Human-Centric and Privacy-Protecting Trust Establishment over the Internet

Holger Kinkelin, Heiko Niedermayer, Simon Mittelberger and Georg Carle

Technische Universität München
Garching bei München, Germany
Email: lastname@net.in.tum.de

***Abstract*—Security needs to be human-centric if the human context and intent have to be known. Security mechanisms should be hidden whenever automation has all information it needs. In this paper, we present a human-centric, assisted trust establishment mechanism over the Internet. This mechanism expands our previous work that required a physical meeting to exchange identities (public keys) and to establish mutual trust. The goal is identification and authentication, which can then be used for authorization. The mechanism was designed to protect privacy while it utilizes existing trust relationships within small social groups to gain evidence of the trustworthiness of a claimed identity. Our technology is in particular suited for interconnecting users and their home networks as well as for smart environments. Competing solutions with respect to trust modeling are the X.509 standard and the PGP (Pretty Good Privacy) web of trust, with respect to applications cloud computing and social networks like Facebook.**

Keywords–trust establishment, authentication, access control

I. INTRODUCTION

Within the last ten years, home computing has gained a lot of momentum due to various reasons. The first important factor is the availability of affordable computer hardware and fast residential Internet connections. Today, energy efficient computing devices suitable for most home computing tasks, such as the Raspberry Pi [1], can be purchased quite cheaply.

The second important factor for home computing is home automation. Homes are becoming increasingly equipped with networked sensors and actuators able to sense or control lighting and temperature, for instance. Various open-source software projects, e.g., openHAB (open Home Automation Bus) [2], emerged that help users to orchestrate their smart home in order to increase comfort or decrease their home's energy consumption.

Lastly, a growing distrust in Internet Cloud services, such as online file storage, mails, and chats, fueled home computing after large-scale monitoring efforts by international intelligence became publicly known in 2013 [3][4]. Thus, local and human-centric solutions have to be found that can replace distant and centralized Internet Cloud services.

On application level, open-source software projects that recreate the functionality of Internet Cloud services became popular. Examples include OwnCloud [5], Seafile [6] and others. This allows building powerful networked service infrastructures within the privacy of a home network. Yet, services are rarely useful if they cannot be securely shared and used beyond the boundaries of isolated home networks.

An important ingredient for that is identification and authentication. The predominant solution today is the usage of user names and passwords, which are hard to handle and insecure due to massive reuse. There are better security mechanisms like asymmetric cryptography. Our work is focused on human-centric concepts for enabling the use of such strong cryptography on the user-side. The information needed from the user are names, public keys, and what the user thinks of them. Given software that assists the user and mechanisms that are human-centric and follow human ways of acting with each other, the user is a better source for such information than a central authority.

The central contribution in this paper is an approach and a protocol for human-centric secure and privacy-protecting trust establishment over an insecure network, i.e., the Internet. Details on the mechanism are described in Section V.

In the following Section II, we introduce our previous work on the subject. In Section III, we give further information and discuss why established trust models are not suitable human-centric solutions. Section IV presents requirements and Section VI an evaluation. Related work is presented in Section VII. An outlook on current work and a conclusion is given in Section VIII.

II. PREVIOUS WORK: ASSISTED KEY MANAGEMENT

In previous work, we designed a human-centric security concept [7]. In the center of our concept we have so-called *Domains*, which are the "digital homes" of users. As a Domain we understand any small network of devices and services. A typical example of a Domain would be a home network or the devices and services owned by a single person. Users can utilize their equipment to establish relationships with other Domains first. Later, services can be shared between such partner Domains.

Technically, we built client/server components for the sketched assisted key management. Our components automate difficult to perform tasks, hide complexity, and take care of security.

Our *Registration Server* acts as a front-end for a Domain-local X.509 certificate authority, called *Domain CA*. With the help of a tool installed on a device called *Registration Client*, a new asymmetric key can be generated and its public part be certified by the local Domain CA in a secure way. The registration process is controlled (permitted/denied) by the Domain owner using an administrative tool. Further parts of our work [8] extended the resulting basic access control scheme with fine-grained authorization based on XACML (eXtensible Access Control Markup Language) [9].

