RZ 3756 (# 99766) 12/04/2009
Computer Science   12 pages

Research Report

Dynamically-changing Interface for Interactive Selection of Information Cards Satisfying Policy Requirements

Patrik Bichsel, Jan Camenisch, Franz-Stefan Preiss, and Dieter Sommer

IBM Research – Zurich
8803 Rüschlikon
Switzerland

E-mail: {pbi,jca,frp,dso}@zurich.ibm.com

Dynamically-changing Interface for Interactive Selection of Information Cards Satisfying Policy Requirements

Patrik Bichsel, Jan Camenisch,
Franz-Stefan Preiss, and Dieter Sommer

IBM Research – Zurich, Switzerland
{pbi,jca,frp,dso}@zurich.ibm.com

January 4, 2010

Abstract

The possibility of carrying out a multitude of transactions, such as shopping, communicating, organizing yourself or your business, in the digital domain amplifies the need for sophisticated identification mechanisms. User-centric identity management implementations, and especially anonymous credential systems, provide the required sophistication. In particular, they support minimal data disclosure and (zero-knowledge) proofs about attributes.

A main challenge in implementing user-centric identity management is to transport the information about each transaction to the user. After all, most users are not yet aware of concepts such as data minimization or (zero-knowledge) proofs about attributes. We propose a visualization of information on the user's side together with the transaction information. We show how established concepts of user interface design can be employed to help users familiarize themselves with these formerly unfamiliar identity management concepts.

1 Introduction

In today's world wide web, users want to get access to specific resources that are provided by so-called relying parties ($\mathcal{R}$s). An $\mathcal{R}$ can specify — within a policy — what its requirements are for giving a requester (e.g., user) access to a resource. The requirements are expressed in terms of attribute information (e.g., the name or address of a user), which has to be provided to $\mathcal{R}$ by the requester. Providing such information to $\mathcal{R}$ means for the user either to reveal the actual value of the attribute or to prove knowledge of the attribute to $\mathcal{R}$. $\mathcal{R}$ can additionally request that the attributes have been certified by a specified identity provider ($\mathcal{I}$) that is trusted for this purpose. The concept of information cards — being similar to identity cards, we know from our every day life — is used in digital life to provide an $\mathcal{R}$ with the requested information (i.e., attributes). An information card thereby is simply a visual representation of a collection of attributes that are certified by a specific $\mathcal{I}$ to a user. A traditional scenario comprises a user who wants to access a restricted resource hosted by $\mathcal{R}$. To authorize the user, $\mathcal{R}$ presents its access policy for that resource to the user. Thus, a user has to choose one or several of her information cards that contain the attributes requested in the policy (for example the user's name and address) and send them to $\mathcal{R}$.

The Microsoft[1] CardSpace technology uses information cards to satisfy the policy of an $\mathcal{R}$. In this technology the supported policies are simple. For example, the user has to provide $\mathcal{R}$ with a number of values for given attributes, such as first name, last name, address, birthday, gender, etc. A paradigm in the current CardSpace technology is that only one information card may be used to satisfy the entire policy, i.e., one information card has to contain all the attributes required by a given policy.

Anonymous credential systems like Identity Mixerenable the user to issue more involved statements than the simple release of selected attributes from one card as in CardSpace. Consequently, the policies are more complex, and the paradigm of providing just one information card to satisfy the policy requirements is no longer true. In particular, Identity Mixergives the possibility to prove (in 'zero-knowledge') the knowledge of a certain attribute signed by a specific $\mathcal{I}$ instead of revealing the attribute itself. Furthermore, it is possible to prove polynomial statements over attributes. As an example, such a policy could state that a user has to provide an information card containing her name and a proof that she is older than eighteen. Additionally, she has to provide an information card issued by a credit card company that contains a credit card number and is issued to the same person as the first information card. To satisfy these more complex policy requirements, a user must use multiple of his or her information cards as the requested attributes are usually not contained in a single card.

Given the requirements formulated in a specific access policy of an $\mathcal{R}$, a user has to find a combination of attributes contained in — possibly multiple — information cards that satisfies the requirements stated in the policy. There might be various such combinations to satisfy the policy. The challenge for the user is to find a combination of information cards and attributes where all policy requirements are fulfilled. Further requirements on the attributes of the cards can be specified, e.g., specific attributes have to be provided by the same card or certain attributes from different cards have to be equal. The specification of a policy being able to express such statements is far from simple, and would exceed the scope of this document. Nevertheless we assume such a policy language as given. We focus on the complexity that has to be handled by each user in finding a suitable set of cards that satisfies the requirements of a given policy.

