Formal Analysis of Privacy in an eHealth Protocol

Naipeng Dong*, Hugo Jonker, and Jun Pang

Faculty of Sciences, Technology and Communication, University of Luxembourg, Luxembourg

Abstract. Given the sensitive nature of health data, privacy of eHealth systems is of prime importance. An eHealth system must enforce that users remain private, even if they are bribed or coerced to reveal themselves or others. Consider e.g. a pharmaceutical company that bribes a pharmacist to reveal information which breaks a doctor's privacy. In this paper, we first identify and formalise several new but important privacy notions on enforcing doctor privacy. Then we analyse privacy of a complicated and practical eHealth protocol (DLVV08). Our analysis shows to what extent these new properties as well as properties such as anonymity and untraceability are satisfied by the protocol. Finally, we address the found ambiguities which result in privacy flaws, and propose suggestions for fixing them.

1 Introduction

Traditionally, data in health care (e.g., patient records) was stored on paper files. Given the sensitive nature of health data, handling this data must meet strict security and privacy requirements. This was relatively easily satisfied by controlling access to the physical documents. Those who had access could be considered trusted not to violate security nor privacy of the data. With the advent of eHealth systems – systems that digitally store and exchange health data – security and particularly privacy requirements were often achieved using access control (e.g., see [1, 2]).

However, the introduction of eHealth systems has changed the setting. The main benefit of eHealth systems is that they facilitate the digital exchange of information amongst various roles in health care. This has two major consequences: the health care data is shared digitally with more parties, such as pharmacists and insurance companies; and, this data can be easily shared by any party with an outsider. Clearly, the assumption of a trusted network can no longer hold in such a setting. Given that it is trivial for a malicious entity to intercept or even alter digital data in transit, access control approaches to privacy and security are no longer sufficient. In this paper, we consider security and privacy of the involved parties with respect to an outsider, the Dolev-Yao adversary [3], who controls the communication network (i.e. the adversary can observe, block, create and alter information). In this setting, communication security and privacy are mainly achieved by employing cryptographic communication protocols. It is well known that designing such protocols is error-prone: time and again, flaws have been found in protocols that claimed to be secure (e.g., electronic voting systems [4, 5] have been broken [6, 7]). Therefore, we believe that the claims of an eHealth protocol must be verified before the protocol is used in practice. Without verifying that a protocol satisfies its security claims, subtle flaws may go undiscovered.

* Supported by a grant from the Fonds National de la Recherche (Luxembourg).

In order to verify whether a protocol satisfies security and privacy requirements, each property must be formally defined. Various security and privacy properties have already been defined in the literature, such as secrecy, authentication, anonymity and untraceability. We refer to these properties as regular security and privacy properties. While they are necessary to ensure security and privacy, by themselves these regular properties are not sufficient. Benaloh and Tuinstra pointed out the risk of subverting a voter [4] to sell her vote. This idea, of coercing or bribing a party into nullfying their privacy, is hardly considered in the literature of eHealth systems (notable exceptions include [8, 9]). However, this notion is important for health care – e.g., a pharmaceutical company may bribe doctors to prescribe only their medicine. Therefore, we consider not only privacy with respect to a Dolev-Yao adversary, but also privacy in the presence of an active coercer – someone who is bribing or threatening parties to reveal their privacy. We refer to these properties as enforced privacy properties. In particular, we identify the following notions of privacy [10] to counter doctor bribery: *prescribing-privacy*: a doctor cannot be linked to his prescriptions; *enforced prescribing-privacy*: preventing doctor bribes; *independency of prescribing-privacy*: preventing others to reduce a doctor's prescribing-privacy; and *independency of enforced prescribing-privacy*: preventing anyone from affecting a doctor's enforced prescribing-privacy.

Contributions. We identify three notions of enforced privacy in eHealth systems and are the first to provide formal definitions for them. In addition, we develop an in-depth applied pi model of the DLVV08 eHealth protocol [9] which is rather complicated and aims for practical usage in Belgium. Furthermore, we formally analyse privacy and enforced privacy properties of the protocol, as well as regular security and privacy properties. We find ambiguities in the protocol which potentially lead to flaws on privacy, and propose suggestions for fixing them. The modelling and full analysis of the DLVV08 protocol can be found in [11].

2 The applied pi calculus

The applied pi calculus is a language designed for modelling and analysing security protocols [12]. It assumes an infinite set of names (modelling channels and data), an infinite set of variables and a set of functions (to model cryptographic primitives). A *term* is a name or variable, or a function applied to other terms. Terms are used to model messages. An equational theory E defines equivalences between terms. A protocol is modelled as a set of roles running in parallel. The behaviour of each role is modelled as a process, defined as follows.

$$P, Q ::= 0 \mid P \mid Q \mid !P \mid \nu n.P \mid \text{in}(u, x).P \mid \text{out}(u, M).P \mid \text{if } M =_E N \text{ then } P \text{ else } Q$$
$$A, B ::= P \mid A \mid B \mid \nu n.A \mid \nu x.A \mid \{M/x\}$$

A plain process P, Q can be the empty process 0, two sub-processes running in parallel $P \mid Q$, a replication $!P$, a name restriction on a process $\nu n.P$, an input or output action followed by a process ($\text{in}(u, x).P$ and $\text{out}(u, M).P$, respectively), or a conditional choice based on the equational theory (if…then…else). To this, extended processes add variable restrictions and active substitution.

The semantics of applied pi consists of three parts: *structural equivalence*, which defines equivalence relations between two processes which only differ in structure; *internal reduction* ($\rightarrow$), which defines sub-process communication rules and *if-then-else* evaluation rules; and *labelled reduction* ($\xrightarrow{\alpha}$), which defines reduction rules to model the communication between the adversary and the protocol. For more details, see [12].

In this paper, we use "$P\{M/x\}$" (and, equivalently, "let $x = M$ in P") to denote syntactical replacement of x with M in process P. We use $=_E$ to denote term equivalence relations introduced by equational theory E. Names and variables are *free* if they are not delimited by restriction and by inputs. The sets of free names, free variables, bound names and bound variables of a process A are denoted as $\mathsf{fn}(A)$, $\mathsf{fv}(A)$, $\mathsf{bn}(A)$ and $\mathsf{bv}(A)$, respectively. A process is *closed* if it does not contain free variables. A *context* $C[\_]$ is defined as a process with a hole, which may be filled with any process. An *evaluation context* is a context whose hole is not in the scope of a replication, a conditional, an input, or an output. A term is *ground* when it does not contain variables. The *frame* $\phi(A)$ of a process A is the static knowledge revealed to the adversary, defined as an extended process where every plain sub-process is replaced with 0. The domain $\mathsf{dom}(\psi)$ of a frame ψ is the set of variables in active substitutions. Finally, $\rightarrow^*$ denotes zero or more internal reductions.

Several equivalence relations on processes are defined in the applied pi calculus. In this paper we mainly use labelled bisimilarity, which is claimed to coincide with observational equivalence [12], but easier to reason with both manually and automatically. Labelled bisimilarity $\approx_\ell$ is based on static equivalence $\approx_s$ of processes. Labelled bisimilarity compares the dynamic behaviour of processes, while static equivalence compares the static states of processes (as represented by their frames).

