A Calculus for Privacy-friendly Authentication

Patrik Bichsel
IBM Research – Zurich
pbi@zurich.ibm.com

Jan Camenisch
IBM Research – Zurich
jca@zurich.ibm.com

Dieter Sommer
IBM Research – Zurich
Technische Universität Darmstadt,
FB Informatik/FG SIT
dso@zurich.ibm.com

ABSTRACT

Establishing authentic channels has become a common operation on the Internet and electronic commerce would not be possible without it. Because traditionally authentication is based on identifying users, the success of electronic commerce causes rapid erosion of their privacy. Privacy-friendly authentication, such as group signatures or anonymous credential systems, could mitigate this issue minimizing the information released during an authentication operation. Unfortunately, privacy-friendly authentication systems are not yet deployed. One reason is their sophistication and feature richness, which is complicating their understanding. By providing a calculus for analyzing and comparing the requirements and goals of privacy-friendly authentication systems, we contribute to a better understanding of such technologies. Our calculus extends the one by Maurer and Schmid [18], by introducing: (1) pseudonyms to enable pseudonymous authentication, (2) a pseudonym annotation function denoting the information an entity reveals about itself, and (3) event-based channel conditions to model conditional release of information used for privacy-friendly accountability.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Communication/Networking and Information Technology—*Network-level security and protection*

General Terms

Security, Design, Algorithms

Keywords

Authentication, accountability, privacy, security, secure channel modeling, anonymous credential systems

1. INTRODUCTION

Electronic communication networks such as the Internet have an enormous merit when it comes to the ease of distributing information. However, the lack of physical presence requires the communication partners to establish mutual trust using mechanisms such as authentication. Today, most service providers use a simple authentication approach in which a user shows knowledge of a username/password combination. The use of such simple authentication mechanisms poses severe security risks [15, 23]. Furthermore, the fact that service providers require users to release excessive amounts of (user-provided) personal information upon their first visit erodes privacy. Authentication mechanisms based on technologies such as anonymous credential systems, originally proposed by Chaum [12], provide strong authentication while requiring a user to disclose only the minimal information necessary in a specific context. The drawback of such technologies is that their complexity makes them hard to understand, explain, and compare with traditional approaches. This seems to be an important factor hindering practical deployment so far.

In this paper, we provide a calculus for describing the establishment of secure, i.e., authentic and confidential, channels. We focus on the properties that are particularly important to privacy-friendly authentication. We envision that the improved understanding will contribute in convincing decision makers to adopt privacy-enhancing technology. As a basis we use the model and channel derivation calculus proposed by Maurer and Schmid [18, 19]. Their calculus analyzes the functionality provided by standard cryptographic primitives, i.e., their requirements and security properties when bootstrapping a secure channel. We extend the Maurer-Schmid calculus to model privacy-friendly authentication and accountability. More concretely, we model *pseudonymous authentication*, *attribute-based statements*, and *conditional release* of information.

First, pseudonymous authentication is a basic concept in privacy-friendly authentication. It allows a user to be known to her communication partner only by a pseudonym instead of her unique identity. Consequently, a user can have several unlinkable connections to the same party allowing her to separate different contexts at her discretion. Second, attribute-based statements enable a user to release attributes selectively or even to reveal only a statement about an attribute and thereby fulfill the authentication requirements in a privacy-optimal manner. As an example, consider a liquor store, which is required to verify the age of its customers by regulation. Through attribute-based statements the store can verify the age of customers without requesting any further information. Conversely, in today's practice, the store would request the date-of-birth attribute as contained in an appropriate credential such as an identity card. Third, conditional release of information to a third party is a feature enabling privacy-friendly accountability. It ensures that attributes become available under well-defined circumstances, such as a user abusing the terms and conditions, to designated parties. Conditional release of information can also help in attaining better privacy in general business processes assuming the existence

SACMAT'12, June 20–22, 2012, Newark, New Jersey, USA.

of a mutually trusted party. For instance, when buying a book, a user could release the payment information only to her bank who uses it to bill her. While the service provider can make sure that the user provides the appropriate payment information to her bank, it does not learn this information. Similarly, the shipping information can only be released to the delivery company, which allows the user to have a pseudonymous connection to the book seller itself. Such processes would relieve service providers from knowledge of sensitive personal information, mitigating the risks associated with possession of the latter.

Related Work.

In the area of modeling security of authentication and communication channels, numerous recent papers are available. Typically, they focus on formally verifying security properties of protocols in an automated fashion. Backes, Maffei, and Unruh [3] have integrated zero-knowledge proofs, a major building block of privacy-friendly authentication, into an automated verification tool. Mödersheim and Vigano [20] have later put forth a formal model of pseudonymous channels. Following their approach of modelling pseudonyms, we could aim at a more formal model of attribute-based statements and conditional release of information. However, we want to focus on the intuition and consider a rigorous formalization to be an interesting future contribution. Notable work addressing pseudonymous authentication channels that provides a formal model and tool-based verification of a subset of the idemix protocols has been published by Camenisch, Mödersheim, and Sommer [10].

Regarding semi-formal models of authentication, Maurer and Schmid [19] have introduced a simple, yet expressive, notation allowing for analyzing and comparing protocols that establish secure channels based on standard cryptographic technologies available at the time. This model represents the starting point for various recent formal approaches towards modeling cryptographic functionality [16, 17]. In our paper, we have the same goals as the original model, but for substantially more complex protocols, the properties of which are harder to grasp and only understood by a small group of privacy or cryptography researchers. In contrast to the aforementioned proposals [3, 10, 20] that use more complex and less intuitive notation for achieving their protocol specification and automated verification goals, our model extends the intuitive notation of the model of Maurer and Schmid while retaining its basic concepts and simplicity. Our model can, like Maurer and Schmid's, be used for comparing and analyzing security properties, especially for today's authentication protocols. In addition, our calculus can act as a teaching model for the goals and properties of complex protocols and thereby contribute to a wider understanding of privacy-friendly authentication and accountability technologies and their future deployment. Therefore, our work closes the gap in the space of semi-formal models of expressing cryptographic schemes for privacy-friendly authentication with an intuitive yet formalized method. In this respect our work is orthogonal to the results in the space of formal protocol verification that have been presented before.

Our extension for attribute-based authentication requires authentication properties to be expressed in a suitable language. Sommer [22] presents a logic-based requirements language and dual specification language for attribute-based authentication supporting advanced schemes such as the idemix anonymous credential system. A closely-related language for specifying attribute-based authentication requirements has been put forth by Camenisch et al. [9]. Both contributions allow for combining anonymity of transactions with user accountability based on the ideas originally put forth by Backes et al. [2].

Structure of this Paper.

We start in Section 2 by introducing the main concepts of privacy-friendly authentication. In addition, we provide a brief overview of the channel calculus proposed by Maurer and Schmid [18]. In Section 3 we introduce our extensions to Maurer and Schmid's model. In the same section we extend the set of channel derivation rules to accommodate our extensions. We show in Section 4 how our extended model applies to the examples of standard X.509 certificates, the privacy-enhanced idemix authentication protocols, and the general example of privacy-friendly accountability. Finally, we conclude with a discussion of the merits of our model and future directions in Section 5.

2. PRELIMINARIES

In this section we introduce the most relevant aspects of privacy-friendly authentication and accountability. In addition, we discuss the Maurer-Schmid model that we extend such that it allows for modeling privacy-friendly authentication and accountability.

2.1 Privacy-friendly Authentication

In recent years, numerous cryptographic systems that allow for the protection of a user's privacy have been proposed [4, 5, 6, 8]. An important feature thereby is the ability of a user to authenticate pseudonymously. As outlined in the introduction, we may use privacy-enhanced protocols for fine-grained control of information to be released. Namely, when a user orders a book, the delivery information is only needed by the delivery company and the payment information is only used by the user's bank. However, the entity requesting the authentication (in our example the book shop) has a legitimate interest in getting security assurances. Group signature schemes [4, 5, 13] or anonymous credential systems [6, 8, 12] are examples of technologies that achieve strong authentication guarantees in combination with substantial privacy guarantees. We will focus on anonymous credential systems as they provide the most general set of features.

Anonymous Credential Systems.

