Privacy Through Pseudonymity in User-Adaptive Systems

ALFRED KOBSA
University of California, Irvine
and
JÖRG SCHRECK
University of Essen

User-adaptive applications cater to the needs of each individual computer user, taking for example users' interests, level of expertise, preferences, perceptual and motoric abilities, and the usage environment into account. Central user modeling servers collect and process the information about users that different user-adaptive systems require to personalize their user interaction.

Adaptive systems are generally better able to cater to users the more data their user modeling systems collect and process about them. They therefore gather as much data as possible and "lay them in stock" for possible future usage. Moreover, data collection usually takes place without users' initiative and sometimes even without their awareness, in order not to cause distraction. Both is in conflict with users' privacy concerns that became manifest in numerous recent consumer polls, and with data protection laws and guidelines that call for parsimony, purpose-orientation, and user notification or user consent when personal data are collected and processed.

This article discusses security requirements to guarantee privacy in user-adaptive systems and explores ways to keep users anonymous while fully preserving personalized interaction with them. User anonymization in personalized systems goes beyond current models in that not only users must remain anonymous, but also the user modeling system that maintains their personal data. Moreover, users' trust in anonymity can be expected to lead to more extensive and frank interaction, hence to more and better data about the user, and thus to better personalization. A reference model for pseudonymous and secure user modeling is presented that meets many of the proposed requirements.

Categories and Subject Descriptors: H.4.m [**Information Systems Application**]: Miscellaneous

General Terms: Human Factors, Security

Additional Key Words and Phrases: User modeling, user-adaptive systems, personalization, security, privacy, secrecy, anonymity, pseudonymity, encryption, access control, reference model, Chaum mix, KQML, personal information

This research has been partly supported by the National Science Foundation (NSF) Center for Research on Information Technology and Organizations (CRITO) at UC Irvine.
Correspondence address: School of Information and Computer Science, University of California, Irvine, Irvine, CA 92697-3425; email: kobsa@uci.edu; Joerg.Schreck@acm.org.

1. INTRODUCTION

User-adaptive (or "personalized") applications aim at anticipating the needs of each individual user, and at adapting to these needs while interacting with the respective user. For instance, a personalized web-based application may tailor the content, structure and presentation of information to each user by for example,

—selecting a brief general or a detailed technical description, depending on the user's presumed background knowledge,
—highlighting recommended links based on the user's presumed interest in the information or in products described on the linked page, and
—choosing between textual, graphical and audio presentation, depending on the users' preferences [Brusilovsky 1998; Kobsa et al. 2001].

These adaptations are performed to supply each user with all relevant information in a form that is suitable for him and does not overtax him.

In order to be able to individually adapt to the current user, user-adaptive applications rely on information about individual characteristics of each user, and characteristics of the user groups to which a user belongs. This information is stored in so-called *user models* (in the case of individual users) and *stereotypes* (in the case of user groups). Frequently employed information about users includes

User Data. Demographic data, user knowledge, skills, capabilities, interests, preferences, goals, and plans

Usage Data. Observable usage (e.g., selective actions and ratings) and usage regularities (e.g., usage frequency and action sequences)

Environment Data. Software and hardware environment, and the user's current location, for example.

This information is individually acquired for, and associated with, each user. The user model persists across different user sessions, and a user must be linked to her user model at the beginning of each session. Chin [1993], Kobsa [1993], and Kobsa et al. [2001] describe the methods that are currently being used for acquiring information about users, and for representing this information and augmenting it through inferences in a user model. Network-wide *user modeling servers* [Fink and Kobsa 2000; Kobsa 2001a] supply several user-adaptive applications ("clients") with user modeling services for their users, and can thereby achieve synergy effects in personalization. They can compare individual user models with information about the whole user population and about user subgroups, and also merge information from different applications about the same user. Table I gives an overview of major generic user modeling systems (including both research prototypes and commercial systems), and some of the methods they employ (see Fink and Kobsa [2000], and Kobsa [2001a] for a more detailed discussion).

In this article, the term *user-adaptive system* will denote a user, a user modeling server, the user's individual model on this server, and the user-adaptive applications that the user employs and that access this user model. The term *user*

Table I. Generic User Modeling Systems (Research Prototypes and Commercial Systems)

System name	References	Characteristics
GRUNDY	[Rich 1979a, 1979b, 1983]	stereotypes, default assumptions
UMFE	[Sleeman 1985]	propositional logic, conceptual hierarchies, numerical gradation of attributes
GUMS	[Finin 1989]	Prolog, stereotypes
GUMAC	[Kass 1991]	assumptions, rules, stereotypes
um	[Kay 1995]	frames, propositional logic, inspection and modification
BGP-MS	[Kobsa and Pohl 1995], [Pohl 1998]	propositional, first-order, and modal logic, stereotypes, partitions, shared user models
Doppelgänger	[Orwant 1995]	shared user models, propositional logic, statistics, machine learning, inspection and modification
TAGUS	[Paiva and Self 1995]	Prolog, inspection
GroupLens	[Net Perceptions 2002]	collaborative filtering
Personalization Server	[Art Technology Group 2001]	production rules
FrontMind	[Manna 2001]	production rules, Bayesian networks
DPS	[Fink 2003]	content-based filtering, collaborative filtering, production rules

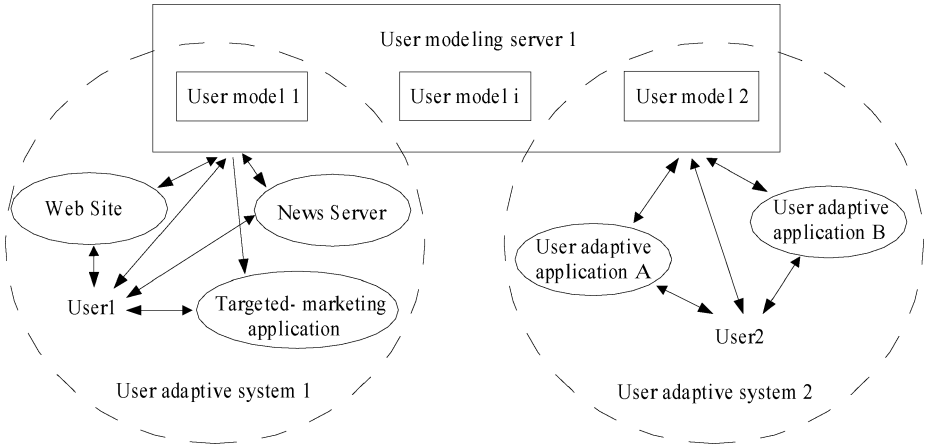

Fig. 1. Two user-adaptive systems and their components.

modeling component denotes a component of a user-adaptive system. Figure 1 shows an example of two user-adaptive systems (they are marked out by dashed circles). User1 interacts with a personalized newsreader and a personalized website. Both communicate information about the user to a user modeling server, and adapt their offerings based on assumptions about user interests that they receive from this server.[1] A targeted-marketing application also receives information about the interests of User1 and supplies her with personalized

[1]An implemented example of such a personalized service is the DPS user modeling server that makes assumptions about users' interests based on their web navigation behavior in several commercial news sites [Fink and Kobsa 2002; Fink et al. 2002].

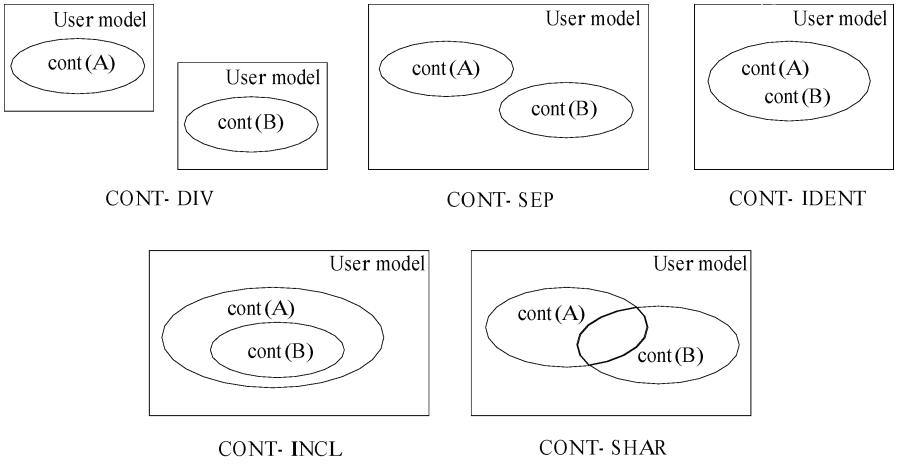

Fig. 2. Modes of cooperation between application systems.

advertisements. User2 also interacts with two user-adaptive applications. Information about this user is stored in a different user model on the same user modeling server.

Figure 2 shows the possible relationships between the user models that two clients A and B maintain about the same user on a user modeling server (*cont*(*A*) and *cont*(*B*) denote the contents of these models):

CONT-DIV. *Cont*(*A*) and *cont*(*B*) are in two different user models, that is, components of different user-adaptive systems. The user modeling server is unaware that they belong to the same user.

CONT-SEP. A and B maintain separate entries in the same user model. These entries do not interfere with each other.

CONT-IDENT. A and B maintain a single common user model.

CONT-INCLUDE. *Cont*(*B*) is a subset of *cont*(*A*), that is, all entries made by B are also known to A.

CONT-SHAR. *Cont*(*A*) and *cont*(*B*) intersect. The contents in this intersection are known to both A and B.

In the example of Figure 1, the three applications with whom User1 is interacting can pairwise be in any of the above relationships. In the case of CONT-IDENT, CONT-INCLUDE and CONT-SHAR, direct sharing of user model contents between two clients, and thus a flow of information, can take place. In CONT-SEP, content sharing and information flow can only occur via the user modeling server (e.g., when it corrects one user model based on the contents of another). We will get back to these models in the remainder of this article when privacy issues are being discussed.

2. PRIVACY IN USER MODELING

Privacy has been defined in many ways, such as the “right of the individual to be let alone” [Warren and Brandeis 1890], the right of people “to determine

for themselves when, how, and to what extent information about them is communicated to others" [Westin 1970, p. 7], and through the demand for "giving people property rights in information about themselves and letting them sell those rights freely" [Posner 1984, p. 336]. A *privacy policy* is a set of specifications that regulate the processing of personal data. Privacy policies can be influenced by many factors, including laws, self-regulatory guidelines, ethics, and user preferences. In this article, we are not going to propose specific privacy policies for user-adaptive systems, but rather present security measures that guarantee that the processing of data about users complies with given policies. In this vein, we will first summarize regulatory measures and documented user concerns that influence privacy policies in user-adaptive systems, and discuss their impacts on the design of such systems.

2.1 Privacy Laws and Self-Regulatory Privacy Principles

Privacy laws regulate the processing of personal data in currently more than 30 countries and some states, provinces and counties. These laws differ considerably and are also still in flux [Rotenberg 2002; Kobsa 2001b; Kobsa 2002]. However, they are usually based on a relatively small number of underlying "privacy principles" (e.g., OECD [1992] and CSA [1996]).

Information about the current user that is kept and processed in a user modeling system falls within the scope of "personal data" as defined in such privacy laws. For instance, the *EU Data Protection Directive* [EC 1995], which sets minimum standards for the national privacy laws of all European Union member states, defines personal data as follows:

> "personal data" shall mean any information relating to an identified or identifiable natural person ("data subject"); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity.

For data to qualify as personal data, it is thus sufficient that they *can* be linked to an identifiable person. It is not necessary that users must fully identify themselves (e.g., by revealing their names and addresses) to have their data protected under this directive, nor that identification actually takes place. For user modeling purposes, this is particularly relevant since user modeling aims at collecting considerable amounts of information about users to support personalized behavior. Models of individual users may thus become sufficiently large and differentiated from the other models that the identification of users becomes *in principle* possible.

