A Security Model for Anonymous Credential Systems

Andreas Pashalidis* and Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
E-mail: {A.Pashalidis,C.Mitchell}@rhul.ac.uk

corrected version

July 5, 2004

Abstract

This paper proposes a formal model of the Bellare-Rogaway type [1] that enables one to prove the security of an anonymous credential system in a complexity theoretic framework. The model abstracts away from *how* a specific instance of anonymous credential system achieves its goals; instead it defines *what* these goals are. The notions of credential unforgeability, non-transferability, pseudonym unlinkability and pseudonym owner protection are formally defined and the relationships between them are explored. The model is a step towards a formal treatment of the level of privacy protection that anonymous credential systems can and should achieve, both in terms of pseudonym unlinkability and user anonymity.

Keywords: anonymous credential systems, pseudonym systems, privacy, anonymity, unlinkability, provable security.

1 Introduction

1.1 Background and motivation

Anonymous credential or 'pseudonym' systems allow users to interact with organisations using distinct and unlinkable pseudonyms. In particular, a user can obtain a credential (a statement of a designated type that attests to one or more of the user's attributes) from one organisation and then 'show' it to another, such that the two organisations cannot link the issuing and showing acts; this renders the user's transactions unlinkable. Of course this unlinkability is limited; if only one credential is ever issued with a particular set of attributes, then clearly all credential showings containing this set of attributes can be linked to each other and to the unique issued credential. Pseudonym systems must prevent users from showing credentials that have not been issued (i.e. they must guarantee 'credential unforgeability'), and prevent users from pooling their credentials (for example, to collectively obtain a new credential that each user individually would not be able to). This latter property is usually referred to as 'credential non-transferability'.

Security models of pseudonym systems, and proofs (where given), do not usually allow reasoning about the resulting degrees of user anonymity and pseudonym unlinkability. This paper, following the ideas first set out by Bellare and Rogaway in [1], proposes a model that is based on complexity theoretic arguments and which potentially leads to information theoretic anonymity

*The author is sponsored by the State Scholarship Foundation of Greece.

metrics. It abstracts away from the particulars of *how* specific pseudonym system instances achieve their goals; instead it focuses on *what* these goals are. The model captures security properties for both organisations (credential unforgeability and non-transferability), and users, both in terms of 'traditional' security (pseudonym owner protection) and privacy (pseudonym unlinkability and user anonymity). The model makes a clear distinction between the different notions and allows the relationships between them to be analysed.

1.2 Related work

Pseudonym systems were first introduced by Chaum in the 1980s [4]. Since then, numerous pseudonym systems have been proposed, each with its own particular set of entities, underlying problems, assumptions and properties. Examples of such systems are given in [2, 3, 5, 6]. The most relevant work to this paper is probably the formal treatment of the anonymous credential system in [3]. There, security is defined based on the indistinguishability between the transcripts of protocols that occur in an 'ideal' world (where a universally trusted party guarantees security), and the 'real world' (where such a party does not exist). In that model, transactions between users and organisations correspond to well-defined events, and the adversary acts like an event scheduler; he can arbitrarily trigger events of his choice. In the model of [3], however, the relationship between the different security notions that a pseudonym system should satisfy is somewhat hidden by the fact that the universally trusted party takes care of them. Also, in that model, the adversary is not allowed to corrupt players in an adaptive fashion. While our model retains the property that the adversary gets to specify the order of events in the system, he can also adaptively corrupt players. Further, the model allows a relatively easy analysis of the relationships between different notions. This is due to the fact that we abstract away from properties that do not lie at the same level of abstraction as that at which a pseudonym system operates.

1.3 What we do *not* do

Our model does not capture 'traditional' communications security properties, such as entity authentication. This is not an omission; these issues are outside the scope of the model (other well-established security models can be used to reason about such issues). Of course, if users do not authenticate organisations, and if the communications in the system are not appropriately protected at the session level[1], then there cannot be any security. However, the way these services are provided lies at a different level of abstraction. We therefore assume that they are provided by the infrastructure that allows users and organisations to communicate. We also assume that, within this infrastructure, users remain anonymous to organisations (i.e. we assume an anonymous user-to-organisation channel).

The remainder of the paper is organised as follows. The next section describes the formal model of pseudonym systems. Section 2.2 establishes the notions of pseudonym owner protection, credential unforgeability and credential non-transferability, which together capture the notions of soundness for a scheme. Further, section 2.3 provides a brief discussion of the notions and explains the relationships between them. Section 2.4 establishes the notion of pseudonym unlinkability which is discussed in section 2.5. Section 2.6 then establishes the notion of pseudonym indistinguishability and shows it is a necessary condition for unlinkability. Finally, section 2.7 addresses the issue of anonymity in pseudonym systems, while section 3 concludes the paper and gives directions for further research.

[1] What it means for communications to be 'appropriately protected at the session level' may vary from system to system. Typically, it means that each protocol execution is bound to exactly one conversation between the involved parties. It may also imply protection of the integrity and confidentiality of the communications occurring within the session.