Anonymous credential systems allow a user to obtain a certification of attributes from an issuing party, called the *issuer* or *identity provider*. The attributes can be arbitrarily chosen and may include identity attributes (e.g., name, date of birth) or access rights. We refer to the set of certified attributes as the *credential* and to the protocol for obtaining a credential as the *issuing protocol*. Note that privacy protection can be achieved as the issuer does not necessarily learn all the information contained in a credential, i.e., he may not learn all attribute values. After a user has obtained a credential, she can use it to selectively reveal the certified attributes or prove statements about those attributes. As the recipient of such proof usually offers a service in exchange for the proof, we name it a *service provider*, *verifier*, or *relying party* and we call the protocol for proving statements over attributes the *proving protocol*. A main merit of anonymous credentials is that a proof can be done anonymously or pseudonymously, i.e., it does not leak any information that can be linked to the issuing protocol. Some systems, such as the Identity Mixer (idemix) library [21], even allow a user to unlinkably issue proofs based on one credential any number of times. Furthermore, a user can release any subset of the certified information. For instance, a user with an anonymous credential containing her name, address, and birth date, may use this credential to prove that she is older than 21 years without revealing any further information.

In our model we distinguish three components offered by anonymous credential systems that improve privacy: pseudonymous

authentication, attribute-based information disclosure, and conditional attribute release.

Pseudonymous Authentication. Systems such as idemix enable a user to choose a pseudonym when authenticating. Whenever a user authenticates using the same pseudonym, the service provider may link the information related to the different transactions. In addition, a service provider may require that one user can only have one pseudonym for a specific domain.

Attribute-based Information Disclosure. While selectively revealing attributes already provides an improvement over standard certification technology w.r.t. the privacy of a user, proving statements about attributes goes even further. Anonymous credentials allow a user to only reveal a statement about attributes instead of the attribute value itself. Such statements include equalities among attributes, or inequalities between attributes or constants.

Conditional Attribute Release. To prevent abuse of the privacy granted to users, service providers may want to ensure that users are accountable for their actions. For example, if a user wants to rent sports equipment, the rental agency does not need any information about the user. Still, in case the user damages or does not return the equipment, the agency wants to have the identity or billing information to claim the damages. In such a situation the user could issue a verifiable encryption containing its identity information on behalf of the local government. As the encryption is verifiable, the agency can verify that it indeed contains the identity information of the user as claimed without learning the specific attribute values. Attached to the verifiable encryption would be the condition that decryption is to be done only if the equipment is damaged or has not been returned at all. The agency as well as the user do trust in the local government to decrypt in case the indicated condition is fulfilled, and only in that case. The example shows how verifiable encryption [1, 11] allows a relying party to attain accountable transactions. During authentication, the relying party requires the user to release a verifiable encryption containing the desired attributes. The encryption to a mutually trusted entity can be verified by the relying party. In addition, the parties agree on a condition defining when the message may be decrypted. The relying party trusts the third party to actually decrypt in case the condition is met and the user trusts it to only decrypt in this case. This assures the relying party that it will learn certain attributes if the user does not act as agreed.

2.2 Formal Model of Secure Channels

Maurer and Schmid [18] define a simple and expressive formal framework for comparing security properties of cryptographic protocols. They propose a channel calculus to compare security properties achieved by standard cryptographic primitives. The capabilities of cryptographic protocols are modeled using channel transformations. Maurer and Schmid highlight the capabilities of their framework by modeling the channel transformations that can be implemented with symmetric-key encryption, message authentication codes, public-key encryption or digital signature schemes. As a further capability, their model simplifies reasoning about trust relations and the transformations enabled through trusted entities. This enables, e.g., the expression of a public key infrastructure in their model.

Maurer and Schmid model two security properties called *authentication* and *confidentiality*. Let us summarize their authentication definition using entities A and B. Informally, if party A is authenticated to party B, the latter is assured that it actually communicates with party A. In other words, if B is convinced that it communicates with a well-defined, unique party A, then there cannot be a party A' that fakes messages to look as if A had sent them. The confidentiality property is dual to the authentication property, thus, a party B knows that her messages can only be read by a party A (and not A') if it has a confidential channel to A.

$$A \longrightarrow B \quad (1) \qquad A \longrightarrow\!\bullet\, B \quad (3)$$

$$A \bullet\!\longrightarrow B \quad (2) \qquad A \bullet\!\longrightarrow\!\bullet\, B \quad (4)$$

Maurer and Schmid use the notation (1) for an insecure channel from A to B, (2) for an authentic channel where A is authentically known to B, and (3) for a confidential channel where A is sure that its messages can be only read by B. A secure channel fulfills the authentication and confidentiality properties of the respective channel endpoints and is denoted as in (4). In their notation a bullet denotes a security property, i.e., either authentication or confidentiality of the respective channel endpoint.

When it comes to channel transformations achieved by using cryptographic protocols, the time at which a channel is available is of importance. Thus, the model defines a channel over which a message, fixed or chosen at time t_1, can be sent at time t_2 to be denoted as $A \bullet\!\xrightarrow{t_2[t_1]} B$, where $t_2 > t_1$ must hold. For example, (5) shows that a message can only be forwarded from a party A to C if the time t_3 at which the relaying party B can choose its message is after the time it has received the original message from party A.

$$A \xrightarrow{t_2[t_1]} B,\ B \xrightarrow{t_4[t_3]} C,\ t_3 > t_2 \implies A \xrightarrow{t_4[t_1]} C \qquad (5)$$

Maurer and Schmid conclude that, using the basic cryptographic primitives they discuss, a security property (i.e., a bullet) at one end of a channel can be re-established at time t_2 using an insecure channel given a bullet on the same side of the channel at time t_1, given $t_2 > t_1$. However, they state that two things cannot be achieved through cryptographic protocols: (1) bullets cannot be created, and (2) bullets cannot be moved from one side of the channel to the other.

3. FORMAL CHANNEL MODEL

We present multiple extensions to Maurer and Schmid's approach of modeling secure channels, which has recently been used for more formal treatments of cryptographic methods [16, 17]. Our extensions aim at modeling privacy-friendly authentication and accountability. First, we extend the notion of authentication such that a party can have multiple different names, denoted as *pseudonyms*. This enhancement accounts for the fact that cryptographic schemes allow a user to authenticate pseudonymously. Second, we enable parties to make *statements about their attributes*. We use those statements to model attribute-based authentication, where the service provider merely learns attributes or predicates about attributes. This allows us to model situations where, e.g., a party presents a statement derived from anonymous credentials. Finally, we use *generic conditions* instead of time semantics to denote when a message has to be chosen by the sender and when a message is sent. Using this generalization allows us to model channels established trough events, which build the basis for privacy-friendly accountability. We now present the extensions to the model of Maurer and Schmid in detail and provide the definitions we build upon.

3.1 Extensions to the Maurer-Schmid Model

We start with presenting the foundations of our model and put it in context with the approach taken by Maurer and Schmid in their work. As Maurer and Schmid, we use channels to model that parties may exchange information and we use a bullet to annotate a security assurance. More concretely, a bullet at the source of

a channel denotes an authenticated communication partner and a bullet at the destination stands for a confidential channel. A channel without bullet annotations does not have any security assurances and is called an insecure channel.

3.1.1 Pseudonyms

In the original Maurer-Schmid model, a party is assumed to have a unique, system-wide identifier. Technology-wise, such an identifier can, e.g., be implemented by the unique public key of the party in a system where each party has exactly one public key. In each authentic or confidential channel, the party with the security annotation (i.e., the bullet) is known to its communication partner by this unique identifier. This is a core property of the model, based on which channels can be composed to obtain a target channel. A major drawback of this modeling approach is that it cannot reflect the capabilities of today's privacy-enhanced authentication technologies. We overcome this limitation by allowing parties to have and act under multiple pseudonyms. Therefore, we define channels to connect two pseudonyms instead of the parties themselves. Intuitively, a cryptographic pseudonym can be seen as the equivalent of a public key in that it (provably) can be related to a secret key. However, a party A can generate an arbitrary number of pseudonyms using a single secret key. Note that a party knowing a set of pseudonyms (without the corresponding secret information) cannot distinguish whether or not they have been generated using the same user secret. Thus, we denote pseudonyms to be *unlinkable*.

