

Diameter WebAuth: An AAA-based Identity Management Framework for Web Applications

Niklas Neumann and Xiaoming Fu
Computer Networks Group, University of Goettingen, Germany
{neumann, fu}@cs.uni-goettingen.de

Abstract—With an increasing number of personalized information and services offered on the Internet, especially the World Wide Web, effective identity management solutions are demanded by application providers. Instead of a web-based stand-alone approach, we extend existing network-based AAA mechanisms to be useable for identity management by web applications. Our proposal, Diameter WebAuth, allows to seamlessly integrate web-based services into a Diameter infrastructure for authentication, authorization, credit-control and identity management purposes. Diameter WebAuth offers comparable features to web-based identity management solutions, benefits from the maturity and wide deployment of the Diameter protocol, and takes advantage of existing AAA setups.

I. INTRODUCTION

Every day countless users are accessing various personal and personalized information on the Internet, especially the World Wide Web. In order to provide each user proper access, web applications need to be able to establish the user's identity. Identity management is a concept to unify and facilitate such user identification. There are several approaches to enable identity management in web applications like OpenID [1], the Liberty Alliance project [2] or Microsoft CardSpace [3]. They emerged rather recently, are specifically designed for web applications and take the requirements and characteristics of a web environment into account. However, those approaches are developed independently of existing technologies and, therefore, introduce additional complexity, dependencies and bottlenecks. Also, they require different knowledge and expertise to be set up, operated and maintained. Moreover, they still rely on external services for authentication such as LDAP or database servers.

On the other hand, the problem of access control has been extensively explored in the networking field. Network access and authentication, authorization and accounting (AAA) protocols [4] have been developed and deployed to verify and control access privileges of network users. Those protocols are well established and mature. They are, however, not designed for the web environment and cannot be used together with web applications without special adaptations. Examples for such network access control protocols are Radius [5], Diameter [6] and Kerberos [7].

In this paper, we propose an AAA-based identity management framework for web applications, called Diameter WebAuth. It is based on Diameter [6] and therefore benefits from the maturity and wide deployment of this protocol. Moreover, it enables web applications to be directly and

seamlessly integrated into Diameter setups. This allows for an integrated enterprise identity management based on a Diameter setup and saves additional management or development effort. Our proposal is a new Diameter application that is oriented at the needs of identity management for web applications. Diameter WebAuth combines authentication and authorization mechanisms with credit-control mechanisms and the support for generic identity attributes. Offering these functions on top of the mechanisms the Diameter base protocol provides, such as inter domain routing, federating and delegation, provides Diameter WebAuth with a set of facilities that are at least comparable to other existing approaches.

Compared with other approaches Diameter WebAuth is not a proprietary or completely new technology. It doesn't require extensive implementation, additional setups, or familiarization. Because it is based on the Diameter protocol, a WebAuth-enabled web site can be seen as just another network access device from the maintenance point of view. A working Diameter setup provided, any web site can be integrated into the AAA infrastructure with minimal effort. Furthermore, there is no need for maintaining an additional identity management infrastructure.

In networks without an existing AAA infrastructure, Diameter WebAuth introduces a basic Diameter setup which can be extended to offer AAA services to a wide range of network-related services. In any case, Diameter as a well-established and mature protocol offers reliability and safety that new approaches still have to achieve.

The remainder of this paper is structured as follows. Section II presents related work to this paper, followed by Section III where the design of the Diameter WebAuth proposal is introduced and explained. Section IV provides a further analysis of our proposal based on a working prototype implementation. Section V concludes this paper.

II. RELATED WORK

Existing approaches can roughly be grouped into three categories. First, there are efforts, which directly aim at identity management in web applications. They are recently created and specifically designed for the web-based environment. OpenID [1], the Liberty Alliance [2] or Shibboleth [8] are part of this category. They are all based on HTTP and follow a redirection pattern. In this pattern the actual authentication is conducted between the end user and the identity provider without interference or relay operations from

the service provider. The end user is redirected between the identity provider and the service provider. There are a number of disadvantages for approaches using the redirection pattern. First, the redirecting relies on the possibility of the user to contact his identity provider. In scenarios where a web application is used to secure general network access, for example for a WiFi HotSpot, the redirect pattern creates a predicament. The user should be authenticated before he is granted network access, but to be authenticated he already needs access to reach his identity provider. Also, the redirection of the user back and forth between the web application and the identity provider is a point of attack for phishing attacks which aim at gaining access to the credentials of the user. Especially for the OpenID framework this was stated as a major issue [9]–[11]. Furthermore, the exact authentication methods are not part of the specifications which leaves a crucial element of the overall security unsettled and out of the control of the service provider.