When services need to be securely shared between two Domains, a problem arises, as a local service is unable to

 ISBN: 978-1-61208-369-8

authenticate the certificate presented by the "foreign" device. Hence, service access will be refused. In order to solve this shortcoming, the foreign Domain's CA certificate needs to be *trusted* in the local Domain.

To setup a *trust relationship* between two Domains, we developed the *personal trust exchange* protocol in our previous work. The protocol is implemented by a tool that can be executed on a mobile device, such as a laptop or smart phone. When representatives of two different Domains meet, they can use this tool to securely and easily exchange their Domain CA certificates via near-field wireless technologies. After the exchange, the certificate is tagged with a human readable identity (hID) of its owner (e.g., full name) and the social peer group the owner belongs to (e.g., friends, colleagues, etc.). Based on this membership, basic access rights of this Domain can later be derived automatically, e.g., all friends might obtain access to the Domain's photo sharing service. Hence, our notion of "trust" goes beyond the traditional understanding of the term and comprises identification, authentication, and authorization. Later, certificate and meta-data of the foreign Domain are imported into the own Domain and provided to all local services.

The outlined trust establishment method is secure and meaningful as certificates are exchanged in person, which allows the representatives to personally identify their exchange partner. Furthermore, the protocol guarantees that attackers are unable to interfere and, for instance, slip in a bogus certificate. This again could finally lead to an undesired trust relationship, which might be exploited. One major problem with the outlined trust establishment solution is that it can only be executed when representatives of two Domains are able to meet in person. Obviously, this is not always possible, e.g., when the Domain owners live far apart.

III. Analysis and Background

A. Applicability of Existing Personal Key Infrastructures

The standard **X.509** [10], also often referred to as the web PKI (Public Key Infrastructure), is today the most often used method to map an identifier to a public key. As identifiers, for instance, mail addresses or DNS (Domain Name System) names can be used. The common approach is that a trusted third party, called certificate authority (CA), verifies a public key holder's identity and then issues a signed (identity) certificate. Once a software component receives a certificate it can easily verify whether the certificate is valid, whether the certificate contains the expected identity, and finally whether the entity that presented the certificate is the legitimate key holder. Typically, CAs are payed services offered by companies like Verisign, Comodo, or Thawte.

In the scenario of home networking, or other small personal Domains, a public CA, for instance operated by the home's Internet provider, might be used to issue a certificate for the local Domain CA. This again would allow other Domains that trust this public CA to verify the identity of a Home CA. However, there are various issues with public CAs in the scenario of small private Domains: 1) no agreed-on standard for identifiers of small Domains, such as home or personal networks, exists yet. Certificates of Domain CAs might be simply associated with their owner's names. But natural names are often ambiguous and therefore no suitable identifier. The just proposed identifier might also be extended to a format like `country/city/street/name`, which would rule out ambiguity in most cases. But still this is insufficient as being able to authenticate a certificate issued by a foreign Domain does not yet answer the question which ones of all authentifiable foreign Domains are allowed to access a service offered in a Domain. 2) there are general issues with the trustworthiness of X.509 certificates as the verification of identities is done by a third party. Examples exist that show that CAs issued valid certificates to entities that are not the legitimate owner of an identity. In such cases the CA did not pay enough attention or was compromised by some adversary.

PGP/GPG (Pretty Good Privacy/GNU Privacy Guard) [11] is a human-centric alternative to X.509, which is mostly used for mail communication. Participants of PGP/GPG issue signed data structures that assert that a public key represents the identity of a particular person. This is done by mapping a name and mail address to this public key. To increase PGP/GPG's convenience, these data structures are typically published alongside the public keys on so-called *key servers*. Persons that require the public key that represents a specific identity or want to verify the authenticity of a public key they already possess, can query a key server and receive public key and signed data structures that vouch for this keys authenticity.

