Trade-off Analysis of Identity Management Systems with an Untrusted Identity Provider

Golnaz Elahi
Department of Computer Science
University of Toronto
Canada, M5S 1A4
Email: gelahi@cs.toronto.edu

Zeev Lieber
SlashID Corporation
Toronto, Canada
Email: zeev@slashid.com

Eric Yu
Faculty of Information Studies
University of Toronto
Canada, M5S 3G6
Email: eric.yu@utoronto.ca

***Abstract*—Internet users interact with multiple Web Service Providers (WSP), and therefore, must remember and manage multiple passwords. Users try to overcome the burden of password management by employing insecure solutions such as reusing the same password with several WSP. Identity management systems provide a solution for such problems. The common "assertion-based" Identity Management systems require a strong trust in the Identity Provider (IdP), which has the power to impersonate any of its users. However, such trust is unlikely to materialize in the global Internet setting. This paper uses a goal-oriented approach for analyzing trust trade-offs of Identity Management systems in the global Internet scenario. We analyze a new proposal for a global Identity Management system named SlashID. SlashID takes advantage of client-side cryptography to eliminate the required trust relationship between the IdP and the end users. We analyze and compare the impact of trust trade-offs of SlashID solution, using the suggested goal-oriented trade-offs analysis approach.**

I. Introduction

As more and more websites add personalized services, users need to manage more and more passwords and profiles. This results in less secure practices employed by the users, such as reusing a single password across different websites, or choosing weak passwords that are easier to remember. As the number of accounts increases, the users will need to keep their information up to date with more and more websites. Whenever a piece of personal information changes, the update has to be made in multiple places. Identity Management systems solve the "password fatigue" problem by providing a single set of credentials that can be used in multiple places. Identity Management systems solve multiple profiles problem by providing a single place to update personal information, which is then propagated to one or more *relying parties*.

Most Identity Management systems available today are based on an *assertion* issued by the Identity Provider (IdP). After authenticating the user, the assertion is generated and passed on to the relying party, which in turn verifies the assertion. This configuration requires the relying party to trust the IdP to issue correct assertions, and to safeguard user's passwords. In fact, the IdP in such a system has the complete power to impersonate any of its users, without the risk of being caught. This absolute power comes at a cost: the mere possibility of IdP's cheating gives a malicious user a way to deny a transaction and try to blame the IdP for it.

Although this setup works well inside a single domain, such as enterprise Identity Management systems, it may break when trying to apply it to a cross-domain authentication. This is due to the fact that the trust relationship required for the system to function may not exist between two arbitrary and unrelated organizations, and between end users and web-based IdP. Therefore, trust should not be dealt as users' and system owners' goal.

This paper suggests using a goal-oriented approach for analyzing trust trade-offs of Identity Management with an untrusted IdP. We analyze how lack of trust between the end users and the IdP, and between the Web Service Providers (WSP) and the IdP causes security and privacy trade-offs in the global Identity Management scenario. This paper analyzes a new proposal for Identity Management called SlashID for eliminating the required trust relationship between the IdP and end users. The SlashID solution employs client-side cryptography, which results in the IdP not having access to any sensitive information. Using the proposed goal-oriented approach, we analyze the impact of employing various approaches of Identity Management on trust, security, and privacy.

The remaining parts of the paper are structured as follows. In section II, we overview existing or related approaches to single password, single profile Identity Management, and analyze trust trade-offs that these impose. Section III describes the SlashID protocol as a new proposal to solve the trust trade-offs in Identity Management systems. In section IV, we describe the goal-oriented approach for modeling and analyzing security and privacy issues resulted from trust trade-offs. Section V employs a goal-oriented approach to analyze the security, privacy, and trust trade-offs that the Identity Management solution proposals impose. Section VI discusses the related work in this area, and finally, section VII draws a conclusion, and presents limitations and future work.

II. Existing Solutions to Identity Management with an Untrusted Identity Provider

There exist various approaches to identity and password management, which we briefly describe in this section. For each approach, we discuss its applicability to the Identity Management with an untrusted IdP in the global Internet scenario.