More formally, given a set of user secrets $\mathfrak{S}$ and a set of parties $\mathfrak{P}$, each party $P \in \mathfrak{P}$ is assigned a secret $s_i \in \mathfrak{S}$ using a function $f : \mathfrak{P} \rightarrow \mathfrak{S}$. Note that extending this situation to using several secrets per user is straightforward. Let us assume a function $nym(\cdot,\cdot)$ that takes a user secret and a randomization factor as input and outputs a pseudonym $n_i \in \mathfrak{N}$. First, we assume that pseudonyms are unique, i.e., $(\forall s_1, s_2 \in \mathfrak{S}\ \forall n_1, n_2 : n_1 = nym(s_1,\cdot), n_2 = nym(s_2,\cdot)) : n_1 \neq n_2$. Second, the unlinkability property of pseudonyms n_1, n_2 is defined as follows: Let $\mathfrak{B}_j = \{nym(s_j,\cdot)\}$ be the set of all pseudonyms based on secret s_j, for $j \in \{1,2\}$. Unlinkability of n_1 and n_2 is equivalent to the following cases (1) $n_1 \in \mathfrak{B}_1, n_2 \in \mathfrak{B}_2$ and (2) $n_1, n_2 \in \mathfrak{B}_1$ being (computationally or information-theoretically) indistinguishable. As suggested by the analogy of pseudonyms with public keys, a party P with secret $s_p = f(P)$, and pseudonym $n_i = nym(s_p, r)$ can prove to a communication partner that she is the legitimate owner of n_i (i.e., that she knows the secrets s_p and r corresponding to n_i). Because of the uniqueness of pseudonyms we can define a mapping function $p(\cdot)$ using a pseudonym as input and providing the corresponding party as output. We denote with $P = p(n_i)$ that party P is the holder of pseudonym n_i, i.e., $n_i = nym(s_p,\cdot)$. This mapping function p between parties and their pseudonyms is needed for expressing our channel composition rules. More concretely, we use this function to compose channels with different pseudonyms, where the composition requires the party having generated those pseudonyms being the same. Note that this function is not available to parties within the system since this would invalidate the unlinkability property.

In our channel model we use a more intuitive notation, where we denote a pseudonym of a party A in a communication as $\mathcal{A}_i$ instead of n_i. Note that this notion closely relates to what is denoted as $[A]_i$ by Mödersheim and Vigano [20]. However, our unlinkability property of pseudonyms goes further than their perspective in which pseudonyms model *sender invariance*, where a recipient is assured to be communicating with the same sender (e.g., through the use of an unauthenticated public key). In any practical system, pseudonyms can be realized through cryptographic mechanisms, e.g., using a commitment scheme as in anonymous credential systems [8]. A user may generate a polynomial number of pseudonyms $\mathcal{A}_i$ such that uniqueness of the pseudonyms is attained with overwhelming probability. Depending on the cryptographic scheme, the unlinkability can hold computationally or even information-theoretically.

Note that certain scenarios merit from a party having a unique pseudonym. As an example, a well-known service provider may profit from having only one pseudonym and it does not benefit from the privacy that multiple pseudonyms offer. In such cases we use *public pseudonyms*, i.e., for a party I we would denote the public pseudonym as $\mathcal{I}$, omitting the index.

3.1.2 Authentication and Confidentiality

As we specify channels between pseudonyms that parties act under, and not between parties themselves, we need to appropriately define authentication for our model.

DEFINITION 1 (PSEUDONYM AUTHENTICATION). *An entity A acting under pseudonym $\mathcal{A}_b$ is* pseudonym authenticated *towards an entity B acting under pseudonym $\mathcal{B}_a$ if B is assured that it communicates with the entity legitimately holding pseudonym $\mathcal{A}_b$.*

The intuition behind this definition is aligned with the original model, with the difference that B is assured that it communicates with a party holding the pseudonym $\mathcal{A}_b$ instead of being assured that it communicates with party A known under its unique identifier. The difference articulates in the situation where a party A repeatedly communicates with another entity. In such case, we can see that using the different pseudonyms $\mathcal{A}_i$ and $\mathcal{A}_{\bar{i}}$ allows A to maintain two authenticated but unlinkable communication channels with her communication partner. Consequently, parties are only linkable when using the same pseudonym on several channels. Note that the definition does not touch on information that is released through the channel, in particular, it does not specify attributes that B knows about the pseudonyms, i.e., about the parties holding them.

In the Maurer-Schmid model, the dual property to authentication is confidentiality. In analogy, we introduce the notion of pseudonym confidentiality.

DEFINITION 2 (PSEUDONYM CONFIDENTIALITY). *A channel between an entity A acting under pseudonym $\mathcal{A}_b$ and an entity B acting under pseudonym $\mathcal{B}_a$ is* pseudonym confidential *if A can be ensured that only the party holding pseudonym $\mathcal{B}_a$ has access to the messages sent on this channel.*

Clearly, authentication and confidentiality as modeled by Maurer and Schmid are a special case of our extended model where every party is constrained to one unique, system-wide identifier. How our changes affect the model can be most easily expressed using the examples of the basic channels, i.e., *insecure*, *authenticated*, *confidential*, and *secure* channel.

Insecure Channel.

We start with an insecure channel from A acting under pseudonym $\mathcal{A}_b$ to B acting under pseudonym $\mathcal{B}_a$. We model this similarly to the Maurer-Schmid model, with the difference that not parties but pseudonyms are denoted as communication partners. Thus, we denote such insecure channel as

$$\mathcal{A}_b \longrightarrow \mathcal{B}_a \ . \tag{6}$$

Note that the index of a pseudonym denotes the *intended* communication partner, e.g., $\mathcal{A}_b$ for A communicating with B.

We can look at the channel in two different ways. First, it visualizes the security information (authentication or confidentiality) available to the communicating parties. From this point of view, the entity $A = p(\mathcal{A}_b)$ may be any party in the system. This results from the fact that the pseudonym does not have a security annotation (i.e., a bullet). $\mathcal{A}_b$ here is simply a name used to refer to the *intended channel endpoint*. Party $B = p(\mathcal{B}_a)$ learns only the unauthenticated pseudonym about its communication partner. This is what we define as an insecure channel: similarly to using an unauthenticated public key, the pseudonym does not imply communication with the party legitimately holding the pseudonym. Second, an insecure channel denotes the availability of a channel. For our channel transformations we often use insecure channels between two pseudonyms to denote that the parties holding the pseudonyms have access to a communication channel.

Authentic Channel.

An example of a channel from A, the holder of $\mathcal{A}_b$, to B, the holder of $\mathcal{B}_a$, where the pseudonym $\mathcal{A}_b$ is authenticated is denoted as

$$\mathcal{A}_b \bullet\!\!\longrightarrow \mathcal{B}_a \ . \tag{7}$$

Note that B does not know which party holds the pseudonym $\mathcal{A}_b$. This results from the unlinkability of pseudonyms as well as the fact that parties within the system do not have access to the function p. Party A can send messages authenticated as $\mathcal{A}_b$ to $\mathcal{B}_a$ over this channel where the former does not have any (authentic) information on the pseudonym it sends its messages to. By extension, A does not have any information on the party $B = p(\mathcal{B}_a)$. This is the natural notation of a pseudonym authenticated channel based on the notation of an authenticated channel in the model of Maurer and Schmid where authentication is defined in a more restrictive way through a party authenticating under its system-wide identifier.

Confidential Channel.

We generalize confidential channels similarly to authentic channels. Instead of knowing that the channel is established with an entity specified by a unique identifier, the message recipient of a pseudonym confidential channel is known to be a party holding a specified pseudonym. In an example, we denote a pseudonym confidential channel from a pseudonymous party $\mathcal{A}_b$ to a party B holding $\mathcal{B}_a$ as

$$\mathcal{A}_b \longrightarrow\!\!\bullet\, \mathcal{B}_a \ . \tag{8}$$

In this example, only the pseudonym $\mathcal{B}_a$ comprises an assurance.

Secure Channel.

