Towards a Mechanism for Incentivating Privacy

Piero A. Bonatti, Marco Faella, Clemente Galdi, and Luigi Sauro

Università di Napoli "Federico II"

Abstract. The economic value of rich user profiles is an incentive for providers to collect more personal (and sensitive) information than the minimum amount needed for deploying services effectively and securely. With a game-theoretic approach, we show that provider competition can reduce such information requests. The key is a suitable mechanism, roughly reminiscent of a Vickrey auction subject to integrity constraints. We show that our mechanism induces rational providers to ask exactly for the user information strictly necessary to deliver their service effectively and securely. In this framework, maximal attribute disclosures become more difficult to achieve.

1 Introduction

Web sites frequently ask their users for personal, sensitive information before granting access to their full functionalities. For example, some of the most popular web sites and services for e-commerce and social networking collect user name, birth date, gender, detailed address, credit card information, and—in some cases—even sex preferences, and political and religious views. Some of these fields can be easily aggregated to form *quasi identifiers* [7,19,16], that is, combinations of attributes such that the probability of their matching a single person is high. (i.e., the values of such attributes uniquely identify an individual with high probability). Releasing fake data is generally not an appropriate or sufficient privacy-preserving measure, as the correctness of some fields may be essential for correct functionality (e.g. credit card information for a purchase and home address for parcel delivery). The ongoing deployment of digital IDs and other cryptographically verifiable documents will further exacerbate this problem.

The above information requests are not parsimonious, in general. Rich user profiles have a significant economic value that constitutes an incentive for increasing the amount of user information collected (and for its disclosure to third parties). Therefore, providers are encouraged to ask for more information than the minimum required to deploy a given service effectively and securely.

Competition may contrast this trend. Indeed, a significant number of users have expressed concern over privacy during online interactions (e.g. user complaints have already influenced Facebook's privacy policy and services), and analysts say that privacy may become a factor of competition [5,17,9].

In this paper we take these analyses seriously and move the first steps towards mechanisms that may increase privacy by exploiting competition between providers. More precisely, such mechanisms should encourage providers to be *truthful*, that is, to ask nothing more than the minimum information sets necessary for correct and secure

V. Atluri and C. Diaz (Eds.): ESORICS 2011, LNCS 6879, pp. 472–488, 2011.

service deployment. Moreover, such approaches should supply users with *all* of the alternative ways of fulfilling the provider's policy, so that users can choose the alternative that they prefer from a privacy viewpoint.

Setting up a widely applicable mechanism of this kind is, of course, a very complex task. Open questions include at least the following: How should users and providers interact? (E.g. direct interaction vs. mediation by trusted third parties; single interactions vs. prolonged negotiations.) What is the interplay of information disclosure and nonfunctional service properties such as cost and quality of service? In particular, can providers compensate for more invasive information requests with such nonfunctional properties? Can the truthful mechanisms developed by economists be adapted to the scenarios described so far?

A complete answer to all of these questions lies beyond the scope of this paper (and any single paper of standard size). As a first step, here we focus on the last of the above questions, which concerns an essential prerequisite for the mechanisms of our interest and, more generally, for all the transactions where "payments" consist in user information disclosures. In these application scenarios, the payment means is naturally discrete and partially ordered, either in quantitative terms (e.g., according to set inclusion), or in qualitative terms (e.g., based on sensitivity). On the contrary, all standard mechanisms developed in microeconomics are based on totally ordered payment domains, e.g., money. Therefore, reconstructing such mechanisms in order to fulfill the requirements of our scenarios is a problem of general interest—and requires nontrivial changes in the underlying mathematics, as discussed below.

The mechanism studied in this paper is analogous to an auction: The user is the auctioneer; the providers "buy" the user's preference and "pay" for it by reducing the amount of (economically valuable) user information requested. The auction mechanism should be *truthful*, that is, information requested by rational selfish providers should match what is *actually* needed to deploy the service correctly and securely.

Perhaps, the most famous truthful auction mechanism is the second-price auction introduced by Vickrey [20]. It is a "one-shot" mechanism (bidders submit their offers in parallel, then the auctioneer evaluates them); the best offer "wins", and the price payed is the second best offer. As we anticipated, some of the main technical differences between Vickrey's mechanism and ours concern the range of bids and utility functions, that here is discrete and partially ordered rather than continuous and totally ordered. Of course, in a partial ordering it is not immediately clear how to generalize the notion of second price.

Even if a user may use a same service multiple times, a one-shot mechanism without memory of previous decisions may be appropriate in many cases. In particular, consider any transaction with no monetary costs, that is, where the only "cost" for the user consist of the personal information released. Usually, after such information has been disclosed and the user's profile created, subsequent service usage requires no further disclosures, so (from a privacy perspective) after the first information release the service can be used for free. Consequently, there is no need for further auctions until services change.

The paper is organized as follows: Section 2 introduces the formal framework and describes its formal properties. Related work is discussed in Section 3. Section 4

concludes the paper with a final discussion of the results and some interesting directions for further work. Proofs can be found in the appendix.

2 Formal Framework

The formal framework is relative to an arbitrary but fixed (implicit) service of interest to the user. The set of agents A is identified with an initial segment of the natural numbers: $A = \{0, 1, 2, \ldots, N\}$. The user is represented by 0 and the providers by $1 \leq i \leq N$. We assume that the services deployed by the providers are all equivalent from the user's perspective. The set of information items that can be requested and released before service access is $C = \{c_1, \ldots, c_z\}$. By analogy with trust negotiation frameworks [2,21,22], information items will be sometimes called *credentials*. The powerset of C is denoted by $\mathscr{P}(C)$.

Credential sets may have different sensitivity. The sensitivity order is modelled with a strict partial order $\prec$. When a credential set r_2 is more sensitive than a credential set r_1, we write $r_1 \prec r_2$. Let $r_1 \preceq r_2$ iff either $r_1 \prec r_2$ or $r_1 = r_2$. We assume that $r_1 \subseteq r_2$ implies $r_1 \preceq r_2$ (intuitively, by enlarging information sets, their sensitivity can only increase).

Several concepts, including policies, will be based on *thresholds* over $\mathscr{P}(C)$, that is, sets of sets of credentials $\theta \subseteq \mathscr{P}(C)$ such that $\theta \neq \emptyset$ and for all distinct $r, r' \in \theta$, $r \not\subseteq r'$. Θ denotes the set of all thresholds over $\mathscr{P}(C)$.

The user's policy is a threshold $pol_0 \in \Theta$. Intuitively, pol_0 represents the maximal sets of information items that the user is willing to disclose to access the service. Formally, a request $r \subseteq C$ is *admissible* iff $\exists r' \in pol_0 : r \subseteq r'$; we denote with $adm(pol_0)$ the set of admissible requests.

