An extended abstract of this paper appeared in the
Proceedings of ACM CCS, Chicago, IL, USA, November 2011.

DEMO: A Comprehensive Framework Enabling Data-Minimizing Authentication

Patrik Bichsel and Franz-Stefan Preiss
IBM Research – Zurich, Switzerland
{pbi, frp}@zurich.ibm.com

ABSTRACT

Authentication is an all-embracing mechanism in today's (digital) society. While current systems require users to provide much personal data and offer many attack vectors due to using a username/passwords combination, systems that allow for minimizing the data released during authentication exist. Implementing such data-minimizing authentication reduces the number of attack vectors, enables enterprises to reduce the risk associated with possession of sensitive user data, and realizes better privacy for users. Our prototype demonstrates the use of data-minimizing authentication using the scenario of accessing a teenage chat room in a privacy-preserving way.

The prototype allows a user to retrieve credentials, which may be seen as the digital equivalent of the plastic cards we carry in our wallets today. It also implements a service provider who requires authentication with respect to a service-specific policy. The prototype determines whether and how the user can fulfill the policy with her credentials, which typically results in various options. A graphical user interface then allows the user to select one of these options. Based on the user's input, the prototype generates an Identity Mixer [12] proof that shows fulfillment of the service provider's policy without revealing unnecessary information. Finally, this proof is sent to the service provider for verification. Our prototype is the first implementation of such far-reaching data-minimizing authentication, where we provide the building blocks of our implementation as open-source software.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Access Controls, Authentication, Cryptographic Controls*;
K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Languages, Security

Keywords

Authentication, Policy Languages, Privacy, Anonymous Credentials, Digital Credentials.

CCS'11, October 17–20, Chicago, Illinois, USA.
ACM 978-1-4503-0948-6/11/10.

1. INTRODUCTION

In the digital world today, authentication is a ubiquitous topic. For the use of virtually any online service, such as music streaming platforms or online bookstores, prior creation of a user account including a username/password pair is a strict necessity or at least strongly encouraged. For performing the actual authentication, users prove knowledge of the previously agreed username and password, which is a simple and cheap mechanism most people are familiar with. However, this approach and it's implementation in practice have various drawbacks for both users and service providers.

First, during registration, users typically have to disclose extensive amounts of personal information, which is often of no direct relevance for the service at hand. However, the accumulated sets of data pose the imminent danger of accidental leakage or theft [1]. Such data loss may not only have serious legal and financial consequences [10], but also results in damage of the service provider's reputation [13]. Second, all transactions of a user with one service provider are inherently linkable. From a user's perspective this is often not desirable as it may lead to identification, thus, reduces her privacy. Third, the fact that many users improvidently reuse their login data [8] makes them, on one hand, vulnerable for being impersonated by dishonest service providers, and, on the other hand, linkable across *multiple* domains. Finally, service providers struggle with low quality of account data as users may provide incorrect registration information. There are various reasons for this behaviour ranging from their frustration of the tedious and perseverative registration procedures or their striving for better privacy to the possibility of doing so, which occurs as the data is unverified. Even though the service providers wind up possibly fake data, they have to protect it as if it were sensitive personal information. Summarized, the drawbacks of classical authentication mechanisms for service providers are the need for data protection and poor data quality, where for users the loss of their privacy is most significant.

A first important step towards overcoming these drawbacks is the use of credential-based authentication systems [6] that allow for a selective disclosure of attributes certified in credentials. Credentials, in this context, can be seen as the digital equivalent of the plastic cards we all carry around in our wallets. More concretely, they contain bundles of certified attributes such as name, nationality, or date of birth. There are various technologies that implement such credential concept. When it comes to the selection of this underlying technology, *anonymous credential systems* [7, 4, 5] are the most privacy-preserving choice as they offer a comprehensive set of features. In particular, they allow users to *prove properties* about the attributes contained in the credentials they own in an *unlinkable* manner. For example, a user can prove that she is younger than a given age according to a credential that certifies her date of birth, without revealing her exact birth date. Moreover, she can generate multiple such proofs without the verifier realizing that the same entity generated them, that is, the proofs are unlinkable. As such anonymous-credential-based setup allows for reducing the disclosed data to a minimum with respect to the scenario at hand, we call it *data-minimizing authentication.*