A secure channel between the pseudonyms $\mathcal{A}_b$ and $\mathcal{B}_a$ assures the parties $A = p(\mathcal{A}_b)$ and $B = p(\mathcal{B}_a)$, holding the pseudonyms $\mathcal{A}_b$ and $\mathcal{B}_a$, that their communication partner is the party holding the denoted pseudonym. We denote a secure channel as

$$\mathcal{A}_b \bullet\!\!\longrightarrow\!\!\bullet\, \mathcal{B}_a \ . \tag{9}$$

Note that we simplify the notation in the remainder of the paper by saying that a pseudonym $\mathcal{A}_b$ having a channel to a pseudonym $\mathcal{B}_a$ as shorthand notation for the party $A = p(\mathcal{A}_b)$, i.e., party A holding pseudonym $\mathcal{A}_b$, having a channel to party $B = p(\mathcal{B}_a)$.

3.1.3 Attribute-based Pseudonym Annotations

For modeling privacy-friendly authentication we not only need to model pseudonymous communication but also the exchange of attributes. This goes well beyond what Maurer and Schmid can express in their model where authentication is a binary property indicated through the bullet and the authentication information remains implicit. Due to the importance of attribute-based authentication in today's information systems and application scenarios, we make an extension to the Maurer-Schmid model to capture this concept.

We implement this by annotating the pseudonyms with a formula ϕ that expresses the attribute statements a party makes.

DEFINITION 3 (ATTRIBUTE-BASED ANNOTATION). *An* attribute-based annotation *of a pseudonym $\mathcal{A}_b$ held by an entity A is defined as the statements ϕ about $\mathcal{A}_b$ being released to the communication partner.*

This definition means that the party B learns attribute statements as defined by ϕ, which are expressed as a logic-based formula. In case the pseudonym is annotated with a bullet, this allows its communication partner to derive that the statements are about a *given* pseudonym, e.g., $\mathcal{A}_b$ in the definition. Without a bullet, the statement is purely a declaration about an (unverified) pseudonym. Consequently, the presence or absence of a bullet annotation of a pseudonym making an attribute-based statement plays the crucial role of defining whether the statement is about the indicated pseudonym or not. Note that the direction of the channel between $\mathcal{A}_b$ and $\mathcal{B}_a$ is orthogonal to the attribute-based pseudonym annotation. That is, we assume the attributes can be learnt by the communication partner even if the channel direction does not suggest so.

Channel Syntax.

To express that $\mathcal{A}_b$ has a pseudonym authenticated channel to $\mathcal{B}_a$ with pseudonym annotation ϕ we use the notation

$$\mathcal{A}_b \overset{\phi}{\bullet\!\!\longrightarrow} \mathcal{B}_a \ . \tag{10}$$

That is, entity $B = p(\mathcal{B}_a)$ is ensured that it has a channel with the party holding pseudonym $\mathcal{A}_b$ and in addition learns attribute information as specified by ϕ about this party. Note that we often annotate the formula to illustrate the pseudonymous entity that is described, e.g., $\phi_{\mathcal{A}_b}$ in case the information is about $\mathcal{A}_b$.

We naturally extend this notion to a secure channel by providing an annotation also at the recipient pseudonym of a channel, the syntax being as follows:

$$\mathcal{A}_b \overset{\phi_{\mathcal{A}_b}}{\bullet\!\!\longrightarrow\!\!\bullet} \overset{\phi_{\mathcal{B}_a}}{\mathcal{B}_a} \ . \tag{11}$$

In addition to the properties of the channel before, this channel ensures that it is pseudonym confidential to $\mathcal{B}_a$ and the party holding $\mathcal{A}_b$ learns the attribute statement $\phi_{\mathcal{B}_a}$. As noted previously, we mean the parties holding the pseudonyms when we talk about pseudonyms exchanging information.

Annotation Formula.

For defining how the attribute annotations relate to the channel transformations, we introduce our approach of expressing the annotation formula ϕ and provide an intuition on how it expresses statements.

The simple approach of expressing attribute statements by modeling them as a set of attribute-value pairs is not powerful enough for expressing data-minimizing statements about parties. Concretely, it is lacking the following features: (1) Revealing partial information about an attribute value, (2) grouping of attributes, and (3) relating attributes without revealing them. Feature (1) is necessary for making data-minimizing statements, e.g., revealing that the date of birth attribute is less than a given reference date to establish a minimum age of a user. Feature (2) can be used to make statements about attributes that conceptually belong together, e.g., about the number and expiration date of a specific credit card of a person. Feature (3) allows for specifying that an attribute of one

attribute collection is in a relation with an attribute of another collection. For instance, the last name of a party's driver's license can be expressed to be the same as the one on its eID card. For realizing those features, we decided to model a pseudonym annotation as a formula ϕ in a logic as explained next.

We start with the basic concept of a *credential* used to group attributes into attribute collections, which was done eralier by Camenisch et al. [9] and Sommer [22], where the attributes can be certified using a suitable technology, or remain uncertified. Suitable technologies for certification are, e.g., X.509 attribute certificates [14], anonymous credential systems such as idemix [21] or U-Prove [7], or identity federation schemes with an online identity provider. A possible example for a credential is one of type Electronic Identity Card (eID Card), issued by the Swiss Government using idemix anonymous credential technology. Such credential could, e.g., comprise the attributes first name, last name, and date of birth of the credential's holder. By referring to the attributes of credential c using the "."-notation, as for example in $\mathsf{c}.a$, we can address attributes of credentials, in this example attribute a of credential c.

To make statements about attributes of credentials, we use *predicates*. A predicate can make a statement about an attribute and a constant or about two attributes. For example, the $\mathit{Eq}(\cdot,\cdot)$-predicate expresses equality between its two arguments, where the arguments may be attributes of credentials or constants. Another example is the $\mathit{Leq}(\cdot,\cdot)$-predicate expressing the relation "less than or equal" between the first and second argument. The predicate

$$\mathit{Eq}(\mathsf{c}.\mathit{dateOfBirth}, \mathsf{1978\text{-}12\text{-}01})$$

expresses, e.g., that the attribute *dateOfBirth* of credential c is equal to the constant date value 1978-12-01.

With privacy-friendly authentication we want to express, e.g., a predicate specifying that the attribute *dateOfBirth* of credential c is less than or equal to the constant date value 1991-06-20 to establish that an entity has an age greater than or equal to 21 years, when being considered on 2012-06-20. This predicate expresses a statement over an attribute, providing less information compared to releasing its value. Such information is sufficient for many scenarios, e.g., where only a minimum age has to be established.

$$\mathit{Leq}(\mathsf{c}.\mathit{dateOfBirth}, \mathsf{1991\text{-}06\text{-}20})$$

We abstract in our syntax from using different predicate terms depending on the argument types, e.g., for expressing equality on strings and integers, but overload those into a single predicate term to simplify the notation without loss of expressiveness. Not all predicates are defined for all argument types due to constraints of the cryptographic proof system of idemix and other protocols. The inequality predicates $\mathit{Leq}(\cdot,\cdot)$, $\mathit{Lt}(\cdot,\cdot)$, $\mathit{Geq}(\cdot,\cdot)$, and $\mathit{Leq}(\cdot,\cdot)$ can by applied to attribute types with a total order (e.g., integers or dates) with their usual semantics. The Eq-predicate is applicable to arguments of any type supported by the underlying technology.

Multiple predicates can be connected with the operators $\wedge$ and $\vee$ as is standard in logic to obtain a sentence or formula ϕ expressing attribute statements:

$$\begin{aligned}\phi_0 = &\mathit{Eq}(\mathsf{c}.\mathit{lastName}, \mathsf{Doe}) \wedge \\ &\mathit{Eq}(\mathsf{c}.\mathit{dateOfBirth}, \mathsf{1978\text{-}12\text{-}01}) \wedge \\ &\mathit{Eq}(\mathsf{c}.\mathit{type}, \mathsf{eID\_Card}) \ .\end{aligned}$$

The formula ϕ_0 expresses values of the attributes of credential c by relating the values with the constants through Eq-predicates. A formula like this can be used to specify the attribute values of a credential of a party as being certified by an identity provider, e.g., for an anonymous credential the party obtains as an eID card.

Our language for expressing ϕ is based on a fragment of the logic of [22] for modeling identity statements with a focus on privacy-preserving identity management through the data minimization features, though we omit multiple features that are not relevant for our model.

