Identity Management and Trusted Interaction in Internet and Mobile Computing [0]

Audun Jøsang
University of Oslo, Norway
Email: josang@ifi.uio.no

***Abstract*—The convergence of Internet and mobile computing enables personalised access to online services anywhere and anytime. This potent access capability creates opportunities for new business models which stimulates vigorous investment and rapid innovation. Unfortunately, this innovation also produces new vulnerabilities and threats, and the new business models also create incentives for attacks, because criminals will always follow the money. Unless the new threats are balanced with appropriate countermeasures, growth in Internet and mobile services will encounter painful setbacks. Security and trust are two fundamental factors for sustainable development of identity management in online markets and communities. The aim of this article is to present an overview of central aspects of identity management in Internet and mobile computing with respect to security and trust.**

I. Introduction

Trust is typically interpreted as a subjective belief in the reliability, honesty and security of an entity on which we depend for our welfare. In online environments we depend on a wide specter of things, ranging from computer hardware, software and data, to people and organisations. A security solution always assumes that certain entities function according to specific policies. To trust is precisely to make this sort of assumptions, so a trusted entity is the same as an entity that is assumed to function according to policy. A consequence of this is that a trusted component of a system must work correctly in order for the security of that system to hold, meaning that when a trusted component fails, then the systems and applications that depend on it can no longer be considered secure. An often cited articulation of this principle is: *"a trusted system or component is one that can break your security policy"* (which happens when the trusted system fails) [46] . The same applies to a trusted party such as a service provider (SP for short), i.e. it must operate according to the agreed or assumed policy in order to ensure the expected level of security and quality of services. A paradoxical conclusion to be drawn from this analysis is that security assurance may decrease when increasing the number of trusted components and parties that a service infrastructure depends on [13]. This is because the security of an infrastructure consisting of many trusted components typically follows the principle of the weakest link, i.e. in many situations the overall security can only be as strong as the least reliable or least secure of all the trusted component. We can not avoid using trusted security components, but the fewer the better. This is important to understand when designing identity management architectures, i.e. the fewer trusted parties required in an identity management model, the stronger the security that can be achieved by it [23].

The transfer of the social constructs of identity and trust into digital and computational concepts help in designing and implementing large scale online markets and communities, and also plays an important role in the converging mobile and Internet environments. Identity management (denoted IdM hereafter) is about recognising and verifying the correctness of identities in online environments. Trust management becomes a component of IdM whenever different parties rely on each other for identity provision and authentication. IdM and trust management therefore depend on each other in complex ways because the correctness of identity itself must be trusted for the quality and reliability of the corresponding entity to be trusted. IdM is also an essential concept when defining authorization policies in personalized services.

Establishing trust always has a cost [23], so that having complex trust requirements typically leads to high overhead in establishing the required trust. In order to reduce costs there will be incentives for stakeholders to "cut corners" regarding trust requirements, which could lead to inadequate security. The challenge is to design IdM systems with relatively simple trust requirements. Cryptographic mechanisms are often a core component of IdM solutions, e.g. for entity and data authentication. With cryptography it is often possible to propagate trust from where it initially exists to where it is needed [41]. The establishment of initial trust usually takes place in the physical world, and the subsequent propagation of trust happens online, often in an automated manner.

Research in identity management must deal with specific challenges such as usability, robustness, privacy, adaptability, scalability and trust. Due to the dynamic constraints of mobile computing environments, limited computing capability, restricted user interface for mobile human-computer interaction, power management issues, radio signal exposure, network tracking capabilities and user privacy requirements, all these challenges are amplified.

This article focuses on security and trust aspects of identity management in mobile and Internet computing. In order to set the scene Section II provides a high level overview of the ICT evolution of the past 40 years. Section III then describes the main principles of identity management in online environments. Section IV and Section V analyse current technologies

[0] IET Information Security. Volume 8, Issue 2, March 2014, pp.67–79.

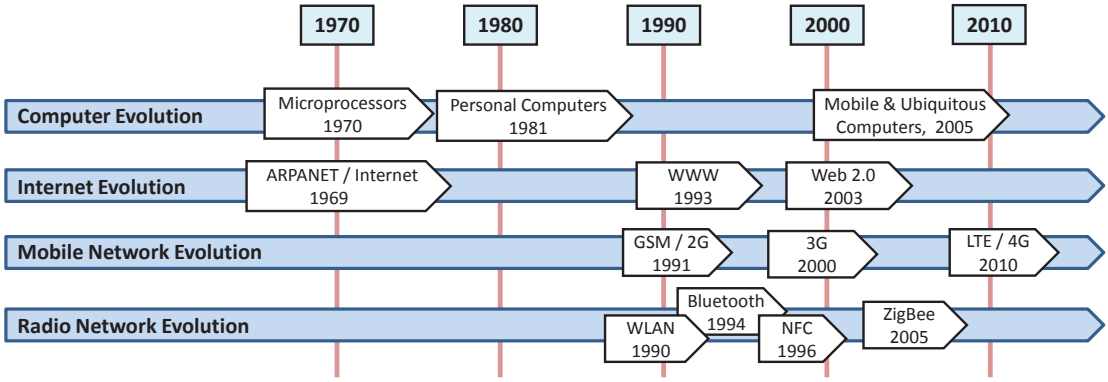

Fig. 1. Major innovation steps in the development of computer and network technologies.

for the management of user identities and SP identities respectively. Section VI provides a brief discussion of the main challenges for identity management with concluding remarks.

II. Background

The ability to exchange information through spoken and written communication was an essential factor for the emergence of advanced human civilisations. The invention of the printing press as a basis for mass distribution of information was another milestone in the evolution of civilisation. Automated processing of digital information in global computer networks is the latest addition to our information handling capability which has fundamentally transformed global markets and social communities, and thereby our whole civilisation, in a time span of only 40 years. Mobile computing brings this evolution to an ever higher level through permanent access to information anytime and anywhere.

A brief retrospective of the past 40 years is interesting for the purpose of identifying the major technologies that form the basis for Internet and mobile computing today. Fig.1 illustrates some important milestones in the evolution of computers, the Internet, and mobile and radio networks.

The first microprocessors were developed in the early 1970s[1], and organisations quickly started to use microprocessor-based computers for relatively intensive computational tasks. The personal computer was launched by IBM in 1981, with which digital computing became accessible to the average consumer. The mobile phone started to become a fully portable general purpose computing platform for the average consumer around 2005[2]. Also somewhere in the mid 2000s we saw the emergence of embedded computers, ubiquitous computing and the Internet of Things[3], which means that sensors, control units and other physical items equipped with microprocessors could be connected to a local network or directly to the Internet. During the period from the 1970 until now, Moore's law [39] has continued to be valid, which implies doubling of microprocessor complexity (in terms of number of transistors) every 18 months. By also accounting for the increase in microprocessor clock speed, we have been – and still are – witnessing more than a doubling of the capacity of standard computing platforms every 18 months. In other words, a microprocessor anno 2013 is at least $2^{28} \approx 400$ million times more powerful than a similarly sized microprocessor anno 1970. It also means that relatively powerful microprocessors can be made extremely small and thereby fit into even the tiniest objects.

The evolution of computer networks has gone through a parallel and similarly spectacular growth as that of microprocessors. The Internet, which started as ARPANET in 1969 was mostly used by researchers during the first decade. Uptake of the Internet by commerce and the average user happened around 1993 with the launch of the Mosaic web browser[4] that could graphically display digital documents encoded in HTML format, and that could navigate the World-Wide Web of linked online documents using the HTTP protocol[5]. The emergence of Web 2.0 from around 2003 transformed what originally was a Web of mostly static information that could only be passively read by users, into a Web of information that could be actively created and edited by users.

In the domain of mobile phone networks, the GSM[6] network, standardised by ETSI[7] in 1990 and commercially launched in 1991, was the first fully digital cellular mobile network. A cellular network consists of adjacent radio coverage areas called cells with a radio coverage radius of between 35km (macro-cell) and 10m (femto-cell), where each cell is served by a base station. Radio signals can be exchanged between the base station and the mobile phones within the cell, and the terminals can keep communicating while moving between cells. GSM is a so-called 2G (2nd Generation) network, where 1G (1st Generation) denotes earlier analogue mobile phone networks, mostly incompatible with each other,

[1] An early example was the Datapoint 2200 Microcomputer, from 1970.

[2] The Nokia 770 Internet Tablet, launched in 2005, was the first general purpose mobile computer, i.e. not just a portable computer.

[3] Wireless weather station, invented by Ambient Devices in 2003 was an early example of the Internet of Things.

[4] Mosaic was developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign (UIUC).

[5] HTTP: HyperText Transfer Protocol, Tim Berners-Lee, 1990.

[6] GSM: Global System for Mobile Communications. The original meaning of GSM was Groupe Spécial Mobile.

