DISCOVERING EPASSPORT VULNERABILITIES USING BISIMILARITY

ROSS HORNE AND SJOUKE MAUW

Department of Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg
e-mail address: ross.horne@uni.lu

Department of Computer Science and SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg
e-mail address: sjouke.mauw@uni.lu

Abstract. We uncover privacy vulnerabilities in the ICAO 9303 standard implemented by ePassports worldwide. These vulnerabilities, confirmed by ICAO, enable an ePassport holder who recently passed through a checkpoint to be reidentified without opening their ePassport. This paper explains how bisimilarity was used to discover these vulnerabilities, which exploit the BAC protocol – the original ICAO 9303 standard ePassport authentication protocol – and remains valid for the PACE protocol, which improves on the security of BAC in the latest ICAO 9303 standards. In order to tackle such bisimilarity problems, we develop here a chain of methods for the applied π-calculus including a symbolic under-approximation of bisimilarity, called open bisimilarity, and a modal logic, called classical $\mathcal{FM}$, for describing and certifying attacks. Evidence is provided to argue for a new scheme for specifying such unlinkability problems that more accurately reflects the capabilities of an attacker.

1. Introduction

Most of us have the option to pass through automatic passport clearance at an airport. Some of us also have electronic national cards that may be used for government services. All of these machine readable documents employ a protocol to authenticate with a reader, establishing that you really hold a valid machine readable document. In order for ePassports to be read internationally, your passport almost certainly implements a standardised protocol for machine readable travel documents, defined by the International Civil Aviation Authority (ICAO) – the UN agency responsible for international aviation standards.

Considerable work has been put into ensuring ePassports satisfy security properties, preventing your ePassport from being read by an unauthorised 3^{rd}-party in the vicinity. However, even if such security properties are satisfied, there may still be ways of exploiting a protocol to mount more subtle attacks on your privacy. Notably, a requirement of ePassports, is that an unauthorised 3^{rd}-party should not be able to use an ePassport to track the document holder. This privacy property is called *unlinkability*, and is an official requirement of the ICAO 9303 standard for machine readable travel documents [MRT15].

It has been debated over the past decade whether or not the ICAO 9303 standard satisfies unlinkability. Vulnerabilities have been discovered by exploiting implementation specific features, such as the different error messages in the French ePassport, or the differences in response time

Key words and phrases: privacy, protocols, bisimilarity, modal logic, ePassports.

Author version of paper appearing in Logical Methods in Computer Science Volume 17, Issue 2, 2021. DOI:10.23638/LMCS-17(2:24)2021

of the ePassports of different nationalities [CS10, ABH^{+}16]. For example, an error message in the French ePassport indicates whether a message authentication code (MAC) test passed, despite authentication failing; hence if a message with the same MAC key is replayed from a previous session with the same ePassport, then we can detect whether or not the same ePassport is present in the current session. Notice this requires no access to the personal data stored inside the ePassport.

Now put implementation-specific side channels aside and consider whether unlinkability holds at the level of the specification of the Basic Access Control (BAC) protocol, as defined in the ICAO 9303 standard. A claim was reported in CSF'10 [ACRR10] that unlinkability does hold for ePassports that conservatively implement the BAC protocol. In particular, the claim concerns implementations where the same plaintext message should be provided for all types of error, as is the case for the UK ePassport for example. That claim was backed up by a formal model of a property that should hold if unlinkability of BAC holds, which is expressed as a bisimilarity problem in the applied π-calculus [ABF17]. The problem is that this original claim was discovered to be false, as reported in ESORICS'19 [FHMS19]. This indicates a failure of the ICAO 9303 BAC protocol to meet its own requirements.

However, this is not the end of the story behind this privacy vulnerability, which has several twists. A twist is that the original claim which we discovered to be flawed was based on a proof in ProVerif, that went through due to a bug, now resolved in Proverif. When the bug was corrected the old proof didn't go through, but no proof or counter-example was reported until ESORICS'19 [FHMS19]. This indicates the need to improve methods and tools for supporting bisimilarity checking in the applied π-calculus, so that false privacy claims about widely deployed protocols do not go undetected for a decade.

There are further twists in this story. In the effort to improve tools and methods for resolving such unlinkability problems, several alternative models of the unlinkability of the BAC protocol have been proposed over the years. Some of these models can be used to prove that there is no attack [HBD19], as first communicated in S&P'16 [HBD16]. The key difference between the original CSF'10 model, for which we discovered an attack in ESORICS'19, and the S&P'16 model, where they prove there is no attack, is the use of trace equivalence rather than bisimilarity in the latter. This is interesting, since there are few, if any, protocols and properties where the use of bisimilarity rather than trace equivalence is essential for discovering vulnerabilities, at least for a widely-deployed protocol. Furthermore, it provokes the question of which equivalence provides the appropriate attacker model: are vulnerabilities discovered using bisimilarity, but undetectable using trace equivalence exploitable; and, if so, are they perhaps less dangerous in some sense than attacks which can be described as a trace?

For the BAC protocol we have answered the question of exploitability in the positive. Using a modified reader and ePassport we have demonstrated how the distinguishing game exposed by the failure of bisimilarity can be exploited to reidentify an ePassport. That particular implementation of the vulnerability discovered using bisimilarity (there are infinitely many mutations of this attack) was reported through a responsible disclosure process to ICAO in June 2019.

ICAO issued a public response made available via numerous press reports [Del19, Lab19a, Lab19b]. In their response, ICAO make the following statement.

> "It's also important to consider here that the described issue, which could be exploited for example at border controls or at other inspection system areas, would only allow adversaries to be able to know that somebody recently passed through a passport check– and even without opening their ePassport. The personal data stored in the contactless chip, however, would not be disclosed."

Understandably, ICAO aim to contain this issue, and we have no interest in creating a scandal, only in ensuring the appropriate agencies receive accurate information. However, please note that the above statement confirms that ICAO agree the vulnerability is real and would, we quote again for emphasis, "allow adversaries to be able to know that somebody recently passed through a passport check– and even without opening their ePassport." This exactly matches our own claims about the capabilities offered to an attacker exploiting the vulnerability discovered. The word "recent" in the above context, means that the ePassport can only be tracked for as long as the attacker can keep open a session with the reader that the ePassport holder recently passed through; which, in practice, can only be a short period of time. This contrasts to more serious implementation-specific vulnerabilities, which can be exploited to track the ePassport holder indefinitely. Being able to reidentify someone within a time-limited period is nonetheless a violation of unlinkability.

It is also important that we clarify the following public response from ICAO, also, understandably, aiming to contain any fallout from a vulnerability affecting citizens using their ePassport standard worldwide.

> "ICAO and experts have thoroughly reviewed this research and their initial analysis is that it is not linked to Doc 9303 specifications in their current version. This is especially the case given that the newest Doc 9303 specifications incorporate the PACE protocol, which is considered a more secure alternative to the BAC protocol."

What the above means is that the vulnerability reported at ESORICS'19 was for the BAC protocol. The BAC protocol is the authentication mechanism used to ensure the ePassport and reader are really talking to each other before exchanging any personal data stored on the ePassport. It has been used since the first generation of ePassports, issued since 2004, and, at the time of writing, is still supported by ePassports. BAC has known security limitations, for example the keys are generated using information such as the passport number and expiry dates, which have low entropy [BFK09]. Thus there are attacks that can enable a user to compromise the *secrecy* of the personal data on an ePassport protected by BAC. For this reason, ICAO have developed the Password Authenticated Connection Establishment (PACE) protocol, addressing such vulnerabilities that can lead to data breaches. Note a data breach would also immediately compromise unlinkability, since the attacker would have direct access to the identity of the ePassport holder.

Thus, PACE is an improvement over BAC from the perspective of secrecy; however, secrecy and privacy are not the same thing. Indeed, we report here that, PACE is also vulnerable to attacks on unlinkability by adopting a similar strategy to the attacks on BAC reported in ESORICS'19. As with BAC, we can formally account for this vulnerability by showing that PACE does *not* satisfy unlinkability, formalised in terms of bisimilarity. Since ePassports implementing BAC or PACE are issued by over 150 countries[1], the impact for society of this vulnerability is current and global.

This paper is an extended version of a paper presented at ESORICS'19. In the conference version of this paper, we explained the privacy vulnerability discovered for the BAC protocol and explained how the attack can be implemented in a real-world setting. This paper complements the conference version by focusing on our methodology for analysing such unlinkability problems rather than the implementation concerns. We explain the formal methods we developed and employed to quickly discover attacks on the unlinkability of the BAC protocol, and, going beyond the ESORICS'19 paper, also the PACE protocol. To approach the bisimilarity problem behind unlinkability, we employ a game between a prover aiming to show unlinkability holds and a disprover aiming to show there is an attack on unlinkability. The prover uses symbolic techniques to try to construct a bisimulation for an under-approximation of bisimilarity (open bisimilarity); while the

[1] Gemalto on ePassport trends: `https://www.gemalto.com/govt/travel/electronic-passport-trends`

disprover aims to verify whether attacks discovered are genuine distinguishing strategies invalidating the bisimilarity problem or whether they are spurious counter-examples due to the fact that open bisimilarity in incomplete. If a spurious counter-example is discovered, then the reason why it is spurious is used to refine and resume the search for a bisimulation. If this game terminates, we should have constructed either a bisimulation (thereby proving the unlinkability property) or a modal logic formula explaining a distinguishing strategy (thereby discovering an attack on unlinkability). Another notable feature of the method in this work is that we show that unlinkability problems, which are traditionally expressed as a weak bisimilarity problem, can be reduced to a strong bisimilarity problem, thereby ensuring the transition system is image finite, considerably simplifying the problem. This is, in itself, a contribution of this work, since notions of strong bisimilarity had not previously been defined for the applied π-calculus, nor had its characteristic modal logic previously been defined, for which we provide soundness and completeness results.

How to read this paper. The focus of this paper is on analysing unlinkability properties of ePassport protocols. During the course of our discussion on unlinkability, we introduce various methods which play a role in our formal analysis. In order to follow these methods some knowledge of the π-calculus and bisimilarity is a prerequisite. It is not necessary to have knowledge of the applied π-calculus, since we introduce a state-of-the-art presentation of the semantics of the applied π-calculus facilitating the translation of recent advances in the theory of the π-calculus to the setting of the applied π-calculus. We move quickly through such definitions, in order to get to the point, which is to explain how the methods are used to discover attacks on unlinkability.

In Sections 2 and 3, we explain our methodology and how it can be used to efficiently analyse problems such as the unlinkability of ePassport protocols, thereby proving that we have closed the question of whether the original formulation of the unlinkability of ePassport protocol BAC is violated. Section 4 reflects on established notions of unlinkability, thereby making a case for instead employing a new notion of unlinkability which makes the realistic assumption that an attacker can distinguish between communications originating from different ePassport sessions. Section 5 demonstrates that there is a similar attack on the unlinkability of the latest ePassport protocol PACE. The existence of a new attack on PACE, similar to the attack we discovered on BAC, is confirmed using our methodology. We conclude in Sections 6 and 7, by acknowledging the wider discussion on ePassport privacy and unlinkability to which this paper contributes.

2. Reducing strong unlinkability to a strong bisimilarity problem

The ICAO 9303 standard recommends two authentication protocols for ePassports. The Basic Access Control protocol (BAC) was the authentication protocol originally proposed. The Password Authenticated Connection Establishment protocol (PACE) was added in the 7^{th} edition of the standard released in 2015.

In this section, here we briefly explain the BAC protocol and show how it can be modelled as processes in the applied π-calculus. We capture a version of the BAC protocol implemented in UK ePassports, as defined in CSF'10 [ACRR10]. We should clarify that the UK version of the BAC protocol captures the way countries should implement the BAC protocol; hence our analysis is not limited to UK ePassports – it applies to ePassport worldwide. We focus, in the next two sections, on a methodology that we used to discover unlinkability attacks on the BAC protocol. An analysis of the PACE protocol appears later in Sec. 5.

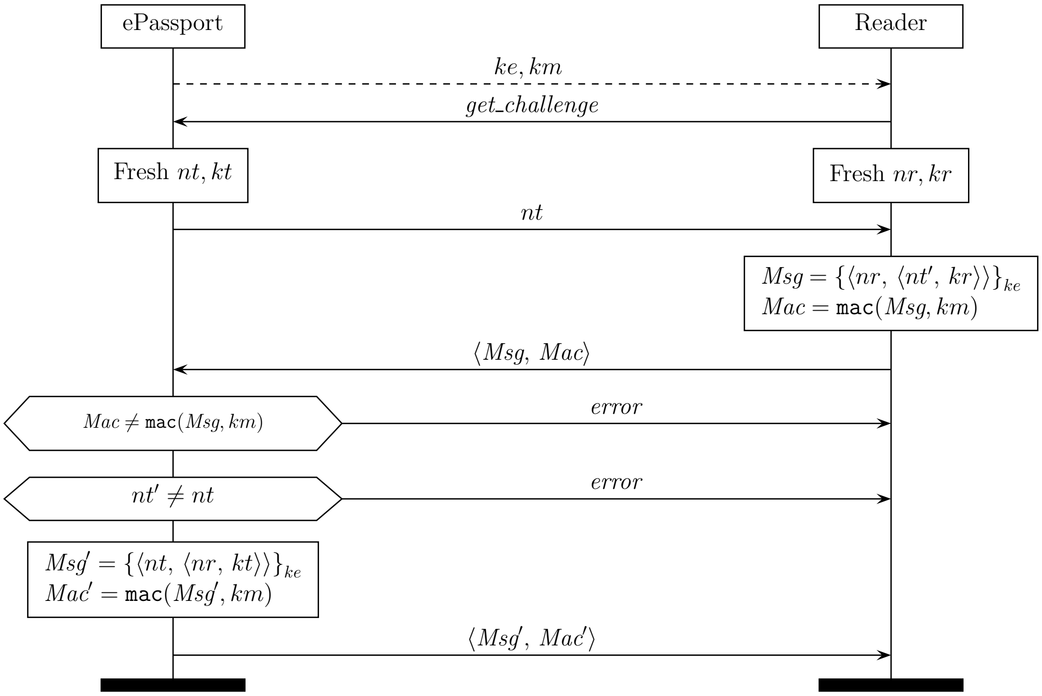
ePassport
Reader
ke, km
$get\_challenge$
Fresh nt, kt
Fresh nr, kr
nt
$Msg = \{\langle nr, \langle nt', kr\rangle\rangle\}_{ke}$
$Mac = \mathtt{mac}(Msg, km)$
$\langle Msg, Mac\rangle$
$Mac \neq \mathtt{mac}(Msg, km)$
$error$
$nt' \neq nt$
$error$
$Msg' = \{\langle nt, \langle nr, kt\rangle\rangle\}_{ke}$
$Mac' = \mathtt{mac}(Msg', km)$
$\langle Msg', Mac'\rangle$

Figure 1: The BAC protocol with one error message for all reasons for failure.

2.1. **The BAC protocol.** The BAC protocol is sketched informally in Fig. 1. Dashed lines (⇢) indicate a message transmitted via an OCR session that reads the personal page of an ePassport. The data read in the initial OCR session is used to calculate the symmetric keys *ke* and *km* used respectively for encryption and as the seed of message authentication codes (MACs). Importantly, these keys are the same for every session involving the same ePassport. Solid lines are wireless communications between a chip embedded in the ePassport and radio frequency reader.

The reader first sends a constant message *get_challenge* requesting a challenge – a nonce *nt* sent by the ePassport – which is used during the mutual authentication of the ePassport and reader. The reader shows it has the keys by responding to the challenge with a message including nonce *nt* encrypted and authenticated using the keys, thereby authenticating the reader from the perspective of the ePassport. In that message, the reader sends its own challenge *nr*, which the ePassport must respond to. The ePassport responds to the reader with a message involving nonces *nr* and *nt* encrypted and authenticated using the keys, thereby authenticating the ePassport to the reader. Notice only the ePassport that shared keys *ke* and *km* and sent challenge *nt* can respond in this way, assuming the keys are never exchanged with a malicious 3^{rd}-party.

We can be precise about the functional properties that BAC achieves. Firstly, BAC achieves an authentication property called (injective) *agreement* [Low97]. Secondly, BAC establishes shared secrets *kr* and *kt* which are used to generate a symmetric key for transmitting personal data. These properties are easily checked using automated tools such as Scyther [Cre08].

We will see that, for unlinkability, the error branches in the protocol have an important role. The ICAO 9303 standard specifies that an “operating system dependent error” [MRT15] should be sent when authentication fails. Such a failure occurs when, upon the ePassport receiving an authentication request, either the message authentication code (MAC) is wrong, or a nonce in the message does not match the challenge *nt* previously sent by the ePassport. In this work, we assume

$$
\begin{array}{rlr}
P, Q ::= & 0 & \text{deadlock} \\
\mid & \overline{M}\langle N\rangle.P & \text{send} \\
\mid & M(y).P & \text{receive} \\
\mid & \mathtt{if}\, M = N \,\mathtt{then}\, P \,\mathtt{else}\, Q & \text{branch} \\
\mid & [M = N]P & \text{match} \\
\mid & [M \neq N]P & \text{mismatch} \\
\mid & \nu x.P & \text{new} \\
\mid & P \mid Q & \text{parallel} \\
\mid & !P & \text{replication}
\end{array}
\qquad
\begin{array}{rlr}
M, N ::= & x & \text{variable} \\
\mid & \mathtt{mac}(M, N) & \text{mac} \\
\mid & \langle M, N\rangle & \text{pair} \\
\mid & \mathtt{fst}(M) & \text{left} \\
\mid & \mathtt{snd}(M) & \text{right} \\
\mid & \{M\}_N & \text{encryption} \\
\mid & \mathtt{dec}(M, N) & \text{decryption}
\end{array}
$$

$$
\begin{array}{ll}
\mathtt{fst}(\langle M,\ N\rangle) =_E M & \mathtt{snd}(\langle M,\ N\rangle) =_E N \\
\mathtt{dec}(\{M\}_K\, , K) =_E M & \{\mathtt{dec}(M, K)\}_K =_E M
\end{array}
$$

Figure 2: A syntax for applied π-calculus processes with a message theory E.

all "operating system dependent error" messages are the same, since distinctions between error messages lead to known serious attacks, such as those discovered for an implementation of the French ePassport [ACRR10,CS10]. Thus we consider the scenario where an ePassport manufacturer does not make the mistake of introducing this well known potential implementation flaw hence any attack we discover is valid for ePassport implementations worldwide.

2.2. **BAC in the applied π-calculus.** We employ the applied π-calculus to model the operational behaviour of participants in the protocol and how they are combined to form a system. The syntax of processes is presented in Fig. 2, along with a message theory featuring pairs and symmetric encryption (encryption using a shared secret key). The message theory also features a MAC function – a cryptographic hash function with no equations.

For readability, we employ the abbreviation $\mathtt{let}\, x = M \,\mathtt{in}\, P \triangleq P\left\{{}^{M}\!/_{x}\right\}$ in the following specifications of an ePassport (*MRTD*) and ePassport reader (*Reader*).

$$
\begin{array}{rl}
\mathit{MRTD} \triangleq & \overline{c_k}\langle ke, km\rangle.d(x).[x = \mathit{get\_challenge}]\nu nt.\overline{c}\langle nt\rangle.d(y). \\
& \mathtt{if}\, \mathtt{snd}(y) = \mathtt{mac}(\mathtt{fst}(y)\, , km) \;\mathtt{then} \\
& \quad \mathtt{if}\, nt = \mathtt{fst}(\mathtt{snd}(\mathtt{dec}(\mathtt{fst}(y)\, , ke))) \;\mathtt{then} \\
& \quad\quad \nu kt.\mathtt{let}\, m = \{\langle nt,\ \langle \mathtt{fst}(\mathtt{dec}(\mathtt{fst}(y)\, , ke)),\ kt\rangle\rangle\}_{ke} \;\mathtt{in} \\
& \quad\quad \overline{c}\langle m, \mathtt{mac}(m, km)\rangle \\
& \quad \mathtt{else}\, \overline{c}\langle \mathit{error}\rangle \\
& \mathtt{else}\, \overline{c}\langle \mathit{error}\rangle
\end{array}
$$

$$
\begin{array}{rl}
\mathit{Reader} \triangleq & c_k(x_k).\overline{c}\langle \mathit{get\_challenge}\rangle.d(nt).\nu nr.\nu kr. \\
& \mathtt{let}\, m = \{\langle nr,\ \langle nt,\ kr\rangle\rangle\}_{\mathtt{fst}(x_k)} \;\mathtt{in}\, \overline{c}\langle m, \mathtt{mac}(\langle m,\ \mathtt{snd}(x_k)\rangle)\rangle
\end{array}
$$

We can express the system and idealised specification, respectively, as follows.

$$
\mathit{System} \triangleq \nu c_k.(!\mathit{Reader} \mid !\nu ke.\nu km.!\mathit{MRTD})
$$

In the system above, the private channel c_k is used to initiate a session between an ePassport and Reader. The use of a private channel in this way is a modelling trick to hide all information exchanged via OCR sessions that we assume cannot be intercepted using wireless technology. Notice that the keys ke and km serve as the identity of each ePassport, since they are fixed for an ePassport when it is manufactured. Thus the innermost replication in $!\nu ke.\nu km.!\mathit{MRTD}$ models the fact that the same ePassport may be used across multiple sessions, while the outermost replication indicates that there are many different ePassports, each employing distinct keys.

2.3. **Strong unlinkability and bisimilarity.** We formulate strong unlinkability as an equivalence problem by setting out to show that *System*, as defined above, is equivalent to an idealised specification of the system that trivially satisfies unlinkability. The idealised specification models a more restricted variant of the system where each ePassport is used only once – as if, once an ePassport is read, it is destroyed and a new ePassport is issued for any future sessions. We employ the following process to model the specification.

$$\mathit{Spec} \triangleq \nu c_k.(!\mathit{Reader} \mid !\nu ke.\nu km.\mathit{MRTD})$$

Notice the only difference between *System* and *Spec* is the absence of replication after the generation of the key. Thus, in *Spec*, each new session is with a new ePassport with a freshly generated key. Trivially, there is no way to link two sessions with the same ePassport in the above specification.

We specify unlinkability by stating that it holds whenever *System* and *Spec* are equivalent from the perspective of an attacker. In principle, the idea is that, if an attacker cannot tell the difference between a scenario where the same tag is allowed to be used in multiple sessions and the scenario where each tag is really used once, then you cannot link two uses of the same tag.

In formulations of strong unlinkability, when we say "equivalent", we mean equivalent with respect to a particular notion of bisimilarity called *weak early bisimilarity*. We should avoid potential confusion of terminology: "strong" in the context of unlinkability does not refer to the process equivalence, but instead the particular formulation of unlinkability as an equivalence problem, rather than as a property of traces used in earlier work on the topic [vDMR08]. In what follows, we briefly present a formulation of weak early bisimilarity for the applied π-calculus. Our presentation makes use of processes extended with the knowledge of the attacker and an early labelled transition system which simplifies the analysis of bisimilarity problems.

We follow the convention that labelled transitions are always defined directly on extended processes in normal form. Adopting normal forms removes the need for several additional conditions that must be imposed in older formulations of bisimilarity for the applied π-calculus [ABF17].

Definition 2.1 (extended processes in normal form)**.** *Extended processes $\nu\vec{x}.(\sigma \mid P)$ consist of a set of restricted names $\vec{x}$, a substitution σ mapping variables to messages, and an applied π-calculus process P. We write $\nu x_1.\nu x_2.\dots\nu x_n.P$ as $\nu x_1, x_2, \dots x_n.P$. The set of free variables for process terms are as standard, where $\nu x.P$ and $M(x).P$ bind x in P, and process terms are always treated modulo α-conversion. We say that a variable x is fresh for a term P (processes or messages) whenever the variable does not appear free in the term, i.e., $x \notin \mathrm{fv}(P)$. A variable x is said to be fresh for a substitution σ whenever $x\sigma = x$ and, for all y, either x is fresh for $y\sigma$ or $x = y$, i.e., σ does not change or use x in any way. Freshness extends in the obvious point-wise fashion to sets of variables, terms and substitutions.*

In this work, we always assume extended processes are in normal form meaning they are subject to the restriction that the variables $\mathrm{dom}(\sigma)$ (i.e., those variables z such that $z \neq z\sigma$) are fresh for $\vec{x}$, $\mathrm{fv}(P)$ and $\mathrm{fv}(y\sigma)$, for all variables y (i.e., σ is idempotent, and substitution σ has already been applied to P). The substitution in an extended process is referred to as an active substitution.