The PGP/GPG web of trust resolves some issues of the X.509 PKI as it is human-centric and obviously suits the demands of human-centric computing well. But it creates its own new problems. One of the most important problems of PGP/GPG is that social relationships are publicly revealed when key servers are used. Furthermore, the mere ability to authenticate public keys is, as explained in the context of X.509, still insufficient in our scenario.

B. Applicability of the Human-centric Trust Exchange

As outlined in the introduction of this paper, our approach is based on the idea of independent Domain-local small CAs and the establishment of trust relationships between Domains based on human interaction when needed. As explained before, a trust relationship is established by exchanging the certificates of two Domain CAs and marking the received certificate as belonging to a "friendly" Domain. Alongside with importing a certificate, a basic set of access rights to local services can be granted automatically. Later, these default rights might be fine-tuned by the Domain owner [8]. Furthermore, a human-readable, locally meaningful identifier and other meta-information (e.g., social community) can be assigned to the received certificate during the trust exchange.

As globally valid cryptographic identifiers for a Domain or an entity (devices or services) that belong to a Domain, we proposed the usage of *cryptographic identifiers (cID)* derived from public keys. A Domain's identity is the hash of the Domain CA's public key (DomainID = H(pubKey$_{Domain}$). The identity of an entity belonging to this Domain is the concatenation of the Domain ID and the hash of the entities public key (EntityID = H(pubKey$_{Domain}$).H(pubKey$_{Entity}$)). The resulting identifier is hierarchic and reflects the belonging of an entity to a specific Domain.

This approach solves trust issues and possible ambiguities of identifiers as received certificates are locally highly meaningful and trustworthy.

 ISBN: 978-1-61208-369-8

IV. REQUIREMENTS

The requirements on a trust exchange mechanism that can be performed over the Internet can be split in two groups, namely security and user-friendliness.

A. Security:

Establishing trust to the outside of a Domain is a critical process. When the security of the exchange cannot be guaranteed, unintended trust relationships might be established. In the worst case the falsely created trust relationship can be abused to access services offered within a Domain.

Requirement R1: Secure Identification/Authentication: The aim of a trust exchange is to receive the certificate of a specific friendly Domain. This aim can only be fulfilled if it can be ensured that the received certificate belongs to this Domain.

Requirement R2: Rating of Certificates: The trust exchange must measure the level of trust into a certificate received from a friendly Domain to expresses the strength of the established trust relationship. Furthermore, this allows reducing the chance to accept untrustworthy Domains over, for the purpose of a trust exchange, trustworthy Domains.

Requirement R3: Security: All precautions need to be taken that a third party is unable to interfere with the trust exchange process, e.g., to trick one or both exchange partners to trust a certificate owned by the third party.

Requirement R4: Human-Centrism: The trust exchange must be human-centric, i.e., may not depend on external, central services.

Requirement R5: Privacy-Protection: Leakage of private information must be minimized, i.e., no third party should be able to easily enumerate trust relationships of a given Domain.

B. User-Friendliness:

Performing a trust exchange between Domains must be easy to understand and user-friendly. Therefore, following requirements must be fulfilled:

Requirement R6: High Degree of Automation: The trust exchange must hide all difficult to understand technical details and run mostly automatically.

Requirement R7: Owner Consent: No trust relationship may be established without a Domain owner's permission.

Requirement R8: Low Interference: The amount of interactions between the trust exchange mechanism and the human user involved in the process must be kept at a minimum to avoid repeated disturbance and prevent annoyances.

Requirement R9: Responsiveness: Long waiting times until the trust exchange has finished need to be avoided. As it cannot be expected that the process is computationally expensive, the responsiveness mostly depends on the protocol design and response times of involved human users.

V. TRUST EXCHANGE

A. Approach and Architecture

Our approach for a secure, privacy-protecting and reliable mechanism to exchange trust between Domains over the Internet is based on the idea to propagate already existing trust relationships between Domains that belong to a social community in a human-controlled manner. The exchange process is supported by a publicly reachable *trust exchange service* running within each Domain. A trust exchange service instance processes incoming trust exchange protocol messages, interacts with the Domain owner over a graphical user interface (GUI) and manages a database of known friendly Domains. The information contained in the Domain database includes a Domain's cryptographic (cID) and human-readable identifier (hID), the Domain's Trust and Identification Level (explanation see below), information to which social community a Domain belongs to (e.g., friends, colleagues, sports club, etc.) and finally the Domain's certificate.