[7] ETSI: European Telecommunications Standards Institute.

that were used in many countries until they were replaced by GSM technology. A GSM subscriber is able to make and receive calls in most regions on our planet, a feature made possible by having compatible technology in every country, and through roaming agreements between operators in different countries, whereby subscribers of one operator are able to access – of course for an additional fee – base stations and mobile networks belonging to other operators in different countries. Interestingly, roaming between different operators within the same country is usually not supported because competing operators have no commercial incentive to establish roaming agreements with each other. Unfortunately, this can be a safety problem in a crisis situation in case one network stops functioning, and the lack of roaming agreements prevent subscribers of that network from making emergency calls through other networks. A simple workaround which can be used in emergency situations in case the home network is unaccessible is to remove the SIM card from the phone, thereby enabling emergency calls through any network. 2G networks are connection-oriented, meaning that a specific communication channel (radio frequency and time slot) is reserved for each call. GPRS-based[8] packet switching for data was introduced with the 2.5G specification (i.e. 2G with additional features) in 1998, and more widely deployed with 3G (3rd Generation) cellular networks from 2000, while still keeping the connection-oriented technology for voice. In LTE/4G (Long Term Evolution) cellular networks, launched in 2010, there is no longer any difference between voice and data, i.e. everything is transmitted as packed switched data.

Mobile networks have the particular characteristic that the mobile terminal always consists of (at least) two separate computers. The first computer is a separate mobile microprocessor commonly called SIM[9] card, which technically speaking is a UICC[10], that is owned by the telco operator. The second computer is the mobile phone itself, which is owned by the user. A specific SIM card which identifies a specific subscriber can thus be placed in (and removed from) any mobile phone. Fig.2 illustrates how the two completely separate microprocessors can be connected to the Internet indirectly through the Mobile/PSTN[11] network. Each of the components can be seen as a separate computing platform consisting of hardware and an appropriate software stack.

The fact that mobile phones contain microprocessors owned by different parties has the potential to cause conflicts of interest. Stakeholders in the computing and Internet industry will try to control aspects of the mobile network infrastructure of Fig.2 in order to protect and cultivate their own business models. Mobile phone vendors and mobile app developers are e.g. interested in having identity management applications installed on the mobile device, possibly supported by cloud services, whereas telcos might be interested in having these

[8]General Packet Radio Service

[9]SIM: Subscriber Identification Module.

[10]UICC: Universal Integrated Circuit Card. SIM is in reality a software module residing on the SIM Card.

[11]PSTN: Public Switched Telephone Network.

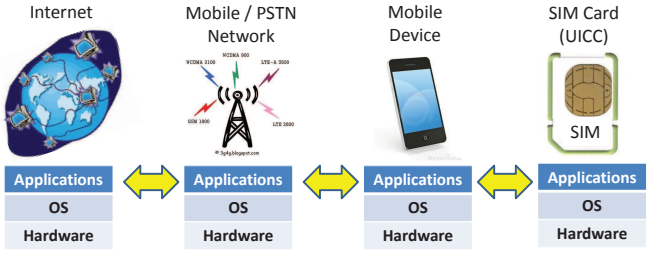

Fig. 2. Integration of mobile phone networks with the Internet

applications installed on the SIM card supported by the mobile network infrastructure. Furthermore, organisations that provide their employees with smartphones might want to control the types of applications that can be installed, and the location where data are being processed/stored, whereas employees might want to install applications of their choice on the smartphone. Hackers and criminals with malicious interests are of course in conflict with all the previously mentioned parties, as illustrated in Fig-3.

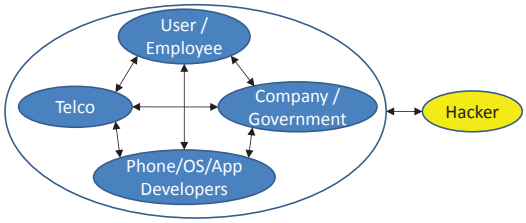

Fig. 3. Potential conflicts of interest in the mobile computing industry

There are several technologies for radio-based network access with limited or no support for mobile roaming between different radio cells/areas. WLAN[12] radio networks based on the IEEE 802.11 standard series is the most used digital radio access network, originally introduced in 1990. A given WLAN has an access point that covers an area with a maximum radius of about 100 meters, typically shorter indoors and potentially even longer outdoors. WLAN is increasingly replacing traditional cable-based LAN access, because it provides high convenience and less cable clutter in home and office environments. *Bluetooth* is a medium range point-to-point radio technology introduced by Ericsson in 1994 to provide wireless communication between local devices that are separated by a few meters, with an upper limit of around 10 meters for typical battery powered class 2 devices, and of around 100 meters for the more powerful mains powered class 1 devices. *NFC*[13] is a short range point-to-point radio technology for communication over a few centimeters up to a maximum of 20 centimeters. NFC was initially specified in 1996[14] and was developed in conjunction with passive RFID[15] tags that receive energy through a magnetic field, which make them able

[12]WLAN: Wireless Local Area Network

[13]NFC: Near Field Communication

[14]NFC is developed by the Near Field Communication (NFC) Forum

[15]RFID: Radio Frequency Identification

to support contactless communication for identification and tracking purposes. ZigBee is a low-cost, low-power, medium range radio mesh network technology introduced in 2005[16] that can support communication over a few meters up to a maximum of 75 meters. Since ZigBee is a radio mesh technology, not just point-to-point, it has a different application profile than those of WLAN, Bluetooth and NFC.

The combination of powerful ubiquitous microprocessors with highly flexible wireless networking technologies provide an extremely fertile ground for innovation and new business models. However, given the many stakeholders involved, as well as the increasing value and sensitivity of mobile and Internet applications, a large number of new and serious threats are emerging that must be balanced with adequate security measures. Identity management represents an important class of technologies for securing mobile and Internet computing.

Trust management can be seen as a general principle for facilitating collaborations among entities in online environments based on a combination of security services, as well as the collection, analysis and dissemination of information about the quality of resources and remote parties. Trust management stimulates collaboration and supports efficient decision making. However, without the proper support of identity management, trust management in online environments becomes impractical. Identity management and trust management depend on each other in complex ways, e.g., because reliable identification is needed for evaluating trust relationships between entities, and because the correctness of identity itself must be trusted for accurate search, discovery, recognition, connection and accounting. Identity management is also an essential factor for access authorization, authentication, access control and for personalized services.

Research in trust and identity management faces a number of challenges, e.g., with regard to usability, robustness, privacy, adaptability and scalability issues, etc. Due to the characteristics of open mobile computing environments, limited computing capability, restricted user interface for mobile human-computer interaction, power management issues, radio signal exposure, network tracking capabilities and user privacy requirements, these research challenges become rather daunting, but also highly interesting. This article provides an overview of identity management, and discusses the role that identity management plays for trusted interaction in mobile and Internet computing.

III. Identity Management Concepts

An identity is a representation of a specific entity such as a system, human user or a SP organisation within a specific application domain. An identity consists of a set of attributes that describe aspects of the entity. A digital identity is a set of digitally represented attributes of an entity, where one of the attributes typically is a name that uniquely identifies the entity within a domain. For example, the registered personal data of a bank customer, and also the customer's physical characteristics that bank staff are able to recognise, constitute aspects of the identity of the customer within the domain of that bank. However, only the digitally registered customer data represents the bank customer's digital identity.

An essential component of identity management is the namespace of unique names, which ideally should be global and memorable. Unfortunately, no namespace can be designed where names are unique, global and memorable simultaneously, which is one of the main challenges in identity management [10]. The legal name of a person or organisation is usually not unique, in which case it can not be used to uniquely identify a person or organisation. Instead some other unique name such as a number must be used as unique identifier, but number identifiers are typically hard to memorise, and are often only applicable within a specif domain.

Identity attributes can have various properties, such as being unique or ambiguous, transient or permanent, self-selected or issued by an authority, suitable for human interpretation or only by computers. The applicable attributes of an identity depend on the type of entity being identified. For example, gender applies to a person but not to organisations, whereas a trademark logo typically applies to a company but usually not to a person. The relationship between entities, identities and attributes/names are shown in Fig.4 below.

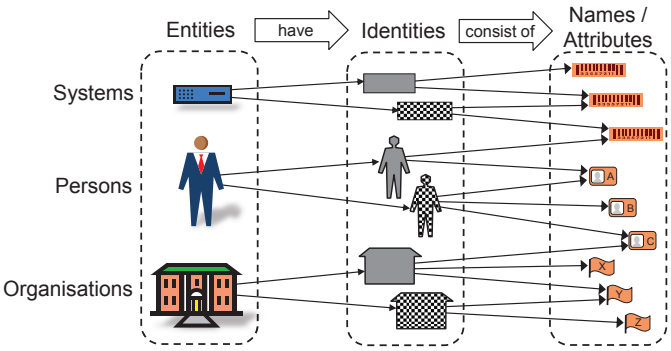

Fig. 4. Relationships between entities, identities and attributes

A simplifying assumption is that a single identity can not be associated with more than one entity. Having multiple owners of the same identity would not only be confusing, it would also create opportunities for misuse. Shared entities may exist, for example a family identity that includes several people. However, from a philosophical point of view a family identity is still owned by one entity, namely the *group* of people in the family, and not by multiple individuals separately.