Example 1. Suppose $pol_0 = \{\{login, passw\}, \{card\_num,\ exp\_date\}\}$. This policy means that the user is willing to disclose either her login-password pair or her credit card number and expiration date. A request for the credit card number alone is admissible, too, as $\{card\_num\}$ is a subset of the second element of pol_0 and hence $\{card\_num\} \in adm(pol_0)$. On the contrary, $\{login, card\_num\}$ is not admissible, because it is not contained in any element of pol_0. □

Symmetrically, each provider i has a policy $pol_i \in \Theta$ that encodes the minimal (alternative) sets of information items that suffice to deliver the service securely and effectively. Formally, a credential set $r \subseteq C$ *fulfills* pol_i (and grants access to the service) iff $\exists r' \in pol_i : r \supseteq r'$; in the following $ful(pol_i)$ is the set of all $r \subseteq C$ that fulfill pol_i. The *policy profile* is the vector $\mathbf{pol} = \langle pol_0, \ldots, pol_N \rangle$.

A provider i may decide to ask users for more information than what is prescribed by pol_i; the actual information request is called a *strategy* and corresponds to what is traditionally called policy in standard access control frameworks (because it determines which conditions must be fulfilled to access the service). Formally, a *strategy profile* for **pol** is any vector $\mathbf{req} = \langle req_1, \ldots, req_N \rangle$ such that (i) $req_i \in \Theta$, and (ii) $req_i \subseteq ful(pol_i)$.

Each strategy req_i is the information request that provider i submits to the user. Each credential set $r \in req_i$ is an alternative way of fulfilling i's request, that is, the user must release a set of credentials $r' \in ful(req_i)$ in order to access the service deployed by i. The requests of provider i are required (by the second condition above) to fulfill the

minimal requirements imposed by pol_i. Therefore the user's response r' is guaranteed to satisfy pol_i, too. Note that req_i might omit some ways of accessing the service. That is, for some $r \in pol_i$ there may be no $r' \in req_i$ such that $r' \supseteq r$. In this way, a provider may force the user to disclose credentials that are of greater interest for the provider.

Example 2. With reference to Example 1, a provider that can technically support both account-based and pay-per-use access would have the policy: $pol_1 = \{\{login, passw\}, \{card\_num, exp\_date\}\}$. The two members of pol_1 represent the two minimal information sets that need to be collected in order to grant the service. The request of provider 1 may in general be different. For instance, if $req_1 = \{\{card\_num, exp\_date, birth\_date\}\}$, the provider is trying to (i) force the user to disclose her credit card information rather than her account information, and (ii) obtain the user's birth date, which is not strictly necessary to service delivery. Note that the only element of req_1 is a superset of the second element of pol_1, therefore the request fulfills pol_1. However, it is not admissible w.r.t. pol_0. Now suppose that $req_1 = \{\{login, passw\}, \{card\_num, exp\_date, birth\_date\}\}$. In this case, due to the additional alternative $\{login, passw\}$, req_1 is admissible for pol_0 and fulfills pol_1. The user can release the set $\{login, passw\}$ and access the service. □

A strategy req_i is *truthful* iff $req_i = pol_i$, that is, the provider's requests match the actual minimal requirements for secure and effective service delivery.

Finally, each agent i is associated to a preference relation $\preceq_i$. We assume that $\preceq_0$ is $\preceq$ (that is, the user's goal is minimizing the sensitivity of disclosed information), while for all providers i, $\preceq_i$ can be either $\preceq$ or $\subseteq$. In the former case, i's goal is maximizing the sensitivity of the information acquired from users,[1] while in the latter case i's goal is maximizing the amount of such information. Let $\vec{\preceq}$ be the vector $\langle \preceq_0, \ldots, \preceq_N \rangle$.

Now a *(full) profile* is a triple $\pi = \langle \mathbf{pol}, \mathbf{req}, \vec{\preceq} \rangle$ where **pol** is a policy profile and **req** a strategy profile for **pol**. The set of all full profiles will be denoted by Π.

2.1 Selection and Response Mechanism

We need preliminary definitions. The set of optimal admissible requests in a profile π is:

$$opt(\pi) = \min_{\preceq} \left(\bigcup_{j=1}^{N} req_j \cap adm(pol_0) \right),$$

where $\min_{\preceq}(X)$ denotes the set of minimal elements of X according to $\preceq$, that is, $\min_{\preceq}(X) = \{r \in X \mid \forall r' \in X, r' \not\prec r\}$.

The user prefers those providers that make minimal information requests. Formally, let the provider i be a *candidate winner* in π iff $opt(\pi) \cap req_i \neq \emptyset$. The set of candidate winners is denoted by $cw(\pi)$.

[1] The rationale is that information value is often correlated with sensitivity. For simplicity, in this first paper we assume a shared (objective) measure of sensitivity – e.g. based on statistics about the identification power of attribute aggregates, cf. quasi-identifiers [7] – so that $\preceq$ may be regarded as common knowledge. Generalizations are discussed in Sec. 4.

Each candidate winner i is associated to a set of possible *responses*, $res(\pi, i)$. Possible responses are credential sets that must satisfy both the user's policy and the provider's request, that is, for all $r \in res(\pi, i)$, $r \in adm(pol_0) \cap ful(req_i)$. Different specific definitions of $res(\pi, i)$ yield different properties in terms of robustness (i.e., lack of unnecessary transaction failures) and amount of information released. Before discussing the alternatives, let us fix the decision making process (provider selection and response selection), consisting of two steps:

1. choose a provider $i \in cw(\pi)$
2. choose a response $r \in res(\pi, i)$.

If $res(\pi, i) = \emptyset$, then the transaction fails. To simplify the discussion, let us assume that the above choices are made at random with uniform probability (different distributions can be adopted, though).

So far, the framework is similar to an auction with some extra constraints posed by policies. In Vickrey's auctions, truthfulness is achieved by setting the price to the second best offer. In this framework, a direct counterpart of this idea consists in defining $res(\pi, i)$ as the set of best requests made by all the providers $j \neq i$. In order to formalize this idea we need a few more auxiliary definitions. First, let $opt_{-i}(\pi)$ denote the best admissible requests of the providers $j \neq i$:

$$opt_{-i}(\pi) = \min_{\preceq}\left(\bigcup_{1 \leq j \leq N}^{j \neq i} req_j \cap adm(pol_0) \right).$$

Then, let $res_0(\pi, i)$ be the set of all the best admissible requests of the providers $j \neq i$ that satisfy i's request, that is: $res_0(\pi, i) = opt_{-i}(\pi) \cap ful(req_i)$.

By setting $res(\pi, i) = res_0(\pi, i)$ one obtains a mechanism that easily leads to failures, because the requests of the winners might not fulfill each other.