The Kerberos [10] protocol is used to authenticate users on local networks. A secret key is shared between each user and an authentication server. The Kerberos server knows secret keys of all participants, which enables it to impersonate any participant in the system. This implies that the Kerberos server is supposed to be trusted; therefore, Kerberos cannot be easily used in cross-domain authentication.

Another popular approach to Identity Management is to manage passwords completely on the client side, with no need for an IdP, which is mostly known as client-side password management. However, the major drawback of this solution is the lack of portability. The user has to manage the password file, and carry it around, as well as the software that uses it.

Hushmail [18] is a privacy oriented web-based email service. While not related to Identity Management, it is an Internet product that aims at removing the trust from the service provider. The cryptographic functionality is implemented on the browser side.

In order to eliminate client-side state, it is suggested to use a one-way hash function to derive a new password for each of the websites that require authentication. This is achieved by hashing the *Master Password* together with the website's URL to create a website-specific password. Examples of such systems are Janus [12] and PwdHash [14]. These systems come close to solving the problem of managing multiple passwords. However, they only provide password management, and not a full identity management solution including multiple profile management.

Generally, the solutions studied above do not provide one common solution for both the single password and single profile management issues. Currently, the main practical solutions for Identity Management are Isolated Identity Management and Assertion-based solutions which we briefly overview.

A. Isolated Identity Management

Isolated Identity Management (as referred to in [11]), also known as "Identity 1.0", is the existing password authentication scheme, implemented by most commercial websites. There is no IdP, and all the interaction is done between the end-user and the WSP. In this approach, the following trust relations exist between the WSP and the user. We based the list on the relationships described in [11]:

- T_SECUREPWD: The user trusts the WSP not to reveal user's password to any third party.
- T_PRIVACY: The user trusts the WSP not to reveal user's private information to any third party.
- T_AUTH: The user trusts the WSP not to provide services the user is entitled to, to anyone except after satisfactory authentication using the password.
- T_HANDLING: The WSP trusts the user not to reveal his credentials to any third party, i.e. to handle the password with care.
- T_AGENT: The WSP trusts the user to have a correctly functioning User Agent, i.e. Browser.

B. Assertion-Based Solutions

In an assertion-based Identity Management system, an IdP verifies identities of the users, and then issues *assertions* using SAML [16] or other technology. The WSP, the *relying party*, verifies the assertion instead of verifying user's credentials directly. Examples of assertion based solutions are most Federated Identity Management systems, as well as Identity 2.0 systems such as OpenID [17]. Assertion-based approaches introduce an additional trust requirements to the system, since the WSP relies on the assertion issued by the IdP. Due to the additional trust relationship, T_ASSERT, the WSP trusts the IdP to only issue "honest" assertions. The same T_ASSERT trust requirement is also found between the user and the IdP, because user's security now fully depends *both* on the IdP making a true assertion, and on the WSP acting only on true assertions. While this trust requirement is acceptable for an enterprise scenario, it may be less acceptable in a global Internet setting. In addition, T_SECUREPWD takes a stronger form, since the password that the user uses is now a key to many more doors, and the IdP or their employees may have much stronger motivation to use it.

III. SLASHID PROTOCOL DESCRIPTION

SlashID provides a cryptographic protocol for managing passwords and user's personal data such as one or more profiles. The main goal of the SlashID solution is to eliminate the required trust relationship between IdP, end users, and WSP. The general idea behind SlashID Identity Management solution is to send the users data, including passwords and profile, as encrypted values to SlashID server from the users browser. The user receives back the same encrypted values from SlashID and sends the decrypted data to the target WSP.

Similar to PwdHash [14] and Janus [12], SlashID protocol employs a separate shared secret between the user and each WSP the user is registered with. However, SlashID generates a completely random secret and encrypts it with user's password which enables changing the secrets. To make the password change easier, SlashID uses a permanent Master Secret for each user, and encrypt all the Shared Keys using this Master Secret. The Master Secret, in turn, will be encrypted with User's password.