2 Security of pseudonym systems

In this section we describe our model of a pseudonym system. We regard a pseudonym system as being comprised of the players in the system and the procedures through which they interact. The players, in particular, are divided into users, issuing organisations and verifying organisations. Since users are known to each organisation under a different pseudonym, indeed possibly under multiple pseudonyms, a procedure must be in place according to which a user and an organisation establish a new pseudonym; we call this the 'pseudonym establishment protocol'. Procedures must also be in place that allow users to obtain credentials (using the pseudonym that was established with the issuer) and to show them (using the pseudonym that was established with the verifier). We call the former the 'credential issuing protocol' and the latter the 'credential showing protocol'.

In our model, credential types are in one-to-one correspondence with (combinations of) user attributes; in other words, each combination of attributes defines a credential type. An organisation that, for example, issues demographic credentials containing the fields `sex` and `age group`, with possible values of {`male`,`female`} and {`18-`,`18-30`,`30-50`,`50+`} respectively, in our model may actually issue credentials of up to eight different types (one for each combination of values).

2.1 The model

A protocol `prot` is assumed to be a tuple of interactive Turing machines; an execution of `prot` is said to be successful if and only if all machines accept. The set of all non-zero polynomial functions in the natural number k is denoted by $\texttt{poly}(k)$. A real-valued function $\epsilon : \mathbb{N} \to \mathbb{R}$, is said to be negligible in k if and only if $0 \leq \epsilon(k) < 1/|q(k)|$ for any $q \in \texttt{poly}(k)$ and for all sufficiently large k.

Remark 1 *We are concerned in this paper with situations where two functions f and g satisfy $f(k) > g(k) + \epsilon(k)$ for any negligible function ϵ and for all sufficiently large k. To simplify the discussion we abuse our notation slightly and simply say that f is greater than $g + \epsilon(k)$, i.e. we omit explicit references to k, and we also omit the rider 'for all sufficiently large k'.*

Definition 1 *A pseudonym system is a tuple*

$$(U, I, V, k, \texttt{init}, P, T, \texttt{peprot}, \texttt{ciprot}, \texttt{csprot})$$

whose elements are as follows.

- *U is the set of users.*
- *I is the set of credential issuing organisations ('issuers' in short).*
- *V is the set of credential verifying organisations ('verifiers' in short).*
- *k (a natural number) is the system security parameter.*
- `init` *is the initialisation algorithm; on input (U, I, V, k), it outputs descriptions of the sets P and T. Depending on the particular scheme, it may also output public parameters and private values for (a subset of) the players in the scheme, i.e. users, issuers and verifiers.*
- *P is the set of pseudonyms.*
- *T is the set of credential types.*
- `peprot` *is the* pseudonym establishment protocol*: any user/organisation pair $(u, o) \in U \times (I \cup V)$ may execute* `peprot`*; if the protocol succeeds, u and o will have established a pseudonym $p \in P$ and we write $\texttt{peprot}_{u,o,p}$. (The user u is called the* owner *of p, and will typically also possess some private output associated with p as necessary to engage in* `ciprot` *and* `csprot`*.)*

- `ciprot` *is the* credential issuing protocol*: any user/issuer pair* $(u, i) \in U \times I$ *may execute* `ciprot` *with respect to a pseudonym* $p \in P$ *associated with* u *and* i *(established using* `peprot`*) and for a particular credential type* $t \in T$*. If successful, we say that* i *has issued a credential of type* t *on pseudonym* p *to* u*, and we write* $\texttt{ciprot}_{i,p,t}$*.*
- `csprot` *is the* credential showing protocol*: any user/verifier pair* $(u, v) \in U \times V$ *may execute* `csprot` *with respect to a pseudonym* $p \in P$ *associated with* u *and* v *(established using* `peprot`*) and for a particular credential type* $t \in T$*; if the protocol succeeds we say that* u *has shown a credential of type* t *on pseudonym* p *to* s *and we write* $\texttt{csprot}_{v,p,t}$*.*

Each issuer $i \in I$ defines a set $T_i \subseteq T$ of credential types that it intends to issue in the future[2]. It is required that, for all distinct $i, i' \in I$, $T_i \cap T_{i'} = \emptyset$[3]. We denote the set of active credential types in the system by $T^* \stackrel{def}{=} \bigcup_{i \in I} T_i$. It is required that $|T^*| \in \texttt{poly}(k)$. Also note that, by definition, $|U| \in \texttt{poly}(k)$, $|I| \in \texttt{poly}(k)$ and $|V| \in \texttt{poly}(k)$.

2.2 The games and soundness

In order to formalise our notions of security for a pseudonym system, we define a series of games between two Turing machines: a Challenger and an Adversary. Each game captures a specific property of the pseudonym system. In this section we define Game 1, which captures 'pseudonym owner protection', Game 2, which captures 'credential unforgeability', and Game 3, which captures 'credential non-transferability'. In sections 2.4 and 2.6 below we define Game 4 and Game 5, which capture 'unlinkability' and 'indistinguishability' of pseudonyms, respectively.

At the beginning of all games, the Challenger sets up the system by selecting the set of users U, issuers I, verifiers V, and a security paramenter k and by running `init` with these as input. At this point, the Challenger controls all the players in the system. He defines the sets T_i for each issuer. The Adversary, which is assumed to be a probabilistic polynomial time (and space) algorithm and is denoted by $\mathcal{A}$, then receives as input the sets U, I, V and T_i, descriptions of the sets P and T, and the system's public information. As explained above, it is assumed that the underlying communication infrastructure appropriately protects the communications between users and organisations at the session level. Thus, $\mathcal{A}$ models a passive adversary that faithfully transmits messages between parties.

