A PRIMER ON USER IDENTIFICATION

Dr. Stefan Brands
McGill University & Credentica
brands@{cs.mcgill.ca, credentica.com}

1. Introduction

An *identifier* is a piece of information that names or indicates a person, a process, an application, a location (such as a place on earth or a CPU memory address), a tangible object (such as a book, a text file, or a device), or any other type of entity or grouping of entities. Within a designated *context*, identifiers enable *relying parties* to distinguish between the entities they interact with; this is known as *identification*. A context can be a sphere of activity, a geographical region, a communication platform, an application, a logical or physical domain, or anything else.

1.1. User identifiers

User identifiers are identifiers that represent *users* (i.e., individuals or groups of individuals) in their interactions with relying parties. Users may present their identifiers verbally, on paper, on plastic cards, or in any other appropriate manner. *Electronic* user identifiers are electronically presented over data communication channels by user-operated computing devices such as PCs, laptops, mobile phones, and smartcards.

Within their designated context, user identifiers serve one or more of the following *purposes* for relying parties:

- Identifier as *contact address*: To enable relying parties to contact users, now or later, to deliver or retrieve services, goods, or information.

- Identifier for *security*: To enable relying parties to *blacklist* users who have engaged in unwanted behavior (so as to be able to deny them future access) and perhaps to enable relying parties to *trace* such users for accountability reasons.[1]
- Identifier as community *membership proof*: To enable relying parties to infer that users have been pre-approved in some sense (typically by means of a one-time *registration* process).
- Identifier as *account index*: To enable relying parties to build user *accounts* (also known as records or profiles) containing user-related information.
- Identifier as *account pointer*: To enable relying parties to retrieve user-related information from indexed user accounts.

In each of the latter three cases, relying parties can use the additional user information they learn to offer better personalized services, to improve business aspects (e.g., better inventory management or direct marketing), or to make better access control decisions.

1.2. Examples

To appreciate the contextual nature of user identifiers, consider the designated contexts and purposes of the following user identifiers:

- Digitized photographs, fingerprint scans, iris or retina scans, and DNA samples;
- Registered birth names, corporate names, and author pseudonyms;
- E-mail addresses, telephone numbers, mail addresses, and URLs;

[1] To trace a person means to unambiguously determine that person's "real identity" through a discovery process that starts with information that is linked to his or her "real identity." The notion of "real identity" is related to that of the self, which is "the total, essential, or particular being of a person." For the pragmatic purposes of this primer, to determine a person's real identity is understood to mean coming up with a description of a person that is sufficiently rich to enable entities that can match descriptions against physically present persons to single out one person that uniquely matches the description.

- User identifiers with ISPs, banks, utility companies, VPNs, and so on;
- Credit cards, debit cards, calling cards, and loyalty tokens;
- Employee badges, sports club membership cards, and hotel key cards;
- Social security numbers and health insurance numbers;
- Passport numbers and driver's license numbers;
- Online usernames (e.g., for instant messaging and chat rooms);
- Mac addresses, IP addresses, smart card serial numbers, Bluetooth identifiers, GSM IMEI numbers, RFID tag identifiers, and other addresses of networked devices that represent users; and
- Cookies, X.500 Distinguishes Names, and SSL certificates.

1.3. Single-domain user identifiers

As the examples illustrate, users traditionally are represented in their interactions with relying parties by a plurality of *local* user identifiers with incompatible formats:

- The symbols that make up an identifier must be *meaningful* to the relying party. Humans are not good at memorizing and recognizing binary strings, while computers are not designed to handle non-numerical data. Thus, traditionally the *encoding* of user identifiers depends on their context.
- A larger *namespace* is required for distinguishing between increasing numbers of users, but lengthier identifiers are less desirable for human processing. Thus, traditionally the length of user identifiers is related to their designated context.
- Absent agreed-upon standards for encoding and generating identifiers, relying parties are likely to use proprietary formats for user identifiers. They may even do so deliberately in order to counter impersonation attacks based on leveraging user identifiers from other domains with a similar look and feel.

Thus, the traditional heterogeneity of communication platforms and the lack of connectivity have historically created an abundance of single-domain user identifiers that are designed to be relied on only by one or a few relying parties that all trust each other to not violate each other's security and privacy interests. Within such a *trust domain*, the only threats that relying parties are concerned with are attacks originating from users and *outsiders* (such as wire-tappers). An example of a single trust domain is a small company with several branches that rely on the same employee identifiers.

