols as trace specifications. This is done by defining a predicate on traces and checking whether every trace of a process satisfies the trace specification. For a process P and a predicate S, P satisfies S if $S(tr)$ holds for every trace tr of P. More formally, P **sat** $S \Leftrightarrow \forall\, tr \in traces(P) \bullet S(tr)$.

We use the above definition to specify a trace specification for a process, in terms of the occurrence of events in its traces. For some sets of events R and T, the trace specification R **precedes** T is defined as

$$P \textbf{ sat } R \textbf{ precedes } T \Leftrightarrow$$
$$\forall\, tr \in traces(P) \bullet (tr \div R \neq \langle\rangle \Rightarrow tr \upharpoonright T \neq \langle\rangle),$$

where a process P satisfies the predicate R **precedes** T if any occurrence of an event from T is preceded by an occurrence of an event from R in every trace tr of P.

5.3 Schneider's Model of the Network

Schneider [26] models the protocol as a network where an arbitrary number of participants engage with each other along. The participants are modelled as CSP processes acting in parallel. An intruder process is also modelled alongside these participants, with capabilities as defined by Dolev and Yao [11]. These capabilities include blocking, replaying, spoofing and manipulating any messages that appear on any of the public channels in the network. In order to give the intruder complete control of the network, Schneider models the network such that all processes communicate with each other through the intruder, that is to say, the intruder becomes the medium.

To express message transmission and reception for each process, Schneider introduces two channels, *send* and *receive*, which are public channels that all processes use to send and receive messages by. The events are structured as $send.i.j.m$ where a message m is sent by source i to destination j on the channel *send* while $receive.j.i.m$ represents a message m being received by j from a source i on the channel *receive*.

We consider a set of users $\mathcal{U}$ to represent all the participants that use the network and *Intruder* to denote the intruder process. For each participant $i \in \mathcal{U}$, a CSP process $USER_i$ represents the behavior of the participant. We specify the complete network **NET** as

$$\mathbf{NET} = (|||_{i \in \mathcal{U}} USER_i) \parallel_{(send,receive)} Intruder,$$

where all participants in U are forced to synchronize with *Intruder* on *send* and *receive* channels. In order to model the capabilities of the intruder according to the Dolev and Yao [11] model, Schneider [26] introduces a generates '$\vdash$' relation to characterize what messages may be generated from a given set of messages. The rules that define this relation are as follows:

- $m \in S$ then $S \vdash m$
- $S \vdash m$ and $S \subseteq S'$ then $S' \vdash m$
- $S \vdash m_i$ for each $m_i \in S'$ and $S' \vdash m$ then $S \vdash m$
- $S \vdash m \wedge S \vdash k \Rightarrow S \vdash \{m\}_k$
- $S \vdash \{m\}_k \wedge S \vdash k \Leftrightarrow S \vdash m$
- $S \vdash m_1.m_2 \Leftrightarrow S \vdash m_1 \wedge S \vdash m_2$
- $S \vdash m_1 \wedge S \vdash m_2 \Leftrightarrow S \vdash m_1.m_2$

Where S is some set of messages, m is a message, and k is some key. The relation can be extended to simulate further properties of cryptography or message extraction. We use this relation to specify a recursive definition of *Intruder* as follows:

$$\begin{aligned} Intruder(S) \quad &= send.i.j.m \rightarrow Intruder(S \cup \{m\}) \\ &\quad \Omega \\ &\quad \Box_{i,j \in U, S \vdash m} receive.i.j.m \rightarrow Intruder(S) \end{aligned}$$

Ω symbolizes the end of an example or proof, while $\Box$ symbolizes a bracket.

The Intruder process is parameterized by a set of messages S that denotes the set of messages in the possession of the intruder. The process is defined such that it has a choice: the first branch models the transmission of a message m, from a participant i to participant j on the channel send, after which the process behaves like the intruder with that additional message m. The second branch allows the intruder to send any message m to any participant i pretending to be some participant j, generated under $\vdash$ from S, after which the process remains with the same knowledge. The above definition of *Intruder* allows us to achieve two things, firstly, model the behavior of an intruder in precise terms, such that it may (or may not) wish to block, spoof or manipulate some (or all) messages, and, secondly, allow the intruder to possess any initial public knowledge about the network such as participant identities and their respective public keys. Such a set of initial knowledge is denoted as *Initial Knowledge*, **IK**, and specifies *Intruder* such that $Intruder(\mathbf{IK})$.

5.4 Rank Functions

Consider the set of participant identities on the network to be $\mathcal{U}$, the set of nonces used by the participants in protocol runs as $\mathcal{N}$ and a set of encryption keys used as $\mathcal{K}$. The set of all such atoms is $\mathcal{A}$, where the atoms are defined as $\mathcal{A} = \mathcal{U} \cup \mathcal{N} \cup \mathcal{K}$. We consider a message space $\mathcal{M}$ to contain all the messages and signals that may appear during a protocol's execution, such that $m \in \mathcal{A} \Rightarrow m \in \mathcal{M}$. Schneider [26] defines a rank function ρ to map events and messages to integers $\rho : \mathcal{M} \rightarrow'$. The message space is then divided into two parts where

$$\begin{aligned} M_{\rho-} &= \{m \in \mathcal{M} | \rho(m) \leq 0\} \\ M_{\rho+} &= \{m \in \mathcal{M} | \rho(m) > 0\}. \end{aligned}$$

The purpose of this partition of the message space is to characterise those messages that the intruder might get hold of without compromising the protocol - assigned a positive rank - and those messages that the enemy should never get hold of - assigned a non-positive rank. It is desirable for a process never to transmit a message of non-positive rank. For a certain process P to maintain positive rank, it is understood that it will never transmit a message with a non-positive rank unless it has previously received a message with a non-positive rank. More formally, for a process P,

$$\begin{aligned} P \textbf{ maintains } \rho \Leftrightarrow \forall tr \in traces(P) \bullet \rho(tr \Downarrow receive) > 0 \\ \Rightarrow \rho(tr \Downarrow send) > 0 \end{aligned}$$

In other words P will never transmit any message m of $\rho(m) \leq 0$ unless it has received some m' of $\rho(m') \leq 0$ previously, with respect to some rank function ρ. It is not important who the message is received from or is sent to.

