THE PRIVACY ADVOCATE (PRIVAD)
A Framework for Negotiating Individual Privacy Contracts

Michael Maaser, Steffen Ortman and Peter Langendörfer
IHP microelectronics GmbH, Im Technologiepark 25, 15236 Frankfurt (Oder), Germany

Keywords: Privacy, negotiation, P3P, privacy contract, user preferences, personalization, mobile devices.

Abstract: Privacy has been a hot topic in research for several years. A lot of different approaches to protect privacy have been proposed recently. Among these are several tools for negotiation of privacy contracts. In this paper we present our privacy negotiation framework called "Privacy Advocat (PrivAd)". It consists of three main parts: the policy evaluation unit, the signature unit and the preferences. In addition our frame supports an interface for negotiation strategies, so that they are independent of the framework. The preferences can be expressed with a combination of P3P and APPEL. The test we executed using a state of the art PDA clearly indicate that our framework can be used by mobile devices. The completion of a negotiation takes about 2 sec. including message transfer via an 802.11b wireless link. The processing itself is done in less than 250 msec.

1 INTRODUCTION

A lot of internet users are concerned about their privacy (Cranor, 2000) . These concerns have led to the development of P3P (Cranor, 2006) and APPEL (Cranor, 2002) by the W3C. P3P provides means to service providers to express which data is gathered by the service provider as well as for which purpose etc. APPEL provides means to service users to express their preferences. The P3P approach is a pure opt in or opt out model (Thibadeau, 2000)(Bennicke, 2003). Thus, if the user preferences and the service provider policy do not fit to each other the only option of the user is not to use this service. The drawbacks of such a static approach are on both sides. On one hand at least some service users have to give up more privacy than what they would prefer or are excluded from using interesting services. On the other hand service providers have to balance their requirements against the user demands for privacy. A recent study (Druecke, 2006) shows that a significant percentage of users is willing to provide additional data if the service provider offers incentives like free trial version of software, reduced prices for the service under negotiation etc. Thus, privacy negotiation tools can help to accommodate the needs of both parties, i.e. more data for service providers, better prices or better privacy for certain groups of users.

Privacy negotiation tools should provide service providers and service users with means to express their own requirements. Note by the term negotiation, we really mean a kind of bargaining about the content of the resulting privacy contract. Unlike other tools or protocols the requirements do not specify a list of alternatives which can be chosen from exclusively. Our approach describes requirements as negotiation intervals by stating their boundaries but support use of discrete values as well. We made both negotiation parties capable to offer counterproposals based on those intervals and former offers. Such intervals are especially useful for measurable data such as charges and position information.

In this paper we present our privacy negotiation framework, named Privacy Advocate (PrivAd). PrivAd provides means for service providers as well as users to negotiate individual privacy contracts. Thereby PrivAd is fully interoperable with already existing P3P based infrastructure. Existing tools such as Privacy Bird (Privacy Bird, 2006) can effectively communicate with servers using PrivAd as well as PrivAd clients can process static P3P policies from regular web servers.

This paper is structured as follows. Section 2 provides a short state of the art. The extensions of P3P that are needed to enable negotiations are briefly discussed in section 3. In the following section we present details of PrivAd and section 5

Maaser M., Ortman S. and Langendörfer P. (2007).
THE PRIVACY ADVOCATE (PRIVAD) - A Framework for Negotiating Individual Privacy Contracts.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Internet Technology*, pages 88-95
DOI: 10.5220/0001282100880095

outlines its implementation. The paper concludes with a summary and an outlook on further research.

2 RELATED WORK

There are several privacy related tools that are based on P3P and APPEL specifications. AT&T's Privacy Bird is a free plug-in for Microsoft Internet Explorer. It allows users to specify privacy preferences regarding how a website collects and stores data about them. When the user visits a website, the Privacy Bird analyzes the provided policy and indicates whether or not the policy matches the user's preferences. The Microsoft Internet Explorer 6 and Netscape 7 embed a similar behavior. They allow users to set some options regarding cookies and are capable of displaying privacy policies in human readable format. PAWS also relies on P3P and APPEL but allows discovery of privacy policies within ubiquitous environments. The policy evaluation is not done on the mobile device (Langheinrich, 2002). All these tools are a valuable step into the right direction, but they still lack means to individualize privacy policies.