Users and relying parties each have their own security and privacy concerns at stake with respect to how single-domain user identifiers are *generated*. Traditionally, user identifiers are either self-generated by their users or certified by relying parties in the form of identity tokens that are endowed with one or more security features. We discuss these two approaches separately in the next two sections.

2. Self-generated user identifiers

Self-generated user identifiers are, as the name implies, generated by their own users (or by their computers). Online usernames are a typical example.

2.1. Privacy of users

User privacy is about the ability of users to minimize what user-related information relying parties can learn beyond what users choose to explicitly disclose. This boils down to a statistical inference problem:

- The difficulty of *statistically correlating* presented user identifiers is proportional to the quality of the randomness used to generate them. To maximize privacy within a designated context, identifiers must be generated independently and uniformly at random from the set of identifiers used within that context.

Randomly self-generated user identifiers cannot be *linked* to other identifiers of the same user and cannot be traced to the user's real identity.

- Any user-related information that relying parties can capture when user identifiers are presented increases their linking and tracing capabilities. This includes not only any personal information that users choose to disclose when presenting their identifiers, but also any circumstantial information (such as the user's location and communication device characteristics) that relying parties may be able to capture. For this reason, repeated use of the same identifier at different occasions typically disproportionately increases the linking and tracing ability of relying parties.

By minimizing the disclosure of additional information (especially of invariant personal information) and by using different self-generated identifiers in different contexts, users can minimize what relying parties can infer about them from the information they choose to disclose.

2.2. Security of users

Plain self-generated identifiers offer little in the way of security to their own users. What prevents another party from presenting someone else's user identifier? Generating user identifiers at random from a large name space reduces the risk of accidental identifier collisions, but does not address interception of presented identifiers nor misuse by relying parties. In many contexts, *impersonation* constitutes an unacceptable security risk to users; if users are fine at all with other parties using their user identifiers, they typically prefer to have control over such *delegated use*.

Secure transport of user identifiers (e.g., physical protection of a plastic identity token or *encryption* of an electronic identifier at the data transport layer) suffices to deal with *outsiders* who attempt to steal or copy identifiers in transit. This measure provides no security against *insiders*, however: nothing stops relying parties from *replaying* presented

user identifiers at other relying parties for their own benefit.[2] Even if there is only a single relying party, from the perspective of the user it may be unacceptable if that relying party is able to cause a false entry in an *audit log* by replaying a presented user identifier.

To counter replay attacks, users must *bind* their self-generated identifiers to something that cannot be stolen or copied, and that must be *verified* at presentation time. This "something" can be:

- Something tangible (e.g., a plastic card with unforgeable *authentication marks* that must be verified at presentation time).
- Something intrinsically unique to the person (e.g., a fingerprint or a handwritten signature specified on a plastic card that must be verified at presentation time).
- Some private knowledge that can be verified at presentation time but that is never disclosed by the user.

The first two techniques require physical proximity at presentation time and are cumbersome to self-implement. The third technique, which applies only to electronic identifiers, does not suffer from either of these drawbacks. It works as follows.

To protect a plain self-generated identifier, the user's device digitally signs it with the *private key* of a randomly self-generated cryptographic *key pair*. At presentation time, the user transmits the identifier, the corresponding *public key*, and the pre-generated signature that attests to the binding between them. In addition, the privacy key is used to digitally sign a *nonce* (i.e., a fresh *challenge message*) of the relying party. This ensures that neither wiretapers nor malicious relying parties can copy or steal presented identifiers and use them at honest relying parties. For each plain user identifier a fresh random key pair should be generated, to preserve the unlinkability between them.

[2] This impersonation attack may or may not constitute a security threat to relying parties themselves.

Another benefit of self-certified electronic identifiers is that users can *digitally sign* documents as well as messages representative of their *actions* with respect to presented identifiers. The latter capability can be exploited to prevent relying parties from making false claims in audit logs and elsewhere as to the actions undertaken by a given user.[3]

In contexts where relying parties are fine with relying on random numbers as user identifiers, there is no need to include a plain "meaningful" identifier: a self-generated public key is with overwhelming probability a universally unique identifier by itself.

2.3. Security of relying parties

While self-protected identifiers offer privacy and security to their own users, they offer no security to relying parties. Whether relying parties have security concerns of their own depends on the context and purpose of presented user identifiers. Most transactions involve some kind of security risk for relying parties relating to user authentication, especially applications that rely on secure *access control* or *data sharing* between relying parties; in these cases, presented identifiers are relied on as secure account indices and account pointers. Even when user identifiers serve merely as contact addresses, liability issues may arise for relying parties when they deliver information or goods to the wrong users.