Another example of a privacy law is the German *Teleservices Data Protection Law* [TSDP 2001], which regulates the processing of personal data in online services. User-adaptive systems on the Internet fully fall under the scope of this law. Among other things, it specifies the circumstances under which usage profiles are permitted, limits the retention period of usage data that is associated with an individual, demands anonymous or pseudonymous access for users to

the extent to which this is technically feasible and reasonable, and requires that users must be able to view their profiles. This law severely limits the user modeling methods that can be applied without having to ask for the user's consent.

Self-regulatory privacy principles control the processing of personal data on the level of an industry sector or a company in a mandatory manner. For instance, according to the principles for online preference marketing of Network Advertising Initiative [2000], network advertisers may not, without prior affirmative consent ("opt-in"), merge personally identifiable information ("PII") with information previously collected as nonpersonally identifiable information, and use personally identifiable information consisting of PII collected offline merged with PII collected online for online preference marketing, unless the consumer has been afforded robust notice and choice about such a merger before it occurs. On the company level, self-regulatory privacy principles have meanwhile become a standard for major web sites under the name "privacy policies."

The Platform for Privacy Preferences Protocol (P3P) [Reagle and Cranor 1999; Cranor et al. 2002b] allows websites to express and communicate their data collection practices and privacy policies both with regard to data explicitly provided by the user and clickstream data. The APPEL language [Cranor et al. 2002a] will allow users to define their own personal privacy preferences. Web users will be alerted when the proposed privacy policy does not meet their requirements, and they can thereupon grant exceptions or leave the site.

2.2 User Demands

Personal demands for privacy in user modeling can be influenced by factors such as

—general personal preferences for privacy in information technology (e.g., whether anonymous or identifiable use of information systems is preferred);
—personal desire to keep different sets of characteristics apart from each other (which has an impact on, e.g., whether different applications may only share a small or rather a large part of the personal data in a user model);
—personal roles that a user assumes when interacting with user-adaptive systems (e.g., that of a company employee at work or a private citizen at home); and
—personal appreciation of the benefits that user-adaptive systems provide (e.g., whether the added value of personalization is deemed worth disclosing personal information).

Numerous surveys were already conducted to determine user preferences regarding online privacy. Table II summarizes findings that relate to the topic of this article. Internet users are apparently very concerned about their privacy online and are reluctant to divulge information about themselves.[2] So far, no

[2]Cranor et al. [1999] and Mabley [2000] show that the reluctance to disclose personal data partly differs with the type of personal data in question. Spiekermann et al. [2001] found that it also depends on the context, and that that discrepancies seem to exist between users' claimed concerns and their actual behavior. The multi-national results of IBM [1999] indicate that in countries that have privacy laws enacted, privacy concerns are not significantly lower.

Table II. User Concerns Regarding Privacy on the Internet

Respondents asserted to	% agreement ("strong"/"very" and "somewhat")
being (very) concerned about threats to their privacy when using the Internet	81% [Westin and Maurici 1998], 87% [Cranor et al. 1999]
being extremely/very concerned about divulging personal information online	67% [Forrester Research 1999], 74% [DePallo 2000], 70% [Culnan and Milne 2001]
being (extremely) concerned about being tracked online	54% [Fox 2000] 77% [DePallo 2000]
having left web sites that required registration information	41% [Boston Consulting Group 1997]
having entered fake registration information	40% [GVU 1998], 27% [Boston Consulting Group 1997], 32% [Forrester Research 1999], 24% [Fox 2000], 34% [Culnan and Milne 2001]
having refrained from shopping online due to privacy concerns, or bought less	32–61% [IBM 1999], 32% [Forrester Research 1999], 24% [DePallo 2000], 64% [Culnan and Milne 2001]
being willing to give out personal data when they get something valuable in return	31% [GVU 1998], 30% [Forrester Research 1999], 51% [Personalization Consortium 2000]
value being able to assume different aliases/ roles on the Internet	58.8% [GVU 1998]
rate privacy as more important than convenience	75.5% [GVU 1998]
value being able to visit sites on the Internet in an anonymous manner	81.1% [GVU 1998]
value being able to communicate over the Internet without people being able to read the content	93.2% [GVU 1998]

survey focused on users' attitude with regard to the collection of usage information and the inference of assumption about users for personalization purposes. However, more than 80% of the respondents in GVU [1998] favored anonymous interaction, nearly 60% of the respondents would like to act in different roles when using the Internet, and 3 out of 4 rate privacy as more important than convenience. These responses make it unlikely that users will tolerate that large amounts of personally identifiable data about them be stored in a central user modeling server, unless the personalization benefits they receive are extremely valuable to them.

2.3 Consequences from Privacy Requirements and Outline of this Article

User-adaptive systems on the web collect far more personal data than regular websites, and do so in a rather surreptitious manner. It can be expected that users' privacy concerns will be exacerbated when such systems become widely put in use. The aim of this article is to develop an architecture that enables users to remain anonymous to personalized websites, but nevertheless allows websites to track users individually, draw inferences about them, and adapt to them individually. Surprisingly, there currently seems to be a belief that anonymity and personalization do not go together. For instance, Schafer et al. [2001] claim the following with regard to recommenders, that is, user-adaptive systems whose recommendations are based on what the current user, as well as

similar users, selected in the past: "Anonymizing techniques are disasters for recommenders, because they make it impossible for the recommender to easily recognize the customer, limiting the ability even to collect data, much less to make accurate recommendations. If recommenders are to be successful in the long-term, alternatives must be developed that alleviate consumer concerns about privacy [. . .]".

We argue in contrast that personalized interaction and anonymous interaction are *not* antagonistic if one can ascertain that actions pertaining to the same user can be linked with each other. We propose an architecture for privacy and security in personalized systems that allows users to benefit from personalization while hiding their identities as well as the identity of the user modeling servers that administers the data that is collected and inferred about users. In this vein, we will first analyze requirements for secrecy in user-adaptive systems. We focus on anonymity, and specifically pseudonymity which reveals no identifiable information about users beyond what they voluntarily supply, allows them to maintain one or more persistent identities, and can optionally be revoked if required (e.g., in the case of abuse). In Section 4, we present a privacy-preserving reference architecture for pseudonymous yet personalized interaction across applications and individual sessions. Its components include secure communication at the application and the transport layers, a mix network, and a hierarchical role-based access control model. In Section 5, finally, we discuss possible extensions to our approach.

3. SECRECY, ANONYMITY AND PSEUDONYMITY IN USER-ADAPTIVE SYSTEMS

Secrecy refers to denying to unauthorized individuals the access to certain information [Simmons 1992]. Denial of access to personal data in user-adaptive systems can be realized by denying to unauthorized components the access to the relationship between the user and her personal data (i.e., the user model entries), and to the personal data themselves. (Examples of such unauthorized components would be, e.g., unauthorized user model clients and imposters who feign to be legitimate components of user-adaptive systems.) We deal with these two cases through *anonymization* and *encryption* of user information, respectively. Anonymization conceals the relationship between a particular user and the data about him. User model entries can no longer be assigned to a particular user, thus ensuring that they will remain secret. Encryption protects personal data from inspection when being exchanged between the user model and its clients. Using an appropriate cryptographic system, the authorized recipients of the information can also be specified in the encryption process.

3.1 Types of Anonymity

To be effective, different types of anonymity must be maintained. For instance, a well designed system providing anonymity or pseudonymity in a secure and provable manner may be futile if it is used by a single person only whose identity is known by means outside of the system. Gavish and Gerdes [1998] distinguish

three types of anonymity which must be considered:

—*Environmental anonymity* is determined by external factors, including the number and diversity of users, and prior knowledge about them. These factors, and hence environmental anonymity, cannot be altered through the design of the system but have to be monitored over time while the system is in operation. Since user-adaptive systems collect far more personal data about users than regular systems, the danger of attacks seems to be higher (cf. Ramakrishnan et al. [2001]).

—*Content-based anonymity* prevails when no identification by means of the exchanged data is possible. De-anonymization may occur on the basis of, for example, the data content (e.g., names, addresses, email addresses, or unique combinations of data values), their structure (e.g., representation of data in a form that is typical for particular users or software they use), or by sequence (e.g., repeating patterns that make it possible to link otherwise unconnected sessions). When the *inference integrity* of a user model is violated, knowledge about unique characteristics of an individual can be exploited for inferring all other characteristics (an example from the domain of user-adaptive systems is given in Schreck [2003]).

—*Procedural anonymity* is determined by the communication protocol and the underlying communication layers. This type of anonymity can be provided by the system and should be planned for in the design phase of the system. *Sender anonymity* is present when the sender of a message cannot be identified by the recipient of a message within the set of potential senders. *Receiver anonymity* means that the identity of the receiver is not known to the sender of a message. The ability to send data to anonymous recipients is especially important for answering queries received under sender anonymity, which is a frequently occurring interaction pattern in user-adaptive systems.

To protect users' privacy through anonymity, all of the above three types of anonymity must be present in a user-adaptive system. In this article, we present solutions for procedural anonymity and outline the integration of mechanisms to improve content-based anonymity. Environmental anonymity must be ascertained with means outside of the system (see Section 5 and Schreck [2003] for existing techniques).

3.2 Levels of Anonymity

For the purposes of user-adaptive systems, it is useful to discern five levels of anonymity, ranging from unequivocal assignment of data to a user to her complete indistinguishability from other users (also see Flinn and Maurer [1995] and Pfitzmann and Köhntopp [2001]).

Super-Identification. The user identifies herself to a certification authority, which in return assigns unique credentials to her (e.g., X.509 certificates). These credentials authenticate the user towards the user-adaptive system (user modeling clients can be authenticated as well).

Identification. The user identifies himself to the system.

Latent Identification. The user identifies himself to a trustee and adopts a unique pseudonym that becomes registered with his identity. Using this pseudonym, he is subsequently able to interact with the system without revealing his identity.

Pseudonymous Identification. The user initially chooses a unique but otherwise uncontrolled pseudonym, which he will also employ in subsequent sessions.

Anonymity. The user uses the system without any identification or identifier that distinguishes her from other users.

From the perspective of user modeling, not all anonymity levels are equally important. Full user *anonymity* does not allow the user-adaptive system to distinguish among users, and to offer differentiated services. *Pseudonymous identification* (or "initially unlinked pseudonyms" [Pfitzmann and Köhntopp 2001]) seems to be the best compromise between privacy demands and the requirements of user modeling. This type of identification differentiates users based on the unique pseudonyms which they themselves have chosen and which also authenticates them (possibly in combination with their chosen passwords). Users are thereby not required to reveal their identities. *Latent identification* (or "escrowed identity" [Kilian and Petrank 1998], "initially nonpublic pseudonym" [Pfitzmann and Köhntopp 2001]) additionally allows the system to determine the identities behind the pseudonyms, in collaboration with one or more registrars/trustees who issued the pseudonym. This revocation of pseudonyms may be desirable in cases of misuse or when the identification of the user becomes necessary for other reasons, such as nonanonymous payment and delivery scenarios. In terms of ISO [1999] and Pfitzmann and Köhntopp [2001]), pseudonymous users of user-adaptive systems should be

Unidentifiable. Neither the user-adaptive system nor third parties should be able to determine the identity of pseudonymous users;

Linkable for the User-Adaptive System. The user-adaptive system can link every interaction with a specific user, even across sessions (users maintain a persistent identity);

Unlinkable for Third Parties. Third parties (including other components of the user-adaptive system) cannot link two interaction steps of the same user;

Unobservable for Third Parties. The usage of a user-adaptive application by a user should not be recognizable by third parties.

Moreover, users should be able to not only adopt a single person pseudonym, but several pseudonyms such as:

Role Pseudonyms. To interact with the same site in different roles (e.g., as an employee at work, or a private citizen at home);

Relationship Pseudonyms (or *Application Pseudonyms*). To interact with different sites under different pseudonyms; and

Role-Relationship Pseudonyms. To combine both of the above.