There are approaches that allow users to choose from a given number of alternatives presented by the service provider (Preibusch, 2005) (El-Khatib, 2003). Such behavior does not characterize a real negotiation to us. These approaches rely on the fact that a non empty intersection of the user requirements and the policies offered by the service provider is given a priori. Such approaches are widely used in protocols such as SSL and TLS. In contrast to this we consider both negotiation opponents as equal and they both are enabled to offer proposals and counterproposals. (Yee, 2004) presents an approach for negotiating privacy policies using negotiation trees. This is similar to our approach but the prototype is not capable of automatic negotiation.

3 NEGOTIATION EXTENSIONS

To enable negotiation scenarios the following documents are needed: Service and user preferences that describe the respective requirements, proposals and counterproposals from preferences and finally a mutually agreed privacy contract as the outcome of a successful negotiation.

Preferences are secret policies of each party that are never exposed to the opposite side. The secret preferences specify boundaries that span a certain room to negotiate. These boundaries do not only define certain values or enumerations but can specify also single- or double-bounded intervals of real numbers or any other value. Surely unique requirements of users and service provider have to be considered too. In order to enable individualization a user has to state permissions and prohibitions whereas the service providers state requirements and facultative requests. This is done in the secret preferences of each party by respective tags that are added to the statements.

Proposals or counterproposals contain complete envisioned contracts or parts of those. Further they may contain requests for deletion of certain parts. Such documents are exchanged alternately during the negotiation process. A privacy contract enumerates all data, purposes, recipients, etc. whose release has been agreed on by both opponents during the negotiation. Privacy contracts must not contain any interval. All values have to be stated explicitly.

Our vision of a privacy contract is mostly resembled by a P3P policy. Therefore we decided to enhance the P3P standard with different language features to enable negotiation possibilities. Since P3P does not support a description of intervals as needed for the preferences, we added tags to describe those intervals first. This is done by specifying boundaries with tags like `<atleast>` and `<atmost>`. Boundaries can be set for measurable data i.e. accuracy of location. For remuneration of service usage we integrated a special element to represent the amount of money to be paid for a service. Besides the necessary interval it contains unit and currency tags to enable respective conversion. Due to space limitation we omitted detailed description of all extensions. Please refer to (Maaser, 2006).

3.1 Permissions and Prohibitions

While requirements and facultative request at service provider preferences are self-explanatory, the permissions and prohibitions in user preferences contain additional implications. Users are able to define statements in their preferences, which are explicitly prohibited or allowed. In this way users are capable to prohibit single data or groups of data for certain purposes, recipients or retentions. It is also possible to prohibit data or groups of data for all purposes or all recipients etc. That is, prohibitions specify domains that are explicitly prohibited. Preferences that are solely composed of prohibitions, implicitly allow all non-stated data.

Similarly, permissions explicitly allow data for certain purposes or recipients. If user preferences contain permissions only, all non-stated data are implicitly prohibited. Obviously, explicitly allowed