Example 3. Suppose that the user's policy permits the simultaneous disclosure of her credit card number (*card_num*), its security code (*sec_code*), user name (*name*), and birth date (*birth_date*), i.e., $pol_0 = \{\{card\_num, sec\_code, name, birth\_date\}\}$. Let the requests of providers 1 and 2 be $req_1 = \{\{card\_num, name, sec\_code\}\}$ and $req_2 = \{\{card\_num, name, birth\_date\}\}$, respectively. If req_1 and req_2 are not comparable with respect to $\prec$, then both requests are $\preceq$-minimal and admissible, so the set of candidate winners is $cw(\pi) = \{1, 2\}$. However, the request of provider 1 does not fulfill the request of provider 2 and viceversa. Then $res_0(\pi, 1) = res_0(\pi, 2) = \emptyset$ and the transaction fails no matter which of the two providers is chosen. □

This problem can be mitigated by adding to the pool of replies the largest admissible requests, that is, the members of pol_0. Let

$$res_1(\pi, i) = \min_{\preceq}(opt_{-i}(\pi) \cup pol_0) \cap ful(req_i),$$
$$res_2(\pi, i) = (opt_{-i}(\pi) \cup pol_0) \cap ful(req_i).$$

The first definition (res_1) still leads to a failures; for instance, in Example 3, it is equivalent to res_0 because $\min_{\preceq}(opt_{-i}(\pi) \cup pol_0) = opt_{-i}(\pi)$, for $i = 1, 2$. The second definition

(res_2) does not lead to any failure in Example 3; however, the user has to disclose all releasable credentials (*card_num*, *sec_code*, *name*, *birth_date*). In general, $res_2(\pi, i)$ contains (at least) all the elements of pol_0 that cover some request of i, therefore some maximal disclosable set of credentials can always be released with probability greater than 0. Then we move over to a more parsimonious definition (in terms of maximal disclosures). The idea consists in "interpolating" intermediate requests between the optimal requests of all $j \neq i$. Such interpolation constitutes a "vault" above req_i, from which possible responses can be selected. Formally, let

$$vault(\pi, i) = \max_{\subseteq} \{ r \subseteq C \mid r \in adm(pol_0) \wedge \forall r' \in opt_{-i}(\pi).\ r' \not\prec r \}\,.$$

In Example 3, assuming for simplicity that $\preceq$ equals $\subseteq$, the vaults are

$$\begin{aligned} vault(\pi, 1) = vault(\pi, 2) = \\ \{\ &\{card\_num, name, sec\_code\}, \{sec\_code, name, birth\_date\}, \\ &\{card\_num, sec\_code, birth\_date\}, \{card\_num, name, birth\_date\}\ \}\,. \end{aligned}$$

They contain the providers' requests, as well as elements that do not fulfill them. Therefore responses should consist of the vault elements that cover some optimal request of i:

$$res(\pi, i) = vault(\pi, i) \cap ful(opt(\pi) \cap req_i)\,.$$

Compare this definition with the standard second price approach: There, the winner pays the minimum price that is not worse (i.e., smaller) than any other offer; analogously, in our framework, the winner gets a maximal response that is not worse (i.e., more sensitive) than any other offer, and satisfies both the user's policy and the winner's request.

We are going to study in depth the framework based on this definition. Before starting its formal analysis, note that in the case of Example 3, $res(\pi, i) = req_i$ ($i = 1, 2$), that is, the selected provider receives nothing more than what it asked for (as opposed to what happens with res_2). In general, however, the user may have to release more than what the winning provider asks for. In general, this is the price to pay for truthfulness. At the end of this section we will characterize a wide class of scenarios in which no unnecessary information is disclosed.

The first formal property of the definition based on vaults concerns its robustness: Unlike res_0 and res_1, it introduces no unnecessary failures. In other words, whenever some request is admissible (equivalently, $cw(\pi) \neq \emptyset$), all candidate winners can be given a response:

Theorem 1. *If there exist a provider j and a request $r \in req_j$ such that $r \in adm(pol_0)$, then for all $i \in cw(\pi)$, $res(\pi, i) \neq \emptyset$.*

It is easy to verify that a similar property holds for res_2. So the second formal property of interest concerns a comparison of the two robust strategies *res* and res_2 with respect to the amount of credentials potentially released. It can be shown that *res* is generally more parsimonious than res_2:

Theorem 2. *For all $r \in res(\pi, i)$ there exists $r' \in res_2(\pi, i)$ such that $r \subseteq r'$.*

Next we investigate the effectiveness of the provider selection mechanism in reducing the amount of information disclosed in the worst case. A first question related to this issue is: Under which circumstances can a maximal releasable set of credentials (i.e., a member of pol_0) be disclosed?

Theorem 3. *Let $r \in pol_0$, $r \in res(\pi, i)$ if and only if there exists $x \in (opt(\pi) \cap req_i)$ such that $x \subseteq r$ and for all the other providers $j \neq i$ and for all $r' \in req_j$, it holds $r' \not\prec r$.*

Note that if $r' \subseteq r$, then $r' \not\prec r$ implies $r' = r$. Then Theorem 3 says that r can be released to i if either there is no competition within the option r (i.e., the other providers' requests are not compatible with r), or the competitors ask exactly for r. Consequently, it appears that competition makes maximal disclosures more difficult to achieve systematically, at least in the absence of detailed information about the user's policy.

Another interesting, related problem is characterizing the circumstances under which the user may have to release all disclosable credentials at once (as it may happen in the formal trust negotiation frameworks studied in the literature).

Corollary 1. *Assume that i makes an admissible request ($req_i \cap adm(pol_0) \neq \emptyset$). Then $\bigcup_{r \in pol_0} r$ can be disclosed to provider i iff the following conditions hold:* (i) $pol_0 = \{r\}$, *and* (ii) $r \in req_j$ *for all providers $j \neq i$ such that $req_j \cap adm(pol_0) \neq \emptyset$.*

According to the first condition in the above corollary, if $|pol_0| > 1$, then it is impossible to release $\bigcup_{r \in pol_0} r$. The reason is clear: whenever $|pol_0| > 1$, the user's policy encodes some integrity constraints that forbid the disclosure of arbitrary unions of disclosable credentials. For example, if $pol_0 = \{\{birthday\}, \{address\}\}$ then both birth date and address can be separately disclosed, but their union is considered too sensitive to be released. Now suppose that $|pol_0| = 1$. Condition 2 says that either i has no competitors (no other provider j makes any admissible request), or i's competitors all ask for r (which is unlikely in practice unless pol_0 is public). This shows how competition helps in reducing complete credential disclosures.