Role pseudonyms were demanded by many respondents in the survey presented in Table II. Using multiple pseudonyms enhances environmental and content-based anonymity. In user-adaptive systems, it enforces the creation and maintenance of multiple separate user models, one for each pseudonym (see CONT-DIV in Figure 2). While separate role pseudonyms may improve personalization if users exhibit different personal characteristics in different roles, separate relationship pseudonyms may hurt the quality of personalization since synergy effects between assumptions made by different user-adaptive applications cannot occur any more. Transaction pseudonyms [Pfitzmann and Köhntopp 2001] that are different for each transaction would even bar any user modeling since linkability is no longer preserved.

Finally, in the case of *identification* by the system, all components are aware of the identity of the respective user. If there is a chance that a user's identity could be usurped by another user or software component, *super-identification* should instead be employed. The responsibility for the assignment of identifying data to the user is thereby delegated to a mutually trusted outside component. In the area of user-adaptive systems, this is especially useful for tutoring systems that eventually assign credentials to users (e.g., certificates of examination) since the identities of the respective users and the components of the assessing system as well as the authenticity of the data must all be verifiable.

No single level of anonymity is suitable for all user-adaptive systems. Depending on the application type (e.g., tutorial systems) and its domain (e.g., electronic commerce), different levels of user anonymity can be required. A specific characteristic of user-adaptive systems is moreover that not only the user but also the user modeling server may need to remain anonymous. User models may reside anywhere on the network, like on the user's platform (as is envisaged, e.g., in the P3P framework [Cranor et al. 2002b]) or on a remote server (such as in Microsoft's Passport or Novell's DigitalMe[3] architectures). A location close to the user (like on hci.ics.uci.edu or even www.jschreck.de) may compromise her anonymity. To safeguard it, mechanisms are therefore also needed that protect the anonymity of user modeling servers.

3.3 Anonymity Complexity

Establishing anonymity requires a component within the user-adaptive system which carries out the anonymization procedure (e.g., a trusted third party). This entity is able to defeat the user's anonymity. A user may therefore prefer to include several entities in the anonymization process, in which he trusts collectively. The user's anonymity can then only be defeated if all entities conspire. To assess this quality of the anonymization process, Gavish and Gerdes [1998] define *anonymity complexity* as the maximum number of colluding entities which cannot defeat the anonymity of the system. *Order-N anonymity*, represented as OA(N), indicates that N+1 entities must collude to defeat anonymity. This measure expresses the degree of protection that is provided in systems that

[3]See www.passport.com and www.digitalme.com.

offer anonymity. Important values are:

OA(0). In systems with anonymity complexity 0, a single entity can defeat the anonymity. This is the case for identification by the system, where each component is aware of the identity of the user and where therefore a single entity can misuse this knowledge.

OA(1). This anonymity complexity can be found in systems that use controlled pseudonyms. Two entities must act jointly to defeat the anonymity, for example, a component of the user-adaptive system and the registrar of the pseudonyms.

OA(N). In this case, N entities are unable to defeat the anonymity of the user. To defend her anonymity, the user has to include at least one trusted entity into every set of N+1 components that might jointly defeat her anonymity.

With the concept of anonymity complexity, individual user requirements regarding the minimal number of entities involved in the anonymization process can be expressed and operationalized. A user-adaptive system which supports complexity OA(N) is most beneficial for users since it can adapt to the number of entities required by each individual user, thereby satisfying different subjective thresholds for trust in the anonymization process.

3.4 Risks and Potentials of Anonymity and Pseudonymity

Anonymity in human-to-human communication harbors several risks, for example, reduced suppression of unsocial or criminal behavior, adoption of fake identities, or missing credit for contributions [Anonymous 1996; Gavish and Gerdes 1998]. These anonymity effects are clearly also observable in people's behavior on the Internet. The best answer seems to be an infrastructure that allows one to revoke users' anonymity in clearly and narrowly defined circumstances.

On the positive side, users' interaction with personalized systems is likely to considerably benefit from anonymity. The results from user polls summarized in Table II show that users' reluctance to divulge information about themselves curtails their interaction with computer systems. We may conclude that users will interact more freely if they trust in their anonymity when interacting with a system.[4] An individual who is not subject to social pressure for conformity may also be more strongly differentiated from others, thus allowing for more discriminating personalization. As a consequence, one can expect that user modeling systems will be able to gain more and better data about users, and that user-adaptive applications will be able to cater better to them, the more users trust the anonymity mechanisms. This in turn will lead to better adaptation by, and hence acceptance of, the user-adaptive application. One may even speculate that the known deinhibiting effects of anonymity will allow users to reveal their "real needs" more openly, and user-adaptive applications to cater

[4]Gavish and Gerdes [1998, p. 314] even note: "If anonymity is being used as a device to encourage a more open and frank exchange of information, a system's perceived level of anonymity may be more important than its actual anonymity."

to them with better information and service. Independent of personalization issues, anonymity finally also allows a freer treatment of sensitive issues. The absence of personal stigmatization when treating sensitive issues anonymously in user-adaptive systems (e.g., retrieving information about a disease) might encourage users to make more profitable use of the system.

These and previous considerations lead to the proposition that a pseudonymous access to user-adaptive systems should be provided since this

—is demanded by users,
—hides the relation between users and their data from user-adaptive applications, and thus effects that these applications are not subject to privacy laws and self-regulatory principles any more (since data can no longer be assigned to identifiable persons), and
—can be expected to foster a more frank and extensive interaction with the system, leading to more information about the user and hence a better basis for personalization.

Negative consequences of pseudonymity can be dealt with by mechanisms that allow for its revocation in very narrowly defined circumstances by specially authorized parties. To increase the perceived level of anonymity when interacting with a user-adaptive system, it appears to be advantageous to also include the user in the anonymization process. This means that part of the anonymization process should be performed under the user's close supervision and control. As a desirable side effect, the anonymity complexity is thereby increased by one.

Supporting privacy through pseudonymity in user-adaptive systems requires a reconsideration of current architectures for the use in user-adaptive systems and user modeling systems, and the development of new means and procedures for safeguarding the secrecy of user information in such systems. Such an architecture will be presented in the remainder of this article.

4. SOLUTIONS FOR SECRECY IN USER-ADAPTIVE SYSTEMS

The previous section discussed requirements for secrecy in user-adaptive systems to guarantee users' privacy. Here we present solutions that draw from experiences in developing both generic user modeling systems [Kobsa and Pohl 1995; Fink and Kobsa 2002] and comprehensive user-adaptive applications [Fink et al. 1998; Fink et al. 2002]. The proposed architecture and its implemented components have been designed to be as generic as possible. If choices had to be made between competing methods or mechanisms, quasi-standards or at least frequently employed solutions were adopted.

With regard to the communication language and protocol that enables user-adaptive application systems to exchange information about the user with the user model server, we opted for the *Knowledge Query and Manipulation Language KQML* [Finin and Weber 1993; Labrou and Finin 1997; Covington 1998] which has been employed in several user modeling systems [Paiva and Self 1995; Kobsa and Pohl 1995; Pohl 1998; Machado et al. 1999]. KQML was also proposed as an interface language at the *UM96 User Modeling Standardization Workshop* [Kobsa et al. 1996]. Below is an example of a KQML expression in

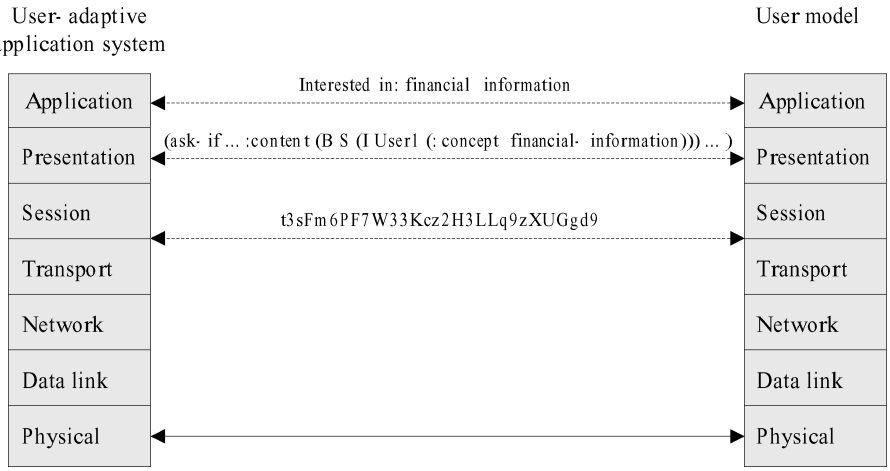

Fig. 3. Encryption through SKAPI below the session level.

which the user-adaptive newsreader of Figure 1 asks the user modeling server whether User1 is interested in financial information. `ask-if` is a *performative* that specifies the type of communicative act (an inquiry in our case). The *parameters* describe the details of the communicative act, namely its sender, its receiver, the language in which the content is expressed, the content itself (namely whether the user modeling system believes that User1 is interested in financial information), and the reference label that should be used in the reply.

```
(ask-if :sender     tcp://adaptive-newsserver:8094
        :receiver   tcp://um-server:8091
        :language   AL
        :content    (B S (I User1 (:concept financial-information)))
        :reply-with query23)
```

4.1 Secrecy Through Encryption

We developed two function libraries (namely, *SKAPI* and *SKQML*) that guarantee the secrecy of exchanged user information during transport when communicating through KQML. The SKAPI functions are located just above the transport layer in the OSI reference model (i.e., the TCP layer, see Figure 3) and require only a few arguments to establish an encrypted channel between two components. This is realized by the *Secure Sockets Layer* (SSL) which encrypts communication by means of sockets [Hirsch 1997; Freier et al. 1996]. We included the SSL library *SSLeay*[5] into the *KQML Application Programmer Interface* (KAPI)[6] to provide transparent encryption. Since encryption is performed just below the session level, the application and the user modeling system need not include cryptographic functions to communicate securely but

[5]See `http://www2.psy.uq.edu.au/~ftp/Crypto/` and *OpenSSL* (`http://www.openssl.org/`).
[6]See `http://www.csee.umbc.edu/kqml/software/`.

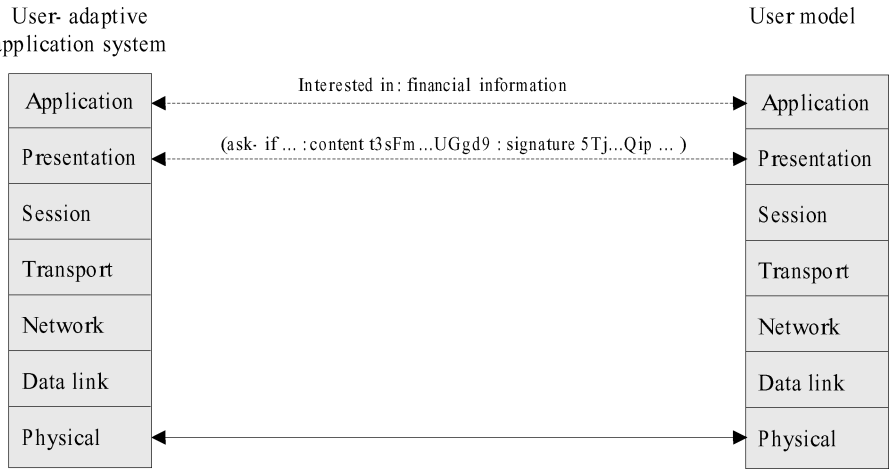

Fig. 4. Encryption through SKQML.

can exchange user information in the same way as before. Nevertheless, communicating processes can take advantage of encryption using various encryption algorithms and key lengths. They can also be authenticated by means of X.509 certificates [Schreck 2003].