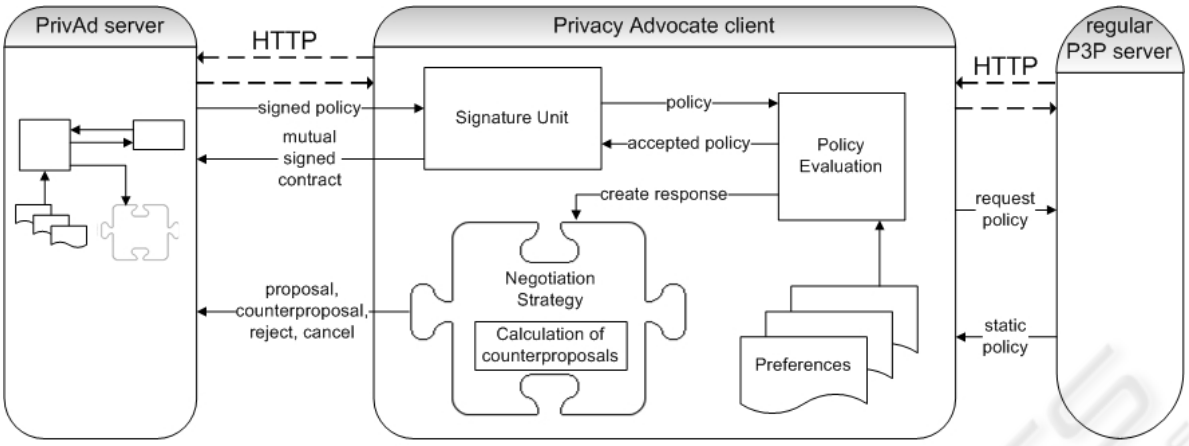

Figure 1: Structure of the Privacy Advocate client and server.

or prohibited content of statements is more significant than comparable implicit ones.

Successful negotiations finish with a mutually agreed policy, called contract. The basis for the used signatures are digital certificates analogous to those used by SSL. Since a contract is an individual document regarding each pair of user and service provider it must be digitally signed by both parties. The P3P standard does not provide an opportunity to integrate digital signatures. Thus, to ensure backward compatibility, we use xml comments to integrate the digital signatures. This way we avoid misinterpreted policies by standard P3P applications. Our negotiation protocol is not capable to choose from a list of signature algorithms, analogue to SSL, yet. Thus, both negotiation parties have to use a predefined signature algorithm. Currently this is fixed to SHA/DSA.

4 PRIVACY ADVOCATE

The Privacy Advocate (PrivAd) is a framework that is capable to negotiate on the basis of P3P privacy policies including the introduced P3P extensions. It is designed for both negotiation parties and may be used by service users as well as the service provider side. Hence we implemented a PrivAd client and a PrivAd server. The framework consists of a Policy Evaluation Unit, a Signature Unit, a Repository for preferences and signed policies, mechanisms for http communications and an interface for exchangeable Negotiation Strategies, see Figure 1. The graphical user interface of the PrivAd client allows the user to monitor and control the negotiation process and was implemented for demonstration purposes only. Surely an embedded Version of PrivAd does not need such a graphical user interface and works unseen in background. The Signature Unit verifies signatures of incoming documents and signs outgoing ones. The Policy Evaluation Unit is the core of PrivAd and the negotiation concept. Every incoming proposal or policy has to be evaluated and compared with the preferences by this unit. The last component, the Negotiation Strategy, is responsible for calculating counterproposals. It is not a part of the framework. We intentionally made this part exchangeable to provide means to use alternative strategies. Exemplary we implemented two strategies described in Section 5.5, but the framework could be be completed with any Negotiation Strategy.

To create a negotiation enabled antagonist we embedded PrivAd into a Tomcat Web Server. To exchange negotiation messages between two negotiation parties or to collect a policy of a static device, the transport protocol http is used. Our Privacy Negotiation Protocol (PNP) used by PrivAd is a stop-and-go protocol in which each incoming policy triggers its evaluation and sending of an answer. Figure 1 displays the interaction of a PrivAd client with a PrivAd server or a regular P3P enabled web server respectively.

4.1 Negotiation Process

This section describes a single turn of the bargaining procedure, when a PrivAd client receives a policy as a proposal or counterproposal from a negotiation enabled service provider using PrivAd server. The Signature Unit verifies the signature of any received policy first. Incorrect signatures lead to stop the negotiation immediately. Policies that contain correct signatures are forwarded to the Policy Evaluation Unit that compares the content of the policy with the client's preferences. The Policy

Evaluation Unit returns a Boolean value that states whether or not the received policy matches the client's preferences. In case of a match PrivAd has 2 options:

1. PrivAd signs the policy and sends it back to the service. If the incoming policy was already single signed both sides have a mutually signed contract now and may continue with fulfilling the service.
2. PrivAd orders the Negotiation Strategy to create a counterproposal although the policy was acceptable. The Negotiation Strategy may create a counterproposal that contains more favorable terms and conditions for the client's side which is then send back.