More generally, relying parties cannot securely rely on self-generated user identifiers regardless of their context and purpose, for the following reasons:

- Anyone can self-generate and present user identifiers.
- Users can self-generate any number of uncorrelated identifiers.

[3] This dispute resolution technique is not perfect, since it cannot be proved to an arbitrator that a particular user identifier corresponded to a particular user-generated public key. Certification of the binding by a party that must be trusted by both users and relying parties introduces new problems, as discussed shortly.

- Users can arbitrarily share self-generated identifiers with others.

As a consequence, relying parties cannot securely perform any of the five purposes of user identifiers – they are completely dependent on the honesty of users. In the next section we examine how relying parties can certify user identifiers in order to endow them with relevant security features.

3. Certified user identifiers

Certified user identifiers are, as the name implies, endowed by or on behalf of relying parties with one or more *security features* and optional *attributes*. (To *certify* means to give out a formal attestation to the truth, accuracy, and genuineness of something.)

3.1. Security features and attributes

Typical security features are:

- The format of the identifier has been approved in some sense.
- The identifier is unique within the specified context.
- The identifier cannot be copied ("cloned").
- The user identifier cannot be transferred to other parties (*non-transferability*).
- The user of the identifier has been approved in some sense.
- No more than *x* certified identifiers have been issued to the user.
- The "real identity" of the user has been associated with the identifier.

During the certification process, a certified user identifier may optionally be bound to one or more attributes (i.e., information of any kind), such as:

- An expiry date or a more general specification of the *lifetime* of the identifier.
- The maximum number of authorized uses.
- The designated use context.
- One or more designated purposes.
- Details about the strength of the certification process.
- User-related information (e.g., a username).

Attributes can be certified along with the identifier itself or be stored in an online account indexed by an account pointer that is certified along with the identifier. The latter approach allows for *dynamic attributes*, which change over the lifetime of the token. The former approach is suitable only for *static attributes*, but has the advantage of allowing relying parties to verify attributes off-line at presentation time.

Static attributes typically fall in one of the following categories:

- Additional information about security features.
- Use limitations for the identifier.
- User-related information attesting to community membership, entitlements, negative statements, and so on.

In all cases, certified attributes primarily serve for the security of relying parties.

3.2. Security of relying parties

The certification of user identifiers is accomplished by specifying them on *identity tokens* that have unforgeable *authenticity marks*. Users are required to present these identity tokens at presentation time, so that relying parties can *verify* the authenticity marks.

Authenticity marks are made as follows:

- For non-electronic user identifiers that are specified on paper documents, plastic cards, and other physical identity tokens, authenticity marks take the form of seals, handwritten signatures, intaglio printing, special paper, watermarks, security threads, color-shifting ink, holograms, and so on. Well-known examples of non-electronic identity tokens are passports, health insurance cards, driver's licenses, employee access cards, credit cards, and debit cards.
- For electronic user identifiers, authenticity marks can be established in one of two ways. If relying parties trust the tamper-resistance of the user's computing device, the user identifier and any attributes can be stored in the device together with a secret key that is used to authenticate the presentation of the identifier and its associated attributes. If the user's device is not tamper-resistant, the identifier and any associated attributes must be *cryptographically certified* by a message authentication code or a digital signature. In practice, tamper-resistance and cryptographic certification may be combined to ensure a degree of *fall-back security* in case one of the two mechanisms gets compromised.

For the security feature of non-transferability, at token issuing time one or more *biometrical characteristics* must be certified along. Photographs and handwritten signatures are the predominant methods, but more secure methods such as fingerprints, hand geometry scans, and iris scans are gradually coming into vogue. Relying parties can compare these certified biometric characteristics with fresh samples taken at presentation time. This technique applies to both non-electronic and electronic identifiers; in the latter case, the biometric clues can be stored within the device chip itself. To prevent a user from replaying a biometric scan of another user, users and relying parties must be in physical proximity.[4]

[4] One way around this is to rely on a tamper-resistant biometric scanner at the user's end that is able to detect replay. Alternatively, for biometric voice scanning the user can be asked from remote to utter a fresh challenge word or sentence. Both methods have clear security and flexibility limitations.