Example 4. Suppose that a user is willing to execute payments either by using her credit card or by bank transfer. In the first case she permits the simultaneous disclosure of her credit card number (*card_num*) and its security code (*sec_code*). For the latter payment form, she is willing to provide the unique ID (*id*) associated to the bank transfer and her own bank account information (*acc_info*). Formally $pol_0 = \{\{card\_num, sec\_code\}, \{id, acc\_info\}\}$. The user can select among three providers for executing a given payment whose requests are, respectively: $req_1 = \{\{card\_num\}\}$, $req_2 = \{\{card\_num\}, \{id\}\}$, $req_3 = \{\{card\_num, sec\_code\}\}$. In such context, there is clear competition among all servers since everyone allows credit card payments. Server 1 and 2 only require *card_num* while server 3 requires both *card_num* and *sec_code*. So, in a parsimonious selection, the user would prefer server 1 or 2. For the second payment method there is no competition at all. Indeed only server 2 allows bank transfers and requires the unique transaction identifier in order to accept the payment. Let π be a profile describing such a scenario. The set of optimal admissible requests is $opt(\pi) = \min_{\subseteq}(\bigcup_{j=1}^{N} req_j \cap adm(pol_0)) = \{\{card\_num\}, \{id\}\}$. Then the set of candidate winners is $cw(\pi) = \{1, 2\}$. Let us focus on server 1. The set of optimal admissible requests made by all providers except 1 is $opt_{-1}(\pi) = \{\{card\_num\}, \{id\}\}$. Then the

vault for provider 1—i.e., the maximal subsets of credentials that are admissible for the client and do not cover optimal requests made by other servers—is $vault(\pi, 1) = \{\{card\_num\}, \{sec\_code\}, \{id\}, \{acc\_info\}\}$. Note that the vault contains no elements of pol_0, because provider 1 competes with provider 2 (whose request $\{card\_num\}$ expunges $\{card\_num, sec\_code\}$ from the vault). Finally, the set of possible responses for server 1 are the members of $vault(\pi, 1)$ that satisfy the server's requests, i.e., $res(\pi, 1) = \{\{card\_num\}\}$. Similarly, for provider 2 we have $opt_{-2}(\pi) = \{\{card\_num\}\}$, $vault(\pi, 2) = \{\{card\_num\}, \{sec\_code\}, \{id, acc\_info\}\}$, and $res(\pi, 2) = \{\{card\_num\}, \{id, acc\_info\}\}$.

Thus, when different servers compete within a specific element of pol_0, (here, $\{card\_num, sec\_code\}$), such element is not entirely disclosed to any server. On the other hand, if exactly one server makes a request compatible with some element in $r \in pol_0$, i.e., if there is no competition within r ($r = \{id, acc\_info\}$ in the example), then r is fully disclosed with nonzero probability. □

2.2 Rational Strategies

We are left to characterize the strategies adopted by ideally rational providers. We consider two kinds of providers: Providers of the first kind are mainly interested in attracting new customers, i.e. their primary goal is maximizing the probability of being selected (or probability of winning) $pw(\pi, i)$, where $pw(\pi, i) = 1/|cw(\pi)|$ if $i \in cw(\pi)$, and $pw(\pi, i) = 0$ otherwise. As a secondary goal, these providers prefer those strategies that better meet their preference $\preceq_i$. Providers of the second kind invert the above priorities. A new player in a given application domain is likely to be a player of the first kind. Similarly, providers whose main income is based on advertisement are likely to be providers of the first kind. On the contrary, when the utility of service usage is dominated by the value of user profiles, providers should be expected to be agents of the second kind.

In order to formalize the perfect strategies for these providers we need some auxiliary notions. First, one needs to compare different responses w.r.t. provider preferences. For this purpose, $\preceq_i$ should be extended from credential sets to *sets* of credential sets (i.e., the range of *res*).

Definition 1. *For all $\rho, \rho' \subset \mathscr{P}(C)$, let $\rho \preceq_i^* \rho'$ iff*

1. *for all $r \in \rho$, there exists $r' \in \rho'$ such that $r \preceq_i r'$, and*
2. *for all $r' \in \rho'$ and $r \in \rho$, $r' \not\preceq_i r$.*

In other words ρ' is preferable to ρ if for all possible responses in ρ there exists an equally preferred or better response in ρ' (according to i's preferences) and none of the responses in ρ' is less preferable than any response in ρ.

Next, we need a handy way of replacing the strategy of an agent: For all strategy profiles **req** and all providers $1 \leq i \leq N$, let

$$\mathbf{req}[i \leftarrow req'] = \langle req_1, \ldots, req_{i-1}, req', req_{i+1}, \ldots, req_N \rangle ,$$

and for all profiles $\pi \in \Pi$, let

$$\pi[i \leftarrow req'] = \langle \mathbf{pol}, \mathbf{req}[i \leftarrow req'], \vec{\preceq} \rangle .$$

Finally, let $pol \in \Theta$ be an arbitrary but fixed policy. The set of profiles where $pol_i = pol$ is denoted by Π^i_{pol}. A *strategy for pol* is any $req \in \Theta$ such that for all $r \in req$, $r \in ful(pol)$. Now the optimal strategies for the two kinds of agents can be formalized.

Definition 2. *A strategy req for pol is a* dominant attraction strategy for i with respect to *pol iff for all* $\pi \in \Pi^i_{pol}$,

1. $pw(\pi, i) \leq pw(\pi[i \leftarrow req], i)$, *and*
2. *if* $pw(\pi, i) = pw(\pi[i \leftarrow req], i)$ *then* $res(\pi, i) \preceq^*_i res(\pi[i \leftarrow req], i)$.

Definition 3. *A strategy req for pol is a* dominant investigation strategy for i with respect to *pol iff for all* $\pi \in \Pi^i_{pol}$,

1. $res(\pi, i) \preceq^*_i res(\pi[i \leftarrow req], i)$, *and*
2. *if* $res(\pi[i \leftarrow req], i) = res(\pi, i)$ *then* $pw(\pi, i) \leq pw(\pi[i \leftarrow req], i)$.

A few explanations are in order here. The universal quantification over Π^i_{pol}, whose only invariant is $pol_i = pol$, means that strategy *req* is optimal w.r.t. all the other possible strategies req_i that might be adopted by i, and this holds in all possible contexts (i.e., no matter what the policies and strategies of the other agents are). Our mechanism yields the desired result: the truthful strategy $req = pol$ is the best strategy a provider can adopt, under both priorities.

Theorem 4. *For all* $pol \in \Theta$ *and all providers i, the unique dominant attraction strategy for i w.r.t. pol is pol itself.*

Theorem 5. *For all* $pol \in \Theta$ *and all providers i, the unique dominant investigation strategy for i w.r.t. pol is pol itself.*

In game-theoretic terms, the previous two theorems prove that being truthful is a *dominant strategy equilibrium* (DSE) [13], i.e., no matter what the other agents' policies and strategies are, being truthful is always the best response. Every DSE is in particular a Nash equilibrium.

