Compositional Analysis of Protocol Equivalence in the Applied π-Calculus Using Quasi-open Bisimilarity

Ross Horne, Sjouke Mauw, and Semen Yurkov(✉)

Department of Computer Science, University of Luxembourg,
Esch-sur-Alzette, Luxembourg
semen.yurkov@uni.lu

Abstract. This paper shows that quasi-open bisimilarity is the *coarsest bisimilarity congruence* for the applied π-calculus. Furthermore, we show that this equivalence is suited to security and privacy problems expressed as an equivalence problem in the following senses: (1) being a bisimilarity is a safe choice since it does not miss attacks based on rich strategies; (2) being a congruence it enables a compositional approach to proving certain equivalence problems such as unlinkability; and (3) being the coarsest such bisimilarity congruence it can establish proofs of some privacy properties where finer equivalences fail to do so.

Keywords: Cryptographic calculi · Bisimilarity · Security · Privacy · Compositionality

1 Introduction

The applied π-calculus [2,5] is a generalisation and extension of the π-calculus [37] useful for verifying security and privacy properties of cryptographic protocols. Some security and privacy properties may be expressed as an equivalence problem, for instance by comparing the actual protocol to an idealised specification that trivially satisfies the desired property [8,23,24,28,33]. This paper employs good-practice principles for designing process equivalences for cryptographic calculi. We define two equivalences: one based on testing and another based on labelled transitions. The two equivalences are proven to coincide; thereby establishing that observables represented by the labels on transitions capture all relevant information about all testing contexts. This paradigm is suited to cryptographic calculi, where the testing environment contains attackers that can be inserted into a network without direct access to key material and other secrets, yet may violate security and privacy requirements of a protocol. By using an equivalence based on labelled transitions for cryptographic calculi we learn that we do not need to know all details of such malicious agents, and that to characterise such attackers it is sufficient to look only at the input and output actions of honest agents

S. Yurkov is supported by the Luxembourg National Research Fund through grant PRIDE15/10621687/SPsquared.

A. Cerone and P. C. Ölveczky (Eds.): ICTAC 2021, LNCS 12819, pp. 235–255, 2021.
https://doi.org/10.1007/978-3-030-85315-0_14

modelled in the specification of the protocol. Considering only the inputs and outputs of honest agents makes the formulation of process equivalence problems in terms of labelled transitions easier to check, compared to checking all malicious agents in full.

Amongst the most powerful testing equivalences proposed over the years is *open barbed bisimilarity* [41], and its equivalent mild simplification *saturated bisimilarity* [13], which were inspired by *dynamic bisimilarity* [38]. These testing equivalences consider all contexts at every execution step, hence, by definition, we obtain a congruence – an equivalence relation preserved in all contexts. Considering all contexts at every execution step reflects that new knowledge about the environment may be discovered during execution. Such testing equivalences have been used to inform the design of labelled transition systems and their corresponding notions of labelled equivalence for a range of process calculi including the π-calculus, which led to the emergence of *quasi-open bisimilarity* [29,41] – the notion of labelled bisimilarity that coincides with open barbed bisimilarity for the π-calculus.

In this work, we make use of the testing regime offered by open barbed bisimilarity to design a labelled transition system and notion of quasi-open bisimilarity for the applied π-calculus. We argue here that employing a bisimilarity that coincides with the testing regime offered by open barbed bisimilarity, is a win-win choice for the applied π-calculus: not only is such an equivalence well-designed according to good-practice process-design principles; but also it is useful for verifying security and privacy properties. We should always be inclined to select an equivalence that is a congruence over one that is not a congruence without a compelling reason not to; and, in the setting of the applied π-calculus, having a congruence relation opens up new proof techniques, such as the ability to reason about equivalence problems compositionally.

It is possible to design other congruence relations for the applied π-calculus, such as the more famous open bisimilarity [28,40]. What we found to be fascinating about quasi-open bisimilarity is that, although, in order to be both a congruence relation and a bisimilarity relation, it is necessary that we obtain a finer equivalence compared to the more common early bisimilarity [5] that is not a congruence, the notion of equivalence is not too fine for security and privacy problems. Moving from early bisimilarity to quasi-open bisimilarity for security and privacy problems that can be formulated as equivalence problems, such as unlinkability, strong secrecy (non-interference), voter privacy, anonymity, does not appear to invalidate established properties. While it is impossible to check and anticipate all possible security and privacy problems that can be formulated as an equivalence problem in the applied π-calculus, there is the following compelling reason why we are confident in making this claim. Despite being finer than early bisimilarity, quasi-open bisimilarity still treats classically the important content of security and privacy problems, which is the treatment of private information such as nonces and keys. In contrast, this is not the case for the finer open bisimilarity, since we, in this paper, will demonstrate a representative example of a scenario in which open bisimilarity discovers a spurious attack, whereas quasi-open bisimilarity discovers the expected proof. Indeed, we are yet to encounter a disadvantage of using open barbed bisimilarity rather than observational equivalence for privacy problems.

Outline of Paper. Section 2 provides motivation and explains minimal examples illustrating why quasi-open bisimilarity is an objective choice of bisimilarity congruence.

The rest of the paper develops a theory of quasi-open bisimilarity for the applied π-calculus. Section 3 introduces *open barbed bisimilarity*, which is, by definition, the greatest bisimilarity congruence (we focus on the strong variant). Section 4 introduces an open variant of labelled bisimilarity called *quasi-open bisimilarity* and proves that it coincides with open barbed bisimilarity. A technical report provides further details [27].

2 Motivating Quasi-open Bisimilarity for the Applied π-Calculus

In the paper, we target properties expressed as a process equivalence. Whether a system satisfies such properties depends not only on the system but also on the choice of equivalence relation, and this choice in fact contributes to the attacker model [28]. In this motivating section, we present the advantages of employing the coarsest bisimilarity congruence and present motivating examples to justify our proposal.

2.1 A Finer Equivalence Discovers Spurious Attacks

Below we provide an example of a property expressed as an equivalence and show that a finer relation can fail to reflect a real attack on the system. Our running example is a cut-down variant of a classic private server example [3,22]. We express the privacy property as an equivalence between the "real" and the "ideal" behaviours.

Consider a server *Server A* that responds with an encrypted message only when it receives a particular public key. Otherwise, it responds with a nonce, indistinguishable from a ciphertext. We assume an attacker knows public key $\mathtt{pk}(k)$ but does not know private key k or nonce r.

Server A can be modelled formally in the applied π-calculus as follows.

$$\begin{array}{ll} \textit{Server A}: & \nu k.\overline{s}\langle \mathtt{pk}(k)\rangle.!\,\nu a.\overline{c}\langle a\rangle.a(x).\nu r. \\ & \qquad\qquad \mathtt{if}\ x = \mathtt{pk}(k)\ \mathtt{then}\ \overline{a}\langle \mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(k))\rangle\ \mathtt{else}\ \overline{a}\langle r\rangle \end{array}$$

In *Server A*, the prefix $\nu k.\overline{s}\langle \mathtt{pk}(k)\rangle$ stands for announcing a public key. The prefix $!\nu a.\overline{c}\langle a\rangle.a(x).\nu r$ represents the start of an unbounded number of sessions on a fresh channel a where, in each session, an input is received and a nonce r is freshly generated. In each session, one of the following decisions is made, based on the input received. If an input is a public key output previously, *Server A* responds with a message-nonce pair encrypted with the public key $\overline{a}\langle \mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(k))\rangle$. Otherwise, *Server A* sends a dummy random message r indistinguishable from a random cyphertext.

Note that in this minimal formulation of the problem, we refrain from modelling the clients (possibly knowing key k). Of course, the fact that clients transmit their public keys in plaintext may introduce further privacy concerns, which we do not model in this minimal illustration.

We approach the problem of proving that the privacy of the owner of secret key k is preserved by providing a reference specification. The reference specification models how the private server should ideally behave from the perspective of an attacker. The specification, defined as *Server B* below, differs from *Server A* in that it transmits a nonce regardless of the message received.

$$\textit{Server B}: \quad \nu k.\overline{s}\langle \mathtt{pk}(k)\rangle.!\nu a.\overline{c}\langle a\rangle.a(x).\nu r.\overline{a}\langle r\rangle$$

Server A and *Server B* are indistinguishable to an external observer – the attacker. An attacker cannot learn that *Server A* responds in a special way to input $\texttt{pk}(k)$. The idea is that an attacker without private key k cannot learn that *Server A* serves some data m to the owner of k. Thus the privacy of the intended recipient of the data is preserved.

We can verify this privacy property by showing *Server A* and *Server B* are bisimilar. The point is that there is a warning: we must take care about which bisimilarity we employ. If we employ the famous *open bisimilarity* (which also is a congruence), the processes are **not** equivalent.

Using a suitable labelled transition system [28], *Server A* can reach the following state, at which point open bisimilarity still allows x (a free variable representing an input) to be instantiated with the message bound to u (i.e., $\texttt{pk}(k)$), representing a previously output message.

$$\begin{aligned}\nu k, a_1, r_1.\Big(\left\{ {}^{\texttt{pk}(k), a_1}/_{u,v} \right\} \mid\ & \texttt{if}\, x = \texttt{pk}(k)\ \texttt{then}\, \overline{a_1}\langle \texttt{aenc}(\langle m, r_1 \rangle, \texttt{pk}(k))\rangle\, \texttt{else}\, \overline{a_1}\langle r_1 \rangle \\ \mid\ & !\nu a.\overline{c}\langle a \rangle.a(x).\nu r. \\ & \ \texttt{if}\, x = \texttt{pk}(k)\ \texttt{then}\, \overline{a}\langle \texttt{aenc}(\langle m, r \rangle, \texttt{pk}(k))\rangle\, \texttt{else}\, \overline{a}\langle r \rangle\ \Big)\end{aligned}$$

Thus, we have not yet committed to $x = \texttt{pk}(k)$ or $x \neq \texttt{pk}(k)$, and hence we cannot proceed until we provide more information about x. Therefore the guard in the *if-then-else* statement above **cannot yet** be resolved; but *Server B* cannot reach an equivalent state, since it can only reach a state which is **immediately ready** to perform an action regardless of whether $x = \texttt{pk}(k)$ or $x \neq \texttt{pk}(k)$. Note we do not assume $x = \texttt{pk}(k) \vee x \neq \texttt{pk}(k)$ holds, which would be an instance of the law of excluded middle; hence we are in an intuitionistic setting [6,7]. The presented distinguishing strategy, does not correspond to a real attack on the privacy of *Server A*; hence open bisimilarity **is not sufficiently coarse** to verify this privacy property.