The Windows CardSpace technology [3], in contrast, specifically aims at the authentication process. It provides a so-called user-controlled identity management system [12] which uses authentication based on electronic signatures. However, the digital user identities, like SSL certificates, are not easily portable and, therefore, not universally useable like, for example, a username and a password. For this reason it seems questionable whether the CardSpace technology is suitable for web services since it would constrict their general accessibility (e.g. from Internet cafes, a friend’s computer or the workplace).

The last category are network access control and AAA protocols such as Radius [5], Diameter [6] or Kerberos [7]. Kerberos needs to be supported by both the client and the server and is not present in common web environments. In September 2007 the MIT announced the launch of the Kerberos Consortium with the goal to advance the propagation of Kerberos. Their plans also include web authentication [13]. Currently, however, there is only very limited support available for Kerberos in web servers [14] or web browsers [15], [16]. Radius and its proposed successor Diameter [6] are intended to be used between a network access point and an authentication server. The end user client is usually not modified and uses only its application-specific protocol (e.g., HTTP in a web environment).

Compared to the other technologies Kerberos and Diameter are both rather mature protocols. They were designed from a network point of view without any focus on web technologies. However, they are intended for similar purposes as the other identity management approaches. They are suited to authenticate and authorize users and to allow them access to (network) services and resources. Our conclusion is, that while the web-based approaches are rather recent, the network-based approaches are well established and proven and it can be assumed, that if Kerberos or Diameter can be adapted to a web-based environment, they would be valid options as basis for an identity management system for web applications. Diameter seems to be more suitable for such an adaptation since it doesn’t require support in the end user client. Our proposal, therefore, is based on the Diameter protocol.

III. DESIGN

This sections describes the “Diameter Application for Authentication and Authorization in Web Applications” (Diameter WebAuth). The intended area of application for Diameter WebAuth are web applications that want to utilize a Diameter server for authentication and authorization of their users. It enables a Diameter server to supply web sites that implement a Diameter WebAuth client with data to authenticate its user via common HTTP authentication methods. Furthermore, it allows the Diameter client to authorize the access to resources or services provided by the web site.

A relevant usage scenario of Diameter WebAuth is deployment in identity management frameworks where there may be different trust relationships between the user, the web application server and the authentication server. This means

- 1) No re-usable authentication credentials are shared with the web application server,
- 2) The authentication server can hold back authentication or authorization information until they are actually needed by the web application server.

Diameter WebAuth specifically addresses the authentication and authorization requirements for the purpose of identity management.

Diameter WebAuth does not rely on other Diameter applications and is designed to be lightweight and straightforward. This makes it feasible in resource-constrained environments, such as embedded systems.

A. Motivation and Goals

Several Diameter applications have been defined for various services, like network access, Mobile IP or the Session Initiation Protocol [17]–[19]. However, the existing applications are not particularly designed for the use in combination with web applications, many of which require authentication and authorization. Specifically, they do not offer methods suitable for authentication and authorization in a web-based environment, for example the HTTP Digest Authentication. Or they are intended for other applications and require extensive and complex implementation work which, however, is not needed for the intended use in web-based environments. Web applications (or web servers itself for that matter), therefore, implement proprietary authentication back-ends or use services that are not primarily designed for extensive authentication operations.

Such services are, for example, LDAP servers, database servers or IMAP servers. This is often the case, even though there is an AAA service like Diameter available within their administrative domain. We, therefore, introduce a new Diameter application that allows web servers and web applications to utilize a Diameter AAA infrastructure for authentication and authorization purposes.