One may wonder whether gaining information on the behavior of the other agents could allow a provider to increase either its winning probability or the amount of credentials received from the client. The answer is negative, regardless of the extra information available to the provider. Indeed, any gained information corresponds to restricting the set of possible profiles Π^i_{pol} in the definition of dominant attraction (resp., investigation) strategy (Definition 2 and 3, respectively). Clearly, by applying any such restriction, the set of dominant attraction (resp., investigation) strategies may only increase. However, since any pair of dominant strategies dominate each other, it is straightforward to see from Definition 2 and 3 that all dominant strategies give rise to the same probability of winning and the same response from the client.

The presence of rational (and hence truthful) providers may induce minimal disclosures in a framework that, in general, releases to providers more information than what they ask for. For simplicity, we analyze this issue in scenarios where all providers have the same policy. In practice, this assumption is naturally satisfied when provider policies are determined by the same technological constraints—for example, all providers supporting VISA credit card payments must provide VISA's servers with the same information for credit card validation (as in Example 6 below).

Theorem 6. *If all providers have the same policy and there are two truthful providers i and j, then* $res(\pi, i) = res(\pi, j) = req_i \cap adm(pol_0) \subseteq pol_i$.

In informal terms, the above theorem ensures that under the uniform policy hypothesis, rational servers are given only elements of their policy, that is, some of the minimal possible credential sets that grant access to the service. Consider the following scenario, for example. It is inspired by real flight reservation portals. Kayak and Momondo ask for no information; tickets are purchased directly from airline companies. On the contrary, eDreams asks for a rich user profile that is then used to make a request to airline companies. Note that eDream user profiles comprise attributes that are not mandatory for airline companies. The following is a formalization of this scenario:

Example 5. Assume that $pol_1 = pol_2 = pol_3 = \{\emptyset\}$, $req_1 = req_2 = \{\emptyset\}$, and $req_3 = \{\{name, address, phone\_num, email\}\}$. Note that providers 1 and 2 are truthful and provider 3 is not. Clearly, only providers 1 and 2 are candidate winners. The response is $\{\emptyset\}$, that is, the user releases no information. Note that the same result is obtained when $pol_3 \neq \{\emptyset\}$ (e.g., $pol_3 = req_3$); thus, in future work, it may be interesting to relax the hypothesis of Theorem 6. □

Moreover, if all policies are the same and there is at least one rational (truthful) server, then the other servers cannot receive more information than what the policy requires.

Theorem 7. *If all providers have the same policy and provider i is truthful, then for all* $j \neq i$, $res(\pi, j) \subseteq pol_j$.

Example 6. Consider an e-commerce application with the same credentials as in Example 3. Assume that all vendors use the same underlying financial institution that requests the credit card number and either the owner's name or the credit card security code, so the providers all share the same policy $pol_i = \{\{card\_num, name\}, \{card\_num, sec\_code\}\}$. Suppose that server i is truthful, while j has strategy $req_j = \{\{card\_num, name, birth\_date\}, \{card\_num, sec\_code\}\}$, i.e., in addition to the credit card number and owner's name, j requests also her birth date. The policy of the client is $pol_0 = \{\{card\_num, name, sec\_code, birth\_date\}\}$, i.e., all credentials are simultaneously releasable. Now j is a candidate winner (as it makes the request $\{card\_num, sec\_code\}$). However, even if j is selected to deliver the service, in accordance with Theorem 7 it will *not* receive the user's birth date, even if it is releasable by the client. Indeed, we have $opt_{-j}(\pi) = \{\{card\_num, name\}, \{card\_num, sec\_code\}\}$. This means that $vault(\pi, j)$ consists of $\{card\_num, name\}$, $\{card\_num, sec\_code\}$, $\{card\_num, birth\_date\}$ and $\{name, sec\_code, birth\_date\}$. Finally, we have that $res(\pi, j) = \{\{card\_num, sec\_code\}\}$. □

3 Related Work

To the best of our knowledge, the approach introduced in the above sections has no analogue in the literature. Standard access control frameworks place no constraints on policies and set up no mechanisms for reducing the extension and sensitivity of user profiles. In the trust negotiation area, privacy is mainly pursued by having

disclosed information match as precisely as possible the provider's request, see for example [18,14,10]. Another work aims at preventing providers from inferring which *types* of credentials are owned by the user when a transaction fails [21]. A major difference between our approach and theirs is that those works do not attempt to influence the providers' requests. There, providers' policies do not represent minimal functional and security requirements (unlike our pol_is); they should rather be considered as part of the specification of req_i. Not only can trust negotiation policies ask for unnecessary information; some interoperable negotiation strategies may further inflate requests by forcing the user agent to disclose all releasable credentials in the attempt to keep the negotiation alive [22,2]. Therefore, the issue of minimizing the difference between req_i and pol_i is not tackled (and there is no precise counterpart of pol_i).

In [8], privacy is tackled in the context of auctions (including second-price auctions). Their purpose is the opposite of ours, namely, minimizing the amount of information that *bidders* have to disclose in order to let the auctioneer compute the optimal outcomes. Payments are fully traditional: continuous and totally ordered. Similarly, another economically inspired model [11] proposes both an estimate of the value of private information and a fair compensation for such information release that may induce *users* to release richer and correct information about their personal preferences.

Concerning the many auction models introduced in the past, the main technical difference is that utility functions and bids always range over a totally ordered domain such as the set of real numbers. Then truthfulness can be obtained with a straightforward implementation of the second-price idea, without resorting to more complex notions such as vaults.

4 Discussion and Perspectives

4.1 Current Achievements

We provided a first formal evidence that the potential competition between equivalent applications may enhance privacy and reduce the amount of personal information requested and collected by application providers. As a starting point, we focussed on scenarios where the competing services are equivalent with respect to functionalities and quality, and the only cost for the user consists of the personal information released. Flight reservation applications provide concrete examples of such scenarios, cf. Example 5 and the preceding paragraph. We argued that one-shot mechanisms (like second-price auctions) are appropriate for such scenarios; indeed, after using the service a first time, subsequent usage is "free" from the point of view of information disclosure; therefore, re-using the same service is always an optimal choice for the user until either demand or offer change.

We proved that a suitable one-shot mechanism regulating provider selection and information disclosure induces truthful behavior in selfish rational providers, resulting in minimal information requests and in a wider range of choices for the user. It is important to note that essentially these results still hold even if agents know each other's policies and strategies.

When functional and security requirements are the same for all providers (e.g., because such requirements are determined by exogenous technological constraints), two rational (and hence truthful) providers suffice to minimize the amount of user information disclosed to *any* provider—and interestingly, if only one provider is rational, then it is the only provider that may receive a non-minimal response. In some cases, the hypothesis that all policies are the same can be relaxed, as shown in Example 5. Another set of results shows that extracting all releasable credentials (or any maximal disclosable set of credentials) from a user is a difficult task, whose systematic achievement, in practice, requires some knowledge of the user's policy.