We require the following definitions for composing extended processes in parallel and with substitutions, defined whenever z is fresh for B and ρ, and also $\mathrm{dom}(\sigma) \cap \mathrm{dom}(\theta) = \emptyset$.

$$\sigma \mid \theta \mid Q \triangleq \sigma \cdot \theta \mid Q \qquad (\sigma \mid P) \mid (\theta \mid Q) \triangleq \sigma \cdot \theta \mid (P \mid Q)$$

$$\rho \mid \nu z.A \triangleq \nu z.(\rho \mid A) \qquad B \mid \nu z.A \triangleq \nu z.(B \mid A) \qquad \nu z.A \mid B \triangleq \nu z.(A \mid B)$$

We require a standard notion of static equivalence, which checks two processes are indistinguishable in terms of the messages output so far.

$$\frac{M =_E M' \quad N =_E N'}{M(x).P \xrightarrow{M'\,N'} P\left\{^{N}/_{x}\right\}}\ \textsc{Inp} \qquad \frac{M =_E M' \quad x \text{ is fresh for } M, N, M', P}{\overline{M}\langle N\rangle.P \xrightarrow{\overline{M'}(x)} \left\{^{N}/_{x}\right\} \mid P}\ \textsc{Out}$$

$$\frac{A \xrightarrow{\pi} B \quad x \notin \mathrm{n}(\pi)}{\nu x.A \xrightarrow{\pi} \nu x.B}\ \textsc{Res} \qquad \frac{P \xrightarrow{\pi\sigma} A \quad \mathrm{bn}(\pi) \text{ is fresh for } \sigma}{\sigma \mid P \xrightarrow{\pi} \sigma \mid A}\ \textsc{Alias}$$

$$\frac{P \xrightarrow{\pi} A \qquad M =_E N}{[M = N]P \xrightarrow{\pi} A}\ \textsc{Mat} \qquad \frac{P \xrightarrow{\pi} A \qquad M =_E N}{\texttt{if}\, M = N \,\texttt{then}\, P \,\texttt{else}\, Q \xrightarrow{\pi} A}\ \textsc{Then}$$

$$\frac{P \xrightarrow{\pi} A \qquad M \neq_E N}{[M \neq N]P \xrightarrow{\pi} A}\ \textsc{Mis} \qquad \frac{Q \xrightarrow{\pi} A \qquad M \neq_E N}{\texttt{if}\, M = N \,\texttt{then}\, P \,\texttt{else}\, Q \xrightarrow{\pi} A}\ \textsc{Else}$$

$$\frac{P \xrightarrow{\pi} A \quad \mathrm{bn}(\pi) \text{ is fresh for } Q}{P \mid Q \xrightarrow{\pi} A \mid Q}\ \textsc{Par-l} \qquad \frac{P \xrightarrow{\pi} A}{!P \xrightarrow{\pi} A \mid !P}\ \textsc{Rep-act}$$

$$\frac{P \xrightarrow{\overline{M}(x)} \nu\vec{z}.\left(\left\{^{N}/_{x}\right\} \mid P'\right) \qquad Q \xrightarrow{M\,N} Q' \qquad \{x\} \cup \vec{z} \text{ are fresh for } Q}{P \mid Q \xrightarrow{\tau} \nu\vec{z}.(P' \mid Q')}\ \textsc{Close-l}$$

$$\frac{P \xrightarrow{\overline{M}(x)} \nu\vec{z}.\left(\left\{^{N}/_{x}\right\} \mid Q\right) \qquad P \xrightarrow{M\,N} R \qquad \vec{z} \text{ are fresh for } P}{!P \xrightarrow{\tau} \nu\vec{z}.(Q \mid R \mid !P)}\ \textsc{Rep-close}$$

Figure 3: An *early* labelled transition system, plus symmetric rules for parallel composition.

Definition 2.2 (static equivalence)**.** *Extended processes in normal form $\nu\vec{x}.(\sigma \mid P)$ and $\nu\vec{y}.(\theta \mid Q)$ are statically equivalent whenever, for all messages M and N such that $\vec{x} \cup \vec{y}$ are fresh for M and N, we have $M\sigma =_E N\sigma$ if and only if $M\theta =_E N\theta$.*

The above definitions are employed in our definition of "early" labelled transitions (Fig. 3), which are defined directly on extended processes in normal form. Labels on transitions are either: τ – an internal communication; $\overline{M}(z)$ – an output on channel M binding the output message to variable z; or $M\,N$ – an input on channel M receiving message N. Define the bound variables such that $\mathrm{bn}(\pi) = \{x\}$ only if $\pi = \overline{M}(x)$ and $\mathrm{bn}(\pi) = \emptyset$ otherwise. Define the free variables such that $\mathrm{fv}(M\,N) = \mathrm{fv}(M) \cup \mathrm{fv}(N)$, $\mathrm{fv}\left(\overline{M}(x)\right) = \mathrm{fv}(M) \cup \{x\}$ and $\mathrm{fv}(\tau) = \emptyset$. These sets are not disjoint, due to the context in which these definitions are used.

Notice, in this labelled transition system, `if-then-else`, match and mismatch inherit their actions from the processes they guard, which is traditional for the π-calculus. This contrasts to established reduction semantics [ABF17] for the applied π-calculus, where `if-then-else` statements perform additional τ-transitions in order to resolve guards. This design decision will enable us to provide genuine "strong" counterparts to the weak equivalences that we define.

The early labelled transition system and static equivalence together can be used to define weak early bisimilarity. Since, initially, we employ a weak formulation of early bisimilarity, we make use of weak transitions $A \overset{\pi}{\Longrightarrow} B$ which allow zero or more τ-transitions to occur before and after the transition π, or zero transitions if $\pi = \tau$.

Definition 2.3 (weak early bisimilarity)**.** *A symmetric relation between extended processes $\mathcal{R}$ is a weak early bisimulation only if, whenever $A \mathrel{\mathcal{R}} B$ the following hold:*

- *A and B are statically equivalent.*
- *If $A \xrightarrow{\pi} A'$ there exists B' such that $B \overset{\pi}{\Longrightarrow} B'$ and $A' \mathrel{\mathcal{R}} B'$.*

Processes P and Q are weak early bisimilar, written $P \approx Q$, whenever there exists a weak early bisimulation $\mathcal{R}$ such that $P \mathrel{\mathcal{R}} Q$.

Now we have the formal tools to express the theorem that confirms that strong unlinkability does not hold for the BAC protocol.

Theorem 2.4. *$System \not\approx Spec$.*

The above is the theorem rectifying the flawed claim, communicated in CSF'10 [ACRR10], that unlinkability holds for this formulation of the BAC protocol. Much of the rest of the paper is dedicated to explaining the methodology we used to prove the above result, by constructing an attack strategy invalidating the claim in Sec. 3. Later in Sec. 4 we will also make a case for adjusting the model and in Sec. 5 we will show how the analysis can be repeated for PACE.

2.4. **Reducing weak to strong bisimilarity.** A challenge with the CSF'10 [ACRR10] specification of unlinkability is that it is formulated using weak transitions, which are not image finite.

Definition 2.5. *A labelled transition system, given by a relation say $\longrightarrow$, is image finite for a process A, whenever for any label π there are finitely many B such that $A \xrightarrow{\pi} B$, up to α-conversion.*

The strong labelled transition relation $\longrightarrow$ defined in Fig. 3 is image finite for all extended processes; whereas its corresponding weak labelled transition relation $\Longrightarrow$ is only image finite for some extended processes. In particular, $\Longrightarrow$ is not image finite for processes *System* and *Spec* that are used to specify the unlinkability problem. To see this observe there are infinitely many states reachable by τ-transitions from *Spec* of the following form, where n sessions have started by communicating on the private channel c_k.

$$\begin{aligned} Spec \Longrightarrow \quad \nu c_k, ke_1, km_1, \ldots ke_n, km_n.\Big(\; & V(ke_1, km_1) \mid \ldots V(ke_n, km_n) \mid !Reader \mid \\ & P(ke_1, km_1) \mid \ldots P(ke_n, km_n) \mid !\nu ke.\nu km.MRTD \;\Big) \end{aligned}$$

where

```
$P(ke, km) \triangleq$   $d(x).[x = get\_challenge]\nu nt.\bar{c}\langle nt\rangle.d(y).$
                 if $\mathtt{snd}(y) = \mathtt{mac}(\mathtt{fst}(y), km)$ then
                     if $nt = \mathtt{fst}(\mathtt{snd}(\mathtt{dec}(\mathtt{fst}(y), ke)))$ then
                         $\nu kt.$let $m = \{\langle nt, \langle \mathtt{fst}(\mathtt{dec}(\mathtt{fst}(y), ke)), kt\rangle\rangle\}_{ke}$ in
                         $\bar{c}\langle m, \mathtt{mac}(m, km)\rangle$
                     else $\bar{c}\langle error\rangle$
                 else $\bar{c}\langle error\rangle$

$V(ke, km) \triangleq$   $\bar{c}\langle get\_challenge\rangle.d(nt).\nu nr.\nu kr.$
                 let $m = \{\langle nr, \langle nt, kr\rangle\rangle\}_{ke}$ in $\bar{c}\langle m, \mathtt{mac}(\langle m, km\rangle)\rangle$
```

When we do not have image finiteness we need to find a finite representation of infinitely many processes reachable by a transition, which can make verification challenging. To simplify verification, we show that the problem of analysing the unlinkability of BAC can be transformed into an equivalent problem where image finiteness does hold, thereby avoiding the need to explicitly deal with reasoning about transitions such as the above. The procedure we employ involves removing the τ-transition from the model – a process known as saturation – thereby reducing the unlinkability problem to a *strong early bisimilarity* problem.

We define an alternative system ***System*** and specification ***Spec***, as follows, in bold.

$$\boldsymbol{System} \triangleq\ !\nu ke.\nu km.!(V(ke, km) \mid P(ke, km))$$

$$\boldsymbol{Spec} \triangleq\ !\nu ke.\nu km.(V(ke, km) \mid P(ke, km))$$

In the above processes, the keys ke and km have been distributed in advance to the relevant parties; hence a τ-transition is not required to initiate a reader and ePassport with the same keys. We then show that each process above is bisimilar to the original system and specification, respectively. It is easier to establish a more general result, stated in the lemma below, which can be used to transform the unlinkability problem for related protocols into a form where we have image finiteness. We make use of the term $a(x_1, x_2, \ldots x_n).P$ as an abbreviation for $a(x).P\left\{{}^{\mathtt{proj}_1(x),\mathtt{proj}_2(x),\ldots\mathtt{proj}_n(x)}/_{x_1,x_2,\ldots x_n}\right\}$, where $\mathtt{proj}_i$ is the obvious generalisation of $\mathtt{fst}()$ and $\mathtt{snd}()$ to n-tuples.

Lemma 2.6. *For any P and Q such that c_k is fresh for P and Q, we have*

$$\nu c_k.\left(!c_k(\vec{k}).P \mid !\nu\vec{k}.\overline{c_k}\left\langle\vec{k}\right\rangle.Q\right) \approx\ !\nu\vec{k}.(P \mid Q)$$

Notice the proof, provided in Appendix A, is just a sketch. To go through all details would be cumbersome, indicating the amount of work that is saved when applying the above lemma to reduce the complexity of the unlinkability problem we aim to solve.

By a similar argument, we can also establish the following lemma.

Lemma 2.7. *For any P and Q such that c_k is fresh for P and Q, we have*

$$\nu c_k.\left(!c_k(\vec{k}).P \mid !\nu\vec{k}.!\overline{c_k}\left\langle\vec{k}\right\rangle.Q\right) \approx\ !\nu\vec{k}.!(P \mid Q)$$

As an immediate consequence of Lemma 2.6 and Lemma 2.7 we obtain.

Proposition 2.8. *System $\approx$ Spec if and only if **System** $\approx$ **Spec**.*

Since, in this model of the BAC protocol, all communications on public channels use channel c for outputs and d for inputs, there are no τ-transitions in ***System*** or ***Spec***. Thereby, we have reduced the problem to a form where we can apply the strong variant of early bisimilarity, defined as follows.

Definition 2.9 (strong early bisimilarity)**.** *A symmetric relation between extended processes $\mathcal{R}$ is a strong early bisimulation only if, whenever $A \mathrel{\mathcal{R}} B$ the following hold:*

- *A and B are statically equivalent.*
- *If $A \xrightarrow{\pi} A'$ there exists B' such that $B \xrightarrow{\pi} B'$ and $A' \mathrel{\mathcal{R}} B'$.*

Processes P and Q are strong early bisimilar, written $P \sim Q$, whenever there exists a strong early bisimulation $\mathcal{R}$ such that $P \mathrel{\mathcal{R}} Q$.

Notice the only difference compared to Def. 2.3 is that, in clause two of the definition above, every transition is matched by a single transition – extra τ-transitions are not permitted. The following theorem summarises the correctness of the transformation of the unlinkability problem described in this section, which is an immediate consequence of Proposition 2.8 and the absence of τ-transitions in ***System*** and ***Spec***.

Theorem 2.10. *System $\approx$ Spec if and only if **System** $\sim$ **Spec**.*

2.5. **A tribute to Jos Baeten in the language of process equivalences.** In this work, we make use of both "weak" and "strong" equivalences, since the original formulation of strong unlinkability was in terms of a weak equivalence, but strong equivalences are easier to work with. Indeed, the authors were inspired by a panel discussion during the 20^{th} edition of CONCUR chaired by Jos Baeten, to whom this paper is dedicated on the occasion of his retirement. The idea that strong equivalences are easier to work with was a point of view raised by Jos Baeten during that panel session.

The above mentioned panel session, during the 20^{th} edition of CONCUR, ended with a question from the audience, "but what can you do with all these process equivalences?" The response from a panellist was one word: "security." Indeed, this paper embodies that panel session, since we go deeper into the spectrum of process equivalences in several dimensions in order to obtain results in the security domain.

Beyond the "weak" v.s. "strong" dimension, another dimension we exploit in this work is the distinction between "early" and "open" equivalences. In the next section, we introduce a notion of "strong open" bisimilarity, which is described in terms of an "open late" labelled transition system. Traditionally, the applied π-calculus has been endowed with a notion of bisimilarity called "labelled bisimilarity," which, is little more than an alias for "weak early" bisimilarity (Def. 2.3). Our primary reason for moving from "early" to "open" is that the open setting enables symbolic methods to be directly applied hence is easier to check systematically. Open bisimilarity should however be applied carefully, since it is strictly finer than early bisimilarity; indeed, open bisimilarity is intuitionistic whereas early bisimilarity is classical [AHT17]. The significance of this insight was emphasised by Jos Baeten himself at the 28^{th} edition of CONCUR during the best paper award ceremony, indicating another way in which his leadership has influenced this paper.

One might say, "well, if it's easier to check, why not just fix open bisimilarity as the target equivalence?" This view doesn't hold up for two reasons. Firstly, the security community are used to weak early bisimilarity, so confidence is increased if we can verify which attacks discovered using open bisimilarity are also valid for weak early bisimilarity. Secondly, taking a fresh position, open bisimilarity is a little too fine for proving some security and privacy properties, so a better target equivalence for open processes (those containing free variables) would be "quasi-open" bisimilarity which balances the qualities of early bisimilarity and open bisimilarity – a discussion on this appears in a companion report [Hor18]. Thus when checking bisimilarity, we require both a notion of open bisimilarity which is easier to explore symbolically, and also a coarser equivalence such as early bisimilarity (or quasi-open bisimilarity) that serves as our target notion of bisimilarity; and, during the search for a proof or a counterexample (an attack), we play a game where we move between these equivalences. This methodology we illustrate in the next section. For the above reasons, we should be aware of how to move between "weak early", "strong early" and "strong open" variants of bisimilarity, since they come together to form a methodology for solving unlinkability problems.

Going further, we could exploit further dimensions in the spectrum of process equivalences – a point we return to in Section 4.4. In particular, we can move along the "linear-time"/ "branching-time" spectrum [vG01] to pick out coarser equivalences than bisimilarity, which can be connected with a spectrum of attacker threat models. In short, the choice of equivalence can control the testing capabilities of an attacker, which can restrict the space of attacks that we range over when we verify a security or privacy property. These intermediate definitions can be obtained by taking any of the above mentioned notions of bisimilarity and restricting them in various ways. Indeed, the linear-time/branching-time spectrum was the main topic of the aforementioned panel discussion chaired by Jos Baeten, and has been a running theme throughout his work [BBK87, ABW06, MDBdV12]. Looking beyond the current paper, there are further uncharted depths to be explored in terms of exploiting the spectrum of process equivalences to both understand attacker/threat models and to enable new methodologies for verification in the security domain. For example, all equivalences in this work "interleave" actions, but there is a spectrum of "non-interleaving" or "truly concurrent" equivalences that make explicit subtle distinctions that occur when there may be multiple attackers that are not co-located or where the duration of events is significant [BB91, BB93, BB98]. This line of inspiration, assimilated into this paper, runs back to the days when the second author was supervised by Jos Baeten at the University of Amsterdam [BBMV91], during which time the interpersonal style of Jos Baeten set a benchmark for the career of the second author.

3. Searching for a bisimulation symbolically

Having reduced unlinkability to a strong bisimilarity problem, we now aim to prove or disprove $\boldsymbol{System} \sim \boldsymbol{Spec}$. To do so, we attempt to construct a strong early bisimulation $\mathcal{R}$ (Def. 2.9) such that $\boldsymbol{System}\ \mathcal{R}\ \boldsymbol{Spec}$. However, naïvely searching for a bisimulation using the early labelled transition system in Fig. 3 is challenging, since we must consider an infinite number of messages which can be received for every input. And, although it has been shown that checking a bounded number of such messages suffices for message theories such as the one we employ, the bound on the number of messages to check is hyper-exponential [Hüt03].

For this reason, it makes sense to approach the problem using symbolic methods, for which we apply open bisimilarity which is an under-approximation of early bisimilarity – that is, if two processes are open bisimilar then they are early bisimilar, but not necessarily vice-versa. Open bisimilarity is suited to symbolic methods, since it uses a call-by-need approach to instantiating inputs where variables representing inputs are only instantiated when they are needed in order to enable a transition. Due to the fact that open bisimilarity is an under-approximation, care must be taken, since open bisimilarity however may discover certain spurious attacks for the BAC unlinkability problem.[2] Hence the use of open bisimilarity must be complemented by a methodology for verifying whether an attack discovered using symbolic methods is a real attack or not.

3.1. Open bisimilarity as a symbolic bisimilarity.

Open bisimilarity is suited to symbolic analysis of protocols, since it permits inputs to be lazily instantiated. Previously, open bisimilarity has been defined for a slightly less abstract cryptographic calculus, called the *spi-calculus* [BN07, Tiu07, AG99]. The spi-calculus is less general since it is hard wired with mechanisms for implementing specific equational theories which are abstracted away in the applied π-calculus, making the applied π-calculus more concise and allowing it to be instantiated with more theories.

For analysing the unlinkability of ePassports we require the additional power of the applied π-calculus, hence introduce a notion of open bisimilarity for the applied π-calculus. This definition of open bisimilarity is a contribution of this paper. The definitions we provide do have many features in common with notions of symbolic bisimilarity [HL95] for the applied π-calculus, particularly the work of Liu and Lin [LL12]. We should clarify that such notions of symbolic bisimilarity were never intended to capture open bisimilarity, due to their classical interpretation of constraints; their objective was to directly implement early bisimilarity (or early congruence – the largest congruence relation contained in early bisimilarity).

Open bisimilarity is defined in terms of an *open late labelled transition system*, presented in Fig. 4, where, like the early labelled transition system in Fig. 3, the rules are only well defined for

[2] The spurious counterexamples arise due to the fact that guards in `if-then-else` statements are treated intuitionistically. We leave it to related work to explain why open bisimilarity is intuitionistic [AHT17, HALT18], and what spurious examples may arise [Hor18]. We will focus here on a counterexample that is not spurious.

extended processes in normal form. We firstly explain the “open late” terminology (in comparison to “closed early”, where “closed” is the antonym for “open” in this setting).

Late v.s. early. A key difference between these labelled transition systems is that, in a *late* labelled transition system, the input labels are of the form $M(x)$, where M is a message representing a recipe for producing a channel and x is variable which acts as a placeholder for some input message. Notice in rule Inp in Fig. 3 the message input is chosen immediately (from infinitely many possible messages) and hence the input message appears on the input label; whereas, in rule oInp in Fig. 4 the input message appears as a variable. The use of a variable means that we do not need to decide immediately which messages should be chosen as inputs; instead, we can instantiate the variable later in a called-by-need fashion, possibly after several steps (subject to some constraints as we will explain below). The key rules that change to accommodate a late approach to inputs compared to the early approach are the rules oInp, oClose-l, oRep-close.

In order to accommodate the late input labels, we must change slightly the definition of the bound names and free names of a label, compared to the corresponding definition for the early labelled transition system, as follows: the bound variables are such that $\mathrm{bn}(M(x)) = \mathrm{bn}\left(\overline{M}(x)\right) = \{x\}$ and $\mathrm{bn}(\tau) = \emptyset$; while the free variables are such that $\mathrm{fv}(M(x)) = \mathrm{fv}\left(\overline{M}(x)\right) = \mathrm{fv}(M) \cup \{x\}$ and $\mathrm{fv}(\tau) = \emptyset$. These definitions are used in the rules of Fig. 4.

Open v.s. closed. The keyword *open* in the term *open late labelled transition system* refers to the fact that we allow free variables to appear. Due to the presence of free variables, we must keep track of a constraint system that determines what messages are allowed to be substituted for each free variable. We succinctly represent these constraints by keeping track of a *history* which records the order in which inputs and outputs occurred, which allows us to determine which messages had already been output before each input occurs and hence were available to use when performing an input. This avoids the possibility of a variable representing an input making use of knowledge from the future. In our representation of constraints, we also employ a set of inequalities between messages $\mathcal{D} = \{M_1 \neq N_1, M_2 \neq N_2, \ldots\}$, called a *distinction*. Distinctions are used to symbolically handle inequality constraints that typically arise due to the presence of `else` branches.

Histories are defined by grammar $h ::= \epsilon \mid h \cdot x^o \mid h \cdot M^i$, representing the order in which messages are sent and received. An annotated variable x^o means some output occurred which we refer to indirectly using an alias x where x appears in the domain of some active substitution θ which is associated with some extended process of the form $\nu\vec{y}.(\theta \mid P)$; thus $x\theta$ is the message term that is output, which possibly contains private names, i.e., variables $\vec{y}$ bound by the ν binder. The annotated variable M^i represents a larger message that has been input, which, initially is a variable, but may later be a message when the input variable is lazily instantiated. Notice, in Fig. 4, each rule carries a history and distinction that may be used to resolve the oElse rule, by providing sufficient evidence that two messages are not equal (i.e., negation is treated intuitionistically). Another key differences compared to Fig. 3 are the updating of the history in rule oRes, which has the effect of further constraining free variables such that none of them may directly refer to any private name. That is, when instantiating inputs, we may not use directly the variables $\vec{x}$ in an extended process of the form $\nu\vec{x}.(\theta \mid P)$; we may only refer to messages containing those variables indirectly via the variables in $\mathrm{dom}(\theta)$ that are used as aliases for outputs.