Diameter WebAuth allows for a Diameter client and a Diameter server to be located either in the same or in different administrative domains. This allows for three party scenarios where, for example, a user has signed up with a dedicated identity management provider which itself provides authentication

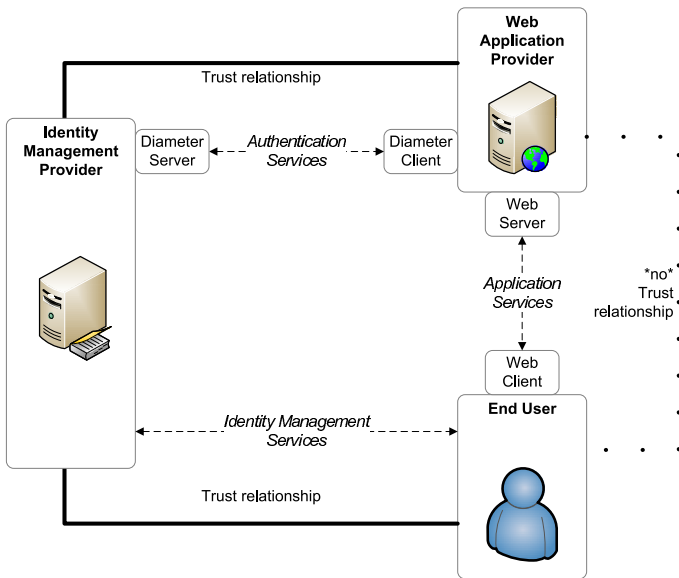


Fig. 1. Trust relationships in a three party scenario

services for a web application provider. As shown in Figure 1 in such three party scenarios, the end-user has a profound trust relationship with the identity management provider but not with the provider of the service he is accessing. Therefore special attention has to be paid to assuring the privacy of the end-user towards the application service provider while enabling the service provider to render its service.

Overall, the goals for the Diameter WebAuth application are as follows: *Lightweight* and easy to implement in Diameter clients as well as Diameter servers. Examples for Diameter clients employing WebAuth are web servers and web applications, or more generally, any entity that provides services using a web-based user interface. *Secure and private* for scenarios where the Diameter server and the Diameter client are not part of the same administrative domain. This is, for example, the case when the Diameter server is operated by a third party such as an identity management provider. *General* with regards to identity information, which are able to transport and manage a wide range of identity information data.

The identity management features of Diameter WebAuth are: *Authentication and authorization* of a user to allow application providers the delegation of those tasks. *Credit-control* to which includes debiting a user for services provided by the application provider. This allows for web applications to process monetary payments. Support for *identity attributes* to enable web applications to query personal information about the user. All of those features are provided within the Diameter WebAuth identity management framework without introducing additional complexity beyond the initial deployment of a well known and established protocol.

B. Authentication and authorization

Diameter WebAuth extends the facilities provided by the Diameter base protocol for a Diameter client to authenticate a user. Two requirements are taken into account regarding the

authentication. First, Diameter WebAuth must use standard authentication methods that are supported by the user client. The reason is that Diameter WebAuth only specifies the protocol between the Diameter client and the Diameter server. It cannot alter or adjust the service specific protocol between the user client and the Diameter client, HTTP in this case. The second requirement is that the authentication method needs to provide protection against unauthorized access to secret credentials. In case of username/password authentication this would be the password. Particularly this means that in scenarios where the Diameter client is outside the trust domain of the Diameter server, the secret credentials needs to be protected against the Diameter client itself.

The most common authentication method supported by web browsers is username/password authentication. RFC 2617 [20] specifies two HTTP authentication methods which are widely supported by web browsers: basic authentication and digest access authentication. While basic authentication exchanges the credentials including the password in cleartext, digest access authentication uses a one-way hash function to prevent sending the password in cleartext. Although digest authentication is not intended to be an absolutely secure authentication scheme, it serves the purpose of protecting the user password against snooping by any entity between the user client and the authenticator, which in this case would be the Diameter server. Besides HTTP digest access authentication, Diameter WebAuth will, nevertheless, support basic authentication as well. It can be used as a fall back in environments where digest authentication is not available or not necessary and to more generally support different authentication mechanisms, for example, HTML-form-based authentication.