Competition may be exploited to address a weakness of automated disclosure techniques, such as OpenID profile sharing and automated trust negotiation. These approaches require a policy to decide when a user profile can be automatically transferred to a new web service, or when an information request is reasonable in a given context. Formalizing a policy that decides on behalf of the user whether a provider is collecting a reasonable set of user attributes is a very difficult task. A mechanism inducing a spontaneous moderation of provider requests may turn out to be less expensive and/or more reliable.

Concerning trust negotiation, it is known that interoperable negotiation strategies are vulnerable to attacks that force a peer to release progressively all disclosable information (cf. [22,2]). It seems that the current one-shot mechanism can address this problem: First, the user agent negotiates with all the providers' agents using a method that does not actually disclose credentials until the end of the negotiation (as in the protocol introduced in [21]). At the end of the negotiations, before really sending credentials, the user can choose among the successful negotiations those with minimal information disclosure, and compute the response with our mechanism, thereby inducing rational providers to reduce information requests.

4.2 Possible Variations to the Current Framework

Several aspects of the framework can be modified without affecting our results. For example, it is not hard to see that different probability distribution can be used in choosing winner and response. As an example of possible applications, skewed distributions over candidate winners may address additional user preferences over providers. Similarly, the preference relations $\preceq_i^*$ employed to compare responses and define dominant strategies can be modified in various ways, in order to model providers with different risk attitudes. For instance, an optimistic (resp. pessimistic) provider may consider only the $\preceq_i$-maximal (resp. $\preceq_i$-minimal) elements of $res(\pi, i)$ (the current definition considers all of $res(\pi, i)$). Our results hold for all such agents.

4.3 Generalizing Preferences

The current theoretical framework should be extended and complemented along several directions. In general, providers compete also on properties such as cost and quality of service. They can easily be modelled by extending credential sets to richer sets, including items that represent the additional quantities of interest. This affects some of the assumptions we adopt in this paper.

For example, in the extended framework, preference relations rank aggregates of credentials, money, and quality of service, thereby reflecting a tradeoff between privacy-related costs and other costs and benefits; consequently, user preferences have a more "personal" nature and it would not be reasonable to make the simplifying assumption that $\preceq$ is based on an objective, shared sensitivity measure. As a consequence, provider preferences could not be restricted to $\subseteq$ and $\preceq$, and the effects of this generalization should be formally analyzed. In this context, users may publish their preference relation to get personalized offers; it should be verified whether a rational user should be truthful, and whether preference disclosure may constitute a privacy violation in itself.

As preference specifications become more sophisticated, the need may arise for usability-enhancing techniques. For instance, a coarse-grained preference relation in a pure information disclosure scenario (where service re-use has no additional "costs") may induce the user to always select the same provider (lock-in effect). The articulated approach to monitoring, refining, and learning preferences introduced in [15] for access control policies can perhaps be adapted to our framework.

4.4 Repeated Auctions

Another consequence of modelling features such as money and quality is that each call to a same service may have additional costs; then it makes sense to repeat the service selection process and abandon the one-shot approach. We conjecture that, in general, a sequence of independent selections may lead to globally suboptimal disclosures (as any greedy strategy). A formal analysis of iterative selections is an important step in our agenda, that may start from the literature on repeated (or sequential) procurement auctions, e.g. [12,1]. It is also interesting to evaluate "globalized" selections over bundles of services as an attempt to optimize information disclosure for a set of commonly used services. It is known that this optimization problem is tractable in some cases [4].

4.5 Deployment

Last but not least, the need is felt for an articulated analysis of deployment solutions; for example, providers may want to make sure that the service selection mechanism is carried out correctly, i.e. users are not cheating and actually disclose an element of the vault. Some application contexts may admit trusted intermediary services. Portals similar to Kayak, Momondo etc. are interesting candidates to fill in this role, that may create new business opportunities and models. In the absence of trusted third parties (i.e., the user is the auctioneer), auctions can be implemented using Secure Multiparty Computation. The instantiation of general MPC constructions can be inefficient; it is, however, possible to design less complex specialized MPC protocols implementing the described variation of second price auctions, cf. [3]. Alternatively, one can resort to ad hoc protocols where credential requests are eventually revealed to all providers (*commitments* and *blind signature* [6] may be employed for this purpose).

Acknowledgments. The authors would like to thank Alessandro Bonatti for helpful preliminary discussions, and Adam J. Lee and the anonymous referees for their detailed and stimulating feedback.

References

1. Bae, J., Beigman, E., Berry, R.A., Honig, M.L., Vohra, R.V.: Sequential bandwidth and power auctions for distributed spectrum sharing. IEEE Journal on Selected Areas in Communications 26(7), 1193–1203 (2008)
2. Baselice, S., Bonatti, P., Faella, M.: On interoperable trust negotiation strategies. In: IEEE POLICY 2007, pp. 39–50. IEEE Computer Society, Los Alamitos (2007)
3. Bogetoft, P., Damgård, I., Jakobsen, T.P., Nielsen, K., Pagter, J., Toft, T.: A practical implementation of secure auctions based on multiparty integer computation. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 142–147. Springer, Heidelberg (2006)
4. Bonatti, P.A., Festa, P.: On optimal service selection. In: Ellis, A., Hagino, T. (eds.) Proc. of the 14th Int. Conf. on World Wide Web, WWW 2005, pp. 530–538. ACM, New York (2005)
5. Broache, A.: Competition is good for search privacy, report says. CNET News (August 8, 2007), http://news.cnet.com/Competition-is-good-for-search-privacy,-report-says/2100-1029_3-6201468.html
6. Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology - Crypto 1982, pp. 199–203. Springer, Heidelberg (1983)
7. Dalenius, T.: Finding a needle in a haystack - or identifying anonymous census records. Journal of Official Statistics 2(3), 329–336 (1986)
8. Feigenbaum, J., Jaggard, A.D., Schapira, M.: Approximate privacy: foundations and quantification (extended abstract). In: Parkes, D.C., Dellarocas, C., Tennenholtz, M. (eds.) ACM Conference on Electronic Commerce, pp. 167–178. ACM, New York (2010)
9. Gray, E.: FTC to boost competition in privacy protection. Global Competition Review (September 23, 2010)
10. He, Y., Zhu, M., Zheng, C.: An efficient and minimum sensitivity cost negotiation strategy in automated trust negotiation. In: Int. Conf. Comp. Sci. and Soft. Eng.,, vol. 3, pp. 182–185 (2008)
11. Kleinberg, J., Papadimitriou, C.H., Raghavan, P.: On the value of private information. In: TARK 2001: Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 249–257. Morgan Kaufmann, San Francisco (2001)
12. Luton, R., McAfee, P.R.: Sequential procurement auctions. Journal of Public Economics 31(2), 181–195 (1986)
13. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)
14. Paci, F., Bauer, D., Bertino, E., Blough, D.M., Squicciarini, A.C.: Minimal credential disclosure in trust negotiations. In: Bertino, E., Takahashi, K. (eds.) Digital Identity Management, pp. 89–96. ACM, New York (2008)
15. Sadeh, N.M., Hong, J.I., Cranor, L.F., Fette, I., Kelley, P.G., Prabaker, M.K., Rao, J.: Understanding and capturing people's privacy policies in a mobile social networking application. Personal and Ubiquitous Computing 13(6), 401–412 (2009)
16. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering 13(6), 1010–1027 (2001)
17. Schwartz, A., Cooper, A.: Search privacy practices: A work in progress. Center for Democracy and Technology report (August 2007)
18. Squicciarini, A.C., Bertino, E., Ferrari, E., Paci, F., Thuraisingham, B.M.: PP-trust-X: A system for privacy preserving trust negotiations. ACM Trans. Inf. Syst. Secur. 10(3) (2007)
19. Sweeney, L.: Guaranteeing anonymity when sharing medical data, the Datafly system. Journal of the American Medical Informatics Association (1997)
20. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. Journal of Finance 16, 8–37 (1961)

