Electronic Identities Need Private Credentials

Jan Camenisch, Anja Lehmann, and Gregory Neven

IBM Research - Zurich

Secure, trustworthy transactions over the Internet require that the communication be encrypted and that the communicating parties authenticate each other. For Web browser transactions, the TLS (Transport Layer Security) protocol routinely and adequately performs encryption and server authentication. The predominant form of user authentication, however, remains usernames and passwords. When creating accounts, users often must additionally provide a list of self-claimed attributes such as name, address, or birth date. Only a few attributes such as email address and credit card information have some (external) mechanism to check their authenticity.

Solutions such as the Security Assertion Markup Language (SAML), OpenID, or X.509 certificates let users authenticate and transfer trusted attributes, certified by issuers, to relying parties. Such technologies are slowly gaining momentum but, as we point out in this article, present considerable security and privacy concerns. Briefly, either the issuer learns the details of all user transactions and unnecessarily exposes the issuance key to online attacks, or the relying party learns more attributes than necessary, thereby becoming an attractive target for hackers.

Private credentials are a superior solution offering the best of both worlds. Issuers don't have to be involved during authentication. Users disclose only those attributes required by the relying parties and can do so without being easily tracked across their transactions.

User-Centric Identity Management

In the following, we view a user's identity as a set of attributes or, more generally, any information a party knows about a user. So, an identity exists only in connection to a party. Because different parties know different things about the same user, every user has many different identities, possibly even multiple identities with each party he or she interacts with. To verify the authenticity of a user's attributes, a party can either perform identity vetting on the attributes itself (for example, require the user to provide physical documents or take an exam) or rely on a specialized issuer whose identity-vetting procedures it trusts.

For example, in Figure 1, John has many different attributes, subsets of which make up John's different identities with the people and institutions he interacts with online. Identities sharing a unique attribute can be linked; for example, his social security number can be linked across his healthcare-related identities. Other identities can't be linked. John should be able to manage these identities the same way he manages them in the paper-based world.

Such user-centric identity management requires two basic mechanisms: one to transfer certified attributes from an issuer to a relying party and one to authenticate (or reauthenticate) a user under an established identity. The former mechanism is essential to conduct trusted electronic transactions and requires cryptography. The latter mechanism can in principle be realized with a simple username and password, but this provides poor security guarantees. Indeed, passwords are well known to be vulnerable to password guessing, phishing, and social-engineering attacks. Their insecurity affects privacy, too. To alleviate these shortcomings, many service providers collect as much side information (for example, location or transaction history) about users as they can and analyze that data to detect suspicious behavior and potential breaches. So, a stronger cryptographic mechanism for authentication involving public-key cryptography seems advisable.

In our paper-based world, attribute transfer and authentication are often folded into one mechanism. For instance, a driver's license transfers the attribute "I'm allowed to drive a car" from the issuer to any relying party and, via the photo on it, provides an authentication mechanism.

When realizing attribute transfer and authentication for the digital world, mimicking the paper-based solutions, as often happens, isn't enough. Instead, you must consider the very different environment: digital data is easily copied and virtually impossible to control once released. So, any digital realization must follow the principle of data minimization. When a user transfers an attribute from an issuer to a relying party, neither party should be able to learn any information that the transferred attribute hasn't already revealed, even if the parties collaborated.

Of course, an identity management system adhering to these principles doesn't eliminate all the digital world's dangers. Communication and stored information should always be encrypted. Sensitive data should be accompanied with usage policies defining how to treat it, who can use it, for what purpose it's to be used, and when to delete it. We don't elaborate on these issues here; we concentrate on the identity management mechanisms.

Existing Solutions to Transfer Attributes

Roughly, existing solutions to transfer certified user attributes from an issuer to a relying party are either offline or online. Offline solutions involve the issuer only at the time of issuance. Online solutions also actively involve the issuer during attribute transfer.

The most prominent offline solution is X.509 v3 certificates with attribute extensions. Here, the issuer or certificate authority (CA) signs the user's public key together with his or her attributes and includes the signature in the certificate. The way X.509 credentials are constructed forces the user to reveal all of the attributes in the certificate when transferring an attribute. Moreover, the user's public key acts as a unique identifier that follows the user across all of his or her online transactions.

In online solutions, the user first authenticates directly to the issuer. The issuer then creates a verifiable token for the specific set of attributes required by the relying party. Popular examples following this approach include SAML and WS-Federation, as well as the more lightweight OpenID.

The advantage of this approach is that only the required attributes are revealed. However, the issuer learns which user authenticates to which relying party at which time. Although some protocols can optionally hide the user's identity from the relying party and hide the relying party's identity from the issuer, this doesn't help when relying parties and issuers compare their transaction logs.