The definitions. The effect of histories on restricting the substitutions that may be applied, as described above, is captured formally in the following definition. Substitutions respecting histories, are key to the lazy approach of open bisimilarity.

$$\frac{M =_E M'\theta \quad x \text{ is fresh for } M, M', h, \mathcal{D}, \theta}{h, \mathcal{D} \colon \theta \mid M(x).P \xrightarrow{M'(x)} \theta \mid P}\ \textsc{oInp}$$

$$\frac{M =_E M'\theta \quad x \text{ is fresh for } M, M', N, P, h, \mathcal{D}, \theta}{h, \mathcal{D} \colon \theta \mid \overline{M}\langle N \rangle.P \xrightarrow{\overline{M'}(x)} \theta \mid \left\{{}^{N}\!/_{x}\right\} \mid P}\ \textsc{oOut}$$

$$\frac{h, \mathcal{D} \colon \theta \mid P \xrightarrow{\pi} A \qquad M =_E N}{h, \mathcal{D} \colon \theta \mid [M = N]P \xrightarrow{\pi} A}\ \textsc{oMat} \qquad \frac{h, \mathcal{D} \colon \theta \mid P \xrightarrow{\pi} A \qquad M =_E N}{h, \mathcal{D} \colon \theta \mid \texttt{if}\, M = N\, \texttt{then}\, P\, \texttt{else}\, Q \xrightarrow{\pi} A}\ \textsc{oThen}$$

$$\frac{h, \mathcal{D} \colon \theta \mid Q \xrightarrow{\pi} A \quad h, \mathcal{D}, \theta \models M \neq N}{h, \mathcal{D} \colon \theta \mid [M \neq N]Q \xrightarrow{\pi} A}\ \textsc{oMis} \qquad \frac{h, \mathcal{D} \colon \theta \mid Q \xrightarrow{\pi} A \quad h, \mathcal{D}, \theta \models M \neq N}{h, \mathcal{D} \colon \theta \mid \texttt{if}\, M = N\, \texttt{then}\, P\, \texttt{else}\, Q \xrightarrow{\pi} A}\ \textsc{oElse}$$

$$\frac{h \cdot x^o, \mathcal{D} \colon A \xrightarrow{\pi} B \quad x \text{ is fresh for } \pi,\ h,\ \mathcal{D}}{h, \mathcal{D} \colon \nu x.A \xrightarrow{\pi} \nu x.B}\ \textsc{oRes}$$

$$\frac{h, \mathcal{D} \colon \theta \mid P \xrightarrow{\pi} A \quad \mathrm{bn}(\pi) \text{ is fresh for } Q}{h, \mathcal{D} \colon \theta \mid P \mid Q \xrightarrow{\pi} A \mid Q}\ \textsc{oPar-l} \qquad \frac{h, \mathcal{D} \colon \theta \mid P \xrightarrow{\pi} A}{h, \mathcal{D} \colon \theta \mid\, !P \xrightarrow{\pi} A \mid\, !P}\ \textsc{oRep-act}$$

$$\frac{h, \mathcal{D} \colon \theta \mid P \xrightarrow{\overline{M}(x)} \theta \mid \nu\vec{z}.\left(\left\{{}^{N}\!/_{x}\right\} \mid P'\right) \quad h, \mathcal{D} \colon \theta \mid Q \xrightarrow{M(x)} \theta \mid Q' \quad \begin{array}{l} x \text{ is fresh for } h,\ \mathcal{D},\ \vec{z} \\ \vec{z} \text{ are fresh for } Q \end{array}}{h, \mathcal{D} \colon \theta \mid P \mid Q \xrightarrow{\tau} \theta \mid \nu\vec{z}.\left(P' \mid Q'\left\{{}^{N}\!/_{x}\right\}\right)}\ \textsc{oClose-l}$$

$$\frac{h, \mathcal{D} \colon \theta \mid P \xrightarrow{\overline{M}(x)} \nu\vec{z}.\left(\left\{{}^{N}\!/_{x}\right\} \mid Q\right) \qquad h, \mathcal{D} \colon \theta \mid P \xrightarrow{M(x)} R \qquad \vec{z} \cap \mathrm{fv}(P) = \emptyset}{h, \mathcal{D} \colon \theta \mid\, !P \xrightarrow{\tau} \nu\vec{z}.\left(Q \mid R\left\{{}^{N}\!/_{x}\right\} \mid\, !P\right)}\ \textsc{oRep-close}$$

Figure 4: An open late labelled transition system, plus symmetric rules for parallel composition.

Definition 3.1 (respects)**.** *Substitution σ respects h, where h is a history, whenever for all h' and h'' such that $h = h' \cdot x^o \cdot h''$, we have $x\sigma = x$, and $y \in \mathrm{fv}(h')$ implies $x \notin y\sigma$ (i.e., x is fresh for $h'\sigma$). In the above,* fv(h') *refers to the set of all variables appearing in any term in h'.*

For an example, consider the following substitutions and history.

$$\sigma = \left\{{}^{u_1,\ u_2,\ u_3}\!/_{x,\ y,\ z}\right\} \qquad \sigma' = \left\{{}^{u_3,\ u_2,\ u_1}\!/_{x,\ y,\ z}\right\} \qquad u_1^o \cdot x^i \cdot u_2^o \cdot y_i \cdot u_3^o \cdot z^i$$

Observe that, σ respects h. In contrast, σ' does not respect h, since $x\sigma' = u_3$, which is forbidden since u_3, represents an output, which, according to the history, did not occur until after the input x.

When applying a respectful substitution σ to an extended processes in normal form, with active substitution θ, we must iteratively apply the two substitutions together in order to recover an idempotent substitution, which is a requirement for normal forms. For example, consider σ defined above and θ defined as $\left\{{}^{n,\ \{x\}_a,\ \{y\}_b}\!/_{u_1,\ u_2,\ u_3}\right\}$. Notice u_2 and u_3 in the domain of active substitution θ represent aliases for messages that have been output, $\{x\}_a$ and $\{y\}_b$ respectively, where each of these messages contain variables, x and y respectively, representing inputs. Thus to find the value of z we must apply σ and θ thrice, that is $z\sigma\theta\sigma\theta\sigma\theta = \{\{n\}_a\}_c$. When σ and θ are such a respectful-active substitution pair, we always obtain an idempotent substitution after applying at most as many iterations as there are inputs in the history. The above observations explain why we require the following standard machinery for defining substitutions.

Definition 3.2 (substitutions)**.** *Given substitutions σ and θ define $\sigma \circ \theta$ to be the standard composition of substitutions (i.e., composition of functions). Acyclic substitutions σ are those for which there exists a strict partial order $\sqsubset_\sigma$ over variables such that if $y \in \mathrm{fv}(x\sigma)$ then $x \sqsubset_\sigma y$. For acyclic substitutions σ, define σ^* to be the substitution obtained by iteratively composing σ with itself until it stabilises, i.e., if $\sigma^0 = id$ (the identity substitution) and $\sigma^{n+1} = \sigma^n \circ \sigma$, then σ^* is σ^m for some m such that $\sigma^m \circ \sigma = \sigma^m$.*

Given an active substitution θ and respectful substitution σ, we can use $(\sigma \circ \theta)^*$ to obtain a new active substitution. This trick is used in the following definition of satisfaction, which is used to resolve inequalities in the labelled transition system in Fig. 4. Defining satisfaction is the reason for carrying around constraints, consisting of a history and distinction, at every step in the labelled transition system, since, for some pairs of messages, we can only determine whether they are not equal by observing that the constraints on their variables forbid them from being made equal.

Definition 3.3 (satisfaction)**.** *Satisfaction $h, \mathcal{D}, \theta \models M \neq N$ holds whenever there does* ***not*** *exist substitution σ respecting h such that:*

- *for all $K \neq L \in \mathcal{D}$, $K(\sigma \circ \theta)^* \neq_E L(\sigma \circ \theta)^*$*
- *and $M(\sigma \circ \theta)^* =_E N(\sigma \circ \theta)^*$.*

Entailment defines a notion of intuitionistic negation, which could be extracted from a Kripke semantics [Kri65], where the "reachable worlds" are those which can be reached by applying substitutions satisfying our constraints (or, equivalently, adding equalities). What is happening is that, since variables subject to constraints may occur in messages compared using equality or inequality, it is possible that we don't yet have enough information to determine whether or not two messages are equal. In general, two messages may be equal under one substitution of variables but not equal under another substitution. Hence it is useful, in this setting, to say that neither holds yet until we have more information, i.e., we do not assume the law of excluded middle.

For an example of a scenario where the law of excluded middle is violated consider, entailment $u^o \cdot y^i \cdot x_o, \{^x/_u\} \models x \neq y$. This entailment does **not** hold yet, since $\left\{^u/_y\right\}$ respects history $u^o \cdot y^i \cdot x^o$, and $y\left\{^u/_y\right\}\{^x/_u\} = x$, thus there exists a respectful substitution under which these messages are equal, and other substitutions that distinguish them. Observe also $x =_E y$ also does **not** hold yet. Thus, clearly, the law of excluded middle is violated.

In contrast to the above example, consider $y^i \cdot u^o \cdot x^o, \{^x/_u\} \models x \neq y$. This entailment holds, since there is no substitution σ respecting $y^i \cdot u^o \cdot x^o$ such that $x\sigma\theta = y\sigma\theta$, i.e., it is impossible for x and y to be made equal under any permitted substitution. In other words, it is impossible for an attacker who manufactures input y, using their knowledge at the time when y was input, to set y to be equal to private name x.

Now consider the following entailments, which make use of distinctions.

$$u^o \cdot v^o \cdot x^i \cdot y^o \cdot z^o, x \neq u, \{^{y,z}/_{u,v}\} \models x \neq y$$
$$u^o \cdot v^o \cdot x^i \cdot y^o \cdot z^o, x \neq u, \{^{y,z}/_{u,v}\} \not\models x \neq z$$

The former entailment above holds since the most general substitution σ respecting history $u^o \cdot v^o \cdot x^i \cdot y^o \cdot z^o$ such that $x\sigma\{^{y,z}/_{u,v}\} =_E y\sigma\{^{y,z}/_{u,v}\}$ is $\sigma = \{^u/_x\}$, but that substitution violates the inequality $x \neq u$. The latter entailment above does not hold since there exists substitution $\{^v/_x\}$ respecting both the history and the distinction such that $x\{^{v}/_{x}\}\{^{y,z}/_{u,v}\} = z$. Thus under the given history and distinction neither $x \neq z$ nor $x = z$ hold, i.e., the law of excluded middle is violated.

We find it insightful to present an explicit definition of reachability with respect to some substitution. This gives all the extended processes that are reachable from some extended process by applying some substitution, subject to constraints given by histories and distinctions.

Definition 3.4 (reachability)**.** *For a set of variables V, let $\sigma\restriction_V$ be the substitution restricted to the variables in V, i.e., if $x \in V$, $x\sigma\restriction_V = x\sigma$, otherwise $x\sigma\restriction_V = x$.*

Reachability $\leq$ is such that, for history h and h', distinction $\mathcal{D}$ and $\mathcal{D}'$ and extended processes in normal form A and B, we have $h, \mathcal{D}, A \leq_\sigma h', \mathcal{D}', B$ whenever the following hold:

- *$A = \nu\vec{y}.(P \mid \theta)$;*
- *σ respects h and $h' = h\sigma$;*
- *For some distinction $\mathcal{E}$, we have $\mathcal{D}' = \mathcal{D}\sigma \cup \mathcal{E}\sigma$;*
- *for all $K \neq L \in \mathcal{D} \cup \mathcal{E}$, we have $K(\sigma \circ \theta)^* \neq_E L(\sigma \circ \theta)^*$;*
- *$\vec{y}$ are fresh for σ, h, $\mathcal{D}$ and $\mathcal{E}$;*
- *and $B = \nu\vec{y}.((\sigma \circ \theta)^* \restriction_{\mathrm{dom}(\theta)} \mid P(\sigma \circ \theta)^*)$.*

Of course, the above is only well defined if $\sigma \circ \theta$ is acyclic. However, acyclicity of $\sigma \circ \theta$ is an invariant property of reachability, assuming that we start θ being *id* and, then generate θ and h from transitions of our labelled transitions system by recording inputs and outputs as they occur (to be made formal in the definition of open bisimilarity below).

Open bisimilarity $\sim_o$ can now be defined as follows, as the largest relation between processes such that there exists an open bisimulation containing the two processes, where all the free variables are treated as initial inputs. Notice this is the strong formulation of open bisimilarity.

Definition 3.5 (open bisimilarity)**.** *A symmetric relation $\mathcal{R}$ indexed by a history and distinction is an open bisimulation whenever: if $A \ \mathcal{R}^{h,\mathcal{D}}\ B$ the following hold, for x fresh for A, B, h, $\mathcal{D}$:*

- *whenever $h, \mathcal{D}, A \leq_\sigma h', \mathcal{D}', A'$ and $h, \mathcal{D}, B \leq_\sigma h', \mathcal{D}', B'$, we have $A' \ \mathcal{R}^{h',\mathcal{D}'}\ B'$.*
- *A and B are statically equivalent.*
- *If $h, \mathcal{D} \colon A \xrightarrow{\tau} A'$ there exists B' such that $h, \mathcal{D} \colon B \xrightarrow{\tau} B'$ and $A' \ \mathcal{R}^{h,\mathcal{D}}\ B'$.*
- *If $h, \mathcal{D} \colon A \xrightarrow{\overline{M}(x)} A'$, for some B', we have $h, \mathcal{D} \colon B \xrightarrow{\overline{M}(x)} B'$ and $A' \ \mathcal{R}^{h \cdot x^o,\mathcal{D}}\ B'$.*
- *If $h, \mathcal{D} \colon A \xrightarrow{M(x)} A'$, for some B', we have $h, \mathcal{D} \colon B \xrightarrow{M(x)} B'$ and $A' \ \mathcal{R}^{h \cdot x^i,\mathcal{D}}\ B'$.*

Open bisimilarity $\sim_o$ is a binary relation over processes defined such that $P \sim_o Q$ holds whenever there exists open bisimulation $\mathcal{R}$ such that $P \ \mathcal{R}^{x_1^i \cdot \ldots x_n^i}\ Q$ holds, where $\mathrm{fv}(P) \cup \mathrm{fv}(Q) \subseteq \{x_1, \ldots x_n\}$.

The second clause checks static equivalence, as in Def. 2.2; but, in contrast to early bisimilarity, due to the first clause we must check static equivalence holds under all substitutions respecting the current history and distinctions, as defined by reachability. Similarly, the clauses for transitions must be checked under all substitutions permitted by reachability. The input and output transitions update the history in order to remember which outputs were available at each moment when an input occurs, thereby constraining the permitted substitutions.

Remark 3.6 (practical benefits)**.** *At first sight, it may appear that closing under all substitutions makes open bisimilarity more difficult to check than early bisimilarity; however, the opposite is true. For many useful equational theories, such as the one featuring basic symmetric encryption used in our model of the BAC protocol, we can calculate a finite set of most general substitutions (and inequalities) that are sufficient to check in order to cover all solutions. This complexity is hidden in the definition of early bisimilarity in the use of early input transitions, where early inputs implicitly ask for all such substitutions and induced inequalities to be checked up front ...but we* *rarely know which to check at the point such inputs occur; hence when checking early bisimilarity we require backtracking that is avoided entirely for open bisimilarity. The feature of intuitionistic logic that is being exploited here is the fact that intuitionistic constraint systems are monotonic, allowing us to progressively close down the set of constraints without missing anything, whereas classical negation violates this monotonicity property.*

3.2. **Discovering unlinkability attacks by calculation.** We demonstrate our methodology, by showing how attacks on unlinkability can be discovered with minimal heuristics simply as a calculation using open bisimilarity.

The steps illustrated in the following subsections are:

3.2.1. The initialisation of two readers and an ePassport, all with the same keys, w.r.t. the system.
3.2.2. The use of respectful substitutions to refine an input to pass a simple guard, ignoring infinitely many other inputs.
3.2.3. Exploiting the game behind this bisimilarity problem, to expose a distinguishing strategy.
3.2.4. Symbolically reasoning about larger messages using the sequent calculus.

Heuristics are required only for selecting which actions to perform (points 3.2.1. and 3.2.3. above). The other steps above are calculations that could be formulated as a decision procedure, building on decision procedures for the spi-calculus [TD10]. Here we begin by starting up two readers, although a more general heuristic searching for a proof would probably start by starting up n readers in order to eventually construct an inductive definition of an open bisimulation covering the whole state space. We provide two reader sessions, since two suffice for the discovery of the particular attack highlighted.

3.2.1. *Initiate two readers with the same ePassport.* Our system, ***System***, makes the first moves by starting two reader sessions, both of which are loaded with the key information of the same ePassport. This can be achieved by triggering two outputs, which must be *get_challenge* messages from readers, and then sending an input to an ePassport, as performed by the following three transitions.

$$h_0 \colon \boldsymbol{System} \xrightarrow{\overline{c}(u_1)} \xrightarrow{\overline{c}(u_2)} \xrightarrow{d(x)} \boldsymbol{System}^{\mathrm{I}}$$

In the above, $h_0 \triangleq \mathit{error}^i \cdot \mathit{get\_challenge}^i \cdot c^i \cdot d^i$ is the initial history, which constrains the initial free variables so that they may not be instantiated with private messages that are output later during execution. We also have $\boldsymbol{System}^{\mathrm{I}}$ defined as follows (employing abbreviations in Fig. 5), where $\theta_1 = \left\{ {}^{\mathit{get\_challenge},\, \mathit{get\_challenge}}\!/_{u_1,\, u_2} \right\}$:

$$\begin{aligned}\nu ke_1, km_1.\big(\theta_1 \mid\ & V1(ke_1, km_1) \mid P1(ke_1, km_1, x) \mid V1(ke_1, km_1) \mid P(ke_1, km_1) \mid \\ & !(V(ke_1, km_1) \mid P(ke_1, km_1)) \mid !\nu ke.\nu km.!(V(ke, km) \mid P(ke, km))\,\big)\end{aligned}$$

Spec can only follow these actions, by starting two reader sessions with different ePassports.

$$h_0 \colon \boldsymbol{Spec} \xrightarrow{\overline{c}(u_1)} \xrightarrow{\overline{c}(u_2)} \xrightarrow{d(x)} \boldsymbol{Spec}^{\mathrm{I}}$$

where $\boldsymbol{Spec}^{\mathrm{I}}$ is defined as follows:

$$\begin{aligned}\nu ke_1, km_1, ke_2, km_2.\big(\theta_1 \mid\ & V1(ke_1, km_1) \mid P1(ke_1, km_1, x) \mid \\ & V1(ke_2, km_2) \mid P(ke_2, km_2) \mid !\nu ke.\nu km.\,(V(ke, km) \mid P(ke, km))\,\big)\end{aligned}$$

Note, since open bisimilarity is preserved by associativity and commutativity of parallel composition and equivariance, we have already also covered the case where, in the specification, the input is received by the ePassport with keys ke_2 and km_2. Note there is a third possible response by the

$$\text{The ePassport (or prover):} \qquad P(ke, km) \triangleq d(x).P1(ke, km, x)$$

$$P1(ke, km, x) \triangleq [x = get\_challenge]\nu nt.\overline{c}\langle nt\rangle.P2(ke, km, nt) \quad P2(ke, km, nt) \triangleq d(y).P3(ke, km, nt, y)$$

$$\begin{array}{rl} P3(ke, km, nt, y) \triangleq & \mathtt{if}\,\mathtt{snd}(y) = \mathtt{mac}(\mathtt{fst}(y)\,, km)\ \mathtt{then} \\ & \quad \mathtt{if}\, nt = \mathtt{fst}(\mathtt{snd}(\mathtt{dec}(\mathtt{fst}(y)\,, ke)))\ \mathtt{then} \\ & \qquad \nu kt.\mathtt{let}\, m = \{\langle nt,\ \langle \mathtt{fst}(\mathtt{dec}(\mathtt{fst}(y)\,, ke)),\ kt\rangle\rangle\}_{ke}\ \mathtt{in} \\ & \qquad \overline{c}\langle m, \mathtt{mac}(m, km)\rangle \\ & \quad \mathtt{else}\,\overline{c}\langle error\rangle \\ & \mathtt{else}\,\overline{c}\langle error\rangle \end{array}$$

$$\text{The Reader (or verifier):} \qquad V(ke, km) \triangleq \overline{c}\langle get\_challenge\rangle.V1(ke, km)$$

$$V1(ke, km) \triangleq d(nt).V2(ke, km, nt) \qquad V2(ke, km, nt) \triangleq \begin{array}{l} \nu nr.\nu kr.\mathtt{let}\, m = \{\langle nr,\ \langle nt,\ kr\rangle\rangle\}_{ke}\ \mathtt{in} \\ \overline{c}\langle m, \mathtt{mac}(\langle m,\ km\rangle)\rangle \end{array}$$

Figure 5: Abbreviations for process used throughout this symbolic analysis.

specification, where a third session with yet another set of keys is started, but the distinguishing strategy in that branch is subsumed by the distinguishing strategy in the cases we explain here.

The updated history, tracking constraints on variables after these initial three transitions, is:

$$h_1 = h_0 \cdot {u_1}^o \cdot {u_2}^o \cdot x^i$$

3.2.2. *Applying respectful substitutions.* Since we are reasoning symbolically, the first input, performed above, is initially a variable x. When unfolding the rules of the labelled transition in Fig. 4, we find that the following transition is enabled for substitutions σ respecting history h_1 equating the messages in the guard $x = get\_challenge$, where a most general unifier is clearly $\left\{{}^{get\_challenge}/_{x}\right\}$.

$$\begin{array}{l} h_1\left\{{}^{get\_challenge}/_{x}\right\} \cdot ke_1^o \cdot km_1^o : \theta_1 \mid \left[x\left\{{}^{get\_challenge}/_{x}\right\} = get\_challenge\right]\nu nt.\overline{c}\langle nt\rangle.P2(ke, km, nt) \\ \hfill \xrightarrow{\overline{c}(v)} \nu nt_1.\,(\theta_1 \mid \{{}^{nt_1}/_{v}\} \mid P2(ke, km, nt_1)) \end{array}$$

The above transition is valid since, the unifier $\left\{{}^{get\_challenge}/_{x}\right\}$ respects history $h_1 \cdot ke_1^o \cdot km_1^o$, which is trivially the case since there are no constraints on unifying variables such as $get\_challenge$ and x.

Using the above transition we induce the following transitions for the system and specification.

$$h_1 : \boldsymbol{System}^{\mathrm{I}}\left\{{}^{get\_challenge}/_{x}\right\} \xrightarrow{\overline{c}(v)} \boldsymbol{System}^{\mathrm{II}} \qquad h_1 : \boldsymbol{Spec}^{\mathrm{I}}\left\{{}^{get\_challenge}/_{x}\right\} \xrightarrow{\overline{c}(v)} \boldsymbol{Spec}^{\mathrm{II}}$$

where $\boldsymbol{System}^{\mathrm{II}}$ is defined as follows (employing abbreviations in Fig. 5),

$$\begin{array}{rl} \nu ke_1, km_1, nt_1.\big(\theta_1 \mid \left\{{}^{nt_1}/_{v}\right\} \mid & V1(ke_1, km_1) \mid P2(ke_1, km_1, nt_1) \mid V1(ke_1, km_1) \mid P(ke_1, km_1) \mid \\ & !(V(ke_1, km_1) \mid P(ke_1, km_1)) \mid !\nu ke.\nu km.!(V(ke, km) \mid P(ke, km))\ \big) \end{array}$$

and $\boldsymbol{Spec}^{\mathrm{II}}$ is defined as follows.

$$\begin{array}{rl} \nu ke_1, km_1, ke_2, km_2, nt_1.\big(\theta_1 \mid \left\{{}^{nt_1}/_{v}\right\} \mid & V1(ke_1, km_1) \mid P1(ke_1, km_1, nt_1) \mid \\ & V1(ke_2, km_2) \mid P(ke_2, km_2) \mid !\nu ke.\nu km.\,(V(ke, km) \mid P(ke, km))\ \big) \end{array}$$

The updated history at this point is $h_2 \triangleq h_1\left\{{}^{get\_challenge}/_{x}\right\} \cdot v^o$, where the substitution records that the most recent input x was a *get_challenge* message. In full, we have at this point:

$$h_2 \triangleq error^i \cdot get\_challenge^i \cdot c^i \cdot d^i \cdot u_1^o \cdot u_2^o \cdot get\_challenge^i \cdot v^o$$

3.2.3. *Alternating play in the distinguishing game.* We now appeal to the symmetry of bisimilarity, allowing the specification $\boldsymbol{System}^{\mathrm{II}}$ to lead with one input. That strategy allows us to trigger the reader which does not have the same keys as the ePassport that outputs a nonce in the previous step. That approach leads to a distinguishing game that quite accurately describes a practical strategy, which can be implemented using NFC enabled phones running a modified ePassport reader app, as first reported in the conference version of this paper [FHMS19]. In this strategy, the attacker deliberately selects the reader that should fail to authenticate an ePassport if unlinkability really holds as modelled by the idealised specification.