In order to inhibit cross-site tracking, SlashID represents the same user with different identitifers to each website. Therefore, SlashID computes a handle by hashing the username with the URL of the website. As a result, if Alice is logging into Bob's online store, she computes her handle as $H_{AB} = Hash(U_A||URL_B)$, where U_A is her username. SlashID system consists basic protocols such as user creation, registration, log in, and profile update, which some of them are briefly specified in this section. All communications are supposed to be performed over a private and secure channel by using HTTPS (HTTP over TLS) requests.

A. User Creation Protocol

1) $A \rightarrow I : H_{AI}, \{M\}_{P_A}, \{S_{AI}\}_M, S_{AI}, \{R_A\}_M$
The user Alice, A arrives at the IdP's (I) website

to create an account. Alice generates a password P_A, Master Secret M and Shared Secret S_{AI} to use with the IdP. She also creates and encrypts her profile (R_A) which contains personal data that she may be disclosing to different websites, but not to the IdP. Alice computes her Handle to be used with IdP (H_{AI}), and identifies herself to the IdP via her handle. She sends the encrypted values to the IdP.

2) $I \rightarrow A : OK$
The IdP stores the values in the database, using H_{AI} as Alice's userid and signals transaction success (OK) back to Alice.

B. Registration Protocol

1) $A \rightarrow I : H_{AI}$
Alice identifies herself to the IdP to prevent unauthorized registrations, and requests a login.
2) $I \rightarrow A : \{M\}_{P_A}, \{S_{AI}\}_M$
IdP sends back encrypted secret values.
3) $A \rightarrow I : S_{AI}, URL_B, \{S_{AB}\}_M$
Alice decrypts the shared secret and sends it back to the IdP to verify her identity. She generates a new Shared Secret to be used between her and Bob, and sends an encrypted version of it to the IdP, which stores it. She also sends Bob's URL to request the registration.
4) $I \rightarrow A : T_{AB}$
The IdP generates a registration ticket to prove that the registration request is coming from the IdP. The ticket is similar to a token of trust in the IdP. This mechanism protects system integrity, rather than user's security.
5) $A \rightarrow B : H_{AB}, S_{AB}, T_{AB}, R_A$
Alice computes her handle to be used with Bob: H_{AB}. Alice decrypts her profile, and sends it along with the Handle, Shared Secret and Registration Ticket to Bob.
6) $B \rightarrow I : T_{AB}$
Bob sends the ticket back to the IdP to verify it.
7) $I \rightarrow B : OK$
IdP responds with an OK status, and Bob saves the shared secret value and Alice's profile in the database. Bob stores the secret and the profile in his database, using Alice's Handle as her userid.
8) $B \rightarrow A : OK$
Bob signals success to Alice and the protocol is over.

C. Login Protocol

1) $A \rightarrow I : H_{AI}$
Alice computes her handle for the IdP and sends it to the IdP.
2) $I \rightarrow A : \{M\}_{P_A}, \{S_{AI}\}_M$
The IdP returns encrypted Master Secret and encrypted Shared Secret for the IdP.
3) $A \rightarrow I : S_{AI}, H_{AB}$
Alice decrypts her Master Secret using her Password, then her Shared Secret using her Master Secret. She sends back a clear text Shared Secret to prove her identity; She also computes her handle for Bob (H_{AB}) and sends it along, to indicate that she wishes to authenticate to Bob.
4) $I \rightarrow A : \{S_{AB}\}_M$
The IdP retrieves the stored value of Alice and Bob's encrypted Shared Secret, and sends it to Alice.
5) $A \rightarrow B : H_{AB}, S_{AB}$
Alice decrypts her and Bob's Shared Secret and sends it to Bob along with her Handle.
6) $B \rightarrow A : OK$
Bob verifies that the shared secret matches his copy and signals OK to Alice. The login protocol logs in the user simultaneously into the IdP and WSP website. This will allow single sign-on functionality next time the user logs in with a different website.

IV. The Goal-Oriented Approach to Trust Trade-off Analysis

Security requirements can be difficult to identify and satisfy; however, the more difficult issue is to analyze the extent to which a system can simultaneously satisfy multiple interacting, and frequently conflicting requirements, including security and privacy. From a requirements perspective, security is often thought to be in conflict with privacy - to attain better security one would have to give up some privacy and vice versa [13].