We describe a method for presenting an intuitive interface to select one or multiple information cards to satisfy a set of requirements that are given by $\mathcal{R}$ in the form of a policy.

2 Communication Model

To describe our approach of presenting the complex policy to the user and assist her in configuring her optimal solution, we briefly elaborate on the important parts of the underlying communication model.

Given a user $\mathcal{U}$, an identity provider $\mathcal{I}$ and an relying party $\mathcal{R}$, the following two parts in the communication flow are of relevance for us. Firstly, in a transaction with $\mathcal{I}$, $\mathcal{U}$ acquires a *credential* consisting of attributes, attribute values and most notably a signature of $\mathcal{I}$ on the attribute value pairs. A credential is presented to the user as an *information card*. Secondly, $\mathcal{U}$ can release attribute values or proofs about the values from any certificate to a relying party $\mathcal{R}$.

[1] Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.

3 Proposed Interface

From a user interface perspective, the issuance process is quite simple to handle. As a credential possibly comprises issuer-chosen, user-chosen and 'cryptographically committed' attributes, $\mathcal{U}$ possibly needs to choose some attributes which are either sent directly to $\mathcal{I}$ or a cryptographic commitment is generated and sent to $\mathcal{I}$, who then signs the credential. Thus, the few possibilities a user has allow a simple user interface design. On the other hand, during the process of convincing a relying party, the user faces a decision with a possibly large number of possible answers. This is because $\mathcal{U}$ is provided with a policy by $\mathcal{R}$ after she requested a service from $\mathcal{R}$. The next step requires the user to find her preferred combination of information cards that still allow her to comply to the policy. Additionally, she can always reject the policy, and abort the transaction. Assuming that the user selected a combination, her host compiles a statement which is sent to $\mathcal{R}$ and verified to decide upon the request of $\mathcal{U}$. Here, we focus on assisting the user in selecting her combination of cards when provided a policy by a relying party $\mathcal{R}$.

Before coming to the detailed description of the proposed interface, we want to elaborate on the tasks that a computer can perform better than a human, that is, the tasks which make our interface dynamically-changing according to a user's choice.

3.1 Machine vs. Human Strengths

The policy of a relying party can contain 'cross-credential requirements', which are cumbersome for a user to analyze and hard to explain to a user in terms of their meaning. At the same time, this task is very easily executed by a computer. Also, it is easy for a computer to scan through all of the user's information cards and verify whether any attribute can be used to partially fulfill a given policy, a task that would, again, be cumbersome for a human.

The most significant step, i.e., selecting a combination of cards that is optimal for a user in a given context, that is, the set of cards that a user prefers to show to a certain relying party $\mathcal{R}$ after receiving a policy from it, cannot be easily executed by a machine. This follows from the difficulty of building a formal model of a wide variety of user preferences. Moreover, soliciting such preferences from a user would be unrealistic. However, we identify the computer's assignments as the following:

a) meeting the cross-credential requirements; and

b) the visual distinction of *preferable* from less preferable cards. Preferability can, for example, be measured in the frequency of the usage of a card, or the number of statements is fulfills in a given policy.

3.2 Basic Interface Details

Our interface consists of two areas, the first one shows the selected attributes that the user is about to submit to the relying party. In addition, this area can show other information such as optional attributes or not submitted attributes of cards, which should be visually distinguishable from the required attributes. We call this area the *summary* as it summarizes the attributes the user has selected for submission to the relying party. Also, it implicitly visualizes the policy provided by the relying party. The second area gives access to the cards that the user possesses. In particular, this area shows cards that allow a part of the policy to be fulfilled. Let us call this area *inventory*. Cards that are not applicable within the context of a given policy are displayed in a visually distinguishable form from the cards that fulfill a policy part or that are even completely omitted from the interface. The inventory is not to be confused with the repository, which is the generally-used term for all cards a user possesses. A possible visualization of the above described ideas is given in Fig. 1.