Fortunately, we will see in this paper that *quasi-open bisimilarity* addresses the above limitation of open bisimilarity. Quasi-open bisimilarity is also intuitionistic [29]. It handles open terms (with free variables) intuitionistically; but handles private messages that an attacker cannot interfere with more subtly. Private information, such as $\texttt{pk}(k)$ (bound to u in the above state), can only be received eagerly by an input action; the effect being that messages such as $\texttt{pk}(k)$ in the above example are treated classically. Thereby, after receiving the input either $x = \texttt{pk}(k)$ or $x \neq \texttt{pk}(k)$ holds; from which we establish that *Server A* and *Server B* are indeed *quasi-open bisimilar*.

The example above, elaborated on in the body of the paper, is selected as a minimal explanation for why quasi-open bisimilarity defines an appropriate attacker model.

A Still More Sophisticated Argument. Those that are not yet satisfied with the above illustration, may question whether the limitation of open bisimilarity is due to a particular lifting of open bisimilarity to the applied π-calculus. This is not the case. There are several possible definitions of open bisimilarity for the applied π-calculus that are, firstly, conservative with respect to the original definition of open bisimilarity for the π-calculus [40] and, secondly, also a congruence relation. However, none of them would be able to prove the privacy property illustrated above. The problem lies with mismatch (the else branches in the above example), which is exactly the problem isolated and explored in related work [29].

We illustrate the argument, by drawing attention to two possible ways of conservatively extending open bisimilarity to the applied π-calculus (which features mismatch or else branches). One approach is to extend the histories in the past (see Sect. 5 [29]); while another is to add explicit inequality constraints [28]. Each of these approaches provide different expressive power, as illustrated by the following pair of processes.

$$\mathit{Server\,C} \triangleq \nu k.\nu l.\overline{s}\langle \mathtt{pk}(k)\rangle.\overline{s}\langle \mathtt{pk}(l)\rangle.P(k) \;\text{ v.s. }\; \mathit{Server\,D} \triangleq \nu k.\nu l.\overline{s}\langle \mathtt{pk}(k)\rangle.\overline{s}\langle \mathtt{pk}(l)\rangle.P(l)$$

$$\text{where } P(t) \triangleq \nu a.\overline{c}\langle a\rangle.a(x).\nu r.\mathtt{if}\, x = \mathtt{pk}(t) \;\mathtt{then}\, \overline{a}\langle \mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(t))\rangle \;\mathtt{else}\, \overline{a}\langle r\rangle$$

When we specify that *Server C* and *Server D* should be equivalent, we specify that two servers that respond to different keys (the first or second output) should be indistinguishable. This differs from our previous private server example, where, recall, the specification is stipulated in terms of another private server, *Server B*, that has no information to leak. Surprisingly, the above processes are equivalent under the notion of open bisimilarity obtained by extending histories [29], yet are not equivalent under the notion of open bisimilarity with inequality constraints [28]. Furthermore, processes *Server A* and *Server B* from the previous sections are not equivalent under either of the notions of open bisimilarity described, and hence neither extension of open bisimilarity is sufficiently coarse to verify that privacy property.

The fundamental insight is that open bisimilarity is heavily tied to the way it symbolically represents constraints, which gives rise to equivalences that differ for technical reasons which have little to do with the semantics of protocols. Quasi-open bisimilarity however is canonical, as we prove in this work via a completeness result that is independent of any internal constraint system. Finally, quasi-open bisimilarity supports proofs of privacy properties that we expect to hold, as illustrated by the equivalence of *Server A* and *Server B* (Sect. 4.2), making it a robust choice that enables compositional reasoning without introducing spurious attacks.

2.2 Too Coarse an Equivalence Misses Real Attacks

Above we have seen an example when a finer property leads to a spurious attack. The situation is mirrored, however, if we attempt to formulate a property using too coarse equivalence: real attacks may be missed. Recent work [26,28], comprehensively explains an attack on ePassports that allows unauthorised observers to track movements of the holder. This attack was overlooked by trace equivalence, which is strictly coarser than bisimilarity. Thus, taking into consideration both ends of the spectrum, we find that quasi-open bisimilarity is neither too coarse, since it is a bisimilarity, nor too fine, since it does not introduce spurious attacks.

2.3 A Congruence Enables Compositional Reasoning

To illustrate the importance of the presented equivalence being a congruence we briefly introduce a discussion on unlinkability, that is the incapability of determining relationships between two observed protocol sessions. The state-of-the-art approach to unlinkability developed in [28] is as follows. If the process *System*, reflecting the actual behaviour of the protocol, is equivalent to the process *Spec*, which specifies the ideal behaviour (from the attacker's perspective), we say that such a protocol is unlinkable.

Consider an abstract authentication protocol with two roles: C and T. The agent playing role C holds credentials signed by the secret key s of the certification authority CA and wants to be able to assume the same identity multiple times without the risk of being reidentified. The goal of the agent playing T is to verify these credentials using the public key $\mathtt{pk}(s)$ of the CA and authenticate C. The real-world behaviour of the system can be modelled as follows.

$$\mathit{System} \triangleq \nu s.\Big(\, !\nu a.!\nu ch_c.\overline{c_C}\langle ch_c\rangle.C(s, ch_c, a) \mid \overline{out}\langle \mathtt{pk}(s)\rangle.!\nu ch_t.\overline{c_T}\langle ch_t\rangle.T(\mathtt{pk}(s)\,, ch_t)\,\Big)$$

Initially, the CA's secret key s is created. The first parallel component above defines agents with identity a that can participate in an **arbitrary number of sessions** of the protocol. Each session begins with advertising a fresh session channel ch_c on the public channel c_C, modelling a new connection to a new session. The leftmost replication models that any number of agents can exist in the system, while the subsequent replication is what allow an agent to appear with the same identity across multiple sessions. The second parallel component above makes the public key $\mathtt{pk}(s)$ of the CA available to the environment via the output on the public channel out. After that, the role T is specified which tries to authenticate a genuine agent in role C making use of $\mathtt{pk}(s)$. Such sessions in role T also begin by advertising a fresh session channel on public channel c_T. The processes $C(s, ch_t)$ and $T(\mathtt{pk}(s)\,, ch_t)$ can be instantiated to model various protocols.

On the other hand, the ideal system is obtained from *System* by removing the second replication, which means that the agent with the identity a can participate in **one protocol run** only.

$$\mathit{Spec} \triangleq \nu s.\Big(\, !\nu a.\nu ch_c.\overline{c_C}\langle ch_c\rangle.C(s, ch_c, a) \mid \overline{out}\langle \mathtt{pk}(s)\rangle.!\nu ch_t.\overline{c_T}\langle ch_t\rangle.T(\mathtt{pk}(s)\,, ch_t)\,\Big)$$

The definition of unlinkability is as follows.

Definition 1 *(unlinkability)***.** *The system satisfies unlinkability if System $\approx$ Spec holds, where $\approx$ is weak early bisimilarity.*

The fact that quasi-open bisimilarity is a congruence allows us to verify an equivalence property for a smaller system and extend the proof to a larger system. Consider a smaller system comprising only agents playing the role C.

$$\mathit{Small\_System} \triangleq \nu s.\overline{out}\langle \mathtt{pk}(s)\rangle.!\nu a.!\nu ch_c.\overline{c_C}\langle ch_c\rangle.C(s, ch_c, a)$$

The corresponding, smaller version of the idealised specification where there is one session per identity is as follows.

$$\mathit{Small\_Spec} \triangleq \nu s.\overline{out}\langle \mathtt{pk}(s)\rangle.!\nu a.\nu ch_c.\overline{c_C}\langle ch_c\rangle.C(s, ch_c, a)$$

We are ready now to prove that if we prove properties using the smaller specification with one role then they hold in the more traditional specification with two roles.

Theorem 1. *If $Small\_System \sim Small\_Spec$, where $\sim$ is quasi-open bisimilarity, then System $\approx$ Spec.*

Proof. Consider the following context, where *out*′ is a fresh variable.

$$\mathcal{C}\{\,\cdot\,\} \triangleq \nu out.\Big(\{\,\cdot\,\} \mid out(pks).\overline{out'}\langle pks\rangle.!\nu ch_t.\overline{c_T}\langle ch_t\rangle.T(\mathtt{pk}(s)\,, ch_t)\Big)$$

Firstly, $\mathcal{C}\{Small\_System\}\{^{out}/_{out'}\} \sim \tau.System$ and $\mathcal{C}\{Small\_Spec\}\{^{out}/_{out'}\} \sim \tau.Spec$ hold. Furthermore, by the assumption, $Small\_System \sim Small\_Spec$ and the fact that since quasi-open bisimilarity is a congruence (Theorem 2), the following holds.

$$\mathcal{C}\{Small\_System\} \sim \mathcal{C}\{Small\_Spec\}$$

Furthermore, since quasi-open bisimilarity is closed under substitutions involving free variables (by definition) we have that the following holds.

$$\mathcal{C}\{Small\_System\}\{^{out}/_{out'}\} \sim \mathcal{C}\{Small\_Spec\}\{^{out}/_{out'}\}$$

Hence, since quasi-open bisimilarity is an equivalence relation, we have the following.

$$\tau.Spec \sim \tau.System$$

Thus there exists a quasi-open bisimulation $\mathcal{R}$ such that $\tau.Spec\ \mathcal{R}\ \tau.System$. Hence, since $\tau.Spec \xrightarrow{\tau} Spec$ it must be the case that $\tau.System \xrightarrow{\tau} System$ and $Spec\ \mathcal{R}\ System$. Therefore $Spec \sim System$. Finally, since $\sim \subseteq \approx$ we have $Spec \approx System$. □

The key difficulty is, of course, to prove that $Small\_System \sim Small\_Spec$, but studying a smaller system significantly reduces the amount of work. This approach to verifying unlinkability for a subsystem was taken in [30], where authors study key agreement for contactless payments and employ only honest cards in their model of unlinkability.

