In Atilla Elçi *et al.* (editors), *Theory and Practice of Cryptography Solutions for Secure Information Systems (CRYPSIS).* IGI Global, May 2013. ISBN13: 9781466640306.

Entity Authentication & Trust Validation in PKI using Petname Systems

Md. Sadek Ferdous[1], Audun Jøsang[2]
[1]*University of Glasgow, UK, Email: m.ferdous.1@research.gla.ac.uk*
[2]*University of Oslo, Norway, Email: josang@mn.uio.no*

ABSTRACT

Recognition of identities and certainty about identity ownership are crucial factors for secure communication in digital environments. Identity Management Systems have been designed to aid users as well as organisations to manage different user identities. However, traditional Identity Management Systems are primarily designed to facilitate the management of identities from the perspective of the service provider, but provide little support on the user side to manage organisational identities. Public-Key Infrastructures (PKI) is the primary tool in aiding users to manage such identities on their sides as well as to establish trust during online transactions. Nevertheless, the complexities and difficulties involved in managing and understanding such certificates from the general public's point of view are overlooked. This causes vulnerabilities that open up for serious attacks such as identity theft and Phishing. Petname Systems have been proposed for managing organisational identities on the user side in order to improve the user friendliness and to strengthen security. This chapter provides an analysis of the Petname Model by describing its history and background, properties, application domains and usability issues and explains how a Petname System can be effectively combined with the PKI to recognise identities and impose certainty by validating the user trust on those identities. The chapter also presents our analysis on two applications that integrate the Public Key Infrastructure with the Petname Model.

INTRODUCTION

Entity identification and trust are two important factors that help people decide whether or not to engage in transaction with other people in the real world. We humans inherit these qualities as part of our human endeavours in the society, and as our boundary of social interactions expand over time so does our ability to utilise those qualities to our benefit. But trust can misleads us while engaging in transactions with other human beings due to the complex and unpredictable nature of human behaviour, and when expectation does not meet in transaction, it results in erosion of trust. With the ever growing expansion of the Internet, technologies have enabled us to engage in transactions much like the way we transact in real world. However, with the absence of the face-to-face interaction, trust assessment through the Internet is typically much more challenging. At the initial growing stage of the Internet, the web and web-based services were not foreseen in its current form and the necessity of formal verification of entity identities was not felt. This led to the omission of the much needed Identity Layer. This causes the identification of entities to be very difficult in online world which in turn makes it difficult to establish and validate trust with other entities.

Authentication was subsequently added for verifying the correctness of claimed and assumed identities. Authentication requires prior registration of identities, and is based on a set of security mechanisms combined with a credential or security token. As authentication became necessary for accessing many online services, more and more identities and credentials were issued, and their management became problematic, both for service providers and for users. Identity Management (IdM, in short) Systems were introduced by the industry to facilitate the server-side management of user identities. Initially, the client-side management of user identities was not considered to be an issue. However, many people currently feel overloaded with identities and passwords that security policies require them to memorise. The growing number of identities that users need to handle and the inability of users to comply with credential management policies now makes client (user) side IdM a critical issue. It is important to consider that users need to manage their own identities as well as SP (Service Provider) identities. The latter aspect of IdM has received relatively little attention. Users have been provided with only PKI and digital certificates for identifying and authenticating SPs. In practice PKI on the Internet is used for automatic authentication of SP entities through their domain names. Although technically sound, PKI suffers from serious usability issues which make it difficult for general people to use it effectively and efficiently. This creates precisely the vulnerability that makes phishing attacks potent and successful. Petname Systems can be an effective solution against such threats. In this chapter we highlight the shortcomings of PKI and show how a Petname System can effectively be used to improve security and usability.

An essential part of an IdM is the namespace which provides a set of unique names (identifiers) for all entities it deals with. Different types of namespaces will have different properties. It is desirable that the namespace enables names to be 1) Global, 2) Memorable and 3) Unique (Called “Secure” in (Wilcox, 2001)). Unfortunately, no single namespace have all the three properties simultaneously (Wilcox, 2001). However, by combining a global namespace with a local namespace, all three properties can be combined (Miller, 2000). A so-called Petname System is a solution for achieving this. The combination of IdM and Petname Systems therefore seems to be an ideal choice for client-side Identity Management.

In this chapter, we present an extensive elaboration of our previous work on Petname Systems that can be found in (Ferdous *et al*., 2009). In addition, we focus on PKI and show how a Petname System and PKI can be combined for entity authentication as well as for establishing and validating trust on the web. The contributions of this chapter in comparison with the previous work are:

- An introductory section that provides definitions of basic terminology of Identity and Identity Management, aimed at audiences that are not familiar with Identity Management.
- A brief introduction on the PKI that describes its mechanisms and highlights its shortcomings.
- The background of the Petname System.
- Real world examples to familiarize readers with the abstract concept of the Petname Systems.
- Fundamental properties of Petname Systems.
- An analysis of how the Petname System and PKI can be combined to remove many shortcomings of the current implementation of PKIs.
- Security usability analysis of Petname Systems combined with PKIs.

The structure of the paper is as follows. Sec.2 explains frequently used basic terms that are necessary for understanding Petname Systems. We use the term Petname Model to denote the abstract properties of Petname Systems. An implementation of the Petname Model is then a Petname System.

Sec.3 provides a very short primer on PKI. It explains the mechanisms behind the digital certificate and techniques highlighting the inner working of the PKI for entity authentication. The section also underlines the shortcomings of the PKI System.

To understand the Petname Model it is essential to understand why Petname Systems were proposed in the first place. The Petname Model was formally described by Marc Stiegler in his 2005 paper (Stiegler, 2005). The potential of the Petname Model, however, was discovered by different people in several successive steps. Elements of the fundamental Petname System concept are scattered among several papers and web articles, and the combined efforts of these authors have shaped the formulation of the Petname Model. Sec.4 aims to summarise the existing literature.

Sec.5 defines the Petname Model by outlining its different components and establishing the connections among them. A Petname System has several properties and its potential applications can span over several disciplines of computing and networking. A long list of properties as well as several application scenarios was listed in (Stiegler, 2005). Sec.6 formalises the properties in a more systematic way by dividing them into two broad categories: 1) Functional properties and 2) Security usability properties, as well as usability requirements. We analyse mechanism for integrating the Petname Model in PKIs in Sec.7. In Sec.8, different applications of the Petname Model are explained. Section 9 analyses the usability issues of two PKI based applications that utilize the Petname Model. Sec.10 provides some hints on potential future works on Petname Systems and concluding remarks are provided in Sec.11.

DEFINITIONS

Entity. An entity is a physical or logical object which has a separate distinctive existence either in physical or in the logical world (Wikipedia Entity, 2012). In the scope of this chapter, a person, an organisation or a machine (computer) operated by any person or organisation will be denoted as an entity. It is to note here that examples of entity could be used for both server-side and client-side. In traditional server-side identity management, an entity can be a person, another organisation or a computer, where the identity of the respective entity typically is managed automatically by the server. In client-side identity management, users manage their own identities as well as identities of SP organisations.

Identity. Different disciplines (Philosophy, Social Science, etc.) interpret identity in different ways. There are also different definitions of identity which can be quite complex to understand and sometimes even contradictory. By putting aside the philosophical debates and contradictory arguments, a simple but intuitive definition can be provided (Thanh & Jørstad, 2007): Identity is the fundamental property of any entity that declares the uniqueness or sameness of itself and makes it distinctive from other entities in a certain context. Readers who would want to explore the philosophical debates and other advanced concepts over Identity such as Identity over time, Absolute and Relative Identity, etc. please refer to the work of (Geach, 1967), (Sider, 2000), (Deutsch, 2008), (Gallois, 2011) and (Gallois, 2012).

In general, an entity can have multiple unique identities, but a unique identity cannot be associated with more than one entity. Each identity can consist of multiple attributes (Jøsang & Pope, 2005). Here, the same attribute can be associated with multiple identities. Attributes can have different properties, such as being transient or permanent, self-selected or issued by an authority, suitable for human interpretation or only by computers. The possible attributes of an identity may differ, depending on the type of real world entity being identified. For example, gender applies to people, but not to organisations; stock exchange listing applies to a company, but not to a person. Some attributes are shared and some are unique within a given identity domain (a logical boundary governed by a single organisation), but each identity has to be unique within a specific identity domain. It is usually the case that one of the attributes is a unique name within a specific namespace, in which case that attribute cannot be shared with other identities. The unique name is then used to identify the identity within the specific identity domain. The diagram below illustrates the conceptual relationship between identities, entities and attributes. It should be noted that the distinction between identity and name is blurred in common language usage. The term “identity” often used in the sense of “name”, especially when an identity is recognised by a single unique name within a given context. For clarity, the terms “identity” and “name” will be used with their separate specific meanings throughout this chapter.

Human beings are equipped with the ability to intuitively identify an entity based on an ad hoc set of characteristics and also in varying contexts, but machines are not. To enable a machine to identify other entities, Digital Identity is required.

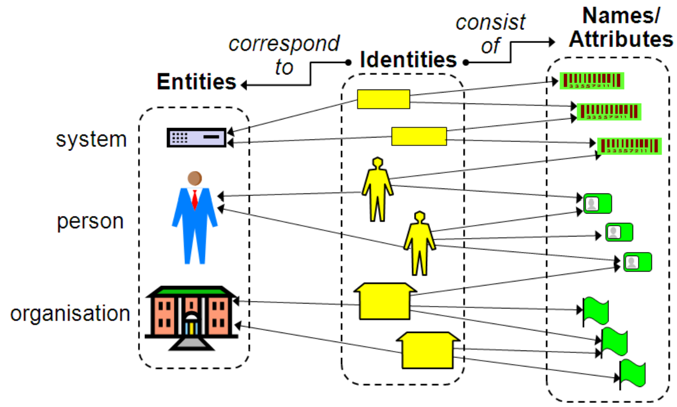

Figure 1. Relationship between Entities, Identities and Names

Digital Identity and Name. The digital encoding of the attributes of an identity can be defined as a digital identity. It is the representation of an identity in a form that is suitable for representation and processing in computer systems. The digital encoding of a name is then a digital name. In all types of digital communication (Internet, telecommunication) digital names are being used in the form of the URL, user-id, phone number, etc. In many digital communications, digital name uniquely identify an entity and are confused to be an identity. Like the identity and the name, digital identity and digital name should be treated separately. For a good introduction to the concepts of identity and digital identity, see (Jøsang & Pope, 2005), (Thanh & Jørstad, 2007).