The *Identification Level (IL)* is a property of a certificate. It is a numerical value in the range from 0 to 10 that measures the confidence a Domain A can have that a given Domain certificate belongs to another Domain B. The IL is either assigned by the Domain owner herself after performing the personal trust exchange or computed during the Internet trust exchange using our trust metric we introduce later in Section V-C.

The *Reputation Level (RL)* is a property of the human Domain owner. It is a numerical value in the range from 0 to 10 that expresses the expectation of Domain owner A how much care a Domain owner B will take when she is performing a personal trust exchange with a Domain owner C. Hence, the RL expresses A's assessment how reliable B will check the Identity of C and assign an appropriate IL to C's certificate.

A Domain's exchange service needs to be addressable and reachable from the public Internet. We assume that Domains participate in a Peer-to-Peer-style network that constitutes a DHT (Distributed Hash Table) storing key/value pairs. The DHT enables a Domain to store its current public IP under their cID in the DHT (`put(cID, IP)`). Other Domains use the DHT to query the current IP address of a known friendly Domain by performing an DHT query (`IP = get(cID)`).

Lastly, we introduce names for the different roles in the Internet trust exchange, see Fig. 1.

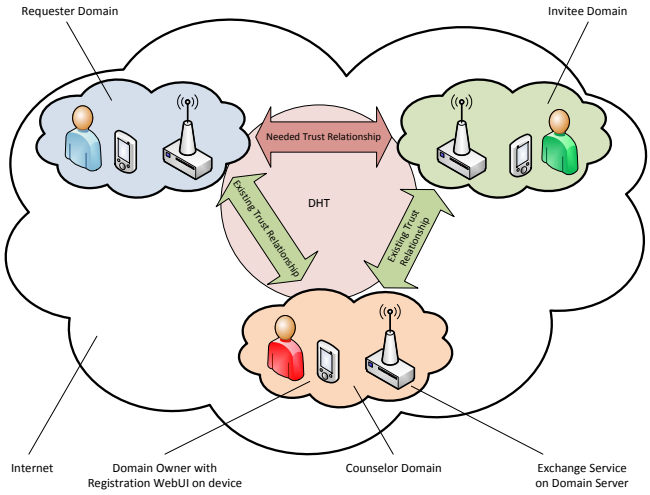

Figure 1. Scenario of the Internet trust exchange

The Domain that wants to establish a new trust relationship between herself and another Domain is called *Requester (R)*. The friend invited to join the new trust relationship is called

 ISBN: 978-1-61208-369-8

Invitee (I). Finally, common friends that assist Requester and Invitee during the Internet trust exchange are referred to as *Counselors (C)*.

B. Protocol Details

Instead of presenting message sequence diagrams that depict every protocol detail, we explain the crucial steps of the Internet trust exchange using the simple example shown in Fig. 2, part 1. In the example, R wants to establish a trust relationship to her friend I. A trust relationship already exists between R and Counselor C and between C and I. Please note that in real life more than just one Counselor might exist.