HTTP Digest Authentication as described in [20] can be requested by the Diameter client in the initial authentication/authorization request (AA-Request) message. The HTTP digest authentication scheme uses a challenge/response mechanism, therefore, multiple protocol round-trips are needed. An example of an authentication session using the HTTP digest authentication scheme is shown in Figure 2. When a user client sends a request for a protected resource without including any credentials (1), the Diameter client starts the authentication process. It sends an AA-Request to the Diameter server (2) with the requested authentication method set to digest authentication. The Diameter server then generates a digest challenge and sends it to the client in an AA-Answer (3). Using the credentials provided by the Diameter server, the Diameter client can construct an HTTP response with the appropriate WWW-Authenticate header and send it to the web user client (4) to challenge an authentication. Next, the client assembles his authentication credentials and sends another request to the web server (5) including the user's credentials. The Diameter client assembles an AA-Request to the Diameter server with the corresponding information from the clients request (6). If the credentials match the records in the Diameter server, it returns an AA-Answer indicating a successful authentication (7). After receiving a positive authentication response, the web server can respond to the user clients request (8) and grant

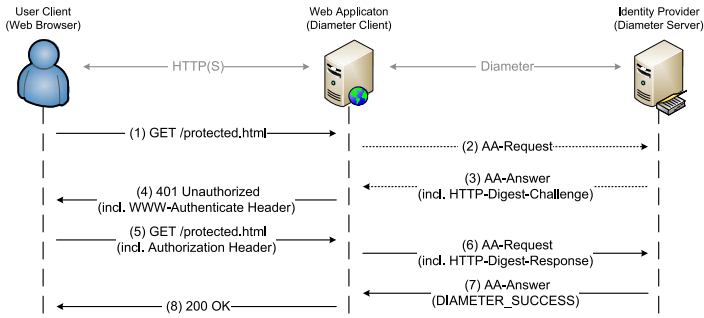


Fig. 2. Diameter WebAuth using HTTP digest authentication

access to the protected resource.

To facilitate different roles and access levels for the authorization process, Diameter WebAuth defines a number of parameters which are passed from a web server to the Diameter server. The Diameter server uses these parameters to determine whether the user is granted access to a requested resource. Such parameters are, for example, the source address of a request, the URI of the requested resource or a service identifier which is associated with the resource. Using these parameters allows a web application to request user differentiated authorization from the Diameter server.

C. Credit-Control

In addition to common identity management functions, Diameter WebAuth provides web applications with some basic accounting support. This allows web applications to relay fees it wants to charge the user for accessing a particular service to the Diameter server. The Diameter server then checks the charge against the user's account and approves or denies the charge. This mechanism offers a number of advantages compared to a solution where the web application has to implement accounting functions on its own. First of all, there is not implementation effort on part of the web application. By utilizing Diameter WebAuth the web application is automatically capable of processing credit charges. Second, the web application provider does not need to concern itself with invoicing, credit verification, transaction processing or security concerns. Thirdly, the end user only needs to trust its identity provider with payment related data. This increases the trust the user has into the payment system and also allows for a, for the web application, transparent handling of payment methods. The user can maintain several different payment methods and switch between them as needed without effecting the web application provider in any way. Last but not least, centralizing the payment processing at the identity provider generates synergy effects for the affiliated web application providers because they effectively share a payment infrastructure. Especially for small web sites, with only a very limited number of services or products they want to charge for, it is beneficial to be able to use the identity provider also as a payment provider for their customers.

The Diameter Credit-Control application [21] specifies extensive support for credit handling in Diameter environments. The Diameter WebAuth specification uses a small subset

of this specification to provide basic credit control support. This also allows to integrate a Diameter WebAuth client into a full-blown Diameter credit-control infrastructure. Diameter WebAuth implements the Credit-Control-Request and Credit-Control-Answer commands to support the credit functions for "Balance Check", "Direct Debiting" and "Refund".

D. Identity attributes

Reliable authentication is probably the most important feature an identity management framework has to provide. However, once the identity of a user is verified by the web application, further personalization of the offered services is possible. For example, the user can be welcome with a personal greeting that includes his name, or a web site can offer the user to use his personal address for shipping goods he orders. Those characteristics describing an identity beyond the login name are called identity attributes. This specification aims at including identity attributes into its framework. This allows to store information that are closely linked to an identity in the same central manner the identity is stored.