In case of a mismatch with own preferences, PrivAd has 3 options to response:

1. PrivAd cancels the negotiation because the policy was unacceptable.
2. PrivAd orders the Negotiation Strategy to create a counterproposal that matches own preferences and returns it to the sender.
3. PrivAd cancels the negotiation because an integrated assessment function indicates that the negotiation process does not seem to advance and become successful.

Note that the Policy Evaluation Unit and the Negotiation Strategies are described in separate subsections later. The PrivAd server handles incoming proposals analogous.

To provide more flexibility, PrivAd uses an independent Negotiation Strategy to create counterproposals. As seen above, PrivAd does not need a strategy to evaluate incoming documents but for appropriate responses only.

It is possible that a complete bargaining needs many negotiation rounds. Therefore PrivAd is capable to store and adjust former proposals if necessary. Thus, counterproposals do not need to contain the complete policy. Only changes need to be submitted. If a negotiation party agrees to the proposal with all changes applied, a new contract proposal containing all involved statements is generated. That contract proposal is signed and send to the opposite side.

Finally the case to «negotiate» with regular P3P enabled web servers has to be considered, see section 5.1 for further details. PrivAd sends no document to such server but receives their static P3P privacy policy. This policy is evaluated by the Policy Evaluation Unit as usual. PrivAd client indicates the user about the acceptance of the policy via a message on the screen. Due to the static server no counterproposal is calculated in case of a mismatch with own privacy requirements. So, PrivAd provides the same functionality as existing P3P tools.

In the case of especially tenacious strategies it may occur that the negotiation will never find an agreement, despite such agreement theoretically exists. These cases cannot be prevented by design. Thus PrivAd needs to provide an assessment function that allows to monitor progress of the negotiation as well as to calculate, whether a negotiation should be cancelled. The assessment function is independent of the strategies. Our current assessment function accounts the percentage of accepted statements to check the convergence of the negotiation process. If this percentage of the current negotiation round is less than the minimum of last five negotiation rounds, PrivAd cancels the current negotiation. In principle each user may define an own assessment function or adjust the number of negotiation rounds of the currently used function.

4.2 Policy Evaluation

This section describes the procedure of checking a proposal against negotiation preferences. It is essential that every incoming document can be checked automatically. Therefore an algorithm and a data-model were developed which will determine acceptance of a document in relation to own preferences. Surely these documents have to conform to P3P with or without the introduced extensions.

As stated before, the two possible results of this check are the acceptance or denial of a proposal. For further explanations the following naming conventions are constituted. Statements of arrived proposals are called paragraphs and statements of preferences are called directives. The policy evaluation passes three different stages starting with the evaluation of individual elements of a statement up to the complete proposal.

4.2.1 Elements of Statements

The paragraphs and directives are interpreted as vectors of sets. These sets are purposes, recipients and data_group. Thus, elements of a statement are represented as follows.

```
statement =
({}purposes,{}recipients,{}data_group)
```

Depending on the use of the statement the sets may also be Ω. Ω specifies all possible members of a set, i.e. all purposes that are imaginable. Ω may be used in the user preferences to prohibit or allow data without referencing a certain purpose or recipient. Hence, it can be assumed that each element of a statement, which works as a paragraph, is neither empty nor Ω. Obviously empty sets or Ω are allowed

to be used in directives only. Considering paragraphs and directives as vectors of sets, just these sets have to be compared. That is, we compare purposes from paragraphs with purposes from directives just etc. We call sets from directives with permissions permitted sets and sets from directives with prohibitions prohibited sets. The result of this comparison is one of the following logical values.

`not affected` → The compared sets are disjoint.

`definitely permitted` → The set of the paragraph is a subset of the permitted set. Hence it allowed.