In addition, with online solutions, the issuance key must be on a system that's permanently connected to the Internet. This considerably increases the issuer's vulnerability to intruders, thus endangering the entire system's security.

Private Credentials

Private credentials are similar to the offline approach in terms of the overall functionality and provided security guarantees, while letting users control and separate their different identities [1–3].

How They Work

Similarly to the approach with X.509 certificates, each user generates a secret key and corresponding public key. The credential is a signature by the issuer on the user's attributes and public key. To transfer attributes, the user signs a challenge message using his or her secret key and sends the signature along with the issuer-signed credential to the relying party. The user can authenticate under a public key by signing a challenge message using the secret key.

However, private credentials have two unique properties. First, the user can generate many public keys from the single secret key. These public keys can't be linked. That is, given two public keys, you can't tell whether they belong to the same user or two different users.

Second, a credential issued to one public key can be (repeatedly) transformed into a credential that's valid on another public key of the same user. Moreover, the transformed credential can contain a selected subset of the attributes in the original credential. Transformed credentials are unlinkable. That is, for two transformed credentials with disjoint sets of revealed attributes, you can't tell whether they originated from the same credential or different credentials. All credentials, both the transformed and original ones, still verify correctly with regard to the issuer's verification key.

These properties are crucial to let users properly manage their identities. They can generate one public key for each identity. To transport attributes from an issuer to a relying party, the user first obtains a credential including those attributes for the public key by which the user is known to the issuer. The user then transforms the credential so that it contains only those attributes the user wants to transfer and so that it's valid for the public key by which the user is known to the relying party.

As you can see, the high-level principles of private credentials and traditional certificates are largely the same. The sole difference is that the two approaches use different cryptographic algorithms to generate public keys and sign certificates and messages. So, you can use private credentials in any situation in which you can use traditional certificates, simply by replacing the algorithms. Private credentials provide the same level of security but also guarantee privacy.

Usage Scenarios

Consider an online journal to which users subscribe to access articles. When subscribing, the user generates a fresh public key from the secret key and obtains from the publisher a private credential on this public key. When the user later wants to access an article, he or she generates another fresh public key, transforms his or her credential, and sends both to the journal website. As with traditional certificates, the journal has the strong cryptographic guarantee that no unregistered users can download articles. However, users can rest assured that no one is tracking or profiling their reading behavior.

As another example, suppose Alice's electronic identity card is a private credential containing as attributes her name, birth date, and address. Further suppose Alice's hometown provides an online feedback system for its residents. Now, by transforming her identity credential into one that contains merely the ZIP code, she can provide her feedback anonymously, while her hometown is ensured she's a valid resident.

Further Features

Private credentials offer all the features of a traditional public-key infrastructure. Moreover, they offer many features that traditional certificates don't. For instance, instead of revealing attribute values, users can choose to merely reveal that some predicate over the attributes holds. In the previous identity card example, Alice could transform her credential into one stating solely that her birthday is before 1994.

Private credentials also let users provide attributes in verifiably encrypted form to the relying party, so that they're available only to a dedicated trusted third party. This mechanism allows, for instance, investigation of abuse of the provided anonymity.

Assume Alice's hometown wants to conduct an anonymous poll, allowing each resident to participate only once. To enforce the latter, the hometown can ask its residents to enable a mode of credential transformation that lets it detect repeated use of the same credential. Of course, this mode preserves the anonymity and unlinkability for honest users.

Conclusion

Private credentials can form the foundation of electronic networks' identity layer, providing the same or even better security for the relying parties while respecting user privacy. The cryptographic literature contains a fair number of proposals on how to implement private credentials based on different cryptographic assumptions. In fact, the research community around privacy-protecting mechanisms is active and growing. Nevertheless, private credentials are ready for deployment in practice. Microsoft's U-Prove [4] and IBM's Identity Mixer [5] are two implementations of private credentials that are publicly available and have been demonstrated to be viable. Currently, both are being integrated and used for two pilots in the EU-funded project ABC4Trust (https://abc4trust.eu).

References

1. D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," Comm. ACM, vol. 24, no. 2, 1981, pp. 84–88.

2. S. Brands, “Rethinking Public Key Infrastructure and Digital Certificates—Building in Privacy,” PhD thesis, Eindhoven Inst. of Technology, 1999.

3. J. Camenisch and A. Lysyanskaya, “Efficient Non-transferable Anonymous Multi-show Credential System with Optional Anonymity Revocation,” Advances in Cryptology—Eurocrypt 2001, LNCS 2045, Springer, 2001, pp. 93–118.

4. Microsoft U-Prove Community Technology Preview R2, Microsoft, 2011; https://connect. microsoft.com/site1188.

5. “Identity Mixer,” blog; http://idemix.wordpress.com.