For user-adaptive systems that can be modified to include cryptographic functions, we developed the *SKQML* library (for *secure* KQML). It combines the *Java Agent Template Lite* (JatLite)[7] with the Cryptix[8] package. The user model information is thereby encrypted at the application level (see Figure 4) and forwarded to the transport mechanism in encrypted form. Encryption at the application level offers more flexibility to the communication partners than encryption at the transport level (e.g., since a component can decide not to employ encryption if the receiver is unable to process encrypted KQML messages). By using different asymmetric key pairs, components are able to maintain different identities. They can also use different cryptographic keys within the same session. To integrate these security elements into the SKQML language, we enhanced KQML by a new performative and new parameters, which will be described in more detail in Section 4.2.2.

4.2 Secrecy Through Anonymization

4.2.1 *Existing Solutions.* Several solutions for anonymity in different Internet protocols (such as HTTP or E-mail) have been developed in the past, each of which has different protection goals (see Berthold et al. [2000] for an analysis). *Anonymizers*[9] and *LPWA* [Gabber et al. 1997, 1999] allow for anonymity while browsing the Web. Their anonymity complexity is OA(0) only, and they do not keep information secret while in transit. *Anonymous Remailers* [Chaum 1981; Gülcü and Tsudik 1996; Mazières and Kaashoek 1998] introduce

[7]See Petrie [1996] and Jeon et al. [2000], and `http://java.stanford.edu`.
[8]See `http://www.cryptix.org`.
[9]See `http://www.anonymizer.com`, `http://www.rewebber.de`, and `https://www.safeweb.com`.

encryption mechanisms to protect the secrecy of the exchanged information. Information is not only kept secret while in transit, but is also concealed from the intermediaries involved. The user is also able to define their number and the sequence in which they should be used. *Onion Routing* [Goldschlag et al. 1999; Syverson et al. 1997] generalizes these mechanisms in a way that allows various application systems to use the Internet anonymously, regardless of the specific protocol that the application system uses. This versatility has two drawbacks: the anonymization process provided to an application cannot be configured, and a proxy is dedicated to a connection between *one* sender and *one* receiver. *Crowds* [Reiter and Rubin 1998, 1999] implements a mechanism similar to that of Anonymous Remailers for the specific case of web browsing via a proxy, which routes the browser's requests through a network of other Crowds participants.

The *mix technique* was originally introduced by Chaum [1981] and provides sender anonymity as well as receiver anonymity through public key cryptography. Senders can chose a sequence of intermediate hosts through which a message should be routed (the so-called *mix components*, or *mixes* for short). The content of the message as well as its routing through a mix network is kept secret by encrypting them recursively with the public keys of the intermediate mixes in such a way that each mix can only decrypt the address of the next mix to which the remainder of the message is to be routed. To prevent observers from relating incoming and outgoing messages, a mix component forwards incoming messages in random order, sends dummy messages if too few messages arrive within a given period of time, and pads messages to uniform length. Mixes must be arranged in a sequence of sufficient length to obtain the required anonymity complexity (N+1 components are needed for an anonymity complexity of O(N), see Section 3.3). In general, mix sequences can differ in length, differ for each sender, differ for message forwarding and message backwarding, vary with each message, include a mix more than once, contain limited loops, and cannot be altered by a mix in the sequence. Since mixes are independent of a specific user modeling system, they can be arranged and put in place prior to the implementation of the user modeling system. A thorough discussion of generalized structures for mix networks is given in Jerichow [1999].

4.2.2 *A Mix Network for Protecting the Anonymity of Users and of User Modeling Servers.* All frameworks described in Section 4.2.1 include elements that are useful and appropriate for user modeling. Here we describe our KQMLmix framework which combines those elements that we consider to be essential for user modeling purposes: sender anonymity, receiver anonymity, secrecy, authenticity, and the dynamic configuration of these factors. For this purpose, we enhanced the KQML language by a new performative *mix-it*, which is described in Table III.

The following example shows an SKQML message that has been prepared for routing through a mix network by the component mix.privacy.org. It includes the immediate recipient, the encrypted content including information on the mix sequence, the sender's signature that authorizes the content, and encrypted

Table III. SKQML Extensions to KQML

mix-it	This performative instructs the mix to process the message either according to the mix technique or, if the keywords *:mix-list* or *:rpi-list* and their values are present, to prepare the message specified by the value of *:content* for routing through other mixes.
:language MIX	The value advises the mix to apply Base64 decoding to the value of *:content*, and then to decrypt it with its secret key (Base64 encoding keeps the message parsable despite encryption).
:content	The value either contains a message that must be prepared for routing through more mixes, or an encrypted message for the current mix which must be decrypted and dispatched.
:mix-list	An application that does not include cryptographic functions (e.g., since it cannot be modified) can nevertheless send a message to a mix, commissioning it to prepare the message for routing through a mix sequence that is specified by the value of :mix-list.
:rpi-list	The return path information indicates the mixes through which the response to this message should be routed.
:signature	The value is a Base64 encoded signature of the sender for the *:content* value which enables the receiving mix to verify the authenticity of the message.
:RPI	The value contains the Base64 encoded *return path information* that is necessary to guarantee receiver anonymity.

return path information.

```
(mix-it :sender    mix.privacy.org
        :receiver  mix.federalreserve.gov
        :language  MIX
        :content   QWNQeHA0...oOOl4=
        :signature BBICDH+8...4D+Yw=
        :RPI       lS8md5lo...LUJTw=)
```

Components of a user-adaptive system that cannot be enhanced to include the necessary cryptographic functions to prepare the message at the application layer are nevertheless able to take advantage of a mix. The mix can be advised to encrypt a message in such a way that it can be routed through the mix as specified by the *:mix-list* and *:rpi-list* parameters:

```
(mix-it :sender   personalnews.cnn.com
        :receiver mix.priv-commissioner.eu
        :language MIX
        :content  RTE1MzdO...GiZQ==
        :mix-list (mix1 mix34 mix2)
        :rpi-list (mix34 mix3 mix5))
```

In user-adaptive systems, mixes can in principle be employed in the traditional manner, namely for concealing the relationship between users and applications (the fact that the applications are user-adaptive is not relevant here). Much more interesting from the point of view of user modeling is however the fact that they can also be employed for hiding the relationship between applications and user models or user model servers. Figure 5 shows user modeling

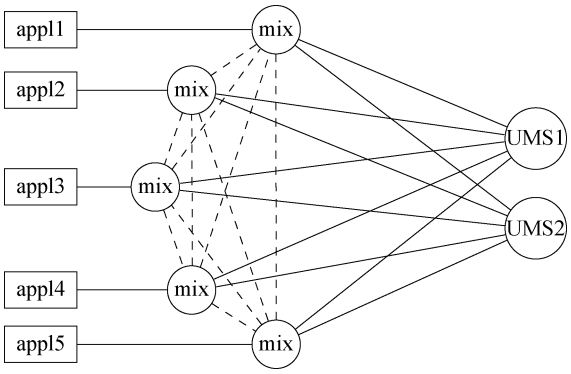

Fig. 5. Mix network.

components using a mix network. Dashed lines symbolize encrypted communication whereas solid lines symbolize communication paths which may or may not be encrypted.

This mix structure can provide sender anonymity as well as receiver anonymity (see Section 3.1) to both the user modeling servers UMS_I and the user-adaptive application systems $appl_i$. Since the user model or user modeling server may reside on the user's network or host rather than a remote host, two of these four possibilities are especially relevant for user modeling purposes:

Sender Anonymity for Messages Sent from UMS_i to $appl_i$. The location (e.g., the network address) of UMS_i must be concealed from $appl_i$ when sending a message to $appl_i$.

Receiver Anonymity for Messages from $appl_i$ to UMS_i. In order to send messages to UMS_i, $appl_i$ must have some means to contact UMS_i without knowing its network address. The return path information (:RPI value, see Table III) in messages received from UMS_i allows $appl_i$ to respond to messages from anonymous user modeling systems. To enable $appl_i$ to initiate a message exchange (as in the case of our targeted-marketing application), the user must supply it with an initial :RPI value, which he can obtain from UMS_i.

In Figure 5, the communication between the mix and the mix clients (i.e., the user modeling servers and the user-adaptive applications) is not necessarily encrypted. To prevent an observer from monitoring the exchanged message, several modifications can be applied to this architecture:

—the communication between the mix and the clients can be encrypted as well;
—the mix, and the mix clients can be placed in a trusted environment; and
—the mix clients can be included into the mix.

At least one of these changes should be made. Most effective is the incorporation of all those clients into the mix network whose communication must be secure. An example based on the usage scenario depicted in Figure 1 is given in Figure 6. Here, the user-adaptive application systems $appl_1$ – $appl_5$ and the user modeling servers UMS_1 and UMS_2 not merely take advantage of a mix

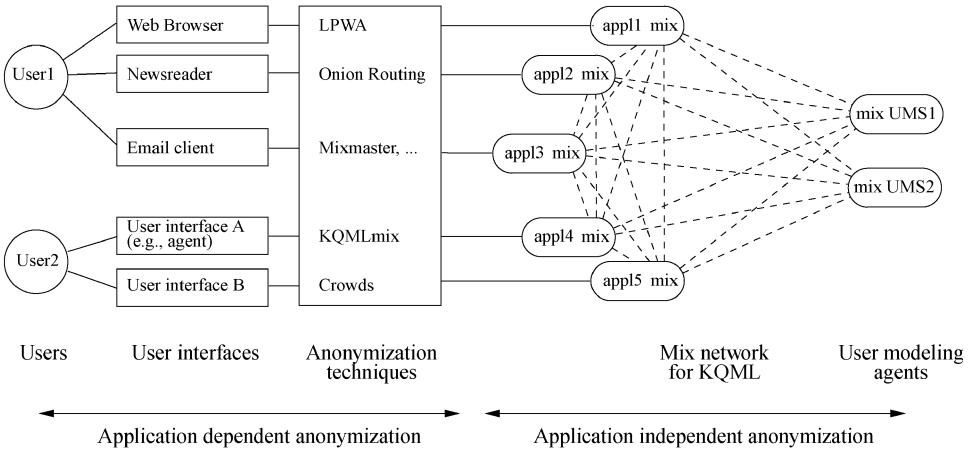

Fig. 6. Mix network with included user modeling components.

network between them but rather constitute the mix network (to increase security, more mixes could be added that have no user modeling function). This yields several improvements:

—The content of messages between the user modeling server viz. user-adaptive application and the first or last mix in a mix sequence cannot be compromised since communication between the mixes is not observable and thus secure.

—Messages exchanged between the user modeling component and the mix network can be authenticated (see Section 4.1).

—A network observer would be unable to distinguish messages which originate from a mix client from messages that are routed through the mix network on behalf of another mix client, and from dummy messages.

So far, we only discussed the righthand side of Figure 6, that is, the application-independent anonymization of KQML messages exchanged between the user-adaptive application system and the user modeling system. To not only keep the user modeling system anonymous from the user-adaptive application but also the user, similar techniques have to be applied between the application and the user. Since these techniques depend on the respective applications and their specific protocols (e.g., HTTP, E-mail), they must be individually selected. Any of the application-dependent anonymization techniques discussed in Section 4.2.1 can be employed on the left-hand side of Figure 6. The KQMLmix implementation can also be used to provide procedural anonymity for HTTP: we developed a proxy which is able to route HTTP requests and the corresponding replies from web servers through a mix network.

4.3 Secrecy Through Selective Access

Secrecy through encryption and anonymization is only sufficient when a user model is made fully available to all user model clients. Often, clients are however only supposed to access and maintain parts of a user model (see Figure 2). Means must therefore be provided to enable users to specify and enforce such a

restricted and possibly partly shared maintenance of user model entries. In the following, we examine well-known security models that control the access to user models and the permissible information flow, and analyze their usefulness for user-adaptive systems. A more detailed study can be found in Schreck [2003].