Diameter WebAuth provides some basic means to transport identity information over the Diameter protocol. This can be attributes like first name, last name, address or other contact information. The scope and value of the potential information is dependent on the schema used to exchange these identity information. The schema is interchangeable and has to be predefined between the web application provider (Diameter client) and the identity management provider (Diameter server). Therefore the actual format for the values of the Identity-Attribute-Request AVP and the Identity-Attribute-Value AVP is dependent on the employed identity information schema and is outside the scope of Diameter WebAuth. Suitable schemes can, for example, be defined in another Diameter application, in standards written by standardization bodies, or in service-specific documentation.

Identity attributes are transported using dedicated Identity-Information-Query/Response AVPs. Multiple such AVPs can be included in a Diameter request or response to query multiple identity information. Every Identity-Information-Query is processed separately by the Diameter server, without a special order.

IV. FURTHER ANALYSIS

An important consideration during the design of Diameter WebAuth was that its features will be comparable with other web-based approaches to identity management. In this section we will evaluate the features of Diameter WebAuth and compare it with other approaches.

A. Diameter WebAuth properties

a) *End user authentication:* The Diameter WebAuth specification covers end user authentication. While the exact backend used is implementation specific (e.g. LDAP, NIS, local database), the procedure to authenticate a user is not. One disadvantage over approaches that only redirect the user to an authentication provider and leave the particular authentication

details open, is that Diameter WebAuth is less flexible. The advantage, however, is that the web application provider can be assured of the authentication procedure which may allow for a tighter overall security. For example, an authentication provider which uses an inherently insecure authentication mechanism like just the input of a last name in a form is not feasible in Diameter WebAuth, but it is in other approaches. Moreover, Diameter WebAuth can be extended to support various other authentication methods by updating the specification.

b) Password-based: At the moment, Diameter WebAuth uses the HTTP authentication methods specified by RFC 2617. By implementing those methods, Diameter WebAuth provides username/password authentication which is probably the most common method used by web applications at the moment. Furthermore it is probably the only standardized authentication method that is available in current web browsers besides derived authentication methods like IP-based authentication or SSL certificate authentication. Since HTTP digest authentication is challenge-response based it does not expose any secret credentials (i.e. the user's password) to any third party including the web application or an attacker that gets access to the authentication data exchange. Furthermore, username/password based authentication methods still have a number of advantages over other proposed methods like portability, mobility, and wide availability.

c) Authorization: Service differentiated user authentication is provided by Diameter WebAuth in order for web applications to support areas with different access restrictions. The particular service identifiers are freely allocatable between the Diameter server provider and the web application provider. This allows for user authentication as fine grained as needed by the web application.

d) Single sign-on/sign-out: Although single sign-on and single sign-out is not explicitly supported by Diameter WebAuth, the HTTP digest access authentication scheme used, allows to define a protection space which can span multiple servers. This mechanism allows the end user client to determine the set of URIs for which the same authentication information may be used. This can be used by the Diameter server to implement a crude single sign-on functionality for the Diameter clients it is responsible for.

e) Accounting: Basic credit related functions are available in Diameter WebAuth as credit control operations, carried over from the Diameter Credit-Control application. They allow a web application to charge and refund its users with currency or application related service units. If and to what extent the Diameter service provider is taking monetary responsibilities for its users needs to be arranged between the web application provider and the Diameter server provider.

f) User client support: The Diameter WebAuth application does not make any fundamental demands on the end user client. Only the HTTP authentication methods used by Diameter WebAuth need to be supported by the client. But first of all, they are more related to the HTTP protocol which support is a basic requirement for every web browser. This

TABLE I
COMPARISON OF IDENTITY MANAGEMENT SYSTEMS

Feature	OpenID	Liberty Alliance	Card-Space	Diameter WebAuth
Authentication	no	no	yes	yes
Password-based	possible*	possible*	no	yes
Authorization	no	yes	yes	yes
Single sign-on/ sign-out	yes/ no	yes/ yes	no/ no	possible**/ possible**
Accounting	no	no	no	yes
Requires user client support	no	no	yes	no
Technologies	HTTP	SAML	XML	IP, HTTP
Identity attributes	no***	yes	yes	yes
Maturity	medium	medium	low	low
Primary focus	Distributed identities (Web)	Trust relations (Web)	Authenti- cation (Web)	AAA + identity information

*Depends on the authentication provider.

