Service Provider Authentication Assurance [♯]

Audun Jøsang[*], Kent A. Varmedal[*], Christophe Rosenberger[†] and Rajendra Kumar[‡]
[*]University of Oslo, Norway [§]
Email: josang@mn.uio.no and kentav@ifi.uio.no
[†]ENSICAEN, France [§]
Email: christophe.rosenberger@ensicaen.fr
[‡]Ministry of Communications and Information Technology, Government of India
Email: rajendra@negp.gov.in

***Abstract*—The concept of authentication assurance traditionally refers to the robustness of methods and mechanisms for user authentication, including the robustness of initial registration and provisioning of user credentials, as well as the robustness of mechanisms that enforce user authentication during operation. However, the user is not the only party that needs to be authenticated to ensure security of online transactions. In fact, online service provision always involves two parties, typically the user on the client side and the service provider on the server side, so that mutual authentication between the two sides is required. In contrast to the unilateral focus on user authentication by industry and academia, it is in fact equally important for the user to correctly authenticate the service provider. Unfortunately, little attention is paid to the problem of correctly authentication the service provider. This paper proposes a framework for server and service provider authentication assurance, similarly to frameworks for user authentication assurance that have already been specified, or are currently under development by many national governments.**

I. Introduction

Entity authentication is considered to be a fundamental security primitive for Internet mediated interaction. Wrong authentication, where an attacker is able to take on the identity of another entity, is a serious security threat that can have significant negative consequences. Authentication assurance expresses the certainty and reliability with which correct authentication is enforced in a system or domain.

There are different forms of entity authentication that serve different purposes. This paper focuses on requirements for server and SP (Service Provider) authentication, and proposes a framework for server authentication assurance. Identity management has traditionally only focused on user authentication, for which there are well established principles for obtaining authentication assurance. Server authentication is often overlooked in the literature, so we believe it is timely to focus on requirements for server authentication assurance.

Standards for network security such as X.800 [7] typically distinguish between the following two types of authentication:

- **Data Origin Authentication:**
  The corroboration that the source of data received is as claimed.
- **Peer Entity Authentication:**
  The corroboration that a peer entity in an association is the one claimed.

The X.800 standard (called ”Security Architecture for Open Systems Interconnection”) assumes that authentication takes place between atomic entities, either as originators of data, or as peer entities in a session. In this paper we are primarily concerned with peer entity authentication, not with data origin authentication. A peer entity can be more complex that what is often assumed, as explained below.

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

By taking into account the distinction between system entity (client or server) and legal/cognitive entity (person or organisation) there are in fact two entities on each side of a communication session, as illustrated in Fig.1. This analysis shows that the implicit assumption of atomic entities made by the X.800 standard is a simplification and abstraction which thereby hides important aspects of entity authentication.

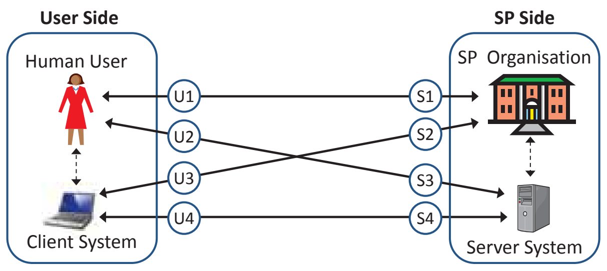


Fig. 1. General entity authentication types

The distinction between the human user and the client system on user side, as well as between the SP organisation and the server system on the SP side illustrated in Fig.1 leads to 8 different forms of peer entity authentication between the two sides, where each side can be authenticated in 4 different ways as explained below.

The four types of user side entity authentication are:
U1) Human user authentication by the SP organisation
U2) Human user authentication by the server system (commonly called *user authentication*)
U3) Client system authentication by SP organisation
U4) Client system authentication by the server system

The four types of SP side authentication are:
S1) SP organisation by the human user
S2) SP organisation by the client system
S3) Server system authentication by the human user (which can be called *Cognitive Server Authentication*)
S4) Server system authentication by the client system

Some of the entity authentication types in Fig.1 are relatively impractical, such as U3 and S2, but they illustrate the generality of entity authentication when assuming non-atomic user and SP sides. U1 is practiced e.g. for authenticating customers over the phone, e.g. by asking questions about address, date of birth, customer number as well as credentials such as PIN codes. The X.800 standard focuses on entity authentication types U4 and S4. However, in Internet and Web services applications the entity authentication types U2 and S3 in Fig.1 are the most important. The importance of entity authentication types U2 and S3 in particular emerges from the requirement of end-to-end security in web service applications. In the typical case where a human user accesses an online service, semantic end-to-end communication takes place between the human user and the server system. It is therefore pragmatic to require mutual authentication between those two entities, as illustrated in Fig.2. The syntactic communication between the server and client systems typically provides communication confidentiality by encryption, but can not provide authentication in a meaningful way.