Each of the games consists of two distinct and successive phases. During the first phase of each game, $\mathcal{A}$ may issue (oracle type) queries to the Challenger; during the second phase he may not. During the first phase of Game 1, 2 and 3, $\mathcal{A}$ may issue the following types of query to the Challenger.

`runpeprot`(u, o)**:** $\mathcal{A}$ may arbitrarily select a user/organisation pair $(u, o) \in U \times (I \cup V)$ and issue this query. When this happens, the Challenger makes u and o execute $\texttt{peprot}_{u,o,p}$. The Challenger replies `true` if the protocol execution is successful and `false` otherwise. (If the execution is successful, u and o will have established a new pseudonym $p \in P$; $\mathcal{A}$, however, does not learn its value.)

`runciprot`(u, i, t)**:** $\mathcal{A}$ may arbitrarily select a user/issuer pair $(u, i) \in U \times I$ and a credential type $t \in T_i$ and issue this query. When this happens, the Challenger selects a pseudonym p from set of pseudonyms that u and i have established[4] and makes u and i execute $\texttt{ciprot}_{i,p,t}$. He replies `true` if the protocol execution is successful and `false` otherwise (including the case where u and i have not established any pseudonym). Note that $\mathcal{A}$ does not learn the value of p.

[2] In certain existing pseudonym systems, credential types are identified with some form of public verification key. These keys are typically published.

[3] This is easily achieved by having a unique identifier of each i embedded into all its types T_i.

[4] We do not specify the probability distribution according to which the Challenger selects p from the set of pseudonyms u has established, since this should not affect security.

`runcsprot`(u, v, t): $\mathcal{A}$ may arbitrarily select a user/verifier pair $(u, v) \in U \times V$ and a credential type $t \in T$ and issue this query. When this happens, the Challenger selects a pseudonym p from the set of pseudonyms that u and v have established and makes u and v execute $\texttt{csprot}_{v,p,t}$. He replies `true` if the protocol execution is successful and `false` otherwise (including the case where u and v have not established any pseudonym). Note that $\mathcal{A}$ does not learn the value of p.

`corruptUser`(u): $\mathcal{A}$ may arbitrarily select a user $u \in U$ and issue this query. When this happens, the Challenger hands all the private information of u to $\mathcal{A}$. This includes u's pseudonyms, credentials and all his past protocol views. From that point on, the control of u is passed from the Challenger to $\mathcal{A}$.

`corruptIssuer`(i): $\mathcal{A}$ may arbitrarily select an issuer $i \in I$ and issue this query. When this happens, the Challenger hands all the private information of i to $\mathcal{A}$. This includes the set of pseudonyms i has established and all its past protocol views. From that point on, the control of i is passed from the Challenger to $\mathcal{A}$.

`corruptVerifier`(v): $\mathcal{A}$ may arbitrarily select a verifier $v \in V$ and issue this query. When this happens, the Challenger hands all the private information of v to $\mathcal{A}$. This includes the set of pseudonyms v has established and all its past protocol views. From that point on, the control of v is passed from the Challenger to $\mathcal{A}$.

In all games, a global and monotonically increasing variable τ counts $\mathcal{A}$'s queries. We say that the query is issued at the time indicated by τ. At some point in time, $\mathcal{A}$ exits the first phase and enters the second phase. The value of τ at that point is denoted by $\tau_{\texttt{max}}$. In the second phase $\mathcal{A}$ may no longer issue any queries; what happens is specific to each game and is described below.

To describe the games we require some additional notation. In the following, $P_{u,o} \subseteq P$ denotes the set of pseudonyms the user $u \in U$ has established with the organisation $o \in (I \cup V)$ at time $\tau_{\texttt{max}}$ (via $\mathcal{A}$'s `peprot` queries), i.e. $P_{u,o} \stackrel{def}{=} \{p \in P \mid \text{a successful } \texttt{peprot}_{u,o,p} \text{ occurred at a time } \tau \leq \tau_{\texttt{max}}\}$. The set of pseudonyms belonging to u is defined as $P_u \stackrel{def}{=} \bigcup_{o \in (I \cup V)} P_{u,o}$ and the set of pseudonyms that o has established is defined as $P_o \stackrel{def}{=} \bigcup_{u \in U} P_{u,o}$. (Since $\mathcal{A}$ does not learn the value of pseudonyms during their establishment, only u knows P_u and only o knows P_o.) The set of active pseudonyms in the system is defined as $P^* \stackrel{def}{=} \bigcup_{u \in U} P_u$, or, equivalently, $P^* \stackrel{def}{=} \bigcup_{o \in (I \cup V)} P_o$. Since $\mathcal{A}$ is polynomially bounded in k, it holds that $|P^*| \in \texttt{poly}(k)$. It is required that, for all distinct $u, u' \in U$, $P_u \cap P_{u'} = \emptyset$[5]. The function $f : P^* \to U$ maps pseudonyms to their owners, which is well-defined by the assumption that $P_u \cap P_{u'} = \emptyset$.