Definition 1 (Static equivalence [12]). *Closed frames ψ and ϕ are statically equivalent, $\psi \approx_s \phi$, if (1) $\mathsf{dom}(\psi)$=$\mathsf{dom}(\phi)$;*
(2) $\forall$ terms M and N, $(M =_E N)$ in $\psi \Leftrightarrow (M =_E N)$ in ϕ.

Extended processes A, B are statically equivalent, $A \approx_s B$, if their frames are statically equivalent: $\phi(A) \approx_s \phi(B)$.

Definition 2 (Labelled bisimilarity [12]). Labelled bisimilarity $(\approx_\ell)$ *is defined as the largest symmetric relation $\mathcal{R}$ on closed extended processes, such that $A\ \mathcal{R}\ B$ implies: (1) $A \approx_s B$; (2) if $A \rightarrow A'$ then $B \rightarrow^* B'$ and $A'\ \mathcal{R}\ B'$ for some B'; (3) if $A \xrightarrow{\alpha} A'$ and $\mathsf{fv}(\alpha) \subseteq \mathsf{dom}(A)$ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(B) = \emptyset$; then $B \rightarrow^* \xrightarrow{\alpha} \rightarrow^* B'$ and $A'\ \mathcal{R}\ B'$ for some B'.*

Applied pi assumes the Dolev-Yao adversary [3], which controls the network and can eavesdrop, block, create, and inject messages, as well as applying cryptographic primitives (e.g., decrypting eavesdropped messages). Notice that normally, dishonest users are considered as part of the adversary. However, coerced/bribed users are not modelled as part of the adversary, as the adversary does not fully trust these users.

3 Formalising privacy properties

In order to formally verify privacy properties of a protocol, the first step is to give precise definitions of privacy properties. Properties such as anonymity and untraceability

have been formally studied in the literature (e.g., [13–18]), which can be lifted to the eHealth domain. In eHealth it is important to protect doctor's prescription behaviour against bribery. Such kinds of privacy properties have not been studied formally so far.

In this section, we first define prescribing-privacy to model protecting a doctor's prescription behaviour without considering bribery. Next, we formally define three new privacy properties to protect a doctor's prescribing-privacy against bribery: *enforced prescribing-privacy*, *independency of prescribing-privacy*, and *independency of enforced prescribing-privacy*. In the end, we briefly show the definitions of anonymity, strong anonymity, untraceability and strong untraceability for eHealth protocols.

In the following discussions, we model an eHealth protocol EHP as a n-role well-formed [17] protocol of the form: $EHP = \nu\tilde{m}.init.(!R_1 \mid \ldots \mid !R_n)$. In particular, we have a doctor role R_{dr} of the form: $R_{dr} = \nu\mathtt{Id}_{dr}.init_{dr}.!P_{dr}$, where $P_{dr} = \nu\mathtt{presc}.main_{dr}$. Essentiality, this formalisation allows us to model an unbounded number of users and represent each user as an instance of a role. We focus on the behaviour of a doctor. Each doctor is associated with an identify and can execute an infinite number of sessions. Within each session, the doctor will create a prescription. Processes $init$ and $init_{dr}$ model the initialisation of the protocol and the doctor role. Process P_{dr} models a session of the doctor role. Furthermore, we use $\mathcal{C}$ to denote a context consisting of honest users; Id_{dr} and $presc$ are free variables; $\mathtt{A}$ and $\mathtt{B}$ are free names, representing doctor identities known to the adversary; and $\mathtt{a}$ and $\mathtt{b}$ are two free names, representing two different prescriptions.

3.1 Prescribing-privacy

Prescribing-privacy aims to protect doctors' prescription behaviour, which can be captured by the unlinkability of a doctor and his prescriptions. Unlinkability is normally modelled as indistinguishability when two honest users swap their actions (or items), e.g., see the formalisation of vote privacy [19]. Thus, prescribing-privacy is modelled as the equivalence of two doctor processes: in the first process, an honest doctor $\mathtt{A}$ prescribes $\mathtt{a}$ in one of his sessions and another honest doctor $\mathtt{B}$ prescribes $\mathtt{b}$ in one of his sessions; in the second one, $\mathtt{A}$ prescribes $\mathtt{b}$ and $\mathtt{B}$ prescribes $\mathtt{a}$.

Definition 3 (Prescribing-privacy). *A well-formed eHealth protocol EHP satisfies prescribing-privacy if*

$$\begin{aligned}
&\mathcal{C}[\big(init_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{a}/presc\})\big) \mid \\
&\quad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{b}/presc\})\big)] \\
\approx_\ell\ &\mathcal{C}[\big(init_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{b}/presc\})\big) \mid \\
&\quad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{a}/presc\})\big)].
\end{aligned}$$

3.2 Enforced prescribing-privacy

Enforced privacy properties have been proposed and formally studied in different domains to prevent bribery and coercion, for instance, receipt-freeness and coercion-resistance in E-voting [19, 20], receipt-freeness in online auction [21]. In eHealth, De

Decker et al. [9] identify the need to prevent a pharmaceutical company from bribing a doctor to favour their medicine. Hence, doctor's prescribing-privacy should be enforced by eHealth protocols to prevent doctors bribery.

This means that intuitively, even if a doctor collaborates, the adversary cannot be certain that the doctor has followed his instructions. Bribed users cannot be modelled as part of the adversary, as they are not trusted by the adversary. In addition, we need to model how bribed users share information obtained from channels hidden from the adversary. Inspired by Delaune et al.'s formalisation of receipt-freeness in electronic voting [19], we define enforced prescribing-privacy to be satisfied if there exists a process where the bribed doctor does not follow the adversary's instruction (e.g., prescribing a particular medicine), which is indistinguishable from a process where she does.

Modelling this property necessitates modelling a doctor who genuinely reveals all her private information to the adversary. In [19], this is achieved by process transformation $P^{\mathtt{chc}}$, which transforms a plain process P into one which shares all private information over the channel $\mathtt{chc}$ with the adversary. In addition, we also use their other transformation $P^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}$. This [19] models a process P which erases all outputs on channel $\mathtt{chc}$. Formally, $P^{\backslash \mathtt{out}(\mathtt{chc},\cdot)} := \nu \mathtt{chc}.(P \mid !\mathsf{in}(\mathtt{chc}, x))$.

Definition 4 (Enforced prescribing-privacy). *A well-formed eHealth protocol EHP satisfies enforced prescribing-privacy, if there exist processes $init'_{dr}$ and P'_{dr}, such that:*

$$
\begin{aligned}
1)\quad & \mathcal{C}[\big(init'_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{A}/Id_{dr}\})\big) \mid \\
& \quad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{a}/presc\})\big)] \\
\approx_\ell\ & \mathcal{C}[\big((init_{dr}\{\mathtt{A}/Id_{dr}\})^{\mathtt{chc}}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid (main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{a}/presc\})^{\mathtt{chc}})\big) \mid \\
& \quad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{b}/presc\})\big)];
\end{aligned}
$$

$$
\begin{aligned}
2)\quad & init'_{dr}\{\mathtt{A}/Id_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}.(P'_{dr}\{\mathtt{A}/Id_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}) \\
\approx_\ell\ & init_{dr}\{\mathtt{A}/Id_{dr}\}.(main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{b}/presc\}),
\end{aligned}
$$

where $init'_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{A}/Id_{dr}\})$ is a closed plain process, and $\mathtt{chc}$ *is a fresh channel name.*

3.3 Independency of prescribing-privacy

Usually, eHealth systems have to deal with a complex constellation of roles: doctors, patients, pharmacists, insurance companies, medical administration, etc. Each of these roles has access to different private information and has different privacy concerns. An untrusted role may be bribed to reveal private information to the adversary such that the adversary can break another roles' privacy. De Decker et al. [9] also note that preserving doctor privacy is not sufficient to prevent bribery: pharmacists could act as *go-betweens*. For instance, pharmacists may have sensitive data which can be revealed to the adversary to break a doctor's prescribing-privacy. To prevent a party (not a doctor) to do this, eHealth protocols are required to satisfy *independency of prescribing-privacy*, meaning that even if another party R_i reveals their information (i.e., $R_i^{\mathtt{chc}}$), the adversary should not be able to break a doctor's prescribing-privacy.