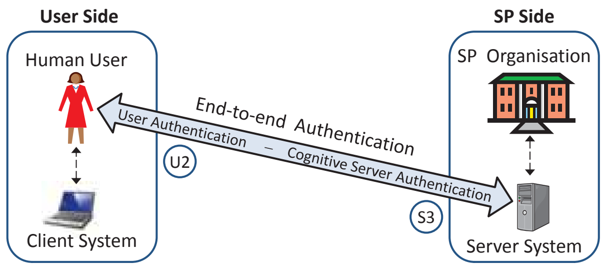

User Side
Human User
Client System
U2
End-to-end Authentication
User Authentication – Cognitive Server Authentication
S3
SP Side
SP Organisation
Server System

Fig. 2. Pragmatic end-to-end authentication

Authentication type U2 in Figs.1&2, commonly known as *user authentication*, is an essential component of Internet security and identity management. User authentication is extensively studied in the literature, with a multitude of solutions in practical use, such as passwords, OTP (one-time password) devices and biometrics. Note that user authentication also can take place locally between the client system and the human user within the user domain, as well as between the server system and a human representative of the SP organisation within the SP domain.

Entity authentication type S3 in Figs.1&2 has largely been ignored by researchers and the industry. Authentication type S3, which we call *cognitive server authentication*, focuses how a human user can verify that the correct and intended system is at the other end of an Internet connection. The lack of robust practical solutions for cognitive server authentication is a serious vulnerability which causes exposure to various types of attacks. This paper proposes a framework for server authentication with focus on cognitive (S3) as well as non-cognitive (S4) methods.

The TLS security protocol [8] mainly provide server authentication type S4 because the authenticity of server certificates, and thereby of the server, are validated by the client system, not by the human user. Additional elements that must be combined with e.g. TLS are are needed to enable server authentication type S3, i.e. to enable the human user to authenticate the server system in a reliable fashion. Other security protocols that can provide server authentication of type S4 are e.g. IPSec [19] and SSH [27].

The concept of *authentication assurance* expresses the estimated reliability of authentication. Several governments have articulated and published frameworks for user authentication assurance, with specific assurance levels defined as a function of factors such as identity registration assurance and authentication mechanism strength. Unfortunately, no similar frameworks have been articulated for server authentication.

This paper proposes a general framework for server authentication assurance, where each specific authentication assurance level is determined as a function of a set of factors. The existing frameworks for user authentication assurance are briefly described next, in order to provide the background and a basis for comparison with our proposed framework for server authentication assurance.

II. Authentication Frameworks for User Authentication

The risk level of a specific service reflects the potential negative impact in case of wrong authentication. The required authentication assurance level shall balance that risk, meaning that as the risk of wrong authentication gets higher, the required authentication assurance level must be higher too. Authentication frameworks typically specify authentication assurance levels according to this principle.

There exists several frameworks for user authentication in public sector online service provision. These frameworks typically specify four or five UAALs (User Authentication Assurance Levels), i.e. from UAAL-1 (or UAAL-0) to UAAL-4, where a high number indicates a high assurance level, and AAL-0 indicates *"No authentication"*. The requirements for each AAL is roughly harmonized across the various national or regional frameworks although there can be minor differences in terminology interpretation.

We briefly review the national/regional frameworks from USA, EU, Norway, Australia and India for user authentication. Although the authentication frameworks are specified for, and aimed at online service provision in the public sector, they are also suitable for the private and commercial sector.

- **US NIST SP800-63.** Title: *Electronic Authentication Guideline* [6]. This framework describes technical requirements for user authentication assurance levels that are specified in the E-Authentication Guidance for U.S. Federal Agencies [5].
- **EU IDABC.** Title: *eID Interoperability for PEGS (Pan-European eGovernment services): Proposal for a multi-level authentication mechanism and a mapping of existing authentication mechanisms* [10]. This is in principle only a proposal, but is nevertheless still widely adopted by subsequent EU policies and technical requirements, such as the STORK Quality Authenticator scheme [13].
- **Norwegian FANR.** Title: *Framework for Authentication and Non-Repudiation in Electronic Communication with and within the Public Sector* [23]. This is the official framework for user authentication in the the Norwegian Government sector. It is clearly inspired by the NIST framework above, but contains far less details.
- **Australian NeAF.** Title: *National e-Authentication Framework* [22]. This framework is well structured and quite comprehensive. NeAF adopts the authentication assurance levels of Queensland Government Authentication Framework (QGAF) [24] which explicitly includes UAAL-0 aimed at anonymous access as well as pseudonymous user authentication.
- **Indian NeAF.** Title: *National e-Authentication Framework* [21]. This framework by the Indian Government is currently published as a draft. It represents an important companion for the Indian UID (Unique Identity) project and the biometric authentication program [4]. It includes UAAL-0 similarly to the earlier Australian NeAF.