3 The Coarsest Bisimilarity Congruence

This section concerns the coarsest (strong) bisimilarity congruence – open barbed bisimilarity. Open barbed bisimilarity is a natural choice of bisimilarity, being, by definition, the greatest bisimilarity congruence. Since open barbed bisimilarity has an objective language-independent definition, there are no design decisions – there is only one reasonable definition as explored in this section.

3.1 An Example Message Term Language and Equational Theory

In the applied π-calculus messages can be defined with respect to any message language subject to any equational theory ($=_E$). The example equational theory we provide in Fig. 1 is for the purpose of providing meaningful examples. Further theories can

$$
\begin{array}{lllr}
M, N, K & ::= & x & \text{variable} \\
 & \mid & \mathsf{pk}(M) & \text{public key} \\
 & \mid & \mathsf{h}(M) & \text{hash} \\
 & \mid & \langle M, N \rangle & \text{tuple} \\
 & \mid & \mathtt{aenc}(M, N) & \text{asymmetric encryption} \\
 & \mid & \mathtt{adec}(M, N) & \text{asymmetric decryption} \\
 & \mid & \mathtt{fst}(M) & \text{left} \\
 & \mid & \mathtt{snd}(M) & \text{right}
\end{array}
$$

$$
\begin{array}{c}
\mathtt{fst}(\langle M, N \rangle) =_E M \\
\mathtt{snd}(\langle M, N \rangle) =_E N \\
\mathtt{adec}(\mathtt{aenc}(M, \mathsf{pk}(K)), K) =_E M \\
\mathtt{aenc}(\mathtt{adec}(M, K), \mathsf{pk}(K)) =_E M
\end{array}
$$

Fig. 1. The applied π-calculus can be instantiated with **any** message language and equational theory for messages. This example message theory is provided *only* to provide meaningful examples.

also be devised not limited to: sub-term convergent theories [1]; blind signatures and homomorphic encryption [20]; and locally stable theories with inverses [9].

The example theory provided in Fig. 1 covers asymmetric encryption. A message encrypted with public key $\mathtt{pk}(k)$ can only be decrypted using private key k. The theory includes a collision-resistant hash function, with no equations. This theory assumes we have the power to detect whether a message is a pair, but cannot distinguish a failed decryption from a random number.

3.2 Active Substitutions and Labelled Transitions

We define a syntax for the applied π-calculus. The syntax is similar to the π-calculus, except messages and channels can be any term rather than just variables. There is no separate syntactic class of terms for names – names are variables bound by new name binders. In addition to processes, *extended processes* are defined, which allow *active substitutions*, denoted σ, to float alongside processes and in the scope of new name binders, defined in Fig. 2.

Extended processes in normal form $\nu \boldsymbol{x}.(\sigma \mid P)$ are subject to the restriction that the variables in $\mathrm{dom}(\sigma)$ are fresh for $\boldsymbol{x}$, $\mathrm{fv}(P)$ and $\mathrm{fv}(y\sigma)$, for all variables y (i.e., σ is idempotent, and substitutions are fully applied to P). We follow the convention that operational rules are defined directly on extended processes in normal form up to α-conversion. This avoids numerous complications caused by the structural congruence in the original definition of bisimilarity for the applied π-calculus. The set of free variables and α-conversion are as standard, where $\nu x.P$ and $M(x).P$ bind x in P.

Intuitionistic mismatch. Mismatch requires special attention. Mismatch models the `else` branch of an `if-then-else` statement with an equality guard. Hence we can encode a conditional branching statement $\mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q$ using process $[M = N]P + [M \neq N]Q$.

As uncovered in related work [29], the trick for handling mismatch such that we obtain a congruence is to treat mismatch intuitionistically. Intuitionistic negation enjoys the property that it is preserved under substitutions; a property that fails for classical negation in general. E.g., there are substitutions under which $[x \neq \mathtt{h}(y)]a(z)$ can perform an input transition and others where it cannot, hence neither $x = \mathtt{h}(y)$ nor $x \neq \mathtt{h}(y)$ holds in the intuitionistic setting until more information is provided about the

SYNTAX OF EXTENDED PROCESSES AND LABELS

$$\begin{array}{lll} P, Q ::= & 0 & \text{deadlock} \\ & \mid \overline{M}\langle N\rangle.P & \text{send} \\ & \mid M(y).P & \text{receive} \\ & \mid [M = N]P & \text{match} \\ & \mid [M \neq N]P & \text{mismatch} \\ & \mid \nu x.P & \text{new} \\ & \mid P \mid Q & \text{parallel} \\ & \mid P + Q & \text{choice} \\ & \mid\, !P & \text{replication} \end{array}$$

Extended processes:

$$\begin{array}{lll} A, B ::= & \sigma \mid P & \text{active substitution and process} \\ & \mid \nu x.A & \text{new} \end{array}$$

actions on labels:

$$\begin{array}{lll} \pi ::= & \tau & \text{internal action} \\ & \mid \overline{M}(z) & \text{bound output} \\ & \mid M\,N & \text{free input} \end{array}$$

LABELLED TRANSITION SYSTEM

$$\frac{M\sigma =_E K}{z\colon \sigma \mid K(x).P \xrightarrow{M\,N} \sigma \mid P\left\{{}^{N\sigma}/_{x}\right\}}\;\textsc{Inp} \qquad \frac{x \# M, N, P, \sigma, z \qquad M\sigma =_E K}{z\colon \sigma \mid \overline{K}\langle N\rangle.P \xrightarrow{\overline{M}(x)} \left\{{}^{N}/_{x}\right\} \circ \sigma \mid P}\;\textsc{Out}$$

$$\frac{z\colon \sigma \mid P \xrightarrow{\pi} A}{z\colon \sigma \mid P + Q \xrightarrow{\pi} A}\;\textsc{Sum-l} \qquad \frac{z\colon \sigma \mid Q \xrightarrow{\pi} A}{z\colon \sigma \mid P + Q \xrightarrow{\pi} A}\;\textsc{Sum-r}$$

$$\frac{z\colon \sigma \mid P \xrightarrow{\pi} A \qquad M =_E N}{z\colon \sigma \mid [M = N]P \xrightarrow{\pi} A}\;\textsc{Mat} \qquad \frac{z\colon \sigma \mid P \xrightarrow{\pi} A \qquad z \models M \neq N}{z\colon \sigma \mid [M \neq N]P \xrightarrow{\pi} A}\;\textsc{Mismatch}$$

$$\frac{z, x\colon \sigma \mid P \xrightarrow{\pi} B \qquad x \# z, \sigma, \mathrm{n}(\pi)}{z\colon \sigma \mid \nu x.P \xrightarrow{\pi} \nu x.B}\;\textsc{Extrude} \qquad \frac{z, x\colon A \xrightarrow{\pi} B \qquad x \# z, \mathrm{n}(\pi)}{z\colon \nu x.A \xrightarrow{\pi} \nu x.B}\;\textsc{Res}$$

$$\frac{z\colon \sigma \mid P \xrightarrow{\pi} \nu\boldsymbol{x}.(\sigma \mid R) \quad \boldsymbol{x} \cup \mathrm{bn}(\pi) \# Q}{z\colon \sigma \mid P \mid Q \xrightarrow{\pi} \nu\boldsymbol{x}.(\sigma \mid R \mid Q)}\;\textsc{Par-l} \qquad \frac{z\colon \sigma \mid Q \xrightarrow{\pi} \nu\boldsymbol{x}.(\sigma \mid R) \quad \boldsymbol{x} \cup \mathrm{bn}(\pi) \# P}{z\colon \sigma \mid P \mid Q \xrightarrow{\pi} \nu\boldsymbol{x}.(\sigma \mid P \mid R)}\;\textsc{Par-r}$$

$$\mathrm{n}(\pi) = \begin{cases} \mathrm{fv}(M) \cup \{x\} & \text{if } \pi = \overline{M}(x) \\ \mathrm{fv}(M) \cup \mathrm{fv}(N) & \text{if } \pi = M\,N \\ \emptyset & \text{otherwise} \end{cases} \qquad \mathrm{bn}(\pi) = \begin{cases} \{x\} & \text{if } \pi = \overline{M}(x) \\ \emptyset & \text{otherwise} \end{cases}$$

$$\frac{z\colon \sigma \mid P \xrightarrow{\overline{M}(x)} \nu\boldsymbol{y}.\left(\left\{{}^{N}/_{x}\right\} \circ \sigma \mid P'\right) \quad z\colon \sigma \mid Q \xrightarrow{M\,N} \nu\boldsymbol{w}.(\sigma \mid Q') \quad \{x\} \cup \boldsymbol{y} \# Q \quad \boldsymbol{w} \# P, \boldsymbol{y}}{z\colon \sigma \mid P \mid Q \xrightarrow{\tau} \nu\boldsymbol{y}, \boldsymbol{w}.(\sigma \mid P' \mid Q')}\;\textsc{Close-l}$$