Schneider [26] presents a general-purpose rank function theorem that ensures the messages that an Intruder gets hold of do not compromise the security property that the protocol provides. Considering that the communication channels are public - under the control of the *Intruder* - any message that flows through them should be of positive rank. If a message with non-positive rank flows through the channel then the intended secrecy of the message is compromised. A protocol is verified to be correct with regard to its security property, if it allows messages of only positive rank to be communicated through the channels.

Theorem 1. (Rank Function Theorem) *If, for sets R and T, there is a rank function $\rho : \mathcal{M} \rightarrow'$ satisfying*

R1. $\forall m \in \mathbf{IK} \bullet \rho(m) > 0$,

R2. $\forall\ \boldsymbol{S} \subseteq \mathcal{M}, m \in M \bullet ((\forall m' \in \boldsymbol{S} \bullet \rho(m') > 0) \wedge \boldsymbol{S} \vdash m) \Rightarrow \rho(m) > 0$,

R3. $\forall t \in T \bullet \rho(t) \leq 0$,

R4. $\forall i \in U \bullet User_i \ \|_R \ Stop$ **maintains** ρ,

then **NET sat** R **precedes** T.

The theorem, the proof of which is available in [26], states that if the rank function, and therefore the underlying **NET**, satisfies the four properties, then no messages of non-positive rank can circulate in **NET** $\|_R$ *Stop*. In particular, an intruder should not be able to generate any illegal messages from the messages it knows at the beginning of the protocol from the set $\boldsymbol{IK}$, nor from the messages it sees during the protocol execution, denoted by a set $\boldsymbol{S}$. Also, honest participants should not be able to generate any illegal messages unless they are sent one, that is, every honest process maintains ρ while being restricted on R. The actual verification of the theorem conditions is performed manually for every rank function constructed for a protocol. Verifying different specifications may require different rank functions to be constructed for the same protocol. This is due to the different events that **NET** may be restricted on for different specifications - sets R and T will contain different events for different cases.

6 Modelling BIO3G in CSP

In this section, we use CSP to present a formal specification of the BIO3G protocol. We clarify the assumptions for the protocol, particularly with regard to the UMTS-AKA mechanism. We then specify the different processes involved and, formalise the authentication and key establishment properties of the protocol as trace specifications.

We identify three entities that participate in the protocol and denote the User as U, the UMTS Subscriber Identity Module as $USIM$ and the Mobile Operator as MO. We represent a non-invertible value B derived from a biometric sample A, provided by U, as $B = f_{fe}(A)$, where f_{fe} is a randomness generating function [10]. For the purpose of our specification, we model the function f_{fe} to have two important properties:

- f_{fe} is one-to-one and error-tolerant (for a specific maximum space d) [10], that is, for some $A' = A+T$, if $T > d$ then $f_{fe}(A) \neq f_{fe}(A')$, else if $T < d$ then $f_{fe}(A) = f_{fe}(A')$;
- f_{fe} is a one-way function, that is, given B it is computationally hard to find A such that $f_{fe}(A) = B$.

The BIO3G protocol relies and enhances UMTS-AKA for some pre-shard values between the entities, which form the core message components in the protocol. We assume both $USIM$ and MO are in secure possession of K, CK, IK and KSB and, are aware of each other with respect to these.

We divide the protocol into two phases. The first phase is concerned with obtaining a biometric sample A from a user U and deriving a non-invertible value B using the randomness generating function described above, such that $B = f_{fe}(A)$. The User Equipment, hereafter known as UE, is responsible for obtaining A from U and calculating B. The UE then passes the value of B onto $USIM$. We assume the communication between UE and $USIM$ is internal and therefore reliable as per the specifications of 3GPP [1].

The second phase involves a single step of transmission from $USIM$ to MO, allowing MO to authenticate U. This is achieved by deriving a new symmetric key K' using B, which will replace the original key K for subsequent UMTS-AKA procedures. Note that the exchange of values between the UE and MO, for the establishment of the new K', takes place only during the initial run of the BIO3G protocol. When the user U reconnects to the network, a new key will be established between $USIM$ and MO using a new biometric measurement and the existing key K. If the value of the new key is correct, UMTS-AKA will succeed and positively authenticate the user to the MO. In case of a mismatched biometric sample, that is a different value of B, UMTS-AKA will fail to authenticate the user. We specify Phase 2 of the protocol informally as follows:

$$USIM \rightarrow MO : KSB \oplus D,\ |KSB \oplus D|IK,$$

where the value D is an offset of B calculated against the exclusive-or'd combination of CK and IK:

$$D = (CK \oplus IK) - B,$$

and the actual message of the protocol is KSB exclusive-or'd with the value of D, $KSB \oplus D$, concatenated with the MAC-I (Message Authentication Code for Integrity) value of $KSB \oplus D$ using the key IK. The purpose of MAC-I is to provide data integrity for the encrypted message $(KSB \oplus D)$ in the protocol. It is calculated using an integrity algorithm, known as f_9 from the UMTS-AKA standard. In more detail, for some message M, we denote MAC-I produced by the f_9 algorithm using a key IK as $|M|_{IK}$, along with the following parameters:

$$|M|_{IK} = f_9(IK, COUNT - I, FRESH, DIRECTION, M),$$

where IK is a pre-shared *Integrity Key*, $COUNT - I$ is a sequence number used for the integrity of the communication between the entities, $FRESH$ is a nonce used to avoid message replay and $DIRECTION$ is used to distinguish between the direction of the message. Once the actual message is received by MO, it performs two main operations:

1) checks the integrity of the message $KSB \oplus D$, by computing the MAC-I value for it and comparing it with the MAC-I value sent. If the values do match, then this validates data integrity for the sent message.