The assurance level alignment of the above referenced authentication frameworks is illustrated in Table I below. It can be seen that there is a general consensus regarding the levels, although some of the frameworks use specific terms differently, such as ”High”, ”Very High” and ”Substantial”. For example, the assurance level USA-NIST ”Very High” is equal to Australia-NeAF ”High”, and ”Substantial” has a different meaning in EU-IDABC and India-NeAF. This might be a source of confusion, so that practitioners who need to map the authentication assurance levels of systems between e.g. USA and Australia, or between EU and India should be aware of the meaning behind the terms used in the respective frameworks. In order to minimize confusion it is advised to simply specify the user authentication levels by their number, e.g. use UAAL-4 instead of e.g Very High (USA-NIST) or High (Australia-NeAF).

The user authentication frameworks listed in Table I describe various factors that contribute to the robustness of the overall user authentication solution, as illustrated in Fig.3 below, where the rounded rectangles represent basic factors and the sharp rectangles represent derived factors. The terms used to describe each factor are generic for the purpose of this

Authentication Framework	***User Authentication Assurance Levels***				
NIST (USA) 2006	Little or no assurance (1)		Some (2)	High (3)	Very High (4)
IDABC (EU) 2007	✕	Minimal (1)	Low (2)	Substantial (3)	High (4)
FANR (Norway) 2008	Little or no assurance (1)		Low (2)	Moderate (3)	High (4)
NeAF (Australia) 2009	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
NeAF (India) 2011	None (0)	Minimal (1)	Minor (2)	Significant (3)	Substantial (4)

TABLE I
CORRESPONDENCE BETWEEN USER AUTHENTICATION ASSURANCE LEVELS IN AUTHENTICATION FRAMEWORKS

study. Other specific terms are often used in the frameworks listed in Table I, but their interpretation is the same as that of the terms of Fig.3.

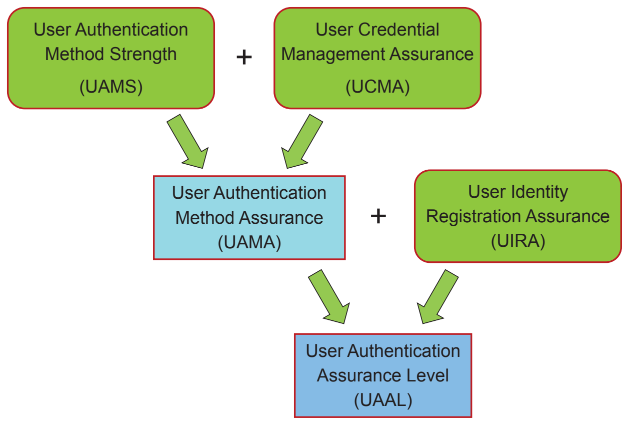

User Authentication Method Strength (UAMS)
+
User Credential Management Assurance (UCMA)
User Authentication Method Assurance (UAMA)
+
User Identity Registration Assurance (UIRA)
User Authentication Assurance Level (UAAL)

Fig. 3. Factors for user authentication assurance

The basic authentication factors indicated as rounded rectangles in Fig.3 are briefly described below.

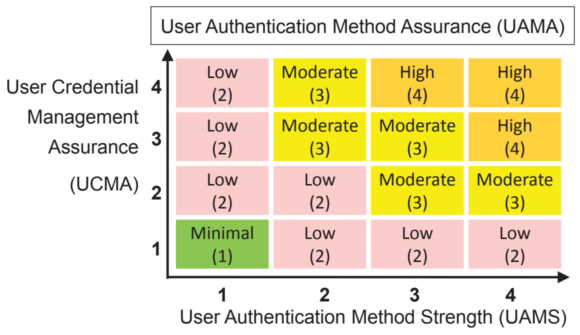
- **User Authentication Method Strength** (UAMS) refers to the intrinsic robustness of the specific solution used for authentication, such as password based, token based or biometrics based authentication, as well as any combination of these to form 2-factor solutions.
- **User Credential Management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

Fig.3 illustrates a general set of authentication factors and their relationships. Some of the frameworks of Table I above only cover a subset of these authentication factors. This is the case for the Norwegian FANR where user identity registration assurance (UIRA) is ignored because the correct registration of citizens is assumed as a prerequisite, in which case UAAL = UAMA.

The derived properties indicated as straight rectangles in Fig.3 are briefly described below.