$$\frac{z\colon \sigma \mid P \xrightarrow{M\,N} \nu\boldsymbol{y}.(\sigma \mid P') \quad z\colon \sigma \mid Q \xrightarrow{\overline{M}(x)} \nu\boldsymbol{w}.\left(\left\{{}^{N}/_{x}\right\} \circ \sigma \mid Q'\right) \quad \{x\} \cup \boldsymbol{w} \# P \quad \boldsymbol{y} \# Q, \boldsymbol{w}}{z\colon \sigma \mid P \mid Q \xrightarrow{\tau} \nu\boldsymbol{y}, \boldsymbol{w}.(\sigma \mid P' \mid Q')}\;\textsc{Close-r}$$

$$\frac{z\colon \sigma \mid P \xrightarrow{\pi} \nu\boldsymbol{x}.(\sigma \mid Q) \quad \boldsymbol{x} \cup \mathrm{bn}(\pi) \# P}{z\colon \sigma \mid\, !P \xrightarrow{\pi} \nu\boldsymbol{x}.(\sigma \mid Q \mid\, !P)}\;\textsc{Rep-act}$$

$$\frac{z\colon \sigma \mid P \xrightarrow{\overline{M}(x)} \nu\boldsymbol{y}.\left(\left\{{}^{N}/_{x}\right\} \circ \sigma \mid Q\right) \quad z\colon \sigma \mid P \xrightarrow{M\,N} \nu\boldsymbol{w}.(\sigma \mid R) \quad \boldsymbol{y} \# P, \boldsymbol{w} \quad \boldsymbol{w} \# P}{z\colon \sigma \mid\, !P \xrightarrow{\tau} \nu\boldsymbol{y}, \boldsymbol{w}.(\sigma \mid Q \mid R \mid\, !P)}\;\textsc{Rep-close}$$

Fig. 2. Syntax of extended processes and an *open early* labelled transition system.

environment. In order to define intuitionistic negation, we require the notion of a fresh substitution.

Definition 2 (fresh). *Consider a set of variables $\boldsymbol{z}$. We say $\boldsymbol{z}$ is fresh for set of variables $\boldsymbol{y}$, whenever $\boldsymbol{z} \cap \boldsymbol{y} = \emptyset$. Given a term, say P, we say $\boldsymbol{z}$ is fresh for P, whenever $\boldsymbol{z}$ is fresh for the free variables of P. Given a substitution σ, we say $\boldsymbol{z}$ is fresh for σ whenever $\boldsymbol{z}$ is fresh for $\mathrm{dom}(\sigma)$, and, for all $y \notin \boldsymbol{z}$, we have $\boldsymbol{z}$ is fresh for $y\sigma$. Freshness extends point-wise to lists of entities and is denoted $u, v, \ldots \# M, N, \sigma, \ldots$.*

We say entailment $\boldsymbol{z} \models M \neq N$ holds whenever there is no σ such that $\boldsymbol{z} \# \sigma$ and $M\sigma =_E N\sigma$.

Consider the following examples that hold or fail to hold for different reasons. Entailment $\emptyset \models x \neq h(x)$ holds, since there exists no unifier, witnessed by a simple occurs check. In contrast, $\emptyset \models x \neq h(y)$ does not hold, since there exists substitution $\left\{{}^{h(y)}/_{x}\right\}$ unifying messages x and $h(y)$, so it is still possible the messages could be equal; thus, there is insufficient information to decide whether the messages are equal or not. By extending the environment such that y is a private name, entailment $y \models x \neq h(y)$ holds, since y is not fresh for the most general unifier $\left\{{}^{h(y)}/_{x}\right\}$ – an observer who can influence x, cannot make x equal to $h(y)$ without access to y.

To define open barbed bisimilarity, we require the labelled transition system for the applied π-calculus in Fig. 2. It is an *early* labelled transition system due to the way inputs are treated, and open early, since it does not assume that free variables are ground names, unless stated so explicitly in the name environment to the left of the transition relation. There are three types of label: τ representing internal progress due to communication; bound output $\overline{M}(x)$ representing that something bound to x is sent on channel M; and free input $M\,N$ representing that we receive N on channel M.

The Mismatch and Res Rules. The Mismatch rule is defined in terms of entailment in Definition 2. The Res rule can also influence mismatches by introducing fresh names. For example, the following derivation shows an input is enabled.

$$\dfrac{\dfrac{\dfrac{}{y\colon z(w) \xrightarrow{z\,w} 0}\;\textsc{Inp} \qquad y \models x \neq h(y)}{y\colon [x \neq h(y)]z(w) \xrightarrow{z\,w} 0}\;\textsc{Mismatch} \qquad y \# z\,w}{\emptyset\colon \nu y.[x \neq h(y)]z(w) \xrightarrow{z\,w} \nu y.0}\;\textsc{Res}$$

Notice, the bound variable y is added to the set of names, enabling $y \models x \neq h(y)$.

Active Substitutions and Labels. For a non-trivial example where the active substitution affects the label, observe that the following transition is derivable.

$$\dfrac{\dfrac{\mathtt{fst}(\langle m, n\rangle) =_E m}{m\colon \left\{{}^{\langle m,n\rangle}/_{w}\right\} \mid m(x) \xrightarrow{\mathtt{fst}(w)\,x} \left\{{}^{\langle m,n\rangle}/_{w}\right\} \mid 0}\;\textsc{Inp} \qquad m \# \mathtt{fst}(w)\,x}{\emptyset\colon \nu m.\left(\left\{{}^{\langle m,n\rangle}/_{w}\right\} \mid m(x)\right) \xrightarrow{\mathtt{fst}(w)\,x} \nu m.\left(\left\{{}^{\langle m,n\rangle}/_{w}\right\} \mid 0\right)}\;\textsc{Res}$$

The conditions on the Res rule ensure bound name m cannot appear in the terms on the label. Fortunately, the Inp rule allows m to be expressed in terms of extruded variable

w. Since we have $m =_E \texttt{fst}(\langle m, n\rangle)$ and the equational theory can be applied in rule Inp, the above input action is enabled, where message $\texttt{fst}(w)$ indirectly refers to channel m.

The Out Rule. The Out rule for the applied π-calculus does not record the message sent on the label; instead, the message is recorded in an active substitution. The domain of the active substitution is chosen to be a fresh variable appearing as the bound variable in the output action on the label.

In the following example a message is sent using the Out rule, then the Res rule is applied such that the private name n in the active substitution remains bound after the transition.

$$\frac{\dfrac{}{n, k \colon \overline{a}\langle \texttt{aenc}(n, \texttt{pk}(k))\rangle.n(x) \xrightarrow{\overline{a}(w)} \left\{ {}^{\texttt{aenc}(n,\texttt{pk}(k))}/_{w} \right\} \mid n(x)}}{k \colon \nu n.\overline{a}\langle \texttt{aenc}(n, \texttt{pk}(k))\rangle.n(x) \xrightarrow{\overline{a}(w)} \nu n.\left(\left\{ {}^{\texttt{aenc}(n,\texttt{pk}(k))}/_{w} \right\} \mid n(x) \right)}$$

Observe, by rule Inp, the following input action is enabled.

$$k \colon a(w).\overline{\texttt{adec}(w, k)}\langle a\rangle \xrightarrow{a\ \texttt{aenc}(n,\texttt{pk}(k))} \overline{\texttt{adec}(\texttt{aenc}(n, \texttt{pk}(k))\,, k)}\langle a\rangle$$

Hence, by Close-l and the above input and output transitions, the following interaction is enabled; and Res is used to bind the key k.

$$\frac{k \colon \nu n.\overline{a}\langle \texttt{aenc}(n, \texttt{pk}(k))\rangle.n(x) \mid a(w).\overline{\texttt{adec}(w, k)}\langle a\rangle \xrightarrow{\tau} \nu n.(n(x) \mid \overline{n}\langle a\rangle)}{\emptyset \colon \nu k.\left(\nu n.\overline{a}\langle \texttt{aenc}(n, \texttt{pk}(k))\rangle.n(x) \mid a(w).\overline{\texttt{adec}(w, k)}\langle a\rangle\right) \xrightarrow{\tau} \nu k.\nu n.(n(x) \mid \overline{n}\langle a\rangle)}\ \textsc{Res}$$

Note this labelled approach to interaction follows closely how interaction traditionally works in the π-calculus. Thus this formulation of labelled transitions facilitates the lifting of results from the π-calculus to the applied π-calculus. An advantage of our labelled transition system is *strong*, *weak*, and other variants of bisimilarity can be studied. In contrast, the original system proposed for the applied π-calculus [2] used a hybrid labelled/reduction system that can only be used to formalise weak equivalences. Furthermore, avoiding a structural congruence avoids having to consider all transitions up to an associative-commutative theory (which can make proofs cumbersome). Also, the use of Rep-act and Rep-close for defining replication, respects image-finiteness (up to α-conversion) [39].

Note, trying to obtain strong bisimilarity by naïvely restricting the original definition of labelled bisimilarity [2] such that every τ-transition is matched by exactly one τ-transition results in an ill-formulated notion of strong bisimilarity. Doing so, would allow processes, such as *Server A* and *Server B* from Sect. 2, to be wrongly distinguished by counting the number of τ-transitions induced by branching statements. The rule Sum-l and its counterpart Sum-r avoid this problem.

3.3 A Testing Regime Defining a Bisimilarity Congruence

A barb represents the ability to observe an input or output action on a channel. Barbs are typically used to define *barbed equivalence*, or *observational equivalence* [36]. However, barbed equivalence is a congruence but not a bisimilarity; while observational equivalence is a bisimilarity but not a congruence. For this reason, we prefer *open barbed bisimilarity* [41], which is, by definition, both a bisimilarity and a congruence. We adopt the convention of writing $A \xrightarrow{\pi} B$ whenever $\emptyset \colon A \xrightarrow{\pi} B$. We say process P has barb M, written $P \downarrow M$, whenever, for some A, $P \xrightarrow{\overline{M}(z)} A$, or $P \xrightarrow{M\,N} A$.

