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 [10] 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, which may result in consequences such as financial loss or damaged reputation of a service provider. Second, all transactions of a user with one service provider are inherently linkable, which reduces her privacy. Third, the fact that many users improvidently reuse their login data [7] makes themvulnerable for impersonation and linkable across *multiple* domains. Finally, service providers struggle with low quality of account data as users may provide incorrect registration 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 [5] that allow for a selective disclosure of attributes certified in credentials. Credentials, 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. Of the various available credential technologies, *anonymous credential systems* [6, 3, 4] provide the most extensive set of privacy-preserving features. In particular, they allow users to *prove properties* about the credentials they own in an *unlinkable* manner. As such anonymous-credential-based setup allows for reducing the disclosed data to a minimum w.r.t. the scenario at hand, we call it *data-minimizing authentication*.

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 [10]. 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 demonstrate is similar to the demonstrators of the U-Prove technology [8]. 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.

2. DATA-MINIMIZING AUTHENTICATION

Let us discuss the components of data-minimizing authentication and their interaction (cf. Figure 1). Users own credentials containing certified attribute values. Credentials are issued to users by so-called identity providers. The figure depicts how a user wants to access a service (e.g., a teenage chat room), where the host requires authentication w.r.t. a service-specific policy. The policy may specify properties of a user's credentials, e.g., a policy could specify that a user needs to be a teenager according to a national ID card, to access 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 it, the user's system determines which *claims*, i.e., statements about a subset of attributes of one or more of the available credentials, fulfilling 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 but is not offered by all underlying credential technologies. 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 is then sent together with the accompanying evidence to the server (3) who verifies whether the claim implies the policy (3a) and whether the claim's evidence is valid (3b). After successful verification, the user is authenticated (4) as someone fulfilling the authentication requirements prescribed by the policy.

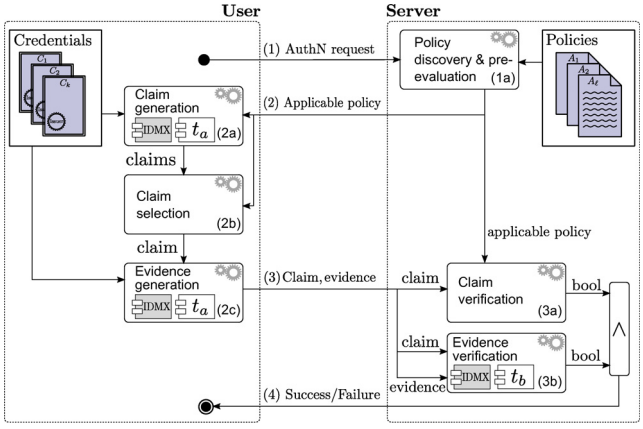

Figure 1: Components & Communication Sequence of Data-Minimizing Authentication

3. AUTHENTICATION FRAMEWORK

We implemented an open-source framework [9] 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. [5] provide the former with their *credential-based authentication requirements language* (CARL) and Bichsel et al. [1] 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 w.r.t. 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 w.r.t. 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

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

rich web-based applications as 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 [2]. 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 a binding of the credentials to a smart card. More specifically, before issuing a credential the identity provider verifies that the user has a smart card with a root credential. The newly issued credential and the root credential share an attribute that never leaves the smart card. Consequently, whenever the user proves knowledge of the ID credential, knowledge of the root credential must be proved in addition. 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 the 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, which 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, i.e., 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.

6. REFERENCES

[1] P. Bichsel, J. Camenisch, and F.-S. Preiss. A comprehensive framework enabling data-minimizing authentication. *To appear*, 2011.

[2] P. Bichsel, J. Camenisch, F.-S. Preiss, and D. Sommer. Dynamically-changing interface for interactive selection of information cards satisfying policy requirements. Technical Report RZ 3756, IBM Research Zurich, 2009.

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

[4] 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.

[5] 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.

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

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

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

[9] F.-S. Preiss. Credential-based authentication framework. http://www.zurich.ibm.com/~frp/com.ibm.zurich.authn.cb/, June 2011.

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