- **User Authentication Method Assurance (UAMA)** refers to the combination of *User Authentication Method Strength* and *User Credential Management Assurance*.
- **User Authentication Assurance Level** refers to the overall robustness of the user authentication solution. The UAAL results from the combination of *User Authentication Method Assurance* and *User Identity Registration Assurance*. In case it can be assumed that all users have been properly registered, the UAAL is equal to the *User Authentication Mechanism Strength*. For example, the Norwegian ANRF ignores identity registration of citizens, because it assumes that every citizen is properly registered in the national person register.

User Authentication frameworks typically define a practical scheme for determining the UAAL as a function of the basic authentication assurance factors. For example, Fig.4 illustrates a look-up matrix for deriving User Authentication Method Assurance (UAMA) as a function User Authentication Method Strength (UAMS) and User Credential Management Assurance (UCMA).



User Authentication Method Assurance (UAMA)

User Credential Management Assurance (UCMA)	1	2	3	4
4	Low (2)	Moderate (3)	High (4)	High (4)
3	Low (2)	Moderate (3)	Moderate (3)	High (4)
2	Low (2)	Low (2)	Moderate (3)	Moderate (3)
1	Minimal (1)	Low (2)	Low (2)	Low (2)

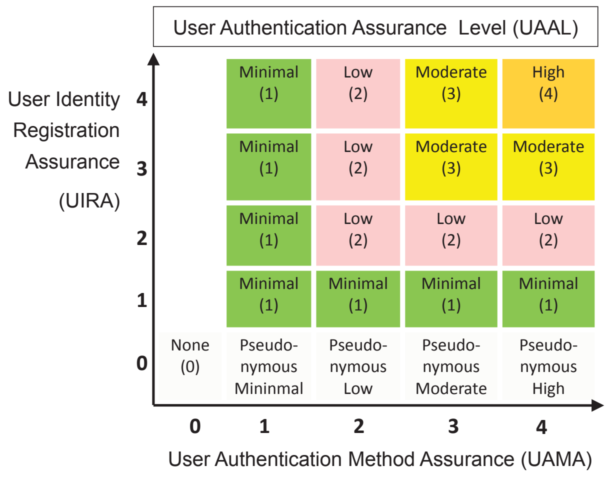
User Authentication Method Strength (UAMS)

Fig. 4. Look-up matrix for determining User Authentication Method Assurance (UAMA)

Similarly, For example, Fig.5 illustrates a look-up matrix for deriving the final User Authentication Assurance Level (UAAL) as a function User Authentication Method Assurance (UAMA) and User Identity Registration Assurance (UIRA).

III. Motivation for Service Provider Authentication Assurance

TLS [8] provides cryptographically strong server authenticated in a technical sense. Unfortunately, typical implementations of TLS do not provide semantic server authentication because they only support server system authentication by the



User Authentication Assurance Level (UAAL)

User Identity Registration Assurance (UIRA)	0	1	2	3	4
4		Minimal (1)	Low (2)	Moderate (3)	High (4)
3		Minimal (1)	Low (2)	Moderate (3)	Moderate (3)
2		Minimal (1)	Low (2)	Low (2)	Low (2)
1		Minimal (1)	Minimal (1)	Minimal (1)	Minimal (1)
0	None (0)	Pseudo-nymous Mininmal	Pseudo-nymous Low	Pseudo-nymous Moderate	Pseudo-nymous High

User Authentication Method Assurance (UAMA)

Fig. 5. Look-up matrix for determining User Authentication Assurance Level (UAAL)

client system, and not by the user. In order to provide meaningful server authentication a method that explicitly allows the user to authenticate the server system or the SP organisation is needed. The difference between server authentication by the client and server authentication by the user, which can be seen in Fig.1, might seem subtle and insignificant, but is nevertheless absolutely crucial. The problem is not due to weak cryptographic authentication mechanisms, but to poor usability of the typical TLS implementation [14], [16].

By analysing the security solution of TLS from a security usability perspective it can easily be seen that there are serious usability vulnerabilities that can easily be exploited by phishing attacks [15], [14]. This is briefly explained below.

The standard implementation of TLS in web browsers provides various information elements to the user. Unfortunately this information is often sufficient to make an informed conclusion about the identity of the web server.

The closed padlock in the corner of a typical browser represents one form of security information elements indicating that the web session is protected with TLS. However, the fact that it does not say anything about the identity of the server is a security usability vulnerability.

Additional security information is contained in the server certificate that can be inspected e.g. by double-clicking on the padlock. The mental load of analysing the content of a server certificate is intolerable for most people, which represents a security usability vulnerability. The following analysis will make this evident.

For example, even the definition of peer entity authentication according to X.800 [7] is inadequate in the case of phishing attacks, where the attackers actually claims its own identity in the formalism of TLS, and the TLS client (the browser) simply verifies the correctness of that claim. However, the claimed identity expressed in the certificate does not necessarily correspond to the identity that the user assumes. Thus, the problem has to do with human cognition of identity, for which cryptography provides no solution.