`possibly prohibited` → The intersection of the paragraph's set and the prohibited set is not empty but the paragraph's set contains elements that are not elements of the prohibited one. So final decision whether or not the paragraph can be accepted requires additional calculations, see section 4.2.2.

`possibly permitted` → The intersection of the paragraph's set and the permitted set is not empty but the paragraph's set contains elements that are not elements of the permitted one. So further processing is required analogous to `possibly prohibited `.

`definitely prohibited` → The set of the paragraph is a subset of the prohibited set. Thus it must be prohibited.

Each element gets one such value assigned and is further processed in the next steps to finally determine the acceptance of the proposal.

Table 1: Logical values of sets in paragraphs to the respective sets in directives.

Directive is		permit	prohibition / $\{\}_+$	prohibition / $\{\}_\&$
Ø ≠ D ≠ Ω	§ ∩ D ≠ Ø, § ∩ D ≠ §, § ∩ D ≠ D	POSSIBLY PERMITTED	DEFINITELY PROHIBITED	POSSIBLY PROHIBITED
	§ ∩ D = Ø	NOT AFFECTED	NOT AFFECTED	NOT AFFECTED
	§ = D	DEFINITELY PERMITTED	DEFINITELY PROHIBITED	DEFINITELY PROHIBITED
	§ ⊂ D	DEFINITELY PERMITTED	DEFINITELY PROHIBITED	POSSIBLY PROHIBITED
	§ ⊃ D	POSSIBLY PERMITTED	DEFINITELY PROHIBITED	DEFINITELY PROHIBITED
D = Ω		DEFINITELY PERMITTED	DEFINITELY PROHIBITED	DEFINITELY PERMITTED
D = Ø		POSSIBLY PERMITTED	DEFINITELY PERMITTED	DEFINITELY PROHIBITED

For easier notation the examined set from the directive is D and the respective set from the paragraph is §. According to this notation a lookup table, see Table 1, is used to determine logical values. Just see the headline to define the kind of directive and the first column to assign the set of the paragraph to determine the result. For determination of logical values it must be taken into account whether a directive is a prohibition or a permit. Since prohibitions forbid the combination of certain data for any of the stated purposes or recipients, we have to distinguish OR-sets $\{\}_+$ and AND-sets $\{\}_\&$. While OR-sets prohibit all possible combinations of the contained elements, an AND-set only prohibits the given combination of the set. Thus, the OR-set is the more restrictive one. In permits this classification of sets is not necessary because every subset of them is also permitted.

4.2.2 Grading of Paragraphs

After determining the logical values of sets in paragraphs, the next step of evaluation has to be done. The logical values of sets in a paragraph have to be concatenated to determine the logical value of the complete paragraph in relation to the directive. We use two additional lookup tables to define this concatenation operation. One lookup table is responsible for prohibiting directives and the other one for permits. Example: A user prohibits the combination of her data A and B for the purpose P by a prohibiting directive. A service requests data A for the purposes P and Q. According to Table 1 the requested data is `possibly prohibited` and the purposes are `definitely prohibited`. These logical values are concatenated resulting in the value `possibly prohibited`. Hence, the paragraph is acceptable because it requests not all of the prohibited data.

A special focus lies on paragraphs, which got the logical values `possibly permitted` or `possibly prohibited`. All paragraphs which have been evaluated to those logical values have to be analyzed again. Please notice that both values are not equivalent. In the following we provide an example for `possibly prohibited` paragraphs. Suppose a directive with prohibited data {name, address, birth date}. The combination of this data could be misused to falsify an eBay account. Additionally assume a proposal containing two paragraphs. One paragraph requests the name and address, the other one requests the birth date. Both paragraphs are evaluated to `possibly prohibited` (refer Table 1) because the requested data_groups are only subsets of the prohibition {name, address, birth date}. So these individual paragraphs would still be acceptable, despite their combination is violating the directive. To avoid such cases, all `possibly prohibited` paragraphs are merged as long as possible in order to determine effectively, whether they are `definitely prohibited` or not.