A person or organisation may have zero, one or more identities within a given domain. For example, a person may have two identities in a school system because he or she is both a parent and a teacher at the school. The rules for registering identities within a domain determine whether multiple identities for one entity are permitted. Even if prohibited by policy, multiple identities for the same entity may still occur in the system, e.g. in error or as a result of a malicious

[16]Supported by the ZigBee Alliance, a group of companies that maintain and publish the ZigBee standard

intent. A person will of course have different identities in different domains. For example, a person can have one identity associated with being a customer in a bank, and another identity associated with being an employee in a company.

Identity management is fundamental to electronic interaction and collaboration by providing a basis for other security constructs, such as authorisation, access control, and reputation. Processes for IdM and access control can be grouped in three separate phases, which are the configuration, operation and termination phases, as illustrated in Fig.5.

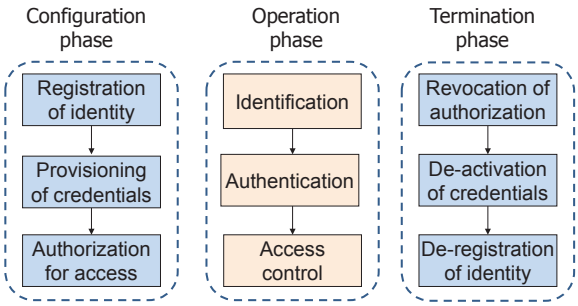

Fig. 5. Phases of IdM and access control

In the configuration phase, a digital identity is registered, and some form of authentication credentials is created and provided to the user.

The term *"authorization"* denotes the process of defining access policy for an entity subsequently to the registration of that entity. The authorization policy is typically translated into formal access control rules that can be automatically interpreted by a computer system. During operation, the access control step approves or rejects access based on these access rules (see e.g. [12, p.387] and [13]).

In the IT security literature the term "authorization" is often incorrectly used with the meaning of access approval, in the sense that the systems grants the user access to the requested resources, see e.g. RFC2904 [47]. However, the definition of e.g. confidentiality is that only authorized entities shall have access to information [20]. The incorrect interpretation of authorization (i.e. that the system grants access) leads to bizarre conclusions and disturbing inconsistencies. For example, assume that a hacker cracks a password, and then logs on to a system with the cracked password in order to steal confidential information. According to the incorrect definition of authorization the hacker would be authorized when accessing the system with the cracked but correctly typed password. An absurd consequence of this would be that theft of information by the hacker does not represent a breach of confidentiality. A further absurd consequence would be that if the victim took the hacker to court, the hacker could claim that he was authorized and thereby innocent, which technically speaking would be correct. In order to avoid the possibility of such scenarios it is important that the term authorization is only used in the sense of access policy definition.

In the operation phase, the subject entity – which e.g. can be a human user or a system entity – requests access to resources controlled by an object entity such as a server. An entity first claims its identity which then is authenticated by the other entity. Mutual authentication requires that the user authenticates the server, and the server authenticates the user.

After authenticating the user, the server control the acces by comparing the requested access with the stored access authorization policy, resulting in a decision to either approve or reject access.

During the termination phase, active authorizations for an identity are revoked and the identity de-registered so that the identity can no longer be used for accessing resources. When entities can have multiple identities in a domain it is possible that an entity with a de-registered identity still has other identities with active authorizations within the same domain.

The terms *client* and *server* are typically used to denote the peer entities in a communication session, see e.g. the X.800 standard [20]. In reality, client and server systems are only agents for legal and/or cognitive entities such as persons or organisations. The human user and the SP organisation are legal entities as well as cognitive entities. A person is assumed to be a cognitive entity because is possesses its own non-deterministic free will, in contrast to system entities that are considered to be deterministic without a free will. A legal organisation can also be considered to be cognitive in the sense that its actions are governed and executed by persons who legally represent the organisation.

The main concern of identity management has traditionally been efficient and secure management of user identities and corresponding credentials [25] for user authentication, where current technical solutions mainly consist of server side processes. However, digital identity management must cover both user-side and server-side identities [21], resulting in two very different types of identity management called *user IdM* and *SP IdM*, as illustrated in Fig.6. For each IdM type there are processes both at the user side and at the SP side.

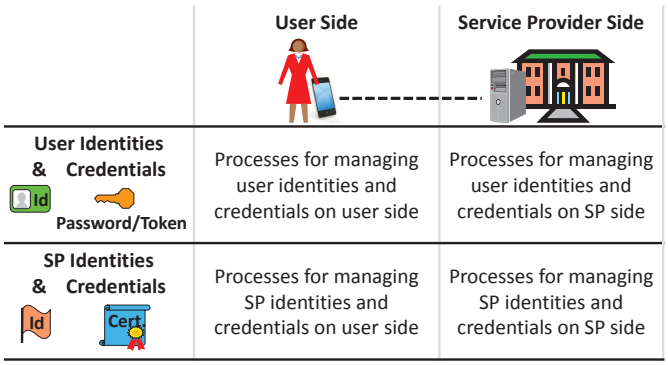

Fig. 6. Identity management types and processes

By taking into account the distinction between system entity (client or server) and legal/cognitive entity (person or organisation) there are in fact two entities on each side of a communication session, as illustrated in Fig.7. With this general architecture it can be seen that there are 8 distinct authentication classes. This analysis shows that the implicit

assumption of atomic entities made by the X.800 standard is a simplifying abstraction which thereby hides important aspects of entity authentication.

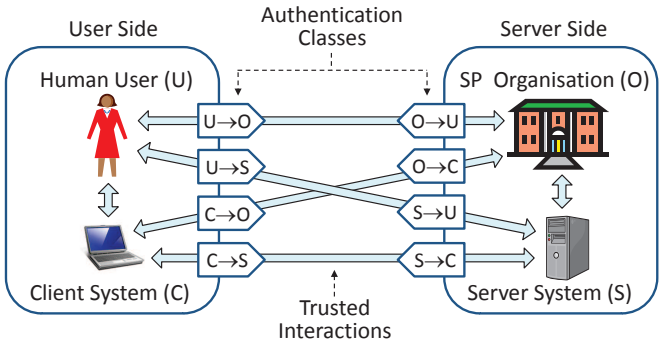

Fig. 7. General entity authentication classes

The distinction between the human user and the client system on the user side, as well as between the SP organisation and the server system on the server side leads to the conclusion that each of the 4 entities can be authenticated in 2 different ways, resulting in 8 different classes of peer entity authentication between the two sides, as illustrated in Fig.7 and described in Table I and Table II below.

Class	Authentication of user-side entities
$[U \rightarrow O]$	User (U) authentication by the SP organisation (O)
$[U \rightarrow S]$	User (U) authentication by the server system (S) (commonly called *user authentication*)
$[C \rightarrow O]$	Client (C) authentication by the SP organisation (O)
$[C \rightarrow S]$	Client (C) authentication by the server system (S)

TABLE I
AUTHENTICATION CLASSES FOR USER-SIDE ENTITIES

Class	Authentication of SP-side entities
$[O \rightarrow U]$	SP organisation (O) authentication by the human user (U)
$[O \rightarrow C]$	SP organisation (O) authentication by the user client (C)
$[S \rightarrow U]$	Server (S) authentication by the human user (U) (called *cognitive server authentication*)
$[S \rightarrow C]$	Server (S) authentication by the user client (C)

TABLE II
AUTHENTICATION CLASSES FOR SP-SIDE ENTITIES

Some of the entity authentication classes in Fig.7 are relatively impractical, such as $[C \rightarrow O]$ and $[O \rightarrow C]$, but they illustrate the generality of entity authentication when assuming non-atomic user and server sides. The authentication class $[U \rightarrow O]$ is e.g. practiced when authenticating customers over the phone by asking questions about customer number, date of birth, etc. The X.800 standard focuses on entity authentication classes $[C \rightarrow S]$ and $[S \rightarrow C]$. However, for online services applications the entity authentication classes $[U \rightarrow S]$ (user authentication) and $[S \rightarrow U]$ (cognitive server authentication) are the most relevant. The importance of these authentication classes emerges from the need for end-to-end security. In the typical case where a human user accesses an online service, semantic end-to-end communication takes place between the human user (U) and the server system (S). It is therefore pragmatic to require mutual authentication between those two entities. Traffic encryption and authentication between the server system (S) and user client (C) typically provides communication confidentiality, but can not provide cognitive server authentication in a meaningful way.