The identity of a SP organisation assumed by the user can be different from the identity validated through TLS. From the user's perspective, the ordinary name and graphical logo of the SP organisation constitutes a significant part of the SP identity. From the client browser's perspective, this identity cannot be used by TLS because normal names can be ambiguous and graphical logos can not be interpreted by TLS.

Certificates, which must be unambiguous, require globally unique names in order to allow efficient automated processing. Domain names mostly satisfy this requirement and have therefore been chosen to represent the identity of the bank in server certificates. Having different identities for the same entity can obviously cause problems. A simple way of solving this problem could be by requiring that users learn to identify online SP organisations by their domain names. Unfortunately this will not work because online banks often use multiple domain names depending on the service being offered.

Many companies' secure web sites have domain names with non-obvious domain names that do not correspond to the domain names of their main web sites. Another vulnerability is the fact that distinct domain names can appear very similar, for example differing only by a single letter, or looking very similar, so that a false domain name may pass undetected. How easy is it for example to distinguish between the following domain names?

```
www.pepes.com/
www.pepespizza.com/
www.pepesnypizza.com/
www.pepespizzeria.com/.
```

The fundamental problem is that, although domain names are designed to be readable by humans, they provide poor usability for identifying organisations in the real world. Ordinary names such as ”Pepe's Pizza”, when expressed in a local semantic context, are suitable for dealing with organisations in the real world, but not for global online identification and authentication. The consequence of this mismatch between names used in the online world and in the real world is that users do not know which unique domain name to expect when accessing online services. Without knowing which domain name to expect, authentication becomes meaningless. In other words, the users do not know what security conclusion to draw.

Another problem with the current implementation of TLS is its dependence on the browser PKI that has a relatively large number of CAs (Certificate Authorities), that in fact create a ”weakest link” situation. Several authors describe the problems with the browser PKI [11], [17], [26]. The specific problem is that any CA can sign any certificate they want. For instance two different CAs can sign a certificate for example.com, and as long as both CAs has their root certificate in the web browsers they will be evaluated as valid. This is problematic when a CAs private key is falling into wrong hands or a CA is issuing a certificate to someone that do not have the right to it. There have been examples of both; [18] and [20]. The problem is mostly how the SSL PKI is built up, and not with the technology itself.

In summary, our analysis of current server authentication has exposed serious vulnerabilities that continue to be exploited by criminals to mount successful phishing attacks. A framework for server authentication assurance is needed to address these vulnerabilities.

IV. Proposed Framework for Server Authentication

The existing authentication frameworks listed in Table I do not specifically focus on how users can make sure that an accessed online service actually is hosted by the correct and intended SP organisation. The draft Indian NeAF describes a method based on a“watermarks” [21], whereby the user selects an image and/or text during registration to a SP. When the user accesses the SP with their use name the next time, then the server of the SP shows the image and text (if it exists), which gives the user a way to confirm that this is the previously accessed SP before the user enters their password.

Research into the usability of such a system has shown that most users do not react if the “watermark” is replaced with a text describing that the service is undergoing an upgrade, and will enter their password anyway [25]. With todays web capabilities it is not a far stretch to imagine this image to be extracted by some kind of proxy server, and delivered to the user by a malicious website.

The US NIST SP800-63 also have similar examples of personalization measures to be implemented by the services. They also mention that there is no foolproof way to prevent the user (Claimant) from revealing any sensitive information to which he or she has access [6].

Because these documents focus on server side solutions, they are missing important points, which are the authentication of the server by the user. The frameworks propose applying simple methods for the user to authenticate the server and try to highlight some “best practices”, at the same time pushing the technology for the user authentication to its limit. However, a specif framework for server authentication assurance would have to specify a more complete set of methods and principles for server authentication.

A. Technologies in Support of Server Authentication

DNSSEC is a solution to ensure that the domain name given by the domain name system (DNS) is not modified to send a user to another place then originally intended by the owner of the domain name. DNSSEC builds on asymmetric cryptographic signing of DNS records in the name servers [3]. This is a strong hierarchical public key infrastructure, where the root-key is managed by The Internet Corporation For Assigned Names and Numbers (ICANN) [1]. ICANN also manage the root-servers for all top level domain names. DNSSEC (if checked by the user or its client) makes it hard for an attacker to make false DNS records and misleading a client system to a wrong server [2]. If there had been a requirement to use DNSSEC then all the major browsers would have supported DNSSEC much faster.

A petname system is a set of personalized nicknames for the services the user is using. [9] describes The Petname Model as a naming system that can implement all three properties of Zooko's triangle; Global, Memorable and Unique. The service nickname (e.g. the domain name system) is global and memorable, and the petname is memorable and unique. This is a user centric system, managed fully by the user.