Noninterference models are most appropriate when several application systems maintaining a common user model need to be completely separated (see CONT-DIV, CONT-SEP in Figure 2):

—The *Chinese Wall security policy* [Brewer and Nash 1989; Kessler 1992] provides a formalism for specifying *conflict classes* (e.g., $cont(A)$ and $cont(B)$ in Figure 2). A user model client can access information from only one of these classes. An assignment of conflict classes to user model clients is not made by this policy. A client that chooses one class from a conflict set (e.g., $cont(A)$) determines for itself which other classes are to be excluded from further requests (for instance, all classes that are in conflict with $cont(A)$). With the Chinese Wall security policy, it is thus not possible to prevent access to a particular class of sensitive information for a particular client.

—The *noninterference model* [Goguen and Meseguer 1982, 1984] makes it possible to separate user model clients that maintains a common user model by assigning clients to groups. In addition to noninterference of clients, noninterference of commands (e.g., the insertion of user model entries) can also be formulated. When combined with one another, particular commands issued by particular clients can be defined as being noninterfering with other clients (i.e., the execution of these commands cannot be detected by other application systems). Crucial for the noninterference model is the history of all issued commands starting from an initial state, and the ability to purge commands from the history. User modeling components usually do not keep a history of all executed commands. However, even if they did, it is likely that a history purged of arbitrarily chosen commands would result in an inconsistent state of the user modeling component. This danger would be particularly high for conflict classes that are not static (e.g., when clients change or join a conflict class).

In contrast to noninterference models, *information flow control models* deal primarily with the information processed within a system, rather than with its clients or the commands they issued. Information flow control models either describe how information can flow within an information system, or which kinds of information flow are prohibited. It is assumed that information flows only within the model described (e.g., within the user modeling system which implements the information flow control model). Information flow between information requesters (e.g., user model clients) is not considered.

—The *multilevel security model* [Bell and LaPadula 1976; Bell 1988] provides means for classifying user model entries and application systems according to content classes and sensitivity levels. The model also defines the permissible information flow in order to prevent information from becoming accessible in content classes that were not assigned to it, or at sensitivity levels that are lower than expected. The *no write-down property* supports

confidentiality within a user-adaptive system but has two counterproductive consequences. First, application systems in which the user trusts (i.e., application systems at a high sensitivity level) are unable to supersede (e.g., *update*, *delete*) entries made by application systems at a lower sensitivity level. This means that trusted application systems are not allowed to correct user model entries made by untrusted application systems. Second, untrusted application systems are not allowed to acquire knowledge of *any* user model entry made by an application in which the user places greater trust. These characteristics lower the trustworthiness of the user model entries and consequently diminish the quality of the user adaptive system.

—The *information flow control model* [Denning 1976, 1982] allows one to specify permissible information flows by arranging security classes in a lattice. This results in a flexible security policy, in contrast to the multilevel security model where the security policy is mandatory. Nevertheless, an information flow between classes caused by a command sequence can only be detected with enormous computational efforts.

Common to the noninterference, multilevel security and information flow models is the characteristic that the defined security policy must be enforced by the user modeling system. For example, the noninterference model relies on the history of commands and on their virtual execution after being purged of certain offending commands. This can only be done by the user modeling system, if at all. The multilevel security model presumes that information is not processed outside the component implementing the security model. Otherwise, it would be possible to retrieve information belonging to a high security level and insert that information at a lower security level (thereby violating the no write-down property). For the information flow control model, it is essential to calculate the conditional entropy for all user model entries before each state transition of the user modeling component. Since inferences within the user modeling component influence the conditional entropy of a user model entry, the calculation can only be performed within this component.

To be implemented within a user modeling system, these security models must be adapted to the specific representation and inference techniques of the particular user modeling system. They are therefore not independent of the user modeling system employed and thus not a good basis for a general security architecture for user-adaptive systems.

4.4 A Hierarchical Role-Based Access Control Model

For the majority of user-adaptive systems, we recommend focusing on the interface between the user modeling system and the user-adaptive system. Controlling the communication between the application and the user modeling system makes it possible to specify the joint maintenance of a user model through selective access to it. This approach makes no assumptions about the internal structure and processing of the user modeling component and is therefore generally applicable.

We have chosen *role-based access control* (RBAC)[10] to filter the communication between application systems and the user model. By means of roles, different access modes can be assigned to user model clients. Roles can be ordered hierarchically, whereby the access permissions become inherited. Figure 7 shows an example in which users' interest are modeled. It includes three times three roles in two role hierarchies. The roles can be assigned as follows:

—*Interest Consumer*. To clients that retrieve information about users' interests;

—*Interest Producer*. To clients that produce assumptions about users' interests; and

—*Interest Maintainer*. To clients that both retrieve and produce assumptions about users' interests.

Each of these three roles comes in three degrees of reliability: untrusted, trusted and verified (their respective names have the suffixes _unt, _tr and _ver). Depending on the kind of role and the degree of trust, different access permissions are assigned to each role, such as for example, "ask-if" for interest_consumer_unt and "ask-all" for interest_consumer_ver. Roles can be ordered in a hierarchy. In Figure 7, interest maintainers inherit the permissions from both interest producers and interest consumers. In a second role hierarchy, trusted roles inherit the permissions of untrusted roles, and verified roles those of trusted roles.

To each application, one or more roles can be assigned. Since the management of numerous roles may be too complicated for users, experts can define typical roles, for example, for typical kinds of application systems. Figure 7 shows three such typical applications, namely a *targeted-marketing application*, a user-adaptive *news server/website*, and the *model owner*. They inherit the permissions associated with the structured roles (like *interest_consumer_unt* for *targeted_marketing_application*). Users can then assign their own applications to these application types, such as those in the example in Figure 1. It is also possible to split the role hierarchy into modules, so that several agencies (e.g., trust centers) can define them and provide their definitions to users.

For defining hierarchies conveniently, we employ and extended NIST's *RBAC/Web* system.[11] Figure 8 shows the RBAC/Web user interface in which the role hierarchy of Figure 7 is being defined on the right-hand side, and the role "interest-consumer" assigned to appl1 (the user-adaptive web site of Figure 1) on the left-hand side. Permissions for roles are represented as regular expressions. Figure 9 shows the definition of the permission that a process may submit KQML ask-if requests for the interests of User1 (i.e., the beliefs of the system about the interests of User1).

Besides representing roles of potential *clients* and their access privileges (*context*-dependent access control), *content*-dependent access control can also be achieved to account for *users'* roles [Zurfluh 1998] when interacting with a

[10]Namely, the base reference model $RBAC_0$ and $RBAC_1$ by Sandhu et al. [1996].

[11]RBAC/Web Release 1.1 `http://hissa.ncsl.nist.gov/rbac/` [Barkley et al. 1997; Ferraiolo et al. 1999].

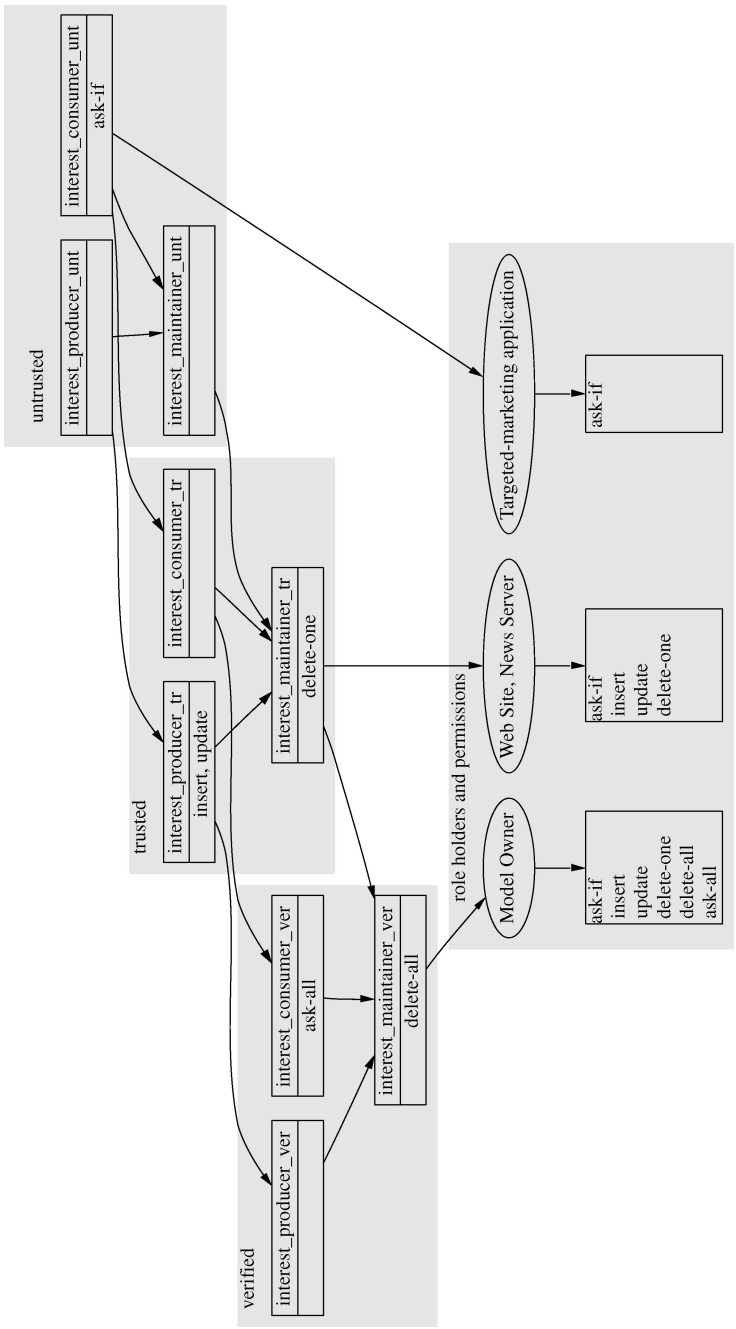

Fig. 7. Role hierarchy with permission inheritance.

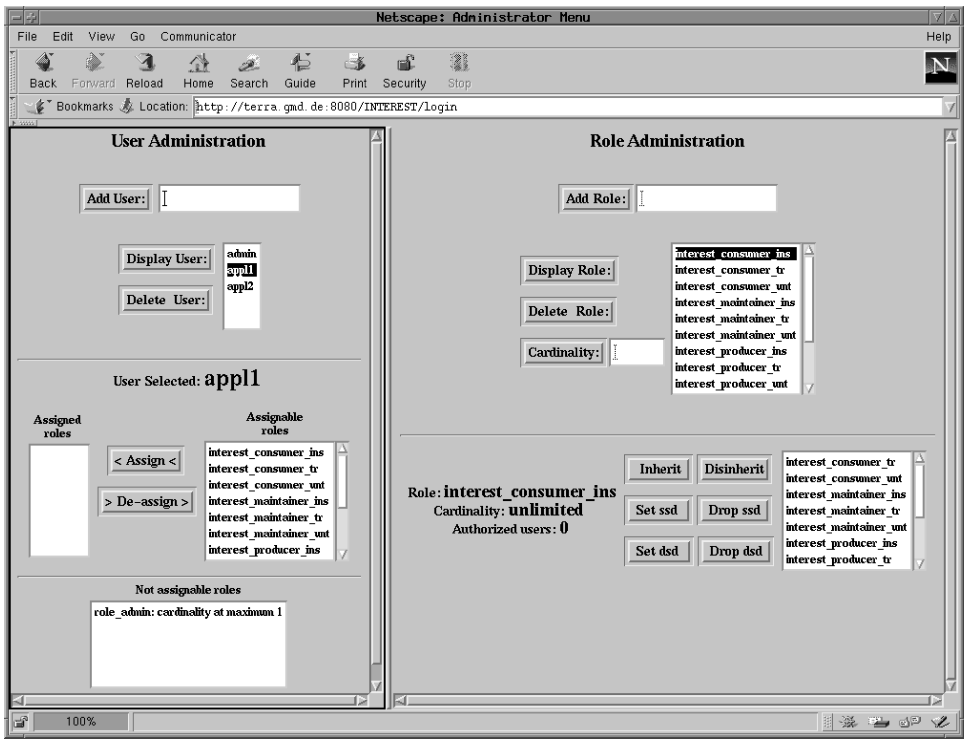

Fig. 8. RBAC/Web user interface for role definition and assignment.