Definition 3 (open barbed bisimilarity). *An open barbed bisimulation $\mathcal{R}$ is a symmetric relation over processes such that whenever $P \mathrel{\mathcal{R}} Q$ holds the following hold:*

- *For all contexts $\mathcal{C}\{\,\cdot\,\}$, $\mathcal{C}\{P\} \mathrel{\mathcal{R}} \mathcal{C}\{Q\}$.*
- *If $P \downarrow M$ then $Q \downarrow M$.*
- *If $P \xrightarrow{\tau} P'$, there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $P' \mathrel{\mathcal{R}} Q'$ holds.*

Open barbed bisimilarity $\simeq$ is the greatest open barbed bisimulation. More specifically, processes P and Q are open barbed bisimilar, written $P \simeq Q$, whenever there exists an open barbed bisimulation $\mathcal{R}$ such that $P \mathrel{\mathcal{R}} Q$.

The power of open barbed bisimilarity comes from closing by all contexts at every step, not only at the beginning of an execution. Closing by all contexts at every step ensures the robustness of open barbed bisimilarity even if the environment is extended at runtime; i.e., we stay within a congruence relation at every step of the bisimulation game.

Recall that a congruence is an equivalence relation preserved in all contexts. Symmetry and context closure of open barbed bisimilarity are immediate from the definition. Reflexivity is trivial since the identity relation is an open barbed bisimulation. Transitivity is only slightly more involved, proven by checking that the composition of two open barbed bisimulation relations is an open barbed bisimulation.

Open barbed bisimilarity is concise – the definition requires only the open labelled transition system in Fig. 2 and the three clauses in Definition 3. Furthermore, it is the coarsest bisimilarity congruence, in the objective sense that it is defined to be a congruence and defined independently of the content of the messages sent and received. Notice, due to the independence of the information on the labels, open barbed bisimilarity applies to any language; indeed open barbed bisimilarity is a generalisation of dynamic observational equivalence [38], that, historically, was used to objectively identify the greatest bisimulation congruence for CCS. Related work also uses the term saturated bisimilarity for such a reference bisimilarity congruence [13,14], which shows that a single barb, say *ok* suffices.

For the above reasons, open barbed bisimilarity is an ideal reference definition. However, it is unwieldy due to closure of the definition under all contexts. This leads us to the notion of quasi-open bisimilarity, defined in the next section.

4 Quasi-open Bisimilarity for the Applied π-Calculus

As highlighted in the previous section, open barbed bisimilarity has a concise and objective definition but is difficult to check, due to the quantification over all contexts. An open variant of labelled bisimilarity, called *quasi-open bisimilarity*, avoids quantifying over all contexts; and furthermore, coincides with open barbed bisimilarity. In this section, we define quasi-open bisimilarity for the applied π-calculus, generalising established results for the π-calculus [29,41].

4.1 Introducing Quasi-open Bisimilarity for the Applied π-Calculus

To extend quasi-open bisimilarity to the applied π-calculus the notion of static equivalence is required. Static equivalence is defined over the static information in an extended process – the active substitutions and name restrictions.

Definition 4 (static equivalence). *Two extended processes $\nu \boldsymbol{x}.(\sigma \mid P)$ and $\nu \boldsymbol{y}.(\theta \mid Q)$ are statically equivalent whenever for all messages M and N such that $\boldsymbol{x}, \boldsymbol{y} \# M, N$, we have $M\sigma =_E N\sigma$ if and only if $M\theta =_E N\theta$.*

In the above definition, messages M and N represent two different "recipes" for producing messages. Two extended processes are distinguished by static equivalence only when the two recipes produce equivalent messages under one substitution, but distinct messages under the other substitution. The concept of static equivalence is no different from original work on the applied π-calculus [5].

Static Equivalence Example. For example, the following extended processes are not statically equivalent.

$$\nu k.\left(\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\,\mathtt{aenc}(x,\mathtt{pk}(k))}/_{v,\,w}\right\} \mid 0\right) \text{ is distinct from } \nu k.\left(\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\,\mathtt{aenc}(z,\mathtt{pk}(k))}/_{v,\,w}\right\} \mid 0\right)$$

The above are distinguished by messages v and w. Notice $v\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\,\mathtt{aenc}(x,\mathtt{pk}(k))}/_{v,\,w}\right\}$ and $w\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\,\mathtt{aenc}(x,\mathtt{pk}(k))}/_{v,\,w}\right\}$ are both equal to $\mathtt{aenc}(x, \mathtt{pk}(k))$; but the following messages are distinct: $v\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\,\mathtt{aenc}(z,\mathtt{pk}(k))}/_{v,\,w}\right\}$ vs. $w\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\mathtt{aenc}(z,\mathtt{pk}(k))}/_{v,w}\right\}$.

In order to define quasi-open bisimilarity, we require the notion of an *open relation* between extended processes. An open relation is preserved under substitutions (respecting bound names, including the domain of active substitution) and extensions of the active substitutions and names in environment. In the following, $\theta{\restriction_D}$ is the restriction of a substitution to a set D.

Definition 5 (open). *A relation over extended processes $\mathcal{R}$ is open whenever we have that if $\nu \boldsymbol{x}.(\theta_1 \mid P) \mathrel{\mathcal{R}} \nu \boldsymbol{y}.(\theta_2 \mid Q)$ and there exist idempotent substitutions σ, ρ and variables $\boldsymbol{z}$ such that: $\boldsymbol{x}, \boldsymbol{y} \# \sigma, \rho$ and $\boldsymbol{z} \# \operatorname{dom}(\rho), \boldsymbol{x}, \boldsymbol{y}$ and $\operatorname{dom}(\theta_i) \# \sigma, \rho, \boldsymbol{z}$ for $i \in \{1, 2\}$, we have the following:*

$$\nu \boldsymbol{z}, \boldsymbol{x}.\left((\theta_1 \circ \sigma){\restriction_{\operatorname{dom}(\theta_1)}} \circ \rho \mid P\sigma\right) \mathrel{\mathcal{R}} \nu \boldsymbol{z}, \boldsymbol{y}.\left((\theta_2 \circ \sigma){\restriction_{\operatorname{dom}(\theta_2)}} \circ \rho \mid Q\sigma\right)$$

Given the definition of an open relation, static equivalence, and the labelled transition system, we can provide the following concise definition of quasi-open bisimilarity for the applied π-calculus.

Definition 6 (quasi-open bisimilarity). *An **open** symmetric relation between extended processes $\mathcal{R}$ is a quasi-open bisimulation whenever, if $A \mathrel{\mathcal{R}} B$ then 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 quasi-open bisimilar, written $P \sim Q$, whenever $P \mathrel{\mathcal{R}} Q$ for some quasi-open bisimulation $\mathcal{R}$.

The keyword in the definition above is "open" in the sense of Definition 5. Without ensuring that properties are preserved under reachability, the above definition would simply be a strong version of the classical *labelled bisimilarity* for the applied π-calculus [5]. We illustrate the impact of insisting on an open relation and allowing messages as channels in the following examples.

Remark 1. The definition of quasi-open bisimilarity above is arguably simpler than in the setting of the π-calculus [41]. In contrast to the original definition, since private names are recorded in extended processes, all types of action are handled by one clause. The π-calculus definition maintains an additional index of extruded private names.

Mobility Example. This work builds on a recent evolution of the applied π-calculus [5], which allows processes such as $\nu z.\overline{x}\langle z, y\rangle.z(w)$ and $\nu z.\overline{x}\langle z, y\rangle$ to be evaluated. These processes should not be equivalent, since they are polyadic π-calculus processes [35] (the π-calculus with tuples), and the applied π-calculus should be conservative with respect to the polyadic π-calculus, which was not the case for older definitions of bisimilarity for the applied π-calculus [2]. The trick to allow processes such as the above to be evaluated is simple: allow channels to be messages. This way, a message, such as $\mathtt{fst}(u)$, can be used to indirectly refer to channels. To see why we can distinguish these processes, firstly, consider the following two transitions with matching actions.

$$\nu z.\overline{x}\langle z, y\rangle.z(w) \xrightarrow{\overline{x}(v)} \nu z.\left(\left\{{}^{\langle z,y\rangle}/_{v}\right\} \mid z(w)\right) \qquad \nu z.\overline{x}\langle z, y\rangle \xrightarrow{\overline{x}(v)} \nu z.\left(\left\{{}^{\langle z,y\rangle}/_{v}\right\} \mid 0\right)$$

The labelled transition $\nu z.\left(\left\{{}^{\langle z,y\rangle}/_{v}\right\} \mid z(w)\right) \xrightarrow{\mathtt{fst}(v)\, x} \nu z.\left(\left\{{}^{\langle z,y\rangle}/_{v}\right\} \mid 0\right)$ is enabled for the process on the left. The process on the right above $\nu z.\left(\left\{{}^{\langle z,y\rangle}/_{v}\right\} \mid 0\right)$ is deadlocked, so cannot match this transition. Notice the use of message $\mathtt{fst}(v)$ on the input label to refer to the private channel output at the first step.

Example Showing Impact of an Open Relation on Static Equivalence. By insisting that a quasi-open bisimulation is an open relation (Definition 5), static equivalence must also be preserved by all fresh substitutions. This has an impact on examples such as the following.