21. Winsborough, W.H., Li, N.: Protecting sensitive attributes in automated trust negotiation. In: WPES, pp. 41–51. ACM, New York (2002)
22. Yu, T., Winslett, M., Seamons, K.E.: Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. ACM Trans. Inf. Syst. Secur. 6(1), 1–42 (2003)

Proofs

Theorem 2. *For all $r \in res(\pi, i)$ there exists $r' \in res_2(\pi, i)$ such that $r \subseteq r'$.*

Proof. Assume $r \in res(\pi, i)$. By definition $r \in vault(\pi, i)$ and $r \in ful(opt(\pi) \cap req_i)$; therefore, $r \in adm(pol_0)$ and $r \in ful(req_i)$. From $r \in adm(pol_0)$, it follows that there exists $r' \in pol_0$ such that $r \subseteq r'$. Then from $r \in ful(req_i)$, derive $r' \in ful(req_i)$, and hence $r' \in pol_0 \cap ful(req_i)$. The theorem follows immediately from the definition of $res_2(\pi, i)$. □

Theorem 3. *Let $r \in pol_0$, $r \in res(\pi, i)$ if and only if there exists $x \in (opt(\pi) \cap req_i)$ such that $x \subseteq r$ and for all the other providers $j \neq i$ and for all $r' \in req_j$, it holds $r' \not\prec r$.*

Proof. (If) Note that since for all $r' \in req_j$, with $j \neq i$, $r' \not\prec r$, *a fortiori* for all $r' \in opt_{-i}(\pi)$, $r' \not\prec r$. Then, the thesis follows by applying the definition of $vault(\pi, i)$ and $res(\pi, i)$.

(Only if) By definition, if $r \in res(\pi, i)$, then $r \in vault(\pi, i)$ and there exists a request x such that $x \subseteq r$ and $x \in (opt(\pi) \cap req_i)$. Assume now that for some provider $j \neq i$ and some request $r' \in req_j$, $r' \prec r$. On one hand, there exists in $opt_{-i}(\pi)$ a request r'' such that $r'' \preceq r'$ and hence $r'' \prec r$. On the other hand, since $r \in vault(\pi, i)$, for all $r' \in opt_{-i}(\pi)$, $r' \not\prec r$ *(absurdum).* □

Corollary 1. *Assume that i makes an admissible request ($req_i \cap adm(pol_0) \neq \emptyset$). Then $\bigcup_{r \in pol_0} r$ can be disclosed to provider i iff the following conditions hold:* (i) $pol_0 = \{r\}$, *and* (ii) $r \in req_j$ *for all providers $j \neq i$ such that $req_j \cap adm(pol_0) \neq \emptyset$.*

Proof. Since the client releases exactly one element in pol_0, the only way in which the client could release $\bigcup_{r \in pol_0} r$ is that pol_0 contains exactly one set r. Then, the proof follows easily from Theorem 3. □

In the following results, we need two auxiliary relations: for all $\theta, \theta' \in \Theta$ let $\theta \sqsubseteq \theta'$ iff $\forall r' \in \theta'\ \exists r \in \theta : r \subseteq r'$. Note that for all providers i, it holds $pol_i \sqsubseteq req_i$. Similarly, let $\theta \trianglelefteq \theta'$ iff $\forall r' \in \theta'\ \exists r \in \theta : r \preceq r'$.

Lemma 1. *For all $r \in req_i \cap opt(\pi)$ there exists $r' \subseteq r$ such that $r' \in pol_i \cap opt(\pi[i \leftarrow pol_i])$.*

Proof. Let $r \in req_i \cap opt(\pi)$. Since $pol_i \sqsubseteq req_i$, there exists $r' \in pol_i$ s.t. $r' \subseteq r$; moreover $r \in opt(\pi)$ implies $r \in adm(pol_0)$ and then $r' \in adm(pol_0)$. Assume per adsurdum that there exists $r'' \in opt(\pi[i \leftarrow pol_i])$ s.t. $r'' \prec r'$. Clearly, r'' cannot belong to pol_i because pol_i is by definition a threshold; therefore there exists a provider $j \neq i$ such that $r'' \in req_j \cap adm(pol_0)$, but in this case, as req_j is the same in both π and $\pi[i \leftarrow pol_i]$, we would have $r'' \prec r$ and hence $r \notin opt(\pi)$ against the hypothesis. Therefore, $r' \in opt(\pi[i \leftarrow pol_i])$. □

Corollary 2. *For all $\pi \in \Pi$ and providers i, we have $opt(\pi[i \leftarrow pol_i]) \trianglelefteq opt(\pi)$.*

Corollary 3. *For all $\pi \in \Pi$, if $i \in cw(\pi)$ then $i \in cw(\pi[i \leftarrow pol_i])$.*

Lemma 2. *For all $\pi \in \Pi$, if $i \in cw(\pi)$ then $cw(\pi[i \leftarrow pol_i]) \subseteq cw(\pi)$.*

Proof. It suffices to show that for all servers $j \neq i$, $j \in cw(\pi[i \leftarrow pol_i])$ implies $j \in cw(\pi)$. Let r be in $req_j \cap opt(\pi[i \leftarrow pol_i])$. By definition, $r \in adm(pol_0)$. Furthermore, because of *opt* minimality, for all $r' \in opt(\pi[i \leftarrow pol_i])$, we have $r' \not\prec r$. Since $opt(\pi[i \leftarrow pol_i]) \trianglelefteq opt(\pi)$ (Corollary 2), for all $r'' \in opt(\pi)$, $r'' \not\prec r$. Therefore, $r \in req_j \cap opt(\pi)$. □

Lemma 3. *For all $\pi \in \Pi$ and providers i, it holds that $res(\pi, i) \subseteq res(\pi[i \leftarrow pol_i], i)$.*