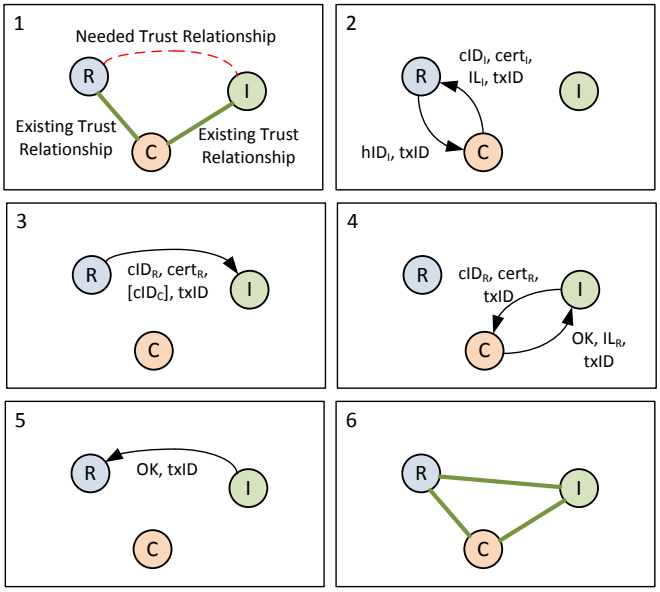

Figure 2. Example and simplified trust exchange message flow

The protocol is started by the Domain owner R by specifying to which Domain I she wants to establish a trust relationship to. This is done by providing I's human readable identifier (hID_I) to the own trust exchange service. Furthermore, the Domain owner must specify which Domains should act as Counselors either by specifying Domains explicitly or implicitly by naming suitable social communities.

The careful selection of Counselors is needed due to multiple reasons. 1) a *Counsel Request* sent to a Counselor will reveal the social contact between R and I. For this reason, the Requester might not want that *all* Domains she has already established a trust relationship to, are queried in order to protect her own privacy. 2) sending a Counsel Request to *all* known and trusted Domains is not useful when we assume the existence of distinct social communities. For instance, it is not useful to send a Counsel Request to a friend when trust should be established to a colleague. The likelihood that a friend might act as Counselor in this case is low. 3) a careful selection of Counselors helps reduce the risk of performing a trust exchange with the wrong Domain. For instance, the Requester might know two John Smiths. By sending Counsel Requests only to Counselor Domains that belong to the right social community, the chance that trust is accidentally established to the "wrong" John Smith's Domain is reduced.

As R and all Counselors already share a trust relationship, R's exchange service knows a Counselor's cryptographic identity. Each cID is resolved to an IP addresses using above explained DHT lookup and a secured TLS connection is established to the Counselor's exchange service. The mutual authentication of this connection is possible, as both Domains have already exchanged their Domain CA certificates. R's exchange service now sends a Counsel Request to C's remote exchange service, which contains hID_I and a random trust exchange ID (txID).

The owner of each queried Counselor Domain must decide whether she wants to reply to a previously received Counsel Request. This permission is required because sending a *Counsel Reply* to the Requester will reveal the existence of a trust relationship between C and I. Various reasons exist why a Counselor might decide not to reply.

The Counsel Reply sent from C's exchange service to R's exchange service contains information needed to securely connect to I, namely I's cryptographic identity (cID_I) and certificate (cert_I). Other meta-information, such as the Invitee's IL (IL_I), as determined by the Counselor, are also included in the message. The IL is important for the R's exchange service to compute the quality of the proposed information using the trust metric explained in Section V-C.

In the worst case, two or more different Invitee Domains might be proposed by different Counselors. In this case, the proposals are sorted and the IL of each proposed Invitee Domain is computed. The result of this operation is presented to the owner of the Invitee Domain. It is the owner's task to select the most promising looking proposal.

In Fig. 2, part 3, the Requester's exchange service contacts the Invitee's exchange service after resolving I's previously received cryptographic identity to her IP address. Over an half-authenticated TLS connection an *Exchange Request* is sent that includes the Requester's own certificate (cert_R) and identity (cID_R), a list of used Counselors ($[\text{cID}_C]$) and txID.

I's exchange service does not yet possess any trusted cryptographic credentials of R that could be used to authenticate the Exchange Request. For this reason I's exchange service will contact the Counselors named in the Exchange Request and ask them to perform the authentication of the Exchange Request by sending an *Authentication Request*. This message includes cID_R, cert_R and txID (Fig. 2, part 4).

The Counselor's exchange service now searches within its state if a trust exchange session was recently started that can be identified by txID, cID_R and cert_R. If the session exists, the legitimacy of the Exchange Request is confirmed. The Counselor Domain's exchange service communicates this fact by sending a positive *Authentication Reply* to I's exchange service. The reply furthermore contains the Identification Level of R (IL_R) as assessed by C and txID.