However, security issues are not limited to protection mechanisms to maintain the integrity, confidentiality, and availability. Recent approaches to security emphasize dealing with security, not only as a system problem, but also as a social and organization issue [1], [2]. Another recent shift in analyzing security requirements is toward the trust assumptions in secure software engineering [3], [2], [4]. Parties in a relationship trust or mistrust each other to deliver the service they require, and to not abuse the permissions which are given to them [2]. Viega et al. argue that basis of trust relationships and formation can dramatically affect the underlying security of any system [3];

In cross-domain, cross-organization, and global Internet environments, several service providers and receivers build a chain of dependencies. However, parties of dependency chains may not trust each other. We analyze the Identity Management problem as an example of situation that the lack of trust between end users and IdP imposes trade-offs on security and privacy. The viability of an Identity Management solution hinges on a proper understanding of the trust relationships among the various parties. Deciding on the security and privacy trade-offs for such cases need analyzing consequences of relying on untrusted IdP that provide Identity Management service. By making trust trade-offs, the user chooses a party to trust over other possible parties, and trusting each alternative party affects security, privacy, and other quality goals.

To analyze the impact of trust or mistrust, we employ the i* framework [19] for modeling the parties in the trust relationships, their goals, alternative ways to achieve their goals, dependencies among parties, and consequences of decisions they make. The i* framework provides a notation for modeling actors, goals, and intentional dependencies and competitions among actors. Actors achieve goals on their own, or depend

on each other for goals to be achieved, tasks to be performed, and resources to be furnished. Quality goals, which do not have clear-cut criteria for satisfaction degree, are modeled as softgoals. Means-ends relation between goals and tasks is used to model alternative ways to achieve a goal. Contribution links in the i* modeling notation provide the means to express the interactions and impacts of goals or decisions on other goals.

For example, the goal model in Fig. 1 gives the intentions of end users and WSP in using single password management through an IdP, alternative solutions that users can employ to manage multiple passwords via means-end links, and their consequences. In this model, the *End user* has four alternative solutions to satisfy *Manage multiple passwords* goal. One of these alternatives is *Single password through an IdP*, which requires the *End user* to depend on *IdP* actor. Later, we discuss how and why this dependency between the *End user* and the *IdP* impose trust requirements. The *IdP* actor has two main alternative ways to provide single password service: *Assertion-based IDM (e.g. OpenID)* and *Client-side Cryptography (SlashID)*.

The model also specifies user's softgoals such as *Service Portability*, *Privacy*, and *Security*. For example, the *End user* can *Write down all passwords* to solve the password fatigue problem, but it has negative contribution (dashed links) on *Password security* softgoal. The *End user* has this alternative option to use *Client-side password management* such as Isolated Identity Management. However, it has negative impact on *Service portability*. In this way, trade-offs that each solution imposes are modeled. One can avoid establishing trust relationships with an IdP by using the alternative ways (like Client-side password management) to solve the password fatigue problem by trading other quality goals (like portability).

To express the consequences of trusting a party, we replace the parties the users need to trust with a malicious actor. The malicious actors have the same capabilities of the trusted actors, but abuse these capabilities and permissions to achieve their malicious goals. We employ the security extensions to the i* [20] to model and analyze the malicious behavior of an actor and consequences of their behavior. This approach enables the analyst to model the malicious goal model of potential attackers to express threats that the security protocol is designed to prevent, and analyze whether the malicious goals are denied and protocol goals are satisfied. Illustrative examples of analyzing and comparing trust trade-offs of the SlashID and assertion-based Identity Management protocols, using the proposed approach, are given in the next section.

V. Trust Trade-offs Analysis of Untrusted Identity Providers

In this section, we apply the goal-oriented approach to develop models of the Identity Management protocols' behavior and analyze trade-offs that using the untrusted IdP imposes. We analyze trade-offs of trusting the SlashID IdP and an assertion-based IdP on stakeholders' goals.