Let $\hat{U} \subseteq U$, $\hat{I} \subseteq I$ and $\hat{V} \subseteq V$ denote the subsets of users, issuers and verifiers respectively that $\mathcal{A}$ corrupted during the first phase. Further, let $P_{u,t}(x) \subseteq P_u$ denote the subset of pseudonyms belonging to user $u \in U$ on which a credential of type $t \in T^*$ has been issued prior to time x, i.e. $P_{u,t}(x) \stackrel{def}{=} \{p \in P_u \mid \text{a successful } \texttt{ciprot}_{\cdot,p,t} \text{ occurred at time } \tau \leq x\}$.

We now describe the second phase of Games 1, 2 and 3. As mentioned above, $\mathcal{A}$ may no longer issue queries to the Challenger in this phase. He may, however, engage in $\texttt{ciprot}_{p,i,t}$ and $\texttt{csprot}_{p,v,t}$ executions directly with organisations (while pretending to be the user $f(p)$).

Game 1 *(pseudonym owner protection): $\mathcal{A}$ selects a pseudonym/verifier/type triple $(p, v, t) \in P^* \times (V - \hat{V}) \times T$ such that $f(p) \in (U - \hat{U})$. We say that $\mathcal{A}$ wins the game iff he can make v accept in a $\texttt{csprot}_{p,v,t}$ execution.*

Game 2 *(credential unforgeability): $\mathcal{A}$ selects a pseudonym/verifier/type triple $(p, v, t) \in P^* \times (V - \hat{V}) \times (T - \bigcup_{i \in \hat{I}} T_i)$ such that $P_{f(p),t}(\tau_{\texttt{max}}) = \emptyset$ and $\bigcup_{u \in \hat{U}} P_{u,t}(\tau_{\texttt{max}}) = \emptyset$. We say that $\mathcal{A}$ wins the game iff he can make v accept in a $\texttt{csprot}_{p,v,t}$ execution.*

[5] This requirement is a technicality that we need in order to define the function f. It practice it can be met by having `peprot` select pseudonyms uniformly at random from a large enough set P. The pseudonym establishment protocols of some existing schemes are of this form.

Game 3 *(credential non-transferability): $\mathcal{A}$ selects a pseudonym/verifier/type triple $(p, v, t) \in P^* \times (V - \hat{V}) \times (T - \bigcup_{i \in \hat{I}} T_i)$ such that $P_{f(p),t}(\tau_{\mathtt{max}}) = \emptyset$. We say that $\mathcal{A}$ wins the game iff he can make v accept in a $\mathtt{csprot}_{v,p,t}$ execution.*

Definition 2 *A pseudonym system is said to offer pseudonym owner protection, credential unforgeability or credential non-transferability if and only if no adversary $\mathcal{A}$ can win Game 1, 2 or 3, respectively, with a probability greater than any negligible function in k.*

2.3 Discussion

Game 1, 'pseudonym owner protection', captures security for users; nobody — even when colluding with users, issuers and verifiers — should be able to successfully show a credential on a pseudonym of which he is not the owner (i.e. on a pseudonym which was not established by himself). The property is typically achieved by having the pseudonym establishment protocol generate some private output for the user. This output is then treated as a secret that enables the user to authenticate himself as the pseudonym owner during the execution of the credential issuing and showing protocols.

Games 2 and 3 capture security for organisations. In particular, Game 2 captures what is usually perceived as 'credential unforgeability'. If a (dishonest) user can construct a credential by himself (i.e. without obtaining it legitimately from an issuing organisation), if, in other words, the user can *forge* the credential, then the system clearly does not offer credential unforgeability. Game 2 captures unforgeability in this sense. There is, however, a simplistic way for a user to 'forge' a credential, namely by 'borrowing' it from another user with whom he colludes (and who legitimately obtained the credential from an issuing organisation). This type of 'forgery' is not captured by Game 2. In some applications credential sharing is not a concern, while forgery is.

Game 3, credential non-transferability, captures the case of credential sharing between users. In a system that offers credential non-transferability, no user can successfully show a credential of a type he was never issued. This holds even in the case the user colludes with other users that have been issued credentials of that type.

It is interesting to observe the relationship between the notions of unforgeability and non-transferability: the latter, being stronger, implies the former. Clearly, if a dishonest user can construct credentials by himself, there is no need for him to collude with other users in order to forge one. In the model, this is simply reflected by the fact that the adversary is more restricted in his choice of the credential type in the (second phase of) Game 2 than he is in the (second phase of) Game 3. A system that offers non-transferability also offers unforgeability.

This relationship between unforgeability and non-transferability motivates the following definition of a sound pseudonym system.

Definition 3 *A pseudonym system is said to be sound if it offers pseudonym owner protection and credential non-transferability.*

As a side comment, note that non-transferability of credentials is probably the most challenging property for a pseudonym system to achieve. How can colluding users be prevented from sharing their credentials? Certainly, if two users share all their secrets, then they can act as each other in all circumstances. Thus, one will always have to assume that users will not share *all* their secrets, either because they will be prevented by some means, e.g. by the use of tamper-resistant hardware, or because they will be given a sufficiently strong incentive not to. Examples of schemes that follow the latter strategy include the ones in [7], where sharing credentials implies sharing a highly valued key (this is called 'PKI-assured non-transferability'), and [3], where sharing one credential implies sharing all credentials (this is called 'all-or-nothing non-transferability').