The flow is as follows, where annotation (†) indicates the player (the system or specification) that leads at each point in the game. Note the system has always been leading up to now, in order to trigger the scenario where two sessions with the same ePassport really started.

$$h_2\colon \boldsymbol{System}^{\mathrm{II}} \xrightarrow{d(nt)} (\dagger)\boldsymbol{System}^{\mathrm{III}} \xrightarrow{\overline{c}(w)}\xrightarrow{d(y)} \boldsymbol{System}^{\mathrm{IV}}$$

$$h_2\colon (\dagger)\boldsymbol{Spec}^{\mathrm{II}} \xrightarrow{d(nt)} \boldsymbol{Spec}^{\mathrm{III}} \xrightarrow{\overline{c}(w)}\xrightarrow{d(y)} \boldsymbol{Spec}^{\mathrm{IV}}$$

where $\boldsymbol{System}^{\mathrm{IV}}$ is defined as follows (employing abbreviations in Fig. 5):

$$\begin{array}{l} \nu ke_1, km_1, nt_1, nr_2, kr_2.\Big(\\ \quad \theta_1 \mid \left\{{}^{nt_1}/_{v}\right\} \mid \left\{{}^{\left\langle \{\langle nr_2, \langle nt, kr_2\rangle\rangle\}_{ke_1}, \mathtt{mac}\left(\{\langle nr_2, \langle nt, kr_2\rangle\rangle\}_{ke_1}, km_1\right)\right\rangle}/_{w}\right\} \mid \\ \quad V1(ke_1, km_1) \mid P3(ke_1, km_1, nt, y) \mid 0 \mid P(ke_1, km_1) \mid \\ \quad !(V(ke_1, km_1) \mid P(ke_1, km_1)) \mid !\nu ke.\nu km.!(V(ke, km) \mid P(ke, km))\ \Big) \end{array}$$

and $\boldsymbol{Spec}^{\mathrm{IV}}$ is defined as follows

$$\begin{array}{l} \nu ke_1, km_1, ke_2, km_2, nt_1, nr_2, kr_2.\Big(\\ \quad \theta_1 \mid \left\{{}^{nt_1}/_{v}\right\} \mid \left\{{}^{\left\langle \{\langle nr_2, \langle nt, kr_2\rangle\rangle\}_{ke_2}, \mathtt{mac}\left(\{\langle nr_2, \langle nt, kr_2\rangle\rangle\}_{ke_2}, km_2\right)\right\rangle}/_{w}\right\} \mid \\ \quad V1(ke_1, km_1) \mid P3(ke_1, km_1, nt, y) \mid 0 \mid P(ke_2, km_2) \mid !\nu ke.\nu km.\,(V(ke, km) \mid P(ke, km))\ \Big) \end{array}$$

Observe that all the above transitions proceed lazily without instantiating the input variable nt. In the specification, the reader with keys ke_2, km_2 is used up entirely, without determining yet what challenge was received. Observe also that $\boldsymbol{System}^{\mathrm{II}}$ has only one option, up to structural rules such as commutativity of parallel composition, for following the specification (without being immediately distinguishable), which is to continue a session with keys ke_1, km_1.

The updated history at this point records the two inputs and the output in the order they occurred in the above transitions, as follows.

$$h_4 \triangleq h_2 \cdot nt^i \cdot w^o \cdot y^i$$

Remark 3.7 (playing this strategy)**.** *A question arising at this point is whether the change of player at this point is meaningful in terms of attacker models. In general, to answer such a question we require domain specific knowledge. Observe that the input action, where the specification leads, selects a specific reader which should ideally behave as if it has different keys from the ePassport issuing the challenge nonce. In reality, the attacker does indeed have the power to choose which reader will receive an input, and so can indeed choose the reader that, according to the specification* *of unlinkability, should not successfully authenticate with the ePassport, i.e., the reader that is not located next to an ePassport that has just engaged in an OCR session with it. Thus the need for a game at this point is partly due to under-specification in the model where there are insufficient observables to determine that the reader is not in proximity to the ePassport issuing a challenge. Note this is far from being the only distinguishing strategy; other distinguishing strategies may require a different domain-specific explanation.*

3.2.4. *Calculating inputs using the sequent calculus.* Now consider whether $P3(ke_1, km_1, nt, y)$, which is a subprocess of $\boldsymbol{System}^{\mathrm{IV}}$ shown in expanded form below, can make progress.

```
$\mathtt{if}\,\mathtt{snd}(y) = \mathtt{mac}(\mathtt{fst}(y)\,,\, km_1)\ \mathtt{then}$
    $\mathtt{if}\, nt = \mathtt{fst}(\mathtt{snd}(\mathtt{dec}(\mathtt{fst}(y)\,, ke_1)))\ \mathtt{then}$
        $\nu kt.\mathtt{let}\, m = \{\langle nt,\, \langle \mathtt{fst}(\mathtt{dec}(\mathtt{fst}(y)\,, ke_1)),\, kt\rangle\rangle\}_{ke_1}\ \mathtt{in}$
        $\overline{c}\langle m, \mathtt{mac}(m, km_1)\rangle$
    $\mathtt{else}\, \overline{c}\langle error\rangle$
```

In what follows, we must take into account the active substitution of $\boldsymbol{System}^{\mathrm{IV}}$, which we recall below and denote by θ_4:

$$\theta_4 \triangleq \left\{ {}^{get\_challenge,\ get\_challenge,\ nt_1,\ \left\langle \{\langle nr_2, \langle nt, kr_2\rangle\rangle\}_{ke_2}, \mathtt{mac}\left(\{\langle nr_2, \langle nt, kr_2\rangle\rangle\}_{ke_2}, km_2\right)\right\rangle} \big/ {}_{u_1,\ u_2,\ v,\ w} \right\}$$

By the rules in Fig. 4, the two `then` branches of the `if-then-else` statements above, which result in a non-error output, can only be triggered for particular substitutions σ. In particular, we are interested in whether there are substitutions σ respecting history h_4 such that the two equations below hold, and, furthermore, σ is fresh for the bound variables $nt_1, ke_1, km_1, nr_2, kr_2$ (a constraint enforced by the oRes rule).

$$\mathtt{snd}(y)(\sigma \circ \theta_4)^* =_E \mathtt{mac}(\mathtt{fst}(y)\,,\, km_1)(\sigma \circ \theta_4)^*$$
$$nt_1(\sigma \circ \theta_4)^* =_E \mathtt{fst}(\mathtt{snd}(\mathtt{dec}(\mathtt{fst}(y)\,, ke_1)))(\sigma \circ \theta_4)^*$$

It is convenient to select fresh variable y' to represent the local view of messages that y can be mapped to by the relevant substitution $(\sigma \circ \theta_4)^*$, i.e., for some substitution σ', instantiating y' we have $y(\sigma \circ \theta_4)^* = y'\sigma'$. This represents the fact that y represents the external view of an observer or attacker when they inject inputs, while y' exposes more of the internal structure of messages that cannot be observed by an attacker. While such additional structure may contain more private information than the attacker is immediately aware of (e.g., because the message represent a cyphertext), that information may be required, internally by the process, in order to enable guards such as the guard in the above `if-then-else` statements. This leads us to the following equations.

$$\mathtt{snd}(y') =_E \mathtt{mac}(\mathtt{fst}(y')\,,\, km_1) \quad \text{and} \quad nt_1 =_E \mathtt{fst}(\mathtt{snd}(\mathtt{dec}(\mathtt{fst}(y')\,, ke_1)))$$

We show how to calculate a most general unifier for the above equations. Firstly, we remove destructors $\mathtt{fst}(\cdot)$, $\mathtt{snd}(\cdot)$ and $\mathtt{dec}(\cdot,\cdot)$ by instantiating variables to which they are applied with the most general form of the constructor to which the destructor is applied. This yields the following substitution, where y_1 and y_2 are fresh variables.

$$\left\{ {}^{\left\langle \{\langle y_1, \langle nt_1, y_2\rangle\rangle\}_{ke_1}, \mathtt{mac}\left(\{\langle y_1, \langle nt_1, y_2\rangle\rangle\}_{ke_1}, km_1\right)\right\rangle} \big/ {}_{y'} \right\}$$

The problem now is to calculate the most general form of y_1 and y_2, refining the above substitution taking into account the history h_4, active substitution θ_4 and bound variables $nt_1, ke_1, km_1, nr_2, kr_2$, as described above. This question can be formulated as the problem of calculating the most general solutions to a system of *deducibility constraints* which are generated from the above mentioned constraints and active substitution.

The first step in this calculation is to generate an intermediate constraint system to solve. The following is an alternative representation of a history, where the names to the left of a turnstile represent the knowledge of the attacker at the moment when the input message to the right of that turnstile is performed.

$$\vdash \mathit{error} \qquad \vdash \mathit{get\_challenge} \qquad \vdash c \qquad \vdash d \qquad u_1, u_2 \vdash \mathit{get\_challenge} \qquad u_1, u_2, v \vdash nt \qquad u_1, u_2, v, w \vdash y$$

In this case, it is sufficient to focus on the final two intermediate constraints, although, in general, the initial constraints are essential for ensuring no private information from outputs during execution are used to instantiate the initial knowledge. Also, u_1 and u_2 provide no new information so can be safely removed from the constraints in order to focus on the essential aspects of the problem. From the intermediate constraints $v \vdash nt$ and $v, w \vdash y$, annotated with messages generated by applying the active substitution θ_4 to each of the variables on the left of the turnstile. We also apply the substitution generated for y' above, resulting in two deducibility constraints described below.

The first deducibility constraint generated is as follows, where nt' is a fresh variable, which is introduced for the same reason as we introduced y', as explained above.

$$v\colon nt_1 \vdash nt\colon nt' \tag{3.1}$$

Thus, nt' represents any message such that for some suitable substitutions σ and σ' we have $nt(\sigma \circ \theta_4)^* = nt'\sigma'$, where $\mathrm{dom}(\theta_4)$ are fresh for σ'. Thus, the difference is that $nt\sigma$ may not refer directly the private names representing various keys and nonces, whereas $nt'\sigma'$ can.

Such deducibility constraint of the form $\Gamma \vdash x\colon x'$, where x, x' are variables, are said to be in *solved form*. This means that x, x' can be any messages produced using information in Γ plus some fresh variables, and x' is the local view of x taking into account the current active substitution, following the principles used to explain the use of y' and nt'. Thus the constraint (3.1) generated above is already in solved form, hence, by itself, does not require further analysis.

The second deducibility constraint, generated from intermediate constraint $v, w \vdash y$ by annotating variables with messages given by the active substitution θ_4, is as follows.

$$\begin{aligned} v\colon nt_1,\ w\colon \Big\langle \{\langle nr_2, \langle nt', kr_2\rangle\rangle\}_{ke_1}, \mathtt{mac}\Big(\{\langle nr_2, \langle nt', kr_2\rangle\rangle\}_{ke_1}, km_1\Big)\Big\rangle & \\ \vdash y\colon \Big\langle \{\langle y_1, \langle nt_1, y_2\rangle\rangle\}_{ke_1}, \mathtt{mac}\Big(\{\langle y_1, \langle nt_1, y_2\rangle\rangle\}_{ke_1}, km_1\Big)\Big\rangle & \end{aligned} \tag{3.2}$$

We find all solutions to the system consisting of the above deducibility constraints (3.1) and (3.2), by calculating the most general substitutions such that there is a proof tree using the sequent calculus rules in Fig. 6, where the leaves of each proof are either *axioms* or are in *solved form*. Fig. 6 extends an existing sequent calculus presentation of deducibility constraints [TGD10] with annotations to the left of a colon representing *recipes* for how a message is deduced.

For this system of constraints, the only possibility is to apply the axiom in Fig. 6. This is achieved by unifying the following messages (recall that nt_1, ke_1, km_1, nr_2, kr_2 are private names hence cannot be unified with other messages):

$$\Big\langle \{\langle nr_2, \langle nt', kr_2\rangle\rangle\}_{ke_1}, \mathtt{mac}\Big(\{\langle nr_2, \langle nt', kr_2\rangle\rangle\}_{ke_1}, km_1\Big)\Big\rangle = \mathtt{mac}\Big(\{\langle y_1, \langle nt_1, y_2\rangle\rangle\}_{ke_1}, km_1\Big)$$

We now use the most general unifier for the above problem, $\sigma' = \left\{ {}^{nt_1, nr_2, kr_2}/_{nt', y_1, y_2} \right\}$ thereby allowing the axiom in Fig. 6 to be applied to both deducibility constraints generated above (firstly ignoring the recipes on the left of each colon). Now, taking into account the recipe on the left of each colon, each of the deducibility constraints is an axiom only if we have $v = nt$ and $w = y$, which leads us to the substitution $\sigma = \left\{ {}^{v, w}/_{nt, y} \right\}$. Notice the domain of this substitution must be $\{nt, y\}$, since v and w are

$$\frac{}{\Gamma, R\colon M \vdash R\colon M}\,\mathit{axiom} \qquad \frac{\Gamma \vdash R_1\colon K_1 \quad \ldots \quad \Gamma \vdash R_n\colon K_n}{\Gamma \vdash f(R_1, \ldots R_n)\colon f(K_1, \ldots K_n)}\,\mathit{intro}$$

$$\text{where } f \in \{\langle\cdot,\ \cdot\rangle\,, \{\cdot\}_{\cdot}\,, \mathtt{mac}(\cdot,\cdot)\,, \mathtt{dec}(\cdot,\cdot)\,, \mathtt{fst}(\cdot)\,, \mathtt{snd}(\cdot)\,, \}$$

$$\frac{\Gamma, \mathtt{fst}(R) : M, \mathtt{snd}(Rr) : N \vdash S : K}{\Gamma, R\colon \langle M,\ N\rangle \vdash S : K}\,\mathit{pair\text{-}elim} \qquad \frac{\Gamma \vdash T\colon K \quad \Gamma, \mathtt{dec}(R, T) : M \vdash S : K}{\Gamma, R\colon \{M\}_K \vdash S : N}\,\mathit{enc\text{-}elim}$$

Figure 6: Deducibility constraints, in sequent calculus style, annotated with messages representing recipes for producing messages to the left of each colon.

treated as names rather then free variables (this is enforced by the constraints on output variables in the notion of a respectful substitution).

Thereby, from deducibility constraints (3.1) and (3.2) where σ is applied to the left of each colon and σ' is applied to the right of each colon, we obtain the following two proofs. Each proof consists of a single axiom, where a proof is a proof tree where all leaves are axioms (hence the set of premises are empty and hence vacuously in solved form).

$$\frac{}{v\colon nt_1 \vdash v\colon nt_1} \qquad \frac{}{\begin{array}{r} v\colon nt_1,\ w\colon \Big\langle \{\langle nr_2,\ \langle nt_1,\ kr_2\rangle\rangle\}_{ke_1},\ \mathtt{mac}\Big(\{\langle nr_2,\ \langle nt_1,\ kr_2\rangle\rangle\}_{ke_1}, km_1\Big)\Big\rangle \\ \vdash w\colon \Big\langle \{\langle nr_2,\ \langle nt_1,\ kr_2\rangle\rangle\}_{ke_1},\ \mathtt{mac}\Big(\{\langle nr_2,\ \langle nt_1,\ kr_2\rangle\rangle\}_{ke_1}, km_1\Big)\Big\rangle \end{array}}$$

Thereby we have calculated the most general respectful substitution $\left\{{}^{v,\,w}/_{nt,\,y}\right\}$, enabling the following transition.

$$h_4\left\{{}^{v,\,w}/_{nt,\,y}\right\}\colon \boldsymbol{System}^{\mathrm{IV}}\left\{{}^{v,\,w}/_{nt,\,y}\right\} \xrightarrow{\bar{c}(z)} \boldsymbol{System}^{\mathrm{V}}$$

where the frame of $\boldsymbol{System}^{\mathrm{V}}$ (ignoring the process) is as follows.

$$\begin{array}{l} \nu ke_1, km_1, nt_1, nr_2, kr_2, kt_1.\Big(\Big\{{}^{get\_challenge,\, get\_challenge}/_{u_1,\, u_2}\Big\} \mid \left\{{}^{nt_1}/_{v}\right\} \mid \\ \quad \Big\{{}^{\left\langle \{\langle nr_2, \langle nt_1, kr_2\rangle\rangle\}_{ke_1}, \mathtt{mac}\left(\{\langle nr_2, \langle nt_1, kr_2\rangle\rangle\}_{ke_1}, km_1\right)\right\rangle}/_{w}\Big\} \mid \Big\{{}^{\left\langle \{\langle nt_1, \langle nr_2, kt_1\rangle\rangle\}_{ke_1}, \mathtt{mac}\left(\{\langle nt_1, \langle nr_2, kt_1\rangle\rangle\}_{ke_1}, km_1\right)\right\rangle}/_{z}\Big\} \mid \ldots\Big) \end{array}$$

Observe that the specification, $\boldsymbol{Spec}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\}$ can also perform an output, either starting a new session, or triggering an error message. In either case, we reach a state that is distinguishable by static equivalence witnessed by the test $z = \mathit{error}$ or $z = \mathit{get\_challenge}$ respectively.

Notice that in the specification, the `else` branch in which an error message is output is enabled by the oElse rule in Fig. 4. That rule is enabled only when the following inequality holds, where h'_4 is the current history extended with the private names by using the oRes rule as follows, i.e., $h'_4 = h_4\left\{{}^{v,w}/_{nt,y}\right\}\cdot ke^o_1\cdot km^o_1\cdot ke^o_2\cdot km^o_2\cdot nt^o_1\cdot nr^o_2\cdot kr^o_2\cdot kt^o_1$ and ρ_4 is the active substitution of $\boldsymbol{Spec}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\}$ at this point.

$$h'_4, \rho_4 \models \mathtt{mac}\Big(\{\langle nr_2,\ \langle nt_1,\ kr_2\rangle\rangle\}_{ke_2}, km_2\Big) \neq \mathtt{mac}\Big(\{\langle nr_2,\ \langle nt_1,\ kr_2\rangle\rangle\}_{ke_2}, km_1\Big)$$

The above inequality is satisfied, since there is no substitution respecting h'_4 equating the two terms in the above inequality, i.e., it holds even under intuitionistic assumptions. To see why, observe that any unifier for the above message equates km_1 and km_2, which must be kept distinct by any substitution respecting the above history, since both km^o_1 and km^o_2 appear in the history.

3.3. **Constructing a distinguishing formula from the distinguishing strategy.** Firstly, we briefly summarise the distinguishing strategy calculated in the previous subsections.

1. ***System*** leads with transitions labelled $\overline{c}(u_1)$ then $\overline{c}(u_2)$ and then $d(x)$ (thereby reaching ***System***$^{\mathrm{I}}$ in which sessions have started with two readers and an ePassport, all using the same keys).
2. ***System***$^{\mathrm{I}}\left\{{}^{get\_challenge}/_{x}\right\}$ leads with transition labelled $\overline{c}(v)$. If ***Spec***$^{\mathrm{I}}\left\{{}^{get\_challenge}/_{x}\right\}$ follows with $v = get\_challenge$, we are done, otherwise continue.
3. ***Spec***$^{\mathrm{II}}$ leads with transition labelled $\overline{d}(nt)$ starting up the wrong reader. If ***System***$^{\mathrm{II}}$ follows by inputting the wrong message into a new ePassport session this can be picked up by performing one more action, otherwise continue.
4. ***System***$^{\mathrm{III}}$ leads with transitions labelled $\overline{c}(w)$ and then $d(y)$. If ***Spec***$^{\mathrm{III}}$ follows with $w = get\_challenge$ we are done, otherwise continue.
5. ***System***$^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\}$ leads with transitions $\overline{c}(z)$. This can only be followed by a transition from ***Spec***$^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\}$ reaching a state where $z = error$ or $z = get\_challenge$.

The problem now is that open bisimilarity (Def. 3.5) does not satisfy any notion of completeness, hence a distinguishing strategy may be a spurious counterexample. Spurious counterexamples, cannot be transformed into counterexamples for strong early bisimilarity (Def. 2.9) and are less likely to indicate the presence of an attack.

The above strategy does not describe a spurious counterexample; and furthermore it can be turned into a real attack. In order to show that it is not a spurious counterexample, our methodology is to construct a modal logic formula from the distinguishing strategy. Instead of using a modal logic characterising open bisimilarity (which would be a generalisation of intuitionistic $\mathcal{OM}$ [AHT17] to the applied π-calculus, making used of the notion of reachability in Def. 3.4) we employ a modal logic characterising strong early bisimilarity called classical $\mathcal{FM}$.

3.3.1. *Introducing classical $\mathcal{FM}$.* The syntax of modal logic *classical* $\mathcal{FM}$ ($\mathcal{F}$ is for free inputs, $\mathcal{M}$ is for match [MPW93]) is presented below.

$$\begin{array}{rlcrl}
\phi ::= & M = N & \text{equality} & \text{abbreviations:} & \mathtt{tt} \triangleq M = M \\
\mid & \phi \wedge \phi & \text{conjunction} & & M \neq N \triangleq \neg(M = N) \\
\mid & \langle\pi\rangle\phi & \text{diamond} & & [\pi]\phi \triangleq \neg\langle\pi\rangle\neg\phi \\
\mid & \neg\phi & \text{negation} & & \phi \vee \psi \triangleq \neg\left(\neg\phi \wedge \neg\psi\right)
\end{array}$$

The semantics of classical $\mathcal{FM}$ is given by the least relation $A \models \phi$ between extended processes A and formulae ϕ satisfying the conditions in Fig. 7.

$$\begin{array}{lll}
\nu\vec{x}.(\theta \mid P) \models M = N & \text{iff} & M\theta =_E N\theta \text{ and } \vec{x} \text{ are fresh for } M \text{ and } N \\
A \models \langle\pi\rangle\phi & \text{iff} & \text{there exists } B \text{ such that } A \xrightarrow{\pi} B \text{ and } B \models \phi. \\
A \models \phi_1 \wedge \phi_2 & \text{iff} & A \models \phi_1 \text{ and } A \models \phi_2. \\
A \models \neg\phi & \text{iff} & A \models \phi \text{ does not hold.}
\end{array}$$

Figure 7: The semantics of modal logic "classical $\mathcal{FM}$".

The following theorem formulates what it means for classical $\mathcal{FM}$ to characterise strong early bisimilarity.

Theorem 3.8. *$P \sim Q$, whenever, for all formula ϕ, we have $P \models \phi$ if and only if $Q \models \phi$.*

The proof is provided in Appendix B. From the contrapositive of the above theorem, whenever $P \not\sim Q$, there exists a formula ϕ such that $P \models \phi$ holds, but $Q \not\models \phi$. Such a formula is called a distinguishing formula.

3.3.2. *The attack on BAC as a formula.* We are now in a position to prove Theorem 2.4, restated below for convenience, which establishes that strong unlinkability of the BAC protocol fails.

Theorem 3.9 (Theorem 2.4 restated)**.** *System* $\not\approx$ *Spec.*

Proof. In order to establish the failure of strong unlinkability of the BAC protocol, we make use of the following classical $\mathcal{FM}$ formula ψ.

$$\begin{aligned}\psi \triangleq\ & \langle \overline{c}(u_1)\rangle\langle \overline{c}(u_2)\rangle\langle d\, \mathit{get\_challenge}\rangle\langle \overline{c}(v)\rangle(\\ & \quad v \neq \mathit{get\_challenge} \wedge \\ & \quad [d\, v](\ \langle \overline{c}(w)\rangle\langle d\, w\rangle\langle \overline{c}(z)\rangle(w \neq \mathit{get\_challenge} \wedge z \neq \mathit{get\_challenge} \wedge z \neq \mathit{error}) \\ & \qquad \vee [\overline{c}(w)](w = \mathit{get\_challenge})\ ))\end{aligned}$$

For this formula we can verify ***System*** $\models \psi$ holds; while ***Spec*** $\not\models \psi$. Hence, by Theorem 3.8, ***System*** $\not\sim$ ***Spec***; thereby by Theorem 2.10, *System* $\not\approx$ *Spec*, as required. □

This closes the initial question of whether or not unlinkability holds for the BAC protocol; the answer is that the BAC protocol does not satisfy unlinkability, at least as specified originally in CSF'10 [ACRR10]. This leads to several immediate questions. Firstly, how do we construct the above formula from the distinguishing strategy given by open bisimilarity (to be addressed in Sec. 3.3.3)? Secondly, can we explain why the formula is distinguishing, and from that explanation describe a practical attack? Thirdly, how do we approach the problem of constructing a formula in general and how do we handle cases when a spurious counterexample is discovered? We focus mainly on the first question in this paper. The second question we we return to in the next section; while the third question is worthy of future work, since it would enable tool support.