User identity management is frequently discussed in the identity management literature, whereas SP identity management is mostly discussed in the network security literature. This article discusses both types because they are both needed for mutual authentication during online service provision.

There are differences in the principles and technologies for user and SP identity management, so these are discussed separately in Section IV and Section V below.

IV. User Identity Management

The principles and technologies for managing user identities and credentials are different from the perspective of the server side and the user side. On the server side there exist mature solutions such as CRM software[17] and database directories such as LDAP[18] for this purpose. Handling of user registration and password recovery is automated and efficient, and is an integral part of user identity management on the SP side. However, there is little technical support for user identity management on the user side. There are various IdM models, which consist of specific technologies and network architectures for handling identities and credentials.

The silo IdM model assumes that each user has a separate identifier-credential pair for each SP that the user accesses. The term ”silo” reflects the characteristic that each SP operates an identity domain like a silo that is isolated from other domains. In this model the trust requirements between user and SP are simple and well understood in the form of specific security and privacy assumptions. In addition, the industry has had several decades of experience with this model, and users are familiar with it. The silo model has traditionally been used for all types of access to online services and resources because it has been relatively simple for SPs, but it is rapidly becoming unmanageable for users. The extreme growth in the number of online services based on the silo model has resulted in identity overload and password fatigue. This is a form of *tragedy of the commons* [15] which is the situation in case of a common resource, such as a common village green, without any centralised or common control. In the original case of the tragedy of the commons the lack of centralised control resulted in too many cows being allowed to graze on the common green. The analogy between the tragedy of the commons and the silo identity management model is that the human brains represent the ”common”, and all SPs want to store user Ids and passwords in each ”common brain”, but the SPs have no common control of how many identities and passwords that each brain has to remember. In this sense a password is the analogy of a cow in the tragedy of the commons.

[17]CRM: Customer Relationship Management
[18]LDAP: Lightweight Directory Access Protocol

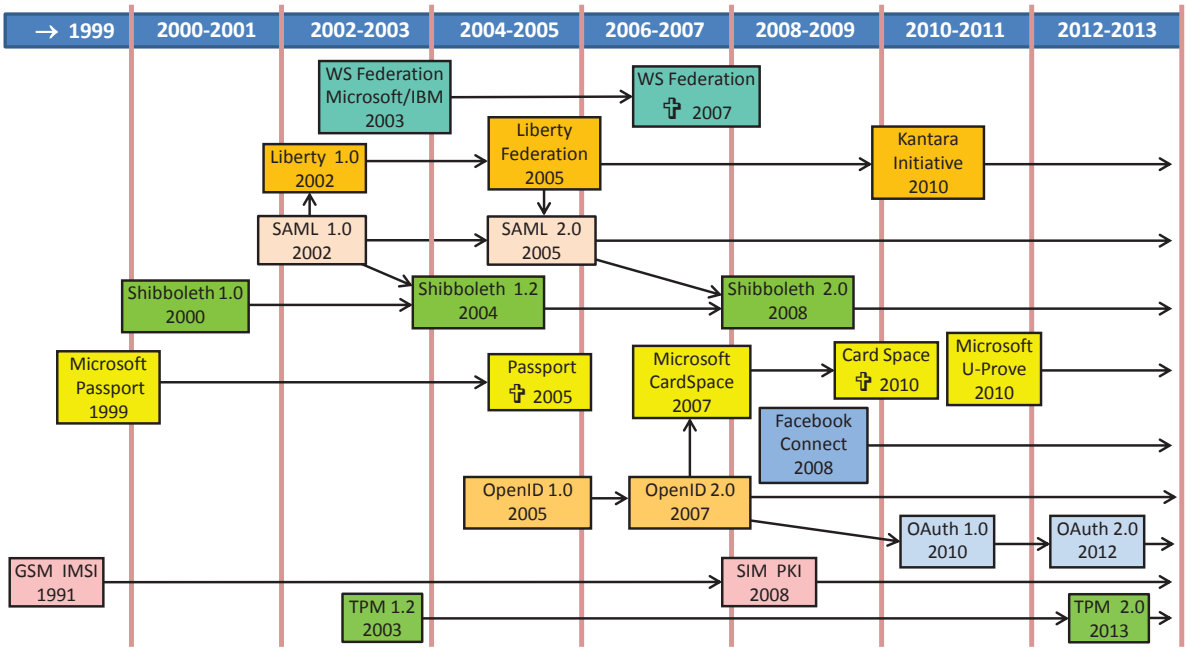

Fig. 8. Evolution of standards and technologies for user identity management.

In order to improve the user experience, and also for the purpose of coordinating related services from different SPs, new identity management models are being proposed and implemented. Some of these models have relatively complex trust requirements which can be a disadvantage for security. Fig.8 illustrates the evolution of models and technologies for user identity management over the last couple of decades.

Microsoft's Passport model introduced in 1999 was among the earliest attempts to solve the identity overload problem for open networks. Passport assumed that every user would register a unique Passport identity, and that SPs would let users access their services with this identity. This also implied that Microsoft would authenticate every user on behalf of the SP (Service Provider). The online industry generally rejected the Passport model because of the market power that this model would have given to Microsoft. Due to minimal market adoption Microsoft abandoned Passport in 2005.

The concept of federated identity management enables different SPs to authenticate users on behalf of each other, and implies some form of shared or mapped user identities so that users can be recognised across different service domains. Identity federation also implies legal contracts between parties that join a particular federation, in order to establish the required trust for accepting authentication and security assertions from each other. Shibboleth is a software implementation of identity federation, and was first released in 2000. In later releases Shibboleth has been adapted to comply with SAML[19], first published in 2002 which is the most prominent standard for identity federation. Parallel but incompatible standardisation initiatives such as Liberty Federation[20] started in 2002, and WS Federation[21] started in 2003, have been abandoned as a result of the widespread acceptance of SAML. Parts of the Liberty Alliance's technical specifications have been integrated in SAML 2.0 published in 2005. The Liberty Alliance changed its name to the Kantara Initiative in 2010, and operates as an industry policy group rather than a technical standardisation group.

OpenId is a model for identity federation where any party can act as an identity provider, and as such is completely open and distributed. OpenId has attracted considerable interest and adoption in the market. However, OpenId in its original form has severe trust and security weaknesses, so a specific usage of OpenId often puts restrictions on which parties are allowed to act as identity providers. CardSpace was a software module integrated with the Explorer browser to support a new version of Passport adapted to a federated identity model, and being compatible with the OpenId model. This effort was again rejected by the industry, and therefore abandoned in 2011. As a replacement for CardSpace Microsoft announced in 2010 the launch of the U-Prove identity and authorization model. U-Prove supports authentication of any type of identity attribute, such as age and gender, not just a name. In case a websites wants to restrict access to services and content based on user characteristics such as age, U-Prove would represent a solution for users to prove their age while remaining anonymous.

As a response to the need to allow closer integration of Web 2.0 services, the recent OAuth[22] standard specifies methods for third party access authorization that e.g. can give a social web-

[19]SAML: Security Assertions Markup Language, published by OASIS.

[20]Liberty Federation was published by the Liberty Alliance.

[21]WS Federation was developed and published by Microsoft and IBM.

[22]OAuth: Open Authorization.

site (such as Facebook) access to e.g. a user's address book at an email SP (such as Gmail) without giving away the password for the email provider. As a result of the explosive growth in popularity of social websites, combined with technical IdM solutions such as OpenId and OAuth, certain prominent social websites such as Facebook and Twitter have become *de facto* federated identity providers for large user groups, although this was never intended when these websites were first started. It is interesting to see that Microsoft was never able to become a major federated identity provider in the same way despite its efforts through Passport and CardSpace.

The massive popularity of the Facebook social community has enabled Facebook to define and impose their own IdM and authorization framework called Facebook Connect. Originally, Facebook Connect was only a user authentication platform that was quickly adopted by many other online service providers, enabling users to access multiple different online services by using their Facebook identities. More recently Facebook Connect has also implemented functionality similar to that of OAuth, which enables a user to authorize others to access specific personal data stored in online accounts belonging to the user.

The IMSI (International Mobile Subscriber Identity) introduced with GSM in 1991 is a unique identifier for mobile subscribers in any mobile network globally. The IMSI contains up to 15 digits, where the first 3 digits represent the MCC (Mobile Country Code), the next 2 digits represent the MNC (Mobile Network Code) within the country, followed by up to 10 digits representing the identification number of the subscriber within the network. The IMSI is privacy protected, so the identity of a subscriber is represented by a derived TMSI (Temporary Mobile Subscriber Identity) which changes every time the subscriber accesses a new mobile network. The IMSI stored on the SIM card, combined with related cryptographic technology running on the SIM card, provides reliable authentication of the subscriber, and has a great potential as a general identifier for user authentication in mobile and Internet applications [26]. However, due to the competitive advantage this would give to the mobile operators, other stakeholders in the Internet and financial industry have resisted initiatives aimed at using the IMSI as a general purpose identifier in online IdM.