Relying parties can also ensure a degree of *non-repudiability* by requiring users to sign documents as well as statements that attest to the actions they undertake with respect to their identity tokens:

- For non-electronic identifiers, relying parties can compare handwritten signatures with handwritten signatures that have been bound to presented identity tokens.
- For electronic identifiers, users can be asked to digitally sign data. This requires user identifiers to be bound at certification time to user-generated asymmetric key pairs. At presentation time, users must present their identity token together with their public key, and use their private key to digitally sign messages. X.509 certificates are a well-known implementation of such *digital identity certificates*.[5]

In addition, when the presentation of user identifiers requires physical proximity, relying parties can rely on witnesses, photographs, and sound recordings.

3.3. Security and privacy of users

While identity tokens offer protection to relying parties vis-à-vis users, they offer no security for users:

- Relying parties can impersonate users by forging identity tokens under their names, possibly without risk of detection. (*Identity theft*.)
- Relying parties can falsely blacklist users so as to block their access to services.
- Relying parties can make discriminatory service decisions on the basis of user-related information that is stored in accounts indexed by the presented identifier.

[5] Like with self-generated key pairs, the public key itself can be used as the user identifier.

- Likewise, relying parties can populate user accounts with false or otherwise inappropriate user-related information.
- Relying parties can hook user identifiers up to wrong account information, so as to cause wrong decisions to be made about them.
- Relying parties can certify along unnecessary attribute information that can be used to discriminate against users at presentation time.

As well, users have no privacy vis-à-vis relying parties: relying parties can link and trace all user actions by simply comparing presented identity tokens against what was disclosed at certification time. When users must identify themselves at token issuance time (as required for non-transferability and other security features, as well as for the specification of user-related attributes), this enables relying parties to trace users in order to take discretionary punitive measures or to find out more about them.

In small-scale contexts, users may not be concerned about these powers of relying parties over them. This is particularly true when identity tokens are relied on by only a single relying party; in this setting, security risks are typically minimal and privacy is about anonymous access to services. When the same user identifiers are relied on by growing numbers of relying parties, however, the situation drastically changes.

4. The use of single-domain identifiers across domains

Efficiency imperatives are currently driving more and more relying parties to rely on the same increasingly secure identity tokens. In this section we examine the nature and implications of these trends, which feed off of each other.

4.1. Trends in technology and security

Technological advances in user computing and electronic networking in the past decade are currently causing a sea change in how user identifiers are generated and processed:

- There is an increasing reliance on electronic identifiers. This trend is both driven and enabled by the fact that users and relying parties are increasingly represented by chip-enabled devices in their interactions with the world around them. Electronic identifiers bring tremendous benefits for relying parties and users alike: they can be generated, managed, presented, verified, and processed at lower cost, with extreme efficiency and accuracy, and without human intervention.
- The same user identifiers are relied on by more and more relying parties with increasingly weaker trust relations. The reuse of identifiers across trust domains saves relying parties and users from repeatedly incurring set-up hassles and related costs. At the same time, advances in on-line connectivity are creating increasingly powerful data sharing imperatives within and, increasingly, across trust domains; this in turn drives the use of identifiers as account indices and account pointers to facilitate the linking of user accounts.

As the traditional heterogeneity barriers of the physical world are disappearing and relying parties increasingly rely on the same user identifiers, rogue users can inflict much greater damage. This holds especially true for electronic identifiers that are fully processed by computer systems. In this environment, a single user may be able to rapidly mass-replicate an attack across all domains that rely on the same user identifiers. Thus, the increasing reliance on the same user identifiers by growing numbers of relying parties is in turn creating a growing demand for identity tokens with stronger security features.

This one-sided demand for better security manifests itself in different ways:

- Self-generated identifiers are gradually phased out.

- Users are increasingly required to disclose their “real identities” at certification time.
- More and more user-related attributes are bound to identity tokens.
- Tamper-resistant user devices are becoming increasingly common.
- Biometric mechanisms to prevent transferability are seeing wider adoption.
- Relying parties within and across trust domains increasingly pool together audit trail data relating to presented user identifiers in order to detect fraud patterns and to be able to centrally blacklist users.

This in turn creates even more incentives for users and relying parties to reuse identity tokens across domains:

- Validating “real identities” and issuing tamper-resistant user devices are costly efforts that are best handled by dedicated authorities.
- The liability risks for certification authorities rapidly go up as more parties rely on the same identity tokens. Again, dedicated authorities are better equipped to deal with these risks.
- It is cumbersome for users to have to use different tamper-resistant devices in different contexts.