Processes $\nu x.\overline{a}\langle \mathtt{aenc}(x, z)\rangle$ and $\nu x.\overline{a}\langle \mathtt{aenc}(\langle x, y\rangle, z)\rangle$ are labelled bisimilar but not quasi-open bisimilar. To see why, observe both processes can perform a $\overline{a}(v)$-transition to the respective extended processes $\nu x.\left(\left\{{}^{\mathtt{aenc}(x,z)}/_{v}\right\} \mid 0\right)$ and $\nu x.\left(\left\{{}^{\mathtt{aenc}(\langle x,y\rangle,z)}/_{v}\right\} \mid 0\right)$. These extended process are **statically** equivalent (recall z cannot be used to decrypt these cyphertexts in asymmetric cryptography). However, since a quasi-open bisimulation must be preserved under fresh substitutions and $v \ \# \left\{{}^{\mathtt{pk}(w)}/_{z}\right\}$, we check static equivalence for $\nu x.\left(\left\{{}^{\mathtt{aenc}(x,z)}/_{v}\right\} \mid 0\right)\left\{{}^{\mathtt{pk}(w)}/_{z}\right\}$ and $\nu x.\left(\left\{{}^{\mathtt{aenc}(\langle x,y\rangle,z)}/_{v}\right\} \mid 0\right)\left\{{}^{\mathtt{pk}(w)}/_{z}\right\}$. After applying the substitution, the extended processes are no longer statically equivalent, witnessed by distinguishing recipes $\mathtt{snd}(\mathtt{adec}(v, w))$ and y. Thus the processes are not quasi-open bisimilar.

Note that the fact that the attack succeeds above suggests the attacker has the power to influence the message bound to z, in order to stage an attack. In the above example the message chosen is a public key $\mathtt{pk}(w)$ for which the attacker knows the

$$\begin{array}{rl}
\nu k.\overline{s}\langle \mathtt{pk}(k)\rangle.\nu a.\overline{c}\langle a\rangle.a(x).\nu r.\overline{a}\langle r\rangle \ \mathcal{S} & \nu k.\overline{s}\langle \mathtt{pk}(k)\rangle.\ \nu a.\overline{c}\langle a\rangle.a(x). \\
& \quad \nu r.\ \mathtt{if}\, x = \mathtt{pk}(k) \\
& \qquad \mathtt{then}\, \overline{a}\langle \mathtt{aenc}(\langle m, r\rangle\,, \mathtt{pk}(k))\rangle \,\mathtt{else}\, \overline{a}\langle r\rangle \\
\nu k.\left(\left\{{}^{\mathrm{pk}(k)}/_{u}\right\} \mid \nu a.\overline{c}\langle a\rangle.a(x).\nu r.\overline{a}\langle r\rangle\right) \ \mathcal{S} & \nu k.\left(\left\{{}^{\mathrm{pk}(k)}/_{u}\right\} \mid \nu a.\overline{c}\langle a\rangle.a(x).\right. \\
& \quad \nu r.\ \mathtt{if}\, x = \mathtt{pk}(k) \\
& \qquad \left.\mathtt{then}\, \overline{a}\langle \mathtt{aenc}(\langle m, r\rangle\,, \mathtt{pk}(k))\rangle \,\mathtt{else}\, \overline{a}\langle r\rangle\right) \\
\nu k, a.\left(\left\{{}^{\mathrm{pk}(k),a}/_{u,v}\right\} \mid a(x).\nu r.\overline{a}\langle r\rangle\right) \ \mathcal{S} & \nu k, a.\left(\left\{{}^{\mathrm{pk}(k),a}/_{u,v}\right\} \mid a(x).\right. \\
& \qquad \nu r.\ \mathtt{if}\, x = \mathtt{pk}(k) \\
& \qquad\quad \left.\mathtt{then}\, \overline{a}\langle \mathtt{aenc}(\langle m, r\rangle\,, \mathtt{pk}(k))\rangle \,\mathtt{else}\, \overline{a}\langle r\rangle\right) \\
\nu k, a, r.\left(\left\{{}^{\mathrm{pk}(k),a}/_{u,v}\right\} \mid \overline{a}\langle r\rangle\right) \ \mathcal{S} & \nu k, a, r.\left(\left\{{}^{\mathrm{pk}(k),a}/_{u,v}\right\} \mid \mathtt{if}\, n\left\{{}^{\mathrm{pk}(k),a}/_{u,v}\right\} = \mathtt{pk}(k)\right. \\
& \qquad\quad \left.\mathtt{then}\, \overline{a}\langle \mathtt{aenc}(\langle m, r\rangle\,, \mathtt{pk}(k))\rangle \,\mathtt{else}\, \overline{a}\langle r\rangle\right) \\
\nu k, a, r.\left(\left\{{}^{\mathrm{pk}(k),a,r}/_{u,v,w}\right\} \mid 0\right) \ \mathcal{S} & \nu k, a, r.\left(\left\{{}^{\mathrm{pk}(k),a,\mathtt{aenc}(\langle m,r\rangle,\mathrm{pk}(k))}/_{u,v,w}\right\} \mid 0\right) \\
\nu k, a, r.\left(\left\{{}^{\mathrm{pk}(k),a,r}/_{u,v,w}\right\} \mid 0\right) \ \mathcal{S} & \nu k, a, r.\left(\left\{{}^{\mathrm{pk}(k),a,r}/_{u,v,w}\right\} \mid 0\right)
\end{array}$$

where s, c, m and n are messages and u, v and w are variables such that $s, c, m, n \# k, a, r$, and $u \# s, c, m, k, a, r$, and $v \# s, c, m, k, a, r, u$, and $w \# s, c, m, n, k, a, r, u, v$.

Fig. 3. Relation $\mathcal{S}$ defining a quasi-open bisimulation verifying the anonymity of *Server A* in the case for a single session, without replication.

secret key w. For another such example, $\nu k.\overline{a}\langle \mathtt{aenc}(x, \mathtt{pk}(k))\rangle.\overline{a}\langle \mathtt{aenc}(y, \mathtt{pk}(k))\rangle$ and $\nu k.\overline{a}\langle \mathtt{aenc}(x, \mathtt{pk}(k))\rangle.\overline{a}\langle \mathtt{aenc}(z, \mathtt{pk}(k))\rangle$ are labelled bisimilar (which assumes x, y, z are distinct names), but not quasi-open bisimilar (which instead assumes x, y, z are variables). To see why, observe the above processes can reach the extended processes $\nu k.\left(\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\mathtt{aenc}(y,\mathtt{pk}(k))}/_{v,w}\right\} \mid 0\right)$ and $\nu k.\left(\left\{{}^{\mathtt{aenc}(x,\mathtt{pk}(k)),\mathtt{aenc}(z,\mathtt{pk}(k))}/_{v,w}\right\} \mid 0\right)$, at which point the attacker has the power to set $x = y$, thereby reaching a scenario explained after Definition 4, where the attacker can observe the same message is output twice for the process on the left but not for the process on the right. This feature of quasi-open bisimilarity is related to the security property of *strong secrecy* [11], where the open nature of secrets represents that the attacker may interfere with messages at runtime.

4.2 Running Example of a Privacy Property

We now have the mechanisms to verify the minimal privacy example from Sect. 2. For greater clarity, firstly consider the case of a single session, i.e., with replication removed. The equivalence of running examples *Server A* and *Server B* for the single session case (without replication) can be established by taking the least *symmetric open* relation satisfying the constraints in Fig. 3. The critical observation is that message n in Fig. 3 ranges over all permitted inputs. Since $n = u$ is permitted, we have the following pair in relation $\mathcal{S}$.

$$\begin{array}{l}
\nu k, a, r.\left(\left\{{}^{\mathtt{pk}(k),a}/_{u,v}\right\} \mid \overline{a}\langle r\rangle\right) \ \ \mathcal{S} \ \ \nu k.a, r.\left(\left\{{}^{\mathtt{pk}(k),a}/_{u,v}\right\} \mid \mathtt{if}\, \mathtt{pk}(k) = \mathtt{pk}(k) \ \mathtt{then}\right. \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \left.\overline{a}\langle \mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(k))\rangle \,\mathtt{else}\, \overline{a}\langle r\rangle\right)
\end{array}$$

In the above, observe the branch sending an encrypted message is enabled. In contrast to the above, if n is any message term not equivalent to u then we have $k, a, r \models n\left\{{}^{\mathtt{pk}(k), a}/_{u,v}\right\} \neq \mathtt{pk}(k)$ since if n were a message term such that $k, a, r \# n$ such that $n\left\{{}^{\mathtt{pk}(k), a}/_{u,v}\right\} = \mathtt{pk}(k)$, then n must be equivalent to u. Thus in all other cases the `else` branch is enabled.

Notice $\nu k, a, r.\left(\left\{{}^{\mathtt{pk}(k), a, \mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(k))}/_{u,v,w}\right\} | 0\right)$ and $\nu k, a, r.\left(\left\{{}^{\mathtt{pk}(k), a, r}/_{u,v,w}\right\} | 0\right)$ are statically equivalent, reachable when $n =_E u$. To see why, observe that an attacker neither has the key k to decrypt $\mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(k))$, nor can an attacker reconstruct the message $\langle m, r\rangle$, without knowing r.