As a first approach to more general SIM-based IdM several mobile operators have since 2008 started collaboration with e.g. national authorities and the financial industry to store private keys on the SIM card. A corresponding public-key certificate is issued by the Id provider, which e.g. can be the national government or a private bank, which thereby represents a PKI commonly called a SIM PKI. The private keys and their public-key certificates support strong user authentication and digital signatures. It is interesting to note that the IMSI is not the unique name stored in SIM PKI certificates, instead the Id provider will use a unique name related to their application, e.g. a social security number or a bank customer number. The low degree of integration between the GSM Id model, and Id models for Internet applications reflects the conflict between stakeholders in the industry as illustrated in Fig.3.

The TPM (Trusted Platform Module) is mentioned here as part of user IdM because it provides a method for cryptographic authentication of the client platforms. The term "TPM" is at the same time the name of a set of specifications [45] issued by the Trusted Computing Group, as well as the name of the hardware chip that implements these specifications. TPM chips are commonly installed in computer systems shipped since 2006, with the purpose of providing a robust hardware based mechanism for obtaining security assurance about various integrity aspects of systems. The TPM chip can e.g. be used to verify that the OS loader has not been modified by malware, and can also be used to authenticate (client) systems to external parties, which corresponds to client authentication (class $[\mathrm{C} \rightarrow \mathrm{S}]$) in Fig.7. TPM based system authentication relies on the EK (Endorsement Key) pair which is a public/private key pair that is unique for each TPM, generated and installed by the manufacturer or the vendor of the TPM. The EK pair is non-migratable and can only be used in carefully controlled ways. Because the TPM chip is physically mounted on the motherboard of the computer system, the EK-pair uniquely identifies the same system. There has so far been limited use of TPM functionality in the industry. One reason for this is that there currently is no PKI for validating data that has been signed by the private key of the EK-pair. Although the principle of TPM-based security and integrity assurance is powerful in theory, the logistics, practical and political issues related to implementing and rolling out TPM-based security solutions are relatively complex, and failures could totally disrupt a system protected by the TPM.

The term *User-Centric Identity Management* is often used in the literature with different meanings. In the most general sense it means identity management that enhances the user experience. The federated identity models described above fall under this category. In a more specific sense user-centric identity management means that there exists technology locally on the client side that assists the users in managing identities, as e.g. proposed in [25], and the term *local user-centric identity management* can be used to designate this interpretation.

Modern IdM solutions based on the above mentioned standards and technologies are currently being implemented by many solutions vendors and deployed in real environments [44]. However, there are still many challenges, as reflected by the many research projects focusing on IdM [34]. One particular area worth mentioning is that of password management which recently has attracted increased interest [17], [1]. While password management is a major security challenge for most people, it is paradoxical that there are hardly any standards or well recognised technologies for personal password management. It is for example a fundamental problem that passwords need to be typed into online terminals where they reside in system memory in cleartext format, which means that they are exposed to potential malware residing on the terminal.

A golden principle – which unfortunately is challenging to follow– is that passwords should never be present in cleartext

in client terminals that are online. Encrypting passwords would not help, because a password will be present in cleartext every time it gets decrypted to be used in an application. As mobile devices have become general purpose computing devices, they are affected with the same security vulnerabilities as traditional commercial computers [11], so it is reasonable to assume that every computer will be infected with some sort of malware after being connected to the Internet over some time. In order to avoid exposing passwords in online terminals, the passwords must be stored in an offline terminal, e.g. in form of the OffPad, an offline personal authentication device [33]. The challenge is then to find practical and user-friendly ways of integrating a device like the OffPad with online identity management solutions.

With regard to trust requirements in user IdM, the silo model is clearly the simplest because it only depends on trust between the SP and the user, whereas federated models require relatively complex trust relationships between multiple parties. However, the advantage of simple trust requirements for the silo model comes at the cost of poor usability and scalability. The aim of local user-centric IdM is to combine the advantage of usability and simple trust relationships.

The risk level of a specific service reflects the potential negative impact in case of wrong authentication. Several national governments have specified requirements for user authentication in order to balance the risk with appropriate authentication assurance. Frameworks for user authentication in the public sector typically specify a set of AALs (Authentication Assurance Levels). The requirements for each AAL are roughly harmonized across the various national or regional frameworks although there can be minor differences in interpretation. We briefly review the following five national/regional frameworks for user authentication.

- **US EAG.** Title: *Electronic Authentication Guideline* (NIST SP800-63-1) [5]. Describes technical requirements for user authentication assurance levels that are specified in the E-Authentication Guidance for U.S. Federal Agencies [4].
- **EU IDABC.** Title: *eID Interoperability for PEGS (Pan-European eGovernment services): Proposal for a multi-level authentication mechanism and a mapping of existing authentication mechanisms* [14]. In principle only a proposed framework for user authentication across the EU, but is nevertheless still widely adopted by subsequent EU policies and technical requirements, such as the STORK Quality authenticator scheme [19].
- **Norwegian FANR.** Title: *Framework for Authentication and Non-Repudiation in Electronic Communication with and within the Public Sector* [38]. User authentication framework for the Norwegian Government sector, similar to the NIST framework, but containing less details.
- **Australian NeAF.** Title: *National e-Authentication Framework* [8]. Detailed and well structured framework for user authentication published by the federal government of Australia. NeAF explicitly includes AAL-0 aimed at anonymous access as well as pseudonymous user authentication.
- **Indian ePramaan.** Title: *ePramaan: Framework for e-Authentication* [37]. Concise e-authentication framework published by the Indian Government. Includes AAL-0 similarly to the earlier Australian NeAF.

The assurance level alignment of the above referenced authentication frameworks is illustrated in Table III below. The specific terms used for each level may differ, and the same term is sometimes used for different levels in different frameworks, which can be a source of confusion. However it can be seen that there is a general consensus regarding the numerical levels. In order to minimize confusion the easiest is to simply refer to user authentication levels by their number, e.g. as AAL-3, because it has the same meaning approximately in every framework.

Authentication Framework	*Authentication Assurance Levels*				
EAG (USA) 2006	Little or no assurance (1)		Some (2)	High (3)	Very High (4)
IDABC (EU) 2007	×	Minimal (1)	Low (2)	Substantial (3)	High (4)
FANR (Norway) 2008	Little or no assurance (1)		Low (2)	Moderate (3)	High (4)
NeAF (Australia) 2009	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
ePramaan (India) 2012	None (0)	Minimal (1)	Moderate (2)	Strong (3)	Very Strong (4)

TABLE III
CORRESPONDENCE BETWEEN AUTHENTICATION ASSURANCE LEVELS.

The user authentication frameworks listed in Table III describe various factors that contribute to the robustness of the overall user authentication solution, such as

- **Authentication Method Strength:** The intrinsic robustness of the specific solution used for authentication, such as password based, token based or biometrics based authentication, as well as any combination of these to form 2-factor solutions.
- **Credential Management Assurance:** The estimated reliability and security of creation, distribution, usage and storage of the authentication credentials such as passwords, tokens and biometric profiles.
- **Identity Registration Assurance:** The thoroughness of the process for enrolling new entities that are to be authenticated by the system. In case an entity is to be registered with identity attributes from other identity domains, such as name and postal address, then the registration strength will depend on the correctness of these attributes when they are imported.

Authentication frameworks typically define a practical scheme for determining the AAL as a function of the basic authentication assurance factors. For example, in order to achieve a specific AAL, there are minimum requirements for the authentication method(s) used, for credentials management and for identity registration.

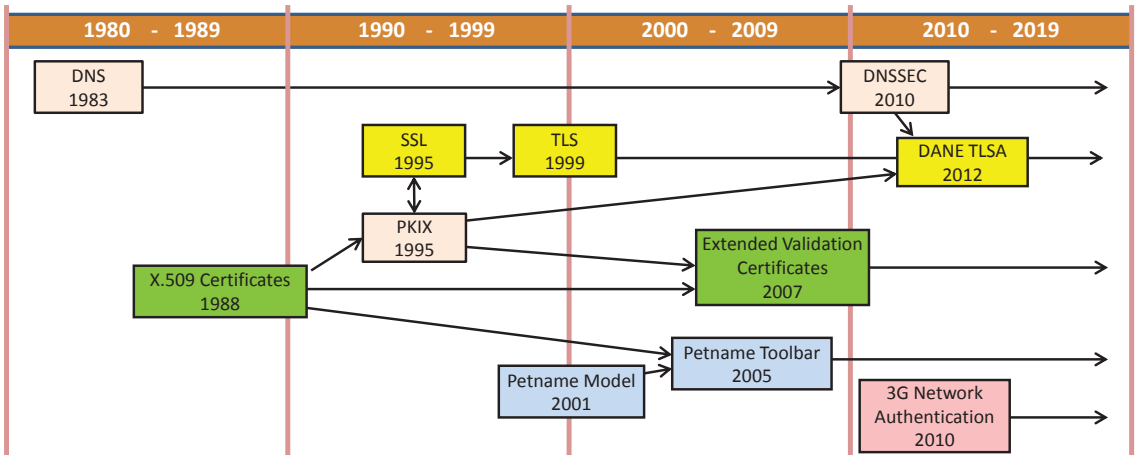

Fig. 9. Evolution of standards and technologies for SP identity management.