2.4 Unlinkability of pseudonyms

We now define Game 4 in order to capture the first privacy property required of pseudonym systems, i.e. the property of pseudonym unlinkability. A second (weaker) privacy property is defined in section 2.6.

In the first phase of the Game 4, $\mathcal{A}$ is allowed to issue queries from the following set of query types, which are similar but not identical to the first three query types of section 2.2.

`runpeprot`(o)**:** $\mathcal{A}$ may arbitrarily select an organisation $o \in (I \cup V)$ and issue this query. When this happens, the Challenger selects a user u according to a probability distribution $\mathcal{D}$ from U and makes u and o execute $\texttt{peprot}_{u,o,p}$. He replies `true` if the protocol execution is successful and `false` otherwise. (If the execution is successful, $\mathcal{A}$ knows that u and o have established a new pseudonym $p \in P$ but learns neither p nor the identity of its owner.)

`runciprot`(p, i, t)**:** $\mathcal{A}$ may arbitrarily select a pseudonym/issuer pair $(p, i) \in P \times I$ and a credential type $t \in T_i$ and issue this query. When this happens, the Challenger selects the owner of p and makes him execute $\texttt{ciprot}_{i,p,t}$ with i. He replies `true` if the protocol execution is successful and `false` otherwise (including the case where p has no owner). Note that $\mathcal{A}$ does not learn who the owner of p is.

`runcsprot`(p, v, t)**:** $\mathcal{A}$ may arbitrarily select a pseudonym/verifier pair $(p, v) \in P \times V$ and a credential type $t \in T$ and issue this query. When this happens, the Challenger selects the owner of p and makes him execute $\texttt{csprot}_{v,p,t}$ with v. He replies `true` if the protocol execution is successful and `false` otherwise (including the case where p has no owner). Note that $\mathcal{A}$ does not learn who the owner of p is.

`corruptUser`(u)**:** As in section 2.2.

`corruptIssuer`(i)**:** As in section 2.2.

`corruptVerifier`(u)**:** As in section 2.2.

We now describe the second phase of the Game 4. We denote the set of pseudonyms that belong to uncorrupted users by $P^{**} \stackrel{def}{=} P^* - \bigcup_{u \in \hat{U}} P_u$.

Game 4 *(pseudonym unlinkability): $\mathcal{A}$ outputs two distinct pseudonyms $p_1, p_2 \in P^{**}$. We say that $\mathcal{A}$ wins the game iff $f(p_1) = f(p_2)$.*

$\mathcal{A}$ may apply a variety of strategies in his effort to correlate pseudonyms. We now consider what is probably the most naive strategy and arrive at the following simple result.

Lemma 1 *If the Challenger, during* `runpeprot`(o) *queries of an instance of Game 4, selects users uniformly at random (i.e. $\mathcal{D}$ is the uniform distribution), and two pseudonyms, p_1, p_2 say, are chosen at random from P^{**}, then the probability that $f(p_1) = f(p_2)$ is $1/|U - \hat{U}|$.*

Proof Suppose $f(p_1) = u \in (U - \hat{U})$. Then the probability that $f(p_2) = u$ is $1/|U - \hat{U}|$, since the pseudonyms are allocated uniformly at random to users, and hence also to uncorrupted users. The result follows. □

Thus it is tempting to define a pseudonym system that offers unlinkability of pseudonyms as a system where $\mathcal{A}$ cannot win the Game 4 with probability greater than $1/|U - \hat{U}| + \epsilon(k)$ for any negligible function ϵ. However, this is only a reasonable definition of unlinkability if $\mathcal{D}$ is the uniform distribution and if no credentials are shown during the first phase of the game, i.e. there are no instances of `runcsprot`. Any instance of `runcsprot` potentially provides the adversary with information about possible links between pseudonyms, and hence potentially increases the adversary's probability of success in linking pseudonyms. Thus, the definition of pseudonym unlinkability needs to take this additional information into account.

Assuming a sound pseudonym system, there are two types of deduction that can be made.

- Suppose a `runcsprot` invocation, say `runcsprot`(p, v, t) for some p, v and t, issued at time τ, returns `true`. Then $\mathcal{A}$ can deduce that there exists some $p' \in \bigcup_{u \in U} P_{u,t}(\tau)$ such that $f(p) = f(p')$.
- Suppose a `runcsprot` invocation, say `runcsprot`(p, v, t) for some p, v and t, issued at time τ, returns `false`. Then $\mathcal{A}$ can deduce that $f(p) \neq f(p')$ for all $p' \in \bigcup_{u \in U} P_{u,t}(\tau)$.

In any instance of Game 4, which in its first phase will involve a series of queries, $\mathcal{A}$ will be able to make a series of deductions about matchings of pseudonyms based on the outcomes ($\{$`true,false`$\}$) of `runcsprot` queries (as above). As a result, for each pair of distinct pseudonyms p_1, $p_2 \in P^{**}$, $\mathcal{A}$ will be able to compute the probability P_{p_1,p_2} that $f(p_1) = f(p_2)$ based on these observations (assuming that $\mathcal{A}$ makes optimal use of the information provided). In computing P_{p_1,p_2} we suppose that $\mathcal{A}$ also takes into account the probability distribution $\mathcal{D}$ used by the Challenger to select the user during `runpeprot` queries.