Thus, these two paragraphs have to be merged into one paragraph if possible. Two paragraphs that differ in one set only, can be merged to a single paragraph where the differing sets are conjoint. In our example the merged paragraph contains data {name, address, birth date} after finishing the merge operation and is checked again. Obviously this turns out to be `definitely prohibited`. Consequentially the logical values of both original paragraphs are changed to `definitely prohibited`.

4.2.3 Aggregation of Acceptance of Proposal

After the logical values for all paragraphs of a proposal are determined the results have to undergo another evaluation step. It is checked whether a certain paragraph was evaluated to `definitely prohibited` or `possibly permitted`. If such a paragraph exists in the proposal, the Policy Evaluation Unit signals the non-acceptance of the proposal. Otherwise it could be agreed to. In (Maaser, 2005), the used data-model and the policy evaluation are defined in more detail.

4.3 Protocol Issues

To exchange negotiation documents we used http as transport protocol. This protocol is widespread and ensures backwards compatibility with P3P. PrivAd uses the http-post method to pick up and send proposals or policies to negotiation enabled servers. If PrivAd detects no authorization to send an http-post request towards the negotiation opponent, a regular P3P enabled web server is assumed. Thus PrivAd will get the static policy, if available, by using the http-get method. That enables PrivAd to verify every existing static P3P policy.

Also our PrivAd server responds with a standard policy at receiving an http-get request. That enables the server to answer requests of clients that are unable to negotiate such as most existing P3P tools. If an http-post request is received, the PrivAd server supposes a negotiation enabled client on the other side. Therefore PrivAd client sends an initial request to negotiation enabled server by http-post to initiate the negotiation process. This post request can be empty or contain a proposal.

5 IMPLEMENTATION

5.1 Mobile PrivAd Client

The main restriction while implementing PrivAd client was to ensure that mobile devices can use it. Therefore the PrivAd client was implemented compatible to JDK 1.1.8. We use the JVM CrEme (CrEme, 2004) to run PrivAd on PDA's. The PrivAd client PDA version contains a graphical user interface to visualize the negotiation process on the PDA for plausibility checks and demonstration. In this GUI every transferred document and the preferences of the client are shown. The user has to choose a file with preferences and a URL as a target for negotiation. Pushing a button to run the next negotiation round enables the user to control the progress of the negotiation process. All policy evaluation and proposal generation is done by the PrivAd client. Note the code size take about 150 Kbytes only.

The privacy negotiation process was tested on a Toshiba Pocket PC e740 WLAN with a 400 MHz xscale processor and 64 MB RAM. For this test we used very simple policies that were compiled with less than 4 statements on the PrivAd server and the mobile PrivAd client. For the air link we used 802.11b with a nominal data rate of 11 Mbit/sec. We measured results for accepted as well as for rejected policies. Comparing the results for different transmission speeds of the network shows clearly that the communication time dominates the time needed to complete the negotiation process. Using a wired connection the complete negotiation cycles were finished in less than 300 msec whereas it took about 2000 msec using the wireless connection. The pure calculation of proposals and counterproposals takes less than 250 msec on the mobile PrivAd client.

5.2 PrivAd Server

The server side is implemented as a Java Servlet that runs in a Servlet container. For our reference application we used Apache Tomcat (Tomcat, 2005). Similar to its client the PrivAd server is backward compatible to existing P3P tools. Therefore the server is able to behave like a regular P3P enabled web server, if it is necessary. On a first request the PrivAd server will respond with its initial policy. If it is an http-post request, the PrivAd server initiates the negotiation process with this client. By receiving a simple http-get request the server only replies with a default policy without any negotiation interests which is fully compatible to P3P standard. Hence the PrivAd server works interoperable with already existing tools like Privacy Bird, Internet Explorer or Netscape.

Using http-post for policy exchange may inhibit options for invaders. This carries limited risk, because incoming messages are parsed and validated against the PrivAd schema. In worst case there will be an error accounted by parsing the element. Since all policies are interpreted as text and not as

executables, the risk, that malicious code can be executed in system, is kept minimal.