Identity Management. Id management consists of technologies, policies and practices for recognising and authenticating entities in online environments. All parties that engage in the online activities have identities that need to be managed. In particular, not only user identities, but also the identities of servers and SPs must be managed. Given that Id management must cover both the identities of the user and service provider, and given that there are always two parties involved (the relying party or the service provider and the target) it is necessary that the Id management have a component both on the client and the server side. This leads to four main types of Id management, as illustrated in table 1. Type 1A Id Management was the first form of Id management, where SPs typically implemented the Silo Model. In the Silo Model, each service is exclusively managed by a separate SP that provides a separate identity service, so that each user must maintain separate user-id and password for each service (Jøsang, Al Zomai, and Suriadi, 2007). THs silo model is still the most widely used IdM model on the Web. Type 1B Id management focuses on how users manage their own identities, and typically consists of memorising user names and passwords, although various types of password wallet applications have been around for many years. Id management of Type 2 focuses on the management of SP identities, where Type 2A is about how SPs manage their own identities, and Type 2B is about how users manage SP identities. Unfortunately Type 2 Id management is hardly ever discussed, although there are serious issues with current Type 2 Id management, as e.g. shown by the relatively high success rate of phishing attacks. The industry's attempt to solve this problem is to compile blacklists of "bad" server names (some commonly used blacklist sites are: Spamhaus (`www.spamhaus.org`), Spews (`www.spews.org`), DSBL (`www.dsbl.org`), MAPS (`www.mail-abuse.org/rbl/`), etc.) that can be used for triggering warnings in browsers, which in fact represents a form of trust management, not Id management. Because server names are often meaningless to users, server authentication based on server certificates and SSL also becomes meaningless which we will show later. Only when server certificates are combined with petnames server names and server certificates become meaningful.

IdM Type 1A: SP-side management of user Ids and credentials	**IdM Type 1B:** User-side management of user Ids and credentials
IdM Type 2A: SP-side management of SP Ids and credentials	**IdM Type 2B:** User-side management of SP Ids and credentials

Table 1. Identity management types

The term "*User-Centric Id Management*" is often used with different meanings. In the most general sense it means Id management that improves the user experience. The so-called federated Id management models fall under this category. In a more specific sense, user-centric Id management means that there is local technology on the client side that assists users in managing identities, as e.g. proposed in (Jøsang & Pope, 2005), and the term local user-centric Id management captures this interpretation. Petname Systems resides on the client side, and therefore represent technology for local user-centric Id management.

Petname Systems are applicable in different domains (to be elaborated in subsequent sections), however, we mainly focus on how the combination of Petname Systems and PKI can provide support for Type 2B IdM, i.e. the management of SP identities on the client side. This specifically solves problems related to the authentication of Web site identities, so that it e.g. can be used to prevent phishing attacks.

PUBLIC KEY INFRASTRUCTURE

A PKI is a framework based on public key cryptosystems (also known as the asymmetric cryptography) and consists of a set of policies that governs how cryptosystems should operate and defines the procedure for generating and publishing digital certificates (Menezes, Oorschot, and Vanstone, 1997). In a public-key cryptosystems an entity generates a key pair known as the public key and the private key. The public key, denoted as *pub* hereafter, is intended to be publicly available while the private key, denoted as *priv* hereafter, is intended for the entity only and to be a kept as a secret strictly. RSA is the most widely used public key cryptosystem however other popular public key cryptosystems exist such as Diffie–Hellman key exchange, ElGamal Encryption, etc. (Menezes, Oorschot, and Vanstone, 1997).

Public key cryptosystems are used to exchange data securely (e.g. using encryption) over any insecure channel such as the Internet. It is suitable for a situation when it is not possible to exchange the shared secret (to be used as a key for encrypting data) between two parties *a priori*. In such cases, the data is encrypted before transmission in the insecure channel using *pub* of the recipient. Upon receiving the data, the recipient uses the corresponding *priv* to decrypt the data. Such mechanism would allow achieving data confidentiality over the transmission medium, however, does not guarantee any data integrity. To achieve data integrity, digital signature is used. To digitally sign a message, a hash function is used to generate a hash of the message which is then encrypted using the *priv* of the sender. This encrypted hash message usually accompanies the message. Upon receiving this pair (message and the encrypted hash message), the recipient uses the *pub* of the sender to decrypt the hash message, uses the same hash function to generate a hashed message of the message and then compares the hash message with the decrypted hash message. If they are equal, the integrity of message is thought to be intact, otherwise not. Typically the operations of encryption and signing are combined to achieve confidentiality and data integrity altogether. To visualise the scenarios, let's assume that we have two parties Alice and Bob and Alice wants to send an encrypted message to Bob accompanied by the digital signature over an insecure channel. They first generate the corresponding key pair (pub_{alice}, $priv_{alice}$) and (pub_{bob}, $priv_{bob}$) respectively. They also share their public keys with other so that Alice gets hold of pub_{bob} and Bob gets hold of pub_{alice}. We also need three operations encryption, decryption and hash and will be denoted by enc(message, key) and dec(message, key) where message is the message to be encrypted and the key is the encryption key and hash(message) respectively. Now the following operations take place:

i) Alice hashes the message m using the hash operation to generate $hash_m$ = hash(m).
ii) Alice encrypts the $hash_m$ using the $priv_{alice:}$ $enchash_m$ = enc($hash_m$, $priv_{alice}$).

iii) Alice combines a pair of the message and hash: pair = (m, $enchash_m$).
iv) Alice encrypts the pair with the pub_{bob} to generate e: e = enc (pair, pub_{bob}).
v) Alice transmits e to Bob over an insecure channel.
vi) Bob uses $priv_{bob}$ to decrypt e and get the pair: pair = dec (e, $priv_{bob}$).
vii) Bob retrieves the message m and $enchash_m$ from the pair.
viii) Bob uses the same hash function to generate hash´: $hash_m´$ = hash(m).
ix) Bob decrypts the $enhash_m$ using the pub_{alice} to get back the $hash_m$: $hash_m$ = dec($enchash_m$, pub_{alice}).
x) Bob compares $hash_m´$ with $hash_m$.
xi) $hash_m´$ = $hash_m$ signifies that the message remained intact during transmission which completes the process.

One of the key challenges in the public key cryptosystem is to share the corresponding public key securely between different entities. PKI has been developed to enable a secure distribution of public with the help of a trusted third party called the Certificate Authority (CA in short) using a key component called Digital Certificate. A digital certificate is used to bind the name of a subject with a piece of information. The name of a subject can be of different types such as email address, DNS name, IP address, URI, etc. (Kesterson II, 2007). There are different types of digital certificates as well. Two major examples of different certificates are public key certificate and attribute certificate. When the name of a subject is bound with its public key, the certificate is known as the public key certificate whereas when the name is bound with an attribute of the subject, the certificate is called the attribute certificate. For the scope of this chapter, we are mainly interested about public key certificates. X.509 version 3 is the most commonly used industry standard for public key and attribute certificate (Housley, Ford, Polk, and Solo, 1999).

Each CA is responsible for generating and issuing digital certificates as well as revoking and archiving certificates that have been generated and signed by the CA. To preserve the authenticity of the binding, each digital certificate is signed by the CA. Additionally, the CA also generates and issues a self-signed certificate called the root certificate. Any entity that wants to use a digital certificate must trust the CA and possess the root certificate to validate the signature on the certificate. A transport layer protocol, known as the Secure Socket Layer/Transport Layer Security (SSL 3.0/ TLS 1.2), has been developed and widely used to exchange and validate digital certificates between communicating entities over the Internet which ultimately is used for entity authentication (Thomas, 2000). We briefly explain this process in the next section.

Entity Authentication using PKI

A public key certificate usually consists of the name of the entity presenting the certificate, name of the CA which has signed it, the public key associated with the entity, a validity period for the certificate and other information regarding the cryptographic algorithms used to sign this certificate. All these information are added to the certificate and signed by the CA. The first step after receiving a certificate is to validate it. A certificate is deemed valid if:

i) The certificate is signed by a trusted a CA.
ii) The certificate has not expired as indicated by the validity period in the certificate.
iii) The certificate has not been revoked by the CA.
iv) The content of certificate is unaltered which can be checked by the signature of the certificate. This requires possessing the public key of the CA itself.

Once the verification is complete, authentication phase begins. During the authentication phase, the sender has to prove its possession of the corresponding private key. This process can be quite complex in nature. However, a very general form is like the following:

The recipient sends a *nonce* (a one-time random data) to the sender. The sender is asked to encrypt it with the private key and return back to the recipient. Upon receiving the encrypted nonce, the recipient decrypts it with the sender's public key and compares it with the previously sent nonce. If both match, the sender is verified to be the holder of the presented certificate and the entity seems to be properly authenticated.

PKI Shortcomings

Despite being a sound technical system, PKI suffers from several critical flaws. A long list of such flaws can be found in (Ellison & Schneier, 2000), (Linn & Branchaud, 2004). Some of the major technical concerns are:

i) Retrieval of Keys and Certificates are difficult.
ii) Complexities in certificate processing.
iii) Management of trust in cross domain scenarios.
iv) Ensuring security at different ends.
v) Naming semantics

All these flaws lead into further significant usability vulnerabilities that allow attackers to launch different types of phishing attacks. The core problem here is the user's lack of knowledge regarding the domain name system and the user's inability to identify a fake domain name from the real one. Attackers 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 PayPa1 (the last character here is number 1) instead of PayPal. When the fake website looks identical to the genuine PayPal website, most users will be tricked into believing that the fake website is genuine. An attacker even may use a legitimate digital certificate for the fake domain and the browser will validate it without any problem whatsoever indicating that the server is fully authenticated by showing a closed padlock sign that it would usually show for other legitimate PKI-validated sites. Such a visual cue entices the user to establish trust with an invalid entity and thereby making the whole point of using PKI useless. In this sense, PKI with TLS is a technically sound solution, but lacks to provide any semantic meaning. In addition, users are suggested to follow a series of careful steps: 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 pose a significant mental load on the user, but also become very tedious when the users need to do it over and over again even for the same entity tempts the user to overlook any warning related to the problem of entity authentication (Jøsang, Al Fayyadh, Grandison, AlZomai, and McNamara, 2007). It is also observed that security is a secondary consideration from the user's point of view (Dhamija & Tygar, 2006). The primary issue is to conclude the transaction and buy the desired item. This leads the user to ignore the required steps and creates precisely the vulnerability that makes phishing attacks potent and successful. Things would improve considerably if the process of trust validation could be incorporated into the system which would allow users to establish trust like before and validate their trust when they would visit the website again. Currently, the Web and browser PKI do not have any such facility and we believe that the Petname System fits perfectly in this scenario.

BACKGROUND OF PETNAME SYSTEMS

The identity management process can roughly be divided into three phases (Wikipedia Identity, 2012):

i) Registration Phase: An identity with a unique name is created. A corresponding credential may also be supplied along with the name. The name and the credential are kept as long as there is a relationship between the entity and any relevant relying party.
ii) Operations Phase: The entity produces the name and the corresponding credential to the IdM system of the relying party for authentication and access control.
iii) Deregistration phase: When the relationship between the user and the relying party(ies) ceases, the identity is normally deregistered so that it can no longer be used for authentication or for accessing services.