2) derives a new shared key K' - by first recovering D from the message such that, $D = KSB \oplus (KSB \oplus D)$,

and then computing B as an offset of D against the exclusive-or'd combination of CK and IK such that, $B = (CK \oplus IK) - D$, and finally, deriving the key K' such that, $K' = f_3(K, B)$ (function f_3 is part of UMTS-AKA standard). Implicit here is also MO's authentication of U, by way of U's sample $A-$ since B is derived as $B = f_{fe}(A)$ and K' is derived as $K' = f_3(K, B)$, the establishment of K' by MO allows it to authenticate U.

We specify the protocol in CSP and model different processes to represent the different entities taking part in the protocol. We define three processes UE, $USIM$ and MO and specify them as follows:

$$UE(a) = biosample?a \rightarrow \\ Running.UE.USIM.a \rightarrow \\ submit.UE.USIM!f_{fe}(a) \rightarrow Stop$$

$$USIM(ksb, ck, ik, k) = accept.USIM.UE?b \rightarrow \\ Running.USIM.mo.b.k' \rightarrow \\ send.USIM!mo!(ksb \oplus d, |ksb \oplus d|ik) \rightarrow Stop$$

$$MO(ksb, ck, ik, k) \\ = receive.MO?usim?(ksbd, |ksbd|ik) \rightarrow \\ Commit.MO.usim.b.k' \rightarrow Stop$$

We model different types of channels for CSP events to represent the different types of communications between the entities. We use a *biosample* channel to abstract away the obtaining of a biometric sample from a biometric reader, which we assume is part of the user equipment. We use *submit* and *accept* channels, which we consider to be private between UE and $USIM$ and, models the internal communication within UE. We use send and receive channels to represent the 3G network, which is considered to be public and hostile.

Observe that we parameterise the CSP processes such that the values of a keystream block ksb, the cipher key ck, the Integrity Key ik and a pre-shared key k are all passed onto $USIM$ and MO as perfect values. The user equipment process UE is modelled to represent obtaining a biometric sample from a user, computing a non-invertible value using the f_{fe} function and passing it onto $USIM$. The $USIM$ process accepts the f_{fe} computed value and essentially executes Phase 2 of the protocol. We use these processes to model the entire network in CSP. We model a network **NET** as

$$\mathbf{NET} = ((UE(a)_{\{submit\}} \parallel_{\{accept\}} USIM(ksb, ck, ik, k)) \parallel \\ MO(ksb, ck, ik, k)) \parallel Medium,$$

where $(UE(a)_{\{submit\}} \parallel_{\{accept\}} USIM(ksb, ck, ik, k))$ represents a combined entity in which UE and $USIM$ operate. This entire system runs in parallel with the mobile operator $MO(ksb, ck, ik, k)$. We combine these two entities in an interleaving parallel composition. We then

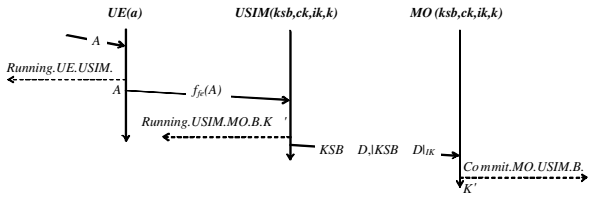

Figure 3: An illustration of a BIO3G protocol run

model the $Medium$ process to represent the actual communication network through which both entities communicate with each other. This $Medium$ process allows us to abstract away a hostile and an unreliable network. The $Medium$ process could be replaced or defined further to model a particular intruder process or features particular to a 3G network. We now illustrate a specific run of the protocol in Figure 3.

In Figure 3 we use signal events to indicate the various stages of the protocol relevant to us. We use a $Running.UE.USIM.A$ signal to indicate $UE(a)'s$ submission of the biometric sample A to $USIM$. Although this sample is submitted in the form of a non-invertible value B, this signal is important as it indicates the submission a biometric sample on behalf of some user U. We use this event later to specify user authentication for this protocol. The $Running.USIM.MO.B.K'$ signal indicates $USIM(ksb, ck, ik, k)'s$ intention to run the protocol with $MO(ksb, ck, ik, k)$ using the non-invertible value B (to identify this unique run of the protocol) and the new key K' to signify the establishment of this key. The $Commit.MO.USIM.B.K'$ signal indicates $MO(ksb, ck, ik, k)'s$ authentication of $USIM(ksb, ck, ik, k)'s$ involvement in this run, where B indicates this unique run of the protocol and K' indicates the key established as a result of this run.

We now use *Running* and *Commit* signal events in the style of [27] to explicitly specify the two different authentication properties for this protocol. In Definition 1, we specify an entity authentication property where the mobile operator authenticates (and establishes a key with) the $USIM$ module using the non-invertible value B. This then allows us to specify the main goal of the protocol in Definition 2, that is the mobile operator's authentication of the biometric sample processed by the user equipment, and hence the user.