As discussed earlier, the trust relationship between the IdP, WSP, end users threatens security and privacy softgoals of the

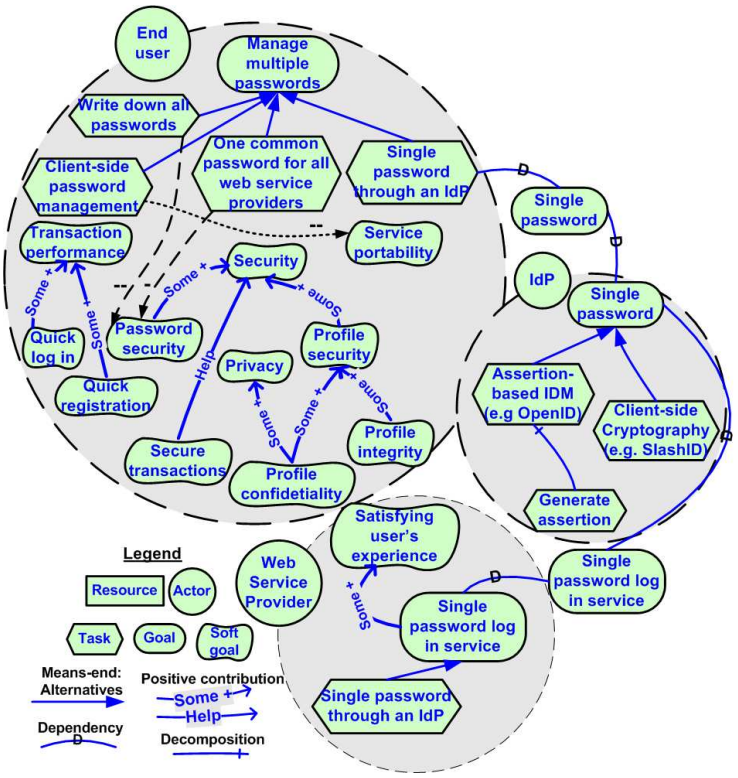

Fig. 1. The i* goal model of end users and WSP intentions for single password management through an IdP.

end uses and WSP, since the IdP has an absolute power over the users identity. Fig. 2 specifies possible security threats that a malicious assertion-based IdP may pose. A malicious IdP can abuse the users' trust in providing the IdP with passwords to *Steal users password* and *Impersonate the user*. Developing the goal model both for the malicious and non-malicious parties provides a basis to evaluate whether the the attacks are sucessful or not. The qualitative goal model evaluation techniques propagate satifaction/denial labels through the models. For example, the checkmark symbols on the model elements express the satisfaction of goals and tasks of malicious IdP. Since the WSP authenticates the users by the assertions that the IdP generates, the *Malicious assertion-based IdP* can *Generate fake assertion* to *Impersonate the user*.

On the other hand, SlashID removes the need that end users trust the IdP to have their plain-text passwords or generate assertions, and the IdP stores and transmits the passwords as encrypted values. The cryptographic calculations of SlashID protocol are performed at the client-side browser, which is currently implemented by JavaScript, and the users need to download the JavaScript to their local machine and run. This requires a T_AGENT trust relation between a WSP and IdP. By providing JavaScript code, the IdP is now responsible for part of user's agent. Therefore, WSP has to believe the IdP will not inject any Trojan code into the JavaScript to get the user's password. This situation presents a trust trade-off between assertion-based solutions and client-side cryptography. Users