We emphasise at this point that there are infinitely many alternative distinguishing formulae for this problem, not only ψ. Some such distinguishing formulae use no box modality and instead employ conjunction, where conjunction also appeals to the branching time nature of bisimilarity. We postpone discussing alternatives until Sec. 4, where we propose an alternative model of unlinkability, where the meaning of distinguishing strategies is clearer.

3.3.3. *How to construct the distinguishing formula.* We construct the formula named ϕ, used as the distinguishing formula in the proof of Theorem 2.4 (reiterated as Theorem 3.9), by induction on the depth of the distinguishing strategy summarised at the top of Sec. 3.3. To do so, we work backwards through the distinguishing strategy. Note we refer to processes representing intermediate states of an execution previously defined throughout Sec. 3.

Firstly, observe that when the system is in state $\textbf{\textit{System}}^{\mathrm{V}}$ the specification must be in a state where either $\mathit{error} = z$, $\mathit{get\_challenge} = z$ or $\mathit{get\_challenge} = w$, where each pair of messages shows static equivalence is violated. Since for $\textbf{\textit{System}}^{\mathrm{V}}$ both w and z cannot be unified with *get_challenge* or *error* with respect to the history at that point, the intuitionistic negation and classical negation coincide for these equalities, leading to the following formula distinguishing $\textbf{\textit{System}}^{\mathrm{V}}$ from any state the specification can reach at that point.

$$\textbf{\textit{System}}^{\mathrm{V}} \models \mathit{error} \neq z \wedge \mathit{get\_challenge} \neq z \wedge \mathit{get\_challenge} \neq w$$

Since the system leads in order to reach this state using transition $\boldsymbol{System}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\} \xrightarrow{\overline{c}(z)} \boldsymbol{System}^{\mathrm{V}}$, we add a diamond modality to the formula, as follows.

$$\boldsymbol{System}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\} \models \langle \overline{c}(z) \rangle (\mathit{error} \neq z \wedge \mathit{get\_challenge} \neq z \wedge \mathit{get\_challenge} \neq w)$$

The above step is standard for constructing modal logic formulae for distinguishing strategies; however the next step requires care. Firstly, note that the substitution $\left\{{}^{v,w}/_{nt,y}\right\}$ concerns input variables. Thus we push these substitutions back through the distinguishing strategy until the relevant input is instantiated. At this point, since the input action in the distinguishing strategy reaching state $\boldsymbol{System}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\}$ introduced variable y, that variable is instantiated immediately, and nt is pushed back through the strategy, refining the distinguishing strategy to obtain the following transitions using the rules of Fig. 3.

$$\boldsymbol{System}^{\mathrm{III}}\{{}^{v}/_{nt}\} \xrightarrow{\overline{c}(w)} \xrightarrow{d\,w} \boldsymbol{System}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\}$$

Since the system was leading, $\boldsymbol{System}^{\mathrm{III}}\{{}^{v}/_{nt}\}$ can be distinguished from $\boldsymbol{Spec}^{\mathrm{III}}\{{}^{v}/_{nt}\}$ by the following formula.

$$\boldsymbol{System}^{\mathrm{III}}\{{}^{v}/_{nt}\} \models \langle \overline{c}(w) \rangle \langle d\,w \rangle \langle \overline{c}(z) \rangle (\mathit{error} \neq z \wedge \mathit{get\_challenge} \neq z \wedge \mathit{get\_challenge} \neq w)$$

The next step in the distinguishing strategy involves a change of leading player, where the specification leads with an action. By pushing back the substitution through the strategy, instantiating the input variable on the label, we have the following transition led by the specification: $\boldsymbol{Spec}^{\mathrm{II}} \xrightarrow{d\,v} \boldsymbol{Spec}^{\mathrm{III}}\{{}^{v}/_{nt}\}$. Since the specification leads at this point and the system follows in any way it can, we write the box modality in the distinguishing formula for the system followed by a disjunction of formulae, where each formula distinguishes $\boldsymbol{Spec}^{\mathrm{III}}\{{}^{v}/_{nt}\}$ from any state reachable from $\boldsymbol{System}^{\mathrm{II}}$ by an input transition labelled with $d\,v$.

Observe that, as well as $\boldsymbol{System}^{\mathrm{III}}\{{}^{v}/_{nt}\}$ for which we constructed a distinguishing formula above, there is another, quite distinct, process reachable from $\boldsymbol{System}^{\mathrm{II}}$ by a $d\,v$ transition that can be distinguished by formula $[\overline{c}(w)](w = \mathit{get\_challenge})$ representing that the case when input transition labelled with $d\,v$ results in feeding v into a new reader session, which kills the possibility of continuing an existing session with an output transition. Note this formula is also constructed algorithmically, as we are describing, but this branch is more due to a limitation of the original model unlinkability communicated in CSF'10, that we address next in Sec. 4. Hence we draw no further attention to that branch at this point.

Thereby, we obtain the following formula distinguishing $\boldsymbol{System}^{\mathrm{II}}$ from $\boldsymbol{Spec}^{\mathrm{II}}$.

$$\begin{aligned}\boldsymbol{System}^{\mathrm{II}} \models [d\,v](\; & \langle \overline{c}(w) \rangle \langle d\,w \rangle \langle \overline{c}(z) \rangle (w \neq \mathit{get\_challenge} \wedge z \neq \mathit{get\_challenge} \wedge z \neq \mathit{error}) \\ & \vee [\overline{c}(w)](w = \mathit{get\_challenge})\,))\end{aligned}$$

The rest of the construction of formula ψ follows the pattern of steps already described above, where a diamond modality is appended whenever the system leads and substitutions are pushed back through the distinguishing strategy until they instantiate the relevant input, or reach the root of the formula.

We present an informal graphical depiction of the game that ψ describes in Fig. 8. In the figure, annotation (†) indicates where a process takes over as the leading process in the strategy. When a process is not leading it may have the option to try more than one counter move, represented by the branches in the strategy.

$$\begin{tikzcd}
(\dagger)\ \boldsymbol{System} \arrow[d, "\bar{c}(u_1)"] & \\
{} \arrow[d, "\bar{c}(u_2)"] & \\
{} \arrow[d, "d\ get\_challenge"] & \\
\boldsymbol{System}^{\mathrm{I}} \arrow[d, "\bar{c}(v)"] & \\
\boldsymbol{System}^{\mathrm{II}} \arrow[d, "d\,v"] \arrow[dr, "d\,v"] & \\
(\dagger)\ \boldsymbol{System}^{\mathrm{III}}\{{}^{v}/_{nt}\} \arrow[d, "\bar{c}(w)"] & {} \arrow[d, "\bar{c}(w)"] \\
{} \arrow[d, "d\,w"] & w = get\_challenge \\
\boldsymbol{System}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\} \arrow[d, "\bar{c}(z)"] & \\
z \neq error \wedge z \neq get\_challenge &
\end{tikzcd}$$

$$\begin{tikzcd}
\boldsymbol{Spec} \arrow[d, "\bar{c}(u_1)"] & \\
{} \arrow[d, "\bar{c}(u_2)"] & \\
{} \arrow[d, "d\ get\_challenge"] & \\
\boldsymbol{Spec}^{\mathrm{I}} \arrow[d, "\bar{c}(v)"] \arrow[dr, "\bar{c}(v)"] & \\
(\dagger)\ \boldsymbol{Spec}^{\mathrm{II}} \arrow[d, "d\,v"] & v = get\_challenge \\
\boldsymbol{Spec}^{\mathrm{III}}\{{}^{v}/_{nt}\} \arrow[d, "\bar{c}(w)"] \arrow[dr, "\bar{c}(w)"] & \\
{} \arrow[d, "d\,w"] & w = get\_challenge \\
\boldsymbol{Spec}^{\mathrm{IV}}\left\{{}^{v,w}/_{nt,y}\right\} \arrow[d, "\bar{c}(z)"] \arrow[dr, "\bar{c}(z)"] & \\
z = error & z = get\_challenge
\end{tikzcd}$$

Figure 8: Distinguishing strategy implied by distinguishing formula ψ.

4. A New Chapter for Unlinkability: Refining the Model of Unlinkability

While the previous sections closed a chapter in the story of unlinkability, by proving that there is an attack on the model of unlinkability of the BAC protocol as originally communicated in the proceedings of CSF'10 [ACRR10], this section opens a new chapter by justifying a new model of unlinkability. This new model of unlinkability is a modest improvement on the model previously proposed. It incorporates some explicit observables reflecting the ability of the attacker to observe and hence control the creation of radio frequency communication channels. We demonstrate here why our proposed model of unlinkability more accurately models the distinguishing power of an attacker; and how descriptions of attacks on the BAC protocol, given by modal logic formulae, become clearer. This section can also been seen as introducing preliminaries required for Section 5, where we show how the model proposed discovers new attacks on the PACE protocol that follow a similar pattern to the attacks discovered on the BAC protocol.

The model of strong unlinkability originally proposed in CSF'10 [ACRR10] has many merits. However, a limitation we would like to draw attention to is that it matters whether or not we include the *get_challenge* message that the reader sends to initiate the protocol. That *get_challenge* message happens to be essential for the attacks on unlinkability described in the previous section since, by controlling the *get_challenge* messages sent and received, we can count the number of sessions that are present and thereby infer when a message sent is a ciphertext in an existing session of the protocol rather than a fresh nonce at the beginning of a new protocol.

This is perhaps clearest in Fig. 8 of the previous section. Observe that in order to reach the bottommost state in the figure, we ensure that no additional sessions are started beyond the two reader sessions and one ePassport session at the beginning triggered by the topmost three actions in the figure. This causes problems, three of which are highlighted below, which are all due to the modelling decision where all parties use the same channel c for outputs and d for inputs.

- Limitation 1. It is inconvenient and confusing to, throughout the strategy, add branches that have the effect of saying "at this point we don't start a new session."
- Limitation 2. In the reality, the attacker can directly observe whether or not two inputs or outputs are performed within the same session of a protocol and, furthermore, can distinguish between a session with a reader or with an ePassport. This is not only because the attacker must be aware of the physical location of each entity, but also because, for each session, the attacker must open a new channel using the underlying transport protocol, as standardised in ISO/IEC 14443 [ISO18].
- Limitation 3. Finally, the fact that the *get_challenge* message is useful for counting the number of each type of session initiated, is rather a misuse of that message. Message *get_challenge* contributes nothing to this authentication protocol (other than impeding the stronger authentication property synchronisation [CMdV06], which is immediately violated in protocols with a constant message). Hence removing it from a model of BAC should not result in attacks ceasing to exist.

The above limitations of existing models used to analyse the unlinkability of the BAC protocol, as employed in previous sections, can be addressed simply by declaring a fresh public channel for each session. To do so, we extend the model with two channels, say *passport* and *reader*, that are used to model the creation of a new channel in the respective roles of either an ePassport or a reader. In the applied π-calculus, we achieve this by sending a fresh channel to the environment on these channels for each session, as in the following new scheme for the system and specification.

$$\begin{array}{ll} \text{Scheme for System:} & !\nu\vec{k}.!\Big(\nu c.\overline{passport}\langle c\rangle.Prover(c,\vec{k}) \mid \nu c.\overline{reader}\langle c\rangle.Verifier(c,\vec{k})\Big) \\ \text{Scheme for Specification:} & !\nu\vec{k}.\Big(\nu c.\overline{passport}\langle c\rangle.Prover(c,\vec{k}) \mid \nu c.\overline{reader}\langle c\rangle.Verifier(c,\vec{k})\Big) \end{array}$$

The above we propose as a general scheme for RFID protocols employing symmetric keys $\vec{k}$, which are shared through another channel that the attacker cannot intercept. Recall, in the case of ePassport protocols, this is usually achieved by an OCR session; but may be achieved by other means such as sending the key to the reader via a secure connection between a personal device and the reader. Thus a fundamental assumption in all these models of unlinkability is that we are considering use cases where intercepting and manipulating RFID communication is easier than intercepting the keys, which would trivially break unlinkability.

Notice we directly employ a presentation of processes that does not involve τ-transitions. This simplifies the problem such that strong notions of bisimilarity may be employed, without loss of modelling power. Of course, to do so, we should assume each instance of *Prover* and *Verifier* is a sequential process (or apply another suitable restriction for forbidding τ-transitions internal to a single reader or ePassport).

4.1. **The unlinkability of the BAC protocol, simplified.** Following the above scheme for the BAC protocol, $\vec{k}$ is ke, km, and $Prover(c, ke, km)$ and $Verifier(c, ke, km)$ are instantiated with the processes

$P_{BAC}(c, ke, km)$ and $V_{BAC}(c, ke, km)$ defined below.

$$
\begin{array}{rl}
P_{BAC}(c, ke, km) \triangleq & \nu nt.\overline{c}\langle nt\rangle.c(y). \\
& \texttt{if}\, \texttt{snd}(y) = \texttt{mac}(\texttt{fst}(y)\,, km)\ \texttt{then} \\
& \quad \texttt{if}\, nt = \texttt{fst}(\texttt{snd}(\texttt{dec}(\texttt{fst}(y)\,, ke)))\ \texttt{then} \\
& \quad\quad \nu kt.\texttt{let}\, m = \{\langle nt,\ \langle \texttt{fst}(\texttt{dec}(\texttt{fst}(y)\,, ke)),\ kt\rangle\rangle\}_{ke}\ \texttt{in} \\
& \quad\quad \overline{c}\langle m, \texttt{mac}(m, km)\rangle \\
& \quad \texttt{else}\, \overline{c}\langle error\rangle \\
& \texttt{else}\, \overline{c}\langle error\rangle \\
V_{BAC}(c, ke, km) \triangleq & c(nt).\nu nr.\nu kr. \\
& \texttt{let}\, m = \{\langle nr,\ \langle nt,\ kr\rangle\rangle\}_{ke}\ \texttt{in}\, \overline{c}\langle m, \texttt{mac}(\langle m,\ km\rangle)\rangle
\end{array}
$$

Notice the above processes are simply $P(ke, km)\{^{c}/_{d}\}$ and $V(ke, km)\{^{c}/_{d}\}$ from the previous sections, but with prefixes concerning the *get_challenge* message removed.

In summary, we propose that the problem of whether there is an attack on the unlinkability of the BAC protocol can be resolved by proving that the following theorem holds.

Theorem 4.1. *$System_{BAC} \nsim Spec_{BAC}$, where*

$$
\begin{array}{rl}
System_{BAC} \triangleq & !\nu ke, km.!\big(\nu c.\overline{passport}\langle c\rangle.P_{BAC}(c, ke, km) \mid \nu c.\overline{reader}\langle c\rangle.V_{BAC}(c, ke, km)\big) \\
Spec_{BAC} \triangleq & !\nu ke, km.\big(\nu c.\overline{passport}\langle c\rangle.P_{BAC}(c, ke, km) \mid \nu c.\overline{reader}\langle c\rangle.V_{BAC}(c, ke, km)\big)
\end{array}
$$

Proof. Consider the $\mathcal{FM}$ formula below.

$$
\begin{array}{rl}
\varphi \triangleq & \langle\overline{reader}(c_1)\rangle\langle\overline{reader}(c_2)\rangle\langle\overline{passport}(c_3)\rangle\langle\overline{c_3}(nt)\rangle\big(\\
& \quad \langle c_1\, nt\rangle\langle\overline{c_1}(w)\rangle\langle c_3\, w\rangle\langle\overline{c_3}(z)\rangle(z \neq error) \\
& \wedge \quad \langle c_2\, nt\rangle\langle\overline{c_2}(w)\rangle\langle c_3\, w\rangle\langle\overline{c_3}(z)\rangle(z \neq error)\ \big)
\end{array}
$$

Since $System_{BAC} \models \varphi$, but $Spec_{BAC} \not\models \varphi$, by Theorem 3.8, $System_{BAC} \nsim Spec_{BAC}$. □

Now compare the distinguishing strategy generated by φ, presented in Fig. 9, to the distinguishing strategy for ψ in the previous section, presented in Fig. 8. The attacks described start in a similar fashion. In both figures, the system starts up two readers and an ePassport with the same keys. The specification can only follow by starting two readers with different keys; hence when the ePassport is initialised is has different keys from at least one of the readers. We draw attention to two key differences between the strategies presented below.

The first notable difference between the strategies is that Fig. 9 does not require several sub-branches of the strategy involving *get_challenge* messages. Those branches that appear throughout Fig. 8 are no longer required, since we can directly observe the number of sessions that are present, rather than implicitly controlling the number of sessions by preventing new sessions from initialising. This difference is beneficial for cleaning up messy strategies, making them easier to explain, and allowing more protocols to be analysed without having to insert constant messages into the model of the protocol.

The second notable difference between the strategies is that, in Fig. 9, the system always leads, including at the point where branching occurs. By the time the system decides which branch to take, the specification has already committed to a state where the the ePassport has different keys from either the reader on channel c_1 or the reader on channel c_2. The strategy of the system is to choose to communicate with the reader which has keys that are different to those of the ePassport. Thereby the system wins the game since it can reach a state where no error is produced by the chosen ePassport — a strategy that cannot be matched by the specification. In contrast, the strategy in Fig. 8 achieved a similar effect but in a different way: the attacker changes perspective by switching to a

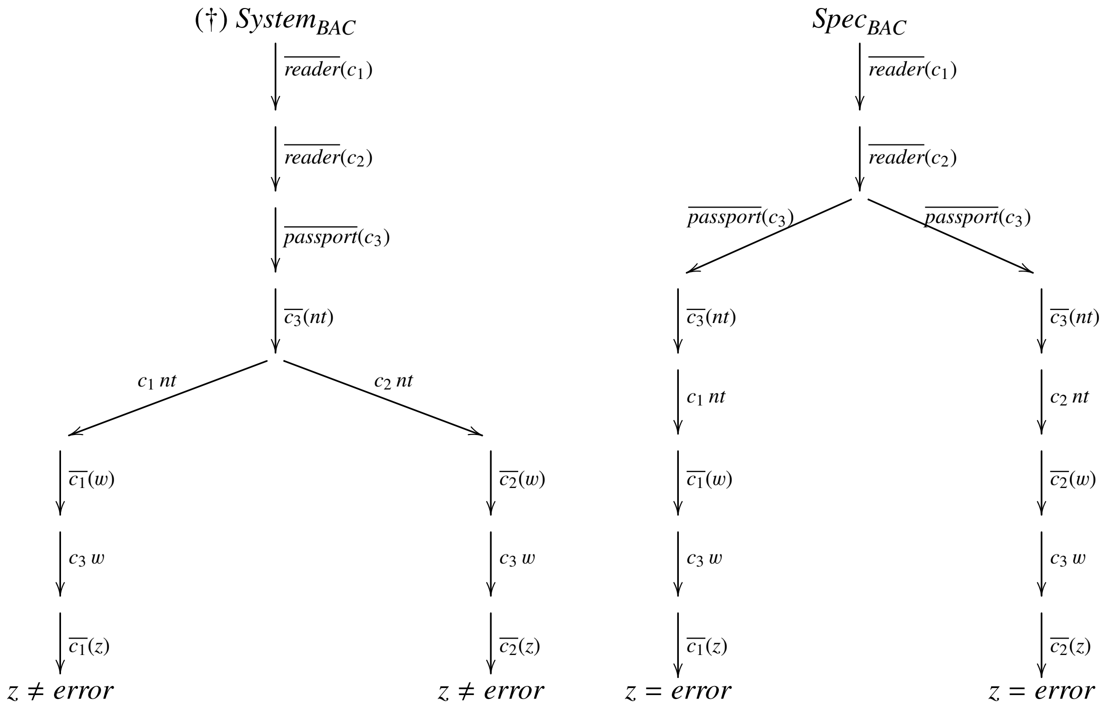

Figure 9: Distinguishing strategy implied by distinguishing formula φ.

view where the leading player is the specification, i.e., what should hypothetically happen. Recall, in Fig. 8, after the change of player, the strategy is for the specification to choose to communicate with a reader that should produce an error message; but the system has no way to produce such an error message hence the system clearly is not equivalent to the hypothetical situation modelled by the specification.

4.2. **Alternative formulas for describing attacks.** The formula in the proof of Theorem 4.1 is not the only formula describing an attack on unlinkability of the BAC protocol. Indeed, in Section 3.3.2 we noted that, when there is an attack, there are infinitely many alternative formulae. In this section, we introduce and explain another formula that is a little longer than φ, defined in the previous section, but is useful for situating the attacks discovered.

Consider the $\mathcal{FM}$ formula below.

$$\begin{aligned}\varsigma \triangleq\ & \langle\overline{reader}(c_1)\rangle\langle\overline{passport}(c_3)\rangle\Big(\\ & \qquad \langle\overline{c_3}(nt)\rangle\langle c_1\ nt\rangle\langle\overline{c_1}(w)\rangle\langle c_3\ w\rangle\langle\overline{c_3}(z)\rangle(z \neq error)\\ & \quad \wedge\ \langle c_1\ nt'\rangle\langle\overline{c_1}(w')\rangle\langle\overline{reader}(c_2)\rangle\langle\overline{c_3}(nt)\rangle\langle c_2\ nt\rangle\langle\overline{c_2}(w)\rangle\langle c_3\ w\rangle\langle\overline{c_3}(z)\rangle(z \neq error)\ \Big)\end{aligned}$$

The above formula also serves as an alternative proof certificate for Theorem 4.1. Observe that $System_{BAC} \vDash \varsigma$, but $Spec_{BAC} \nvDash \varsigma$; and hence, by Theorem 3.8, $System_{BAC} \not\sim Spec_{BAC}$. The distinguishing strategy described by ς is depicted in Fig. 10.

Both formulas ς and φ describe attack strategies on the unlinkability of the BAC protocol. The advantage of φ is that it is more compact. Also, declaring two readers at the top of the strategy and making a choice between them makes clear the key idea: that there is a strategy for testing whether two readers are capable of authenticating the same ePassport.

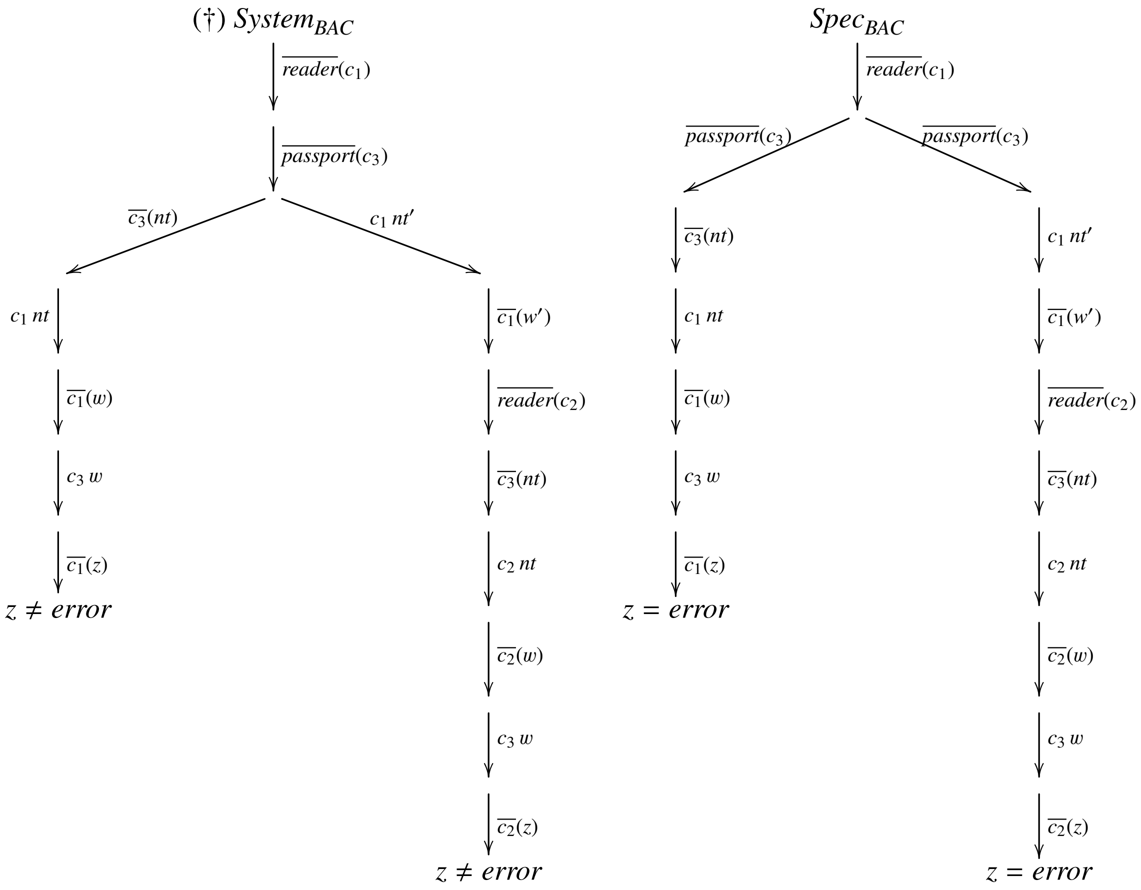

Figure 10: Another distinguishing strategy ς, where readers are created sequentially.