The TrustBar [12] for the Mozilla and Firefox browsers seems to represent a promising approach to the problem by making authentication semantically meaningful. The TrustBar solution consists of personalising every server certificate that the user wants to recognise by defining a personal petname for it [9]. The petname can e.g. consist of an image or a audible tune that the user can easily recognise. Unfortunately solutions like the TrustBar are not widely used.

B. *Factors for Server Authentication Assurance*

It is important to involve the user when he or she registers to a new service, both to check if the place is correct and to show the user what to expect the next time. As most of the technical authentication processing is transparent, the users are easily deceived. It is therefore important to have solutions that allow users to understand the processes taking place during the authentication. As [25] pointed out, all 63 subjects who participated in an experiment entered user name and password even when TLS was not used, because they did not know the difference between the browser using TLS (as indicated by https) and not using TLS (as indicated by http).

Users have to understand how the security mechanisms work, not in detail, but have a general overview, and need to know and understand the required security actions and security conclusions [14]. The higher the authentication risk level, the more this understanding is required.

The three basic factors for user authentication assurance, as indicated in Fig.3 does not cover the concept of identity cognition, because the authentication is executed by a system which simply applies syntactic identity recognition. However, for server authentication by humans it is necessary to introduce the basic factor ICA (Identity Cognition Assurance). Identity cognition by the human user consists of paying attention to the presented server identity, understanding its nature, and making a decision whether it is the expected or desired identity for the specific communication session. Similarly to the case of user authentication frameworks, for server authentication frameworks the basic factors SAMS (Server Authentication Method Strength), SCMA (Server Credential Management Assurance) and SIRA (Server Identity Registration Assurance) are also included, as indicated in Fig.6 where where the rounded rectangles represent basic factors and the sharp rectangles represent derived factors.

The combination of the input factors SIRA and ICA result in the derived factor SIUA (Server Identity Usage Assurance), which when combined with SAMA results in the overall SAAL. The interpretation of the derived SIUA factor in in Fig.6 is that the usage of the server identity in business processes is safe because the identity has been properly registered and because human operators clearly understand what it represents.

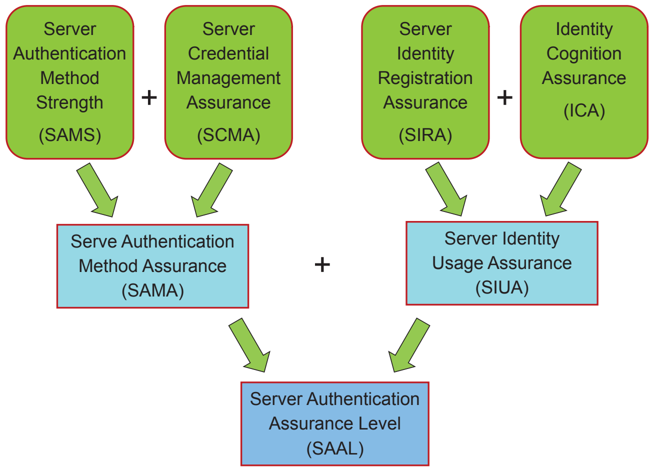

Server Authentication Method Strength (SAMS)
+
Server Credential Management Assurance (SCMA)
Server Identity Registration Assurance (SIRA)
+
Identity Cognition Assurance (ICA)
Serve Authentication Method Assurance (SAMA)
+
Server Identity Usage Assurance (SIUA)
Server Authentication Assurance Level (SAAL)

Fig. 6. Factors for server authentication assurance

The effect of introducing the basic ICA factor in Fig.6 is to explicitly require that the human user take a conscious choice after having recognised the specific server identity with its unique name. High ICA is not needed in case of low SAAL. It is acceptable to have a low SAAL when it would have little consequence that the user misunderstands the server identity and accesses the wrong server without knowing it. However, the SAAL must be higher when the consequence of accessing the wrong server is more severe. This also forces the ICA to be higher, which puts stronger requirements on the authentication methods and their usability, so that that human users can be certain that they access the intended server.

The security of a service extends beyond the server system and SP domain, and should include all users enrolled into it. This is difficult to achieve when serving millions of users. The responsibility to educate the users on how to use the services with adequate identity cognition assurance lies with the SP, as the SP can not expect all their users to have this knowledge prior to enrollment. When planning a service there is always a trade-off between cost and security level, where ensuring ICA and educating users might be seen as a significant cost, but it may be worth the effort when the risk of wrong server authentication is high.

Instead of using a look-up matrix to combine the SAMA and SIUA, similarly to Fig.4 and Fig.5 we find it more appropriate to simply apply the principle of the weakest link, i.e. that for a specific required SAAL, all the input factors (SAMS, SCMA, SIRA, ICA) must have at least the same level as the SAAL. Another way to express the same principle is that the SAAL is dictated by the minimum level of SAMS, SCMA, SIRA and ICA as illustrated in Fig.7.