We now define $\bar{P}$ to be the maximum of these probabilities, i.e.

$$\bar{P} \stackrel{def}{=} \max_{\substack{p_1,p_2 \in P^{**} \\ p_1 \neq p_2}} (P_{p_1,p_2}).$$

We can now define the notion of pseudonym unlinkability.

Definition 4 *A sound pseudonym system is said to offer pseudonym unlinkability iff no $\mathcal{A}$ can win Game 4 with probability greater than $\bar{P} + \epsilon(k)$ for any negligible function ϵ.*

An example scenario of how the two types of deduction might be applied in order to calculate $\bar{P}$, is given in the Appendix.

2.5 Discussion

In real life, colluding organisations could come up with many more effective strategies in order to correlate pseudonyms. Examples include attacks that take into account information such as the time or the geographical location of events that occur in the system. These attacks, however, are not captured by the model, simply because they lie at a different level of abstraction. Protection against, say, timing attacks, de-anonymising traffic analysis or social engineering, is required irrespectively of which particular pseudonym system is being used. The only adversarial strategies to correlate pseudonyms that are inherent in the system, and therefore lie at the same level of abstraction, are the following.

1. If some user is asked for but fails to produce a credential of a given type, the colluding organisations know that none of the pseudonyms on which a credential of that type was previously issued belongs to that user.
2. If some user successfully shows a credential of a given type on one of his pseudonyms, the colluding organisations know that at least one of the pseudonyms on which a credential of that type was previously issued belongs to that user.

These strategies are captured by the probability bound $\bar{P}$. A pseudonym system cannot protect against these strategies without breaching one of its essential properties: that of credential non-transferability. In other words, if a (sound) pseudonym system satisfies Definition 4, this means that the probability that pseudonyms can be successfully linked does not exceed the given bound (by a non-negligible quantity), provided that no 'out-of-scope' attacks place.

2.6 Indistinguishability of pseudonyms

We now establish our second privacy property, namely the notion of indistinguishability of pseudonyms and show that it is a necessary condition for pseudonym unlinkability.

Consider the following game between a Challenger and a polynomial time (and space) adversary $\mathcal{A}$. First, the Challenger chooses sets of users U, issuers I, and verifiers V, a sound pseudonym system and a security parameter k. On input U, I, V, k, he runs `init` and gives the set U of users to $\mathcal{A}$. $\mathcal{A}$ then chooses two users $u_0, u_1 \in U$ and gives them to the Challenger. The Challenger now flips an unbiased random bit $b \in \{0, 1\}$ and makes u_b execute $\texttt{peprot}_{u,o,p}$ with some organisation $o \in (I \cup V)$. He then gives o's private information (including the protocol view and the resulting pseudonym p) to $\mathcal{A}$.

Game 5 *(pseudonym indistinguishability): $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$. We say that $\mathcal{A}$ wins the game iff $b' = b$.*

Definition 5 *A pseudonym system is said to offer indistinguishability of pseudonyms iff no adversary $\mathcal{A}$ can win the above game with probability $\mathbf{Pr} > 1/2 + \epsilon(k)$, for any negligible function ϵ.*

Theorem 1 *If a sound pseudonym system offers pseudonym unlinkability then it also offers pseudonym indistinguishability.*

Proof Suppose the converse, i.e. suppose the pseudonym system offers pseudonym unlinkability but does not offer pseudonym indistinguishability. Given $\mathcal{A}^{\mathtt{i}}$, an adversary that breaks pseudonym indistinguishability, we construct $\mathcal{A}^{\mathtt{u}}$, an adversary that breaks pseudonym unlinkability, as follows. While playing Game 4 (unlinkability) with the Challenger, $\mathcal{A}^{\mathtt{u}}$ plays the role of the Challenger in Game 5 (indistinguishability) with $\mathcal{A}^{\mathtt{i}}$.

Choose a negligible function ϵ. Let $\mu(k) = \sqrt{\epsilon(k)/2}$, which, by definition, is also negligible. In Game 4, $\mathcal{A}^{\mathtt{u}}$ corrupts all but two users, say u_0 and u_1, and one organisation, say o, i.e. $(U - \hat{U}) = \{u_0, u_1\}$ and $\hat{I} \cup \hat{V} = \{o\}$. Then $\mathcal{A}^{\mathtt{u}}$ issues $\texttt{runpeprot}(o)$ queries until three pseudonyms, say p_1, p_2 and p_3, are established between o and $\{u_0, u_1\}$. $\mathcal{A}^{\mathtt{u}}$ does not issue any other queries.

$\mathcal{A}^{\mathtt{u}}$ then plays three instances of Game 5 (indistinguishability) with $\mathcal{A}^{\mathtt{i}}$; in all these games he defines the set of users to be $U = \{u_0, u_1\}$ and the collection of organisations to be $(I \cup V) = \{o\}$. $\mathcal{A}^{\mathtt{u}}$ will use $\mathcal{A}^{\mathtt{i}}$'s ability to win instances of Game 5 in order to win, with non-negligible probability, the instance of Game 4. To this end, in the first instance of Game 5, he gives the pseudonym p_1 together with o's private information and corresponding `peprot` view to $\mathcal{A}^{\mathtt{i}}$. Similarly, in the second and third instances he gives p_2 and p_3 respectively (together with o's private information and corresponding `peprot` views) to $\mathcal{A}^{\mathtt{i}}$. Denote $\mathcal{A}^{\mathtt{i}}$'s output occurring in the three instances of Game 5 by b_1, b_2 and b_3 respectively. Now, since we have assumed that $\mathcal{A}^{\mathtt{i}}$ breaks pseudonym indistinguishability, we suppose that $\mathcal{A}^{\mathtt{i}}$ wins all instances of Game 5 with probability $1/2 + \delta(k)$, where $\delta(k) > \mu(k)$ for all sufficiently large k.