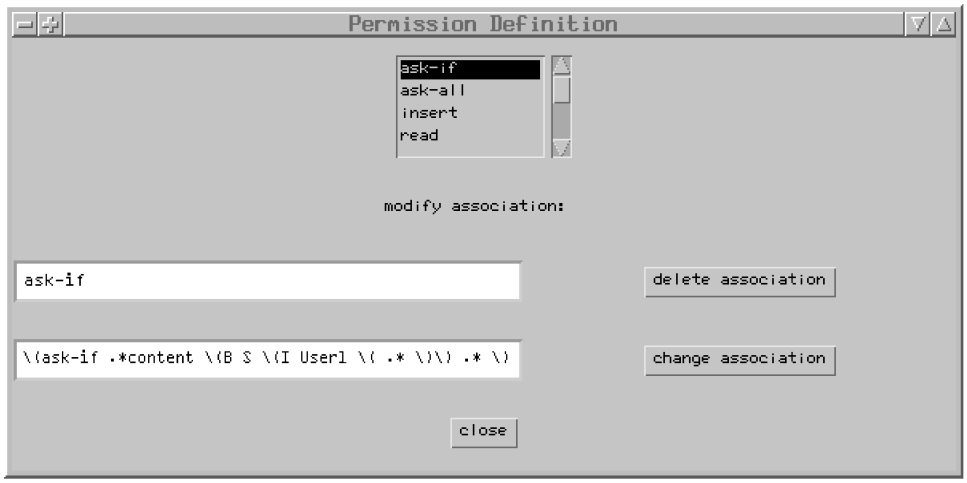

Fig. 9. Definition of an access permission.

user-adaptive system (e.g., *anonymous customer* and *identified customer*). These user roles can then be associated with certain data categories that may be revealed. The data categories can be mapped to a role hierarchy as described above. Utilizing data categories from P3P [Reagle and Cranor 1999; Cranor et al. 2002b], a role *anonymous customer* can group permissions regarding, for example, the data categories "Online Contact information," "Purchase Information," "Financial Information," "Computer Information," and "Preference Data," whereas a role *identified customer* can additionally add permissions regarding the data categories "Physical Information" and "Demographic Information."

Other advantages of RBAC include the support of basic security principles:

Separation of Duties. For a user model that is divided into anonymous and nonanonymous data, two different roles can be defined. Application systems should be assigned to one of these roles only, to prevent the linkage of anonymous and personal data.

Data Abstraction. By authorization via roles, user model clients become assigned to roles that specify access restrictions to user model entries. By using roles, the authorization can abstract from concrete user model entries. For instance, the role *interest consumer* may specify all permissible access modes to obtain information about the user's interests without specifying each individual user model entry to which access granted (see Figures 8 and 7).

4.5 A Reference Architecture for Pseudonymous and Secure Interaction with User-Adaptive Systems

In this section, the interaction of the three components described so far will be discussed, and a recommendation for positioning them between the user-adaptive application and the user model will be made. This arrangement can serve as a default architecture for pseudonymous and secure interaction. It can be modified and extended according to the requirements of the specific application.

In Figure 6, we showed how application-dependent anonymization techniques and application-independent anonymization by means of the KQMLmix component must be combined in a user-adaptive system. Figure 10 focuses on application-independent anonymization (i.e., the right-hand side of Figure 6), super-identification (see Section 3.2), encryption and authorization, and includes further components necessary for basic security services (e.g., a certificate directory; see Section 4.1).

Our reference architecture has the following components:

UM Client. The client of a user model (UM) can be one or several user-adaptive application systems, or the user. The UM clients as well as the UM communicate with the *user model reference monitor* (e.g., through TCP) and operate on a trusted computing base [Pfleeger 1989; Summers 1997].

Certificate Directory. The certificate directory contains X.509 certificates for the UM and for each UM client, which can be used to verify their identities. Public keys extracted from certificates can also be used to verify the authenticity

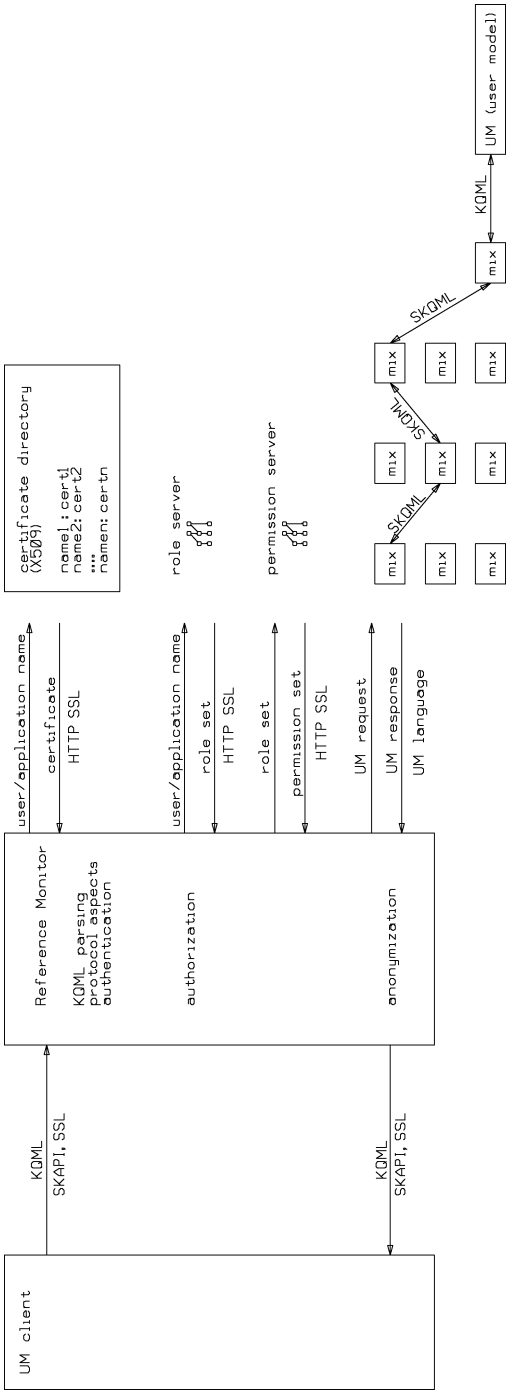

Fig. 10. Reference architecture for pseudonymity in user-adaptive systems.

of the information which is exchanged between the UM clients and the *user model reference monitor*.

Role Server. The role server provides an interface for the definition of the roles and the role hierarchy in the role-based access control model (see Figure 7). Furthermore, it manages the assignment of clients to roles. Since the role server is co-located on a common web server, communication can take place with SSL encryption and authentication.

Permissions Server. The permissions server, which is also colocated on a common web server, handles the assignment of permissions (e.g., *update*, *delete*) to roles. For a given set of roles, the server determines the set of permissions assigned to the roles or inherited from superordinate roles. For each permission in the set, the *permission definition* is ascertained and the set of permission definitions returned to the user model reference monitor. If the request of the UM client matches one of the permission definitions, the request is authorized by the user model reference monitor and can be processed.

Mix. By connecting the user model reference monitor with the user model through a mix network, procedural anonymity can be achieved. The mix network consists of KQMLmix components that provide sender and receiver anonymity.

User Model Reference Monitor. It can be placed between the UM clients and the UM (which exchange UM entries via KQML) and controls their information exchange. Since it imposes no demands on the internal mechanisms of the user modeling system, it can be applied to any KQML-enabled user-adaptive system. The user model reference monitor performs the following actions:

Parsing of KQML Messages. Messages from UM clients must be accepted and parsed.

Handling of Protocol Aspects. Messages have to be stored and be answered with the reply of the UM or with an error message.

Authentication. The sender of a message (and its content) can be authenticated through super-identification by means of certificates. Also, senders acting under a pseudonym can be authenticated if their certificate contains the pseudonym.

Authorization. The request of the UM client is checked for compliance with the definitions for access to the UM entries.

Anonymization. By routing KQML messages containing user model entries through a mix network, procedural anonymity is provided. The relationship between the user and her user model is thereby concealed.

User Model (UM). The user model only processes requests that have been authenticated, authorized, dispatched and anonymized by the user model reference monitor.

Using a mix network to isolate the user model reference monitor and the user model ensures procedural anonymity or procedural pseudonymity of the user

and the user model client towards the user model. As long as content-based and environmental anonymity also prevail, the user's identity cannot be uncovered by the user model and the user modeling system. Conversely, the mix network can also be used to hide from the UM client the location of the monitor (which can coincide with the location of the user). Both methods can be combined to hide from the user model client the user's identity as well as the location of her user model.

Since the user model reference monitor can enhance the security of user-adaptive systems without undue burden on the user modeling system, it can be applied to a wide range of components that exchange information about the user by KQML messages. For lower security demands, some components may be omitted, like, for example, encryption through SSL, the certificate directory, authorization through the access control model, or the mix network. Individual components can be provided either as software packages that must be included into user model clients (e.g., for encryption and authentication), or as third-party services (e.g., for the authorization of information requests and the anonymization of exchanged messages).

5. SUMMARY AND DISCUSSION

This article discussed requirements for secrecy in user-adaptive systems, to guarantee the anonymity of both users and user modeling servers. Giving users the option to conceal their identities seems a viable way to alleviate users' privacy concerns whilst preserving the benefits of personalized interaction. Users' trust in anonymity can moreover be expected to lead to more extensive and frank interactions, thus to more data about the user, and hence to better personalization.

A reference model for pseudonymous and secure user modeling was developed that includes a permissions server for role based access control, a mix network for hiding the identities of users and of user modeling servers, secure transport, a certificate directory, and a reference monitor that safeguards the access of user modeling clients to user models located in the user modeling server. Table IV summarizes the possible privacy threats in this architecture as well as the mechanisms to fend them off (methods listed in square brackets have not been implemented but could be integrated, as is further explained in Schreck [2003]).

The architecture imposes little demands on the internal design of user-adaptive systems (basically just that communication between user modeling components, and particularly between user modeling clients and servers, must be carried out using KQML). This should considerably facilitate the integration of existing user adaptive systems into this framework, or into parts thereof. SKAPI and KQMLmix have been implemented in C and Java and are available from the second author of this article.

Below, we outline possible future enhancements of this work.

Additional Security Mechanisms to Enhance Content-Based Anonymity. A number of additional security mechanisms can be integrated into the reference architecture to further enhance content-based anonymity. These mechanisms

Table IV. Attacker Model and Response

Type of attack / Attacker	Local area network/ wide area network	Internet service provider	User model client	User model server
Eavesdropping	Encryption (SKAPI, SKQML)		n.a.	
Unauthorized change of information	Encryption (SKAPI, SKQML)		Role based access control, signature	Authentication of content
Deanonymization by environment	n.a.			[Statistical anonymity, inference integrity]
Deanonymization by content	Encryption (SKAPI, SKQML)		Role based access control	[Filtering of content, exchange through pseudonyms]
Deanonymization by procedural apects	Procedural anonymity (KQMLmix)			n.a.
Unintended inferences	Encryption (SKAPI, SKQML)		Role based access control	[Statistical anonymity, inference integrity]
Linkage of different identities	Encryption (SKAPI, SKQML)		Role based access control, pseudonyms	Different user models
Impostering	Signature (SKAPI, SKQML)			
Repudiation	n.a.		Signature (SKAPI, SKQML)	

are generally very application dependent though and must possibly be modified with every new user-adaptive system that the user wishes to employ. For instance, the disclosure of highly identifying data to user-adaptive applications can be avoided by automatically assigning different passwords, E-mail addresses etc. for use with different applications (as, e.g., in LPWA [Gabber et al. 1999]). Or, the user modeling servers or access control mechanism can override access privileges granted by the user when this might jeopardize the user's anonymity based on additional information that is available to these components. Examples for such additional information include:

Built-In Knowledge about the Sensitivity of Certain User Data. The system can refuse that such data be stored in the user model server or accessed by certain types of applications.