V. Service Provider Identity Management

Service providers have identities and credentials that need adequate management. Interestingly, the technologies used for SP IdM are very different from those used for user IdM, because traditional server authentication (class $[S \rightarrow C]$) mainly takes place on the transport layer, whereas user authentication (class $[U \rightarrow S]$) mainly takes place on the application layer. Fig.9 illustrates the evolution of fundamental standards and technologies for SP IdM over the last four decades.

The names used to uniquely identity SPs online are typically domain names and IP addresses that are managed under the DNS[23] which was introduced in 1983 to handle the growing number of host names on the Internet. The mnemonic quality of domain names with their mapping to numerical addresses became a practical necessity when in the early 1980s the Internet technology moved out of the research labs into mainstream society, and the user base started to grow exponentially. Many new companies and organisations became known to their customers by their distinctive domain names (e.g. amazon.com or wikipedia.org), and almost any other organisation needed to select a domain name that appeared as attractive as possible while at the same time being syntactically distinct from all other domain names. Consequently, domain names gained economic, social, cultural as well as political value, and often became the object of competition and dispute. This source of conflict still puts the management and the evolution of the DNS, a central element of Internet governance, under constant political and legal pressure. The DNS has to be carefully developed and managed to best serve its billions of users of all nationalities with their different languages and special needs.

Security threats against the DNS are many [3], [32], which reduces the assurance in DNS responses such as IP address translations from domain names. The technical solution to this problem is DNSSEC (DNS Security Extension), described in RFC4033 [2], which was designed to protect Internet resolvers (clients) from forged DNS data, e.g. due to DNS cache poisoning attacks. All answers received with DNSSEC are digitally signed. Through validation of the digital signature a DNS resolver gets the assurance that the information received is identical (correct and complete) to the information on the authoritative DNS server, i.e. that the information has not been corrupted. While protecting the integrity of returned IP addresses is the immediate concern for many users, DNSSEC can protect other information too, and RFC4398 describes how to use it to protect standard public-key certificates stored as CERT records [31], thereby making it possible to use DNSSEC to distribute such certificates. However, the scheme proposed in RFC4398 [31] does not exploit the potential of DNSSEC for direct certification of domain names and IP addresses. The recent RFC6698 [18] proposes the ”DANE TLSA” protocol which uses DNSSEC as a basis for distributing certificates for TLS. This would allow the elimination of trust required in third party CAs, and would therefore provide a significantly stronger security assurance than that which currently can be provided by the browser PKIX [7], the PKI with X.509 certificates currently used with web browsers. With a proper implementation and widespread deployment of the DANE TLSA protocol it would be possible to phase out the problematic browser PKIX described below.

Since its introduction in 1995, TLS/SSL combined with PKIX based on X.509 public-key certificates has been the most common form of server authentication. While TLS provides cryptographically strong server authentication, and optionally client authentication as well, it provides very weak real authentication in a semantic sense. This is because the actual implementation of TLS and PKIX has vulnerabilities that make them relatively easy targets of attack.

Fig.10 illustrates the PKIX implementation in browsers which consists of multiple hierarchical PKIs where each root certificate is stored in the browser, thereby enabling the browser to automatically validate any certificate issued under any of the roots [28].

The fact that PKIX in browsers consists of multiple separate independent PKIs means that its reliability follows the

[23]DNS: Domain Name System

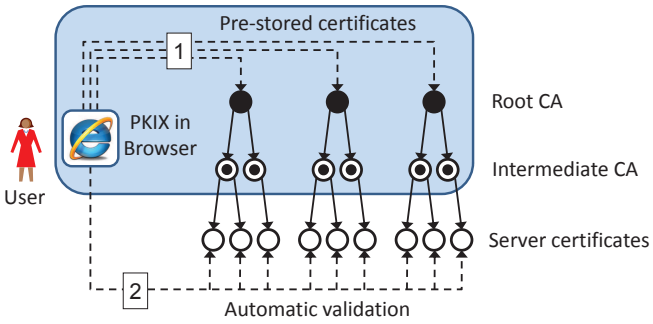

Fig. 10. Browser PKIX trust structure

principle of the weakest link, i.e. the whole PKIX is only as secure as the weakest of each separate PKI, and each separate PKI is only as strong as the weakest of its CA members. In other words, each CA is a single point of failure as explained in [16], [42].

The PKIX certificate market is primarily held by a small number of multinational companies. This market has significant barriers to entry since new providers must undergo annual security audits (such as WebTrust[24] for Certification Authorities) to be included in the list of web browser trusted authorities. Once approved as a provider of PKIX certificates for browsers, a CA will get its root certificate distributed with the major browsers to billions of users worldwide. More than 50 root certificates are installed and thereby automatically trusted in the most popular web browser versions. A 2009 market share report from Netcraft [35] showed that VeriSign and its acquisitions (which include Thawte and Geotrust) held a 47.5% share of the PKIX certificate market, followed by GoDaddy (23.4%), and Comodo (15.44%).

Crititicism of the relatively weak authentication assurance provided by PKIX prompted the CA industry to introduce the so-called Extended Validation (EV) Certificates in 2007. An EV Certificate is an X.509 public key certificate that contains an additional tag to indicate that it is issued according to a specific policy that sets requirements for verifying the real world identity of the requesting entity before an EV certificate is issued. In case of EV certificates the owner domain name can be traced back to a real world legal entity. When a https connection is based on an EV certificate the browser typically displays the address line on a green background instead of the normal white background. EV certificates are more expensive than ordinary certificates due to the required additional identity check of certificate owners. The CA industry promotes EV certificates by saying that they provide higher trust than traditional certificates. Some websites – such as online banks – feel forced to buy EV certificates because they assume that customers expect to see the green address line as a sign of trustworthiness, so that they could lose customers by not buying EV certificates. However, criminal organisations can also buy EV certificates, so an EV certificate obviously gives no indication with regard to the honesty and reliability of the certificate owner. EV certificates are also affected by the same vulnerabilities as those of traditional certificates described above. EV certificates represent a source of increased revenue for the CA industry, but do not make PKIX any more secure.

Despite being based on strong cryptography, there are a number of security exploits that TLS combined with PKIX can not prevent [27]. For example, phishing attacks normally start by sending email messages that trick people to access a fake web site masquerading as a genuine web site that e.g. prompts the user to provide user Id and password. In a technical sens the fake phishing website can be correctly authenticated through TLS. However, from a semantic point of view this is not authentication because the website's identity is different from that of the intended website. The problem is due to the poor authentication usability provided by current implementations of TLS [22], [24].

According to the X.800 standard [20], entity authentication is defined as: *"The corroboration that a peer entity in an association is the one claimed"*. So in case a victim user intends to connect to `https:\\www.paypal.com`, but is tricked into connecting to a phishing website called `https:\\www.peypal.com`, then the server certificate claims that the server identity is `www.peypal.com` which then is correctly authenticated according to X.800. Nevertheless, something is clearly wrong here, and the failure to capture this security failure indicates that the above definition of entity authentication is inadequate. What is needed is a stronger modality of authentication called *cognitive authentication* [30], [29] which can be defined as follows: *"The verification by the cognitive relying party that the identity of the other entity in a communication session is as claimed, and in addition the examination by the cognitive relying party of the true nature of the other entity in order to decide if it is acceptable to connect to that entity."*

When analysing the current browser implementation of TLS from a security usability perspective it can be seen that it provides week assurance of cognitive server authentication, which is precisely why phishing attacks often succeed [30].

Most phishing sites are set up without server certificates, but some phishing sites do use server certificates that normally are automatically validated by the browser. Most attackers probably see server certificates as an unnecessary expense because phishing works fine without certificates anyway. The typical phishing victim often does not know the difference between a http connection (without certificate) and a https connection (with certificate and TLS). The above example clearly shows that current certificate based TLS implementations in browsers do not provide practical server authentication. It would make sense to use TLS in Anonymous Diffie-Hellman mode without server certificates because it provides communication encryption without the cost of buying certificates. However, the Anonymous Diffie-Hellman mode is disabled by default in most browsers, presumably because theoretically it does not support server authentication. Ironically, current browser implementations of TLS with certificates also do not support server authentication, so using TLS with or without

[24]http://www.webtrust.org/

certificate makes no difference in practice.