As in standard logic we can derive new formulae from an existing formula, e.g., a formula ϕ_1 from a formula ϕ_0, which is denoted as $\phi_0 \vdash \phi_1$. Continuing the example from before, the holder of credential c specified through ϕ_0 can derive a formula ϕ_1 that comprises partial information about the credential's attribute values and is "consistent" with the statements in ϕ_0. Using appropriate technology such as an anonymous credential system, a party can prove this formula correct, i.e., consistent with the issued anonymous credential, to a recipient party.

$$\begin{aligned}\phi_1 = &\mathit{Leq}(\mathsf{c}_1.\mathit{dateOfBirth}, \mathsf{1991\text{-}06\text{-}20}) \wedge \\ &\mathit{Eq}(\mathsf{c}_1.\mathit{type}, \mathsf{eID\_Card})\end{aligned}$$

The credential must be renamed to prevent undesired linkability of formulae, e.g., c is renamed to c_1 in the above example.

3.1.4 Channel Conditions

The timing annotation $t_2[t_1]$ of a channel in the model of Maurer and Schmid has the semantics that the message has to be fixed at time t_1 and can be sent at time t_2 over the channel where $t_2 > t_1$ must always hold. We extend this purely time-based semantics with *event-based semantics* for modeling more general *conditions*. The original timing semantics is a special case of our extended notion. This extension particularly allows for realizing conditional release of data, e.g., to model privacy-friendly accountability. Events are specified through monotone formulae in a generic manner.

We define the function $\tau(c)$ for specifying the time at which the event or event formula c occurs. For event formulae c_1 and c_2, we recursively define for $c = c_1 \wedge c_2$, $\tau(c) = \mathrm{Max}(\tau(c_1), \tau(c_2))$, and for $c = c_1 \vee c_2$, $\tau(c) = \mathrm{Min}(\tau(c_1), \tau(c_2))$. For an atomic event c, $\tau(c) = t$ for a constant time value t from a totally-ordered set that indicates the time the event occurs. This defines the function $\tau(c)$ recursively for all monotone formulae for specifying events. Extracted time components of event formulae can be compared using the binary relations $=, <, \leq, \geq$ and $>$ as in the Maurer-Schmid model. For example, $\tau(c_3) > \tau(c_2)$ expresses that the event formula c_3 must have been fulfilled strictly after c_2. An event can model fulfillment of any condition, e.g., a condition used for modeling conditional release as in Section 4.3 or a simple time condition for specifying times.

A channel symbol in our model is annotated with $c_2[c_1]$ where the message on the channel needs to be fixed before $\tau(c_1)$ and the message is sent over the channel at $\tau(c_2)$. In the following example we can see how the general conditions naturally extend the time-based notion of Maurer and Schmid, where we generalize the conditions using the example introduced in Equation (5).

$$\mathcal{A}_b \xrightarrow{c_2[c_1]} \mathcal{B}_a, \quad \mathcal{B}_c \xrightarrow{c_4[c_3]} \mathcal{C}_b, \quad p(\mathcal{B}_a) = p(\mathcal{B}_c), \quad \tau(c_3) > \tau(c_2) \implies \mathcal{A}_b \xrightarrow{c_4[c_1]} \mathcal{C}_b$$

Thus, we can see that if we assume $B = p(\mathcal{B}_a) = p(\mathcal{B}_c)$ to be reliable, then $\mathcal{A}_b$ attains a channel to $\mathcal{C}_b$. A reliable party, as in the original model, states that a party forwards the received messages. We assume that all parties are reliable. When it comes to the events of the resulting channel, we can see that messages have to be fixed by $\mathcal{A}_b$ before $\tau(c_1)$ such that they can be sent via B. Still, the target channel only is ready to transmit the message after $\tau(c_4)$, i.e., at the time the channel from $\mathcal{B}_c$ to $\mathcal{C}_b$ becomes available. Note that in the

originating channels we require that B can select the message sent to $\mathcal{C}_b$ only after having received the message from $\mathcal{A}_b$.

All previously discussed examples can be extended with channel conditions to express the conditions under which the message has been fixed or can be sent over the channel in a straightforward manner. Consequently, whenever there are no specific requirements on channel conditions we omit them for simpler notation. An example for such a situation being that the only requirement on the conditions is that the message on the target channel cannot be fixed before the last message of any source channel has been sent.

3.2 Channel Transformations

We have to define the transformation rules with annotated channels as per our extension to obtain a sensible channel transformation algebra. All transformation rules of Maurer and Schmid can be carried over to our model and they need to be adapted to our notation. We refer to those transformations as basic channel transformations and provide some examples of how we change the original rules to the setting of our model. Additionally, we show how attribute-based annotations are transformed when we apply channel transformations. The rule set comprising the basic channel transformation rules and the new rules presented in this section are the basis for our extended Maurer-Schmid channel composition algebra. This small rule set is sufficient for the channel derivation calculus we propose and is minimal.

3.2.1 Basic Channel Transformations

Let us show by the example of a transformation enabled through public-key cryptography how we amend the transformations of Maurer and Schmid.

$$\mathcal{A}_b \overset{c_2[c_1]}{\bullet\!\!\longrightarrow} \mathcal{B}_a,\ \ \mathcal{A}_b \overset{c_4[c_3]}{\longleftarrow} \mathcal{B}_a,\ \ \tau(c_3) > \tau(c_2) \implies \mathcal{A}_b \overset{c_4[c_3]}{\bullet\!\!\longleftarrow} \mathcal{B}_a$$

The example transformation rule specifies that an authenticated channel from $\mathcal{A}_b$ to $\mathcal{B}_a$ over which a message fixed when condition c_1 holds and sent when c_2 holds, can be used to create a confidential channel from $\mathcal{B}_a$ to $\mathcal{A}_b$.

We use the same notion of *trust* as Maurer and Schmid, i.e., if we say that $\mathcal{R}$ trusts $\mathcal{I}$ we mean that $R = p(\mathcal{R})$ trusts $I = p(\mathcal{I})$ to correctly authenticate entities. In an example where $\mathcal{B}_a$ trusts $\mathcal{I}$, an authenticated channel can be built from two authenticated channels using $\mathcal{I}$ to forward the message from one channel to the other.

$$\mathcal{A}_i \overset{c_2[c_1]}{\bullet\!\!\longrightarrow} \mathcal{I},\ \ \mathcal{I} \overset{c_4[c_3]}{\bullet\!\!\longrightarrow} \mathcal{B}_a,\ \ \mathcal{B}_a \text{ trusts } \mathcal{I},\ \ \tau(c_3) > \tau(c_2) \implies \mathcal{A}_i \overset{c_4[c_1]}{\bullet\!\!\longrightarrow} \mathcal{B}_a \tag{12}$$

Similarly as in these examples, all transformations due to Maurer and Schmid can be adapted by denoting channels between pseudonyms as well as adapting the time constraints to generic constraints. As stated previously, we use the notion of *reliability* as in the Maurer-Schmid model, i.e., a reliable party dependably forwards received messages. Thus, reliability can be seen as a weak form of trust, which we assume to hold for all parties.

3.2.2 Attribute-based Transformations

A main aspect of our model is to allow parties to provide attribute information through the pseudonym annotation function ϕ. We next present channel transformation rules that define how attribute-based pseudonym annotations are propagated between channels. Note that those rules are orthogonal to the rules about the propagation of security annotations (i.e., bullets). This is relevant as channels with both, security and pseudonym, annotations are the most common ones in practice.

Combining Pseudonym Annotations.

A basic rule defines how pseudonym annotations of two channels between the same entities can be combined. This is relevant in a scenario where two parties A and B repeatedly communicate using the same pseudonyms $\mathcal{A}_b$ and $\mathcal{B}_a$.

$$\mathcal{A}_b \overset{\phi_1}{\longrightarrow} \mathcal{B}_a, \mathcal{A}_b \overset{\phi_2}{\longrightarrow} \mathcal{B}_a \implies \mathcal{A}_b \overset{\phi_3}{\longrightarrow} \mathcal{B}_a,\ \phi_3 = \phi_1 \wedge \phi_2$$

The intuition behind this rule is that statements about a party A acting under a pseudonym $\mathcal{A}_b$ with a party $\mathcal{B}_a$ over different channels can be combined into a new channel revealing a statement that is the conjunction of both statements. To the best of our knowledge, no cryptographic protocols that would combine ϕ_1 and ϕ_2 with other operations than conjunction exist. This rule can, like any other rule, be applied recursively to combine security properties from $k > 2$ channels into a single newly-created channel.