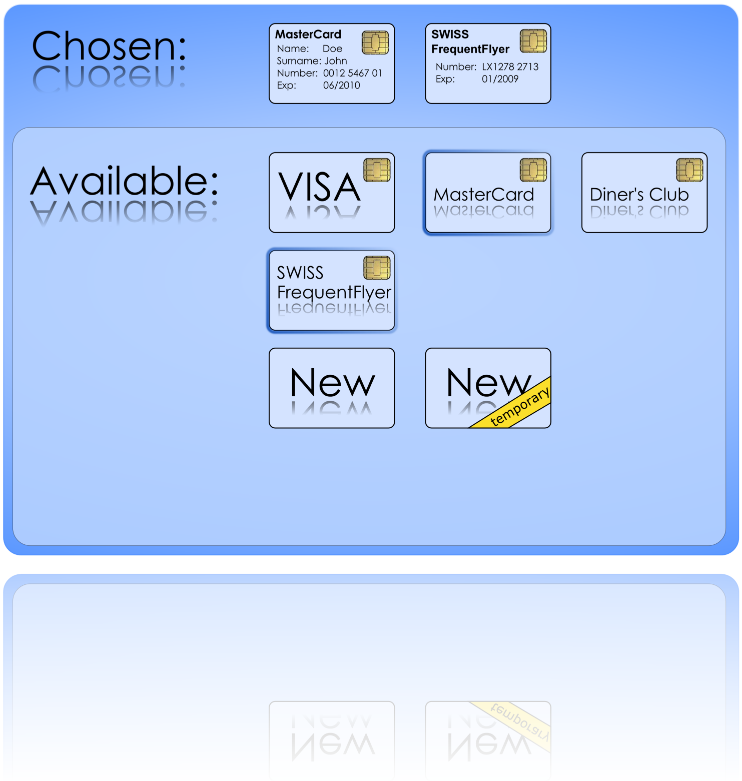


Figure 1: A possible visualization of our interface. It shows the summary presented above the inventory. In addition to the two areas, it shows the situation after the pre-selection algorithm has chosen a suitable combination of cards. The cards selected are shown in the summary and the visual distinction in the inventory is implemented using frames. We propose the choice of the cards to be influenced by the usage of the cards with respect to the different service providers (analysis of user behavior).

The two main areas align with the essential information that needs to be conveyed to the user. First, the user needs to know what information will be given away. This information corresponds to visualizing the policy received from the relying party and is shown in the summary. Second, the choices a user has need to be presented in an intuitive way. This is achieved through the inventory. Clearly, there is more information that the user needs to be aware of. For example, it needs to be clear which attributes the relying party requires to be released, which ones are used to prove a statement, and which ones can optionally be released and what benefits arise from their release. As this is information retrieved from the policy, we propose it to be visualized in the summary. In addition, we propose the use of a summary page after the user has selected a combination of information cards.[2] This page shows the information released as well as which information of the cards involved is not sent to the relying party. In particular, users nowadays are used to all attributes of a card being released if the respective card is presented to a relying party. This has been learnt by people when using real-world cards, like credit or identity cards. Showing that certain attributes are not sent to the relying party,

[2] Such a summary page may possibly be required in the future by European legislation in the context of *informed consent*.

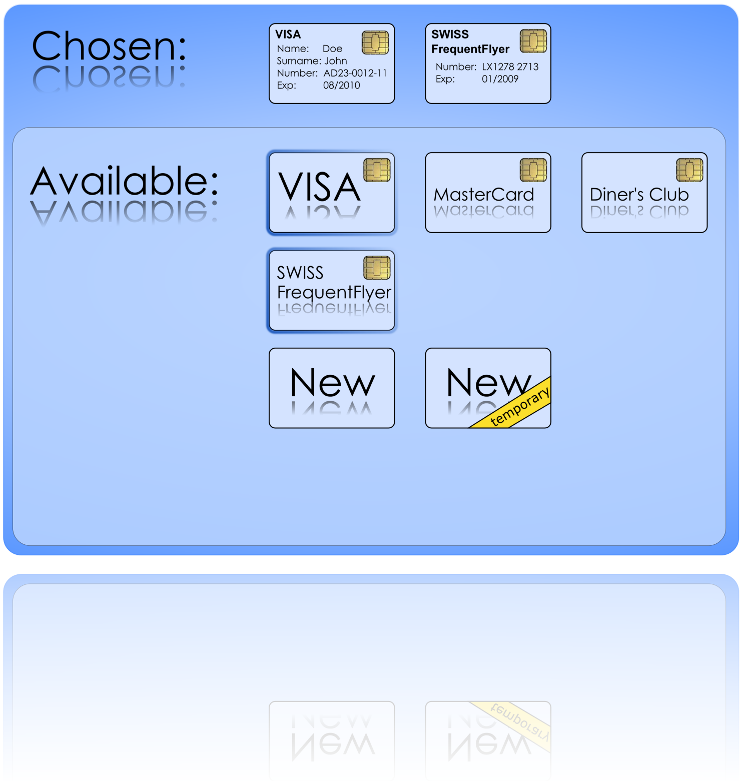
thus, is an essential part of our interface. Graphical methods to do so are, for example, fusing the cards selected into one card or letting the non-submitted attributes dissolve. It is essential that the user realizes that the cards (and the information thereon) can be sent *selectively.* This is made possible by technologies supporting selective release of attributes, such as Identity Mixeror U-Prove.