Reliable identification and authentication of online entities require globally unique names that can be understood by people. Domain names partially satisfy this requirement and have therefore been chosen to represent the online identity of organisations. However, confusion arises in case different domain names appear similar, or when an organisation uses multiple domain names. This is precisely the situation we saw in the phishing example above.

The fundamental problem is that, although domain names are designed to be readable by humans, they are not user-friendly for identifying organisations in the real world. Ordinary names are suitable for dealing with organisations in the real world, but not for global online identification and authentication. The consequence of this mismatch between names used in the online world and in the real world is that users do not know which unique domain name to expect when accessing online services. Without knowing which domain name to expect, authentication becomes meaningless. In other words, the users do not know what security conclusion to draw.

Three desirable properties of a name were identified by Bryce ”Zooko” Wilcox-O’Hearn in his influential web article published in 2001 [48]. According to Wilcox-O’Hearn a name should ideally be Global[25], Unique[26] and Memorable[27]. To be memorable, a name has to pass the so-called “passing bus test”[36]. That is, if one can correctly remember a name written on a moving bus for a definite amount of time, e.g. 10 minutes after the bus passed, then that name can be considered memorable. A name will be unique if it is collision-free within the domain [43] and has the property that it cannot be “forged or duplicated”or “mimicked”. Wilcox-O’Hearn also claimed with supporting evidence that no name could have all the three desirable properties simultaneously, and suggested to choose any two of them according to different scenarios. Any attempt to define a name space that combines all three properties could lead to the following problems:

1) Dependency on a third party which could monopolize the system and create a single point of failure [48].
2) Political and legal conflicts arising when a name becomes a trademark for different companies locally in different regions and those companies compete for the same name when it reaches the global scale [40].
3) Unintentional confusion between almost similar names, for example typical confusion between two web addresses, e.g. wikipedia.org and wikipetia.org, are widely exploited by cyber-squatting and typo-squatting attacks. Intentional confusion caused by e.g. phishing attacks represent serious security threats [43].

The triangle of Fig.11 where each of the three desirable properties of names are placed in one of the corners is commonly known as Zooko’s triangle, and represents the basic foundation for the Petname Model [48], [10].

[25]Called *Decentralized* in [48]
[26]Called *Secure* in [48]
[27]Called *Human-Meaningful* in [48].

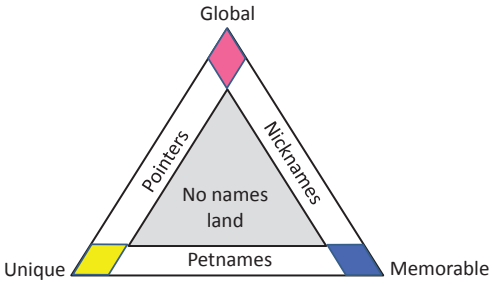

Fig. 11. Zooko’s triangle

The idea of placing the three properties at the three corners of a triangle can be explained as follows. In a triangle the three corners are never connected by a single line, only pairs of corners are connected. Placing those three properties in the three corners of the triangle provides a visual analogy to the fact that a name can only achieve two of the desirable properties at any one time.

The Petname Model allows all three desirable properties to be unified by using a combination of petnames and pointers. It requires each user to create personal petnames to represent globally unique pointers for services that she intends to access in the future. When a specific service is accessed, the user recognises it by its petname, not by its pointer. For this purpose a petname tool is typicality used, which makes it easy to manage the list of mappings between petnames and pointers, and which automatically translates pointers into petnames. The Petname Toolbar for the Firefox web browser has e.g. been developed to support petname based trust management [6]. The main motivation was to show the potential implementation of the Petname Model to counter phishing attacks.

Whenever we move from one website to another by clicking a hyper-link at the first site, there are two types of transitions that take place. One is the website transition that takes us to the next website. The second one is the transition of trust which enables us to retain or discard the trust relationship with the next website. We have different types of trust relationships with different entities, and may trust one entity more than another for specific trust scopes. For example, when a user wants to buy something from an e-commerce website, he may not trust to give his credit card credentials to that site, but he may trust PayPal. In this case, after choosing the item, the website may take him to the PayPal web page and he completes the transaction there. But the problem here is to make sure that the e-commerce site takes him to the right PayPal site, not to a fraudulent one. Currently, users are supposed to follow a set of steps to validate the identity of a website: 1) check if the target URL in the address uses the encrypted https protocol instead of the unencrypted http protocol, 2) check if the received server certificate is issued by some trusted authority, and 3) check if the domain of the accessed site matches the domain specified in the certificate.

Not only do these steps put a significant mental load on the user who therefore often omits them, they even fail to

consider whether the website is the one that the users intend to access [22]. These vulnerability then allow phishing attacks to succeed. Security is often a secondary consideration for users [9], where the primary goal normally is to complete a transaction to buy the desired item. This induces the user to ignore crucial security factors. So a malicious e-commerce site may exploit the technique of typo squatting, a technique in which similar domain names that only vary in one or two letters are utilized, e.g. as represented by `www.peypal.com`. When the fake website looks identical to the genuine PayPal website, most users will be tricked into believing that the fake website is genuine. That is, transition of trust may not take place as desired.

By using a petname system this type of confusion – and phishing attack – is avoided through enabling manual trust evaluation by the user while the transition takes place [6], [10]. More specifically, a petname system in combination with e.g. TLS can provide cognitive server authentication, which is precisely the security service needed to prevent phishing attacks.

2G/GSM mobile networks were subject to attacks through the use of rogue base stations and cryptanalysis of weak ciphers. This allowed attackers to intercept and control cryptographic material which in turn enabled the attackers to eavesdrop on the mobile radio channel and to create false SIMs. The original GSM security design did not support mobile network authentication, because the designers of GSM had overlooked the threat of false base stations. One of the priorities in the security design of 3G was therefore to provide mobile network authentication, as illustrated in Fig.9. This security feature is mostly transparent to users because they will normally not verify that the phone is connected to any specific mobile network. Instead the SIM will automatically verify that it is accessing a mobile network that is trusted by the user's own mobile operator. Assuming that the necessary feature is available on the mobile phone, the user is normally able to inspect the name of the mobile network accessed by the phone at any one time. However, this solution does not support strong human cognition of the mobile network identity. A mobile network us uniquely identified by the combination of the 3 digit-MCC (Mobile Country Code) and the 2-digit MNC (Mobile Network Code) relative to each country. Although this identifier is unique it is cognitively difficult for users to recognise a specific mobile operator by this number only. For that reason there is usually an associated network name, but this name is not globally unique, thereby creating a potential for confusion. In case the user requires strong assurance and cognition of the mobile network identity then the petname model could be used here as well.

In contrast to the existence of frameworks for user authentication as described in Section IV there are no government frameworks for SP authentication. Also, the lack of focus on SP authentication by the user is a blind spot in the Internet security literature. More focus on robust and user-friendly solutions for SP authentication is therefore required, e.g. by specifying requirements for SP authentication assurance levels [30]. From a usability perspective there is a great potential for developing solutions for combining both user and SP IdM on the user side. One approach in this regard is to use the OffPad – already described for user IdM in Section IV – to also handle SP IdM, e.g. by installing a petname application on the OffPad.

VI. Discussion

The intention of this article is to review identity management technologies and analyse their ability to support trusted interaction in Internet and mobile computing. In particular we have shown that there is a significant difference between user identity management and service provider identity management. Mobile applications have similar security and privacy issues as Internet applications. However, as new services are accessible through mobile computing new challenges emerge, ranging from usability, security and privacy. This article provides a snapshot of current technologies with new technologies that can provide trust assurance in identity management solutions. The evolution timelines show that this is a rapidly evolving area where the continuous innovation regularly brings disruptve market changes. Our study shows that identity management solutions with relatively complex trust requirements, such as federated identity systems, can provide a relatively good user experience, but that they can be challenging to implement and operate in a secure way. On the other hand, identity management solutions with simple trust requirements, such as silo identity systems, traditionally provide relatively poor user experience and do not scale well, but can provide strong security and are simple to implement. The challenge is thus to develop identity management solutions with simple trust requirements that at the same time scale well and provide satisfactory user experience.

References

[1] Bander AlFayyadh, Per Thorsheim, Audun Jøsang, and Henning Klevjer. Improving usability of password management with standardized password policies. In *7ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information (7th Conference on Network and Information Systems Security) (SAR-SSI 2012)*, Cabourg, May 2012.

[2] R. Arends, R Austein, M Larson, D. Massey, and S. Rose. *RFC 4033 – DNS Security Introduction and Requirements*. IETF, March 2005. Available at: http://www.rfc-editor.org/.

[3] Steven M. Bellovin. Using the domain name system for system breakins. In *Proceedings of the Fifth Usenix Unix Security Symposium*, 1995.