We developed a prototype application that demonstrates the feasibility of this innovative authentication technology. In particular, we created a Java framework that provides implementations for all the components necessary to perform data-minimizing authentication on the basis of certified credentials, which we publish as open-source software. While the framework may be combined with any credential technology (e.g., X.509, U-Prove, Kerberos, OpenId), we provide a plug-in for the Identity Mixer (Idemix) anonymous credential system [12]. As data-minimizing authentication in general, and our implementation in particular, eliminates all above-mentioned disadvantages of classical authentication, we believe it has the potential to induce a *paradigm shift* for authentication away from username/password pairs.

The technology we implemented is similar to the demonstrators of the U-Prove technology [9]. The two main differences are that, (1) our demonstrator implements a much broader range of authentication statements, which results from the expressivity of the used authentication language as well as the features offered by Idemix, and (2) all computations requiring a user's credentials are executed entirely on the equipment owned by the user, which allows for the best possible privacy protection. The drawback of this far-reaching protection being that it requires the user to install software on her computing equipment or the identity providers need to distribute devices holding credentials (e.g., smart cards) which are powerful enough to compute Idemix proofs.

2. DATA-MINIMIZING AUTHENTICATION

Let us discuss the components of data-minimizing authentication and their interaction (cf. Figure 1). Users own credentials (in [6] called "cards") containing certified attribute values, which are issued to them by so-called identity providers. The figure depicts how a user wants to access a service (e.g., a teenage chat room) hosted by some server. For using its service, the server requires the user to authenticate with respect to a service-specific authentication policy. An important aspect of data-minimizing authentication is that these policies are formulated in terms of properties of the user's credentials. For example, a policy could specify that only users who are teenagers according to a national ID card may use the service. Upon receiving an authentication request (1) for a service, a server determines and pre-evaluates (1a) the applicable policy and sends it to the user (2). During this pre-evaluation, references to static content such as the current date are resolved to generate the applicable policy. After receiving the policy, the user's system determines which *claims*, that is, statements about a subset of attributes of one or more of the available credentials, that fulfill the policy can be made (2a). For example, a policy requiring the user to be a teenager according to an ID card may be fulfilled by means of a user's national ID card or her student ID. The statement of being a teenager can be made by disclosing the exact date of birth or by proving that the date of birth lies at least thirteen but less then twenty years in the past. The latter option minimizes the disclosed information, thus, it is significantly more privacy-preserving. Note that the claims a user can make depend on the capabilities of the underlying credential technology. The user interactively selects the favoured claim (2b) using the graphical user interface (GUI). Based on her decision the credential technology is used to generate *evidence* (also called proof) that supports this claim (2c). To this end, a technology-specific *proof specification* (e.g., an Idemix proof specification) is generated. The resulting technology-specific evidence is the basis for the server to verify the claim's validity. The claim together with the accompanying evidence are then sent to the server (3) who verifies whether the claim implies the policy (3a) and whether the claim's evidence is valid (3b). Depending on the credential technology, the evidence may be generated and verified with or without the identity provider being involved. After successful verification, the user is authenticated (4) as someone fulfilling the authentication requirements prescribed by the policy.