Formula ς is presented to point out that the two readers need not be simultaneously active. Observe that in ς the event representing the creation of the second reader, indicated in both strategies as $\overline{reader}\langle c_2\rangle$, is pushed later in the attack strategy compared to in φ. The distinguishing strategy then proceeds as follows.

(1) A run of a reader and ePassport is created. What is important at this point is that the system has an opportunity to start a reader and ePassport run that match, i.e., the reader is loaded with the key of the ePassport in question. The specification has two choices, which are the first and second respective branches taken by $Spec_{BAC}$ in Fig. 10:
   (a) the specification can start a run with an ePassport with different keys to the reader;
   (b) or using the same keys as the reader.

(2) Now consider the two branches of the conjunction in the formula ς, which occurs after one reader and ePassport are created.
   (i) In the first branch of the conjunction, the second reader is never used. There are only the events required for the run of the ePassport and reader, initially created on channels c_3 and c_1 respectively, to authenticate. This branch of the conjunction can be played by the attacker whenever the specification takes its first branch (1a), where the run of an ePassport involves keys different from those loaded into the reader initially created.

(ii) In the second branch of the conjunction, we do make use of a second reader. To emphasise that no actions of readers need be concurrent for this attack strategy, we first consume the actions of the reader on channel c_1 by using the dummy nonce nt' and effectively ignoring the response w' from that reader. At that point, the second reader is created on channel c_2 such that it is again loaded with the same keys. This allows the second reader to successfully authenticate the ePassport created at the beginning of the attack. This branch of the conjunction is played in response to the specification taking its second branch (1*b*), where the run of the ePassport and first reader on c_1 match, and hence, since no further run may use the same keys, the second reader must fail to authenticate in that idealised setting.

The first thing to observe about the attack described by ς is that, since no two reader runs are concurrently active, we know that designing a system to force reader runs involving the same ePassport to be conducted sequentially will not prevent attacks on unlinkability. We raise this point, since, as explored in related work [Bae21] sequentialising reader sessions does prevent certain kinds of attacks on the unlinkability of certain protocols (such as a previously known attack on the PACE protocol that we will come to in the next Section 5). In that work, an operator, with symbol ¡, is used as a prefix for reader sessions, which acts like a Kleene star [BLMvT16], creating infinitely many copies of a thread sequentially rather than in parallel. The strategy ς, shows clearly that such a strengthened model of the system, where runs of readers with the same ePassport are sequentialised using a Kleene star, will not prevent the attack we describe. Thus strengthening the specification of strong unlinkability only by sequentialising readers will not allow strong unlinkability to be verified. Furthermore, clearly only one ePassport is needed for all attacks we present, so also sequentialising the runs of an ePassport will not affect our analysis.

The second thing to observe about ς is that we can extract message sequence charts (MSCs) from Fig. 10 describing the actions of an attacker required to realise either branch of this attack strategy. Such an MSC is presented in Fig. 11. An MSC is fundamentally not designed to represent

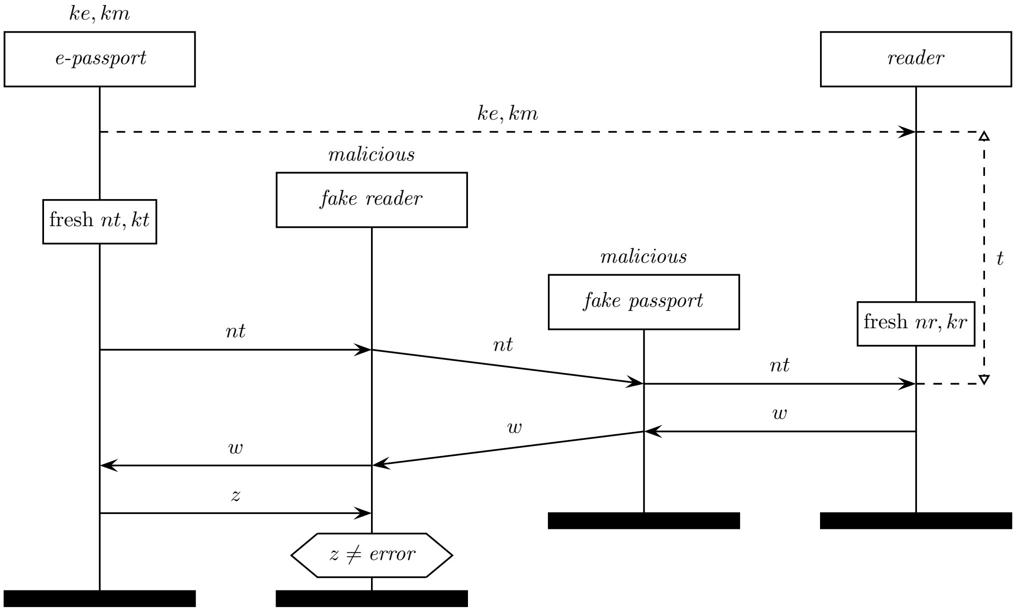

Figure 11: Message sequence chart representation of the left branch in Fig. 10 for $System_{BAC}$.

the branching time and different perspectives in games, however the MSC presented is an accurate depiction of the left branch of the strategy in Fig. 10 from the perspective of the system. Observe that, in Fig. 11, the reader concerned is loaded with the keys of the ePassport present, corresponding to action $\overline{passport}(c_1)$ and $\overline{reader}(c_3)$, and then the attacker relays messages between the ePassport and the reader, where the actions of the honest reader and ePassport correspond to the five actions in the left branch of the strategy, which eventually result in a non-error message. In contrast, if the run of the reader was loaded with different keys, as in the left branch of the specification in Fig. 10, then the final message would be an error. Hence this is the distinguishing strategy, when, according to the specification, the honest session of the reader will be, or should have been with high likelihood, present elsewhere. This is how we interpret what the right branch of the strategy accounts for.

The MSC conveys different information to the depiction of the strategy. The fake reader and fake ePassport are implicit in this symbolic model, since the attacker is the observer interacting with the observables of the honest participants. However, in the MSC we make the steps that an attacker must perform explicit, which assists us with communicating the attack to stakeholders. ICAO and ISO experts acknowledged they understood the attack presented in this way for the purpose of responsible disclosure [Del19].

The MSC diagram in Fig. 11 was easy for students to understand as the basis of an implementation to demonstrate the feasibility of the attack and for the evaluation of the wider socio-technical context[3]. The dissemination of that broader study is ongoing. We found off-the-shelf readers typically keep the keys of an ePassport loaded until new keys are loaded, making the attack strategy easier than expected by allowing multiple attempts at reidentifying ePassports. The attack may even be triggered inadvertently, simply by having an ePassport in the proximity of a reader loaded with the wrong keys, in fact, this happened live in a lecture where the objective was just to show ePassports could be read by off-the-shelf apps – the fact that an ePassport in the vicinity inadvertently triggered an error message told us it was not the same as the one the was previously used. This makes the vulnerability uncovered a real risk, particularly, when the ICAO 9303 standard is being deployed for multiple purposes, not just at airport gates; a risk confounded by a proliferation of powerful RFID readers and covert components, such as 180 micron thick overlay cards [And21]. Stakeholders implementing and deploying ePassport readers should be aware of this risk and possible mitigation strategies. One may argue that there are more serious side-channel attacks; however, side-channel attacks may be addressed by better implementations of ePassports, while the vulnerability we uncover will remain since it is tied to the specification.

In science in general, interpreting the outputs of a model requires domain expertise. Security is no exception; hence we expect that there exist further attacks on the BAC protocol that can be uncovered by selecting a distinguishing formula and interpreting it meaningfully in a range of socio-technical scenarios where eDocuments play a role. For example, for the attacks presented several actions can be permuted, such as the first two actions in ς creating the ePassport and reader channels, without changing fundamental strategy. Some of these permutations may give rise to slightly different scenarios to those described. The essence will however be the same – the attacker can chose between multiple combinations of devices in order to attempt an authentication session, when, ideally, only one combination of ePassport and reader run should work.

4.3. Without else branches there is still an attack.

Some papers that analyse the BAC protocol drop the else branch [CDS20]. This is convenient since not all tools and methods handle else

[3] https://github.com/bboyifeel/bac-protocol-unlinkability-exploitation Repository with implementations of fake reader and fake ePassport for testing the feasibility of attacks. Maintained by Igor Filimonov.

branches. For example, we may instead try to use the following model of an ePassport in the system and specification processes.

$$\begin{aligned} P_{BAC}^{no\_else}(c, ke, km) \triangleq \quad & \nu nt.\overline{c}\langle nt\rangle.c(y). \\ & [\mathtt{snd}(y) = \mathtt{mac}(\mathtt{fst}(y)\,, km)][nt = \mathtt{fst}(\mathtt{snd}(\mathtt{dec}(\mathtt{fst}(y)\,, ke)))] \\ & \nu kt.\mathtt{let}\, m = \{\langle nt,\ \langle \mathtt{fst}(\mathtt{dec}(\mathtt{fst}(y)\,, ke)),\ kt\rangle\rangle\}_{ke}\ \mathtt{in}\, \overline{c}\langle m, \mathtt{mac}(m, km)\rangle \end{aligned}$$

However, even for this variant, using bisimilarity, we discover an attack on unlinkability. To see why, observe that if authentication succeeds, instead of checking the final message sent by the ePassport is not an error, it is sufficient to check that some message is sent at that point. If no message was transmitted, then we can infer the tests on the nonce failed.

A classical $\mathcal{FM}$ formula distinguishing the system from the specification, obtained using the process above to model the ePassport in the scheme at the top of this section, is as follows.

$$\begin{aligned} \varphi' \triangleq \quad & \langle \overline{reader}(c_1)\rangle\langle \overline{reader}(c_2)\rangle\langle \overline{passport}(c_3)\rangle\langle \overline{c_3}(nt)\rangle\Big(\\ & \qquad \langle c_1\, nt\rangle\langle \overline{c_1}(w)\rangle\langle c_3\, w\rangle\langle \overline{c_3}(z)\rangle\mathtt{tt} \\ & \quad \wedge \quad \langle c_2\, nt\rangle\langle \overline{c_2}(w)\rangle\langle c_3\, w\rangle\langle \overline{c_3}(z)\rangle\mathtt{tt}\ \Big) \end{aligned}$$

Notice the difference compared to φ is that φ' does not need to test to check that z is not an error. Indeed, when we reach the final action of the distinguishing strategy, the specification is unable to perform any action on channel c_3, hence cannot simulate the behaviour of the system.

4.4. **From bisimilarity to notions of similarity.** As we have already pointed out for Fig. 9, the leader in the strategy in Fig. 10 is always the same. Thus we do not require the full power of bisimilarity to discover either of these attack strategies, nor even the strategy described by φ' in Sec. 4.3. Indeed, to specify this unlinkability problem, it is sufficient to use a similarity preorder, which is obtained from strong early bisimilarity in Def. 2.9 by dropping the requirement that the relation is symmetric. To be explicit, in the specification of unlinkability, we could employ the following notion of similarity instead of bisimilarity.

Definition 4.2 (similarity)**.** *A relation between extended processes $\mathcal{R}$ is a strong early simulation only if, whenever $A\ \mathcal{R}\ B$ the following hold:*

- *A and B are statically equivalent.*
- *If $A \xrightarrow{\pi} A'$ there exists B' such that $B \xrightarrow{\pi} B'$ and $A'\ \mathcal{R}\ B'$.*

Process P is simulated by processes Q, written $P \leq Q$, whenever there exists a strong early simulation $\mathcal{R}$ such that $P\ \mathcal{R}\ Q$.

In both the old scheme for unlinkability, as communicated in CSF'10, and in our new updated scheme introduced in this section, it is trivial that the process modelling the specification, e.g., $Spec_{BAC}$, is simulated by the system process $System_{BAC}$, i.e., $Spec_{BAC} \leq System_{BAC}$ holds. To see why, intuitively, observe that if we have full control of the system we can always make it behave like the specification by never using the same ePassport twice thus the specification can be simulated by the system. Hence, the problem of checking unlinkability, when cast as a similarity problem, can be formulated by the problem of checking whether $System_{BAC} \leq Spec_{BAC}$, i.e., checking whether any observable behaviours the system can perform can be simulated by behaviours of the specification. Stated as a theorem, we have the following which tightens Theorem 2.4, where the proof follows from the same strategy as presented in Fig. 9.

Theorem 4.3. *$System_{BAC} \leq Spec_{BAC}$ does **not** hold.*

It is helpful to know that similarity is sufficient for this problem, since similarity preorders have compelling attacker models, e.g., in terms of probabilistic testing semantics [DvGHM08]. Indeed, it is standard in cryptography to assume that an adversary has the power of a probabilistic polynominal-time Turing machine, even if the protocol does not contain probabilistic choices, which is reflected in power made available to the attacker by adopting similarity.

Remark 4.4. *There are several variants of similarity in the linear-time / branching-time spectrum [vG01], which, as touched on in Sec. 2.5, could be compelling choices for modelling the capabilities of attackers. We argue that a more broadly applicable design decision would be to employ a stronger notion of similarity called failure similarity. For example, failure similarity can distinguish process* $!\overline{a}\langle go\rangle.\overline{a}\langle error\rangle \mid !\overline{a}\langle go\rangle$ *from* $!\overline{a}\langle go\rangle.\overline{a}\langle error\rangle$*, whereas similarity in Definition 4.2 cannot distinguish these processes. Thus we can test that an event does not happen, e.g., by using a timeout [vG21], which is a distinction that cannot be made using similarity.*

Such additional expressive power is not required in order to detect unlinkability attacks on the BAC protocol, but might be useful in some scenarios where, for example, the attacker can explicitly observe an error due to the presence of a message, but cannot explicitly observe a success. In such scenarios, a success can be inferred by observing that an error does not occur within an expected time window. For example, the ICAO 9303 specification of the PACE protocol [MRT15] only requires the ePassport to send an error message at the end of an unsuccessful authentication session, but does not require it to send any message if the session results in authentication being successful; hence successful authentication from the perspective of an ePassport can be inferred by the absence of an error message at the end of the session within an expected time window. However, when we model the PACE protocol in Sec. 5, we use explicit observables for success so as to align with related work, thereby avoiding unnecessary debate about whether our attacks are particular to how we model the PACE protocol (they are not). Thus, it is safest to verify unlinkability with respect to bisimilarity, which covers all such attacks, including the richer strategy in Fig. 8. Recall, in Sec. 3, that, by using domain knowledge, we were able to assign a practical meaning to the strategy in Fig. 8, which failure similarity does not detect.

5. Unlinkability of the PACE protocol

We address the public communication from the office of the secretary general of ICAO, discussed in the introduction, which challenges whether the unlinkability vulnerability discovered on the BAC protocol is valid for more recent versions of the ICAO 9303 ePassport standard [MRT15]. This is a reasonable question, since the 7^{th} edition of the ICAO 9303 standard recommends the Password Authenticated Connection Establishment protocol (PACE), as a more secure alternative to BAC.

The PACE protocol does improve on the security of the BAC protocol, making attacks giving access to private data stored on the chip more difficult. For example, the PACE protocol satisfies *forward secrecy* [BFK09, CGIP12]; whereas the BAC protocol does not, that is: if an attacker intercepts ciphertexts in anticipation of, in the future, discovering the key for the ePassport, then she cannot go back and use the key to discover the session key and reveal the encrypted secrets from those old runs of the protocol.

The PACE protocol also eliminates unlinkability vulnerabilities caused by using different errors when the protocol fails for different reasons, as was the case for an implementation of the BAC protocol for French ePassports. Thus we believe that some of the most serious types of attack on unlinkability exploiting the BAC protocol have been addressed in the PACE protocol, where such attacks on the BAC protocol allow an ePassport holder with an implementation interpreting the specification in a particular way to be tracked forever after the messages from one session with a trusted reader have been intercepted.

The clause of the standard that restricts the use of error messages is the following line in section 4.4.2 of part 11 of the ICAO 9303 standard.

> "An eMRTD chip that supports PACE SHALL respond to unauthenticated read attempts (including selection of (protected) files in the LDS) with "Security status not satisfied" (0x6982)."

The above explicit statement is an improvement over the specification of the BAC protocol; however there are still unlinkability vulnerabilities in the PACE protocol as specified in the ICAO 9303 standard. Similarly to the vulnerability in the BAC protocol, studied throughout previous sections, there are vulnerabilities in the PACE protocol valid due to differences between a successful and failed authentication session observable to an attacker, such as the presence of the error message highlighted above. We formally analyse this vulnerability using bisimilarity following our revised approach to unlinkability justified in Sec. 4.

5.1. **The PACE protocol.** There are multiple ways to interpret the PACE protocol since it has various operational modes that permit a number of cryptographic primitives to be used at each stage for establishing shared keys. We model here the generic mapping which uses a Diffie-Hellman key exchange. The message exchange, presented in Fig. 12, follows closely related work communicated

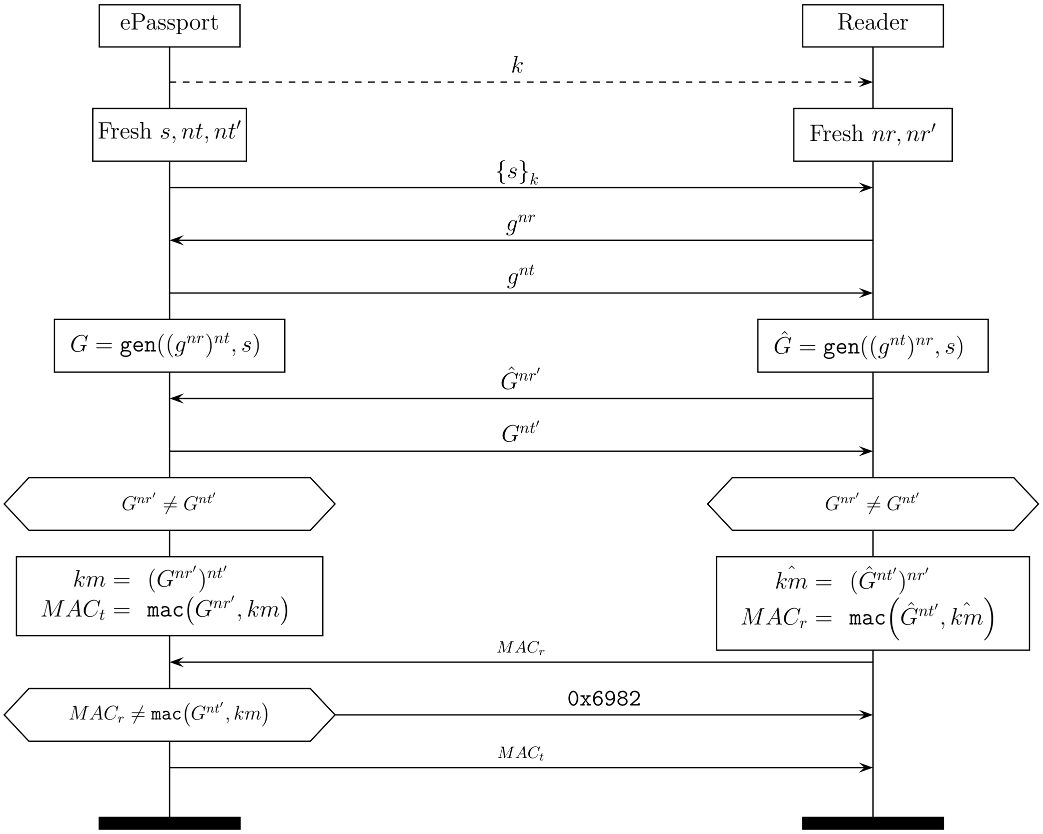
ePassport
Reader
k
Fresh s, nt, nt'
Fresh nr, nr'
$\{s\}_k$
g^{nr}
g^{nt}
$G = \mathtt{gen}((g^{nr})^{nt}, s)$
$\hat{G} = \mathtt{gen}((g^{nt})^{nr}, s)$
$\hat{G}^{nr'}$
$G^{nt'}$
$G^{nr'} \neq G^{nt'}$
$G^{nr'} \neq G^{nt'}$
$km = (G^{nr'})^{nt'}$
$MAC_t = \mathtt{mac}(G^{nr'}, km)$
$\hat{km} = (\hat{G}^{nt'})^{nr'}$
$MAC_r = \mathtt{mac}\left(\hat{G}^{nt'}, \hat{km}\right)$
MAC_r
$MAC_r \neq \mathtt{mac}(G^{nt'}, km)$
0x6982
MAC_t

Figure 12: The PACE protocol, using a generic mapping based on Diffie-Hellman Key Agreement.

in the Journal of Computer Security [HBD19], thereby avoiding unnecessary debate on how the protocol is interpreted.

The message flow in Fig. 12 is as follows.

(1) The ePassport shares information for generating a key k with the reader, usually via an OCR session with the biometric page of the ePassport. This is represented by the dotted line at the top of the figure. PACE uses better sources of randomness than BAC, however this does not affect our unlinkability analysis.
(2) The ePassport key uses the key k to transmit an encrypted nonce $\{s\}_k$ to the reader.
(3) The ePassport and reader employ one of several operational modes to create additional randomness for each session. We model the "generic mapping" operational mode which employs a Diffie-Hellman handshake. This information is used to generate shared key $G = \mathtt{gen}((g^{nr})^{nt}, s)$, where $\mathtt{gen}(\cdot, \cdot)$ is key generation function. Notice $G = \hat{G}$ in Fig. 12.
(4) A Diffie-Hellman handshake is performed using G as the generator, which is used to compute a MAC key km. Also an encryption key for the secure messaging phase, which we do not model, is generated at this point. Again $km = \hat{km}$. The checks $G^{nr} \neq G^{nt}$ at this point avoid reflection attacks, where an ePassport or reader is used to authenticate itself.
(5) Finally the ePassport and reader exchange and verify MACs, using the MAC key km, authenticating the public keys exchanged in the previous step. If authentication fails at this point an error message is produced. Notice we include the above mentioned error message (0x6982) if authentication fails at the end of the protocol.

5.2. **PACE in the applied π-calculus.** For the PACE protocol we require an extended message theory. We require symmetric encryption, where decryption is not detectable (modelled by the same equations as employed for the BAC protocol). For the Diffie-Hellman exchanges we require exponentiation and also a key generating map, which acts like a two parameter hash function. As for the BAC protocol, MACs are modelled as a two parameter hash function. This message theory is presented below.

$$
\begin{array}{rllr}
M, N ::= & x & & \text{variable} \\
\mid & \mathtt{mac}(M, N) & & \text{mac} \\
\mid & \mathtt{gen}(M, N) & & \text{generator} \\
\mid & M^N & & \text{exponentiation} \\
\mid & \{M\}_N & & \text{encryption} \\
\mid & \mathtt{dec}(M, N) & & \text{decryption}
\end{array}
\qquad
\begin{array}{c}
\mathtt{dec}(\{M\}_K, K) =_{E'} M \\[1em]
\{\mathtt{dec}(M, K)\}_K =_{E'} M \\[1em]
\left(M^N\right)^K =_{E'} \left(M^K\right)^N
\end{array}
$$

The ePassport and reader for the PACE protocol can be modelled in the applied π-calculus as follows.

$$\begin{array}{rl@{\qquad\qquad}rl}
P_{PACE}(c,k) \triangleq & \nu s.\overline{c}\langle\{s\}_k\rangle.c(x). & V_{PACE}(c,k) \triangleq & c(x).\nu nr.\overline{c}\langle g^{nr}\rangle.c(y). \\
& \nu nt.\overline{c}\langle g^{nt}\rangle.c(y). & & \mathtt{let}\, G = \mathtt{gen}(\mathtt{dec}(x,k)\,, y^{nr})\ \mathtt{in} \\
& \mathtt{let}\, G = \mathtt{gen}(s, x^{nt})\ \mathtt{in} & & \nu nr'.\overline{c}\left\langle G^{nr'}\right\rangle.c(z). \\
& \nu nt'.\overline{c}\left\langle G^{nt'}\right\rangle & & \left[G^{nr'} \neq z\right]\mathtt{let}\, km = z^{nr'}\ \mathtt{in} \\
& \left[G^{nt'} \neq y\right]c(z). & & \overline{c}\langle\mathtt{mac}(z,km)\rangle \\
& \mathtt{let}\, km = y^{nt'}\ \mathtt{in} & & c(m).\left[m = \mathtt{mac}\left(G^{nr'}, km\right)\right]c(n) \\
& \overline{c}\langle\mathtt{mac}(z,km)\rangle & & \\
& \mathtt{if}\, z = \mathtt{mac}\left(G^{nt'}, km\right) & & \\
& \mathtt{then}\, \overline{c}\langle\mathtt{mac}(z,km)\rangle & & \\
& \mathtt{else}\, \overline{c}\langle error\rangle & &
\end{array}$$

Notice only the ePassport features an error message if authentication fails in the final step. Also, we add a dummy event $c(n)$ at the end of the reader session, for the sake of modelling that a reader will proceed to do something after successfully authenticating. This is to align with related work in communicated in the Journal of Computer Security [HBD19], in order to facilitate a comparison of results obtained.