C. *Requirements for Server Authentication Assurance Levels*

We believe that it would require a major study to specify in detail requirements for the four factors of server authentication assurance levels according to Fig.6. The requirements we provide below must therefore be seen as indicative. First, the

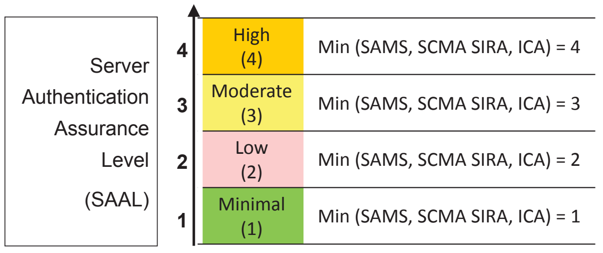

Server Authentication Assurance Level (SAAL)
4
3
2
1
High (4)
Moderate (3)
Low (2)
Minimal (1)
Min (SAMS, SCMA SIRA, ICA) = 4
Min (SAMS, SCMA SIRA, ICA) = 3
Min (SAMS, SCMA SIRA, ICA) = 2
Min (SAMS, SCMA SIRA, ICA) = 1

Fig. 7. Determining Server Authentication Assurance Level (SAAL)

server authentication assurance levels with their corresponding relative risk levels are indicated.

Server Authentication Assurance Levels (SAAL)

- **SAAL-1.** Minimal server authentication assurance is required when wrong server authentication would have minimal or no negative impact for services in this level, and attackers have little incentive to spoof the server, e.g. because the user does not provide sensitive information to the SP.
- **SAAL-2.** Low server authentication assurance is required when wrong server authentication would have some negative impact for services in this level, and attackers could have some incentive to spoof the server to mislead or steal user credentials or user information.
- **SAAL-3.** Moderate server authentication assurance is required when wrong server authentication would have significant negative impact, and attackers could have strong financial or political incentive to spoof the server to mislead or steal user credentials or user information.
- **SAAL-4.** High server authentication assurance is required when wrong server authentication would have severe negative impact for services in this level, and attackers could have strong financial or political incentive to spoof the server to mislead or steal user credentials or user information.

At this point in time there exist several different server authentication methods that can be used for the different assurance levels, some are not widely deployed although theoretical and implemented solutions exist. The list below is indicative for possible server authentication method strengths.

Requirements for Server Authentication Method Strength (SAMS)

1) **SAMS-1.** The server is identified by its domain name or IP address. Otherwise there are no specific requirements.
2) **SAMS-2.** It is sufficient to have a SSL/TLS connection with a valid SSL certificate. It must be possible to check if the certificate is valid, and to inspect the unique domain name of the SP.
3) **SAMS-3.** Server certificates based in DNSSEC is required in order to avoid the weakest link vulnerability of the browser PKI. DNSEC also ensures that results returned by DNS are authentic. Currently, DNSSEC is not widely implemented, and is not supported by all web browsers.
4) **SAMS-4.** The same requirements as for SAMS-3 above. In addition, a petname system is required to support identity cognition for the SP identity and name.

The management of server authentication credentials is quite different from that of user authentication credentials. The list below is indicative for possible server credentials management assurance.

Requirements for Server Credentials Management Assurance (SCMA)

1) **SCMA-1.** No specific requirements.
2) **SCMA-2.** Online installation on the client systems of PKI root public keys used for server certificate validation. Private server keys can be stored in server memory with adequate protection.
3) **SCMA-3.** Online installation on the client system of the the DNSSEC PKI root public key used for server certificate validation. Private DNSSEC server keys can be stored in server memory with adequate protection.
4) **SCMA-4.** The same requirements as for SCMA-3 above, but the PKI root public key must be installed manually on client systems, and private server keys must be installed, stored and processed in trusted hardware.

The registration of server identities is quite different from that of user identities. The list below is indicative for possible methods for achieving server identity registration assurance.

Requirements for Server Identity Registration Assurance (SIRA)

1) **SIRA-1.** No specific requirements.
2) **SIRA-2.** Representatives from SP organisations can obtain server certificates online by providing evidence of legally representing the specific SP organisation.
3) **SIRA-3.** Server certificates must be part of the DNSSEC PKI hierarchy, so that server certificates can be obtained from the DNS registrar only. Requirements for obtaining server certificates are the same as for obtaining a domain name.
4) **SIRA-4.** The same requirements as for SIRA-3 above, but representatives of SP organisation must present in person at DNS registrar.