Definition 5 (Independency of prescribing-privacy). *A well-formed eHealth protocol EHP satisfies* prescribing-privacy independent of role R_i *if*

$$\begin{aligned}
&\mathcal{C}[!R_i^{\mathtt{chc}} \mid \big(init_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{a}/presc\})\big) \mid \\
&\qquad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{b}/presc\})\big)] \\
\approx_\ell\ &\mathcal{C}[!R_i^{\mathtt{chc}} \mid \big(init_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{b}/presc\})\big) \mid \\
&\qquad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{a}/presc\})\big)].
\end{aligned}$$

where R_i is a non-doctor role.

Note that we assume a worst situation in which a pharmacist genuinely cooperates with the adversary. For example, the pharmacist forwards all information obtained from channels hidden from the adversary.

3.4 Independency of enforced prescribing-privacy

We have discussed two situations where a doctor prescription behaviour can be revealed when either the doctor or another different party cooperates with the adversary. It is natural to consider the conjunction of these two, i.e., a situation in which the adversary coerces both a doctor and another party (not a doctor). Since the adversary obtains more information, this constitutes a stronger attack on doctor's prescribing-privacy. To address this problem, we define *independency of enforced prescribing-privacy*, which is satisfied when a doctor's prescribing-privacyis preserved even if both the doctor and another party reveal their private information to the adversary.

Definition 6 (Independency of enforced prescribing-privacy). *A well-formed eHealth protocol EHP satisfies* enforced prescribing-privacy independent of role R_i*, if there exist processes $init'_{dr}$ and P'_{dr}, such that:*

$$\begin{aligned}
1)\quad &\mathcal{C}[!R_i^{\mathtt{chc}} \mid \big(init'_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{A}/Id_{dr}\})\big) \mid \\
&\qquad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{a}/presc\})\big)] \\
&\approx_\ell \mathcal{C}[!R_i^{\mathtt{chc}} \mid \big((init_{dr}\{\mathtt{A}/Id_{dr}\})^{\mathtt{chc}}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid (main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{a}/presc\})^{\mathtt{chc}})\big) \mid \\
&\qquad \big(init_{dr}\{\mathtt{B}/Id_{dr}\}.(!P_{dr}\{\mathtt{B}/Id_{dr}\} \mid main_{dr}\{\mathtt{B}/Id_{dr}, \mathtt{b}/presc\})\big)];
\end{aligned}$$

$$\begin{aligned}
2)\quad &init'_{dr}\{\mathtt{A}/Id_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}.(P'_{dr}\{\mathtt{A}/Id_{dr}\}^{\backslash \mathtt{out}(\mathtt{chc},\cdot)}) \\
&\approx_\ell init_{dr}\{\mathtt{A}/Id_{dr}\}.(main_{dr}\{\mathtt{A}/Id_{dr}, \mathtt{b}/presc\}),
\end{aligned}$$

where $init'_{dr}\{\mathtt{A}/Id_{dr}\}.(!P_{dr}\{\mathtt{A}/Id_{dr}\} \mid P'_{dr}\{\mathtt{A}/Id_{dr}\})$ is a closed plain process, R_i is a non-doctor role, and chc *is a fresh channel name.*

We conjecture that independency of enforced prescribing-privacy implies independency of prescribing-privacy and enforced prescribing-privacy, each of which also implies prescribing-privacy.

3.5 Anonymity and strong anonymity

Anonymity is a property that protect users' identities. We model anonymity as indistinguishability of processes initiated by two different users.

Definition 7 (Doctor anonymity). *A well-formed eHealth protocol EHP satisfies doctor anonymity for a doctor* $\mathtt{A}$ *if there exists another doctor* $\mathtt{B}$*, such that*

$$\mathcal{C}[init_{dr}\{\mathtt{A}/Id_{dr}\}.!P_{dr}\{\mathtt{A}/Id_{dr}\}] \approx_\ell \mathcal{C}[init_{dr}\{\mathtt{B}/Id_{dr}\}.!P_{dr}\{\mathtt{B}/Id_{dr}\}].$$

A stronger notion of anonymity is defined in [17], capturing the situation that the adversary cannot even find out whether a user (with identity $\mathtt{A}$) has participated in a session of the protocol or not.

Definition 8 (Strong doctor anonymity [17]). *A well-formed eHealth protocol EHP satisfies strong doctor anonymity, if*

$$EHP \approx_\ell \nu\tilde{m}.init.\big(!R_1 \mid \ldots \mid !R_n \mid (init_{dr}\{\mathtt{A}/Id_{dr}\}.!P_{dr}\{\mathtt{A}/Id_{dr}\})\big).$$

Similarly, we can define anonymity and strong anonymity for patient and other roles in an eHealth protocol, by simply replacing the doctor role with a different role.

3.6 Untraceability and strong untraceability

Untraceability is a property preventing the adversary from tracing a user. It is defined as the adversary cannot tell whether two executions are initiated by the same user.

Definition 9 (Doctor untraceability). *A well-formed eHealth protocol EHP satisfies doctor untraceability if, for any two doctors* $\mathtt{A}$ *and* $\mathtt{B} \neq \mathtt{A}$*,*

$$\begin{aligned}&\mathcal{C}[init_{dr}\{\mathtt{A}/Id_{dr}\}.(P_{dr}\{\mathtt{A}/Id_{dr}\} \mid P_{dr}\{\mathtt{A}/Id_{dr}\})]\\ \approx_\ell\ &\mathcal{C}[(init_{dr}\{\mathtt{A}/Id_{dr}\}.P_{dr}\{\mathtt{A}/Id_{dr}\}) \mid (init_{dr}\{\mathtt{B}/Id_{dr}\}.P_{dr}\{\mathtt{B}/Id_{dr}\})].\end{aligned}$$

A stronger notion of untraceability is proposed in [17] that captures the adversary's inability to distinguish the situation in which one user executes the protocol multiple times from a situation in which no user executes the protocol more than once.

Definition 10 (Strong doctor untraceability [17]). *A well-formed eHealth protocol EHP satisfies strong doctor untraceability, if*

$$EHP \approx_\ell \nu\tilde{m}.init.\big(!R_1 \mid \ldots \mid !R_{i-1} \mid !R_{i+1} \mid !R_n \mid !(\nu\mathtt{Id}_{dr}.init_{dr}.P_{dr})\big).$$

Similarly, we can define untraceability and strong untraceability for patient and other roles in an eHealth protocol, by simply replacing the doctor role with a different role.