Statistical Disclosure Constraints. The system can refuse to disclose user characteristics should their value not be within, for example, two sigma of the populations means.

Extensional Prediction of Possible Inferences. The system can refuse to reveal antecedents from which conclusions can be derived that the user chose not to disclose. If the user modeling server is additionally being used as a statistical database (e.g., for analyzing users' web navigation and shopping behavior), then mechanisms for statistical disclosure limitation have to be applied as well [Domingo-Ferrer 2002].

Parsimonious Access. Even when user-adaptive applications are authorized to access certain data in a user model, one may want them to do this in a frugal manner. Economical models can be employed to achieve this effect [Posner 1984]. For example, an application can be endowed with funds in a virtual currency, from which the user model deducts a fee for every request it receives from this application. User acceptance of the resulting personalization quality and success in the marketplace would be the counterweights that prevent applications from requesting insufficient or useless information about users.

Integration into User Modeling Servers. While quite a few generic user modeling systems were developed over the past ten years (see Table I for an overview of major research prototypes and commercial systems), none of them includes comprehensive security mechanisms. The addition of a security framework as presented in this article would considerably facilitate the deployment of user adaptive applications that respect users' privacy by offering anonymous yet fully personalized interaction with user-adaptive systems.

Identity Management. Identity managers [Jendricke and Gerd tom Markotten 2000; Clauß and Köhntopp 2001; Berthold and Köhntopp 2001] are envisaged high-level mechanisms that assist users in managing their privacy online. They are based on some underlying anonymous communication infrastructure, like the one that was described in this article. Identity managers would add many useful services to our architecture. They would allow users, for example, to select among anonymous, pseudonymous or indentified interaction. In the case of pseudonymous interaction, they would advise on the choice between person pseudonyms, role pseudonyms and relationship pseudonyms (see Section 3.2), and on the data that can be disclosed under each pseudonym. The latter would be directly translated into access restrictions in our reference model. Identity managers could also help users select the degree of anonymity (which would impact the minimum number of mixes in our framework), to select protection goals and to verify their compatibility [Wolf and Pfitzmann 1999].

While this article presented a comprehensive technical solution for anonymous yet fully personalized user interaction with web services, a number of obstacles must still be addressed that may complicate its deployment in practice. Hardly any readily available distributed anonymization infrastructures, such as mixes, have as yet been put in place. Anonymous interaction is currently difficult to maintain when payments, physical goods and non-electronic services are being exchanged. Anonymity on the Internet may harbor the risk of misuse and currently even seems to have an air of disreputability. Finally, web retailers also have a considerable interest in identified customer data as a business asset. While pseudonymous data would be equally helpful for an analysis of shopping behavior and customer segmentation, they cannot be used for "cross-channel" personalization (sending a web customer a targeted brochure by mail, recognizing him in a brick and mortar store and serving her individually). This becomes increasingly important since the number of web-only businesses continues to decline [BCG 2002]. The factual deployment of personalized anonymous interaction will thus strongly hinge on social factors, such as regulatory provisions

that mandate anonymous and pseudonymous access to electronic services (such as TSDP [2001] and EC [2002]), and articulated consumer demand that gives businesses offering personalized anonymous interaction a competitive advantage that outweighs its commercial downsides.

ACKNOWLEDGMENTS

We would like to thank Lorrie Cranor, Günter Pernul, Gene Tsudik, and the three anonymous TOIT reviewers for their valuable comments on an earlier version of this article.

REFERENCES

ANONYMOUS. 1996. Risks of anonymity. *Commun. ACM 39*, 12, 162.

ART TECHNOLOGY GROUP. 2001. ATG dynamo e-business platform. `http://www.atg.com/products`.

BARKLEY, J. F., CINCOTTA, A. V., FERRAIOLO, D. F., GAVRILLA, S., AND KUHN, D. R. 1997. Role based access control for the world wide web. In *Proceedings of the 20th National Information System Security Conference (NIST/NSA)*.

BELL, D. AND LAPADULA, L. 1976. Secure computer systems: Unified exposition and multics interpretation. Tech. Rep. MTR 2997. MITRE Corporation, Bedford, Mass.

BELL, D. E. 1988. Concerning "modeling" of computer security. In *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, Calif. Apr. 18–21). IEEE Computer Society Press, Los Alamitos, Calif.

BERTHOLD, O., FEDERRATH, H., AND KÖHNTOPP (HANSEN), M. 2000. Project "Anonymity and unobservability in the internet." In *Proceedings of the 10th Conference on Computers, Freedom & Privacy 2000* (Westin Harbour Castle, Toronto, Ont., Canada, Apr. 4–7), 57–68.

BERTHOLD, O. AND KÖHNTOPP (HANSEN), M. 2001. Identity management based on P3P. In *Anonymity 2000*, H. Federrath, Ed. Lecture Notes in Computer Science, vol. 2009. Springer-Verlag, Berlin—Heidelberg, Germany, 141–160.

BOSTON CONSULTING GROUP. 1997. eTRUST Internet Privacy Study: Summary of Market Survey Results. Tech. rep.

BOSTON CONSULTING GROUP. 2002. *The State of Retailing Online 5.0*. Shop.org, http://www.shop.org.

BREWER, D. F. AND NASH, M. J. 1989. The chinese wall security policy. In *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, Calif., May 1–3). IEEE Computer Society Press, Los Alamitos, Calif., 206–214.

BRUSILOVSKY, P. 1998. Methods and techniques of adaptive hypermedia. In *Adaptive Hypertext and Hypermedia*, P. Brusilovsky, A. Kobsa, and J. Vassileva, Eds. Kluwer Academic Publishers, Dordrecht, Netherlands, 1–43.

CHAUM, D. L. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM 24*, 2, 84–88.

CHIN, D. N. 1993. Acquiring user models. *Artif. Intel. Rev. 7*, 3-4, 185–197.

CLAUß, S. AND KÖHNTOPP (HANSEN), M. 2001. Identity management and its support of multilateral security. *Computer Netw. 37*, 205–219.

COVINGTON, M. A. 1998. Speech acts, electronic commerce, and KQML. *Deci. Supp. Syst. 22*, 3, 203–211.

CRANOR, L., LANGHEINRICH, M., AND MARCHIORI, M. 2002a. A P3P preference exchange language 1.0 (APPEL 1.0): W3C working draft 15 April 2002. `http://www.w3.org/tr/p3p-preferences/`, World Wide Web Consortium.

CRANOR, L., LANGHEINRICH, M., MARCHIORI, M., PRESLER-MARSHALL, M., AND REAGLE, J. 2002b. The platform for privacy preferences 1.0 (P3P1.0) specification: W3C recommendation 16 April 2002. `http://www.w3.org/tr/p3p/`, World Wide Web Consortium.

CRANOR, L. F., REAGLE, J., AND ACKERMAN, M. S. 1999. Beyond concern: Understanding net users' attitudes about online privacy. AT&T labs-research technical report TR 99.4.3, http://www.research.att.com/library/trs/trs/99/99.4/.

CSA 1996. Model Code for the Protection of Personal Information. Tech. rep., Canadian Standards Association. http://www.csa.ca/standards/privacy/code/.

CULNAN, M. J. AND MILNE, G. R. 2001. The Culnan-Milne survey on consumers & online privacy notices: Summary of responses. In *Interagency Public Workshop: Get Noticed: Effective Financial Privacy Notices* (Washington, D.C., Dec. 4, 2001). 47–54.

DENNING, D. E. 1976. A lattice model of secure information flow. *Commun. ACM 19*, 5, 236–243.

DENNING, D. E. R. 1982. *Cryptography and Data Security*. Addison-Wesley, Reading, Mass.

DEPALLO, M. 2000. AARP national survey on consumer preparedness and e-commerce: A survey of computer users age 45 and older. Tech. rep., AARP, Washington, D.C.

DOMINGO-FERRER, J. 2002. *Inference Control in Statistical Databases: From Theory to Practice*. Springer, Berlin, Heidelberg, New York.

EC 1995. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the European Communities of 23 November 1995 No L. 281*.

EC 2002. Directive 2002/58/EC of the European Parliament and of the Council of 24 October 1995 Concerning the Processing of Personal Data and the Protection of Privacy in the Electronic Communications Sector. *Official Journal of the European Communities of 31 July 2002 No L 201/37*.

FERRAIOLO, D. F., BARKLEY, J. F., AND KUHN, D. R. 1999. A role-based access control model and reference implementation within a corporate intranet. *ACM Trans. Inf. Syst. Sec. 2*, 1, 34–64.

FININ, T. AND WEBER, J. 1993. Draft specification of the KQML agent-communication language. Tech. Rep.

FININ, T. W. 1989. GUMS—A general user modeling shell. In *User Models in Dialog Systems*, A. Kobsa and W. Wahlster, Eds. Springer, Berlin, Heidelberg, New York, 411–430.

FINK, J. 2003. User modeling servers: Requirements, design, and evaluation. Ph.D. dissertation, Department of Mathematics and Computer Science, University of Essen, Germany.

FINK, J. AND KOBSA, A. 2000. A review and analysis of commercial user modeling servers for personalization on the world wide web. *User Mod. User-Adapted Interact. 10*, 3-4, 209–249. http://www.ics.uci.edu/~kobsa/papers/2000-UMUAI-kobsa.pdf.

FINK, J. AND KOBSA, A. 2002. User modeling in personalized city tours. *Artif. Intel. Rev. 18*, 1, 33–74. http://www.ics.uci.edu/~kobsa/papers/2002-AIR-kobsa.pdf.

FINK, J., KOBSA, A., AND NILL, A. 1998. Adaptable and adaptive information provision for all users, including disabled and elderly people. In *New Review of Hypermedia and Multimedia 4*. Taylor Graham, London, 163–188. http://www.ics.uci.edu/~kobsa/papers/1998-NRHM-kobsa.pdf.

FINK, J., KOENEMANN, J., NOLLER, S., AND SCHWAB, I. 2002. Putting personalization into practice. *Commun. ACM 45*, 5, 41–42.

FLINN, B. AND MAURER, H. 1995. Levels of anonymity. *J. Univ. Comput. Sci. 1*, 1, 35–47.

FORRESTER RESEARCH. 1999. The privacy best practise. Tech. rep., Cambridge, Mass.

FOX, S. 2000. Trust and privacy online: Why Americans want to rewrite the rules. Tech. rep. The Pew Internet & American Life Project, Washington, D.C.

FREIER, A. O., KARLTON, P., AND KOCHER, P. C. 1996. The SSL protocol, version 3.0. Tech. rep., Netscape Communications Company.

GABBER, E., GIBBONS, P. B., KRISTOL, D. M., MATIAS, Y., AND MAYER, A. 1999. Consistent, yet anonymous, web access with LPWA. *Commun. ACM 42*, 2, 42–47.

GABBER, E., GIBBONS, P. B., MATIAS, Y., AND MAYER, A. 1997. How to make personalized web browsing simple, secure, and anonymous. In *Financial Cryptography, Proceedings of the 1st International Conference (FC'97)* (Anguilla, British West Indies, February 24–28). R. Hirschfeld, Ed. Springer-Verlag, New York, 17–31.

GAVISH, B. AND GERDES, J. H. 1998. Anonymous mechanisms in group decision support systems communication. *Dec. Supp. Syst. 23*, 4, 297–328.