To the extent that self-generated user identifiers are not phased out for security reasons, many are for technological reasons. User devices without a CPU cannot self-certify and randomly self-generate identifiers; thus, user identifiers must be generated and pre-installed by device manufacturers and application providers. User devices that do have a CPU may not have sufficient computing power to perform the cryptographic operations needed for self-certification. In either case, pre-installed user identifiers generated by third parties have implicit security features due to the fact that they are encapsulated in user devices in a manner that their users have very little control over. Indeed, hardware-bound electronic identifiers are often relied on by relying parties as certified identifiers.

4.2. Privacy and security implications

For users, there are severe security and privacy implications of the accelerating shift towards identity tokens that are reused by more and more relying parties. No longer is the damage due to identity theft (whether by outsiders or insiders) contained to a narrow domain, and no longer are identity thieves impaired by the inherent slowdowns in the non-electronic world. As the traditional segmentation of activity domains disappears, users also lose all control over the extent to which other parties can monitor their actions.

Ironically, the one-sided security battle of relying parties versus users has serious security, privacy, and autonomy implications for relying parties themselves. As identity tokens are relied on across trust domains, the security and privacy threats to relying parties shift from users and traditional outsiders to insiders – insiders in other domains, that is. In a single trust domain all relying parties fully trust each other, and so any one of them can play the role of the certification authority. In contrast, when identity tokens are to be relied on by multiple relying parties that do not fully trust each other when it comes to their own security and privacy, the certification function must inevitably be performed by a *certification authority* over which each relying party can at best have only partial control. This authority is either a new party that was not there before, or its role is played by a relying party in a different trust domain. The more autonomous relying parties rely on the same identity tokens, the less control each relying party has over the proper functioning of that certification authority; an increasing reliance on the same user identifiers implies a decreasing trust alignment between relying parties and the certification authority.

Consider the security and privacy-invasive powers that are housed in the certification authority, from the perspective of relying parties:

- It can create identity tokens with fictitious user identifiers for its own use, possibly without risk of detection.
- It can selectively impersonate the clients of a relying party by creating new identity tokens for existing user identifiers, possibly without risk of detection.

- It can by-pass many of the security features of the identity tokens that it issues, possibly without risk of detection.
- It can issue identity tokens that specify false attribute information, possibly without risk of detection. As well, it can arbitrarily change dynamic attribute information stored in on-line accounts, on a selective basis.

To the extent that the certification authority is involved in real-time in the interaction between users and relying parties (whether to validate the revocation status of presented identifiers or to reconcile or process secure audit data provided by relying parties), it gains additional powers. Specifically:

- For certified identifiers that were issued anonymously in batch (as needed for efficiency and/or security reasons), the certification authority can cross-correlate all of a user's identifiers.
- For identifiers issued to users who identified themselves at certification time (as required for many security features), the certification authority can trace and link all user actions in real time.

The incredible ease with which digital information can be certified and processed makes these threats much more serious for electronic identifiers than for non-electronic identifiers. In particular, a hacker, a virus, or any other kind of intruder who is able to gain insider status may be able to do any of the above – the threats do not only come from those in charge of operating the certification authority. It is one thing for a relying party to trust its own internal security measures, it is quite another to have to trust another organization's organizational security measures.

The increasing trust misalignment between relying parties and certification authorities is driven not only by the increasing reliance on identifiers across traditional trust domains: traditional trust domains themselves are rapidly disintegrating. Firstly, contract workers, mergers, outsourcing, geographic dispersion, and other corporate trends are increasingly

leading to separate trust domains within the same organization. Secondly, digitization and migration of corporate resources to online systems render organizations increasingly vulnerable to hackers and computer viruses with insider status. Even when the same identifiers are relied on only within a single organization, that organization may consist of multiple divisions each with their own concerns about security and privacy.

5. Conclusion

In sum, the automation of communication and transaction systems is causing increasing numbers of relying parties to rely on traditional single-domain identity tokens that are endowed with more and more security features. The result: users lose all the security and privacy benefits of self-generated user identifiers, while relying parties are increasingly vulnerable to attacks originating from other relying parties and from certification authorities.

These problems are caused by the fact that single-domain user identifiers are designed only to be secure in the traditional *two-party threat model* of users versus a single relying party. Thus, relying parties and users are confronted with a dilemma: either they live with the growing security and privacy risks caused by using single-domain user identifiers across trust domains, or they stop doing so and thereby forfeit all the efficiency and business benefits of reusing identifiers.

There is, however, a way out of this dilemma, namely by modernizing the technology of user identifiers itself. This requires the introduction of a new kind of user identifier that combines all the benefits of self-generated identifiers with those of certified identifiers, without inheriting any of their respective drawbacks: a *cross-domain* user identifier. But that is the topic of another paper.