Proposed requirements for identity cognition assurance are given below. **Requirements for Identity Cognition Assurance (ICA)**

- **ICA-1.** No specific requirements.
- **ICA-2.** It is required that the authentication system enables to inspect the unique name of the authenticated server.
- **ICA-3.** It is required that the unique name of the server be mapped to a local petname specified by user. The petname must be unique within the local domain.
- **ICA-4.** Same requirements as for ICA-3 above. In addition it is required that the user explicitly specifies the

server identity, either directly with the globally unique server name, or indirectly through the local server petname, in order to indicate the exact server identity that is expected in the communication.

V. Concluding Remarks

The lack of focus on service provider authentication by the user is a blind spot in the academic and information industry security communities. The intention of this paper is to put the focus on the importance of ensuring adequate server authentication. We believe that it is timely to establish frameworks for server authentication in order to reflect and balance the existing frameworks for user authentication assurance.

An important question to ask is whether different user authentication and server authentication assurance levels will have consequences for the overall security in a session between a user and a server system. For example, assume that a user is required to use strong user authentication when accessing a network server, while at the same time being unable to authenticate the server identity. An obvious attack in this situation would be to trick the user to access a server controlled by the attacker, which could enable the attacker to collect user credentials which in turn could be used to masquerade as the user. It therefore seems reasonable to have a balance between UAAL and SAAL on online applications. Frameworks for user authentication and server authentication should therefore be considered in conjunctions, and we encourage national governments with programs for online service provision to consider frameworks for both user authentication as well as server authentication, and to consider their relationship and interdependencies.

References

[1] J. Abley and J. Schlyter. DNSSEC Trust Anchor Publication for the Root Zone. http://data.iana.org/root-anchors/draft-icann-dnssec-trust-anchor.txt, 2010.

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

[3] Giuseppe Ateniese and Stefan Mangard. A new approach to dns security (dnssec). In *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, pages 86–95, New York, NY, USA, 2001. ACM.

[4] Yojana Bhavan and Sansad Marg. *Biometrics Design Standards For UID Applications*. Unique Identification Authority of India, Planning Commission, New Delhi 110001, India, December 2009.

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

[6] William E. Burr, Donna F. Dodson, and W. Timothy Polk. Electronic Authentication Guideline – NIST Special Publication 800-63. Technical report, National Institute of Standards and Technology, 2006.

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

[8] T. Dierks and E. Rescorla. *RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2*. IETF, August 2008. URL: http://www.ietf.org/rfc/rfc5246.txt.

[9] Md Sadek Ferdous, Audun Jøsang, Kuldeep Singh, and Ravi Borgaonkar. Security Usability of Petname Systems. In *Proceedings of the 14th Nordic Workshop on Secure IT systems (NordSec 2009)*, Oslo, October 2009.

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

[11] James M. Hayes. The problem with multiple roots in web browsers - certificate masquerading. In *7th Workshop on Enabling Technologies, Infrastructure for Collaborative Enterprises (WETICE '98)*, pages 306–313. CAUSA Proceedings, IEEE Computer Society, Palo Alto, June 17-19 1998.

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

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

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

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

[16] Audun Jøsang. Trust Extortion on the Internet. In *7th International Workshop on Security and Trust Management (STM 2011)*, Copenhagen, June 2011.

[17] Audun Jøsang and Kashif Sana Dar. Server Certificates based on DNSSEC. In *Proceedings of NordSec2011*, Tallin, October 2011.

[18] Gregg Keizer. Computerworld: DigiNotar dies from certificate hack caper. http://www.computerworld.com/s/article/9220175/ DigiNotar_dies_from _certificate_hack_caper, 2011.

[19] S. Kent and K. Seo. *RFC 4301 - Security Architecture for the Internet Protocol*. IETF, December 2005. URL: http://www.ietf.org/rfc/rfc4301.txt.

[20] Microsoft. Microsoft Security Bulletin MS01-017 (March 22, 2001): Erroneous VeriSign-Issued Digital Certificates Pose Spoofing Hazard. http://www.microsoft.com/technet/security/bulletin/MS01-017.asp, 2001.

[21] Ministry of Communications and Information Technology. *National e-Authentication Framework (NeAF)*. Government of India, Version 1.0, 1 September 2011.

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

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

[24] Office of Government ICT. *Queensland Government Authentication Framework (QGAF)*. Queensland Government, October 2006.

[25] S.E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators. In *Security and Privacy, 2007. SP '07. IEEE Symposium on*, pages 51 –65, may 2007.

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

[27] T. Ylonen and C. Lonvick. *RFC 4251 - The Secure Shell (SSH) Protocol Architecture*. IETF, January 2006. Available at: http://www.rfc-editor.org/.

(♯) Appears in the proceedings of the 10th Annual Conference on Privacy, Security and Trust (PST 2012). Paris, July 2012.

(§) The work reported in this paper has been partially funded by the Franco-Norwegian Foundation Project 1/11-FNS, and by the EUREKA Project 7161 Lucidman.