Definition 1.

$$BIO3G\_Auth\_Key(tr) = tr'^{\frown}\langle Commit.MO.USIM.B.K'\rangle \\ \leq tr \Rightarrow \langle Running.USIM.MO.B.K'\rangle \\ \text{in } tr' \wedge \#(tr \upharpoonright Running.USIM.MO.B.K') \\ \geq \#(tr \upharpoonright Commit.MO.USIM.B.K').$$

The first clause in the $BIO3G\_Auth\_Key$ specification specifies the causal precedence of $Running.USIM.MO.B.K'$ over $Commit.MO.USIM.B.K'$. This means that every

time a $Commit.MO.USIM.B.K'$ event occurs, it is preceded by a $Running.USIM.MO.B.K'$ event. The second clause specifies an injective agreement between the two runs, that is, for every $Commit.MO.USIM.B.K'$ there is a unique $Running.USIM.MO.B.K'$ that precedes it. The second clause is important because the protocol provides key establishment between the two entities; the use of the unique value of B ensures this. The network **NET** is now said to satisfy the property of $BIO3G\_Auth\_Key$ if all of its traces satisfy $BIO3G\_Auth\_Key$:

$$\textbf{NET sat } BIO3G\_Auth\_Key \Leftrightarrow \forall tr \in traces(\textbf{NET}) \bullet BIO3G\_Auth\_Key(tr).$$

Definition 2.

$$\begin{aligned} BIO3G\_User\_Auth(tr) = tr'^{\frown}\langle Commit.MO.USIM.B.K'\rangle \\ \leq tr \Rightarrow \langle Running.UE.USIM.A. \\ \text{in } tr' \wedge \#(tr \upharpoonright Running.UE.USIM.A) \\ \geq \#(tr \upharpoonright Commit.MO.USIM.B.K'). \end{aligned}$$

Observe that MO receives no direct data from the user U or UE - it only receives a single message from $USIM$. It does, however, receive the unique value of B, which can be used to confirm the validity of biometric sample A obtained from the user U. The derivation of a new key K', such that $K' = f_3(K, B)$, and its subsequent use (note that we assume K is already shared between MO and $USIM$), allows MO to be assured of the validity of U's biometric sample.

We use $Running.UE.USIM.A$ to indicate U's submission of a biometric sample A to UE in this run. We use $Commit.MO.USIM.B.K'$ to indicate MO's run with $USIM$ (using B and K' both of which point to a valid biometric sample). The specification then requires every time a $Commit.MO.USIM.B.K'$ event occurs it is preceded by a $Running.UE.USIM.Aevent$. As in Definition 1, this clause alone does not provide injective agreement and so we place a second clause, which does provide a one-to-one relationship between these two runs. In terms of user authentication, this means that every time the mobile operator validates a biometric sample for U, U should have taken part in that run of the protocol earlier. The network **NET** is said to satisfy the property of $BIO3G\_User\_Auth$ if all of its traces satisfy $BIO3G\_User\_Auth$:

$$\textbf{NET sat } BIO3G\_User\_Auth \Leftrightarrow \forall tr \in traces(\textbf{NET}) \bullet BIO3G\_User\_Auth(tr).$$

7 Analysing BIO3G Using Rank Functions

We use the CSP model from the previous section and analyse BIO3G using the rank functions approach from Section 5.4. In Section 7.1, we define our proof strategy and construct a rank function for BIO3G in Section 7.2. We verify the protocol using rank functions in Section 7.3. Finally, we comment on the injective agreement between protocol runs in BIO3G in Section 7.4.

7.1 Proof Strategy

Observe that the definition of **NET** in Section 4 uses a Medium process to abstract away the entire communication medium between the user equipment containing the UMTS subscriber module, that is, UE and $USIM$ and the mobile operator MO. We consider a specific run of the protocol and redefine **NET** such that

$$\begin{aligned} \textbf{NET} = ((UE(A)_{\{submit\}} \parallel_{\{accept\}} USIM(KSB, CK, IK, K)) \\ \lll MO(KSB, CK, IK, K)) \parallel Intruder, \end{aligned}$$

where the Medium process with an Intruder process. This allows us to do things. First, represent an intruder with specific capabilities as discussed in Section 5.3. We enhance these capabilities in Section 7.3 to model all the functionalities relevant to our analysis. Second, give an intruder complete control of the communication medium. Verifying the network **NET** for correctness would then mean that the BIO3G protocol can withstand all the attacks that such an intruder may launch on this protocol.

Our proof strategy is then as follows. We intend to verify the trace specification in Definition 2, that is, every time MO authenticates a valid biometric sample A from UE, indicated by $Commit.MO.USIM.B.K'$, the biometric sample A has been provided to UE, indicated by $Running.UE.USIM.A$. To achieve this, we require the trace specification in Definition 1 to hold. Now in order for us to check whether every $Commit.MO.USIM.B.K'$ is preceded by a $Running.USIM.MO.B.K'$ (see Definition 1), we restrict **NET** on the $Running.USIM.MO.B.K'$ signal event and check whether the following signal event $Commit.MO.USIM.B.K'$ is allowed to appear in the restricted **NET**. More formally,

$$\begin{aligned} \forall tr \in traces(\textbf{NET} \parallel_{Running.USIM.MO.B.K'} Stop) \\ \bullet \textbf{NET} \parallel_{Running.USIM.MO.B.K'} Stop \\ \textbf{sat } tr \upharpoonright Commit.MO.USIM.B.K' = \langle\rangle. \end{aligned}$$

Observe that the $BIO3G\_Auth\_Key$ trace specification is slightly stronger (see Definition 1) than what we check in the proof strategy above. This is important as it allows us to only verify non-injective agreement between protocol runs, that is to say, we do not check whether for every $Commit.MO.USIM.B.K'$ there is a unique $Running.USIM.MO.B.K'$ that preceding it. This is due to the limitation of the rank function theorem, which does not verify injective agreement for protocol runs; we comment on this further in Section 7.4.

7.2 Constructing a Rank Function for BIO3G

We now construct a rank function for the BIO3G protocol as per the proof strategy and evaluate the different con-

ditions provided in the theorem to judge the correctness of the protocol.