In the first phase the Identity Management System (IdMS) has to generate and issue a unique name for each entity. The IdMS uses a namespace from which a name is selected or chosen. Simply, a namespace is a logical and abstract lot of names that can be used to uniquely select an entity. The main requirement for a name is uniqueness such that each name maps to a unique entity. It is obvious that the same name can be used to represent different entities in different namespaces. The larger the namespace, the more unique names it contains. However, a global namespace will normally suffer from the shortcoming that interpretation and memorization by humans becomes problematic. IP address is an example of such a global namespace. While it is possible to remember a few IP addresses, the mental load of remembering and accessing a large number of web sites by their IP addresses would be intolerable for normal users.

Three desirable properties of a name were defined by Zooko Wilcox-O'Hearn in his influential web article published in 2001 (Wilcox, 2001). According to him, a name should ideally be Global, Unique and Memorable (Called "Decentralized" "Secure" and "Human-Meaningful" respectively in (Wilcox, 2001)) (Internet, 2012), (Wilcox, 2001). To be memorable, a name has to pass the so-called "moving bus test" (Miller, 2000). That is, if one can correctly remember a name written on a moving bus for a definite amount of time, that name can be considered memorable. A name will be unique if it is collision-free within the domain (Stiegler, 2005) 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. Clay Shirky in his web article also came up with the same conclusion (Shirky, 2002). Any attempt to achieve all the three properties by any name could lead into the following problems:

i) Dependency on a third party which could monopolise the system and create a single point of failure (Wilcox, 2001).
ii) Political and legal conflict may arise when a name becomes a trademark for different companies locally in several region and those companies compete for the same name when it reaches the global scale (Shirky, 2002).
iii) Unintentional confusion between almost similar names, for example any confusion between two email addresses, e.g. rahim@bd.com and rahim@bd.net, can be very dangerous in a life critical situation. Intentional confusion caused by e.g. phishing attacks can also be disastrous (Stiegler, 2005).

A triangle where the three properties are placed in the three corners is commonly known as the Zooko's triangle, and represents the basic foundation for the Petname Model. Zooko's triangle is illustrated in Fig.2.

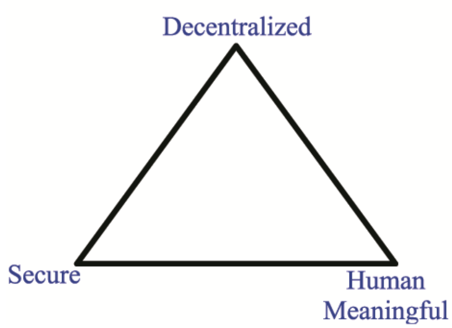

Figure 2. Zooko's triangle

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.

In 2000, Jonathan S. Shapiro, being inspired by the idea of Marc Miller et al. while at Electric Communities, described in a web article his scheme of adopting a system which utilised three types of naming conventions: Petname, True Name and Nickname (Shapiro, 2000). He adopted this idea for a configuration management system. A True Name is synonymous to a global unique name, the Nickname is a global memorable assigned name of an entity by its creator, and the Petname is a memorable and locally unique user-assigned name for that entity.

A few months later, Mark Miller published another article (Miller, 2000) in which he, for the first time, documented the structure of the Petname Model with three components: Petname, Key and Nickname. These three components are essentially equivalent to Shapiro's Petname, True Name and Nickname respectively. Miller suggested to use the term Key instead of True Name, and pointed out that the Petname Model satisfies all the three desirable properties of Zooko's triangle. This idea was actually elaborated by Marc Stiegler when he formalised the Petname Model. Tyler Close suggested adopting the term Pointer instead of Key (Close, 2003) and the term Pointer will be adopted instead of Key in this chapter. This topic will be described in greater details in the subsequent sections.

In 2003, Tyler Close of Waterken Inc. pointed out the possibility of using Petname Systems for better trust management (Close, 2003a). Waterken Inc. developed the Petname Toolbar for the Firefox web browser. The main motif was to show the potential implementation of the Petname Model to counter phishing attacks. According to Tyler Close, humans are not capable of managing the transition of trust from one entity to another in digital communications and this leads to identity-theft as a result of phishing attacks. The next paragraph explains his view on the rationale behind Petname Systems.

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 and 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. We may trust one entity more than another and with different scopes. As an analogy, when a user wants to buy something from an e-commerce website, she may not trust to give her credit card credentials to that site but she may trust PayPal. In this case, after choosing the item, the website may take her to the PayPal web page and she completes the transaction there. But the problem here is to make sure that the e-commerce site takes her to the right PayPal site, not to a fraudulent one. As mentioned earlier, users find it difficult to evaluate and validate their trust during transitions of website. That is, transition of trust may not take place as desired. So Tyler Close concluded that it was unwise to perform both transitions on the recommendation from a non-trustworthy entity, and therefore suggested to use Petname Systems to enable manual trust evaluation by the user while the transition takes place.

It is interesting to note at this point the relationship between identity management and trust management, where applying them improperly may lead to identity theft attacks. A realistic scenario can be used as an example. In the brick-and-mortar world, we come across different people where the different biological differences help us identify each person uniquely. Interactions with them enable us to decide who to trust. Sometimes recommendations play a crucial role. When our near and dear tell us not to trust somebody, we usually do not trust him or her, though this perspective may change over time. So we usually identify a person at first and place trust afterwards. Now in the digital world this scenario is somewhat different. To trust a digital entity, recommendation is the best and sometime the only option. We read website reviews, blogs, etc. and receive advice from relatives and friends on which digital entity to trust for online transactions. We may learn from them that there is a website www.paypal.com (there are also other trusted websites for online transactions) which we can trust for online transactions, even before we have accessed and identified it. Once the trust is placed, the only thing remaining is to identify the website which is truly the recommended one. It can also be the other way around, as for example we may browse and identify several unknown websites that are potentially suitable for a specific transaction, and then choose to transact with a specific one that subsequently will be trusted based on positive experiences. The first way obviously is the most hassle-free, and the second one requires the user to accept a certain risk of transacting with an unknown entity. Whichever is the best option, trust management and identity management are closely tied to each other when we try to derive a solution for the identity theft. As we will see, the Petname Model provides a solution for both scenarios.

In 2005, Marc Stiegler extended the Petname Model based on Mark Miller's suggestion and also explained the detailed interaction among the components of the Petname Model (Stiegler, 2005). He also formalised the properties and requirements for the Petname Model and gave examples of some applications of Petname Systems. The evolutionary time-line in this section illustrates how the different topics of namespace, identity management and trust management are interrelated, and how they were combined to formulate the Petname Model.

THE PETNAME MODEL

Real World Example

Before we analyse the principle of the Petname Model, let us go through a real world example in which the Petname Model is so naturally integrated that we hardly notice its existence. It will help to link back the abstract concept of the Petname model with the real world scenarios and to grasp some of the key concepts of the Petname Model. Let us first analyse how people actually recognise each other. This process is very simple and natural to us: through several physical attributes like face, voice, physique or maybe combinations of them. These combinations can be thought of as the Pointer in the Petname terminology (see below) to uniquely identify a single person. That single person introduces herself to us by stating her name XYZ which is actually a Nickname in the Petname Model terminology (See below). From then on we may perceive that the person's identity as Mrs XYZ, which actually represents a Petname in the Petname terminology (see below). Now if another person also introduces herself as XYZ, then our mind does not simply assign that name as her Petname because it was already assigned to another person. Here things may evolve in different directions. One possible direction can be that our mind distinguishes between those two persons and changes the Petname for the first person as Mrs XYZ of London and Mrs XYZ of Paris for the second person or whatever seems practical.

Rationale

As mentioned in the previous section, the Zooko's triangle visualises the hypothesis that no name can at the same time be Global, Memorable and Unique, but can only have two of the properties. Three unique pairs can be created using these three properties: 1) Global-Memorable, 2) Memorable-Unique and 3) Global-Unique. Even if no name can have all the three properties, a naming system can be designed to achieve all the three properties. The Petname Model represents one such naming system.

Components

The Petname Model uses three different types of names that in our terminology are called: Pointer, Nickname and Petname. These three name types actually represent the three sides of the Zooko's triangle and hence are synonymous to the three pairs discussed above. Detailed explanation of each of them is given below.

Pointer. The Pointer was defined as "True Name" in Shapiro's interpretation and as "Key" in Miller's interpretation. A Pointer implies a globally unique and securely collision free name which can uniquely identify an entity. In this sense, it is actually a Name (Identifier). It inter-connects the *Global* and *Unique* corners of the Zooko's triangle. The security of the Petname Model mainly depends on these factors: 1) Difficulty to forge a Pointer (meaning it should be difficult to duplicate one pointer from another), 2) Difficulty to mimic a Petname (meaning it should be difficult to create two petnames so similar that users will have difficulty to differentiate them) later and 3) It should reasonably difficult in spoofing the trusted path and context (e.g. the browser chrome within the scope of this paper) in which the petname is displayed. A public/private key pair and a fully qualified pathname of a file in an Internet file server are good examples of Pointers. They are globally unique and difficult to forge. However, a Pointer (e.g. a public key, IP address, etc.) may not be memorable to the human.

Nickname. The Nickname inter-connects the *Global* and *Memorable* corners of the Zooko's triangle. It is an optional non-unique name created by the owner of the Pointer. The purpose of the Nickname is to aid in identifying the entity easily. The title of a web page that is displayed in the title bar of the browser is an example of a Nickname. Users may remember that web page by the title, but another website may have the same title and can create a collision on the user's mind. Thus a Nickname is not necessarily unique.

Petname. The Petname is a name created by the user to refer to a specific Pointer of an entity. Within the domain of a single user a bi-directional one-to-many mapping exists between Petnames and Pointers. A Petname connects the *Memorable* and *Unique* corners of the triangle. Petnames only have a local scope and may only be relevant for local jurisdiction. The trusted path and context mentioned in the Pointer section above highlights the importance of confining the Petname only in a local context. The same Petname can be used by different users to refer to either the same Pointer or to different Pointers. The security of a Petname System also depends on the privacy of Petnames and the difficulty to mimic a Petname. Here it is interesting to note that a Petname does not necessarily mean a text-based name. In addition to text, it can also be an image or a sound or any combination of all of the items in different ways.

The concept of *Referral* is also related to the Petname Model (Stiegler, 2005). A Referral from a third party can consist of a Pointer and a so-called *Alleged Name* which is the introductory/referred name for an entity, like the Nickname. The distinction between a Nickname and an Alleged Name is that the Nickname is created by the owner of the entity and the Pointer, whereas the Alleged Name is provided by a third party. In a trivial case, the Nickname and the Alleged Name can be identical. If your friend sends you a message with the text "Best e-auction site" with the link `www.ebay.com`, then it can be thought as Referral where the text "Best e-auction site" can be interpreted as the *Alleged* name.