In the final phase of the trust exchange, depicted in Fig. 2, part 5, the Invitee's exchange service computes the Identification Level of the new trust relationship using the trust metric explained in Section V-C and asks the Domain owner if the trust relationship should be accepted. The purpose of this step is again to assist the Domain owner and to protect her from establishing a trust relationship with a Domain whose identity is not assured. Furthermore, the Domain owner must have the choice to accept the new trust relationship or not.

 ISBN: 978-1-61208-369-8

Finally in Fig. 2, part 6, the new trust relationship between Domains R and I is established.

C. Trust Metric

Trust metrics are generally used to minimize the risk of falsely trusting claims of other peers in a web of trust. Our trust metric reduces the likelihood of falsely established trust relationships in the case of Counselors accidentally or intentionally making wrong assertions about the identity of another Domain. The protocol previously explained makes use of this trust metric in two situations: 1) after the Requester received Counsel Replies from Counselors and 2) after the Invitee received Authentication Responses from Counselors.

The aim of our metric is to compute the Identification Level (IL) of a new trust relationship, i.e., to rate the trustworthiness of a received certificate. As a basis for this computation, Counselors include in their Counsel and Authentication Replies the IL of the Invitee or of the Requester as assessed by herself.

If only one Counsel Reply was received, the Identification Level IL_{RI} of a trust relationship between Requester (R) and Invitee (I) can be computed as follows, see (1) and Fig. 3. Please note that the same formula can be applied to rate IL_{IR} based on one Authentication Reply.

$$IL_{RI} = IL_{CI} \cdot \frac{RL_{RC}}{RL_{Max}} \cdot \frac{IL_{RC}}{IL_{Max}} \cdot d \qquad (1)$$

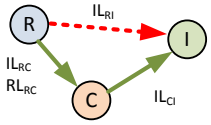

Figure 3. Identification and Reputation Levels between Domains

Explanation of (1): The credibility of a propagated IL_{CI} depends on two properties of the trust relationship to the Counselor. 1) the Reputation of the Counselor, i.e., how sure the Requester can be that the Counselor assigned the right IL to the propagated trust relationship. 2) the IL the Requester has assigned to the Counselor by himself. So, the formula expresses that the IL_{CI} of a trust relationship established over a *securely* identified Counselor that has a *very good* reputation has *high* credibility. If the Counselor is rated *not* to be very reliable or was *insecurely* identified herself, IL_{CI} is decreased by the formula.

Additionally, a dampening factor d decreases the ILs established over the Internet as we argue that exchanging trust indirectly without personal human interaction can never be as reliable as exchanging trust directly in person. Hence, the maximal IL_{RI} of a trust relationship that can be established over the Internet is $IL_{Max} \cdot d$. In the current implementation we use a static dampening factor $0,9$, so the maximum IL will be $10 \cdot 0,9 = 9$.

If more than one Counselor has sent Counsel Replies, the single ILs can be aggregated as follows, see (2). Please note that the same calculation can be applied to rate IL_{IR} based on several Authentication Replies

$$IL_{RI} = \frac{\sum_{C \in Counselors} IL_{CI} \cdot \frac{RL_{RC}}{RL_{Max}} \cdot \frac{IL_{RC}}{IL_{Max}}}{count(Counselors)} \cdot d \qquad (2)$$

Formula 2 computes the algebraic average of the IL_{CI} values propagated by each single Counselor. As in (1) each IL_{CI} is weighted by the Counselor's Reputation and Identification Level. We furthermore propose to adjust the dampening factor d dynamically according to the amount of received Counsel Replies. If more than three Counsel Replies were received that all proposed the same Invitee Domain, the dampening can be set to 1, resulting in no dampening at all.

In Fig. 4, part 1, we have depicted a simple example scenario with IL and RL values where R has established a trust relationship to I.