[4] Joshua B. Bolten. E-Authentication Guidance for Federal Agencies – Memorandum to the Heads of All Departments and Agencies (OMB M-04-04). Technical report, Executive Office of Tthe President, Office of Management and Budget, Washington, D.C. 20503, 2004.

[5] William E. Burr et al. Electronic Authentication Guideline – NIST Special Publication 800-63 Rev. 1. Technical report, National Institute of Standards and Technology, December 2011.

[6] Tyler Close. Trust management for humans. Waterken YURL, WaterkenInc. http://www.waterken.com/dev/YURL/Name/, 12 July 2004.

[7] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. *RFC 5280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. May 2008.

[8] Department of Finance and Deregulation. *National e-Authentication Framework (NeAF)*. Australian Government Information Management Office, Canberra, January 2009.

[9] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 581–590, New York, 2006. ACM.

[10] Md. Sadek Ferdous and Audun Jøsang. Entity authentication & trust validation in pki using petname systems. In Atilla Elçi et al., editors, *Theory and Practice of Cryptography Solutions for Secure Information Systems (CRYPSIS)*. IGI Global, Hershey, PA, USA, May 2013.

[11] Mohamed Ghallali, Driss El Ouadghiri, Mohammad Essaaidi, and Mohamed Boulmalf. Mobile phones security: the spread of malware via MMS and Bluetooth, prevention methods. In *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '11, pages 256–259, New York, NY, USA, 2011. ACM.

[12] Dieter Gollmann. *Computer Security, 3rd Edition*. Wiley, December 2010.

[13] Dieter Gollmann. From access control to trust management, and back - a petition. In Ian Wakeman et al., editors, *Trust Management V, 5th IFIP WG 11.11 International Conference (IFIPTM 2011)*, pages 1–8. Springer, 2011.

[14] Hans Graux and Jarkko Majava. eID Interoperability for PEGS (Pan-European eGovernment services) – Proposal for a multi-level authentication mechanism and a mapping of existing authentication mechanisms. Technical report, EU IDABC (Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens.), 2007.

[15] Gian Maria Greco and Luciano Floridi. The tragedy of the digital commons. *Ethics and Inf. Technol.*, 6(2):73–81, June 2004.

[16] James M. Hayes. The Problem with Multiple Roots in Web Browsers - Certificate Masquerading. In *7th Workshop on Enabling Technologies, Infrastructure for Collaborative Enterprises (WETICE '98)*, pages 306–313. CAUSA Proceedings, IEEE Computer Society, Palo Alto, June 17-19 1998.

[17] Cormac Herley and Paul Van Oorschot. A Research Agenda Acknowledging the Persistence of Passwords. *IEEE Security and Privacy*, 10(1):28–36, 2012.

[18] P Hoffman and J. Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. IETF, August 2012. URL: http://www.ietf.org/rfc/rfc6698.txt.

[19] B. Hulsebosch, G. Lenzini, and H. Eertink. Deliverable D2.3 - STORK Quality authenticator scheme. Technical report, STORK eID Consortium.), 2009.

[20] ITU. *Recommendation X.800, Security Architecture for Open Systems Interconnection for CCITT Applications*. International Telecommunications Union (formerly known as the International Telegraph and Telephone Consultantive Committee), Geneva, 1991. (X.800 is a re-edition of IS7498-2).

[21] ITU-T. *Recommendation Y.2720: NGN identity management framework*. International Telecommunication Union, Telecommunication Standardization Sector, Series Y: Next Generation Networks - Security, 2009.

[22] A. Jøsang, B. AlFayyadh, T. Grandison, M. AlZomai, and J. McNamara. Security Usability Principles for Vulnerability Analysis and Risk Assessment. In *The Proceedings of the Annual Computer Security Applications Conference (ACSAC'07)*, Miami Beach, December 2007.

[23] A. Jøsang, J. Fabre, J. Hay, J. Dalziel, and S. Pope. Trust Requirements in Identity Management. In R. Buyya et al., editors, *The Proceedings of the Australasian Information Security Workshop (AISW) (Volume 44 of Conferences in Research and Practice in Information Technology)*, Newcastle, Australia, January 2005.

[24] A. Jøsang, P.M. Møllerud, and E. Cheung. Web Security: The Emperors New Armour. In *The Proceedings of the European Conference on Information Systems (ECIS2001)*, Bled, Slovenia, June 2001.

[25] A. Jøsang and S. Pope. User-Centric Identity Management. In Andrew Clark., editor, *Proceedings of AusCERT 2005*, Brisbane, Australia, May 2005.

[26] A. Jøsang and G. Sanderud. Security in Mobile Communications: Challenges and Opportunities. In *The Proceedings of the Australasian Information Security Workshop*, Adelaide, February 2003.

[27] Audun Jøsang. Trust Extortion on the Internet. In *Proceedings of the 7th International Workshop on Security and Trust Management (STM 2011)*, Copenhagen, 2012. Springer LNCS 7170.

[28] Audun Jøsang. PKI Trust Models. In Atilla Elçi et al., editors, *Theory and Practice of Cryptography Solutions for Secure Information Systems (CRYPSIS)*. IGI Global, Hershey, PA, USA, May 2013.

[29] Audun Jøsang, Christophe Rosenberger, Laurent Miralabé, Knut Eilif Husa, Jérôme Daveau, and Petter Taugbøl. Local User-Centric Identity Management. *Journal of Trust Management*, (in press), 2013.

[30] Audun Jøsang, Kent A. Varmedal, Christophe Rosenberger, and Rajendra Kumar. Service Provider Authentication Assurance. In *Proceedings of the 10th Annual Conference on Privacy, Security and Trust (PST 2012)*, Paris, July 2012.

[31] Simon Josefsson. *RFC 4398 – Storing Certificates in the Domain Name System (DNS)*. IETF, March 2006. Available at: http://www.rfc-editor.org/.

[32] Dan Kaminsky. Details. Dan Kaminsky's blog at dankaminsky.com http://dankaminsky.com/2008/07/24/details/, 24 July 2008.

[33] Henning Klevjer, Audun Jøsang, and Kent A. Varmedal. Extended HTTP Digest Access Authentication. In *Proceedings of the 3rd IFIP WG 11.6 Working Conference on Policies & Research in Identity Management (IFIP IDMAN 2013)*, London, April 2013. Springer.

[34] Konstantinos Lampropoulos and Spyros Denazis. Identity Management Directions in Future Internet. *IEEE Communications Magazine*, 49(12):74–83, December 2011.

[35] Netcraft Ltd. Certification Services. Netcraft Report. https://ssl.netcraft.com/ssl-sample-report//CMatch/certs, 2010.

[36] Mark S. Miller. Lambda for Humans: The Pet-Name Markup Language. Resources library for E, http://www.erights.org/elib/capability/pnml.html, 2000.

[37] Ministry of Communications and Information Technology. *e-Pramaan: Framework for e-Authentication*. Government of India, Delhi, Version 1.0, October 2012.

[38] Ministry of Government Administration Reform. Framework for Authentication and Non-Repudiation in Electronic Communication with and within the Public Sector (in Norwegian: Rammeverk for autentisering og uavviselighet i elektronisk kommunikasjon med og i offentlig sektor). Technical report, Norwegian Government, 2008.

[39] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.

[40] Clay Shirky. Domain Names: Memorable, Global, Non-political? Clay Shirky's Writings About the Internet. http://shirky.com/writings/domain_names.html, 2002.

[41] G.J. Simmons. An introduction to the mathematics of trust in security protocols. In *Proceedings of the 1993 Computer Security Foundations Workshop*, pages 121–127. IEEE Computer Society Press, Los Alamitos, CA, USA, 1993.

[42] Christopher Soghoian and Sid Stamm. Certified lies: Detecting and defeating government interception attacks against ssl (short paper). In *Financial Cryptography*, pages 250–259, 2011.

[43] Marc Stiegler. Petname Systems. Technical Report HPL-2005-148, HP Laboratories Palo Alto, 15 August 2005.

[44] Jack Suess and Kevin Morooney. Identity Management & Trust Services: Foundations for Cloud Computing. *Educause review*, 44(5):25–42, September/October 2009.

[45] TCG. *Trusted Platform Module Library, Part 1: Architecture, Family 2.0 (draft for public review)*. Trusted Computing Group, Beaverton, Oregon, USA, March 2014.

[46] USDoD. *Trusted Computer System Evaluation Criteria (TCSEC)*. US Department of Defence, 1985.

[47] J. Volbrecht et al. *RFC 2904 – AAA Authorization Framework*. IETF, August 2000. URL: http://www.ietf.org/rfc/rfc2904.txt.

[48] Bryce (Zooko) Wilcox-O'Hearn. Names: Decentralized, secure, human-meaningful: Choose two. http://www.zooko.com/distnames.html, 2005.