A couple of naming conventions, global names and local names, found in the Simple Public Key Infrastructure (SPKI, in short) (Ellison, Frantz, Lampson, Rivest, Thomas, and Ylonen, 1999) has strong similarities with the Petname model. Therefore it is useful to analyse the similarities and dissimilarities between these names of the two models. The suggested globally unique name (or global name) in the SPKI model is essentially a Pointer in the Petname model. Both share the same properties of a name; they are global and unique, however may not be memorable. The local name of the SPKI, also known as the Simple Distributed Security Infrastructure (SDSI, in short) name, is similar to the concept of the Petname as both are locally created name residing only the domain of a user with one major dissimilarity: a SDSI name can be used globally by prefixing the local namespace with the local name, however a Petname will never reach global scale and will always be kept under local jurisdiction. The SPKI has no concepts of the Nickname or Alleged name whatsoever.

Relationship among the Components

There is a bidirectional one-to-many mapping between Petnames and Pointers within the domain of each user. A Nickname has a one-to-many relationship to the set of Pointers. A Pointer is assumed to map to a single Nickname, but can map to several Alleged Names in the global domain. The relationship between Petnames and Nicknames can be confusing sometimes when first described without a good example. In some situations, a Nickname can be used as a Petname or in other situations a Petname can be derived from the Nickname. A single Nickname can always be uniquely resolved from the Petname, but the Nickname is not necessarily unique for the Petname. For that reason, a Petname cannot be uniquely resolved from a Nickname. Figure 3 illustrates this relationship. As seen from the figure, the Petname Model is actually a naming convention built on top of the Zooko's triangle.

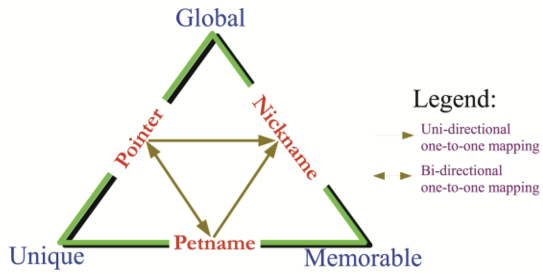

Figure 3. Petname Model

It is fascinating to note here that other than providing a trivial bi-directional mapping, the relationship between the Pointer and the Petname offers a subtle indication of the trust transition that was mentioned previously. Thus a Petname can also be thought of as a trust indicator for the Pointer. In Sect. 7 it will be explained how Petnames can act as a trust indicator for Pointers.

PROPERTIES OF PETNAME SYSTEMS

The properties of a Petname System can be divided into two broad categories: Functional properties and Security Usability properties.

Functional Properties

F1. A Petname System must consist of at least a Pointer and a Petname.
F2. Nickname is optional.
F3. Pointers must be strongly resistant against forgery so that the Pointer cannot be used to identify a false entity, meaning that there should not be any second pointer that has been created from another pointer and at the same time both pointers refer to different entities, or simply, two same pointers must refer to the same entity. However, it is always possible to have two different pointers referring to the same entity.
F4. For every user there must be a bi-directional one-to-many mapping between the Petname and the Pointer of each entity only if these pointers refer to the same entity, otherwise a bidirectional one-to-one mapping between the Petname and Pointer of each entity has to be enforced. That is, the same Petname can be used for different pointers only if all these pointers refer to the same entity. It is suitable for situations when an entity has different pointers and the user wants to use the same Petname for all these pointers. It also enforces that the same Petname cannot be used for different pointers if each of those pointers refers to different entities.

Security Usability Properties

Security usability will ensure the reliability of using the system and enables the user to draw conclusion on the actual security of the system. These properties will ensure that the Petname System is not affected by usability vulnerabilities. Usability properties can again be categorized in two types (Jøsang, , Zomai, and Suriadi, 2007):

Security Action. A security action is when users are required to produce information and security tokens, or to trigger some security relevant mechanisms. Security actions enable a user to interact securely with an entity. For example, typing and submitting a password is a security action. Properties related to the security action in the Petname System are (Stiegler, 2005):

SA1. It is the user who must assign the Petname for each Pointer.
SA2. Users must assign the Petname for the Pointer with explicit actions.
SA3. As the relationship between the user and other entities evolve, the user should be able to edit the previously applied Petname for a Pointer to a new Petname.

SA4. Suggestion on the Petname based on the Nickname can be provided as an aid for the user to select a Petname for a Pointer. If the Nickname is missing, other criteria could be chosen for the suggestion.

SA5. If a suggestion is provided and the user wants to accept it as the Petname, then she must do so with explicit actions. This is to ensure that the suggestion is not automatically assigned as the Petname and users are well informed that a suggested name is being assigned as the Petname.

SA6. Petname Systems must make sure that the user-selected, created or suggested Petname is sufficiently distinct from the Nickname so that the user does not confuse them with each other. This is needed to ensure that two same nicknames do not result in the same Petname and thus violating the F4. It might be acceptable that a Petname is equal to the Nickname in case a specific Nickname is unique within the user's local domain, but it would cause confusion and security usability vulnerabilities in case two or more Pointers correspond to the same Nickname in the user's domain. An alternative formulation of the SA6 property can therefore be that the Petname System must enforce that a Petname is different from the Nickname in case the Nickname is non-unique.

SA7. Petname Systems must make sure that the user-selected, created or suggested Petname must be sufficiently different from existing Petnames unless they refer to the same entity. In that case, the Petname even be the same as any existing one. This is needed to reduce the risk of mimicry of the Petname upon which the security of the Petname System largely depends.

SA8. If the user chooses a Petname that may resemble a Nickname or other Petnames, she should be warned explicitly. This property actually supplements SA6 and SA7.

SA9. The User should be alerted to apply a Petname for the entity that involves in highly sensitive data transmission.

Security Conclusion. A security conclusion is when users observe and assess security relevant evidence in order to derive the security state of systems. Security conclusions enable the user to conclude on the security state of the system by observing security relevant evidence and assessing this together with assumptions. For example, observing a closed padlock on a browser, and concluding that the communication is protected by TLS is a security conclusion. Properties related to the security conclusion are (Stiegler, 2005):

SC1. The Pointer and the corresponding Petname must be displayed at all times through the user interface of the Petname System. This will make the user confident about her interaction and help to draw the security conclusion easily.

SC2. The Petname for a Pointer should be displayed with enough clarity at the user interface so that it can attract the user's attention easily.

SC3. The absence of a Petname for a Pointer should be clearly and visually indicated at the user interface so that the user is surely informed about its absence.

SC4. The visual indications distinguishing actual petnames from suggestions (like Nicknames) should be unambiguous enough so that the user does not confuse them with each other.

SC5. The warning message that will be provided when there is a direct violation of any of the above properties should be clear enough so that the user can understand the problem and take the necessary security action.

PKI & PETNAME SYSTEMS

Now, let us analyse the ways a Petname System can be used for entity authentication as well as for trust validation. In line with context of this chapter, the ideal place to utilise any Petname System is inside the chrome of a browser as a browser add-on or extension much like the Petname Tool (Close, 2005), developed by Tyler Close, TrustBar (Herzberg & Gbara, 2004), developed by the TrustBar team at the Dept. of Computer Science in the Bar Ilan University, Israel and Passpet (Yee & Sitaker, 2006), developed at the University of California, Berkeley. All of them are Firefox extensions and work only with the Firefox. They allow the user to define a Petname for a website and display the Petname when she visits it later. The first step of using the Petname System is, as we call it and happens to be the most important one, Trust Bootstrapping in which the user visits the intended (secure) website with an intention to involve in an online transaction. It follows the Entity Authentication phase as described in the PKI Section in which the certificate is validated and the website is authenticated. During the bootstrapping phase, the user needs to investigate the domain name and the certificate carefully. When she is confident that this is the correct entity, she will define a Petname against a Pointer for this website satisfying the functional and security usability requirements mentioned above. Different Petname Systems utilise different mechanisms to define a Pointer. For example, the Petname Tool uses the pair CA public key fingerprint, end entity Organisation name or Common Name if absent to define the Pointer for that site. It does not work with non-https sites because it depends on certificate to retrieve the public key. Passpet extends the idea of the Petname tool also for non-https sites. It utilises the combination of root key, field name and field value to generate the Pointer. For https sites, root key is the hashed public key of the site, field name is “O” and field value is the organisation name if organisation name is available in the certificate, otherwise field name is “CN” and field value is the certificate’s common name. For the non-https sites, root key is empty, field name is “D” and field value is the last n+1 level for the n-level TLD (Top level domain). Users can assign a Petname for each site by clicking an icon in the browser. The domain name represents the Pointer in case of TrustBar.

The second step is called the Trust Validation in which the user visits the aforementioned secure site once again. If the user visits the same site (identified and authenticated by the certificate) and the calculated Pointer matches with a pointer for which a Petname was assigned beforehand, her defined Petname will be displayed in the Petname System. The absence of the Petname would indicate that her trust has not been validated properly and would provide the required visual cue that this site may be a fraudulent site and hence she needs to be more vigilant in engaging any sort of transaction with this site. Different systems may utilise different visual cues which will be explained in the next section. As mentioned previously, a Petname needs not to be a Text only; it can be an image, sound or any combination of them.
To summarise, PKI and the Petname Model should be considered as supplementary technologies for entity authentication and trust validation on the web; not as complimentary technologies. The PKI lacks significantly in providing a semantically consistent trust bootstrapping and validation mechanism which can be provided consistently by the Petname Model as long as the bootstrapping process is carefully carried out. On other hand, the Petname model may not function properly without the automatic entity authentication mechanism service provided by the PKI.

At this point it is worth exploring recently popular trust ensuring mechanisms based on recommendation systems where the reputation data is provided by the community of users of the respective system. There are different such systems namely McAfee's SiteAdvisor and SiteAdvisor Plus (McAfee, 2012), Trend Micro's TrendProtect (Trend, 2012) and WOT Services Ltd.'s Web of Trust (WoT) (WOT, 2012). To keep our analysis concise, here we analyse the WoT only.

WoT provides a browser add-on for different browsers. Using this browser add-on, WoT allows users to rate the trustworthiness, vendor reliability, privacy and/or child safety of a website as they perceive while using the websites or the services they offer. The rating scale ranges from poor to excellent. The rating input is transferred to the WoT Server where the reputation of any websites are computed algorithmically through a combination of user ratings and data from trusted sources such as hpHosts Legit Script Panda Security , PhishTank , and TRUSTe. The reputation data for a specific website is displayed on the WoT add-on using the traffic-light style color rating system (Green for provable safe sites, Red for provable dangerous sites, etc.) when a user visits that site.