Connecting Pseudonym-annotated Channels.

Another new rule specifies how annotations propagate to a new channel that is created through a party connecting two channels. For example, similar to the trust-based rule in Equation (12) that allows for propagation of a security annotation, trust enables propagation of pseudonym annotations.

$$\mathcal{A}_i \overset{\phi_1}{\underset{c_2[c_1]}{\longrightarrow}} \mathcal{I},\ \ \mathcal{I} \overset{c_4[c_3]}{\bullet\!\!\longrightarrow} \mathcal{B}_a,\ \ \mathcal{B}_a \text{ trusts } \mathcal{I},\ \ \tau(c_3) > \tau(c_2) \implies \mathcal{A}_i \overset{\phi_1}{\underset{c_4[c_1]}{\longrightarrow}} \mathcal{B}_a$$

The rule shows how a pseudonym annotation of $\mathcal{A}_r$ can be transferred from the first channel to the target channel, using a party $\mathcal{I}$ as trusted intermediary. A noteworthy aspect of this rule is that the second prerequisite channel of the rule needs a bullet annotation on the side of $\mathcal{I}$ because otherwise the trust relation does not have any meaning as the pseudonym could be employed by any party.

K-fold Transfer of Certified Information.

Let us investigate the transformation required to model a setting of an identity provider I issuing certificates, e.g., anonymous credentials. The parties receiving the credentials use them to authenticate to other entities releasing attribute statements. In particular, we focus on a situation where several credentials are used to generate one attribute-based annotation.

The following transformation rule allows a party $A = p(\mathcal{A}_b)$ to use the authentications with k parties $\mathcal{I}_i$ with $1 \leq i \leq k$ to establish a new channel to $B = p(\mathcal{B}_a)$ with the pseudonym annotation comprising a combination of the annotations of the channels with $\mathcal{I}_i$.

$$\left(\mathcal{A}_i \overset{\phi_i}{\overset{c_{i_2}[c_{i_1}]}{\bullet\!\!\longleftarrow}} \mathcal{I}_i, \mathcal{I}_i \overset{c_{i_4}[c_{i_3}]}{\bullet\!\!\longrightarrow} \mathcal{B}_i,\ \tau(c_5) > \tau(c_{i_2}), \tau(c_5) > \tau(c_{i_4}), \mathcal{B}_i \text{ trusts } I_i\right)_{\forall 1 \leq i \leq k}$$
$$\left(p(\mathcal{A}_i) = p(\mathcal{A}_b), p(\mathcal{B}_i) = p(\mathcal{B}_a)\right)_{\forall 1 \leq i \leq k}$$
$$\mathcal{A}_b \overset{\phi'}{\underset{c_6[c_5]}{\longrightarrow}} \mathcal{B}_a,\ \left(\textstyle\bigwedge_{i=1}^{k} \phi_i\right) \vdash \phi' \implies \mathcal{A}_b \overset{\phi'}{\bullet\!\!\underset{c_6[c_5]}{\longrightarrow}} \mathcal{B}_a$$

This rule models establishing an attribute-based authentication of $\mathcal{A}_b$ with $\mathcal{B}_a$ where the attribute statement ϕ' is composed from multiple attribute statements ϕ_i. The latter are the annotation functions A has established using possibly different pseudonyms with parties $\mathcal{I}_i, 1 \leq i \leq k$. Note that this rule is atomic and cannot be derived from the basic rule and composition of channel rules be-

cause in this case it would be only possible to have $\phi' = \bigwedge_{i=1}^{k} \phi_i$. However, this would not be in line with privacy-preserving attribute statements. The above rule reflects what technologies such as the idemix credential system can achieve. Namely, they allow for stating relations between attributes enclosed in the credentials a party holds, not only their combination.

The above procedure can be integrated with public key infrastructures such that there need not exist a direct authenticated channel between $\mathcal{I}_i$ and $\mathcal{B}_i$, but between $\mathcal{I}$ and a certification authority $\mathcal{C}$ as well as $\mathcal{C}$ and $\mathcal{B}_c$ such that $\mathcal{C}$ is taking the role of a trust mediator for ensuring authenticity of the public keys of $\mathcal{I}_i$. We can model this derivation following the standard channel transformation rules of the Maurer-Schmid model.

4. EXAMPLES

In this section we illustrate the expressivity of our extensions with several examples. First, we discuss how to model the issuing and use of a standard X.509 certificate. Second, we extend the first example to one using a privacy-friendly anonymous credential, e.g., using idemix or U-Prove. Finally, we show how we model privacy-friendly accountability achieved by verifiable encryption.

4.1 X.509 Certificates

A standard X.509 certificate gets issued by an identity provider I to a user A. The identity provider uses a public pseudonym $\mathcal{I}$ and the user creates a pseudonym $\mathcal{A}_i$ that it only uses in this transaction. After having received the certificate, A may use the cryptographic token to present the certified attributes to a relying party R. The latter uses its public pseudonym $\mathcal{R}$.

Certificate Issuing.

The requirements for issuing a standard certificate using technology such as X.509 must allow the identity provider I to verify that $\mathcal{A}_i$ possesses the attributes $\phi_{\mathcal{A}_i}$ it will certify. In addition it will need a channel to $\mathcal{A}_i$ to send the certificate. The confidentiality of the target channel can be achieved using public-key cryptography as described in Section 3.2.1.

$$\mathcal{I} \longleftarrow\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i,\ \mathcal{I} \longrightarrow \mathcal{A}_i \quad \Longrightarrow \mathcal{I} \longrightarrow\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i$$

Through the available channels, the recipient does not get any security assurance about the issuer. In real world scenarios an identity provider, e.g., a state or a bank, may base the issuing of a credential on a strongly identifying transaction where the user needs to physically visit the issuer. Through such visit, the user authenticates the identity provider. Even if the authentication is not strictly necessary, the user may want to only provide her attributes after establishing a confidential channel, i.e., $\mathcal{I} \bullet\!\longleftarrow \mathcal{A}_i$. Using public-key cryptography, such a confidential channel can be transferred into an authentic one. Consequently, in a setting where the user wants the assurance of revealing its attributes to and getting a credential from $\mathcal{I}$, issuing would be modeled as shown next.

$$\mathcal{I} \bullet\!\longleftarrow \mathcal{A}_i,\ \mathcal{I} \longleftarrow\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i,\ \mathcal{I} \longrightarrow \mathcal{A}_i \quad \Longrightarrow \mathcal{I} \bullet\!\longrightarrow\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i$$

Release of Certified Attributes.

After having received a certificate, the user A can release the certified attributes to a relying party R. Using certification technology such as X.509 forces A to release all the certified information as the certificate can otherwise not be verified. In our channel model this corresponds to A not being able to change the endpoint annotation function $\phi_{\mathcal{A}_i}$ after the issuing process. The channel modeling the release of certified information from A to R is denoted by $\mathcal{A}_i \overset{\phi_{\mathcal{A}_i}}{\bullet\!\longrightarrow} \mathcal{R}$. If the user wants confidentiality of her data, as in the issuing process, she would need an authenticated channel $\mathcal{A}_i \bullet\!\longleftarrow \mathcal{R}$ and use public key cryptography (see Section 3.2.1). For the verification of the statement $\phi_{\mathcal{A}_i}$ the relying party needs an authentic channel with the identity provider.

$$\mathcal{I} \longrightarrow\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i,\ \mathcal{I} \bullet\!\longrightarrow \mathcal{R},\ \mathcal{A}_i \overset{\phi_{\mathcal{A}_i}}{\longrightarrow} \mathcal{R}, \mathcal{R} \text{ trusts } \mathcal{I} \Longrightarrow \mathcal{A}_i \overset{\phi_{\mathcal{A}_i}}{\bullet\!\longrightarrow} \mathcal{R}$$

We can see that due to the authentication of $\mathcal{A}_i$ at the identity provider and the authentic channel to $\mathcal{I}$, the relying party achieves the authentication of $\mathcal{A}_i$. For the same reason, the statement $\phi_{\mathcal{A}_i}$ can be transferred to the resulting channel. While we can see how attribute statements can be transferred to new channels, we do not see the full flexibility of our model due to limitations of X.509 credentials.