$\mathcal{A}^{\mathtt{u}}$ then selects $j, j' \in \{1, 2, 3\}$, $j \neq j'$, such that $b_j = b_{j'}$, where the pair (j, j') exists by the pigeonhole principle, and outputs $(p_j, p_{j'})$. Now, since $b_j = b_{j'}$ and $f(p_j), f(p_{j'}) \in \{u_0, u_1\}$, we know that $f(p_j) = f(p_{j'})$ if either ($f(p_j) = u_{b_j}$ and $f(p_{j'}) = u_{b_j}$) or ($f(p_j) \neq u_{b_j}$ and $f(p_{j'}) \neq u_{b_j}$). Hence:

$$\begin{aligned}
\mathbf{Pr}(f(p_j) = f(p_{j'})) &= \mathbf{Pr}(f(p_j) = u_{b_j}) \cdot \mathbf{Pr}(f(p_{j'}) = u_{b_j}) \\
&\quad + \mathbf{Pr}(f(p_j) \neq u_{b_j}) \cdot \mathbf{Pr}(f(p_{j'}) \neq u_{b_j}) \\
&= (1/2 + \delta(k))^2 + (1/2 - \delta(k))^2 \\
&= 1/2 + 2\delta(k)^2 \\
&> 1/2 + 2\mu(k)^2 \text{ (for all sufficiently large } k) \\
&= 1/2 + \epsilon(k)
\end{aligned}$$

where ϵ was assumed to be negligible. Thus $\mathcal{A}^{\mathrm{u}}$ breaks unlinkability, contradicting our assumption, and the result follows. □

2.7 Anonymity of users

Consider a sound pseudonym system that offers pseudonym unlinkability. The owner $u \in (U - \hat{U})$ of pseudonym p ($u = f(p)$) is hidden in the anonymity set $U - \hat{U}$ because, from $\mathcal{A}$'s point of view, any user in that set could potentially be the owner of p. The effective size of the anonymity set, however, depends on the probability distribution $\mathcal{D}$ according to which users are selected during pseudonym establishment. Using the information-theoretic anonymity metric of [8, 9], this is given by $-\Sigma_{p \in P^{**}} \mathbf{Pr}(f(p) = u) \log_2[\mathbf{Pr}(f(p) = u)]$ and is maximised if $\mathcal{D}$ is the uniform distribution. In this case the effective size of the anonymity set for all pseudonyms is $\log_2 |U - \hat{U}|$. It is worth observing that, in the general case, it makes sense to consider the anonymity of the user *while acting using a particular pseudonym*. In other words, it is likely that the anonymity a user enjoys will depend on the pseudonym under which he is acting.

The above measure of anonymity only applies to a naive adversary; it only takes into account the *a priori* knowledge (i.e. the distribution $\mathcal{D}$). After observing the system for some time, in the sense of Game 4, $\mathcal{A}$ may decrease the unlinkability between pseudonyms. This decrease in unlinkability yields an *a posteriori* probability distribution $\mathcal{D}'$, that $\mathcal{A}$ is able to construct using deductions that he can make due to the scheme's soundness. While it is the distribution $\mathcal{D}'$ that defines the (effective) size of the anonymity set in which users are hidden (while acting under one of their pseudonyms), this does not necessarily mean that a reduction in unlinkability implies a reduction in anonymity in the theoretical definition of the term. Of course, in practice, any linking of pseudonyms is likely to lead to an increased risk of loss of anonymity because of 'out of scope' attacks. As a result, unlinkability is a property of great importance in its own right.

3 Future work and concluding remarks

In this paper we have introduced a complexity theoretic model for anonymous credential systems. We have formally defined the notions of pseudonym owner protection, credential unforgeability, credential non-transferability and pseudonym unlinkability. A key challenge is thus to construct scheme(s) that meet the definitions in this model, and/or to prove, under appropriate assumptions, the security of existing ones. There is, however, room to refine and extend the model itself; determining the probability $\bar{P}$ by which colluding organisations should be bound when trying to correlate pseudonyms, given a specific history of events in the system, is clearly of importance. Naive strategies for computing $\bar{P}$ appear to be of exponential complexity. Hence, incorporating efficient strategies for computing, approximating or bounding $\bar{P}$ into the model is a desirable refinement. It is envisaged that a refined version of the model described above will combine complexity theory and probability theory in order to describe the resulting degrees of unlinkability and anonymity using recently proposed information theoretic metrics [8, 9]. This should provide further insight into the inherent limits of unlinkability and anonymity in credential systems. We believe that this will also provide insight as to what such systems have to achieve in order not to be considered the weakest link with respect to the overall system of which they form part. An extended version of the model could capture additional properties of pseudonym systems, for example credentials that can be shown only a limited number of times, or a capability for anonymity revocation.