4 Description and modelling of the DLVV08 protocol

De Decker. et al develop a complex healthcare protocol for the Belgium situation [9], which captures most aspects of the current Belgian healthcare practice and aims to provide a strong guarantee of privacy for patients and doctors.

To ensure security and privacy properties, the DLVV08 protocol employs cryptographic primitives such as privacy-preserving credential systems and verifiable public key cryptography. We briefly describe the used primitives and explain how to model them in applied pi. Then we briefly discuss the DLVV08 protocol and focus on the modelling of two sub-protocols in details.

4.1 Cryptographic primitives

Zero-knowledge proofs. A zero-knowledge proof (ZKP) is a cryptographic scheme in which one party (the prover) proves to another party (the verifier) that a statement is true, without leaking any information on the statement. A ZKP scheme can be either interactive or non-interactive. We model non-interactive ZKPs as $\mathsf{zk}(secrets, pub\_info)$, where $secrets$ models private information and pub_info models public information [22, 23]. Verification of a ZKP is modelled as $\mathsf{Vfy\text{-}zk}(\mathsf{zk}(secrets, pub\_info), verif\_info)$, with a proof $\mathsf{zk}(secrets, pub\_info)$ to be verified, and some verification information $verif\_info$. Since the private information in a ZKP is known only by the prover, only he can construct a correct ZKP. To verify a ZKP is to check whether a specific relation between the secret information and the verification formation is satisfied. Since pub_info and $verif\_info$ happen to be the same in all ZK proof verifications in this paper, the generic structure of a verification is $\mathsf{Vfy\text{-}zk}(\mathsf{zk}(x, \mathsf{f}(x, y)), \mathsf{f}(x, y)) = \mathsf{true}$, where x denotes private information and y denotes public information.

In DLVV08, both anonymous authentication and verifiable encryption are essentially ZKPs. Anonymous authentication is modelled as a ZKP with a credential as public information, while verifiable encryption is modelled as a ZKP with the encrypted message as part of the public information. The specific function to check a ZKP of type x is denoted as $\mathsf{Vfy\text{-}zk}_x$, e.g., verification of a patient's anonymous authentication is modelles by function $\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{pt}}$.

Signed proofs of knowledge. Signed proofs of knowledge uses proofs of knowledge as a digital signature scheme (for details see [24]). Intuitively, a prover signs a message using some private information, which can be considered as a secret signing key. The prover uses a proof of knowledge to convince the verifier that he possesses the private signing "key". We denote a signed proof of knowledge as $\mathsf{spk}(secrets, pub\_info, msg)$, which models a message *msg* and public verification information pub_info signed with signing key *secrets* [25]. What knowledge is proven depends on the instance of the proof and is captured by the verification functions for the specific proofs. These proofs are verified by checking that the signature is correct given the signed message and the verification information, generically: $\mathsf{Vfy\text{-}spk}\left(\mathsf{spk}\left(x, \mathsf{f}(x, y), m\right), \mathsf{f}(x, y), m\right) = \mathsf{true}$. Note that specific verification functions depend on the proof to be verified.

Further cryptographic primitives used. A *digital credential* proves that the owner possesses some specific properties. We model a doctor credential as a private function drcred with the doctor's private information as parameter. Similarly, a patient's credential is modelled as a private function ptcred. Functions getpublic, getSpkVinfo and getSpkMsg model retrieving public information from a ZKP, from a signed proof of knowledge, and obtaining the message from a signed proof of knowledge, respectively. Bit-commitments, hash functions, encryptions and signing messages are modelled by functions commit, hash, enc, and sign, respectively. Correspondingly, opening a commitment, decryption and retrieving the message from a signature are modelled as functions open, dec and getsignmsg.

4.2 Description of the DLVV08 protocol

The protocol involves five roles: doctor, patient, pharmacist, medicine prescription administrator (MPA) and health insurance institute (HII).

– A doctor has an identity ($\mathtt{Id}_{dr}$), a pseudonym ($\mathtt{Pnym}_{dr}$), and an anonymous doctor credential ($Cred_{dr}$) issued by trusted authorities.
– A patient has an identity ($\mathtt{Id}_{pt}$), a pseudonym ($\mathtt{Pnym}_{pt}$), an HII (Hii), a social security status (Sss), a health expense account (Acc) and an anonymous patient credential ($Cred_{pt}$) issued by trusted authorities.
– Pharmacists, MPA, and HII are public entities, each of which has an identity ($\mathtt{Id}_{ph}$, $\mathtt{Id}_{mpa}$, $\mathtt{Id}_{hii}$), a secret key ($\mathtt{sk}_{ph}$, $\mathtt{sk}_{mpa}$, $\mathtt{sk}_{hii}$) and an authorised public key certificate ($\mathtt{pk}_{ph}$, $\mathtt{pk}_{mpa}$, $\mathtt{pk}_{hii}$) issued by trusted authorities.

The DLVV08 protocol works as follows: a doctor prescribes medicines to a patient; next the patient obtains medicine from a pharmacist according to the prescription; following that, the pharmacist forwards the prescription to his MPA, the MPA checks the prescription and refunds the pharmacist; finally, the MPA sends invoices to the patient's HII and is refunded.[1] Each step is described as a sub-protocol in [9]. Due to space limitations and the fact that the studied privacy properties mainly involve doctors, patients and pharmacists, we focus on the first two sub-protocols: the doctor-patient sub-protocol and the patient-pharmacist sub-protocol.

4.3 Underspecification of the DLVV08 protocol

The DLVV08 protocol leaves the following issues unspecified:

a1 whether a zero-knowledge proof is transferable;
a2 whether an encryption is probabilistic;
a3 whether a patient/doctor uses a fresh identity and/or pseudonym for each session;
a4 whether credentials are freshly generated in each session;
a5 what a patient's social security status is and how it can be modified;
a6 how many HIIs exist and whether a patient can change his HII;
a7 whether a patient/doctor can obtain a credential by requesting one;
a8 what type of communication channels are used (public or untappable).

To be able to discover potential flaws on privacy, we make the following (weakest) assumptions in our modelling of the DLVV08 protocol:

s1 the zero-knowledge proofs used are non-interactive and transferable;
s2 encryptions are not probabilistic;
s3 a patient/doctor uses the same identity and pseudonym in every session;
s4 a patient/doctor has the same credential in every session;
s5 a patient's social security status is the same in every session;
s6 there are many HIIs, different patients may have different HIIs, and a patient's HII is fixed and cannot be changed;
s7 a patient/doctor's credential can be obtained by requesting one;
s8 the communication channels are public.

[1] As we do not focus on properties such as revocability and reimbursement, we do not consider the other two roles: public safety organisation (PSO) and social security organisation (SSO).

4.4 Modelling the doctor-patient sub-protocol

This sub-protocol is used for a doctor, whose steps are labelled $\mathbf{d}i$ in Fig. 1, to prescribe medicine for a patient, whose steps are labelled $\mathbf{t}i$ in Fig. 2.

First, the doctor anonymously authenticates to the patient using credential $Cred_{dr}$ (**d1**). The patient reads in the doctor authentication (**t1**), obtains the doctor credential (**t2**), and verifies the authentication (**t3**). If the verification in step (**t3**) succeeds, the patient anonymously authenticates himself to the doctor using his credential (**t5**, the first zk function), generates a nonce r_{pt} (**t4**), computes a commitment with the nonce as opening information, and proves that the patient identity used in the patient credential is the same as in the commitment, thus linking the patient commitment and the patient credential (**t5**, the second zk).