We identify the ranks on the message space for our **NET** and construct the rank function shown in Figure 4 below. We consider the identities of UE, $USIM$ and MO to be possibly impersonated by the Intruder process and therefore known to the intruder. We exclude, however, the identity of U and assign it a non-positive rank. We denote the values of the four parameters ksb, ck, ik and k as KSB, CK, IK and K and, assume them to be perfectly shared between $USIM$ and MO, for a particular run of the protocol and not available to an intruder - we assign all such values a non-positive rank. A derived key K', established as a result of this particular protocol run, is also assigned a non-positive rank. We consider all other keys to be available to the intruder, along with other values of the parameters and therefore assigned a positive rank. We assign a non-positive rank to a biometric sample A since it is unique to U. Recall that the rank function theorem is defined in terms of general sets R and T. For our analysis, we assign sets R and T to $Running.USIM.MO.B.K'$ and $Commit.MO.USIM.B.K'$ respectively:

$$\begin{aligned} R &= \{Running.USIM.MO.B.K'\} \\ T &= \{Commit.MO.USIM.B.K'\}. \end{aligned}$$

This corresponds to the proof strategy described in Section 7.1, where we need to check for the occurrence of $Commit.MO.USIM.B.K'$ in **NET** restricted on $Running.USIM.MO.B.K'$. We assign $Commit.MO.USIM.B.K'$ a non-positive rank and assign $Running.USIM.MO.B.K'$ a positive rank. Moreover, due to the proof strategy, we assign the messages $KSB \oplus D$ and $|KSB \oplus D|_{IK}$ a non-positive rank as they are not supposed to appear in the restricted network $\mathbf{N}ET \parallel_{Running.USIM.MO.B.K'} Stop$.

In the rank function shown above, we also consider ranks for some general cases:

- We assign a concatenation of two messages m_1 and m_2 a positive rank only if both m_1 and m_2 are of positive rank.
- We assign two messages exclusive-or'd with each other a positive rank if they both have the same rank, that is to say, they both have either positive or non-positive rank. So, if they both have a positive rank it means they are both available to the intruder anyway or, if they both have a non-positive rank, then it means the exclusive-or'd message can be sent out on a public channel (without the intruder retrieving either of the messages).
- We consider the fuzzy extractor function f_{fe} and assign the output of a function $f_{fe}(m)$ a positive rank only if the input m is of positive rank. This means, if the intruder is in possession of only $f_{fe}(m)$ and not m, then it is impossible to retrieve m. This allows us to model f_{fe} as a one-way function as defined earlier in Section 6.
- We assign the MAC-I value $|m|_k$, for some m, a positive rank only if both m and the key k are of positive rank. So an intruder cannot generate any MAC-I value without possessing both the message and the key, and finally.
- We consider the function f_3 (part of the UMTS-AKA standard) and assign the output of a function $f_3(k, m)$ a positive rank only if both inputs k and m are available to the intruder.

7.3 Verifying BIO3G Using Rank Functions

Before we proceed to consider each of the conditions of the rank function theorem and check whether our rank function in Figure 4 satisfies them, we alter the generates '$\vdash$' relation (see Section 5.3) to enhance the capabilities of the intruder (which is based on the Dolev-Yao [11] model). This is important as it reflects the current scenario where an intruder is in possession of extra functionalities outlined in Section 7.2.

We consider the generates '$\vdash$' relation to follow all the rules originally defined in Section 5.3 and add the following rules, where m, m_1 and m_2 are some messages and, k is a key:

- $\{m_1, m_2\} \vdash m_1 \oplus m_2$
- $\{m_1 \oplus m_2, m_2\} \vdash m_1$
- $\{m_1 \oplus m_2, m_1\} \vdash m_2$
- $\{m\} \vdash f_{fe}(m)$
- $\{m, k\} \vdash |m|_k$
- $\{m, k\} \vdash f_3(k, m)$

Recall the rank function theorem from Section 5.4 We proceed to consider each condition **R1-R4** and check whether the rank function in Figure 4 satisfies them.

R1. $\forall m \in \boldsymbol{IK} \bullet \rho(m) > 0$:
The set **IK** contains all the information that the intruder is aware of at the start of the protocol. We consider the intruder an insider, such that it is also a valid subscriber of MO and therefore, is in possession of a corresponding set of parameters that U is. We denote such parameters as KSB', CK' and IK'_I. We also consider an initial key K_I that the intruder shares with MO. The set **IK** then includes all this information, $\mathbf{IK} = \{UE, USIM, MO, KSB', CK', IK'_I, K_I\}$. There is nothing in this set that is of non-positive rank. The condition is deemed satisfied.

R2. $\forall S \subseteq \mathcal{M}, m \in \mathcal{M} \bullet ((\forall m' \in \boldsymbol{S} \bullet \rho(m') > 0) \wedge \boldsymbol{S} \vdash m) \Rightarrow \rho(m) > 0$:
This conditions checks whether a message of non-positive rank can be generated under the '$\vdash$' relation from a set of messages of positive rank. None of the messages