Theoretically, any such recommendation system coupled with the PKI can provide the trust bootstrapping and validation similar to the Petname Model. However, any such recommendation system suffers from some significant disadvantages such as:

i) Like any other recommendation systems, the accuracy of the rating of a website largely depends on the active participation of users and on the aggregated trust users have in that website. Therefore, they system may rate a website as a potentially unsafe one even though the website may be technically safe if a significant enough portion of the community has indicated lack of trust for the site (WOT Wiki, 2012). Conversely, the system may rate a website as a potentially safe one even though the website may be technically unsafe if a significant enough portion of the community has (maliciously) given a positive feedback for the site. That is, the displayed result may be significantly biased.

ii) A new website which is not properly recognised in the system may do significant damage before it is negatively rated creating a window of opportunity for malicious websites.

iii) Such systems may alternate the ratings of different websites to gain financial incentives (e.g. via sponsored result, etc.). For example, it has been reported that McAfee's figures show a higher percentage of potentially dangerous sites among sponsored results in their systems (Rubenking, 2009).

iv) Relying on a specific recommendation system run by a third party for ensuring trust may be potentially problematic for users in case the respective company goes bankrupt and all the reputation data is gone.

Having no central dependency on a specific third party, the Petname Model is free from almost all these disadvantages. It does not depend on any third party that can influence the outcome of any trust bootstrapping and trust validation result. There is no window of opportunity for malicious websites as long as the user can bootstrap the trust. There is no need for a significant amount of participation from the users of a community to establish and validate trust in the Petname Model. In such we believe that the Petname Model coupled with the PKI can be a more suitable choice for entity authentication and trust validation on the web.

OTHER APPLICATION DOMAINS

The presence of the Petname Model is so ubiquitous that people may sometimes be unaware of its existence. Here we will highlight the possible domains in which the Petname Model is used, intentionally or unintentionally, or could be used. For each of the applications we will try to determine the suitability of applying the Petname Model (Stiegler, 2005).

Phone/E-mail Contact List

A phone/email contact list is another classic example of a Petname System. The phone number with international format (preceding the number with + or 00 and country code) may represent the Pointer and it is unforgeable and globally unique. We save the number in our contact book by placing a name for it which is nothing but a Petname for that number. Nicknames are absent here. The same analogy applies for email contact lists. Email addresses represent Pointers. A From-field in an email header may contain only the email address: xyz@yahoo.com or a given name by the sender with her email address: Mr. XYZ <xyz@yahoo.com>. Here the given name (Mr. XYZ) represents the Nickname. After receiving a mail from a new sender one can save the sender’s email address in the email contact list. At that time a Petname is created by inserting a name suitable to identify that person, or by simply keeping the Nickname. It should be noted that many email systems violate the petname model by not adequately distinguishing between a nickname that was automatically accepted by the system without any user action, and a petname that the user explicitly accepted. In such systems, it was not uncommon to start a series of email with one person and end the day with a completely different person who happens to have the same nickname and thus causing lots of confusion just because the system has tried to be helpful. This particular example stresses out the need for SA1 and SA2 properties.

IM Buddy List

In the domain of a particular Instant Messaging Service each entity has a unique Id (email Id for yahoo, Hotmail or Passport service) which represents the Pointer for that entity. But sometimes those Ids can have quite close resemblance (logicman and 1ogicman, the second one actually is a 1 not a small L) to each other and thus can be quite confusing for the user to differentiate. A better option is used in the interface of the Instant Messenger where one can put a name for each of the IDs. Such a name is actually a Petname. In the user interface all the interactions with the Id is usually done with the Petname and thus making the IM Buddy list a good example of a Petname System. Nicknames are absent here.

IP Address

Not all IP addresses have domain names. If one would like to communicate only utilizing IP address, a Petname Model can be applied locally as a substitute for domain names. IP addresses are hard to remember, and Petnames will make it easy to refer to them. IP addresses will represent the Pointer, and the corresponding Petname will be used at the user interface. All communication from the user's side will be based on Petnames.

CapDesk and Polaris

CapDesk is a desktop environment that applies the principle of least authority and utilises the Petname Model to provide security to the user for applications (Corporation, 2012). Whenever a new application is installed, CapDesk will feature a Pet Text and Pet Graphic for that application. The user may accept it or modify it. Once provided, Pet Text and Graphics will be used in the window of the application while it runs. Like CapDesk, Polaris is also based on the principle of least authority and also uses Pet Text similar to CapDesk and attaches it to the window of the application while it runs (Stiegler, Karp, Yee, and Miller, 2004).

OpenPGP

The OpenPGP key is the Pointer and it carries the Nickname given by the owner of the Pointer. Some implementations of OpenPGP allow the user to change the Nickname and implement a Petname System (Stiegler, 2005).

Process Handling

Every modern OS runs a number of processes simultaneously. *ps -e* command in Linux or the process tab in the task manager for Windows shows a long list of processes. Some of the process names are so obscure that it is impossible for the user to understand their functionalities. A Petname Model could be applied to improve the situation significantly. When a process would run for the first time it would present a short description of what it would do. Then the user could create an informative Petname for that process. This Petname would be displayed in the memory map, for example in the process tab in task manager or with *ps -e* command. In this case the Pointer does not have to be global. It is simply the unique process name or unique command used to run the process.

EVALUATION OF SECURITY USABILITY FOR PKI-BASED PETNAME SYSTEM

The usability of security is crucial for the overall security of the system, but is still a relatively poorly understood element of IT security. Therefore it is important to evaluate the Security Usability of Petname Systems as it is directly related to the security of client-side Identity Management. A set of general Security Usability principles related to Identity Management were proposed in (Jøsang, Al Zomai, and Suriadi, 2007). We will use these principles as a basis to evaluate the Security Usability of the Petname System by analysing if the Security Usability properties of the Petname System satisfy these principles. The Security Usability principles are described below:

Security Action Usability Principles:

A1. Users must understand which security actions are required of them.
A2. Users must have sufficient knowledge and the ability to take the correct security action.
A3. The mental and physical load of a security action must be tolerable.
A4. The mental and physical load of making repeated security actions for any practical number of instances must be tolerable.

Security Conclusion Usability Principles:

C1. Users must understand the security conclusion that is required for making an informed decision.
C2. The system must provide the user with sufficient information for deriving the security conclusion.
C3. The mental load of deriving the security conclusion must be tolerable.
C4. The mental load of deriving security conclusions for any practical number of instances must be tolerable.

The Security Usability properties of Petname Systems can now be analysed according to these security principles. When a Petname System satisfies SA1-SA3 and SA6-SA9 of the Security Action properties, it implicitly implies that principles A1 and A2 are also satisfied, because the former properties enable a user to select a unique and unambiguous Petname for a Pointer. This selection of a unique and unambiguous Petname for a Pointer can be thought of as the correct security action as it enables the user to securely identify an entity. Security Action properties SA4-SA8 will act as the aid for the user to select a Petname for a Pointer. We believe that selecting an unambiguous Petname will pose the most significant mental load for the user in the Petname System when repeated for several entities. Such mental load will be reduced significantly if these five properties are satisfied in a Petname System because users do not have to think about the ambiguity of the new Petname with other existing Petnames. Automated suggestion could also be a great aid in such selection. Therefore satisfying these five properties will implicitly lead to the principles A3 and A4 also being satisfied.

To analyse the Security Conclusion properties of the Petname System, we have to first define Security Conclusion in the Identity Management perspective. Security Conclusion in the Identity Management perspective is to correctly identify a specific entity. Displaying the Petname for a Pointer that points to the desired entity at the user interface will enable the user to draw conclusion that this Pointer and in turn the entity the user is interacting with is the intended one. The presence and absence of the Petname will provide the user with enough information to draw the security conclusion easily. So whenever a Petname System satisfies SC1-SC3, it will explicitly satisfy C1 and C2. Different visual techniques should be applied to help the user reduce their mental load in deriving security conclusions. Using different eye-catching colours to indicate the presence or absence of a Petname for a specific Pointer can be an example of one such visual technique. The security conclusion properties SC2-SC5 should be applied to enable a user to draw conclusion with ease and thus if followed will satisfy principles C3 and C4.

From the above analysis we can conclude that a complete implementation of all the properties of a Petname System will satisfy all the security usability principles.

Having formalized the properties of Petname Systems, and having analysed security usability issues on a general level, the security usability for two existing Petname System applications are analysed with the Cognitive Walkthrough method. The applications to be analysed are: 1) Petname Tool and 2) TrustBar. Both toolbars are designed only to work with the Firefox browser, and are aimed at simplifying client-side management of SP identities and at providing a better defence mechanism against Phishing attacks. Though the application domains for the Petname System is much broader, as described in Sect. 8, we have decided to confine our evaluation only to these two in order to focus on managing SP identities at the client side. These two particular applications exactly meet this criterion. The Passpet mentioned earlier could not be evaluated as it was not compatible with the Firefox that was used for evaluation or with the current version of the Firefox.

The Cognitive Walkthrough method is a usability evaluation method in which an evaluator or a group of evaluators participate to identify the usability issues of an application by visually inspecting the user interface. It focuses on evaluating the understandability and the ease of use for a user at the user interface to accomplish a task using that application. Among several usability evaluation methods the Cognitive Walkthrough was chosen as our preferred method because of its main focus on the understandability of the user at the user interface (Whitten & Tygar, 1999). Because Petname Systems affect the user interface, Cognitive Walkthrough is a suitable method for evaluating their usability. While performing the Cognitive Walkthrough for each application, it will be noted if the application satisfies the usability properties discussed in Sect. 5. The degree of compliance with the specified security usability properties will give an indication of the level of security usability of each application. For the evaluation, Firefox with the Nightly Tester Tool, a Firefox add-on, was used. It is important to note here that the evaluations were performed by the authors of this chapter.

The Petname Tool

Setup

The Petname Tool is available as a Firefox add-on in (Close, 2012). The current version of the Petname Tool is compatible with the latest Firefox version, and can be easily installed by just clicking the "Add to Firefox" button in the respective Firefox Add-on website. Once installed the toolbar will look like Fig.4.

Figure 4. The Petname Tool in Firefox

Functionality

The first thing to note about the Petname Tool is its simplicity. It consists of only a text field in the navigation toolbar of the browser. Its main purpose is to allow a user to assign a Petname for a website that she wants to correctly recognize and to display that Petname in the text field when she visits the site later. The Petname will be absent if the visited site is not the intended one. A user can judge if a webpage comes from a previously identified website by observing the presence or absence of the Petname. The Petname Tool utilizes different font properties and graphical user interface elements to achieve its goal: 1) The text in the text field, 2) The typeface of the text, 3) Color of the text field, 4) Tooltip and 5) Dialog box. Different texts with different typefaces are displayed in the text field in different situations, color of the text field change, different tooltips are provided accordingly when mouse pointer is placed over the text field and warnings are displayed using dialog boxes.