The doctor reads in the patient authentication as rcv_Auth_{pt} and the patient proof as $rcv\_PtProof$ (**d2**), obtains the patient credential from the patient authentication (**d3**), obtains the patient commitment c_Comt_{pt} and the patient credential from the patient proof, tests whether the credential matches the one embedded in the patient authentication (**d4**), then verifies the authentication (**d5**) and the patient proof (**d6**). If the verification in the previous item succeeds, the doctor generates a prescription[2] $\mathtt{presc}$ (**d7**), generates a nonce $\mathtt{r}_{dr}$ (**d8**), computes a prescription identity $PrescriptID$ (**d9**), and computes a commitment $Comt_{dr}$ using the nonce as opening information (**d10**). Next, the doctor signs the message ($\mathtt{presc}$, $PrescriptID$, $Comt_{dr}$, c_Comt_{pt}) using a signed proof of knowledge. This proves the pseudonym used in the credential $Cred_{dr}$ is the same as in the commitment $Comt_{dr}$, thus linking the prescription to the credential. The doctor sends the signed proof of knowledge together with the open information of the doctor commitment $\mathtt{r}_{dr}$ (**d10**).

The patient reads in the prescription as $rcv\_PrescProof$ and the opening information of the doctor commitment (**t6**), obtains the prescription c_presc, prescription identity $c\_PrescriptID$, doctor commitment c_Comt_{dr}, and tests the patient commitment signed in the receiving message (**t7**). Then the patient verifies the signed proof of prescription (**t8**). If the verification succeeds, the patient obtains the doctor's pseudonym c_Pnym_{dr} by opening the doctor commitment (**t9**).

$$
\begin{array}{ll}
 & \mathtt{let}\ P_{dr} = \\
\mathbf{d1}. & \quad \mathsf{out}(\mathtt{ch}, \mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr}))); \\
\mathbf{d2}. & \quad \mathsf{in}(\mathtt{ch}, (rcv\_Auth_{pt}, rcv\_PtProof)); \\
\mathbf{d3}. & \quad \mathtt{let}\ c\_Cred_{pt} = \mathsf{getpublic}(rcv\_Auth_{pt})\ \mathtt{in} \\
\mathbf{d4}. & \quad \mathtt{let}\ (c\_Comt_{pt}, = c\_Cred_{pt}) = \mathsf{getpublic}(rcv\_PtProof)\ \mathtt{in} \\
\mathbf{d5}. & \quad \mathtt{if}\ \mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{pt}}(rcv\_Auth_{pt}, c\_Cred_{pt}) = \mathsf{true}\ \mathtt{then} \\
\mathbf{d6}. & \quad \mathtt{if}\ \mathsf{Vfy\text{-}zk}_{\mathsf{PtProof}}(rcv\_PtProof, (c\_Comt_{pt}, c\_Cred_{pt})) = \mathsf{true}\ \mathtt{then} \\
\mathbf{d7}. & \quad \nu\mathtt{presc}; \\
\mathbf{d8}. & \quad \nu\mathtt{r}_{dr}; \\
\mathbf{d9}. & \quad \mathtt{let}\ PrescriptID = \mathsf{hash}(\mathtt{presc}, c\_Comt_{pt}, \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}))\ \mathtt{in} \\
\mathbf{d10}. & \quad \mathsf{out}(\mathtt{ch}, (\mathsf{spk}((\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}, \mathtt{Id}_{dr}), \\
 & \qquad\qquad (\mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{Id}_{dr})), \\
 & \qquad\qquad (\mathtt{presc}, PrescriptID, \mathsf{commit}(\mathtt{Pnym}_{dr}, \mathtt{r}_{dr}), c\_Comt_{pt})), \\
 & \qquad \mathtt{r}_{dr})).
\end{array}
$$

Fig. 1. The doctor process P_{dr}.

[2] A medical examination of the patient is not part of the DLVV08 protocol.

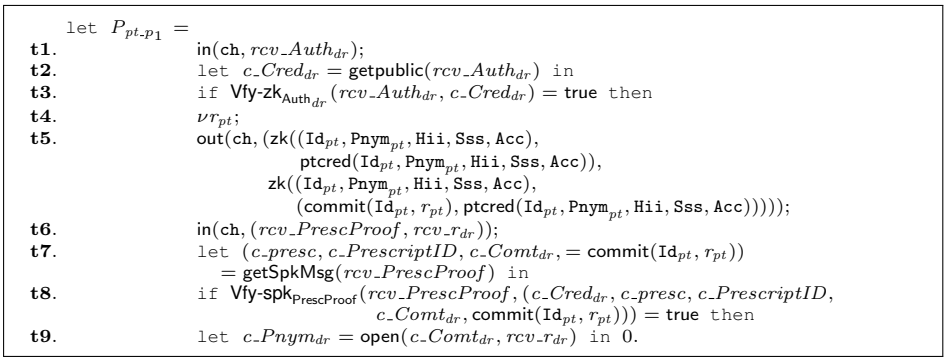

Fig. 2. The patient process P_{pt} in doctor-patient sub-protocol.

4.5 Modelling the patient-pharmacist sub-protocol

This sub-protocol is used for a patient, whose steps are labelled **t**i in Fig. 3, to obtain medicine from a pharmacist, whose steps are labelled **h**i in Fig. 4.

First, the pharmacist authenticates to the patient using a public key authentication (**h1**). Note that the pharmacist does not authenticate anonymously, and that the pharmacists's MPA identity is embedded. The patient reads in the pharmacist authentication rcv_Auth_{ph} (**t10**) and verifies the authentication (**t11**). If the verification succeeds, the pharmacist obtains the pharmacist's MPA identity from the authentication (**t12**), thus obtains the public key of MPA (**t13**). Then the patient anonymously authenticates himself to the pharmacist, and proves his social security status using the proof $PtAuthSss$ (**t14**). The patient generates a nonce which will be used in a signed proof of knowledge (**t15**), and computes verifiable encryptions vc_1, vc_2, vc_3, vc'_3, vc_4 and vc_5 (**t16-t21**). These divulge the patient's HII, the doctor's pseudonym, and the patient's pseudonym to the MPA, the patient's pseudonym to the HII, and the patient pseudonym and HII to the social safety organisation, respectively. The patient encrypts vc_5 with MPA's public key as c_5 (**t22**). The patient computes a signed proof of knowledge $PtSpk$ which proves that the patient identity embedded in the prescription is the same as in his credential[3]. The patient sends the prescription $rcv\_PrescProof$, the signed proof $PtSpk$, and vc_1, vc_2, vc_3, vc'_3, vc_4, c_5 to the pharmacist (**t23**). The pharmacist reads in the authentication $rcv\_PtAuthSss$ (**h2**), obtains the patient credential and his social security status (**h3**), verifies the authentication (**h4**). If the verification succeeds, the pharmacist reads in the patient's prescription $rcv_{ph}\_PrescProof$, the signed proof of knowledge rcv_{ph}_PtSpk, the verifiable encryptions rcv_vc_1, rcv_vc_2, rcv_vc_3, $rcv\_vc'_3$, rcv_vc_4, and cipher text rcv_c_5 (**h5**); and verifies $rcv_{ph}\_PrescProof$ (**h6-h8**), rcv_{ph}_PtSpk (**h9-h10**), and rcv_vc_1, rcv_vc_2, rcv_vc_3, $rcv\_vc'_3$, rcv_vc_4 (**h11-h20**). If all the verifications succeed, the pharmacist charges the patient, and delivers the medicine (neither are

[3] In the prescription, this identity is contained in a commitment. For simplicity, we model the proof using the commitment instead of the prescription. The link between commitment and prescription is ensured when the proof is verified (**h10**).