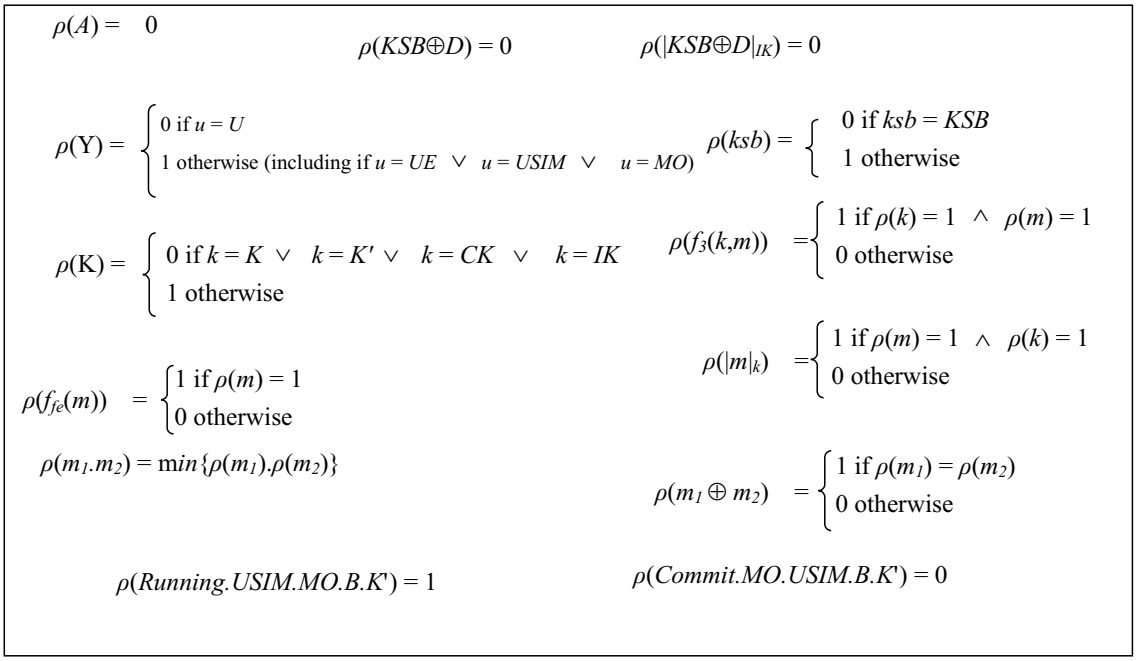

Figure 4: A rank function for the BIO3G protocol

identified as of positive rank, shown in Figure 4, let the Intruder generate any messages that are of non-positive rank. Both U and corresponding biometric sample A are considered to be out of reach of the intruder. The parameters KSB, CK, IK, K and K' are all assumed to be perfectly shared between $USIM$ and MO, that is to say, cannot be retrieved by the intruder. None of this information allows the intruder to generate $KSB \oplus D$ or $|KSB \oplus D|_{IK}$ or, anything else of non-positive rank. The condition is deemed satisfied.

R3. $\forall t \in \boldsymbol{T} \bullet \rho(t) \leq 0$:
This condition requires none of the events in T to be of positive rank. The only event in set T is the signal event $Commit.MO.USIM.B.K'$ of non-positive rank (see Figure 4). This condition is deemed satisfied.

R4. $\forall i \in \mathcal{U} \bullet\ User_i \parallel_R Stop$ **sat maintain** ρ:
For this condition to be satisfied every process in **NET** needs to **maintain** ρ while being restricted on the events in set R, where $R = \{Running.USIM.MO.B.K'\}$. Recall the definition of **NET** from Section 7.1. We consider each of the process $UE(A)$, $USIM(KSB, CK, IK, K)$ and $MO(KSB, CK, IK, K)$, restrict them on $Running.USIM.MO.B.K'$ and check whether they maintain ρ. Since only the $USIM(ksb, ck, ik, k)$ process can perform $Running.A.B.NB$, the other two processes remain unaffected. The process $UE(A)$ with a biometric sample A behaves as follows:

$$\begin{aligned} UE(A) = {} & biosample?A \rightarrow \\ & Running.UE.USIM.A \rightarrow \\ & submit.UE.USIM!f_{fe}(A) \rightarrow Stop. \end{aligned}$$

The process $UE(A)$ is not restricted to pass on the value $f_{fe}(A)$ (that is of non-positive rank) onto the channel submit. But since the submit channel is a private channel between UE and $USIM$, the intruder cannot observe $f_{fe}(A)$. The process $UE(A)$, therefore, succeeds to maintain ρ. The process $USIM(KSB, CK, IK, K)$ is restricted on $Running.USIM.MO.B.K'$, which simplifies to

$$\begin{aligned} & USIM(KSB, CK, IK, K) \parallel_{Running.USIM.MO.B.K'} Stop \\ & = accept.USIM.UE?B \rightarrow \\ & \quad \text{if } b = B \wedge k = K' \\ & \quad \text{then } Stop \\ & \quad\quad \text{else } Running.USIM.MO.b.k' \rightarrow \\ & \quad\quad send.USIM!MO!(ksb \oplus d, |ksb \oplus d|_{ik}) \rightarrow Stop. \end{aligned}$$

Now if the restricted process $USIM(KSB, CK, IK, K) \parallel_{Running.USIM.MO.B.K'} Stop$ is passed on the value of B that leads onto the value of K', then it is instructed to stop. If, however, $b \neq B$, then the process continues to behave normally with appropriate values. Due to the restriction on the process, it does not transmit $KSB \oplus D$ or $|KSB \oplus D|_{IK}$ (both of which are of non-positive rank) and therefore succeeds to **maintain** ρ. The process $MO(KSB, CK, IK, K)$ remains unaffected by the restriction.

$$\begin{aligned} & MO(KSB, CK, IK, K) \\ & \quad = receive.MO?USIM?(KSB \oplus D, |KSB \oplus D|_{IK}) \\ & \quad \rightarrow Commit.MO.USIM.B.K' \rightarrow Stop. \end{aligned}$$

Upon observation, if $MO(KSB, CK, IK, K)$ does not

receive the concatenated message $KSB \oplus D, |KSB \oplus D|_{IK}$ then it does not perform $Commit.MO.USIM.B.K'$. The only way it can perform $Commit.MO.USIM.B.K'$ is for it to receive $KSB \oplus D, |KSB_o plus D|_{IK}$. The process therefore succeeds to **maintain** ρ.