Some examples can illustrate how the Petname Tool operates. It is worth noticing here that the Petname Tool does not work for non-https sites, as it uses the pair CA public key fingerprint, end entity Organisation name or Common Name if absent as the Pointe for that site. While visiting a non-https site, e.g. `www.wikipedia.org`, the text in the text field will be *unauthenticated* with italic typeface and it will be disabled with grey colour so that nobody can assign a Petname for the site (Fig.5). The corresponding tooltip is: *Don't give this page sensitive information; it was not received securely* (Fig.6). During the visit to a https site for the first time, e.g. `www.paypal.com`, the text in the text field becomes *unknown site* with italic typeface and the text field colour changes to white (Fig.7) with the corresponding tooltip-*Assign a Petname to this site before exchanging sensitive information* (Fig.8). At this point, user can assign a Petname by just writing it in the text field and hitting the Enter key. The colour of the text box changes from white to light green and type face becomes normal (Fig.9). When the user visits that site later, the Petname with normal typeface is displayed in the green text field. Different dialog boxes are prompted to warn users whenever something goes wrong.

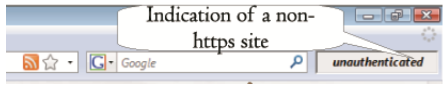

Figure 5. Disabled text field for a non-https site

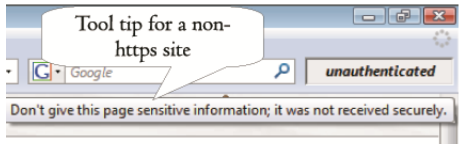

Figure 6. Tooltip for a non-https site

Figure 7. Indication of an https site

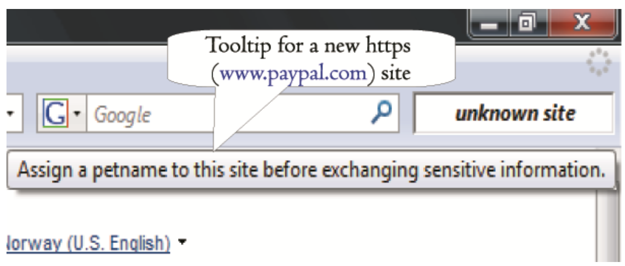

Figure 8. Tooltip for an https site

Figure 9. Assigned Petname for a new https site

Evaluation

As mentioned earlier, the Petname Tool is very simple, however, one may almost feel that it is too simple. It does not come with any text label; only a text field to enter Petnames. Absence of a text label can confuse unfamiliar users because they might not understand its purpose. The Petname Tool does not work for non-https sites; therefore it will not be possible for a user to assign Petnames to non-https sites. Many sites with server certificates do not use https in the initial log-in stages, though the log-in name and the related password may be encrypted before transmission. An example is the famous social networking website `www.facebook.com`. A potential vulnerability is caused by Facebook because email addresses represent user names. People often use the same passwords for different accounts, so a password used on Facebook will often allow access to the user's web email account. The lack of support, therefore, for non-https sites in the Petname Tool is a major drawback. Another thing is worth to note that the Petname Tool uses the pair CA public key fingerprint, end entity Organisation name or Common Name if absent as a Pointer. Therefore if the site receives a new certificate and thus a new public key, the Petname Tool will fail to map between the already assigned Petname and the Pointer. A possible solution could be to let URL or domain name be the Pointer that will also remove the restriction of applying Petnames for https sites only.

In the following, the Petname Tool will be analysed for compliance with the Petname System properties. The Petname Tool, obviously, deploys Petnames. The pair, CA public key fingerprint, end entity Organisation name or Common Name if absent, is used to define the Pointer and is strongly resistant against forgery. Therefore we conclude that the Petname Tool satisfies F1 and F3. But a serious restriction of the Petname Tool is that it allows users to assign exactly the same Petname for different entities as demonstrated in the next paragraph, thus violates F4. It does not deploy Nicknames and therefore does not satisfy the optional property F2.

The Petname Tool enables users to explicitly assign a Petname for each entity, e.g. to select the text field, write down a Petname and hit the Enter key. This satisfies SA1 and SA2. Users can change any Petname any time, thereby satisfying SA3. No suggestion is provided for aiding the user to select a Petname, thereby, is not compliant with SA4 and SA5. Also Nicknames are not used in the Petname Tool, resulting in non-compliance with SA6. Whenever a user selects a Petname that closely resembles existing Petnames, the user is alerted with an informative dialog box (Fig.10). The dialog box displays the existing Petnames to which the current Petname has close resemblance. The user can ignore the alert by clicking the *Assign petname* button or she can cancel this current Petname by clicking the "*Don't assign petname"* button. If a user assigns a Petname that is similar to an existing Petname, the Petname Tool displays the

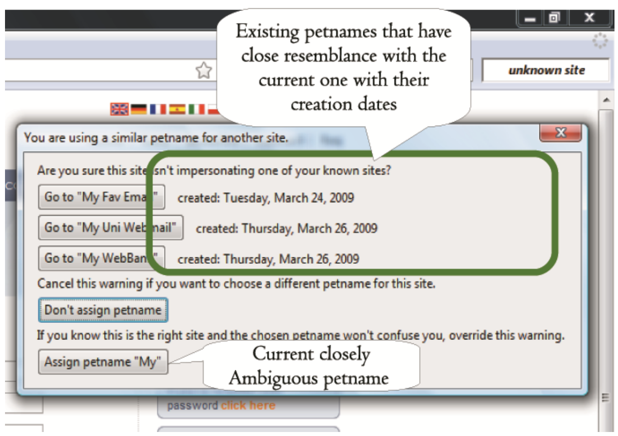

Figure 10: Dialog box warning about the close ambiguity among different Petnames

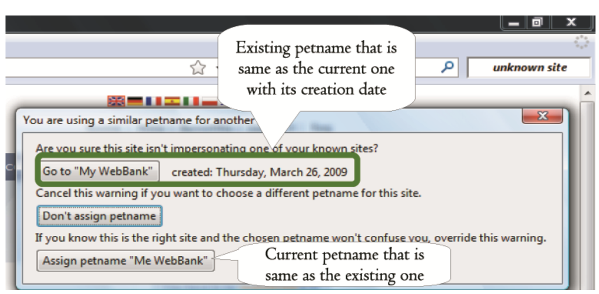

Figure 11: Dialog box warn about the similarity between two Petnames

dialog box (Fig.11). The dialog box contains the name of the existing similar Petname with its creation date. The user has the option to discard the current Petname by clicking the *Don't assign petname* button. If the user clicks the *Assign petname* button, the Petname will be assigned for the current entity. In this case, the same Petname will be displayed for both websites when she visits them later. Therefore, the Petname Tool is compliant with SA8 (showing the dialog box with the warning), but directly violates SA7 as the same Petname is possible for two different entities. The Petname Tool allows a user to assign a Petname at her will whenever she feels and does not show any alert when there is highly sensitive data transmission and therefore indicates the absence of SA9.

The Petname, if already supplied by the user, is displayed on the Petname Tool toolbar, thereby satisfying SC1. Different typefaces, tooltips and colors have been used in the Petname Tool to catch the user attention to indicate the presence or absence of a Petname. White and light green as used by the Petname Tool is less visible than Red, Yellow or Green, as suggested in (Drelie Gelasca, Tomasic, and Ebrahimi, 2005). In addition, blinking text or different text colors could be used to draw more user attention. Nevertheless, we can conclude that the Petname Tool is compliant with SC2 and SC3. As there is no suggested Petname or Nickname in the Petname Tool, it does not satisfy SC4. The Petname Tool provides warning through dialog boxes when there are conflicts with other Petnames or if there is an ambiguity between Petnames and thus satisfies SC5. However, it does not provide a warning message when there is a violation for other properties.

Apart from security usability issues, there are some other weaknesses in the Petname Tool. For example, there is no help button that could explain what the user has to do to utilise it properly. It does not provide the standard *About* menu item that could explain the purpose of the Petname Tool.

TrustBar

Setup

The TrustBar Tool is available as a Firefox add-on in (Herzberg, 2006). The current version of TrustBar is not compatible with the latest Firefox version. Therefore the Nightly Tester Tool, another Firefox add-on, was used to resolve the compatibility issues. Once installed the toolbar looks like Fig.12.

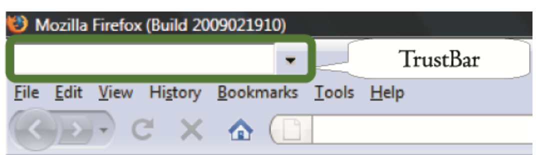

Figure 12: TrustBar installed in Firefox

Functionality

TrustBar consists of a text field, a menu and a Security Status field. The text field allows users to enter a Petname for an entity, the menu provides the user with options, and the Security Status field provides visual indication of various security statuses based on the certificate. Unlike the Petname Tool, it also allows user to assign a logo as a Petname for an entity. When a logo is used, an image replaces the text field and such logos can be called Petlogos. A user can assign a Petname text or Petlogo for a website that she wants to correctly recognise and to display that Petname or Petlogo when she visits the site later. The Petname or Petlogo will be absent if the visited site is not the intended one. A user can judge if a webpage comes from a previously identified website by observing the presence or absence of the Petname or Petlogo. TrustBar utilizes different graphical user interface elements to achieve its goal: 1) The Petname field for text or logo, 2) Colour of the Petname field, 3) Drop down menu, 4) Tooltip and 5) Security Status field. The Petname field changes as a user visits different sites. At the same time the colour of the Petname field changes and different tooltips over the Security Status field are provided. The Security Status field provides a visual indication of the status of the server certificate, and changes according to different circumstances. Options in the menu allows users to set the Petname or Petlogo, to edit the

Petname or Petlogo, remove the defined Petname(s), report fraudulent websites and display help regarding TrustBar (Fig.13). The menu also contains an *About* menu item that, if clicked, displays some relevant information regarding TrustBar, e.g. what is TrustBar, why is it used for, etc.