For the unbounded case, consider the least symmetric open relation $\mathcal{T}$ satisfying the constraints in Fig. 4. This generalises the finite case by defining all scenarios where there are l parallel sessions that are either in the state of having just announced the communication channel a, having just received a message, or have responded already. This definition is closed under all transitions and reachability, as required to establish that $\mathcal{T}$ is a quasi-open bisimulation. Indeed $\mathcal{T}$ is a quasi-open bisimulation such that *Server A* $\mathcal{T}$ *Server B*. Hence the desired privacy property of *Server A*, first mentioned in Sect. 2, is verified.

$$Responder \triangleq \nu a.\overline{c}\langle a\rangle.a(x).\nu r.\mathtt{if}\, x = \mathtt{pk}(k)\ \mathtt{then}\, \overline{a}\langle \mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(k))\rangle\, \mathtt{else}\, \overline{a}\langle r\rangle$$

$$\nu k.\overline{s}\langle \mathtt{pk}(k)\rangle.!\nu a.\overline{c}\langle a\rangle.a(x).\nu r.\overline{a}\langle r\rangle \ \ \mathcal{T}\ \ \nu k.\overline{s}\langle \mathtt{pk}(k)\rangle.!Responder$$

$$\begin{array}{l} \nu k, a_1, \ldots, a_l, r_1, \ldots\ r_l.\big(\ \sigma \mid P_1 \mid \ldots P_l \\ \qquad\qquad\qquad \mid\ !\nu a.\overline{c}\langle a\rangle.a(x).\nu r.\overline{a}\langle r\rangle\big) \end{array} \ \mathcal{T}\ \begin{array}{l} \nu k, a_1, \ldots, a_l, r_1, \ldots\ r_l.\big(\ \theta \mid Q_1 \mid \ldots Q_l \\ \qquad\qquad\qquad\qquad \mid\ !Responder\ \big) \end{array}$$

for any I, J, I', J' partitioning $\{1, \ldots\ l\}$ such that the following hold

$$\begin{array}{ll} u\sigma = \mathtt{pk}(k) & \\ v_i\sigma = a_i & \text{if } i \in \{1, \ldots\ l\} \\ w_i\sigma = r_i & \text{if } i \in I' \cup J' \end{array} \qquad \begin{array}{ll} u\theta = \mathtt{pk}(k) & \\ v_i\theta = a_i & \text{if } l \in \{1, \ldots\ l\} \\ w_i\theta = \mathtt{aenc}(\langle m, r_i\rangle, \mathtt{pk}(k)) & \text{if } l \in I' \\ w_i\theta = r_i & \text{if } l \in J' \end{array}$$

$$P_i \triangleq \begin{cases} a_i(x).\nu r.\overline{a_i}\langle r\rangle & \text{if } i \in I \\ \overline{a_i}\langle r_i\rangle & \text{if } i \in J \\ 0 & \text{if } l \in I' \cup J' \end{cases}$$

$$Q_i \triangleq \begin{cases} a_i(x).\nu r.\mathtt{if}\, x\theta = \mathtt{pk}(k)\ \mathtt{then}\, \overline{a_i}\langle \mathtt{aenc}(\langle m, r\rangle, \mathtt{pk}(k))\rangle\, \mathtt{else}\, \overline{a_i}\langle r\rangle & \text{if } i \in I \\ \mathtt{if}\, n_i\theta = \mathtt{pk}(k)\ \mathtt{then}\, \overline{a_i}\langle \mathtt{aenc}(\langle m, r_i\rangle, \mathtt{pk}(k))\rangle\, \mathtt{else}\, \overline{a_i}\langle r_i\rangle & \text{if } i \in J \\ 0 & \text{if } l \in I' \cup J' \end{cases}$$

s, c, m, n_i are messages such that $s, c, m, n_i \# k, a_1, \ldots\ a_l, r_1, \ldots\ r_l$ and $u, v_1, \ldots, v_l, w_1, \ldots w_l$ are distinct variables such that $u, v_1, \ldots\ v_l, w_1 \ldots\ w_l \# s, c, m, a_1, \ldots\ a_l, k, r_1, \ldots\ r_l$

Fig. 4. Relation $\mathcal{T}$ verifying *Server B* $\sim$ *Server A* in the unbounded case.

A subtlety is that $\mathcal{T}$ is not the least quasi-open bisimulation witnessing *Server A* $\sim$ *Server B*, since we *over approximated* by allowing inputs to possibly use outputs from the future. This over approximation is correct, since we can always have additional redundant terms in a bisimulation set, as long as they are also closed under the relevant conditions. Indeed, this illustrates a practical benefit of bisimilarity – we can find abstractions that reduce the amount of verification work.

4.3 Quasi-open Bisimilarity is Sound and Complete

As illustrated in the previous sub-section, a core guarantee offered by quasi-open bisimilarity is that it is a congruence relation. We prove quasi-open bisimilarity is preserved by all contexts, notably under input prefixes; and, furthermore, coincides exactly with open barbed bisimilarity, which is the coarsest (strong) bisimilarity congruence.

Theorem 2 (contexts). *If $P \sim Q$ then for all contexts $\mathcal{C}\{\,\cdot\,\}$, we have $\mathcal{C}\{P\} \sim \mathcal{C}\{Q\}$.*

The most involved cases of Theorem 2 are those showing quasi-open bisimilarity is preserved under parallel composition and replication; while the most novel case is for mismatch, which relies on the notion of an open relation given in Definition 5. Given Theorem 2, the soundness of quasi-open bisimilarity with respect to open barbed bisimilarity follows immediately.

Corollary 1 (soundness). *If $P \sim Q$ then $P \simeq Q$.*

Completeness, expressed in Theorem 3, supports our claim that quasi-open bisimilarity in Definition 6 is a correct and canonical (strong) bisimilarity congruence for the applied π-calculus. This theorem is the fundamental property of quasi-open bisimilarity that does not hold for open bisimilarity.

Theorem 3 (completeness). *Quasi-open bisimilarity coincides with open barbed bisimilarity.*

It is interesting to compare the proof of Theorem 3 to the corresponding proof for the π-calculus [41]. In the corresponding proof for the π-calculus checks are built into bound output transitions to ensure extruded private names are fresh. In the proof of Theorem 3 no such checks are required for output transitions; such checks are subsumed by checking static equivalence.

Strong v.s. Weak Bisimilarity. Observe Theorem 3 is for a strong formulation of quasi-open bisimilarity. The weak/strong dimension [44] (as with other dimensions such as interleaving v.s. true concurrency [45], for instance) is a perpendicular issue to the focus of this paper. Quasi-open variants of various equivalences and preorders can also be defined, so this scientific discussion on attacker models should not be limited to strong bisimilarity. Sometimes weak equivalences can be avoided. For example, for privacy properties, such as unlinkability of ePassports, the traditional formulation in terms of a weak bisimilarity problem [8] has been shown to be reducible to an equivalent strong bisimilarity problem that is easier to check, since we have image finiteness [28], i.e., for any label π each process has finitely many π-labelled transitions.

5 Comparison to Related Work on Observational Equivalence

Most notions of bisimilarity previously introduced for cryptographic calculi (e.g., hedged bisimilarity, labelled bisimilarity, early bisimilarity) coincide with observational equivalence [2,4,5,10,15–17,31,32,34]. Observational equivalence is a restriction of open barbed bisimilarity (Definition 3), considering only contexts of the form $\{\,\cdot\,\} \mid P$

that add a new process in parallel at every step of the bisimulation game. This makes the equivalence strictly coarser than open barbed bisimilarity, however observational equivalence is not a congruence relation. Intermediate results on symbolic bisimulations [18,25] also closely approximate observational equivalence.

The gap between observational equivalences and open barbed bisimilarity is thoroughly explored in the context of the π-calculus [7,29,40,41]. Open barbed bisimilarity is finer than observational equivalence since, $\pi.P + \pi.Q$ is observationally equivalent to $\pi.P + \pi.Q + \pi.\mathtt{if}\, x = y\, \mathtt{then}\, P\, \mathtt{else}\, Q$, but these processes are not open barbed bisimilar in general. Yet these processes are equivalent if we take barbed equivalence [36], which is the largest congruence contained within observational equivalence, lying strictly between open barbed bisimilarity and observational equivalence. In Sect. 4.1, we did mention there are examples of noninterference properties that can be formulated using a congruence. However, it remains an open question whether there exists a realistic privacy property, as opposed to the toy equation immediately above, that cannot be verified using open barbed bisimilarity but can be analysed using barbed equivalence.

If one does insist that a property is defined in terms of observational equivalence, we may still use quasi-open bisimilarity as an under-approximation. If an attack is discovered, we can check whether an attack is also valid classically (possibly making use of modal logic intuitionistic $\mathcal{FM}$ described in the extended technical report [29]). If the attack is also classically valid it is also a counterexample for observational equivalence. This methodology was used to resolve the problem of whether there is an attack on the BAC protocol for ePassports [26,28], as originally stated in terms of observational equivalence [8].

6 Conclusion

This paper justifies the bisimilarity congruence quasi-open bisimilarity as a method for reasoning about protocols expressed using the applied π-calculus. The equivalence we converge on, *quasi-open bisimilarity*, can be seen as an enhancement of existing methods, balancing between the strengths of *labelled bisimilarity* [4,10,15–17,31,32,34] and *open bisimilarity*.

The bisimilarity congruence, *open bisimilarity*, has previously been introduced for the spi-calculus [19,42,43]. However, the spi-calculus could not verify privacy properties demanding mismatch, and is less abstract, being hard-wired with a fixed message theory; which were problems addressed in recent work that lifts open bisimilarity to the more general setting of the applied π-calculus [28]. By moving to the coarser equivalence *quasi-open bisimilarity* we are able to verify more privacy properties, such as the typical privacy-preserving protocol in Sect. 2, involving *if-then-else* with a guard depending on private information. Some equivalences, such as differential equivalence [12,21], which compares two structurally identical processes that differ only in the terms they exchange, are incomplete and hence may report attacks that trivially do not exist. Hence when differential equivalence reports an attack, it may not exist for trivial reasons – a problem minimised by the fact that quasi-open bisimilarity adheres to a completeness criterion for observational congruences (Theorem 3).

Equivalences coarser than quasi-open bisimilarity are either not congruences or are not bisimilarities, meaning that some corresponding proof techniques cannot be applied.

The gap between quasi-open bisimilarity and classical *labelled bisimilarity* is small—we insist on an open relation (Definition 2). However, the gap is significant, since we obtain a complete congruence. In an extended version of this paper in a technical report [27], we go further by demonstrating that we are able to logically characterise quasi-open bisimilarity, using an intuitionistic modal logic useful for describing attacks.