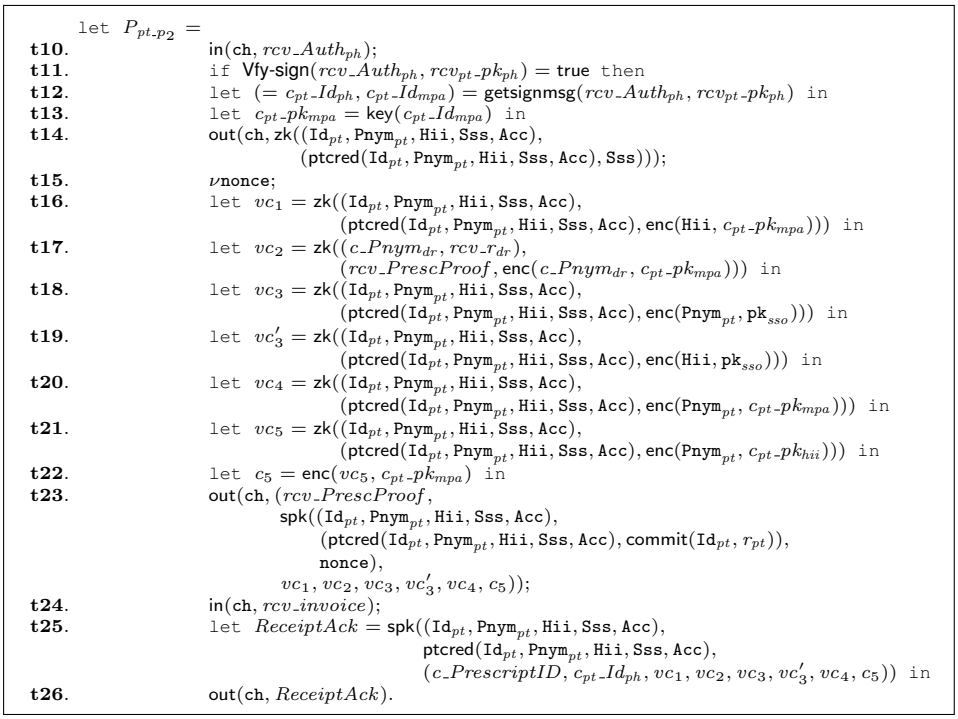

Fig. 3. The patient process P_{pt} in patient-pharmacist sub-protocol.

modelled as they are out of DLVV08's scope). Then the pharmacist generates an invoice with the prescription identity embedded in it and sends the invoice to the patient (**h21**).

The patient reads in the invoice (**t24**), computes a receipt: a signed proof of knowledge $ReceiptAck$ which proves that he receives the medicine (**t25**); and sends the signed proof of knowledge to the patient (**t26**). The pharmacist reads in the receipt $rcv\_ReceiptAck$ (**h22**) and verifies its correctness (**h23**).

4.6 Claimed privacy properties

The DLVV08 protocol claims to satisfy the following privacy properties:

- Prescribing-privacy: the protocol protects a doctor's prescription behaviour.
- Enforced prescribing-privacy: the protocol prevents pharmaceutical companies from rewarding doctors for prescribing their medicine.
- Independency of prescribing-privacy: pharmacists are not able to provide evidence to pharmaceutical companies about doctors' prescription.
- Patient anonymity: no party should be able to determine a patient's identity.
- Patient untraceability: prescriptions issued to the same patient should not be linkable to each other.

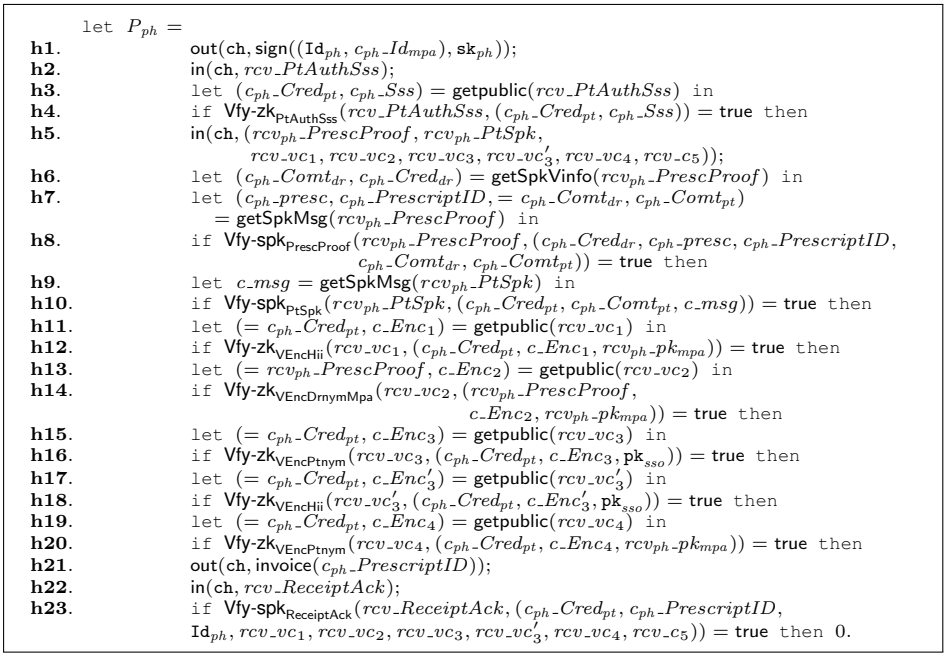

Fig. 4. The pharmacist process P_{ph} in Patient-Pharmacist sub-protocol.

5 Analysis

In this section, we analyse (enforced) prescribing-privacy, independence of (enforced) prescribing-privacy, (strong) patient and doctor anonymity, (strong) patient and doctor untraceability of the DLVV08 protocol, under the assumptions stated in Sect. 4.3. Doctor anonymity and untraceability are not required by the protocol but are still interesting to analyse. The verification results are summarised in Tab. 1.

The above privacy properties are modelled using equivalences in the applied pi calculus (see Sect. 3). To verify them is to check the satisfiability of the corresponding equivalence between processes, which can be captured by a bi-process and automatically checked in the tool ProVerif [26]. A bi-process models two processes sharing the same structure and differing only in terms or destructors. The two processes are written as one process with choice-constructors which tells ProVerif the spots where the two processes differ. For example, choice$[x, y]$ means that the first process uses x to replace choice$[x, y]$ while the second process uses y. The context $\mathcal{C}$ in the DLVV08 protocol for the analysis of privacy properties is defined as $\mathcal{C} = \nu\tilde{m}.init.(!R_{pt} \mid !R_{dr} \mid !R_{ph} \mid \_)$.