Let us elaborate on how the information in the summary is displayed. We outline several ways to show this information. For example, the attributes used from one card could be shown enclosed in a card-like object. The optional attributes could be omitted or greyed-out. An alternative method is to simply list all those attributes that are about to be released. Indication of the origin of the attribute could then be visualized by an icon next to the attribute. The icon could, for example, be a picture of the respective card. Also, the attributes originating from the same card could be grouped and only one icon could be shown next to the group of attributes. A further extension is the idea of making the icons *clickable*, which would cause alternative cards to be displayed. Apart from the visualization of the mandatory attributes and their origin, there are several ways to deal with optional attributes. As an example, showing that an attribute is optional could be achieved by displaying a check box next to it (as done in Microsoft's CardSpace). Activating the check box causes the optional attribute to be released and the summary shows the benefit arising from the release as specified in the policy by $\mathcal{R}$.

Whereas the summary visualizes the attributes selected, the inventory is the area where most of the user interaction will take place. Having the objective of an easily understandable interface in mind, it is essential to reduce the interactions with the user to a minimum. Therefore, an essential feature of our approach is the *pre-selection* of a suitable set of information cards. This also has the advantage that the summary is already populated, and the user can efficiently assess whether she wants to change something. There will be situations where the repository (i.e., all the cards the user possesses) does not allow the building of a combination of attributes that satisfies the policy. In this case the user needs to understand that only by adding an existing card to the repository or obtaining a new card she will be able to fulfill the policy. The interface conveys this by displaying, for example, an empty card, a card with some text (e.g., "new" or "add"), or a card showing a question mark in the summary.

Pre-selection. The pre-selection algorithm deserves some additional comments. First of all it is not used to solely determine the first combination of cards that is suggested to the user. It is in addition able to construct a suitable set of cards after the user has expressed her preference. That is, it comes up with a combination of information cards that fulfills the policy using a number of attributes indicated by the user. For example, whenever the user clicks a card that is currently not selected, the algorithm chooses a new set of cards making minimal adjustments to the current situation and including the clicked card/attribute in the selection. The algorithm should always prefer user-indicated attributes. The implication here is that the user can choose any attribute or card shown in the inventory and the interface will adapt dynamically such that the summary contains a set of cards that meets the policy requirement. An example of an adaptation that can be done by the user is depicted in Fig. 2.

In addition, there are several techniques to allow the algorithm to adapt its strategy. That is, the combination of cards a user would select is influenced by multiple factors, for instance, by the relying party $\mathcal{R}$ the data is about to be disclosed to, or by the frequency of using a particular card. Another idea is to let it minimize the number of cards used, that is, the solution in which every card used satisfies as many policy requirements as possible is determined. Clearly, only a combination of those ideas can lead to the best suited algorithm that can predict the user's behavior in the best possible way.



Name: Doe
Surname: John
Number: AD23-0012-11
Exp: 08/2010

Number: LX1278 2713
Exp: 01/2009

Figure 2: Given the situation in Fig. 1 the user decides to use a different credit card. The summary reflects the change and shows the attributes that would be submitted to the relying party.

It is essential that even though we call the inventory and the summary an "area", the interface does not require those areas to be visually separated. For example, all cards that fulfill a part of the policy could be shown in one area. Whenever the user clicks on a card, this card is selected and changes its appearance. This change indicates that the card is now part of the set the user wants to submit, i.e., it now is part of the summary. An example of a visualization of the two areas not being visually distinguishable objects is given in Fig. 3 and Fig. 4. Note that we assume all cards that fulfill a part of the policy to be in the inventory area. However, it is not necessary that they are shown in the inventory. Figure 3 and Fig. 4 are examples where cards that are in the summary are omitted in the inventory. On the other hand, in Fig. 1 and Fig. 2 the cards belonging to the summary are still shown in the inventory. Consequently one card might be depicted in both areas, in one of the areas, or in neither of them.

3.3 Additional Features

De-selection. A user should have the possibility to de-select attributes (cf. Figure 7 in Section 4).

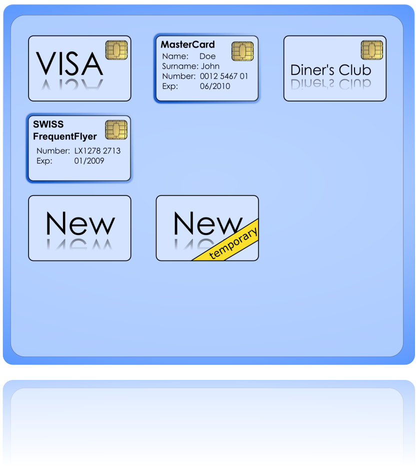


Figure 3: A view of the interface where the summary and the inventory are not separated. The cards that are selected show the additional information that in Fig. 1 only appears in the upper part of the interface.