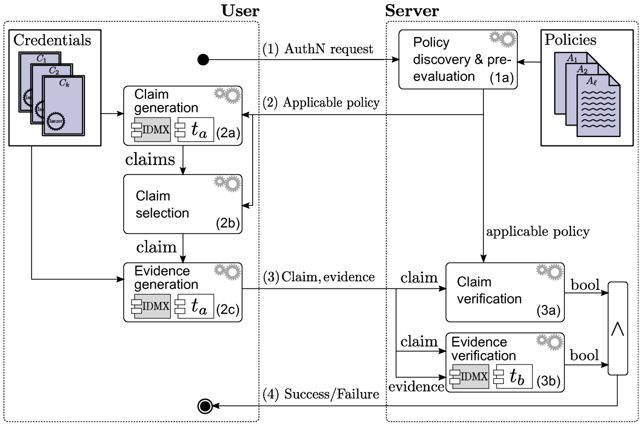

Figure 1: Components & Communication Sequence of Data-Minimizing Authentication

3. AUTHENTICATION FRAMEWORK

We implemented an open-source framework [11] for performing data-minimizing authentication transactions as outlined in Section 2. On top of our framework, we created a prototype application (cf. Section 4) that demonstrates such authentication transaction by means of a comprehensive example. In particular, our framework implements policy pre-evaluation (1a), claim generation and verification (2a, 3a), claim selection (2b), and evidence generation as well as its verification (2c, 3b). It defines and operates on Java interfaces, which open it up for extensions with any credential technology that provides technology-specific plug-ins for claim and evidence generation. To enable the implementation of fully data-minimizing authentication scenarios, we provide such plug-in for the Idemix anonymous credential technology (denoted as IDMX in Figure 1).

A vital prerequisite for implementing the mentioned components is the availability a policy language for expressing the server's authentication requirements as well as a claim language to make statements about (attributes of) the user's credentials. Camenisch et al. [6] provide the former with their *credential-based authentication requirements language* (CARL) and Bichsel et al. [2] show how a constraint version of CARL can be used as claim language. We provide an implementation of both languages based on the *Xtext language framework*[1] of Eclipse. The main challenge for implementing CARL was the fact that it allows for specifying an arbitrary boolean predicate over credential's attributes by means of a mathematical formula expressed in unquantified predicate logic. As the attributes have *data types* that are determined by so-called *credential types*, we implemented a type system that ensures type correctness of the formula. Our authentication framework features a policy editor for the CARL policy language. The editor is provided as Eclipse platform plug-in and eases authoring policies that are correct concerning syntax and the formula's data types.

We further created an interpreter that evaluates a formula with respect to a set of credentials. During claim generation, a technology-independent *assignment finder* component first determines whether and how a user can fulfill a given policy with respect to her available credentials. The resulting *assignments* are then transformed into claims by plug-ins specific to the credential technologies of the assignments' credentials. This differentiation is necessary as not all technologies support the same set of privacy-preserving features.

The GUI for claim selection is implemented by means of the Rich Ajax Platform (RAP)[2], which allows for building rich web-based applications by means of the Eclipse development model. RAP is similar to the Eclipse Rich Client Platform (RCP), but it renders the GUI widgets in a Web browser rather than in an operating system window. In our prototype, a user is redirected to a local Web page displaying a RAP GUI for selecting a claim suitable to fulfill a given policy. The choice of Idemix as underlying credential technology has implications on the design of the GUI, where we follow the ideas proposed in [3]. The goal of the GUI being to assist the user in assessing the possible choices and facilitate the selection of the most privacy-preserving option.

We released the current version of our *credential-based authentication framework* under the Eclipse Public License. To use the framework you will also need the Idemix credential technology implementation. You can download both components from http://www.primelife.eu/.

4. PROTOTYPE APPLICATION

To demonstrate the use of our authentication framework we created a prototype application where we use the scenario of a user authenticating to a teenage chat room. While this scenario does not allow us to exhibit the full expressivity of the authentication framework, it shows the potential of credential-based authentication and builds the basis for discussion of more elaborate scenarios.

The prototype features an identity provider issuing credentials with attributes typical for national identity (ID) cards (e.g., name, date of birth, or address of a user). As a particular strength of the prototype we implemented one approach for binding credentials to a smart card. More specifically, before issuing a credential the identity provider verifies that the user has a smart card containing a credentials, the so-called *root credential*. The newly issued credential and the root credential share an attribute that never leaves the smart card. Note that not even the user knows the value of this attribute creating the binding of credentials. Consequently, whenever the user proves knowledge of the ID credential, knowledge of the root credential must be proved in addition, which implies that the smart card must be present. This binding to a tamper resistant device copes with the fact that digital credentials by themselves are inherently easy to copy, thus, allow for misuse.