All four conditions of them theorem **R1-R4** are satisfied by the rank function in Figure 4. This ensures that

$$\forall tr \in traces(\mathbf{NET} \parallel_{Running.USIM.MO.B.K'} Stop) \bullet \mathbf{NET} \parallel_{Running.USIM.MO.B.K'} Stop \ \mathbf{sat}\ tr \upharpoonright Commit.MO.USIM.B.K = \langle\rangle.$$

We do, however, need to verify that **NET** also satisfies the trace specification given in Definition 2. More formally,

$$\forall tr \in traces(\mathbf{NET} \parallel_{Running.UE.USIM.A} Stop) \bullet \mathbf{NET} \parallel_{Running.UE.USIM.A} Stop \ \mathbf{sat}\ tr \upharpoonright Commit.MO.USIM.B.K = \langle\rangle.$$

Recall the definition of **NET**

$$\mathbf{NET} = ((UE(A)_{\{submit\}} \parallel_{\{accept\}} USIM(KSB, CK, IK, K)) \lll MO(KSB, CK, IK, K)) \parallel Intruder.$$

Let us focus on the component $((UE(A)_{\{submit\}} \parallel_{\{accept\}} USIM(KSB, CK, IK, K))$ that represents UE running alongside $USIM$. This component, due to its design, satisfies the condition that every time $USIM$ receives the value B, UE has sent the value B (where $B = f_{fe}(A)$) to $USIM$ prior to that. This essentially represents the inner operations of the user equipment, where it is understood to function correctly and not leak any information to any other channel. We denote this as a trace specification. For some trace tr:

$$tr' \frown \langle Running.USIM.MO.B.K' \rangle \leq tr \Rightarrow \langle Running.UE.USIM.A \rangle \text{ in } tr'$$

and state

$$\forall tr \in traces((UE(A)_{\{submit\}} \parallel_{\{accept\}} USIM(KSB, CK, IK, K)) \bullet tr' \frown \langle Running.USIM.MO.B.K' \rangle \leq tr \Rightarrow \langle Running.UE.USIM.A \rangle \text{ in } tr'$$

and as the rest of network **NET** cannot interfere with this property, the entire **NET** also satisfies this property

$$\forall tr \in traces(\mathbf{NET}) \bullet tr' \frown \langle Running.USIM.MO.B.K' \rangle \leq tr \Rightarrow \langle Running.UE.USIM.A \rangle \text{ in } tr'$$

and since we have already proved

$$\forall tr \in traces(\mathbf{NET}) \bullet tr' \frown \langle Commit.MO.USIM.B.K' \rangle \leq tr \Rightarrow \langle Running.USIM.MO.B.K' \rangle \text{ in } tr'.$$

It follows that

$$\forall tr \in traces(\mathbf{NET}) \bullet tr' \frown \langle Running.USIM.MO.B.K' \rangle \leq tr \Rightarrow \langle Running.UE.USIM.A \rangle \text{ in } tr'$$

which essentially states that every time the mobile operator authenticates the value B, the corresponding biometric sample A was accepted by the user equipment prior to that.

7.4 A Note on Injective Agreement

Observe that Definition 2 is slightly stronger than what we have proved in Section 7.3. The second clause in Definition 2 requires injective agreement between runs

$$\#(tr \upharpoonright Running.UE.USIM.A) \geq \#(tr \upharpoonright Commit.MO.USIM.B.K'),$$

such that every time $Commit.MO.USIM.B.K'$ appears, there is a unique $Running.UE.USIM.A$ event preceding it. The rank function theorem does not verify injectivity [26] so we cannot use rank functions to prove this. Note, however, that the design of the protocol ensures injective agreement in a number of ways:

- The integrity algorithm f_9 described in Section 6 uses $FRESH$ and $COUNT - I$ as parameters that prevent the MAC-I value of the message (and therefore the message) being replayed. The use of $FRESH$ prevents any intentional replay on behalf of an intruder whereas $COUNT - I$ prevents any confusion due to transmission errors.

- At the end of the protocol run, both $USIM$ and MO derive a new key as described in Section 4. The new key is derived using the existing key between the two parties and then used for subsequent communication. If for some reason MO is somehow led to accept a replayed message from a previous run, any subsequent communication would then not be possible as the newly derived key by $USIM$ and MO would not match. Such an arrangement, therefore, inherently enforces synchronization between the protocol runs of $USIM$ and MO, and finally.

- Recall the component $((UE(A)_{\{submit\}} \parallel_{\{accept\}} USIM(KSB, CK, IK, K))$ that represents UE running alongside $USIM$. We consider this component to work perfectly and specify its behavior as

  $$tr' \frown \langle Running.USIM.MO.B.K' \rangle \leq tr \Rightarrow \langle Running.UE.USIM.A \rangle \text{ in } tr',$$

  for some trace tr. Due to the design of the component, every time UE sends the value of some B to $USIM$, it accepts it. Conversely, every time $USIM$ accepts the value of some B, UE has actually passed

B onto $USIM$. We can therefore refine the behavior of this component further as, for some trace tr,

$$tr'^{\frown}\langle Running.USIM.MO.B.K'\rangle \leq tr \Rightarrow \langle Running.UE.USIM.A\rangle \text{ in } tr' \wedge \#(tr \upharpoonright Running.UE.USIM.A) \geq \#(tr \upharpoonright Running.USIM.MO.B.K')$$

that ensures injective interaction between UE and $USIM$ for this component.

The three mechanisms combine to ensure that BIO3G provides injective agreement between UE and MO. This is important not only for authentication purposes, but also for key establishment between the two parties.

8 Conclusions

Due to the increased sensitivity of biometric data, the introduction of biometrics for security in 3G networks is a challenging process. BIO3G was created by following a design approach that identified the necessary requirements and defined the corresponding specifications, through the detailed study of biometric technologies within the framework of their incorporation in a 3G environment. BIO3G is essentially a user authentication protocol, along with providing key derivation. The use of a single message makes the protocol less prone to cryptanalysis. The generation of non-invertible values for biometric samples is particularly important as it avoids actual biometric samples being sent over public channels. The use of the existing facilities of the $USIM$, for key generation and simple exclusive-or functions, makes BIO3G lightweight and compatible to the existing 3G infrastructures. These features, as well as the simplicity of the exchanged messages over the air interface, allow BIO3G to be ideal for mobile communications where computational resources are limited, the medium is noisy and ever increasingly hostile and, more importantly, actual biometric data is too risky to be stored on user equipment (whether stationary or mobile).