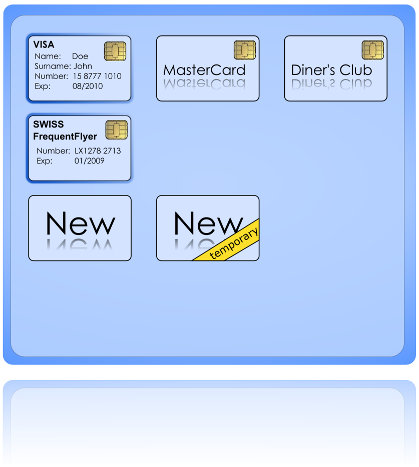


Figure 4: The user decided to use a different credit card (cf. Figure 6 in Section 4). Therefore, the newly selected card has been added to the summary without changing its place. Cards that belong to the Summary are highlighted.

Browsing. A feature that makes it easier for the user to assess the combinations that fulfill the policy is to allow her to browse the available combinations of cards or even attributes. As an example, an algorithm could change the cards selected, that is, the cards in the summary. Another algorithm could try to retain the currently selected cards and only change the attributes. Probably the user is best assisted if these different algorithms for browsing the suitable combinations are all accessible simultaneously and he can choose the algorithm that is the most suited. For example, several 'browsing bars' (e.g., visualized as standard scroll-bars) could indicate the various algorithms that allow a different browsing style. Alternatively, the user chooses the algorithm that is used to browse through the combinations, and only one set of arrows is used for all algorithms.

In the example depicted in Fig. 1 browsing through the different combinations of cards would mean that the combinations using different *credit cards* would be shown. More concretely, assuming that first the MasterCard and the Swiss FrequentFlyer cards are selected, the algorithm would then select the VISA card instead of the MasterCard. The third combination that can be explored in this situation is the Diner's club card in combination with the Swiss FrequentFlyer card.

Hovering. Another capability we find desirable is a preview mechanism. It can be activated by hovering over a card. One example where the preview is useful shows when a user is hovering over a currently selected card. This action could highlight all the alternative cards or sets of cards that can be used to replace this card (cf. Fig. 8). Another application of the preview mechanism is hovering over a card in the inventory. In this case, the changes that would happen to the summary if this card or attribute was selected could be visualized (cf. Fig. 9a or Fig. 9b).

Addition of Cards. There are several other ideas about how the user can be assisted in selecting the attributes to fulfill the policy. One such feature is allowing for easy addition of cards to the inventory and, more generally, to the repository.

This could be done by showing a card with a tag such as “add” or “new” on it.

Note that the user interfaces depicted here need to be amended with standard user interface elements, such as buttons for closing the dialogue or for submitting the combination selected.

4 Example

Let us give a concrete example, where we consider a user named “John Doe” who has both the German and the Swiss citizenship[3]. As a frequent international traveller, he possesses a Visa, a MasterCard, and a Diner’s Club credit card. Let us assume he has to fulfill the following policy issued by $\mathcal{R}$ (we use an intuitive pseudo-formal notation for presenting the policy). Let $x.y$ refer to the value of attribute y of credential x. Let $x[y_1, \ldots, y_k]$ mean the request for revealing attributes $y_1, \ldots, y_k$ of credential x.