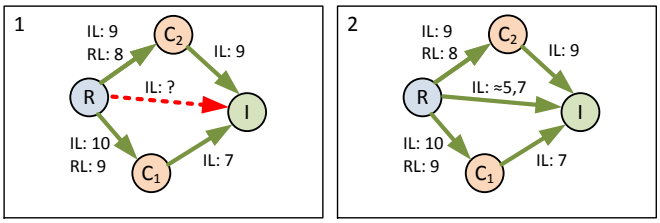

Figure 4. Example: Aggregation of propagated Identification Levels.

The IL_{RI} of the new trust relationship can be calculated by inserting known IL and RL values and propagated IL values into (2). d is set to 0,9 as only two Counselors participated, see (3):

$$IL_{RI} = \frac{7 \cdot \frac{9}{10} \cdot \frac{10}{10} + 9 \cdot \frac{8}{10} \cdot \frac{9}{10}}{2} \cdot 0,9 \approx 5,7 \qquad (3)$$

VI. Evaluation

In the following, we want to evaluate the presented mechanism using the requirements we defined in Section IV. Security requirements R1 and R3 are satisfied as all communication channels are encrypted and authenticated. No third party can interfere with the trust exchange. The trust exchange furthermore queries different independent counselors and rates the authenticity of received certificates with our trust metric (R2). This reduces the likelihood that a willingly or accidentally proposed unintended certificate gets accepted. Furthermore, it reduces the maximum distance in the social graph that can be bridged with repeated Internet trust exchanges. However, we still suggest to strengthen trust relationships established over the Internet with the personal trust exchange when the opportunity is given. Existing trust relationships within a social community cannot be enumerated by attackers that are not already part of this group (R5). The reason for this is that outside attackers are unable to contact exchange services that belong to this social community as their cID and IP are unknown. Additionally, an attacker would not be able to successfully authenticate towards an exchange service. Within the community, trust relationships can be hidden on demand of a Domain owner acting as Counselor. However, the current protocol does not protect an Invitee's privacy yet. In case a Counselor decides to take part in the Internet trust exchange, the trust relationship between this Counselor and the Invitee is revealed to the Requester. However, this issue can be fixed with a slightly modified protocol. The mechanism is also strongly human centric (R4) as there are no central authorities involved and as we always leave the last word to the Domain owner, i.e.,

 ISBN: 978-1-61208-369-8

whether she wants to assist in a trust exchange or whether she wants to accept a new trust relationship as an Invitee (R7). The proposed system is furthermore fairly easy to use and does not require special skills as we provide a high degree of automation (R6). The Domain owners only need to control their exchange service, which takes care of the rest. The amount of interaction between the exchange service and its owner is minimized. Each involved Domain owner is only bothered once when a trust exchange is performed (R8). This design principle is also good for performance as for each phase of the trust exchange only one waiting period for user input exists. The responsiveness of our system is therefore optimized as well. Nevertheless, we need to point out that performing the Internet trust exchange might need a considerable amount of time when human users do not respond quickly (R9).

VII. Related Work

The mechanism we presented has some similarities with FOAF+SSL (Friend of a Friend + Secure Socket Layer) [12]. Both leverage a user's social graph to establish trust into keying material. Whereas our mechanism is targeted to establish trust in long-term keys, FOAF+SSL is an on-the-fly authorization mechanism for the semantic web. FOAF+SSL's aims can best be subsumed with "friends of my friends may access my website". One problem of FOAF+SSL we see is that the system requires publicly available data structures called WebIDs, which reveal a user's social graph.

The SecBook project, which is to our knowledge a discontinued student project, followed a different approach for trust establishment over social communities. SecBook used the Facebook API to store a public key and other information in a user's Facebook profile. This approach made it quite easy for users to identify their friend and obtain her public key. However, this mechanism was limited to Facebook users and was only as secure as a user's Facebook password. When a weak password was used, an attacker was easily able to hijack the profile and exchange the public key.