Some examples can illustrate the TrustBar functionality. Unlike the Petname Tool, TrustBar works both with https and non-https sites. When users visit a non-https site, e.g. www.wikipedia.org, the Petname field contains the domain name for that site (Fig.14). A user can assign a Petname by writing directly in the text box and hitting the enter key. The colour of the text field will turn from white to light green. The Security Status filed displays a No lock icon indicating that the site does not have a server certificate and that TLS is not used, and also provides the tooltip "This site is not protected. Click here for more information" (Fig.15). Clicking the icon will redirect the user to the TrustBar website that explains the necessary concept on TrustBar. A user can edit the Petname later just by writing the new one and hitting the enter key. The drop-down menu also provides methods to assign, edit or delete Petnames. Assigning and editing a Petlogo happens in a similar way, except that the user has to select an image from her computer. When the user visits an https site, e.g. mail.yahoo.com, the text field contains the organisation name from the certificate. The colour of the text field turns to pale yellow. The Security Status field is

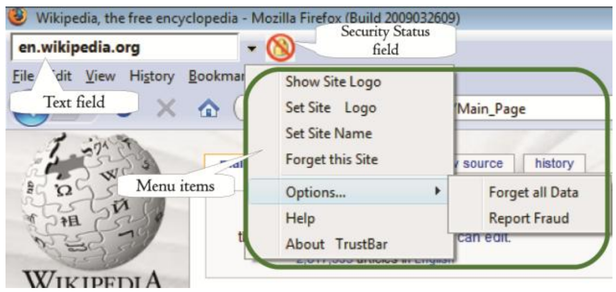

Figure 13. Components of TrustBar

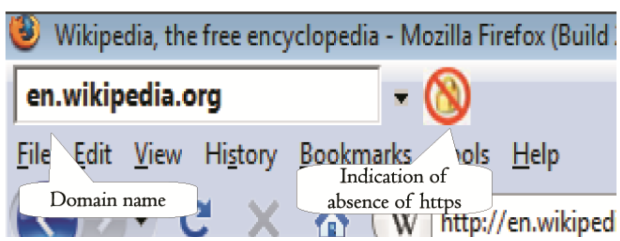

Figure 14. Indication of a non-https site in TrustBar

modified with a lock icon and the text "Identified By:". The name of the CA and another drop-down menu are displayed adjacent to the Security Status field. This second menu allows the user to set, edit or delete a logo for CA, to ignore the CA, and some other options (Fig.16). The user can assign a Petname or Petlogo to override the organisation name like before. Once a Petname is assigned, the Petname field turns to light green (Fig.17).

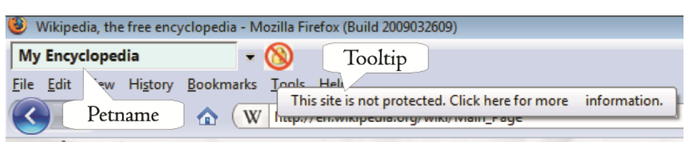

Figure 15. Assigning a Petname for a non-https site in TrustBar

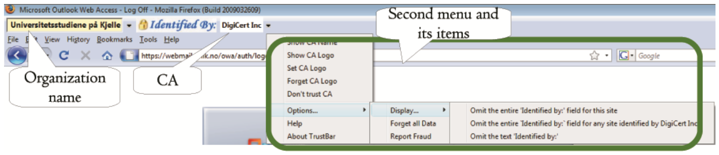

Figure 16. TrustBar interaction with an https site in TrustBar

Figure 17: Assigning a Petname for an https site in TrustBar

Evaluation

TrustBar overcomes some of the shortcomings of the Petname Tool. For example, it works for non-https sites, provides an excellent "*Help*" feature and also comes with the standard *About* menu item that provides a short description of what it does.

The following simple analysis of TrustBar gives an indication of how it satisfies the properties of the Petname Model. TrustBar utilizes Petnames, and thereby complies with F1. The domain name or URL represents the Pointer. Therefore we conclude that TrustBar satisfies F1 and F3. TrustBar also displays a Nickname in the form of the organisation name, if a certificate is available or in the form of the domain name for non-https sites and thus satisfies F2. However, a serious restriction of TrustBar is that it allows users to assign exactly the same Petname for different entities as demonstrated in the next paragraph, thus violates F4.

TrustBar enables a user to assign a Petname for each entity so she has to act explicitly, e.g. select the text field, write down a Petname and hit the Enter key, to enable the Petname and this satisfies SA1 and SA2. Users can change any Petname any time and thus TrustBar meets the requirement of SA3. A suggestion is provided in the form of a Nickname for aiding the user to select a Petname if a server certificate is available and this satisfies SA4 partially, and the user has to act explicitly, e.g. by hitting the Enter key so the text field turns to light green (an indication for accepting the Petname) to accept the Nickname as the Petname. This satisfies SA5 too. However, it is important to note here that if the Nickname is accepted as the Petname without any modification then it represents a temporary Petname, because when the user visits it again, the Petname turns into the Nickname, also indicated by the pale yellow colour of the text field. This means that TrustBar tries to ensure S6. But this approach is rather contradictory and that a better approach could be taken that would not allow users to accept the Nickname as the Petname without any modification. As mentioned earlier, a serious restriction of TrustBar is that it allows users to assign ambiguous Petnames or even equal Petnames for different entities. It does not provide any sort of warning to users about the ambiguity or similarity of the Petnames and thus directly violates SA7 and SA8. TrustBar allows a user to assign a Petname at her will whenever she feels and does not show any alert when there is highly sensitive data transmission and therefore violates SA9.

The Pointer and the related Petname in the TrustBar, if already supplied by the user, are displayed all the time in the browser toolbar, thereby satisfying SC1. Different icons, tooltips and colours have been used in the TrustBar to catch the user attention to indicate the presence or absence of Petnames. It would have been better to use more flashy colours like Red, Yellow or Green instead of pale yellow and light green. Blinking text or different text colours could be used to draw more user attention to potential security problems in websites. Nevertheless, we can conclude that TrustBar satisfies SC2 and SC3. White, pale yellow or light green colour has been used to differentiate among non-https Nicknames, https Nicknames and Petnames respectively, thereby satisfying SC4. TrustBar does not provide any sort of warning to the user and this indicates the complete absence of SC5.

Summary

Table 2 summarises the distinction between the Petname Tool and TrustBar in terms of the properties of Petname Systems.

Tool Name	F				SA									SC				
	1	2	3	4	1	2	3	4	5	6	7	8	9	1	2	3	4	5
Petname Tool	Y	N	Y	N	Y	Y	Y	N	N	N	N	Y	N	Y	Y	Y	N	Y
TrustBar	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	N

Table 2: Comparison between the Petname Tool and TrustBar

It can be noted that TrustBar satisfies more properties of the Petname model than the Petname Tool, though TrustBar has one major shortcoming: absence of any type of warning message. Both tools suffer from the absence of the crucial property F4. As neither of them satisfies all the main properties of Petname Systems, we can conclude that none of them fully satisfies the security usability principles.

FUTURE WORK

The next natural step is to conduct a large-scale usability study of the mentioned Petname Systems performed by several users. It would be very interesting to compare both results and also to determine the understandability and ease of use for Petname Systems by general users.

No tool (assuming the inapplicability of the Passpet tool with any current browser) is currently available for Petname based identity management that satisfies all the properties of Petname Model and the corresponding security usability principles, as indicated by our analysis in the previous section. Developing a Petname Model based tool that satisfies the security usability principles should be a priority in future research and development. The extensions described earlier work only with the Firefox. The same functionalities should be made available for other browsers as well. However, it may be tricky to develop a Petname System that could be integrated into every browser since different browsers use different mechanisms and frameworks for their extensions. A better approach could be to develop a central Petname System and provide services to each browser when required. Unfortunately, it will require a considerable amount of research and development effort to build such a central system.

As the Petname Model is based on the Zooko's triangle, any shortcut in the triangle may collapse the relationships among the components of the Petname Model or may create a new dimension of relationship. Bob Wyman in his web blog proposed to update the Zooko's triangle into a pyramid by inserting a new attribute called "Persistent" and connecting it to the other corners. The new attribute was proposed to signify the longevity of each name (Wyman, 2006). This proposal to change the shape of the Zooko's triangle can be another potential topic for research which could give Petname Systems additional security properties.

Smart phones are becoming increasingly popular and the number of people that access the Internet from their smart phones is growing every day. Investigations into how the Petname Model can be implemented and adapted for the tiny screen of a mobile phone can be a challenging task and another scope for future research. Typing in mobile phones is still very challenging. Sound or image based Petname Systems could be an ideal choice in this regard.

CONCLUSION

The Petname Model is naturally embedded in human perception to identify different entities. Implementing it in computer networks and system is a natural extension of human cognitive capabilities and represents a great aid for humans in digital environments. This fact has been demonstrated through several applications, experiments and proposals. It could actually be a necessary component of digital certificates and has been recommended in the W3C Security Context: User Interface Guidelines (Roessler & Saldhana, 2010). A large scale adaptation of the Petname Model is therefore timely.

In this chapter, we have focused on providing the link between PKI and Petname Systems. We have provided a brief overview of Petname Systems starting from the history and evolution of the Petname Model. We have formally defined the properties of Petname Systems and explained how this set of properties can satisfy essential security usability principles. It is our belief that the integration of the Petname Model into applications will improve the user experience and improve overall security by removing security vulnerabilities related to poor usability. We have also explained the necessity, suitability and applicability of the Petname Model in combination with traditional PKIs. The chapter has also analysed two available Petname-based applications for SP identity management on the client side, and have shown that they represent a significant improvement in usability, but unfortunately do not fully satisfy every desirable security usability principle.

REFERENCES

Close, T. (2003). Naming vs. pointing. Retrieved April 17, 2012, from http://www.waterken.com/dev/YURL/Analogy/.

Close, T. (2003a). Waterken YURL:Trust management for Humans. Retrieved April 17 2012, from http://www.waterken.com/dev/YURL/Name/.

Close, T. (2005). Petname tool: Enabling web site recognition using the existing ssl infrastructure. Retrieved April 17, 2012, from http://www.w3.org/2005/Security/usability-ws/papers/02-hp-petname/.

Close, T. (2012). Petname tool 1.7. Retrieved April 17 2012, from https://addons.mozilla.org/en-US/firefox/addon/957.

Corporation, Combex. (2012). Capdesk. Retrieved April 17 2012, from http://www.skyhunter.com/marcs/CapDeskSpec.html.

Deutsch, Harry (2008). Relative Identity. The Stanford Encyclopedia of Philosophy (Winter 2008 Edition), Edward N. Zalta (ed.), from http://plato.stanford.edu/archives/win2008/entries/identity-relative/.

Dhamija, R. & Tygar, J.D. (2006). Why phishing works. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 581–590. ACM Press.

Drelie Gelasca, E., Tomasic, D. & Ebrahimi, T. (2005). Which Colors Best Catch Your Eyes: a Subjective Study of Color Saliency. First International Workshop on Video Processing, Retrieved April 17, 2012, from http://infoscience.epfl.ch/getfile.py?mode=best&recid=87215

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B. & Ylonen, T. (1999). SPKI certificate theory. RFC 2693.

Ellison, C. & Schneier, B. (2000). Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure. Computer Security Journal, Volume XVI, Number 1.

Ferdous, M. S., Jøsang, A., Singh, K., Borgaonkar R. (2009). Security Usability of Petname Systems. Proceedings of the 14th Nordic Workshop on Secure IT systems (NordSec 2009) Oslo, Norway, October 2009

Gallois, André (2011). Occasional identity: Thereby hangs the tale. Analytic Philosophy 52 (3):188-202.

Gallois, Andre (2012). Identity Over Time, The Stanford Encyclopedia of Philosophy (Summer 2012 Edition), Edward N. Zalta (ed.), from http://plato.stanford.edu/archives/sum2012/entries/identity-time/.