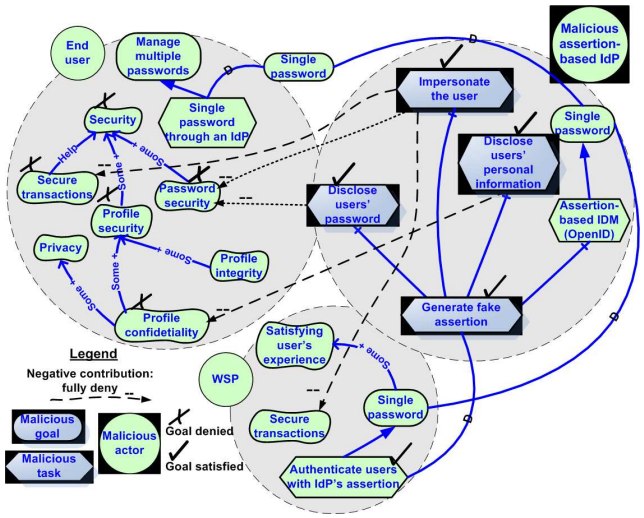

Fig. 2. Security threats that an assertion-based IdP poses and compromises the end users' goals who trust the IdP.

need to decide on trusting the client-side cryptography-based IdP not to inject Trojan JavaScript, or trusting the assertion-based IdP to issue "honest" assertions. To analyze the trade-off between these two options we prove two statements:

1) From the trust perspective, T_AGENT between IdP and WSP is *not worse* than T_ASSERT; and
2) From the technology perspective, T_AGENT is overwhelmingly preferable over T_ASSERT.

The first statement is true, simply because with T_ASSERT the IdP already has user's password or equivalent, while with T_AGENT they need to go through an extra step of stealing it. The second statement arises from the fact that T_ASSERT is embedded in the *protocol itself*, while T_AGENT is embedded in a particular *implementation* of a protocol. Currently SlashID protocol is implemented by JavaScript; therefore, to remove T_AGENT, it is enough to implement the SlashID protocol as part of the browser. As long as the specifications of the protocol are open, the Open Source community can implement a secure browser core or plug-in which resolves the trust trade-offs. On the other hand, T_ASSERT cannot be removed from an assertion-based system without completely changing the architecture of that system.

Fig. 3 provides a visual representation of the discussed arguments about the impact of T_AGENT trust between IdP and WSP. The malicious SlashID IdP which works based on JavaScript pose the threats to *Steal users password* by *Trojan Horse through JavaScript* and *Offline Dictionary attack*. Other alternative implementations of SlashID by *Browser plug-in* or *Core browser modification* prevent the threats of Trojans. However, SlashID protocol is still vulnerable to offline dictionary attacks.

A malicious SlashID IdP can run *Offline Dictionary Attack* to *Steal users' passwords*. However, this attack is not successful for all attempts; therefore, the *Password security* softgoal of the *end user* is indicated as partially denied, as shown in Fig. 3.

A malicious assertion-based IdP does not need to perform the extra effort for offline dictionary attack. Hence, the result of malicious assertion-based IdP behavior is fully denial of the *security* softgoal, shown in Fig. 2. To completely prevent an offline dictionary attack by the IdP, one can employ several unrelated servers, such as solutions described in [21], [22].

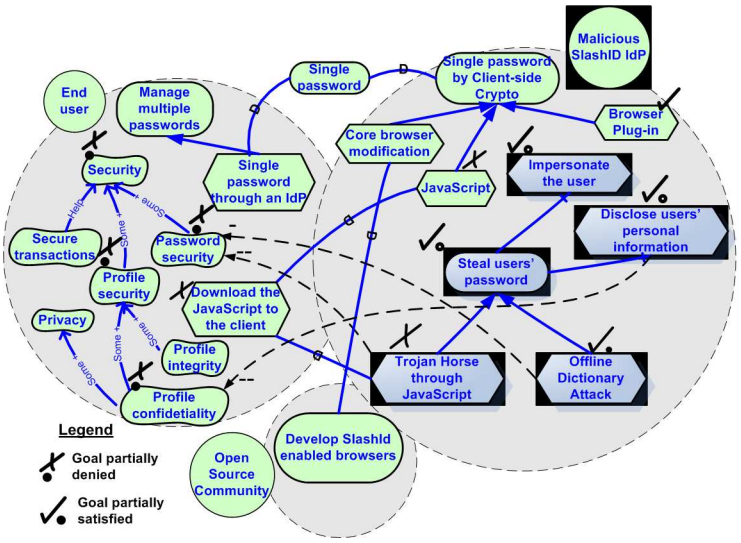

Fig. 3. Security threat that a client-side cryptography-based IdP can pose. The threat depends on the end user downloads and run the IdP's malicious JavaScript