References

1. Abadi, M., Cortier, V.: Deciding knowledge in security protocols under equational theories. Theor. Comput. Sci. **367**(1–2), 2–32 (2006). https://doi.org/10.1016/j.tcs.2006.08.032
2. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: POPL, pp. 104–115 (2001). https://doi.org/10.1145/360204.360213
3. Abadi, M., Fournet, C.: Private authentication. Theor. Comput. Sci. **322**(3), 427–476 (2004). https://doi.org/10.1016/j.tcs.2003.12.023
4. Abadi, M., Gordon, A.D.: A bisimulation method for cryptographic protocols. Nord. J. Comput. **5**(4), 267–303 (1998)
5. Abadi, M., Blanchet, B., Fournet, C.: The applied pi calculus: mobile values, new names, and secure communication. J. ACM **65**(1), 1–41 (2017). https://doi.org/10.1145/3127586
6. Ahn, K.Y., Horne, R., Tiu, A.: A characterisation of open bisimilarity using an intuitionistic modal logic. In: Meyer, R., Nestmann, U. (eds.) 28th International Conference on Concurrency Theory, CONCUR 2017, 5–8 September 2017, Berlin, Germany, vol. 85 of LIPIcs, pp. 7:1–7:17 (2017). https://doi.org/10.4230/LIPIcs.CONCUR.2017.7
7. Ahn, K.Y., Horne, R., Tiu, A.: A characterisation of open bisimilarity using an intuitionistic modal logic. Log. Meth. Comp. Sci. (2021). https://arxiv.org/abs/1701.05324. In press
8. Arapinis, M., Chothia, T., Ritter, E., Ryan, M.: Analysing unlinkability and anonymity using the applied pi calculus. In 23rd IEEE Computer Security Foundations Symposium, pp. 107–121 (2010). https://doi.org/10.1109/CSF.2010.15
9. Ayala-Rincón, M., Fernández, M., Nantes-Sobrinho, D.: Intruder deduction problem for locally stable theories with normal forms and inverses. Theor. Comput. Sci. **672**, 64–100 (2017). https://doi.org/10.1016/j.tcs.2017.01.027
10. Bengtson, J., Johansson, M., Parrow, J., Victor, B.: Psi-calculi: a framework for mobile processes with nominal data and logic. Log. Meth. Comp. Sci. **7**(1) (2011). https://doi.org/10.2168/LMCS-7(1:11)2011
11. Blanchet, B.: Automatic proof of strong secrecy for security protocols. In: 2004 Proceedings of IEEE Symposium on Security and Privacy, pp. 86–100. IEEE (2004). https://doi.org/10.1109/SECPRI.2004.1301317
12. Blanchet, B., Abadi, M., Fournet, C.: Automated verification of selected equivalences for security protocols. J. Log. Algebr. Program. **75**(1), 3–51 (2008). https://doi.org/10.1016/j.jlap.2007.06.002
13. Bonchi, F., König, B., Montanari, U.: Saturated semantics for reactive systems. In: 21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12–15 August 2006, Seattle, WA, USA, Proceedings, pp. 69–80. IEEE Computer Society (2006). https://doi.org/10.1109/LICS.2006.46
14. Bonchi, F., Gadducci, F., Monreale, G.V.: A general theory of barbs, contexts, and labels. ACM Trans. Comput. Log. **15**(4), 35:1-35:27 (2014). https://doi.org/10.1145/2631916

15. Boreale, M., De Nicola, R., Pugliese, R.: Proof techniques for cryptographic processes. SIAM J. Comput. **31**(3), 947–986 (2001). https://doi.org/10.1137/S0097539700377864
16. Borgström, J.: A complete symbolic bisimilarity for an extended Spi calculus. Electron. Notes Theor. Comput. Sci. **242**(3), 3–20 (2009). https://doi.org/10.1016/j.entcs.2009.07.078
17. Borgström, J., Nestmann, U.: On bisimulations for the Spi calculus. Math. Struct. Comput. Sci. **15**(3), 487–552 (2005). https://doi.org/10.1017/S0960129505004706
18. Borgström, J., Briais, S., Nestmann, U.: Symbolic bisimulation in the Spi calculus. In: Gardner, P., Yoshida, N. (eds.) CONCUR 2004. LNCS, vol. 3170, pp. 161–176. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28644-8_11
19. Briais, S., Nestmann, U.: Open bisimulation, revisited. Theor. Comput. Sci. **386**(3), 236–271 (2007). https://doi.org/10.1016/j.tcs.2007.07.010
20. Bursuc, S., Comon-Lundh, H., Delaune, S.: Deducibility constraints and blind signatures. Inf. Comput. **238**, 106–127 (2014). https://doi.org/10.1016/j.ic.2014.07.006
21. Cheval, V., Blanchet, B.: Proving more observational equivalences with ProVerif. In: Basin, D., Mitchell, J.C. (eds.) POST 2013. LNCS, vol. 7796, pp. 226–246. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36830-1_12
22. Cheval, V., Comon-Lundh, H., Delaune, S.: A procedure for deciding symbolic equivalence between sets of constraint systems. Inf. Comput. **255**(Part 1), 94–125 (2017). https://doi.org/10.1016/j.ic.2017.05.004
23. Cortier, V., Smyth, B.: Attacking and fixing Helios: an analysis of ballot secrecy. In: 2011 IEEE 24th Computer Security Foundations Symposium, pp. 297–311, June 2011. https://doi.org/10.1109/CSF.2011.27
24. Delaune, S.: Analysing privacy-type properties in cryptographic protocols. In: Kirchner, H. (ed.) 3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018), volume 108 of LIPIcs, Dagstuhl, Germany, pp. 1:1–1:21. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/LIPIcs.FSCD.2018.1
25. Delaune, S., Kremer, S., Ryan, M.D.: Symbolic bisimulation for the applied pi calculus. J. Comput. Secur. **18**(2), 317–377 (2010). https://doi.org/10.3233/JCS-2010-0363
26. Filimonov, I., Horne, R., Mauw, S., Smith, Z.: Breaking unlinkability of the ICAO 9303 standard for e-passports using bisimilarity. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) ESORICS 2019. LNCS, vol. 11735, pp. 577–594. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29959-0_28
27. Horne, R.: A bisimilarity congruence for the applied pi-calculus sufficiently coarse to verify privacy properties. Arxiv, arXiv:1811.02536, pp. 1–31 (2018.). https://arxiv.org/abs/1811.02536
28. Horne, R., Mauw, S.: Discovering ePassport vulnerabilities using bisimilarity. Logical Methods in Computer Science **17**(2), 24:1-24:52 (2021). https://doi.org/10.23638/LMCS-17(2:24)2021
29. Horne, R., Ahn, K.Y., Lin, S., Tiu, A.: Quasi-open bisimilarity with mismatch is intuitionistic. In: Dawar, A., Grädel, E. (eds.) Proceedings of LICS 2018: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2018), Oxford, United Kingdom, 9–12 July 2018, p. 10 (2018). https://doi.org/10.1145/3209108.3209125
30. Horne, R., Mauw, S., Yurkov, S.: Breaking and fixing unlinkability of the key agreement protocol for 2nd gen EMV payments (2021). https://arxiv.org/abs/2105.02029
31. Johansson, M., Bengtson, J., Victor, B., Parrow, J.: Weak equivalences in psi-calculi. In: 2010 25th Annual IEEE Symposium on Logic in Computer Science, pp. 322–331, July 2010. https://doi.org/10.1109/LICS.2010.30
32. Johansson, M., Victor, B., Parrow, J.: Computing strong and weak bisimulations for psi-calculi. J. Logic Algebraic Program. **81**(3), 162–180 (2012). https://doi.org/10.1016/j.jlap.2012.01.001

33. Kremer, S., Ryan, M.: Analysis of an electronic voting protocol in the applied pi calculus. In: Sagiv, M. (ed.) ESOP 2005. LNCS, vol. 3444, pp. 186–200. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31987-0_14
34. Liu, J., Lin, H.: A complete symbolic bisimulation for full applied pi calculus. Theor. Comput. Sci. **458**, 76–112 (2012). https://doi.org/10.1016/j.tcs.2012.07.034
35. Milner, R.: The polyadic π-calculus: a tutorial. In: Bauer, F.L., Brauer, W., Schwichtenberg, H. (eds.) Logic and Algebra of Specification. NATO ASI Series, vol. 94, pp. 203–246 (1993). https://doi.org/10.1007/978-3-642-58041-3_6
36. Milner, R., Sangiorgi, D.: Barbed bisimulation, pp. 685–695 (1992). https://doi.org/10.1007/3-540-55719-9_114
37. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, Part I and II. Inf. Comput. **100**(1), 1–100 (1992). https://doi.org/10.1016/0890-5401(92)90008-4
38. Montanari, U., Sassone, V.: Dynamic congruence vs. progressing bisimulation for CCS. Fundam. Inf. **16**(2), 171–199 (1992)
39. Sangiorgi, D.: On the proof method for bisimulation. In: Wiedermann, J., Hájek, P. (eds.) MFCS 1995. LNCS, vol. 969, pp. 479–488. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60246-1_153
40. Sangiorgi, D.: A theory of bisimulation for the π-calculus. Acta Inf. **33**(1), 69–97 (1996). https://doi.org/10.1007/s002360050036
41. Sangiorgi, D., Walker, D.: On barbed equivalences in π-calculus. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 292–304. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44685-0_20
42. Tiu, A.: A trace based bisimulation for the Spi calculus: an extended abstract. In: Shao, Z. (ed.) APLAS 2007. LNCS, vol. 4807, pp. 367–382. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76637-7_25
43. Tiu, A., Nguyen, N., Horne, R.: SPEC: an equivalence checker for security protocols. In: Igarashi, A. (ed.) APLAS 2016. LNCS, vol. 10017, pp. 87–95. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47958-3_5
44. Glabbeek, R.J.: The linear time—branching time spectrum II. In: Best, E. (ed.) CONCUR 1993. LNCS, vol. 715, pp. 66–81. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57208-2_6. ISBN 3-540-57208-2
45. van Glabbeek, R.J., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Inf. **37**(4/5), 229–327 (2001). https://doi.org/10.1007/s002360000041