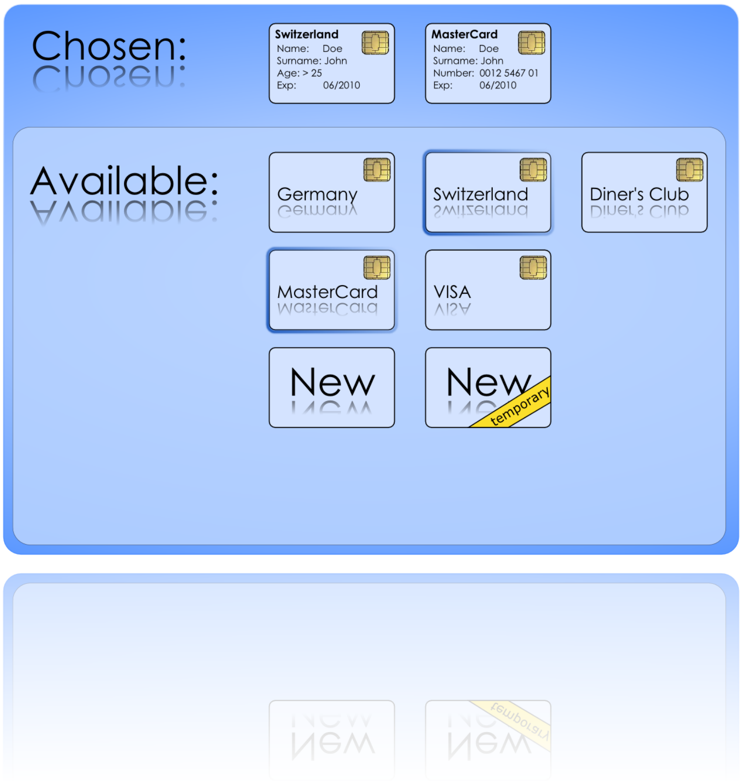
$$\begin{aligned}
&c[firstName, lastName]\wedge\\
&(c.issuer = \text{‘switzerland’} \vee c.issuer = \text{‘austria’} \vee \ldots)\wedge\\
&d[cardNo, expirationDate]\wedge\\
&(d.issuer = \text{‘}visa' \vee d.issuer = \text{‘amex’} \vee \ldots)\wedge\\
&c.firstName = d.firstName \wedge c.lastName = d.lastName\wedge\\
&(c.age > 25 \vee d.age > 25)
\end{aligned}$$

In words, the policy presented above means the following:

- Release the attributes “first name” and “last name” using an information card issued by an European government.
- Release the attributes “card number” and “expiration date” using an information card issued by one of a set of trusted credit card companies.
- Both cards need to be issued to the same person. This property is ensured in the example by ensuring the “last name” and “first name” attribute of both credit and identity card have to match. Essentially, this translates to releasing the attributes of both cards.
- Prove that the person is older than 25 years of age, using one of the cards already used.

Upon receiving the policy, the algorithm for computing the pre-selection, for example, collects all cards having an issuer matching the requirements. Figure 5 shows the situation after the pre-selection algorithm has been executed. We assume that John has never connected to the specific relying party before and that he, in general, prefers using the Swiss over the German identity card. Also, he has better conditions on the MasterCard credit card than on the other credit cards he owns. Those behavioral preferences have been taken into account by the pre-selection mechanism. The summary shows a possible combination of attributes fulfilling the policy issued by $\mathcal{R}$. We have chosen to hide cards that are not useful for the given policy from the inventory. Thus, only John’s identity cards and credit cards are listed. Given his general preference, it is likely that he would submit the required attributes from the pre-selected cards and thereby finish the transaction.

[3] This is a rare case, but it translates to a user having several cards with a very similar set of attributes.



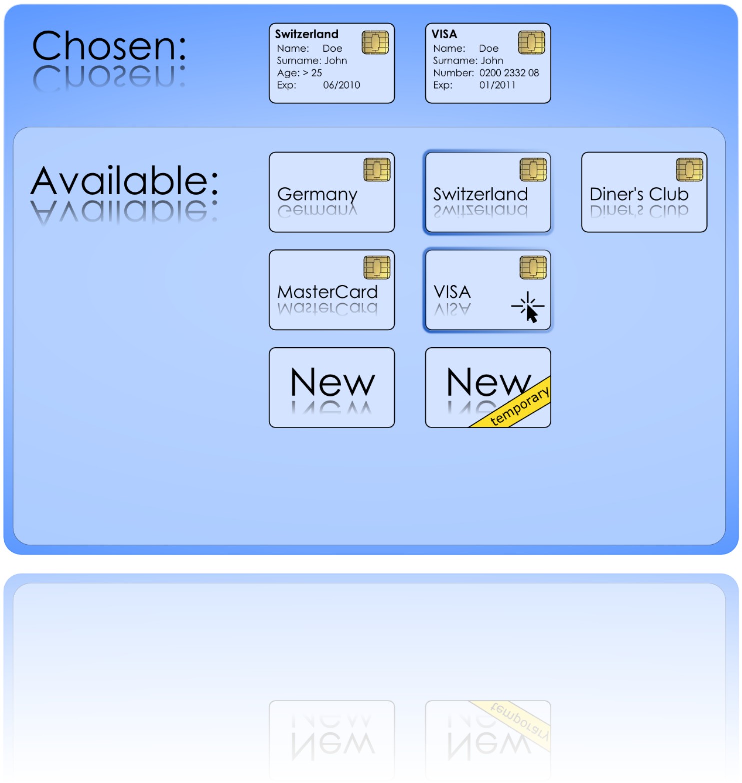
Switzerland
Name: Doe
Surname: John
Age: > 25
Exp: 06/2010

MasterCard
Name: Doe
Surname: John
Number: 0012 5467 01
Exp: 06/2010

Figure 5: The user interface after the pre-selection algorithm has selected a suitable combination of cards.