Monkeyspere [13] is a tool that addresses (amongst other aims) problems with not authentifiable certificates in the public World Wide Web. Once a user reaches a web page whose certificate cannot be authenticated, as it is self-signed, for instance, friends of the user are queried by the Monkeysphere tool and asked whether they trust this certificate. If they do, trust into this certificate is established locally as well.

VIII. Conclusion and Future Work

In this paper, we pointed out the importance of small-scale private networks and their need for secure identification and authentication as basis for strong network and service access control. We outlined various parts of our past work that focused on secure and user-friendly mechanisms for key and identity management within unmanaged network Domains. Another important concept we outlined is a human-centric trust establishment mechanism performed personally between owners of Domains. As this mechanism is limited to situations where a personal meeting is possible, we introduced in this paper a new trust exchange mechanism that can be performed over an insecure network requiring no personal meeting of participants. The central idea of this mechanism is to leverage Domains that already share a trust relationship to each of the Domains that want to establish a new trust relationship with each other. These so-called Counselor Domains act in our protocol as a specific type of trusted third party and assist the Domains that want to exchange trust. Although these mechanisms were presented in the context of private home networks all concepts are generic and can be adapted to various scenarios.

Currently, we work on including the described mechanisms in our "Living Lab" as security infrastructure. In the Living Lab we explore this and other technologies related to smart buildings and how humans interact with it in a secure and privacy-protecting manner. Furthermore, we work on an extension of the proposed trust exchange mechanism able to establish trust between Domains over more than one hop.

Acknowledgments

Our work has been supported by the German Federal Ministry of Education and Research, projects IDEM (grant 01LY1217C) and BaaS (grant 01IS13019G) and by the European Commission FP7 projects EINS (grant 288021) and SecFuNet (grant 288349).

References

[1] S. Hickey, "The Raspberry Pi computer," Article in The Guardian online, http://www.theguardian.com/technology/2014/mar/09/raspberry-pi-computer-eben-upton-cambridge [retrieved August 2014], March 2014.

[2] openHAB UG, "openHAB," Website, http://www.openhab.org/ [retrieved August 2014].

[3] G. Greenwald and E. MacAskill, "NSA Prism program taps in to user data of Apple, Google and others," Article in The Guardian online, http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data [retrieved August 2014], June 2013.

[4] G. Greenwald, "XKeyscore: NSA tool collects 'nearly everything a user does on the internet'," Article in The Guardian online, http://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data [retrieved August 2014], June 2013.

[5] ownCloud Inc., "ownCloud," Website, https://owncloud.org/ [retrieved August 2014].

[6] Seafile Ltd., "Seafile," Website, http://seafile.com [retrieved August 2014].

[7] A. Müller, H. Kinkelin, S. K. Ghai, and G. Carle, "An Assisted Device Registration and Service Access System for Future Home Networks," in proceedings of IFIP Wireless Days 2009, Paris, France, 2009.

[8] A. Müller, H. Kinkelin, S. K. Ghai, and G. Carle, "A Secure Service Infrastructure for Interconnecting Future Home Networks based on DPWS and XACML," in proceedings of ACM SIGCOMM Workshop on Home Networks (HomeNets) 2010, New Delhi, India, 2010.

[9] T. Moses, "eXtensible Access Control Markup Language (XACML) Version 3.0," OASIS Standard, 2013.

[10] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280 (Proposed Standard), Internet Engineering Task Force, May 2008, updated by RFC 6818. [Online]. Available: http://www.ietf.org/rfc/rfc5280.txt

[11] P. Zimmermann, "Why OpenPGP's PKI is better than an X.509 PKI," Website, 2012, online articile, http://www.openpgp.org/technical/whybetter.shtml [retrieved August 2014].

[12] H. Story, B. Harbulot, I. Jacobi, and M. Jones, "FOAF+SSL: RESTful Authentication for the Social Web," in proceedings of European Semantic Web Conference, Workshop: SPOT2009., Heraklion, Greece, 2009.

[13] "The Monkeysphere Project," Website, 2013, online article, http://web.monkeysphere.info [retrieved August 2014].

 ISBN: 978-1-61208-369-8