Using the above processes, and our revised scheme for unlinkability in the previous section we obtain the following result confirming there are attacks on the unlinkability of PACE.

Theorem 5.1. *$System_{PACE} \not\sim Spec_{PACE}$, where*

$$\begin{aligned}
System_{PACE} &\triangleq\ !\nu k.!\left(\nu c.\overline{passport}\langle c\rangle.P_{PACE}(c,k) \mid \nu c.\overline{reader}\langle c\rangle.V_{PACE}(c,k)\right) \\
Spec_{PACE} &\triangleq\ !\nu k.\left(\nu c.\overline{passport}\langle c\rangle.P_{PACE}(c,k) \mid \nu c.\overline{reader}\langle c\rangle.V_{PACE}(c,k)\right)
\end{aligned}$$

Proof. Consider the following $\mathcal{FM}$ formula.

$$\begin{aligned}
\xi \triangleq\ & \langle\overline{reader}(c_1)\rangle\langle\overline{reader}(c_2)\rangle\langle\overline{passport}(c_3)\rangle\langle\overline{c_3}(t)\rangle\Big(\\
& \quad \langle c_1\, t\rangle\langle\overline{c_1}(u)\rangle\langle c_3\, u\rangle\langle\overline{c_3}(v)\rangle\langle c_1\, v\rangle\langle\overline{c_1}(w)\rangle\langle c_3\, w\rangle\langle\overline{c_3}(x)\rangle\langle c_1\, x\rangle\langle\overline{c_1}(y)\rangle\langle c_3\, y\rangle\langle\overline{c_3}(z)\rangle(z \neq error) \\
& \quad \wedge \langle c_2\, t\rangle\langle\overline{c_2}(u)\rangle\langle c_3\, u\rangle\langle\overline{c_3}(v)\rangle\langle c_2\, v\rangle\langle\overline{c_2}(w)\rangle\langle c_3\, w\rangle\langle\overline{c_3}(x)\rangle\langle c_2\, x\rangle\langle\overline{c_2}(y)\rangle\langle c_3\, y\rangle\langle\overline{c_3}(z)\rangle(z \neq error)\ \Big)
\end{aligned}$$

Since $System_{PACE} \models \xi$, but $Spec_{PACE} \not\models \xi$, by Theorem 3.8, $System_{PACE} \not\sim Spec_{PACE}$. □

The formula ξ proving that unlinkability does not hold for the PACE protocol follows a similar pattern to the formula φ, used in the previous section to certify that BAC fails unlinkability. In this strategy, the system starts two readers and an ePassport with the same keys. The specification can only follow using a strategy where one of the readers will fail authentication. To win this game, the system simply chooses to authenticate with the reader that is expected to fail in the specification. That reader will obviously successfully authenticate in the system, thereby concluding our distinguishing strategy.

5.3. **Another attack strategy from related work.** Infinitely many distinguishing strategies exist violating the unlinkability of the PACE protocol. Indeed related work [HBD19] discovered a violation of the unlinkability of the PACE protocol, that can be described as a trace. An attack on unlinkability formulated using trace equivalence is always also an attack on bisimilarity. The added value that our methodology brings to that attack, is that we can certify their attack by using the following classical $\mathcal{FM}$ formula, describing the distinguishing trace discovered in the above mentioned related work. The $\mathcal{FM}$ formulae characterising trace equivalence are those consisting of diamond modalities only ending with some formula that does not involve modalities.

$$\begin{aligned}\vartheta \triangleq\ & \langle \overline{reader}(c_1)\rangle\langle \overline{reader}(c_2)\rangle\langle \overline{passport}(c_3)\rangle\langle \overline{c_3}(t)\rangle\langle c_1\, t\rangle\langle c_2\, t\rangle \\ & \langle \overline{c_1}(u_1)\rangle\langle \overline{c_2}(u_2)\rangle\langle c_2\, u_1\rangle\langle c_1\, u_2\rangle\langle \overline{c_1}(v_1)\rangle\langle \overline{c_2}(v_2)\rangle\langle c_2\, v_1\rangle\langle c_1\, v_2\rangle \\ & \langle \overline{c_1}(w_1)\rangle\langle \overline{c_2}(w_2)\rangle\langle c_2\, w_1\rangle\langle c_1\, w_2\rangle\langle \overline{c_1}(x_1)\rangle\langle \overline{c_2}(x_2)\rangle\langle c_2\, x_1\rangle\langle c_1\, x_2\rangle \\ & \langle \overline{c_1}(y_1)\rangle\langle \overline{c_2}(y_2)\rangle\langle c_2\, y_1\rangle\langle c_1\, y_2\rangle\langle \overline{c_1}(z_1)\rangle\langle \overline{c_2}(z_2)\rangle\langle c_2\, z_1\rangle\langle c_2\, m\rangle\mathtt{tt}\end{aligned}$$

In the strategy described by the formula above, the encrypted nonce sent by the ePassport at the beginning of the protocol is replayed to two different readers. The two readers are then used to authenticate each other, exploiting the fact that the protocol is symmetric in the role of the reader and ePassport. Authentication will only be successful if both readers have the same keys, otherwise either reader will fail the check on the MAC at the final step of the PACE protocol. Thus, assuming that a reader does something after authentication, we learn whether both readers talked with the same ePassport.

The attack ϑ is quite different from attack ξ. Notice ϑ requires two readers to be fully active during the attack, not simply present in principle, as a choice in a game, suggesting that it may be more difficult to exploit, despite being described as a trace. Also, notice the attack can be mitigated in several ways, e.g., by initially responding to the ePassport in the same way regardless of whether it authenticates or not, hence, since an ePassport cannot also be authenticated at the same time, there is no way to continue with the secure messaging phase. An alternative fix preventing this attack is proposed in related work [HBD19], modifying the protocol such that additional role specific information is added to the handshake. However, these fixes will not mitigate the more serious problem described in ξ, so do not really improve unlinkability.

6. Related and future work

A closely related paper, that is not already covered by remarks in the body of the paper was communicated in S&P'18 [CKR18]. That paper announces the discovery of attacks on the BAC protocol using the bounded trace equivalence checker DeepSec, but without further discussion. An interesting difference between that paper and the current work is that, while we build on the original formulation of unlinkability, as communicated in CSF'10 [ACRR10]; the S&P'18 paper proposes another model where, instead of using a specification process, unlinkability is modelled in terms of two systems where the number of identical users in the system differ. In their alternative model, one process models two sessions featuring the same ePassport twice, which is compared to another process featuring two sessions each featuring a different ePassport. That is, unlinkability is formulated such that the following two processes are compared, using trace equivalence.

$$Diff \triangleq \nu ke.\nu km.\Big(V_{BAC}(c_r, ke, km) \mid P_{BAC}(c_p, ke, km)\Big) \mid \nu ke.\nu km.\Big(V_{BAC}(c_r, ke, km) \mid P_{BAC}(c_p, ke, km)\Big)$$

$$Same \triangleq \nu ke.\nu km.\Big(V_{BAC}(c_r, ke, km) \mid P_{BAC}(c_p, ke, km) \mid V_{BAC}(c_r, ke, km) \mid P_{BAC}(c_p, ke, km)\Big)$$

In the above c_r and c_p are used as fixed channels for communications with all ePassports or all readers respectively. Attack traces discovered using the above method, communicated in S&P'18, can be used to confirm that two sessions are certainly not with the same ePassport. In particular, consider the following formula describing a trace that holds for the first process but does not hold for the second process.

$$\begin{aligned}Diff \vDash\ & \langle \overline{c_p}(n_1)\rangle\langle \overline{c_p}(n_2)\rangle\langle c_r\, n_2\rangle\langle c_r\, n_2\rangle \\ & \qquad\langle \overline{c_r}(m_1)\rangle\langle \overline{c_r}(m_2)\rangle\langle c_p\, m_1\rangle\langle c_p\, m_2\rangle\langle \overline{c_p}(e_1)\rangle\langle \overline{c_p}(e_2)\rangle(e_1 = error \wedge e_2 = error)\end{aligned}$$

In the first line of the above trace, the first two messages correspond to sending two nonces, and only the second nonce is fed as an input to both readers. On the second line of the formula, the protocol continues for both sessions and the protocol ends with both ePassports sending error messages. If both readers are using the same nonce, then both ePassports can only send error messages at the last step if both ePassports are different; if both were the same ePassport then one of the two sessions would successfully authenticate, hence there could not have been two error messages. I.e., it is impossible for *Same* to satisfy the above formula.

The limitation we see is the above mentioned approach communicated in S&P'18 discovers attack traces that cannot be used to positively confirm that two sessions are with the same ePassport. What we mean is that there is no trace that holds for the process *Same* that does not hold for the process *Diff*; but for a trace-like attack on unlinkability we should surely be able to provide a trace that links two sessions. Thus we should be careful interpreting the above result — it does not mean that the above method discovers an attack on unlinkability that is in the form of a trace.

Further discussion on the above model of unlinkability appears in the conference version of this paper [FHMS19], where it is clarified that the above limitation is due to modelling decisions and is not a feature of the DeepSec tool that the S&P'18 paper showcases. The DeepSec tool can also be used to verify finite formulations of the unlinkability problem using our preferred "system v.s. specification" approach — in which case DeepSec discovers no attacks that are in the form of a trace for the BAC protocol. In particular, DeepSec can verify that *Diff* is trace equivalent to the following process where either there is a choice between starting the second session with the same keys as the first or with the new keys, which is a bounded approximation of the system following established schemes for unlinkability.

$$\begin{aligned} \nu ke.\nu km.\Big(\; & V_{BAC}(c_r, ke, km) \mid P_{BAC}(c_p, ke, km) \mid \\ & \big(V_{BAC}(c_r, ke, km) \mid P_{BAC}(c_p, ke, km)\big) + \nu ke, km.\big(V_{BAC}(c_r, ke, km) \mid P_{BAC}(c_p, ke, km)\big)\Big) \end{aligned}$$

In the above process, + is non-deterministic choice, which is easy to add to the applied π-calculus.

An approach similar to the approach communicated in S&P'18, where two systems are compared in which users in a system are permuted, has been thoroughly investigated and demonstrated to be the preferable approach for formulating voter privacy, which is a property of eVoting systems [DKR09]. We should clarify that we are not arguing against using a "permutations of a system" approach to voter privacy. What we are arguing is that the "system vs. specification" approach adopted in the current paper and in the CSF'10 paper is appropriate for unlinkability, since if an attack trace is discovered the attack trace will be able to positively confirm that two sessions are with the same ePassport, i.e., the sessions will be linked.

6.1. **A preliminary discussion on mitigation strategies.** We summarise here three quite different mitigation strategies and present some preliminary findings.

6.1.1. *Timeouts.* A mitigation strategy, which we have recommended to stakeholders, is to guarantee that implementations of readers only hold ePassport keys for a short period of time. This would render the reader useless in an attack strategy where the goal is to reidentify someone in the future. In Fig. 11, the time to keep small is indicated by t: the time between the reader being prepared with they keys of a particular passport and the RFID session being triggered. This eliminates use cases where an ePassport holder who has recently passed through a checkpoint is reidentified, but might not mitigate other use cases. For example, a user may be expected to approach a reader, e.g., to provide their identity at a service desk, and hence the reader is loaded with a particular key in a time window. In that case, an attacker may attempt to find who was intended to be at the service desk without them being present. An extensive study of use cases emerging we push to future work.

The above mitigation strategy can be modelled by imposing causal dependencies in our model. A timeout built into the reader would allow a single run of an ePassport to occur before the reader session times out. Thus the creation of a new run of an ePassport causally depends on any events of an ongoing run of a reader involving the same ePassport, which must first be completed. In addition, we can assume that ePassport runs themselves are sequentially ordered, by the nature of the chip. We may also assume that the readers are loaded with the keys of a single ePassport sequentially, e.g., along a path through checkpoints. Surprisingly, forcing all these causal assumptions would still not be enough to prevent the attack ς on unlinkability presented in Section 4.2, since we have not stipulated that a single ePassport run can be held open for a length of time sufficient for the attacker to choose between relaying message to one of two readers in a sequence, both of which may successfully authenticate the ePassport. Thus there must be a fourth causal dependency imposed to prevent attacks, which is not enforced by a timeout for the reader: we should prevent a new reader from being initiated before any run of an ePassport involving the same keys terminates. This does not model a timeout imposed by the reader, but instead a timeout imposed in practice by how long an attacker can keep a device in the vicinity of an ePassport holder.

If we make all four causal assumptions in the paragraph above, the effect is each run of an ePassport and its corresponding reader session are sequentialised together, in parallel. That is, a run of an ePassport and a reader must both be used entirely before any new run can be created involving the keys of the same ePassport. This can be modelled by making use of a Kleene star operator ¡, proposed in related work [Bae21], in the following alternative scheme for the system.

$$!\nu\vec{k}.¡\Big(\nu c.\overline{passport}\langle c\rangle.\mathit{Prover}(c,\vec{k}) \mid \nu c.\overline{reader}\langle c\rangle.\mathit{Verifier}(c,\vec{k})\Big)$$

Our hypothesis is that the above process is bisimilar to the scheme for the specification for both the BAC and PACE protocols. An account of the semantics of ¡ in this context and a proof of this claim are pushed to future work.

6.1.2. *Obscuring messages.* Another alternative is to probabilistically encrypt the error message, or produce random noise when the ePassport fails to authenticate the reader. Note this is within the scope of the BAC protocol specification, since the specification does not fix the form of the error message. Nevertheless, to our best knowledge, real ePassports implementing the BAC protocol send errors as constant plaintext messages. This mitigation strategy would only be effective for use cases where the attacker cannot observe the consequences of using an eDocument. For example, one may consider a “polite” registration system where you are offered to register using your electronic identity, but choosing not to or not succeeding to do so is permitted. Perhaps those who do not provide their electronic identities will be picked up by other safety nets such as a human attendant who later checks participants against a register rather than as an immediate effect such as the opening of a gate.

Such a refinement of the BAC protocol where error messages are obscured, does in fact satisfy unlinkability, as long as nothing happens after executing the BAC protocol. We can model one such variant of the BAC protocol as follows.

$$\begin{array}{ll}
P^{fixed}_{BAC}(c, ke, km) \triangleq & \nu nt.\overline{c}\langle nt\rangle.c(y). \\
& \quad \texttt{if}\,\texttt{snd}(y) = \texttt{mac}(\texttt{fst}(y), km)\ \texttt{then} \\
& \qquad \texttt{if}\, nt = \texttt{fst}(\texttt{snd}(\texttt{dec}(\texttt{fst}(y), ke)))\ \texttt{then} \\
& \qquad\quad \nu kt.\texttt{let}\, m = \{\langle nt,\ \langle \texttt{fst}(\texttt{dec}(\texttt{fst}(y), ke)),\ kt\rangle\rangle\}_{ke}\ \texttt{in} \\
& \qquad\quad \overline{c}\langle m, \texttt{mac}(m, km)\rangle \\
& \qquad \texttt{else}\,\nu r.\texttt{let}\, m = \{\langle r,\ error\rangle\}_{ke}\ \texttt{in}\,\overline{c}\langle m, \texttt{mac}(m, km)\rangle \\
& \quad \texttt{else}\,\nu r.\texttt{let}\, m = \{\langle r,\ error\rangle\}_{ke}\ \texttt{in}\,\overline{c}\langle m, \texttt{mac}(m, km)\rangle
\end{array}$$

Let $System^{fixed}_{BAC}$ and $Spec^{fixed}_{BAC}$ denote, respectively, the system and specification for unlinkability of BAC where the above model of the ePassport role is used. For this model, we have a proof that unlinkability does hold, i.e., $System^{fixed}_{BAC} \sim Spec^{fixed}_{BAC}$. The publication of a proof for this claim is pushed to future work, in the interest of focussing on attacks in this paper.

A problem with the above refinement of the BAC protocol satisfying unlinkability is that making BAC unlinkable does not guarantee that the whole ePassport protocol satisfies unlinkability. The BAC protocol is just for authentication and establishing a session key. After authenticating the ePassport proceeds with a *secure messaging phase* that uses the session key to transmit personal data stored in the ePassport [MRT15]. Thus it is sufficient for an attacker to look at whether the protocol proceeds with secure messaging or not in order to determine whether authentication was successful. That knowledge can be used to the same effect as observing whether or not an error message was sent. Thus, for the above fix to be fully effective even in a "polite" system, the secure messaging phase should proceed even if the ePassport does not authenticate, transmitting dummy data indistinguishable to an observer from the real data. Such a mitigation strategy is outside the scope of the current ICAO specification and does not significantly improve unlinkability for the standard use cases for ePassports.

6.1.3. *Using one-time keys.* An arguably better mitigation strategy is to use Time-based One-time Passwords (TOTP) to make the six-digit key for PACE change periodically, as explored in the thesis of our student [Fil20]. Verification of that strategy is immediate, since TOTP has the effect of generating a new key for every run of the protocol, making the system and specification trivially bisimilar. Implementing TOTP has further security and privacy advantages, forgoing other attacks on the system, including some social attacks, and has been deployed in card form for ePayments, so could be easily integrated into electronic ID cards implementing the ICAO 9303 standard. The challenge for ePassports is more likely to be at the policy level, since questions may be raised at international checkpoints if countries do not agree that TOTP is acceptable technology. Dissemination of the socio-technical evaluation of this mitigation strategy we push to future work.

6.2. **Further risks to unlinkability.** There are many potential privacy risk for the PACE protocol that are not directly captured by the symbolic models in this work. Some are simple to exploit, such as the fact that the PACE protocol offers several different operational modes. In order for a reader to determine the appropriate operational model, before starting the protocol, the ePassport declares the operational modes of the PACE protocol that it implements. In an environment, such as an airport, where many different ePassports implementing different operational modes coexist, a user may be tracked with a probability better than a random guess.

The actual probability of guessing correctly that the same ePassport is involved in two sessions depends on the number of different implementations of ePassport and the expected movements of their holders. We expect the advantage gained by such a strategy to be non-negligible and it is standard in security and privacy models that gaining a non-negligible advantage counts as an attack. Permitting may different implementations of protocols is an oversight of the ICAO 9303 standard.

7. Conclusion

This paper confirms there are attacks on the authentication protocols proposed in the latest ICAO 9303 specification for ePassports. Both the BAC and PACE protocols feature attacks that can be described as a distinguishing strategy in a game played according to a specification of what it means to satisfy unlinkability. Attacks on the BAC protocol, communicated in Theorem 2.4 and Theorem 4.1, are interesting since the former resolves flawed claims that no such attack exists according to a formulation of unlinkability dating back to CSF'10; while the latter irons out limitations of that original model concerning the capabilities of an attacker to distinguish messages from different readers and ePassports.

An interesting aspect of the new attack we discover on the PACE protocol, as formulated in Theorem 5.1, is that it is not mitigated by defensive strategies that may be introduced to mitigate attacks previously discovered using trace equivalence as communicated in the Journal of Computer Security [HBD19] (e.g., sequentialising reader sessions will not mitigate the newly discovered attack, nor will adding role information distinguishing messages of the same form originating from a reader and from an ePassport). Furthermore, the attacks discovered previously using trace equivalence require two honest readers to actively participate in the attack, whereas the new attack discovered using bisimilarity requires only one honest reader to actively participate in the attack, meaning that the attack can be realised in a broader range of scenarios. This observation challenges claims communicated in the Journal of Computer Security, where it is argued that bisimilarity is too strong and hence trace equivalence should be employed. Their argument is provided to support their model that relies on trace equivalence in order to prove that unlinkability of the BAC protocol holds. Since their results lead to contradictory advice compared to ours, differences are worth clarifying.

The crux of their argument is based on remarks communicated in CSF'10 [ACRR10] claiming that bisimilarity may distinguish processes due to their internal state — an argument we contest since bisimilarity is all about the games played between the adversary and its environment using observations only; and never can distinctions be made based on differences in internal state (that is the point of such observational equivalences). Presenting multiple viewpoints is healthy for academic debate; however, the fact that the attacks we discover using bisimilarity are not spurious and furthermore are easier to realise is evidence that trace equivalence is insufficient for verifying interactive systems such as security protocols. To further support our argument that the use of bisimilarity (or, as a compromise, a suitable notion of similarity, as discussed in Sec. 4.4) is important for such security and privacy problems, we remark that it should not be a surprise to cryptographers that the adversary can play a strategy in a game to gain a non-negligible advantage, since related assumptions about the adversary are standard in the long-established school of *computational security* used to formally reason about cryptographic primitives. We quote R.L. Rivest on the topic of games in cryptography [MvOV96]:

> "Cryptography is also fascinating because of its game-like adversarial nature. A good cryptographer rapidly changes sides back and forth in his or her thinking, from attacker to defender and back. Just as in a game of chess, sequences of moves and countermoves must be considered until the current situation is understood."

We argue that the above remark should also hold for the *symbolic verification of cryptographic protocols*, which encompasses the methodology employed in this work. In the setting of this work, the game is defined by a bisimilarity problem specifying what it means for a protocol to satisfy unlinkability, while attacks are strategies improving the chances of an attacker winning the game. Such strategies can be conveniently described using modal logic formulae.

The insight obtained, concerning the existence of unlinkability attacks on the latest ePassport standards, is impactful for society, since ePassports and electronic ID cards are used by the citizens of over 150 countries at the time of writing. This amounts to an estimated 4 billion eDocuments in circulation that implement the ICAO 9303 standard. The manufactures and operators of readers should be made aware of mitigation strategies. Some preliminary ideas on mitigation strategies are discussed in Sec. 6.1.

Further to uncovering the above mentioned attacks, this paper makes multiple technical contributions. We proposed and justified a new scheme for unlinkability problems in Sec. 4. We proposed a definition of open bisimilarity for the applied π-calculus (Def. 3.5). Our formulations of weak and strong early bisimilarity and similarity (Defs. 2.3, 2.9, and 4.2) for the applied π-calculus are also new for the applied π-calculus, incorporating minor improvements facilitating verification, such as the adoption of a set of rules that make labelled transitions image finite. These improvements are conventional from the perspective of established work on the π-calculus; thus, in that direction, we simply modernise the applied π-calculus with respect to advances in the π-calculus literature. The formulation of the modal logic "classical $\mathcal{FM}$" is also new, as is the rather short and neat proof (in Appendix B) of the fact that classical $\mathcal{FM}$ characterises strong early bisimilarity for the applied π-calculus (Theorem 3.8). The methodology of using a classical $\mathcal{FM}$ formula to certify attacks is new, as is the use of open bisimilarity to discover distinguishing strategies that are transformed into attacks whenever the distinguishing strategy is not spurious. A reformulation of unlinkability removing τ-transitions has appeared in related work [HBD19], but the proof that the new specification preserves the original specification in terms of bisimilarity (Theorem 2.10) is new, as is the observation that this transformation reduces the unlinkability problem to a problem were image finiteness holds and strong notions of bisimilarity may be applied. In short, in order to solve this problem, we have set up a rich tool chain of methods that can be applied beyond the problem of analysing the unlinkability of the ICAO 9303 standard.