4.2 Anonymous Credentials

Anonymous credential systems such as the ones proposed by Brands [6] or Camenisch and Lysyanskaya [8] provide the features for demonstrating the flexibility of our model. Indeed, modeling such systems was the reason for extending the model in the first place. We will discuss two main features of anonymous credential systems. First, we model that transactions of issuing an anonymous credential and release transactions of attribute values of this credential are all unlinkable. We visualize this feature using a distinct pseudonym for each transaction. To create channel transformations we need to make sure that pseudonyms belong to the same party. We use the function p to attain this goal. Second, we capture the capability of selectively revealing the certified attributes. To model this possibility, we allow the recipient of an anonymous credential to change the endpoint annotation function ϕ.

Selective Release of Attributes.

Let us start with the selective release of attributes. Similar to standard certification technology, the issuer uses a pseudonym authenticated channel to assert that the recipient $A = p(\mathcal{A}_i)$ holds the attributes $\phi_{\mathcal{A}_i}$ it will certify. In contrast to the example discussed in Section 4.1, the technology allows A to select a pseudonym $\mathcal{A}_r$, *different* from the pseudonym used in the issuing process, when releasing the information. Note that the transformation is only possible if both pseudonyms belong to the same entity, i.e., $p(\mathcal{A}_i) = p(\mathcal{A}_r)$. Furthermore, anonymous credentials allow a user to only reveal a subset of its certified attributes. Consequently, the relying party learns $\phi_{\mathcal{A}_r}$, which is a statement derived from $\phi_{\mathcal{A}_i}$. Therefore, the issuing and use of an anonymous credential can be modeled as presented next.

$$\begin{aligned} &\mathcal{I} \longrightarrow\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i,\ \mathcal{I} \bullet\!\longrightarrow \mathcal{R},\ \mathcal{A}_r \overset{\phi_{\mathcal{A}_r}}{\longrightarrow} \mathcal{R}, \\ &\mathcal{R} \text{ trusts } \mathcal{I},\ \ \phi_{\mathcal{A}_i} \vdash \phi_{\mathcal{A}_r},\ p(\mathcal{A}_i) = p(\mathcal{A}_r) \\ &\qquad\qquad \Longrightarrow \mathcal{A}_r \overset{\phi_{\mathcal{A}_r}}{\bullet\!\longrightarrow} \mathcal{R} \end{aligned} \tag{13}$$

The semantics of the resulting channel is that using, (1) the authentic connection between $\mathcal{R}$ and $\mathcal{I}$, (2) the authentic connection between the identity provider and the $\mathcal{A}_i$, as well as (3) the trust of $\mathcal{R}$ in $\mathcal{I}$, allows $\mathcal{R}$ to create a pseudonym authenticated channel with $\mathcal{A}_r$. The statements $\phi_{\mathcal{A}_r}$ need to be derived from the original statements under which the party A has been authenticated to $\mathcal{I}$. Consequently, we can model that the relying party does not get all certified statements but only the part that is relevant for the given purpose.

Similar to the X.509 example, the need for an authenticated channel between $\mathcal{I}$ and $\mathcal{R}$ can be met using a public key infras-

tructure. Concretely, using the channels $\mathcal{I} \bullet\!\!\longrightarrow \mathcal{C}$ and $\mathcal{C} \bullet\!\!\longrightarrow \mathcal{R}$ as well as trust of $\mathcal{R}$ in $\mathcal{C}$, we can derive the channel $\mathcal{I} \bullet\!\!\longrightarrow \mathcal{R}$.

4.3 Conditional Release of Information

As already mentioned, modern cryptographic primitives allow a user A to release attributes such that they become available to a recipient party R only if a well-defined condition is fulfilled. Technically, this is achieved by the user verifiably encrypting attributes under the public key of a trusted entity T. The user needs to trust T that it will only decrypt the attributes if the condition is fulfilled. The recipient R of the verifiable encryption can verify the correctness of its content and it has to trust T to provide the information in case the condition holds. Clearly, the user and the relying party have to agree on the mentioned condition. Assuming that the condition is fulfilled, T decrypts the attributes and sends them to R to finalize the conditional release. In such a conditional release setting it may happen that the condition is never reached and the verifiably encrypted information is not learnt by R. Assuming that R only communicates the encrypted information in case the condition holds, the trusted party T does not learn the values either.

Consider as example a customer A who wants to rent a car from a car rental agency. The car rental agency acts as relying party R in a selective attribute disclosure transaction with A. Conversely to how such a transaction is carried out today, where R would require A to release personal data such as her name, address, or driver's license number, the rental agency will only request the attributes that are strictly necessary for renting a car. That is, it will require a proof that A has a valid driver's license to drive the car she wants to rent as well as information that allows R to bill A. Note that the latter could be released in a way that does not leak information about the user A, e.g., through anonymous e-cash. For simplicity we only consider an attribute statement based on the driver's license in the remainder of this scenario. Using the rule for k-fold transfer of certified information as stated in Section 3.2.2, we can easily generalize this setting to information from multiple identity providers. Through the release of a proof of owning a valid driver's license combined with the use of anonymous payment, the agency does not learn the identity of A. In fact, the transaction is carried out anonymously with R only learning required attributes of A.

However, in case of a violation of the terms and conditions of the car rental agency as well as if the user commits illegal actions (e.g., violation of traffic regulations), the agency wants to ensure accountability of A, e.g., by being able to obtain her name and address information. This goal is achieved by A creating a verifiable encryption towards the mutually-trusted entity T (e.g., the local government or a notary service) and R checking its correctness without learning the encrypted information. The encryption has a (cryptographically-associated) condition attached under which T should decrypt and provide the information to R.

We can model such scenario as follows: Let A be the party acting under pseudonyms $\mathcal{A}_i, \mathcal{A}_r$, and $\mathcal{A}_t$ with the other parties. Further, let R be the relying party acting under public pseudonym $\mathcal{R}$, the identity provider I acting under public pseudonym $\mathcal{I}$, and the trusted party T acting under $\mathcal{T}$. The channel $\mathcal{I} \longrightarrow\!\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i$ models the issuing of an anonymous credential from the identity provider $\mathcal{I}$ to party A. The authentic channel $\mathcal{I} \bullet\!\!\longrightarrow \mathcal{R}$ models that R has obtained the authentic public key of $\mathcal{I}$ and is therefore able to authenticate attribute statements made by the identity provider $\mathcal{I}$. Finally, the channel ${\mathcal{A}_r}^{\phi_{\mathcal{A}_r}} \longrightarrow \mathcal{R}$ models the release of attribute statements $\phi_{\mathcal{A}_r}$ to $\mathcal{R}$. As in example (13), we can derive a channel modeling the release of certified attributes from A to R and becoming authenticated under a pseudonym $\mathcal{A}_r$.

The channel ${\mathcal{A}_t}^{\phi_{\mathcal{A}_t}} \xrightarrow{c_{\mathrm{dec}}}\!\!\bullet\, \mathcal{T}$ models the conditional release of attributes $\phi_{\mathcal{A}_t}$ from $\mathcal{A}_t$ to $\mathcal{T}$, conditioned on c_{dec}. This is the crucial channel for modeling the conditional release of identifying attributes $\phi_{\mathcal{A}_t}$ from $\mathcal{A}_t$ to $\mathcal{T}$, which only happens once condition c_{dec} is satisfied. Thus, such channel exactly models the trusted party obtaining the conditionally released information. After T receives the information it will use an authentic channel to transfer it to R. Consequently, conditional release of information can be modeled as follows:

$$\begin{aligned} &\mathcal{A}_t \,{}^{\phi_{\mathcal{A}_t}} \xrightarrow{c_{\mathrm{dec}}}\!\!\bullet\, \mathcal{T},\ \ \mathcal{T} \bullet\!\!\longrightarrow \mathcal{R},\ \ \mathcal{R} \text{ trusts } \mathcal{I}, \\ &p(\mathcal{A}_r) = p(\mathcal{A}_t),\ \phi_{\mathcal{A}_i} \vdash \phi_{\mathcal{A}_t} \\ &\qquad\qquad \Longrightarrow \mathcal{A}_r \,{}^{\phi_{\mathcal{A}_t}} \xrightarrow{c_{\mathrm{dec}}} \mathcal{R} \quad . \end{aligned}$$