VI. Related Work

We argued that a systematic approach for analyzing and incorporating the impact of (mis)trust on security and privacy requirements of stakeholders and parties is necessary. In this regard, Giorgini et al. [2] suggest modeling security requirements based on the concepts of ownership, permission, and delegation within the normal functional requirements model and actor dependencies. This approach employs an analysis procedure on formally specified ownership, delegation, and trust models developed by Secure Tropos modeling notation. The analysis evaluates issues such as if actors have assigned permission and obligation to trusted actors; and if actors have assigned duties to actors that have capabilities and permission to achieve the goals. The proposed approach is useful to identify the dependencies, permissions, and delegations that cause security problems due to the untrustworthiness of dependee. However, the analysis stops at this stage, and it does not provide means to express the consequences of trusting parties which do not have capabilities and permission to achieve the goals.

Haley et al. [4] analyze the effect of trust assumptions on elaborating security requirements. They propose employing and combining problem frames [5], threats description, and trust assumptions for deriving, elaborating, and analyzing security requirements. In this approach, trust assumptions are added to the problem frames models of the system, which helps documenting the decisions about security requirements,

and defining and limiting the scope of the analysis. However, this approach does not consider analyzing the consequences of violation of trust assumptions on requirements.

Yu and Liu [6] treat trustworthiness as a quality goal to be satisfied from the viewpoint of each stakeholder depending on others. They employ the i* framework to model the dependencies and trust requirements among stakeholders. The trustworthiness softgoal is refined in both depender and dependees actors boundaries, and impact of other goals and decisions of stakeholders on trustworthiness is modeled and evaluated. In a similar approach in [7], trust is treated as a softgoal for analyzing technology strategies. This approach suggests analyzing the impact of malicious parties on goal model evaluation from the point of view of opponents and proponents of the trusted technology. These approaches provide an explicit way to model and analyze the trust issue as one of stakeholders goals. However, in the global Internet scenario, end users are reluctant to trust other parties; therefore, service providers, for instance an IdP, may need to remove or mitigate the trust requirements, and trust and establishing a trust relationship are not the softgoals of the interacting parties.

In a similar contribution to our work, the i* framework and elaborated version of strategic actors of i* framework, called Goal-oriented Requirements Language (GRL), are used in [8], [9] for modeling and analyzing Identity Management issues. The approach in [8] proposes a generic identity management meta-model, with which the requirements and identity management architecture designs can be analyzed. The emphasis of this approach is on intentions, capabilities, and dependencies of parties, which facilitates reasoning about trust distribution and dependency relationships. However, it does not provide means to reason about alternative parties to trust, and trade-offs that each trust alternative imposes.

VII. Conclusion, Limitations, and Future Work

This paper discusses that existing Identity Management solutions require strong trust relationship between users and IdP that are not found in today's Internet scenario. The required trust is likely to have negative effect on global adoption of any of these solutions as the Global Identity Management protocol for the Internet. This brings the need for suitable models and analysis of trust trade-offs that different solutions may impose.

In this paper, security extensions to the i* framework are used to model and analyze the consequences of relying on alternative untrusted parties. We analyzed the trust trade-offs by replacing the trusted parties with malicious actors. We described and analyzed an Identity Management protocol named SlashID which alleviates these trust issues partially if implemented in JavaScript. We applied the modeling and analysis technique to the SlashID solution, and compared it with assertion-based solutions. The resulted models can be used to compare the consequences of malicious capabilities of different IdP.

Developing the goal model for the malicious actors helps understanding and reasoning about consequences of trusting untrusted parties and security and privacy threats that each malicious party may pose. This approach facilitate comparing untrusted parties and trade-offs that trusting each one imposes. Goal model evaluation techniques help analyzing if the top goals of the Identity Management protocol are satisfied when the goal model of the protocol scales.