5.3 Digital Signature

All contracts have to be signed digitally. This will block man-in-the-middle-attacks, give proof of identity and offers more integrity. Incorrect signatures lead to cancellation of the negotiation process. In addition the Signature Unit signs an acceptable policy to generate a single signed contract proposal. Accepted contract proposals from the opponent are signed to generate a mutually signed contract. For digital signing we implemented the SHA/DSA algorithm in the Signature Unit. We rely on existing PKI as SSL/TLS also does. For future versions we intend to integrate selection means for different signature algorithms.

5.4 Negotiation Strategies

Creation of a counterproposal depends on the Negotiation Strategy in PrivAd and its secret preferences. Simple Negotiation Strategies, for both negotiation parties are developed and integrated. For demonstration we implemented two strategies where both sides try to find a mutual contract by accommodating as fast as possible. To start we point out operation methods of a simple service provider strategy for PrivAd server. Consequently, the service provider creates and proposes an initial policy, which contains all requirements and facultative requests. Additionally, most profitable values from the intervals in the preferences are chosen in this policy to gather a maximum of information and money. On receiving a counterproposal it is applied to previous proposals to get a combined proposal and the provider strategy has to react in a more complex way:

- The strategy acquires the logical values of all statements that the Policy Evaluation Unit has determined before. Logically statements that are neither explicitly nor implicitly prohibited do not require a counterproposal.
- All acceptable statements in the proposal are mapped to respective statements in the preferences.
- If all requirements are satisfied already, the strategy creates a new contract proposal from the combined proposal. All additionally acceptable statements are surely added to gain as many data as possible. The PrivAd server signs it and sends that new contract proposal back to the user.
- If not all requirements are satisfied, a new contract proposal or counterproposal has to be created and proposed. Therefore all requirements are requested.
- Unacceptable proposed statements are updated by new values matching the intervals in the preferences. Here the strategy may reduce the proposed charge for the provided service by 10 percent or similar.
- If proposals instruct to discard statements that are requirements, the provider strategy proposes exactly these statements again since otherwise the service cannot be provided. However, the service provider never signs a policy, where not all minimum requirements from its preferences are included.
- The created counterproposal is passed to PrivAd server that sends it back to the opponent.

While service providers normally try to retrieve the highest possible amount of data, the users naturally negotiate in the opposite way.

- The user gets an initial policy by querying the service provider at first time.
- The PrivAd client checks the signature. Proposals are applied to previous ones if those exist. Then the acceptance of the combined proposal is evaluated. Thereby the Policy Evaluation Unit marks all statements in the proposal with logical values to determine prohibited ones.
- If the proposed policy does not contain any prohibited statement this policy is signed by the Signature Unit and is returned. If this policy was a contract proposal, it is also signed by the PrivAd client and becomes a mutually signed contract. In that case both sides have a mutually signed contract now and may start using the service. Surely also an acceptable proposal could be counterproposed here. Thereby the strategy may cut the prizes for the service or adapt the proposed values to improved ones.
- If it contains prohibited statements the strategy component generates a counterproposal. Prohibited paragraphs are the base for calculating the counterproposal. Below we display an operation breakdown on a single prohibited paragraph. As mentioned before, statements in preferences are called directives whereas statements in proposals are called paragraphs.
    - First, the prohibiting directives for this paragraph are searched in the client's preferences.
    - Special focus is set on merged paragraphs. Certainly all prohibited paragraphs that were merged into one single paragraph have to be rejected in their isolated form as well. Since all paragraphs are numbered a rejection can be accomplished by returning an empty paragraph with that particular number.

  - Finally, the prohibited paragraph is intersected with the prohibiting directive. Here intersection means that all values of the paragraph are adjusted to match the intervals in the directive. The strategy may try to approximate the values of the counterproposal to the prefered boundary of the interval.
  - If the strategy has still not found an appropriate intersection, the prohibited paragraph has to be rejected without a replacement.
  - This process is repeated for every prohibited paragraph of the unaccepted policy.
- Last step is to create a counterproposal containing all changed and newly generated paragraphs. All paragraphs, that were not listed or explicitly rejected in counterproposal are supposed to be accepted.
- The created counterproposal is passed to the PrivAd client that sends it to the opponent.