The prototype also implements a teenage chat room, which acts as a service provider. In our scenario the authentication policy of the service provider requires a user to prove that she is between 13 and 20 years old. Receiving this policy the framework checks for credentials available on the user's host. Note that even if there are several ID credentials available only the one with a corresponding root credential on a connected smart card is considered available. Using the available credentials the framework determines the options a user has to fulfill the policy and presents those option to the user in the GUI. Based on the selection of the user the framework issues an Idemix proof. The prototype sends the proof to the service provider who verifies it using the framework. The latter bases it's authentication decision on the verification of this proof and on the relation of the claim to the original authentication policy.

5. CONTRIBUTIONS

We provide a generic and extensible software framework for data-minimizing authentication. This framework uses the simple, yet expressive, authentication policy language CARL. In addition, it comes with a policy editor that greatly simplifies the authoring of authentication policies. Furthermore, it provides a GUI that allows a user to interactively select a combination of credentials. During the selection process we convey to the user which attributes are released and what is *not* transmitted to the service provider.

The framework addresses the downsides of classical authentication in the following way. Property proofs allow users to disclose just the relevant bits of information while service providers are assured that their authentication policies are fulfilled. In many cases this eliminates the need for the creation of user accounts including the associated disclosure of personal data. Consequently, service providers can avoid protecting sensitive user information while increasing the security of their authentication process. The latter results from the use of certified attributes, that is, the service provider's data quality issues are addressed, and due to the absence of usernames and password, the issue of user impersonation is mitigated.

In some cases service providers have a legitimate need to recognize recurring users. Here, data-minimizing authentication offers the possibility to execute transactions in a pseudonymous manner. Therefore, user linkability is no longer mandatory but rather an optional feature.

Our prototype application shows the use of our authentication framework on an example that is simple, yet, shows the innovative capabilities of data-minimizing authentication.

[1] See http://www.xtext.org/.

[2] See http://www.eclipse.org/rap/.

6. REFERENCES

[1] Liana B. Baker and Jim Finkle. Sony playstation suffers massive data breach. http://www.reuters.com/article/2011/04/26/us-sony-stoldendata-idUSTRE73P6WB20110426, April 2011.

[2] Patrik Bichsel, Jan Camenisch, and Franz-Stefan Preiss. A comprehensive framework enabling data-minimizing authentication. In *ACM DIM '11*, 2011.

[3] Patrik Bichsel, Jan Camenisch, Franz-Stefan Preiss, and Dieter Sommer. Dynamically-changing interface for interactive selection of information cards satisfying policy requirements. Technical Report RZ 3756, IBM Research Zurich, 2009. Available at domino.research.ibm.com/library/cyberdig.nsf.

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

[5] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT '01*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.

[6] Jan Camenisch, Sebastian Mödersheim, Gregory Neven, Franz-Stefan Preiss, and Dieter Sommer. A card requirements language enabling privacy-preserving access control. In *Proceedings of SACMAT 2010*, pages 119–128, 2010.

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

[8] Blake Ives, Kenneth R. Walsh, and Helmut Schneider. The domino effect of password reuse. *Comm. of the ACM*, 47:75–78, April 2004.

[9] Microsoft Corp. U-prove community technology preview. https://connect.microsoft.com/site1188, August 2011.

[10] Ponemon Institute and Symantec Corp. 2010 annual study: Global cost of a data breach. http://www.symantec.com/content/en/us/about/media/pdfs/symantec_cost_of_data_breach_global_2010.pdf, May 2011.

[11] Franz-Stefan Preiss. Credential-based authentication framework. http://www.zurich.ibm.com/~frp/com.ibm.zurich.authn.cb/, June 2011.

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

[13] TheFirewall.co.uk. Research reveals the folly of sony's handling of its data breach double whammy. http://www.thefirewall.co.uk/news/71/, May 2011.