5.1 Prescribing-privacy

The verification result shows that the DLVV08 protocol does not satisfy prescribing-privacy, i.e., the adversary can distinguish whether a prescription is prescribed by doctor A or doctor B. In the prescription proof, a prescription is linked to a doctor credential. And a doctor credential is linked to a doctor identity. Thus, the adversary can link a

checked privacy property	initial model	cause(s)	improvement	revised model
prescribing-privacy	$\times$	**s4**	**s4'**	$\surd$
enforced presc.-priv.	$\times$ (with **s4'**)		**s8'**	$\surd$
independency of presc.-priv.	$\surd$ (with **s4'**)			$\surd$
independency of enforced presc.-priv.	$\times$(with **s4'**)		**s8'**	$\times$
patient anonymity	$\surd$			$\surd$
strong patient anonymity	$\surd$			$\surd$
doctor anonymity	$\times$	**s4**	**s4'**	$\surd$
strong doctor anonymity	$\times$	**s4**	**s4'**	$\surd$
patient untraceability	$\times$	**s2**, **s4**, **s5**, **s6**	**s2'**, **s4''**, **s5'**, **s6'**	$\surd$
strong patient untraceability	$\times$	**s2**, **s4**, **s5**, **s6**	**s2'**, **s4''**, **s5'**, **s6'**	$\surd$
doctor untraceability	$\times$	**s3**	**s3'**	$\surd$
strong doctor untraceability	$\times$	**s3**	**s3'**	$\surd$

Table 1. Verification results of the DLVV08 protocol with original and revised assumptions.

doctor to his prescription. To break the link, one way is to make sure that the adversary cannot link a doctor credential to a doctor identity. This can be achieved by adding randomness to the credential (**s4'**).

5.2 Enforced prescribing-privacy

The definition of enforced prescribing-privacy is modelled as the existence of a process P'_{dr}, such that the two equivalences in Def. 4 are satisfied. Due to the existence quantification, we cannot verify the property directly using ProVerif – as we can use ProVerif to verify equivalences, but not to show the existence of such processes.

By examining the DLVV08 protocol, we find an attack on enforced prescribing-privacy, even after fixing prescribing privacy (with assumption **s4'**). A bribed doctor is able to prove to the adversary of his prescription as follows:

1. A doctor communicates with the adversary (a pharmaceutical company) to agree on a bit-commitment that he will use, which links the doctor to the commitment.
2. The doctor uses the agreed bit-commitment in the communication with his patient. This links the bit-commitment to a prescription.
3. Later, when the patient uses this prescription to get medicine from a pharmacist, the adversary can observe the prescription being used. This proves that the doctor has really prescribed the medicine.

Formally, using ProVerif, we can show that if a doctor reveals all his information to the adversary, the doctor's prescribing-privacy is broken. To prove that there exist no alternative precesses for a doctor to cheat the adversary, we assume that there exists a process P'_{dr} which satisfies the definition of enforced prescribing-privacy, and then derive some contradiction. A bribed doctor reveals the nonces used in the commitment and the credential to the adversary. Thus, the adversary links a bribed doctor to his commitment and credential. In the prescription proof, a prescription is linked to a doctor's commitment and credential. Suppose there exists a process P'_{dr} in which the doctor lies to the adversary that he prescribed `a`, while the adversary observes that the commitment or the credential is linked to `b`. The adversary can detect that the doctor has lied.

5.3 Doctor's (enforced) prescribing-privacy independent of pharmacist

The doctor's prescribing-privacy independent of the pharmacist is modelled by replacing R_i with R_{ph} in Def. 5. The verification result shows that the protocol (after fixing the flaw on prescribing-privacy with assumption **s4'**) satisfies this property.

Similarly, the doctor's enforced prescribing-privacy independent of pharmacist is defined as replacing R_i with R_{ph} in Def. 6. The flaw described in the previous section is also applied here. Intuitively, when a doctor is able to prove his prescription without the pharmacist sharing information with the adversary, the doctor can certainly prove it when the pharmacist genuinely cooperates with the adversary.

5.4 (Strong) patient and doctor anonymity

Our verification results show that the protocol satisfies patient anonymity and strong patient anonymity but not doctor anonymity, nor strong doctor anonymity (see Def. 7 and Def. 8).

For strong doctor anonymity, the adversary can distinguish a process initiated by an unknown doctor and a known doctor. Given a doctor process, where the doctor's identity is $\mathtt{A}$ and his pseudonym is $\mathtt{Pnym}_{dr}$, his credential is $\mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{A})$. $\mathtt{Pnym}_{dr}$ and $\mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{A})$ are revealed. We assume that the adversary knows another doctor identity $\mathtt{B}$. The adversary can fake an anonymous authentication by faking the zero-knowledge proof as $\mathsf{zk}((\mathtt{Pnym}_{dr}, \mathtt{B}), \mathsf{drcred}(\mathtt{Pnym}_{dr}, \mathtt{A}))$. If the zero-knowledge proof passes the corresponding verification $\mathsf{Vfy\text{-}zk}_{\mathsf{Auth}_{dr}}$ by the patient, then the adversary knows that the doctor process is executed by the doctor $\mathtt{B}$. Otherwise, not.

For the same reason, doctor anonymity fails the verification. Both flaws can be fixed by requiring a doctor to generate a new credential in each session (**s4'**).

5.5 (Strong) patient and doctor untraceability

The DLVV08 protocol does not satisfy patient/doctor untraceability (see Def. 9), nor strong untraceability (see Def. 10).

The adversary can distinguish sessions initiated by one doctor and by different doctors. The doctor's pseudonym is revealed and a doctor uses the same pseudonym in all sessions. Sessions with the same doctor pseudonyms are initiated by the same doctor. For the same reason, doctor untraceability also fails. Both of them can be fixed by requiring a doctor to freshly generate his pseudonym in each session (**s3'**).

For strong patient untraceability, the adversary can distinguish sessions initiated by one patient (with identical social security statuses) and initiated by different patients (with different social security statuses). Second, the adversary can distinguish sessions initiated by one patient (with identical cipher texts $\mathsf{enc}(\mathtt{Pnym}_{pt}, \mathtt{pk}_{sso})$ and identical $\mathsf{enc}(\mathtt{Hii}, \mathtt{pk}_{sso})$) and initiated by different patients (with different cipher texts $\mathsf{enc}(\mathtt{Pnym}_{pt}, \mathtt{pk}_{sso})$ and different $\mathsf{enc}(\mathtt{Hii}, \mathtt{pk}_{sso})$). Third, since the patient credential is the same in all sessions and is revealed, the adversary can also trace a patient by the patient's credential. Fourth, the adversary can distinguish sessions using the same HII and sessions using different HIIs. For the same reasons, patient untraceability fails. Both flaws can be fixed by revising the assumptions (**s5'**, **s2'**, **s4''** and **s6'**).

5.6 Addressing the flaws of the DLVV08 protocol

To summarise, we modify assumptions in Sect. 4.3 to fix the flaws found in our analysis.

s2' The encryptions are probabilistic.
s3' A doctor's pseudonym is freshly generated in every session.
s4' A doctor freshly generates an unpredictable credential in each session. We model this with another parameter (a random number) of the credential. Following this, anonymous authentication using these credentials proves knowledge of the used randomness.
s4'' A patient freshly generates a credential in each session.
s5' A patient's social security status is different in each session.
s6' All patients share the same HII.