6 SUMMARY & OUTLOOK

This article determined the situation in stating and contracting privacy in the Internet. Lacking negotiation capability has been identified as a shortcoming. A privacy negotiation framework that allows mobile users and service providers to negotiate about data and its potential recipients, purpose etc. was presented. This framework PrivAd is downwards compatible with P3P and enables negotiation scenarios between PrivAd clients and static servers as well as between known P3P tools and PrivAd servers. The process of evaluating a privacy policy and the calculation of counterproposals was explained. Tests, done with a state of the art mobile device, prove that our mobile PrivAd client is functional. Successful negotiations, finishing with a mutually signed privacy contract, take about 2 seconds via wireless link and about 0.3 seconds via a wired link, including all message transfer. In the future the two negotiation parties and their types of requirements shall be merged, in order to create tools which can perform both sides of negotiation at a time. This will enable negotiations between enterprises and in the peer-to-peer domain.

REFERENCES

Bennicke, Langendörfer, 2003. *Towards Automatic Negotiation of Privacy Contracts for Internet Services.* Proceedings of 11th IEEE Conference on Computer Networks, IEEE Society Press.

Cranor, 2000. *Beyond Concern: Under-standing Net Users' Attitudes About Online Privacy*. Ingo Vogelsang and Benjamin M. Compaine, eds. The Internet Upheaval: Raising Questions, Seeking Answers in Communications Policy. Cambridge, Massachusetts: The MIT Press, p. 47-70.

Cranor, Langheinrich, 2002. *A P3P Preference Exchange Language 1.0.* http://www.w3.org/TR/P3P-preferences/.

Cranor, Langheinrich, Marchiori M., 2006. *The Platform for Privacy Preferences 1.1 (P3P1.1) Specification.* http://www.w3.org/TR/P3P11/.

CrEme, 2004. CrEme version 3.25, NSICOM, www.nsicom.com.

Druecke, 2006. *Attitudes to Privacy at using mobile phones (in german)*, Technical Report, "Mobile Internet Business", Nr. 3, ISSN 1861-3926.

El-Khatib, 2003. *A Privacy Negotiation Protocol for Web Services*, Workshop on Collaboration Agents; Autonomous Agents for Collaborative Environments Halifax, Nova Scotia, Canada.

Langheinrich, 2002. *A Privacy Awareness System for Ubiquitous Computing Environments*. In: Gaetano Borriello, Lars Erik Holmquist (Eds.): 4th International Conference on Ubiquitous Computing (Ubicomp 2002), LNCS No. 2498, Springer-Verlag, pp. 237-245.

Maaser, Langendoerfer, 2005. *Automated Negotiation of Privacy Contracts* 29th Annual International Computer Software and Applications Conference (COMPSAC 2005). Edinburgh, Scotland, UK. IEEE Computer Society

Maaser, Ortmann, Langendoerfer, 2006. *NEPP: Negotiation Enhancements for Privacy Policies*, W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement. Ispra, Italy.

Preibusch, 2005. *Implementing Privacy Negotiation Techniques in E-Commerce,* 7th IEEE International Conference on E-Commerce Technology, IEEE CEC 2005, Technische Universität München, Germany

Privacy Bird, 2006. AT&T Corporation, http://privacybird.com.

Thibadeau, 2000. *A Critique of P3P: Privacy on the Web.* The eCommerce Institute, School of Computer Science, Carnegie Mellon University

Tomcat, 2005. Tomcat version 5.5, Apache Software Foundation, tomcat.apache.org.

Yee, Korba, 2004. *Privacy Policies and their Negotiation in Distance Education.* Instructional Technologies: Cognitive Aspects of Online Programs. Idea Group Inc. NRC 46555.