The use of CSP to model such protocols has many advantages. The notion of channels allows us to distinguish between the different types of channels and make different processes selectively synchronize on them. It also allows modelling certain events to appear on local or private channels so that we can distinguish between private and public events. It is easy to model the medium of communication separately so as to either explicitly model or abstract away the various features of wireless transmission such as noise and error, traffic analysis or specific intruder models, such as the Dolev-Yao intruder model [11]. For the purpose of verification, for example, we can specify the various Dolev-Yao capabilities for the $Medium$ process. The use of trace specifications allows us to model the exact nature of the security property being specified, in terms of the actual data being used in the protocol and injective or non-injective relationship between participant (protocol) runs.

References

[1] 3rd Generation Partnership Project, *TS 22.022 - Personalisation of Mobile Equipment (ME)*, Mobile functionality specification, 2005.

[2] 3rd Generation Partnership Project, *TS 33.102 - 3G Security*, Security architecture, 2004.

[3] *Atricle 29- EC Data Protection Working Party*, Working document on biometrics, 2003.

[4] O. Benoit, N. Dabbous, L. Gauteron, P. Girard, H. Handschuh, D. Naccache, S. Socile, and C. Whelan, *Mobile Terminal Security*, Cryptology ePrint Archive: Report 2004/158, 2004.

[5] CC-Protection Profile, *UK Biometric Device - Draft*, 2002.

[6] CC-Protection Profile, *US Government Biometric Verification Mode Protection V Profile for Medium Robusness Environment*, 2003.

[7] Common Criteria Biometric Evaluation Methodology Working Group, *Biometric Evaluation Methodology*, 2002.

[8] C. Dimitriadis and D. Polemi, "Application of multicriteria analysis for the creation of a risk assessment knowledgebase for biometric systems," in *International Conference on Biometric Authentication (ICBA)*, LNCS 3072, pp. 724-730, Springer-Verlag, 2004.

[9] C. Dimitriadis and D. Polemi, "Biometrics VRisks and controls," *Information Systems Control Journal*, vol. 4, pp. 41-43, 2004.

[10] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Eurocrypt 2004*, LNCS 3027, pp. 523-540, Springer-Verlag, 2004.

[11] D. Dolev and A. C. Yao "On the security of public key protocols," *IEEE Transaction on Information Theory*, vol. 29, no. 2, pp. 198-208, Mar. 1983

[12] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results,"in *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, LNCS 2162, pp. 251-261, Springer-Verlag, 2001.

[13] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice-Hall International, 1985

[14] ISO/IEC 15408 Information technology, *Security techniques-Evaluation criteria for IT security*, 1999.

[15] ISO/IEC JTC1, SC37/SG1, *Biometric Vocabulary Corpus*, 2004.

[16] IST V 2002 V001766, *Biometrics and Security (BIOSEC): Deliverable D3.3 V Security Recommendations: Biometric Systems Integration, Basic Research on Security, Network Protocols and PKI*, Biosec consortium, 2005.

[17] ST-1999-20078, *Business Environment of Biometrics Involved in E-Commerce V BEE: Deliverable D7.1 Conclusions and Recommendations*, http://expertnet.net.gr/bee, 2002.

[18] J. P. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *Audio and Video-Based Biometric Person Authentication (AVBPA)*, pp. 393-402, 2003.

[19] T. Matsumoto, "Gummy finger and paper iris V an update," in *Proceedings of Workshop on Information Security Research*, pp. 187-192, Japan, 2004.

[20] V. Neimi and K. Nyberg, *UMTS Security*, John Wiley & Sons, 2003.

[21] S. Prabhakar, S. Pankanti, and A, K. Jain, "Biometric recognition security and privacy concerns," *IEEE Security and Privacy*, vol. 1, no. 2, pp. 33-42, 2003.

[22] A. W. Roscoe, *The Theory and Practice of Concurrency*, Prentice-Hall International, 1997.

[23] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe, *Modelling and Analysis of Security Protocols*, Addison-Wesley, 2001.

[24] S. Schneider, *Concurrent and Real-time Systems: The CSP Approach*, Addison-Wesley London, 1999.

[25] S. Schneider, "Verifying authentication protocol implementations," in *Proceedings of the Second Workshop on Automated Verification of Critical Systems*, pp. 239-253, University of Birmingham, 2002.

[26] S. Schneider, "Verifying authentication protocols in CSP," *IEEE Transactions on Software Engineering*, vol. 24, no. 9, pp. 741-758, Sept. 1998.

[27] S. Shaikh, V. Bush, and S. Schneider "Kerberos V Specifying authenticity properties using signal events," in *Proceedings of the Indonesia Cryptology and Information Security Conference*, pp. 87-93, Mar. 2005.

Christos Dimitriadis is a researcher at the University of Piraeus, participating in European Union and National projects, including the security assessment of major mobile operators, banks and governmental authorities. Dimitriadis has 34 publications in the fields of biometrics and security. He has been invited by several organizations to provide lectures, including the ITU, NIST and the EC. His research interests include 3G and 4G security architectures, biometrics, identity management, honeynets and security protocol design and testing. Dimitriadis received a diploma of Electrical and Computer Engineering from the University of Patras-Greece and a PhD from the University of Piraeus-Greece in the field of 3G security. He is a member of IEEE and the Technical Chamber of Greece.

Siraj Shaikh is a Lecturer at the Department of Computing at the University of Gloucestershire, UK, where he teaches Computer Networking and Security. He holds a MSc in Computer Networking and is currently pursuing a PhD in the area of formal analysis of authentication protocols using the CSP and Rank functions approach. His main research interests include formal design and analysis of distributed security systems and protocols. His other interests include performance analysis of security protocols such as IPSec and SSH, and information security education. He is a Member of the British Computer Society.