**Depends on the browser implementation of RFC 2617 [20].

***Available through an OpenID service extension [22].

means, that Diameter WebAuth rather uses basic features of its end user's service protocol than expecting exceptional and incoherent traits. And second of all, even in the case HTTP authentication is not available in the end user's client, Diameter WebAuth can fallback on form-based authentication.

g) Identity attributes: Diameter WebAuth supports the exchange of identity attributes by implementing the Identity-Information command. It allows a web application to query the Diameter server for attributes describing a persons identity. The extend of the available attributes, their format and the query/response syntax to exchange them is not fixed and needs to be predefined between the web application provider and the Diameter server provider. This allows to use virtually any identity information schema with Diameter WebAuth.

h) Primary focus: Diameter WebAuth focuses on bringing network-based AAA mechanisms into the realm of web applications. The intention is to provide means for web applications to use AAA infrastructures for identity management purposes. This is done by adapting network-based technology to be used in the application layer of web environments. Furthermore, the identity management aspect is emphasized by the means to transport identity information in addition to the classical AAA tasks.

B. Comparison with other approaches

Table I shows a feature comparison of Diameter WebAuth with web-based identity management approaches. The result of that comparison is that Diameter WebAuth is able to offer most of the features that can be demanded from an identity management approach. Solely the support for single sign-on/sign-out can be enhanced, compared to solutions like OpenID and Liberty. However, OpenID and Liberty were designed with a special focus on single sign-on mechanisms. Compared to the Microsoft CardSpace technology, Diameter WebAuth does not rely on the support of the user client. This makes it easier to deploy and to maintain.

Because Diameter WebAuth also explicitly covers end user authentication it can be considered much more versatile than the OpenID and Liberty approach. Unlike those approaches, Diameter WebAuth can be used as an authentication backend. This makes it viable in scenarios where no identity management but local authentication is the primary task to accomplish. For example, in the case where a company-internal webserver should be enabled to authenticate users against an existing AAA infrastructure. The authentication capabilities of Diameter WebAuth also allow for it to be employed to complement identity management approaches that do not cover authentication.

A unique advantage of Diameter WebAuth is that, because it superimposes on the Diameter protocol, it seamlessly integrates in existing AAA infrastructures and that it supports accounting mechanisms. While the later might not yet be excessively interesting for most web applications, the utilization of existing AAA structures, especially existing Diameter servers, brings a number of advantages. It reduces expenses and maintenance effort, avoids data redundancies or discrepancies and allows to consolidate know-how. Especially for organizations that provide different access controlled services, this makes Diameter WebAuth a very attractive approach to expand existing network-based identity management to web applications.

V. CONCLUSION

In this paper, we presented a new approach to identity management for web applications based on existing network-based protocols. The proposed Diameter WebAuth is an AAA-based identity management framework that has been developed with the same requirements that are made to web-based solutions. It closes the gap between network authentication and application authentication by effectively bringing network-based access control concepts to the application layer. WebAuth it is based on the well established and mature Diameter protocol. It, therefore, benefits from the propagation of Diameter setups and the general experience with the protocol in terms of deployability implementability and maintainability. Diameter WebAuth offers authentication, authorization and billing facilities and is also able to handle identity attributes based on various schemas.

The proposed approach can be used to enable any Diameter system to be part of an identity management system. Especially network providers, which already operate Diameter setups can easily extend their systems to offer more services and consolidate their infrastructure respectively. In closed networks, Diameter WebAuth can also be established as a common authentication method to implement a central and secure authentication system that is not limited to the web-environment. Since not all identity management solutions include authentication protocols, even identity providers offering services using other approaches, like OpenID or the Liberty Alliance, can employ Diameter servers for their backend authentication using Diameter WebAuth. The work presented in this paper is also partially available as an internet draft [23].