Alternatively, John can adjust the current solution to his needs. For example, he can select another credit card out of the set of credit cards presented (see Fig. 6). The summary is adapted accordingly, and the attributes released are updated with the values from the cards selected.

Assuming that John makes use of the de-selection feature requires the system to look for alternative attributes (or cards) that allow satisfying the policy. If there is no card fulfilling all requirements, the interface shows that a card needs to be added or acquired from an issuer, and the submit button vanishes (cf. Fig. 7). We also visualize what happens when John hovers over various cards. Figure 8 shows that hovering over a card in the summary emphasizes cards with similar attributes in the inventory. In contrast, we show in Fig. 9a the case where John hovers over a card in the inventory, and the user interface visualizes the consequences that clicking at the current location would have.



Switzerland
Name: Doe
Surname: John
Age: > 25
Exp: 06/2010

VISA
Name: Doe
Surname: John
Number: 0200 2332 08
Exp: 01/2011

Figure 6: The user chooses to select the VISA credit card instead of the pre-selected card.

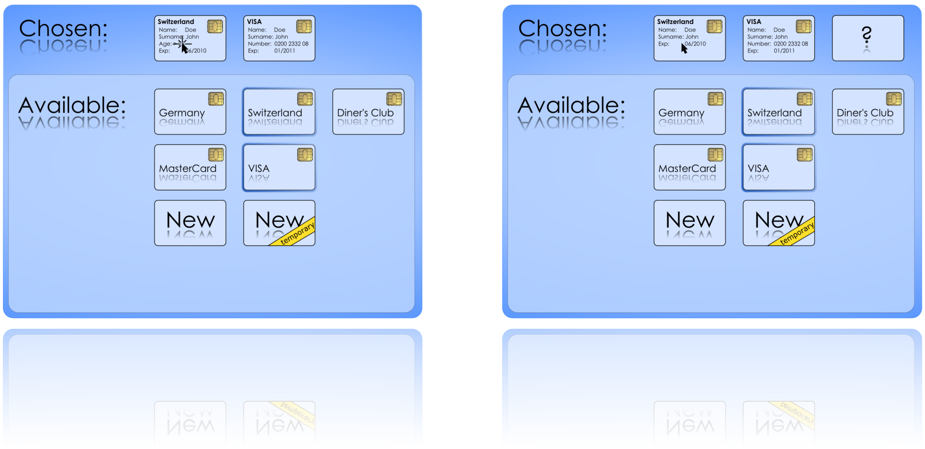
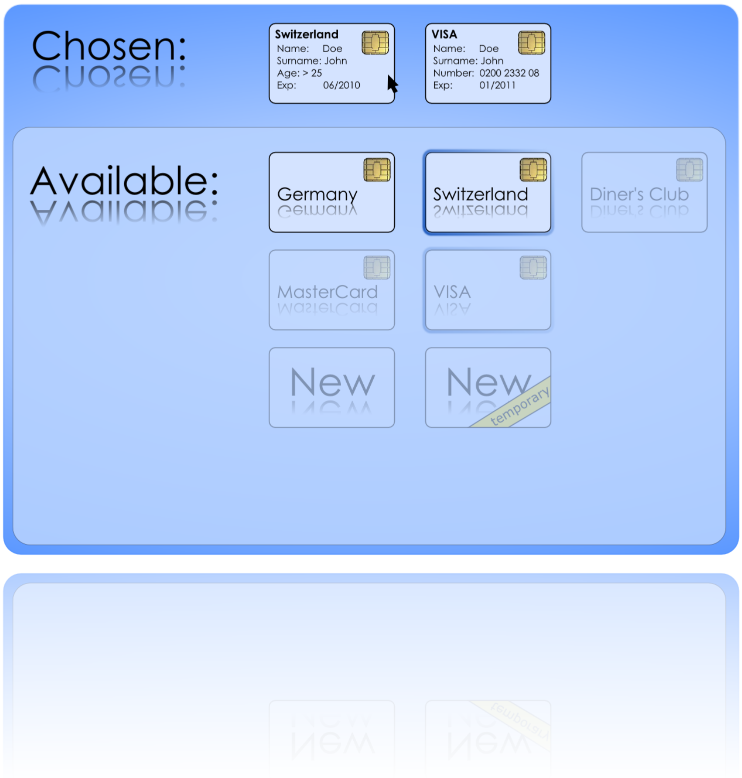


Figure 7: Clicking on an attribute (here the age) causes it not to be used. We visualize this by letting it disappear. Howevre, the policy requires this attribute and the user has currently no information card containing this attribute. Thus, the interface shows a card with a question mark in the summary.