Another direction for future research is the analysis of real-world distributions $\mathcal{D}$ of pseudonym-to-user mappings. This might lead to the description of strategies that users might follow, in a realistic setting, in order to maximise the unlinkability of their pseudonyms. Given the statistical properties of the context, this could also lead to descriptions of how long any given pseudonym can be kept before it should be renewed (if the context allows for this).

Acknowledgments

We would like to thank Sattam Al-Riyami, Alex Dent, Caroline Kudla and Kenny Paterson for their helpful comments on earlier versions of this paper.

Appendix

The following example scenario illustrates how the adversarial strategies are captured by the probability bound $\bar{P}$. For the sake of simplicity, in the example are only one issuer which issues only two types of credential, one verifier and three users. It is assumed that, during the first phase of Game 4 (unlinkability), the adversary corrupts all parties except for the three users, i.e. $I = \hat{I} = \{i\}$, $V = \hat{V} = \{v\}$, $U = \{u_1, u_2, u_3\}$, $\hat{U} = \emptyset$ and $T^* = T_i = \{t_1, t_2\}$.

Table 1 depicts the queries that $\mathcal{A}$ issues in this example scenario. From the first `runcsprot` query, $\mathcal{A}$ can deduce that $f(p_4) = f(p_1)$ or $f(p_4) = f(p_2)$ or $f(p_4) = f(p_3)$. From the second `runcsprot` query, $\mathcal{A}$ can deduce that $f(p_5) \neq f(p_1)$ and $f(p_5) \neq f(p_2)$ and $f(p_5) \neq f(p_3)$. From the third `runcsprot` query, $\mathcal{A}$ can deduce that $f(p_6) \neq f(p_1)$ and $f(p_6) \neq f(p_2)$ and $f(p_6) \neq f(p_3)$.

Table 1: Example scenario: `runpeprot` queries that returned `true`.

Time	*Query type*	*Org*	*Pseudonym*	*Type*	*Outcome*
1	`runpeprot`	i	p_1	n/a	`true`
2	`runpeprot`	i	p_2	n/a	`true`
3	`runpeprot`	i	p_3	n/a	`true`
4	`runpeprot`	v	p_4	n/a	`true`
5	`runpeprot`	v	p_5	n/a	`true`
6	`runpeprot`	v	p_6	n/a	`true`
7	`runciprot`	i	p_1	t_1	`true`
8	`runciprot`	i	p_1	t_2	`true`
9	`runciprot`	i	p_2	t_1	`true`
10	`runciprot`	i	p_2	t_2	`true`
11	`runciprot`	i	p_3	t_1	`true`
12	`runcsprot`	v	p_4	t_1	`true`
13	`runcsprot`	v	p_5	t_1	`false`
14	`runcsprot`	v	p_6	t_1	`false`

Combining the three `runcsprot` queries, $\mathcal{A}$ can deduce, with certainty, that $f(p_4) \neq f(p_5)$ and that $f(p_4) \neq f(p_6)$. It follows that p_5 and p_6 must belong to the two users $\{u_1, u_2, u_3\} - \{f(p_4)\}$. So, the probability P_{p_5,p_6} that $f(p_5) = f(p_6)$ is $1/2$. This happens to be the maximum over all distinct pseudonym pairs and thus, in the example, $\bar{P} = 1/2$. In other words, if $\mathcal{A}$, at the end of the game, outputs (p_5, p_6), he has a 50% chance of winning the game. If a (sound) pseudonym system offers pseudonym unlinkability, then no $\mathcal{A}$ should be able to break this bound by a non-negligible quantity.

References

[1] M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. Stinson, editor, *Advances in Cryptology — Crypto 93 Proceedings*, volume 773 of *Lecture Notes in Computer*

Science, pages 232–249, 1994.

[2] Stefan Brands. *Rethinking Public Key Infrastructures and Digital Certificates — Building in Privacy*. The MIT Press, Cambridge, Massachusetts, 2000.

[3] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceedings*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer Verlag, Berlin, 2001.

[4] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.

[5] D. Chaum and J.-H. Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. In A. M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, number 263 in Lecture Notes in Computer Science, pages 118–167. Springer Verlag, Berlin, 1987.

[6] I.B. Damgard. Payment systems and credential mechanisms with provable security against abuse by individuals. In S. Goldwasser, editor, *Advances in Cryptology — CRYPTO '88: Proceedings*, number 403 in Lecture Notes in Computer Science, pages 328–335. Springer Verlag, 1990.

[7] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In H. M. Heys and C. M. Adams, editors, *Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, Kingston, Ontario, Canada, August 9-10, 1999, Proceedings*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199. Springer Verlag, Berlin, 2000.

[8] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In R. Dingledine and P. F. Syverson, editors, *Privacy Enhancing Technologies, Second International Workshop, PET 2002, San Francisco, CA, USA, April 14-15, 2002, Revised Papers*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer-Verlag, Berlin, 2002.

[9] S. Steinbrecher and S. Koepsell. Modelling unlinkability. In R. Dingledine, editor, *Privacy Enhancing Technologies, Third International Workshop, PET 2003, Dresden, Germany, March 26-28, 2003, Revised Papers*, volume 2760 of *Lecture Notes in Computer Science*, pages 32–47. Springer-Verlag, Berlin, 2003.