Acknowledgements. The formulation of the sequent calculus in Fig. 6 for calculating most general recipes is new, but should be attributed to Alwen Tiu. He made the observation that annotating deducibility constraints in a sequent calculus presentation allows us to calculate the recipes required for an attacker to deduce a message. We thank Davide Sangiorgi and the jury Luca Aceto, Jos Baeten, Patricia Bouyer-Decitre, Holger Hermanns, and Alexandra Silva for the explicit mention of the results communicated in this paper in the report on his CONCUR 2020 Test-of-Time Award [ABBD$^+$20]. Last, but not least, we congratulate Jos Baeten on the occasion of his retirement and dedicate this work to his leadership in the field of concurrency.

References

[ABBD$^+$20] Luca Aceto, Jos Baeten, Patricia Bouyer-Decitre, Holger Hermanns, and Alexandra Silva. CONCUR Test-Of-Time Award 2020 Announcement (Invited Paper). In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:3, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2020.5.

[ABF17] Martín Abadi, Bruno Blanchet, and Cédric Fournet. The applied pi calculus: Mobile values, new names, and secure communication. *Journal of the ACM*, 65(1):1:1–1:41, 2017. doi:10.1145/3127586.

[ABH+16] Gildas Avoine, Antonin Beaujeant, Julio Hernandez-Castro, Louis Demay, and Philippe Teuwen. A survey of security and privacy issues in epassport protocols. *ACM Comput. Surv.*, 48(3):47:1–47:37, 2016. doi:10.1145/2825026.

[ABW06] Suzana Andova, Jos C. M. Baeten, and Tim A. C. Willemse. A complete axiomatisation of branching bisimulation for probabilistic systems with an application in protocol verification. In Christel Baier and Holger Hermanns, editors, *CONCUR 2006 - Concurrency Theory, 17th International Conference, CONCUR 2006, Bonn, Germany, August 27-30, 2006, Proceedings*, volume 4137 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2006. doi:10.1007/11817949_22.

[ACRR10] Myrto Arapinis, Tom Chothia, Eike Ritter, and Mark Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *23rd IEEE Computer Security Foundations Symposium*, pages 107–121, 2010. doi:10.1109/CSF.2010.15.

[AG99] Martin Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 1999. doi:10.1006/inco.1998.2740.

[AHT17] Ki Yung Ahn, Ross Horne, and Alwen Tiu. A characterisation of open bisimilarity using an intuitionistic modal logic. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory, CONCUR 2017, September 5-8, 2017, Berlin, Germany*, volume 85 of *LIPIcs*, pages 7:1–7:17, 2017. doi:10.4230/LIPIcs.CONCUR.2017.7.

[And21] Ross Anderson. *Security Engineering (third edition)*. John Wiley & Sons, Inc., 2021.

[Bae21] David Baelde. *Contributions à la Vérification des Protocoles Cryptographiques*. habilitation thesis, Paris-Saclay, 2021.

[BB91] Jos C. M. Baeten and Jan A. Bergstra. Real time process algebra. *Formal Aspects of Computing*, 3(2):142–188, 1991. doi:10.1007/BF01898401.

[BB93] Jos C. M. Baeten and Jan A. Bergstra. Non interleaving process algebra. In Eike Best, editor, *CONCUR'93*, pages 308–323. Springer, 1993. doi:10.1007/3-540-57208-2_22.

[BB98] Jos C. M. Baeten and Jan A. Bergstra. Deadlock behaviour in split and ST bisimulation semantics. In Ilaria Castellani and Catuscia Palamidessi, editors, *Fifth International Workshop on Expressiveness in Concurrency, EXPRESS 1998, Satellite Workshop of CONCUR 1998, Nice, France, September 7, 1998*, volume 16 of *Electronic Notes in Theoretical Computer Science*, pages 61–74. Elsevier, 1998. doi:10.1016/S1571-0661(04)00117-3.

[BBK87] Jos C. M. Baeten, Jan A. Bergstra, and Jan Willem Klop. Ready-trace semantics for concrete process algebra with the priority operator. *Comput. J.*, 30(6):498–506, 1987. doi:10.1093/comjnl/30.6.498.

[BBMV91] Jos C. M. Baeten, Jan A. Bergstra, Sjouke Mauw, and Gert J. Veltink. A process specification formalism based on static COLD. In Jan A. Bergstra and Loe M. G. Feijs, editors, *Algebraic Methods II: Theory, Tools and Applications*, pages 303–335. Springer, 1991. doi:10.5555/109462.109475.

[BFK09] Jens Bender, Marc Fischlin, and Dennis Kügler. Security analysis of the PACE key-agreement protocol. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio A. Ardagna, editors, *Information Security*, pages 33–48. Springer, 2009. doi:10.1007/978-3-642-04474-8_3.

[BLMvT16] Jos C. M. Baeten, Bas Luttik, Tim Muller, and Paul van Tilburg. Expressiveness modulo bisimilarity of regular expressions with parallel composition. *Mathematical Structures in Computer Science*, 26(6):933–968, 2016. doi:10.1017/S0960129514000309.

[BN07] Sébastien Briais and Uwe Nestmann. Open bisimulation, revisited. *Theoretical Computer Science*, 386(3):236–271, 2007. doi:j.tcs.2007.07.010.

[CDS20] Véronique Cortier, Stéphanie Delaune, and Vaishnavi Sundararajan. A decidable class of security protocols for both reachability and equivalence properties. Technical Report hal-02446170, Loria & Inria Grand Est; Irisa, 2020. URL `https://hal.inria.fr/hal-02446170/`.

[CGIP12] Jean-Sébastien Coron, Aline Gouget, Thomas Icart, and Pascal Paillier. Supplemental access control (PACE v2): Security analysis of PACE integrated mapping. In David Naccache, editor, *Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*, volume 6805 of *Lecture Notes in Computer Science*, pages 207–232. Springer, 2012. doi:10.1007/978-3-642-28368-0_15.

[CKR18] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. DEEPSEC: Deciding equivalence properties in security protocols theory and practice. In *2018 IEEE Symposium on Security and Privacy (S&P)*, pages 529–546, 2018. doi:10.1109/SP.2018.00033.

[CMdV06] Cas Cremers, Sjouke Mauw, and Erik P. de Vink. Injective synchronisation: An extension of the authentication hierarchy. *Theoretical Computer Science*, 367(1):139–161, 2006. doi:10.1016/j.tcs.2006.08.034.

[Cre08] Cas Cremers. The Scyther tool: Verification, falsification, and analysis of security protocols. In *International Conference on Computer Aided Verification*, pages 414–418. Springer, 2008. doi:10.1007/978-3-540-70545-1_38.

[CS10] Tom Chothia and Vitaliy Smirnov. A traceability attack against e-passports. In Radu Sion, editor, *Financial Cryptography and Data Security, 14th International Conference, FC 2010, Tenerife, Canary Islands, Spain, January 25-28, 2010, Revised Selected Papers*, volume 6052 of *Lecture Notes in Computer Science*, pages 20–34. Springer, 2010. doi:10.1007/978-3-642-14577-3_5.

[Del19] Uni researchers discover e-passport flaw. *Delano Magazine, Luxembourg*, September 2019. URL `https://delano.lu/d/detail/news/uni-researchers-discover-e-passport-flaw/207929`.

[DKR09] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009. doi:10.3233/JCS-2009-0340.

[DvGHM08] Yuxin Deng, Rob van Glabbeek, Matthew Hennessy, and Carroll Morgan. Characterising testing preorders for finite probabilistic processes. *Logical Methods in Computer Science*, 4(4), 2008. doi:10.2168/LMCS-4(4:4)2008.

[FHMS19] Ihor Filimonov, Ross Horne, Sjouke Mauw, and Zach Smith. Breaking unlinkability of the ICAO 9303 standard for e-passports using bisimilarity. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *Computer Security – ESORICS 2019*, pages 577–594. Springer, 2019. doi:10.1007/978-3-030-29959-0_28.

[Fil20] Ihor Filimonov. *Analysis of privacy attacks on ePassports and a mitigation strategy using TOTP*. master thesis, University of Luxembourg, 2020.

[HALT18] Ross Horne, Ki Yung Ahn, Shang-Wei Lin, and Alwen Tiu. Quasi-open bisimilarity with mismatch is intuitionistic. In Anuj Dawar and Erich Grädel, editors, *In Proceedings of 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, Oxford, United Kingdom, July 9-12, 2018*, pages 26–35, 2018. doi:10.1145/3209108.3209125.

[HBD16] Lucca Hirschi, David Baelde, and Stéphanie Delaune. A method for verifying privacy-type properties: the unbounded case. In *Security and Privacy (S&P), 2016 IEEE Symposium on*, pages 564–581. IEEE, 2016. doi:10.1109/SP.2016.40.

[HBD19] Lucca Hirschi, David Baelde, and Stéphanie Delaune. A method for unbounded verification of privacy-type properties. *Journal of Computer Security*, 27(3):277–342, 2019. doi:10.3233/JCS-171070.

[HL95] Matthew Hennessy and Huimin Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138(2):353–389, 1995. doi:10.1016/0304-3975(94)00172-F.

[Hor18] Ross Horne. A bisimilarity congruence for the applied π-calculus sufficiently coarse to verify privacy properties. *CoRR*, (arXiv:1811.02536), 2018. URL `https://arxiv.org/abs/1811.02536`.

[Hüt03] Hans Hüttel. Deciding framed bisimilarity. *Electronic Notes in Theoretical Computer Science*, 68(6):1–18, 2003. doi:10.1016/S1571-0661(04)80530-9.

[ISO18] Cards and security devices for personal identification — contactless proximity objects — part 3: Initialization and anticollision. Technical Report 14443-3, ISO/IEC, 2018. URL `https://www.iso.org/standard/73598.html`.

[Kri65] Saul A. Kripke. Semantical analysis of intuitionistic logic I. In J.N. Crossley and M.A.E. Dummett, editors, *Formal Systems and Recursive Functions*, volume 40 of *Studies in Logic and the Foundations of Mathematics*, pages 92–130. Elsevier, 1965. doi:https://doi.org/10.1016/S0049-237X(08)71685-9.

[Lab19a] Thierry Labro. Une faille dans les passeports électroniques. *Paperjam, Luxembourg*, September 2019. URL `https://paperjam.lu/article/faille-dans-passeports-electro`.

[Lab19b] Thierry Labro. Une faille qui devrait alerter les autorités. *Paperjam, Luxembourg*, September 2019. URL `https://paperjam.lu/article/faille-qui-devrait-alerter-aut`.

[LL12] Jia Liu and Huimin Lin. A complete symbolic bisimulation for full applied pi calculus. *Theoretical Computer Science*, 458:76–112, 2012. doi:https://doi.org/10.1016/j.tcs.2012.07.034.

[Low97] Gavin Lowe. A hierarchy of authentication specifications. In *Proceedings 10th Computer Security Foundations Workshop*, pages 31–43, June 1997. doi:10.1109/CSFW.1997.596782.

[MDBdV12] Jasen Markovski, Pedro R. D'Argenio, Jos C. M. Baeten, and Eric P. de Vink. Reconciling real and stochastic time: the need for probabilistic refinement. *Formal Aspects of Computing*, 24(4):497–518, 2012. doi:10.1007/s00165-012-0230-y.

[MPW93] Robin Milner, Joachim Parrow, and David Walker. Modal logics for mobile processes. *Theor. Comput. Sci.*, 114(1):149–171, 1993. doi:10.1016/0304-3975(93)90156-N.

[MRT15] Machine readable travel documents. part 11: Security mechanisms for MRTDs. Technical Report Doc 9303. Seventh Edition, International Civil Aviation Organization (ICAO), 2015. URL https://www.icao.int/publications/Documents/9303_p11_cons_en.pdf.

[MvOV96] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. doi:10.1201/9781439821916.

[TD10] Alwen Tiu and Jeremy Dawson. Automating open bisimulation checking for the spi calculus. In *2010 23rd IEEE Computer Security Foundations Symposium*, pages 307–321. IEEE, 2010. doi:10.1109/CSF.2010.28.

[TGD10] Alwen Tiu, Rajeev Gore, and Jeremy Dawson. A Proof Theoretic Analysis of Intruder Theories. *Logical Methods in Computer Science*, Volume 6, Issue 3, 2010. doi:10.2168/LMCS-6(3:12)2010.

[Tiu07] Alwen Tiu. A trace based bisimulation for the spi calculus: An extended abstract. In *Programming Languages and Systems. APLAS 2007*, volume 4807 of *Lecture Notes in Computer Science*, pages 367–382. Springer, 2007. doi:10.1007/978-3-540-76637-7_25.

[TNH16] Alwen Tiu, Nam Nguyen, and Ross Horne. SPEC: An equivalence checker for security protocols. In Atsushi Igarashi, editor, *Programming Languages and Systems. APLAS 2016*, pages 87–95. Springer, 2016. doi:10.1007/978-3-319-47958-3_5.

[vDMR08] Ton van Deursen, Sjouke Mauw, and Sasa Radomirovic. Untraceability of RFID protocols. In Jose Antonio Onieva, Damien Sauveron, Serge Chaumette, Dieter Gollmann, and Constantinos Markantonakis, editors, *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks, Second IFIP WG 11.2 International Workshop, WISTP 2008, Seville, Spain, May 13-16, 2008. Proceedings*, volume 5019 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2008. doi:10.1007/978-3-540-79966-5_1.

[vG01] Rob van Glabbeek. The linear time – branching time spectrum I. In Jan A. Bergstra, Alban Ponse, and Scott A. Smolka, editors, *Handbook of Process Algebra*, pages 3 – 99. Elsevier Science, Amsterdam, 2001. doi:https://doi.org/10.1016/B978-044482830-9/50019-9.

[vG21] Rob van Glabbeek. Failure trace semantics for a process algebra with time-outs. *Logical Methods in Computer Science*, 17(2):11:1–11:40, 2021. doi:10.23638/LMCS-17(2:11)2021.

Appendix A. Reducing weak to strong bisimilarity

We provide here a proof for Lemma 2.6, which is used to prove that unlinkability when expressed in terms of a strong bisimilarity problem, is equivalent to a formulation of unlinkability in terms of weak bisimilarity (Theorem 2.10). In the proof of the lemma below we employ equivariance, which simply allows names to be swapped.

Definition A.1. *Equivariance is the least congruence extending α-conversion such that $\nu x.\nu y.P \equiv \nu y.\nu x.P$, thereby allowing the order of name binders to be ignored.*

Working up to equivariance has been shown to significantly reduce the search space when constructing a bisimulation [TNH16].

Lemma A.2 (Lemma 2.6)**.** *For any P and Q such that c_k is fresh for P and Q, we have*

$$\nu c_k.\left(!c_k(\vec{k}).P \mid !\nu\vec{k}.\overline{c_k}\left\langle\vec{k}\right\rangle.Q\right) \approx !\nu\vec{k}.(P \mid Q)$$

Proof. Define $\mathcal{R}$ to be the least symmetric relation, upto equivariance, such that for any R_i and S_i such that c_k is fresh for R_i and S_i and $\vec{r}$ is fresh for P and Q, we have that the following extended process

$$A \triangleq \; \nu c_k, \vec{k_1}, \vec{k_2}, \ldots \vec{k_n}, \vec{r}.\Big(\, \sigma \mid R_1 \mid \ldots R_n \mid !c_k(\vec{k}).P \mid S_1 \mid \ldots S_n \mid !\nu\vec{k}.\overline{c_k}\left\langle\vec{k}\right\rangle.Q \,\Big)$$

is related by $\mathcal{R}$ to the following extended process

$$B \triangleq \; \nu\vec{k}_{f(1)}, \vec{k}_{f(2)}, \ldots \vec{k}_{f(m)}, \vec{r}.\Big(\, \sigma \mid R_{f(1)} \mid S_{f(1)} \mid \ldots \mid R_{f(m)} \mid S_{f(m)} \mid !\nu\vec{k}.\,(P \mid Q) \,\Big)$$

where $f\colon \{1..m\} \to \{1..n\}$ is injective and $R_i = P\left\{{}^{\vec{k_i}}/_{\vec{k}}\right\}$ and $S_i = Q\left\{{}^{\vec{k_i}}/_{\vec{k}}\right\}$ for $i \in \{1..n\} \setminus f(\{1..m\})$.

There are two cases to pay attention to concerning extra τ-transitions in B. Firstly, consider

$$A \xrightarrow{\tau} \nu c_k, \vec{k_1}, \vec{k_2}, \ldots \vec{k_n}, \vec{k_{n+1}}, \vec{r}.\big(\, \sigma \mid R_1 \mid \ldots R_n \mid P\left\{{}^{\vec{k_{n+1}}}/_{\vec{k}}\right\} \mid !c_k(\vec{k}).P \mid S_1 \mid \ldots S_n \mid !\nu\vec{k}.\overline{c_k}\left\langle\vec{k}\right\rangle.Q\,\big)$$

This can be matched by B by performing zero transitions, whilst staying in the relation $\mathcal{R}$.

The second important case to consider is when for some $j \in \{1..n\} \setminus f(\{1..m\})$ we have $P\left\{{}^{\vec{k_j}}/_{\vec{k}}\right\}$ or $Q\left\{{}^{\vec{k_j}}/_{\vec{k}}\right\}$ acts (or indeed they interact), possibly extruding some active substitution σ and fresh names $\vec{s}$, as follows.

$$\begin{aligned}
&\nu c_k, \vec{k_1}, \vec{k_2}, \ldots \vec{k_n}, \vec{r}.\big(\, \sigma \mid R_1 \mid \ldots P\left\{{}^{\vec{k_j}}/_{\vec{k}}\right\} \ldots \mid R_n \mid !c_k(\vec{k}).P \mid S_1 \mid \ldots Q\left\{{}^{\vec{k_j}}/_{\vec{k}}\right\} \ldots \mid S_n \mid !\nu\vec{k}.\overline{c_k}\left\langle\vec{k}\right\rangle.Q\,\big) \\
&\xrightarrow{\pi} \nu c_k, \vec{k_1}, \vec{k_2}, \ldots \vec{k_n}, \vec{r}, \vec{s}.\big(\, \sigma \mid \theta \mid R'_1 \mid \ldots R'_j \ldots \mid R'_n \mid !c_k(\vec{k}).P \mid S'_1 \mid \ldots Q'_j \ldots \mid S'_n \mid !\nu\vec{k}.\overline{c_k}\left\langle\vec{k}\right\rangle.Q\,\big)
\end{aligned}$$

In this case, B stays within relation $\mathcal{R}$ by using transition

$$\begin{aligned}
B \xrightarrow{\pi} \nu\vec{k}_{f(1)}, \vec{k}_{f(2)}, \ldots \vec{k}_{f(m)}, \vec{k}_{f(m+1)}, \vec{r}, \vec{s}.\big(\, \sigma \mid \theta \mid \\
R'_{g(1)} \mid S'_{g(1)} \mid \ldots \mid R'_{g(m)} \mid S'_{g(m)} \mid R'_{g(m+1)} \mid S'_{g(m+1)} \mid !\nu\vec{k}.\,(P \mid Q)\,\big)
\end{aligned}$$

where $g\colon \{1..m+1\} \to \{1..n\}$ such that $g(i) = \begin{cases} j & \text{if } i = m+1 \\ f(i) & \text{otherwise} \end{cases}$, which is clearly injective. Note we should also consider when two distinct $j, j' \in \{1..n\} \setminus f(\{1..m\})$ interact in A, which has a similar pattern, except we require pairs of processes to be added to B using the rule Rep-close. □

Appendix B. Classical $\mathcal{FM}$ characterises strong early bisimilarity

In this paper, we prove that unlinkability properties are violated by exhibiting a distinguishing formula in classical $\mathcal{FM}$. A distinguishing formula is sufficient evidence to show that two processes specifying the unlinkability property are not bisimilar, as long as the modal logic characterises bisimilarity. Therefore the proof of soundness and completeness of strong early bisimilarity with respect to classical $\mathcal{FM}$ is critical for this work. Indeed, other parts of our reasoning may be incomplete, e.g., using open bisimilarity to seek a distinguishing strategy, but if our method discovers an attack that can be described using an $\mathcal{FM}$ formula that we confirm is distinguishing, then we are certain that unlinkability does not hold as formulated.

We reiterate Theorem 3.8. The proof is standard for a classical Milner-Parrow-Walker logic, for which reason it appears in this appendix. In fact, the use of static equivalence simplifies the analysis compared to the π-calculus, since there are no special cases for bound actions.

Theorem B.1 (Theorem 3.8). *$P \sim Q$, whenever, for all ϕ, we have $P \models \phi$ if and only if $Q \models \phi$.*

Proof. Let $\mathcal{R} = \{(A, B) \colon \forall\phi, A \models \phi \text{ iff } B \models \phi\}$. We aim to prove $\mathcal{R}$ is a strong early bisimulation. Symmetry is immediate. In the following cases assume $A \mathrel{\mathcal{R}} B$.

Case of static equivalence. By definition of $\mathcal{R}$ for any M and N, we can apply α-conversion to A and B such that $A = \nu\vec{x}.(\theta \mid P)$ and $B = \nu\vec{y}.(\sigma \mid Q)$ and $(\vec{x} \cup \vec{y}) \cap (\mathrm{fv}(M) \cup \mathrm{fv}(N)) = \emptyset$. If $M\theta = N\theta$, then by definition of satisfaction, $A \models M = N$ hence, by definition of $\mathcal{R}$, $B \models M = N$, hence by definition of satisfaction, $M\sigma = N\sigma$. Therefore A and B are statically equivalent.

Case of actions. Suppose $A \xrightarrow{\pi} A'$. Hence $A \models \langle\pi\rangle\mathtt{tt}$, so by definition of $\mathcal{R}$, we have $B \models \langle\pi\rangle\mathtt{tt}$ and hence for some B' we have $B \xrightarrow{\pi} B'$. By image-finiteness, there are finitely many B_i such that $B \xrightarrow{\pi} B_i$. Suppose for contradiction that $A' \mathrel{\mathcal{R}} B_i$ does not hold for all i. Then for all i, there exists

ϕ_i such that $A' \models \phi_i$ and $B_i \not\models \phi_i$. Hence $A \models \langle\pi\rangle(\bigwedge_i \phi_i)$ but $B \not\models \langle\pi\rangle(\bigwedge_i \phi_i)$, contradicting the assumption that $A \mathrel{\mathcal{R}} B$. Hence for some i, $A' \mathrel{\mathcal{R}} B_i$, as required.

Thus, $\mathcal{R}$ is a strong early bisimulation. Hence, if for any processes P and Q it holds that, for all formula ϕ, we have $P \models \phi$ if and only if $Q \models \phi$, then we have $P \mathrel{\mathcal{R}} Q$, and hence $P \sim Q$.

The converse direction follows by induction on the structure of ϕ. Assume $P \sim Q$, hence there is some strong early bisimulation $\mathcal{S}$ such that $P \mathrel{\mathcal{S}} Q$. In the following, assume that $A \mathrel{\mathcal{S}} B$ holds.

Case of equality. Consider when $A \models M = N$. By α-conversion, $A = \nu\vec{x}.(\theta \mid P)$ such that $\vec{x} \cap (\mathrm{fv}(M) \cup \mathrm{fv}(N)) = \emptyset$. hence $M\theta = N\theta$. Now, by α-conversion we have $B = \nu\vec{y}.(\sigma \mid Q)$ such that $\vec{y} \cap (\mathrm{fv}(M) \cup \mathrm{fv}(N)) = \emptyset$. So, by static equivalence, $M\sigma = N\sigma$; and hence $A \models M = N$, as required.

Case of conjunction. Consider when $A \models \phi \wedge \psi$ hence $A \models \phi$ and $A \models \psi$. So, by the induction hypothesis $B \models \phi$ and $B \models \psi$ and hence $B \models \phi \wedge \psi$.

Case of negation. Consider when $A \models \neg\phi$, hence $A \not\models \phi$. Hence, by the induction hypothesis, $B \not\models \phi$ hence $B \models \neg\phi$.

Case of action. Consider when $A \models \langle\pi\rangle\phi$. Hence $A \xrightarrow{\pi} A'$ such that $A' \models \phi$. Since $\mathcal{S}$ is a strong early bisimulation, there exists B' such that $B \xrightarrow{\pi} B'$ and $A' \mathrel{\mathcal{S}} B'$. Hence, by the induction hypothesis, $B' \models \phi$. Hence $B \models \langle\pi\rangle\phi$.

Hence, by induction on the structure of ϕ, for all formulae ϕ, and for all A, B such that $A \mathrel{\mathcal{S}} B$, we have $A \models \phi$ iff $B \models \phi$; and hence $P \models \phi$ iff $Q \models \phi$, since $P \mathrel{\mathcal{S}} Q$. □