Switzerland
Name: Doe
Surname: John
Age: > 25
Exp: 06/2010

VISA
Name: Doe
Surname: John
Number: 0200 2332 08
Exp: 01/2011

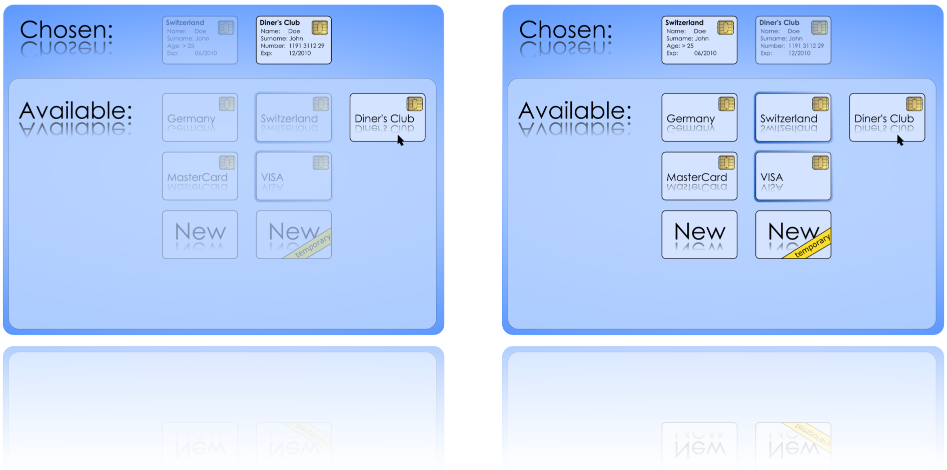
Figure 8: Hovering over a card in the summary shows the alternative cards in the inventory. To highlight those cards the non-applicable cards are faded out.

The user interface supports the creation of new information cards. This is useful if a policy is not satisfiable with the set of cards the user has, or if he does not want to use the cards he has. Special 'New Card' and 'New One-time Card' entries are therefore shown together with the alternative cards. A click on such a card lets the user create a new self-issued information card. Note that such cards are only applicable in a restricted set of scenarios. If a one-time card is created, this card will remain available only for a predefined period of time.

If a policy is not satisfiable with the current set of cards, the user interface shows special "New" and "New temporary" cards. John can create new or add existing cards using the corresponding buttons to satisfy the missing parts of the policy. Temporary cards will be discarded after the transaction, which is a feature that people like when interacting (presumably) only one time with a specific $\mathcal{R}$.

This example omits visualization of, for example, the distinction between mandatory, optional, and implicitly-released attributes. This distinction could be achieved, by using different fonts or by grouping the attributes according to their types. Also, the browsing bars for switching between different solutions (i.e., combinations of cards or attributes) are not presented. We believe that adding those features is straightforward.

To summarize, the interface shows the policy in a human-readable form in the summary. In addition, the information released is presented in the same area. The inventory helps the user assess the possibilities of complying with the given policy. Thus, the user is enabled to reach an informed decision about the release of personally identifiable information in any transaction guided through the interface.



(a) (b)

Figure 9: Two possibilities to support a user in showing which cards will be changed upon clicking in the current area. Hovering over a non-selected card shows which card would be replaced in the summary. The first version fades out all cards except the one that could be selected. The second version only shows a transparent version of the card, the user is hovering on, in the summary.

5 Conclusion

Anonymous credential systems, i.e., Identity Mixer, are able to generate statements that are counterintuitive for users because they are not familiar with concepts such as zero-knowledge proofs that the technology exploits. Accordingly, the following information must be conveyed to the user:

- Attributes can be selectively disclosed.
- Attributes from different cards can be combined into one statement.
- The policy might have constraints on the combination of cards/attributes, those can be proved in zero-knowledge.
- Arithmetic relations about attributes can be proved in zero-knowledge, i.e., without releasing the attribute itself.

We believe that by cleverly visualizing released vs. hidden attributes, and using visualization techniques, such as dissolving attributes or merging the summary into the (legally required) summary page, we can make the non-intuitive concepts easily understandable. By reducing the number of cards to the inventory, that is, by letting the computer analyze which cards are applicable and by having the pre-selection mechanism taking care of cross-requirements, we allow the user to indicate her preference, and the mechanism takes care of the important but cumbersome details.