Geach, P.T. (1967). Identity. Review of Metaphysics, 21: 3-12. Reprinted in Geach 1972, pp. 238-247.

Herzberg, A. (2006). Trustbar firefox addon. Retrieved April 17, 2012, from http://u.cs.biu.ac.il/~herzbea/TrustBar/.

Herzberg, A. and Gbara, A. (2004). Protecting (even Naïve) Web Users from Spoofing and Phishing Attacks. Technical Report 2004/155, Cryptology ePrint Archive.

Housley, R., Ford, W., Polk, T., & Solo, D. (1999, January). Internet X.509 Public Key Infrastructure – Certificate and CRL Profile. RFC 2459 (RFC 3280–2002).

Internet Archive Wayback Machine (2012). Snapshot on zooko's writing. Retrieved on April 17 2012, from http://web.archive.org/web/*/http://zooko.com/distnames.html.

Jøsang, A. & Pope, S. (2005). User centric identity management. In Asia Pacific Information Technology Security Conference, AusCERT2005, Austrailia, pp. 77–89.

Jøsang, A., Al Zomai, M. & Suriadi, S. (2007). Usability and privacy in identity management architectures. In L. Brankovic, C. Steketee (Ed.), Fifth Australasian Information Security Workshop (Privacy Enhancing Technologies) (AISW 2007), *CRPIT*, vol. 68, pp. 143–152. ACS, Ballarat, Australia.

Jøsang, A., Al Fayyadh, B., Grandison, T., AlZomai, M. & McNamara, J. (2007). Security usability principles for vulnerability analysis and risk assessment. In Computer Security Applications Conference, Annual, pp. 269-278, Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007).

Kesterson II, H. L. (2007). PKI & Identity: Technical and legal aspects. Retrieved April 17, 2012, from http://www.itu.int/dms_pub/itu-t/oth/15/04/T15040000110001PDFE.pdf.

Linn, J. & Branchaud, M. (2004). An Examination of Asserted PKI Issues and Proposed Alternatives. In Proceedings of the 3rd Annual PKI R&D Workshop, NIST, Gaithersburg MD, USA.

McAfee SiteAdvisor and SiteAdvisor Plus (2012). Retrieved September 09, 2012, from http://home.mcafee.com/store/siteadvisor-live?ctst=1

Menezes, A., Oorschot, P. van, & Vanstone, S. (1997). Handbook of Applied Cryptography. CRC Press.

Miller, M. (2000). Lambda for humans: The PetName Markup Language. Retrieved April 17, 2012, from http://www.erights.org/elib/capability/pnml.html.

Roessler, T. & Saldhana, A. (2010, August). W3C Security Context: User Interface Guidelines. Retrieved April 17 2012, from http://www.w3.org/TR/wsc-ui/

Rubenking, N. J. (2009). Web of Trust Review and Rating, August 13, 2009. Retrieved September 09, 2012, from http://www.pcmag.com/article2/0,2817,2351536,00.asp

Sider, Theodore (2000). Recent Work on Identity Over Time. Philosophical Books 41 (2):81–89.

Shapiro, J.S. (2000). Pet names, true names, and nicknames. Retrieved April 17 2000, from http://www.eros-os.org/~majordomo/dcms-dev/0036.html.

Shirky, C. (2002). Domain names: Memorable, global, non-political? Retrieved April 17, 2012, from http://shirky.com/writings/domain_names.html.

Stiegler, M. (2005). Petname systems. Retrieved April 17 2012, from http://www.financialcryptography.com/mt/archives/000499.html.

Stiegler, M., Karp, A.H., Yee, K.P. & Miller, M. (2004). Polaris: Virus safe computing for windows xp. Retrieved April 17 2012, from http://www.hpl.hp.com/techreports/2004/HPL-2004-221.pdf.

Thomas, S. (2000). SSL and TLS Essentials, Securing the Web, Wiley.

Thanh, D.V. & Jørstad, I.J. (2007). The ambiguity of identity. Telektronikk issue on Identity Management 103(No.3/4-2007, ISSN:0085-7130), 3–10.

Trend Micro's TrendProtect (2012), Retrieved September 09, 2012, from http://www.trendsecure.com/portal/en-US/tools/security_tools/trendprotect

Whitten, A. & Tygar, J.D. (1999). Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In 8th USENIX Security Symposium.

Wikipedia Entity (2012). Wikipedia entry on entity. Retrieved April 17 2012, from http://en.wikipedia.org/wiki/Entity.

Wikipedia Identity (2012). Wikipedia entry on identity management. Retrieved April 17 2012, from http://en.wikipedia.org/wiki/Identity_management.

Wilcox-O'Hearn, Z. (2001). Names: Decentralized, secure, human-meaningful: Choose two. Retrieved April 17 2012, from http://www.zooko.com/distnames.html.

WOT Services Ltd.'s Web of Trust (2012), Retrieved September 09, 2012, from http://www.mywot.com/

WOT Wiki (2012). Retrieved September 09, 2012, from http://www.mywot.com/wiki/WOT

Wyman, B. (2006). The persistence of identity. Retrieved April 17, 2012, from http://www.wyman.us/main/2006/12/the_persistence.html.

Yee, K.P. & Sitaker, K. (2006). Passpet: convenient password management and phishing protection. In SOUPS, pp. 32–43.

FURTHER READING

Alpár, G., Hoepman, J. & Siljee, J. (2011). The Identity Crisis. Security, Privacy and Usability Issues in Identity Management. CoRR, abs/1101.0427, 2011.

Camp, J.L. (2004). Digital identity. Technology and Society Magazine, IEEE, 23(3):34–41, fall 2004.
Chadwick, D. W. (2009). Federated Identity Management. In A. Aldini, G. Barthe, and R. Gorrieri, editors, FOSAD 2008/2009, number 5705 in LNCS, pages 96–120. Springer-Verlag, Berlin, January 2009.

Ferdous, M. and Poet, R. (2012). A comparative analysis of Identity Management Systems, In the Proceedings of the International Conference on High Performance Computing and Simulation (HPCS), 2012, pages 454-461, 2012.

Future of Identity in the Information Society WP3. Study on Mobile Identity Management, May 2005. http://www.fidis.net/fileadmin/fidis/deliverables/fidis-wp3-del3.3.study_on_mobile_identity_ management.pdf.

ITU-T. Baseline capabilities for enhanced global identity management and inter-operability, September 2009. http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.1250.

Jøsang, A., Fabre, J., Hay, B., Dalziel, J. & Pope, S. (2005). Trust Requirements in Identity Management. Proceedings of the Australasian Information Security Workshop (AISW'05), Newcastle, Australia, January-February 2005.

Jøsang, A., Keser, C. & Dimitrakos, T. (2005). Can We Manage Trust?. Proceedings of the Third International Conference on Trust Management (iTrust'05), 2005.

Kölsch, T., Zibuschka, J. & Rannenberg,K. (2011). Digital privacy. Chapter: Privacy and Identity Management Requirements: An Application Prototype Perspective, pages 735–749. Springer-Verlag, Berlin, Heidelberg, 2011.

Modinis – Common Terminological Framework for Interoperable Electronic Identity Management. Accessed on 28th June, 2011. https://www.cosic.esat.kuleuven.be/modinis-idm/twiki/bin/view.cgi/Main/GlossaryDoc.

NIST. Electronic authentication guideline: Information security, April 2006. http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf.

OASIS Standard. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. 15 March, 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

OAuth 2.0. http://oauth.net/2.

OpenID Authentication 2.0 - Final. 5 December, 2007. http://openid.net/specs/openid-authentication-2\0.html.

Pötzsch, S., Borcea-Pfitzmann, K., Hansen, M., Liesebach, K., Pfitzmann, A. & Steinbrecher, S. (2011). Requirements for Identity Management from the Perspective of Multilateral Interactions. In Jan Camenisch, Ronald Leenes, and Dieter Sommer, editors, Digital Privacy, volume 6545 of Lecture Notes in Computer Science, pages 609–626. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-19050-6 22.

Pötzsch, S., Meints, M., Priem, B., Leenes, R. & Husseiki, R. (2009). D3.12: Federated Identity Management – what's in it for the citizen/customer? 10 June 2009. http://www.fidis.net/fileadmin/fidis/deliverables/new_deliverables/fidis-wp3-del3.12.Federated_Identity_Management.pdf.

Priem, B., Leenes, R., Kosta, E. & Kuczerawy, A. (2011). The Identity Landscape. In Jan Camenisch, Ronald Leenes, and Dieter Sommer, editors, Digital Privacy, volume 6545 of Lecture Notes in Computer Science, pages 33–51. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-19050-6 3.

Shibboleth. http://shibboleth.internet2.edu/.

Suriadi, S., Ashley, P. & Jøsang, A. (2007). Future standardization areas in identity management systems. In Proceedings 2nd PRIME Standardization Workshop, Zurich, Switzerland, 2007.

Tatli, E. I. & Lucks, S. (2009). Mobile Identity Management Revisited. Electron. Notes Theor. Comput. Sci., 244:125–137, August 2009.

KEY TERMS AND DEFINITIONS

Keywords

Petname System, Identity Management, PKI, Authentication, Security, Trust Validation, Security Usability and Cognitive Walkthrough.

Definitions of Keywords

Petname System: The Petname System is an implementation of the Petname Model which is a naming system that is designed to achieve all the three properties (Global, Memorable and Unique) of the Zooko's triangle.

Identity Management: Identity management consists of technologies, policies and practices for recognising and authenticating entities in online environments.

PKI: Public Key Infrastructure is a framework based on the public key cryptosystem (also known as the asymmetric cryptography) and consists of a set of policies that governs how the cryptosystem should operate and defines the procedure for generating and publishing digital certificates.

Authentication: Authentication is the process of proving an association between a name (identifier or an attribute) and an entity supplying the name (identifier). To prove the association, the entity usually has to supply a credential that accompanies the name (identifier). Authentication is the process of verifying the association using the credential.

Security: Security is the inherent property of each information system that safeguards the system from unauthorised access, use, disclosure, modification, etc. Security of a system is governed by three essential properties: Confidentiality, Integrity and Availability, combinedly known as the CIA property. It is extremely essential to maintain the security of a system which is involved in online transactions especially during financial transactions.

Trust Validation: Trust validation is the mechanism for evaluating the trust that is placed on an entity during a particular transaction or operation.

Security Usability: The usability of security is crucial for the overall security of the system since it ensures the reliability of using the system and enables the user to draw conclusion on the actual security of the system.

Cognitive Walkthrough: The Cognitive Walkthrough method is a usability evaluation method in which an evaluator or a group of evaluators participate to identify the usability issues of an application by visually inspecting the user interface. It focuses on evaluating the understandability and the ease of use for a user at the user interface to accomplish a task using that application.