One crucial step in privacy-friendly accountability is not yet taken care of. Namely, $\mathcal{R}$ does not attain any guarantees about the attribute statements $\phi_{\mathcal{A}_t}$ that it learns. The use of an anonymous credential in combination with conditionally revealing information solves such issue. Using a sequence of channel transformation rules, we can obtain the target channels ${\mathcal{A}_r}^{\phi_{\mathcal{A}_r}} \bullet\!\!\longrightarrow \mathcal{R}$ and ${\mathcal{A}_r}^{\phi_{\mathcal{A}_t}} \bullet\!\!\xrightarrow{c_{\mathrm{dec}}} \mathcal{R}$. Those channels model both the attributes released directly to R, i.e., $\phi_{\mathcal{A}_r}$, and the ones that have been conditionally released, namely $\phi_{\mathcal{A}_t}$.

$$\begin{aligned} &\mathcal{I} \longrightarrow\!\!\bullet^{\phi_{\mathcal{A}_i}} \mathcal{A}_i,\ \ \mathcal{I} \bullet\!\!\longrightarrow \mathcal{R},\ \ \mathcal{A}_r \,{}^{\phi_{\mathcal{A}_r}} \longrightarrow \mathcal{R}, \\ &\mathcal{A}_t \,{}^{\phi_{\mathcal{A}_t}} \xrightarrow{c_{\mathrm{dec}}}\!\!\bullet\, \mathcal{T},\ \ \mathcal{T} \bullet\!\!\longrightarrow \mathcal{R},\ \ \mathcal{R} \text{ trusts } \mathcal{I}, \\ &p(\mathcal{A}_i) = p(\mathcal{A}_r) = p(\mathcal{A}_t),\ \phi_{\mathcal{A}_i} \vdash \phi_{\mathcal{A}_r},\ \phi_{\mathcal{A}_i} \vdash \phi_{\mathcal{A}_t} \\ &\qquad\qquad \Longrightarrow \mathcal{A}_r \,{}^{\phi_{\mathcal{A}_r}} \bullet\!\!\longrightarrow \mathcal{R},\ \mathcal{A}_r \,{}^{\phi_{\mathcal{A}_t}} \bullet\!\!\xrightarrow{c_{\mathrm{dec}}} \mathcal{R} \end{aligned}$$

This example nicely shows the capabilities of our model to express privacy-friendly authentication and accountability. We strongly believe that transactions as shown in this example, where a user may remain pseudonymous as long as she complies with rules and regulations, while being accountable in case of well-defined misbehaviour, will be important for the future of the Internet.

For a scenario of conditionally releasing information, we next relate the concrete information flow in a system realized with cryptographic protocols and the idealized model of the functionality as presented. Technically, the verifiable encryption towards T is sent from the user A to R. The latter can verify the encrypted attributes w.r.t. attributes certified in credentials and make sure that the user can be held accountable in case of misconduct. Once the decryption condition is fulfilled, R may request the decryption of the encrypted attributes from T. Note that in a system based on verifiable encryption, the relying party may send the verifiable encryption to T already when it receives it or it may wait until c_{dec} is fulfilled. Both flows realize the same semantics under our assumption that the trusted party T follows its protocol. Thus, this difference in the message flow in a system is not reflected in our model.

Relating this discussion to the car rental scenario, the channel ${\mathcal{A}_r}^{\phi_{\mathcal{A}_r}} \longrightarrow \mathcal{R}$ conveys the attribute statements that the user has a valid driver's license for the car intended to rent. The channel ${\mathcal{A}_t}^{\phi_{\mathcal{A}_t}} \xrightarrow{c_{\mathrm{dec}}}\!\!\bullet\, \mathcal{T}$ models the communication of the conditionally-released identity attributes to $\mathcal{T}$ to allow for user accountability. As mentioned before, the actual flow of information in a system, when using verifiable encryption as technical mechanism, is from $\mathcal{A}$ to the relying party $\mathcal{R}$ and not to the trusted party $\mathcal{T}$ itself which reflects the intended functionality. The target channels are a combination of the channel conveying the directly revealed statement and the conditional channel releasing the identity attributes once c_{dec} is fulfilled. The latter channel can thus only be derived in the case of a need to hold the user accountable and obtain her conditionally-released identity information. This is exactly what realizes the accountability feature of a transaction in which normally a user can

be known only under some attribute statement, while under a well-defined condition her identity can be obtained and the target channel be derived.

5. CONCLUSION

We have presented a simple and intuitive model for expressing the semantics of privacy-friendly authentication and accountability technologies such as anonymous credential systems and verifiable encryption. It allows for expressing the precise relations as well as the authentication and accountability properties between parties.

The concepts we cover with our model comprise pseudonyms, attribute-based authentication, as well as conditional release of information. As a result, our model can express the relevant primitives for privacy-preserving authentication and accountability at the same time.

A formalization of our model, similar to the work aiming at a more formal treatment of the Maurer-Schmid calculus [16, 17], is an interesting piece of future work. Through such formal approach, one may be able to express more precisely the functionality of cryptographic protocols and analyze, e.g., their composability.

6. REFERENCES

[1] G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. *Proc. 6th ACM CCS*, p.138–146. Nov. 1999.

[2] M. Backes, J. Camenisch, and D. Sommer. Anonymous yet accountable access control. *Proceedings of ACM WPES 2005*, November 2005.

[3] M. Backes, M. Maffei, and D. Unruh. Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol. *IEEE Symposium on Security and Privacy*, p.202–215, 2008.

[4] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get shorty via group signatures without encryption. *SCN '10*, v.6280 of *LNCS*, p.381–398. Sept. 2010.

[5] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. *CRYPTO '04*, v.3152 of *LNCS*, p.41–55. 2004.

[6] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.

[7] S. Brands and C. Paquin. U-prove cryptographic specification v1.0, Mar. 2010.

[8] J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. *EUROCRYPT '01*, v.2045 of *LNCS*, p.93–118. 2001.

[9] J. Camenisch, S. Mödersheim, G. Neven, F.-S. Preiss, and D. Sommer. A card requirements language enabling privacy-preserving access control. *Proceedings of SACMAT 2010*, p.119–128, 2010.

[10] J. Camenisch, S. Mödersheim, and D. Sommer. A formal model of identity mixer. *FMICS 2010*, LNCS. 2010.

[11] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. *CRYPTO '03*, v.2729 of *LNCS*, p.126–144, 2003.

[12] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Comm. of the ACM*, 24(2):84–88, Feb. 1981.

[13] D. Chaum and E. van Heyst. Group signatures. *EUROCRYPT '91*, v.547 of *LNCS*, p.257–265. 1991.

[14] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard), May 2008.

[15] B. Ives, K. R. Walsh, and H. Schneider. The domino effect of password reuse. *Comm. of the ACM*, 47:75–78, Apr. 2004.

[16] U. Maurer. Constructive cryptography – a new paradigm for security definitions and proofs. *Theory of Security and Applications (TOSCA 2011)*, v.6993 of *LNCS*, p.33–56. Apr. 2011.

[17] U. Maurer, A. Rüedlinger, and B. Tackmann. Confidentiality and integrity: A constructive perspective. *Theory of Cryptography — TCC 2012*, LNCS. 2012.

[18] U. Maurer and P. Schmid. A calculus for security bootstrapping in distributed systems. *Journal of Computer Security*, 4(1):55–80, 1996.

[19] U. M. Maurer and P. E. Schmid. A calculus for secure channel establishment in open networks. *ESORICS '94*, v.875 of *LNCS*, p.173–192. Nov. 1994.

[20] S. Mödersheim and L. Viganò. Secure pseudonymous channels. *Proceedings of Esorics'09*, number 5789 in LNCS, p.337–354. 2009.

[21] Security Team, IBM Research Zurich. Specification of the identity mixer cryptographic library. IBM Research Report RZ 3730, IBM Research Division, Apr. 2010.

[22] D. Sommer. Architecture. *Digital Privacy: PRIME – Privacy and Identity Management for Europe*, LNCS Volume 6545. 2011.

[23] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: Empirical results. *IEEE Security and Privacy*, 2:25–31, Sept. 2004.