Future work will include additional authentication schemes such as certificate-base authentication, the extension of single sign-on/sign-out capabilities and a comprehensive performance evaluation based on our prototype implementation. Furthermore, a secure and dedicated feedback channel between the identity provider and the user can enhance the overall security of Diameter WebAuth and is subject to further investigations.

ACKNOWLEDGMENT

The authors would like to thank Hannes Tschofenig from Nokia Siemens Networks for his vital inputs.

REFERENCES

- [1] "OpenID Authentication 2.0," Finalized OpenID Specification, Dec. 2007.
- [2] "The Liberty Alliance," Oct. 2007, (accessed 2008-08-15). [Online]. Available: <http://www.projectliberty.org/>
- [3] Microsoft Corporation, "Windows CardSpace," 2006, (accessed 2008-08-15). [Online]. Available: <http://cardspace.netfx3.com/>
- [4] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence, "Generic AAA Architecture," RFC 2903 (Experimental), Aug. 2000.
- [5] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Draft Standard), Jun. 2000.
- [6] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," RFC 3588 (Proposed Standard), Sep. 2003.
- [7] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)," RFC 4120 (Proposed Standard), Jul. 2005.
- [8] T. Scavo and S. Cantor, "Shibboleth Architecture - Technical Overview," Working Draft, Jun. 2005.
- [9] B. Laurie. (2007, Jan.) OpenID: Phishing Heaven. (accessed 2008-08-15). [Online]. Available: <http://www.links.org/?p=187>
- [10] M. Slot. Beginners' guide to OpenID phishing. (accessed 2008-08-15). [Online]. Available: <http://marcoslot.net/apps/openid/>
- [11] S. Brands. The Identity Corner - The problem(s) with OpenID. (accessed 2008-08-15). [Online]. Available: <http://www.idcorner.org/?p=161>
- [12] M. Hansen, "User-Controlled Identity Management the Future of Privacy," in *Identity in a Networked World*. FidiS consortium, Aug. 2006.
- [13] The MIT Kerberos Consortium. (2007) MIT Kerberos Consortium. [Online]. Available: <http://www.kerberos.org/index.html>
- [14] D. Kouril, "Kerberos Module for Apache," (accessed 2008-08-15). [Online]. Available: <http://modauthkerb.sourceforge.net/>
- [15] —, "Negotiateauth Project: HTTP Negotiate authentication for Mozilla-based browsers," (accessed 2008-08-15). [Online]. Available: <http://negotiateauth.mozdev.org/>
- [16] J. Garman, "Single Sign-on for Your Web Applications with Apache and Kerberos," Nov. 2003, (accessed 2008-08-15). [Online]. Available: <http://www.onlamp.com/pub/a/onlamp/2003/09/11/kerberos.html>
- [17] P. Calhoun, G. Zorn, D. Spence, and D. Mitton, "Diameter Network Access Server Application," RFC 4005 (Proposed Standard), Aug. 2005.
- [18] P. Calhoun, T. Johansson, C. Perkins, T. Hiller, and P. McCann, "Diameter Mobile IPv4 Application," RFC 4004 (Proposed Standard), Aug. 2005.
- [19] M. Garcia-Martin, M. Belinchon, M. Pallares-Lopez, C. Canales-Valenzuela, and K. Tammi, "Diameter Session Initiation Protocol (SIP) Application," RFC 4740 (Proposed Standard), Nov. 2006.
- [20] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication," RFC 2617 (Draft Standard), Jun. 1999.
- [21] H. Hakala, L. Mattila, J.-P. Koskinen, M. Stura, and J. Loughney, "Diameter Credit-Control Application," RFC 4006 (Proposed Standard), Aug. 2005.
- [22] J. Bufu and J. Hoyt, "OpenID Attribute Exchange 1.0," Dec. 2007. [Online]. Available: <http://cardspace.netfx3.com/>
- [23] N. Neumann and X. Fu, "Diameter Application for Authentication and Authorization in Web Applications," draft-neumann-dime-webauth-00 (work in progress), Feb. 2008.