However, the analysis and arguments of this paper are limited to trade-offs between only two example solutions: the assertion-based solution and SlashID. The analysis is limited to trust relation between the end users and the IdP, and the IdP and the WSP, while one may raise the argument that WSP parties are not trustworthy either from the point of view of the end users. In future work, we need to analyze each party, IdP, WSP, and end user as an attacker to Identity Management system as part of the trust trade-off analysis.

References

[1] L. Liu, E. Yu, J. Mylopoulos, Security and Privacy Requirements Analysis within a Social Setting. In IEEE Joint Int. Conf. on Requirements Engineering, 2003, 151-161.

[2] P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone, Modeling Security Requirements Through Ownership, Permission and Delegation, In proc. of RE, 2005, 167-176.

[3] J. Viega, T. Kohno, B. Potter, Trust (and Mistrust) in Secure Applications, Communications of the ACM 44(2), 2001, 31-36.

[4] C. B. Haley, R. C. Laney, J. D. Moffett, B. Nuseibeh, Using Trust Assumptions in Security Requirements Engineering, Second Int. iTrust Workshop On Trust Management In Dynamic Open Systems, 2003.

[5] M. Jackson, Problem Frames, Addison Wesley, 2001.

[6] E. Yu, L. Liu, Modelling Trust for System Design Using the i* Strategic Actors Framework, In: Trust in Cyber-Societies - Integrating the Human and Artificial Perspectives, LNAI-2246. Springer Verlag, 2001, 175-194.

[7] J. Horkoff, E. Yu, L. Liu, Analyzing Trust in Technology Strategies, In Proc. of Int. Conf. on Privacy, Security, and Trust (PST'06) 2006.

[8] L. Liu, E. Yu, Intentional Modeling to Support Identity Management, In 23rd Int. Conf. on Conceptual Modeling (ER 2004), LNCS 3288 Springer, 2004, 555-566.

[9] L. Liu, E. Yu, Modeling Identity Management Architecture within a Social Setting In Proc. of the 8th Asia Pacific Web Conference (APWeb 2006), LNCS 3841, 2006, 917-922.

[10] J. Kohl, C. Neuman, The Kerberos Network Authentication Service V, 1993.

[11] A. J. Sang, J. Fabre, B. Hay, J. Dalziel and S. Pope, Trust Requirements in Identity Management, Australasian Information Security Workshop, 2005.

[12] E. Gabber, P. Gibbons, Y. Matias, and A. Mayer. How to Make Personalized Web Browsing Simple, Secure, and Anonymous, Proc. of Financial Cryptography, Springer-Verlag, LNCS 1318, 1997.

[13] B. Schneier, Beyond Fear, Springer, 1st ed., New York, 2003.

[14] B. Ross, C. Jackson, N. Miyake, D. Boneh and J.Mitchell Stronger Password Authentication Using Browser Extensions, Proceedings of the 14th Usenix Security Symposium, 2005.

[15] C. Caleiro, L. Vigan, D. Basin, Deconstructing Alice and Bob, in: Proc. ARSPA'05, ENTCS, Vol. 135(1), 2005, 3-22.

[16] Security Assertion Markup Language 2.0, OASIS standard, 15 March 2005. Available from http://docs.oasis-open.org/security/saml/v2.0/

[17] OpenID: an actually distributed identity system *http://openid.net*

[18] HushMail, Free Email with Privacy. *http://www.hushmail.com*.

[19] E. Yu, Modeling Strategic Relationships for Process Reengineering, PhD thesis, Department of Computer Science, University of Toronto, Canada, 1995.

[20] G. Elahi, E. Yu, A goal oriented approach for modeling and analyzing security trade-offs, In Proceeding of 26th International Conference of Conceptual Modeling, 2007, 375-390.

[21] W. Ford, B. S. Kaliski Jr., Server-Assisted Generation of a Strong Secret from a Password, In Proc. of the IEEE 9th Int. Workshop on Enabling Technologies (WET-ICE), IEEE Press, 2000.

[22] J. Brainard, A. Juels, B. Kaliski, M. Szydlo, A New Two-Server Approach for Authentication with Short Secrets, In Proc. of the 12th USENIX Security Symposium, 2003.