GOGUEN, J. AND MESEGUER, J. 1982. Security policies and security models. In *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, Calif., Apr. 26–28). IEEE Computer Society Press, Los Alamitos, Calif., 11–20.

GOGUEN, J. A. AND MESEGUER, J. 1984. Unwinding and inference control. In *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, Calif., Apr. 29–May 2). IEEE Computer Society Press, Los Alamitos, Calif., 75–86.

GOLDSCHLAG, D., REED, M., AND SYVERSON, P. 1999. Onion routing for anonymous and private internet connections. *Commun. ACM 42*, 2, 39–41.

GÜLCÜ, C. AND TSUDIK, G. 1996. Mixing email with BABEL. In *Proceedings of the Internet Society Symposium on Network and Distributed Systems Security (SNDSS'96)* (San Diego, Calif., Feb. 22–23). IEEE Computer Society Press, Los Alamitos, Calif., 2–16.

GVU 1998. GVU's tenth WWW user survey. Tech. rep., Graphics, Visualization, & Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, Ga. http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-10/.

HIRSCH, F. J. 1997. Introducing SSL and certificates using SSLeay. *WWW J.*, 141–173.

IBM. 1999. IBM multi-national consumer privacy survey. http://www.ibm.com/services/files/privacy_survey_oct991.pdf.

ISO. 1999. ISO/IEC 15408-2, Information technology—Security techniques—Evaluation criteria for IT Security: Part 2: Security functional requirements.

JENDRICKE, U. AND GERD TOM MARKOTTEN, D. 2000. Usability meets security: The identity-manager as your personal security assistant for the internet. In *Proceedings of the 16th Annual Computer Security Applications Conference* (New Orleans, La).

JEON, H., PETRIE, C., AND CUTKOSKY, M. R. 2000. JATLite: A java agent infrastructure with message routing. *IEEE Inter. Comput. 4*, 2, 87–96.

JERICHOW, A. 1999. Generalisation and security-improvement of mix-mediated anonymous communication. Ph.D. dissertation. Department of Computer Science, Dresden University of Technology, Dresden Germany.

KASS, R. 1991. Building a user model implicitly from a cooperative advisory dialog. *User Mod. User-Adapted Interact. 1*, 3, 203–258.

KAY, J. 1995. The um toolkit for cooperative user modelling. *User Mod. User-Adapted Interact. 4*, 3, 149–196.

KESSLER, V. 1992. On the Chinese wall model. In *Computer Security—ESORICS 92, Proceedings of the 2nd European Symposium on Research in Computer Security* (Toulouse, France, Nov. 23–25). Y. Deswarte, G. Eizenberg, and J.-J. Quisquater, Eds. Springer-Verlag, New York, 41–54.

KILIAN, J. AND PETRANK, E. 1998. Identity escrow. In *Advances in Cryptology (CRYPTO '98)* (Santa Barbara, Calif.), H. Krawczyk, Ed. Lecture Notes in Computer Science, vol 1462. Springer-Verlag, Berlin, Heidelberg, New York, 169–185.

KOBSA, A. 1993. User modeling: Recent work, prospects and hazards. In *Adaptive User Interfaces: Principles and Practice*, T. K. M. Schneider-Hufschmidt and U. Malinowski, Eds. North-Holland, Amsterdam, The Netherlands, 111–128.

KOBSA, A. 2001a. Generic user modeling systems. *User Mod. User-Adapted Interact. 11*, 1-2, 49–63. http://www.ics.uci.edu/~kobsa/papers/2001-UMUAI-kobsa.pdf.

KOBSA, A. 2001b. Tailoring privacy to users' needs (invited keynote). In *User Modeling 2001: 8th International Conference*, M. Bauer, P. J. Gmytrasiewicz, and J. Vassileva, Eds. Springer, Berlin, Heidelberg, New York, 303–313. http://www.ics.uci.edu/kobsa/papers/2001-UM01-kobsa.pdf.

KOBSA, A. 2002. Personalized hypermedia and international privacy. *Commun. ACM 45*, 5, 64–67. http://www.ics.uci.edu/~kobsa/papers/2002-CACM-kobsa.pdf.

KOBSA, A., KOENEMANN, J., AND POHL, W. 2001. Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowl. Eng. Rev. 16*, 2, 111–155. http://www.ics.uci.edu/~kobsa/papers/2001-KER-kobsa.pdf.

KOBSA, A. AND POHL, W. 1995. The user modeling shell system BGP-MS. *User Mod. User-Adapted Interact. 4*, 2, 59–106.

KOBSA, A., POHL, W., AND FINK, J. 1996. A standard for the performatives in the communication between applications and user modeling systems (draft). http://www.ics.uci.edu/~kobsa/papers/1996-kobsa-pohl-fink-rfc.pdf.

LABROU, Y. AND FININ, T. 1997. A proposal for a new KQML specification. Tech. rep., Computer Science and Electrical Engineering Department, University of Maryland, Baltimore County, Md.

MABLEY, K. 2000. Privacy vs. personalization: Part III. Tech. Rep. Cyber Dialogue, Inc. http://www.cyberdialogue.com/library/pdfs/wp-cd-2000-privacy.pdf.

MACHADO, I., MARTINS, A., AND PAIVA, A. 1999. One for all and all in one: A learner modelling server in a multi-agent platform. In *User Modeling, Proceedings of the 7th International Conference (UM99)* (Banff, Canada, June 20–24), J. Kay, Ed. Springer-Verlag, Wien, New York, 211–221.

MANNA. 2001. Frontmind. http://www.mannainc.com/products.html.

MAZIÈRES, D. AND KAASHOEK, M. F. 1998. The design, implementation and operation of an email pseudonym server. In *Proceedings of the 5th ACM Conference on Computer and Communications Security* (San Francisco, Calif., Nov. 2–5). ACM, New York, 27–36.

NET PERCEPTIONS. 2002. Net Perceptions. http://www.netperceptions.com.

NETWORK ADVERTISING INITIATIVE. 2000. Self-regulatory principles for online preference marketing by network advisers. http://www.ftc.gov/os/2000/07/nai 7-10 final.pdf.

OECD 1992. Guidelines for the Security of Information Systems. Tech. Rep. Organisation for Economic Cooperation and Development (OECD).

ORWANT, J. 1995. Heterogeneous learning in the Doppelgänger user modeling system. *User Mod. User-Adapted Interact. 4*, 2, 107–130.

PAIVA, A. AND SELF, J. 1995. Tagus—A user and learner modeling workbench. *User Mod. User-Adapted Interact. 4*, 3, 197–226.

PERSONALIZATION CONSORTIUM. 2000. Personalization & privacy survey. Tech. Rep. Personalization Consortium, Edgewater Place, Mass, http://www.personalization.org/SurveyResults.pdf.

PETRIE, C. J. 1996. Agent-based engineering, the web, and intelligence. *IEEE Expert 11*, 6, 24–29.

PFITZMANN, A. AND KÖHNTOPP (HANSEN), M. 2001. Anonymity, unobservability, and pseudonymity: A proposal for terminology. In *Anonymity 2000*, H. Federrath, Ed. Lecture Notes in Computer Science, vol. 2009. Springer-Verlag, Berlin-Heidelberg, Germany, 1–9.

PFLEEGER, C. P. 1989. *Security in Computing*. Prentice-Hall, Englewood Cliffs, N.J.

POHL, W. 1998. *Logic-Based Representation and Reasoning for User Modeling Shell Systems*. Academic Publishing Corporation—Infix, Berlin, Germany.

POSNER, R. A. 1984. An economic theory of privacy. In *Philosophical Dimensions of Privacy*, F. Schoeman, Ed. Cambridge University Press, 333–345.

RAMAKRISHNAN, N., KELLER, J. B., MIRZA, B. J., GRAMA, A. Y., AND KARYPIS, G. 2001. Privacy risks in recommender systems. *IEEE Internet Comput.* Nov.-Dec., 54–62.

REAGLE, J. AND CRANOR, L. F. 1999. The platform for privacy preferences. *Commun. ACM 42*, 2, 48–55.

REITER, M. K. AND RUBIN, A. D. 1998. Crowds: Anonymity for web transactions. *ACM Trans. act. Inf. Syst. Sec. 1*, 1, 66–92.

REITER, M. K. AND RUBIN, A. D. 1999. Anonymous web transactions with crowds. *Commun. ACM 42*, 2, 32–38.

RICH, E. 1979a. Building and exploiting user models. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI'79)* Vol. 2 (Tokyo, Japan, Aug. 20–23). 720–722.

RICH, E. 1979b. User modeling via stereotypes. *Cogn. Sci. 3*, 329–354.

RICH, E. 1983. Users are individuals: Individualizing user models. *Int. J. Man-Machine Stud. 18*, 199–214.

ROTENBERG, M. 2002. *The Privacy Law Sourcebook 2002*. Electronic Privacy Information Center, Washington DC.

SANDHU, R., COYNE, E. J., FEINSTEIN, H. L., AND YOUMAN, C. E. 1996. Role-based access control models. *IEEE Comput. 2*, 2, 38–47.

SCHAFER, J. B., KONSTAN, J. A., AND RIEDL, J. 2001. E-commerce recommendation applications. *Data Min. Knowl. Disc. 5*, 1, 115–153.

SCHRECK, J. 2003. *Security and Privacy in User Modeling*. Kluwer Academic Publishers, Dordrecht, Netherlands. http://www.security-and-privacy-in-user-modeling.info.

SIMMONS, G. J. 1992. *Contemporary Cryptology, The Science of Information Integrity*. IEEE Press, Los Alamitos, Calif.

SLEEMAN, D. 1985. UMFE: A user modelling front-end subsystem. *Inter. J. Man-Machine Stud. 23*, 71–88.

SPIEKERMANN, S., GROSSKLAGS, J., AND BERENDT, B. 2001. E-privacy in 2nd generation e-commerce: Privacy preferences versus actual behavior. In *EC'01: Proceedings of the 3rd ACM Conference on Electronic Commerce*. ACM, New York, 38–47.

SUMMERS, R. C. 1997. *Secure Computing, Threats and Safeguards*. McGraw-Hill, New York.

SYVERSON, P. F., GOLDSCHLAG, D. M., AND REED, M. G. 1997. Anonymous connections and onion routing. In *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, Calif., May 4–7). IEEE Computer Society Press, Los Alamitos, Calif., 44–54.

TSDP 2001. Act on the protection of personal data used in teleservices (Article 3 of the law on the legal requirements for electronic business dealings of 14 Dec. 2001). German Federal Law Gazette 1, 3721. `http://www.iid.de/iukdg/aktuelles/fassung_tddsg_eng.pdf`.

WARREN, S. D. AND BRANDEIS, L. D. 1890. The right to privacy. *Harvard Law Review IV*, 5, 193–220.

WESTIN, A. F. 1970. *Privacy and Freedom*. Atheneum.

WESTIN, A. F. AND MAURICI, D. 1998. E-commerce & privacy: What net users want. `http://www.pwcglobal.com/gx/eng/svcs/privacy/images/e-commerce.pdf`.

WOLF, G. AND PFITZMANN, A. 1999. Empowering users to set their security goals. In *Multilateral Security for Global Communication—Technology, Application, Business*, G. Mller and K. Rannenberg, Eds. Addison-Wesley-Longman, Berlin—Heidelberg, Germany, 113–135.

ZURFLUH, U. E. 1998. People, their Roles, and their Action Potential in Electronic Environments (in German). In *Sicherheit in Informationssystemen, Proceedings der Fachtagung SIS'98, Universität Hohenheim, Germany, 26.-27.3.*, K. Bauknecht, A. Büllesbach, H. Pohl, and S. Teufel, Eds. Hochschulverlag AG ETH Zürich, 43–64.

Received October 2001; revised June 2002 and September 2002; accepted November 2002