The modified protocol is verified again using ProVerif. The verification results show that the protocol with revised assumptions satisfies prescribing-privacy, doctor anonymity and strong anonymity, patient and doctor untraceability and strong untraceability.

To make the protocol satisfies enforced prescribing-privacy, we apply the following assumption on communication channels.

s8' The communication channels are untappable, except that communication channels for authentications remain public.

Our model of the protocol is accordingly modified as follows: replacing channel $\mathtt{ch}$ in lines **d10**, **t6** with an untappable channel $\mathtt{ch}_{dp}$, replacing channel $\mathtt{ch}$ in lines **t23**, **t26**, **h5**, **h22** with an untappable channel $\mathtt{ch}_{ptph}$, and replacing channel $\mathtt{ch}$ in lines **t24**, **h21** with an untappable channel $\mathtt{ch}_{phpt}$. We prove that the protocol (with **s4'** and **s8'**) satisfies enforced prescribing-privacy by showing the existence of a process P'_{dr} such that the equivalences in Def. 4 are satisfied.

However, with the above assumptions the DLVV08 protocol does not satisfy independency of enforced prescribing-privacy. We first show that P'_{dr} is not sufficient for proving this with ProVerif. Then we prove (analogous to the proof in Sect. 5.2) that there is no alternative process P'_{dr} which satisfies Def. 6. Intuitively, all information sent over untappable channels are received by pharmacists and can be genuinely revealed to the adversary by the pharmacists (do not lie by assumption). Hence, there still exist links between a doctor, his nonces, his commitment, his credential and his prescription, when the doctor is bribed/coerced to reveal the nonces used in the commitment and the credential to the adversary.

6 Conclusion

In this paper, we have identified new privacy requirements for eHealth systems and formalised them in the applied pi calculus. Then we took the DLVV08 protocol as a case study. We have found ambiguities in the protocol and privacy flaws as consequence, and proposed possible solutions for fixing them. We hope that our findings can help to clarify and improve the design of the DLVV08 protocol, satisfying a number of necessary privacy requirements.

References

1. Reid, J., Cheong, I., Henricksen, M., Smith, J.: A novel use of rBAC to protect privacy in distributed health care information systems. In: Proc. 8th Australian Conference on Information Security and Privacy. Volume 2727 of LNCS., Springer (2003) 403–415
2. Currim, F., Jung, E., Xiao, X., Jo, I.: Privacy policy enforcement for health information data access. In: Proc. 1st ACM Workshop on Medical-grade Wireless Networks, ACM (2009) 39–44
3. Dolev, D., Yao, A.C.C.: On the security of public key protocols. IEEE Transactions on Information Theory **29**(2) (1983) 198–207
4. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections (extended abstract). In: Proc. 26th Symposium on Theory of Computing, ACM (1994) 544–553
5. Lee, B., Kim, K.: Receipt-free electronic voting through collaboration of voter and honest verifier. In: Proc. Japan-Korea Joint Workshop on Information Security and Cryptology. (2000) 101–108
6. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: Proc. 19th Conference on the Theory and Application of Cryptographic Techniques. Volume 1807 of LNCS., Springer (2000) 539–556
7. Lee, B., Kim, K.: Receipt-free electronic voting with a tamper-resistant randomizer. In: Proc. 4th Conference on Information and Communications Security. Volume 2513 of LNCS., Springer (2002) 389–406
8. Matyáš, V.: Protecting doctors' identity in drug prescription analysis. Health Informatics Journal (3-4) (1998) 205–209
9. De Decker, B., Layouni, M., Vangheluwe, H., Verslype, K.: A privacy-preserving eHealth protocol compliant with the Belgian healthcare system. In: Proc. 5th European Workshop on Public Key Infrastructures, Services and Application. Volume 5057 of LNCS., Springer (2008) 118–133
10. Dong, N., Jonker, H.L., Pang, J.: Challenges in eHealth: from enabling to enforcing privacy. In: Proc. 1st Symposium on Foundations of Health Information Engineering and Systems. Volume 7151 of LNCS., Springer (2012) 195–206
11. Dong, N., Jonker, H.L., Pang, J.: Formal analysis of an eHealth protocol. Technical report, University of Luxembourg (2012) Report and ProVerif code are available at `http://satoss.uni.lu/naipeng/publication.php`.
12. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: Proc. 28th ACM Symposium on Principles of Programming Languages, ACM (2001) 104–115
13. Schneider, S., Sidiropoulos, A.: CSP and anonymity. In: Proc. 4th European Symposium on Research in Computer Security. Volume 1146 of LNCS., Springer (1996) 198–218
14. van Deursen, T., Mauw, S., Radomirović, S.: Untraceability of RFID protocols. In: Proc. 2nd Workshop on Information Security Theory and Practices. Smart Devices, Convergence and Next Generation. Volume 5019 of LNCS., Springer (2008) 1–15
15. Backes, M., Hriţcu, C., Maffei, M.: Automated verification of remote electronic voting protocols in the applied pi-calculus. In: Proc. 21st IEEE Computer Security Foundations Symposium, IEEE CS (2008) 195–209
16. Küsters, R., Truderung, T.: An epistemic approach to coercion-resistance for electronic voting protocols. In: Proc. 30th IEEE Symposium on Security and Privacy, IEEE CS (2009) 251–266
17. Arapinis, M., Chothia, T., Ritter, E., Ryan, M.: Analysing unlinkability and anonymity using the applied pi calculus. In: Proc. 23rd IEEE Computer Security Foundations Symposium, IEEE CS (2010) 107–121

18. Küsters, R., Truderung, T., Vogt, A.: A game-based definition of coercion-resistance and its applications. In: Proc. 23rd IEEE Computer Security Foundations Symposium, IEEE CS (2010) 122–136
19. Delaune, S., Kremer, S., Ryan, M.D.: Verifying privacy-type properties of electronic voting protocols. Journal of Computer Security **17**(4) (2009) 435–487
20. Jonker, H.L., Mauw, S., Pang, J.: A formal framework for quantifying voter-controlled privacy. Journal of Algorithms in Cognition, Informatics and Logic **64**(2-3) (2009) 89–105
21. Dong, N., Jonker, H.L., Pang, J.: Analysis of a receipt-free auction protocol in the applied pi calculus. In: Proc. 7th Workshop on Formal Aspects in Security and Trust. Volume 6561 of LNCS., Springer (2011) 223–238
22. Backes, M., Maffei, M., Unruh, D.: Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol. In: Proc. IEEE Symposium on Security and Privacy, IEEE CS (2008) 202–215
23. Li, X., Zhang, Y., Deng, Y.: Verifying anonymous credential systems in applied pi calculus. In: Proc. 8th Conference on Cryptology and Network Security. Volume 5888 of LNCS., Springer (2009) 209–225
24. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press (2000)
25. Delaune, S., Ryan, M., Smyth, B.: Automatic verification of privacy properties in the applied pi-calculus. In: Proc. 2nd Joint iTrust and PST Conferences on Privacy, Trust Management and Security. Volume 263 of IFIP Conference Proceedings., Springer (2008) 263–278
26. Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules. In: Proc. 14th IEEE Computer Security Foundations Workshop, IEEE CS (2001) 82–96