Proof. $r \in res(\pi, i)$ implies that $r \in vault(\pi, i)$ and there exists $r' \in opt(\pi) \cap req_i$ s.t. $r' \subseteq r$. Note that $vault(\pi, i)$ does not depend on the request of i, therefore $vault(\pi, i) = vault(\pi[i \leftarrow pol_i], i)$. Moreover, due to Lemma 1, there exists r'' s.t. $r'' \subseteq r'$ and $r'' \in opt(\pi[i \leftarrow pol_i]) \cap pol_i$. Therefore, $r \in vault(\pi[i \leftarrow pol_i], i) \cap ful(opt(\pi[i \leftarrow pol_i]) \cap pol_i)$, i.e. $r \in res(\pi[i \leftarrow pol_i], i)$. □

Lemma 4. *For all $\pi \in \Pi$ and providers i it holds that $res(\pi, i) \preceq_i res(\pi[i \leftarrow pol_i], i)$.*

Proof. If $res(\pi, i) \neq res(\pi[i \leftarrow pol_i], i)$, we need to show that both conditions in Definition 1 are met. By Lemma 3, $res(\pi, i) \subseteq res(\pi[i \leftarrow pol_i], i)$. Condition 1 follows by reflexivity of relation $\preceq_i \in \{\preceq, \subseteq\}$. Indeed, for every $r \in res(\pi, i)$, there exists $r' = r \in res(\pi[i \leftarrow pol_i], i)$ such that $r \preceq_i r'$.

As for condition 2, we observe that $res(\pi[i \leftarrow pol_i], i)$ is a set of maximal elements w.r.t. $\preceq$. This means that every pair of elements in $res(\pi[i \leftarrow pol_i], i)$ are incomparable w.r.t. $\preceq$, and hence w.r.t. $\subseteq$. In particular, for every $r' \in res(\pi[i \leftarrow pol_i], i)$ and $r \in res(\pi, i) \subseteq res(\pi[i \leftarrow pol_i], i)$, it holds that $r' \not\prec r$ and hence $r' \not\subset r$. □

Lemma 5. *If $opt(\pi) \cap req_i \neq \emptyset$, then $res(\pi, i) \neq \emptyset$.*

Proof. Let $V = \{r \subseteq C \mid r \in adm(pol_0) \wedge \forall r' \in opt_{-i}(\pi).\ r' \not\prec r\}$, so that $vault(\pi, i) = \max_{\subseteq} V$. Let $r \in opt(\pi) \cap req_i$, we prove that $V \neq \emptyset$. Since $r \in opt(\pi)$, we have $r \in adm(pol_0)$. Let $r' \in opt_{-i}(\pi)$; if by contradiction $r' \prec r$, we would have $r \notin opt(\pi)$. Hence, $r \in V$ and $vault(\pi, i) \neq \emptyset$. □

Theorem 4. *For all $pol \in \Theta$ and all providers i, the unique dominant attraction strategy for i w.r.t. pol is pol itself.*

Proof. First we prove that $pol_i = pol$ is a dominant attraction strategy (membership), then that for all strategies $req_i \neq pol_i$ there exists a full profile $\pi \in \Pi^i_{pol}$ such that $pw(\pi[i \leftarrow req_i], i) < pw(\pi, i)$, therefore req_i is not a dominant attraction strategy (uniqueness).

(Membership). Let π be a full profile. By the (contrapositive of) Corollary 3 and Lemma 2, it is straighforward to see that $pw(\pi, i) \leq pw(\pi[i \leftarrow pol_i], i)$. From Lemma 4, $res(\pi, i) \preceq_i res(\pi[i \leftarrow pol_i], i)$ always holds, therefore in particular when $pw(\pi, i) = pw(\pi[i \leftarrow pol_i], i)$.

(Uniqueness). Consider $req_i \neq pol_i$, for some $r \in req_i$ and $r' \in pol_i$, $r' \subset r$. Choose $\pi \in \Pi^i_{pol}$ such that *(i)* $pol_0 = \{r'\}$; *(ii)* for two providers $i \neq j$, the requests of i and j in π are $\{r'\}$; *(iii)* for all the other providers $k \neq i, j$, it holds $r' \notin req_k$. Clearly, as $r' \subset r$, $r \notin adm(pol_0)$ and since req_i is a threshold, for the other $r''(\neq r) \in req_i$, $r'' \notin adm(pol_0)$. This implies that $opt(\pi[i \leftarrow req_i]) \cap req_i = \emptyset$ and hence $pw(\pi[i \leftarrow req_i], i) = 0$. On the contrary, due to *(iii)*, $\{r'\} = opt(\pi)$ and hence $pw(\pi, i) > 0$. □

Theorem 5. *For all $pol \in \Theta$ and all providers i, the unique dominant investigation strategy for i w.r.t. pol is pol itself.*

Proof. (Membership) By analogy with Theorem 4, membership is a straightforward consequence of Corollary 3 and Lemma 4 and 3.

(Uniqueness). Assume a request req_i for the server i such that $req_i \neq pol_i$ and choose a full profile π as in Theorem 4. Since $opt(\pi[i \leftarrow req_i]) \cap req_i = \emptyset$, then $res(\pi[i \leftarrow req_i], i) = \emptyset$, whereas from $i \in cw(\pi)$ and Lemma 5, $res(\pi, i) \neq \emptyset$. Therefore, $res(\pi[i \leftarrow req_i], i) \prec_i res(\pi, i)$. □

Lemma 6. *For all $\pi \in \Pi$ and providers i, we have $opt_{-i}(\pi) \subseteq vault(\pi, i)$.*

Proof. Let $V = \{r \subseteq C \mid r \in adm(pol_0) \wedge \forall r' \in opt_{-i}(\pi).\ r' \not\prec r\}$. It holds $vault(\pi, i) = \max_{\subseteq} V$. Let $r \in opt_{-i}(\pi)$. By definition of $opt_{-i}(\pi)$, we have $r \in adm(pol_0)$ and $r' \not\prec r$ for all $r' \in opt_{-i}(\pi)$. Hence, $r \in V$.

It remains to prove that r is a maximal element of V. By contradiction, assume that $r' \in V$ is such that $r \subset r'$, which implies $r \prec r'$. Since $r \in opt_{-i}(\pi)$, this leads to the contradiction that $r' \notin V$, and we obtain the thesis. □

Theorem 6. *If all providers have the same policy and there are two truthful providers i and j, then $res(\pi, i) = res(\pi, j) = req_i \cap adm(pol_0) \subseteq pol_i$.*

Proof. First, notice that $opt(\pi) = opt_{-i}(\pi) = req_j \cap adm(pol_0) = req_i \cap adm(pol_0)$. Then,

$$\begin{aligned} res(\pi, i) &= vault(\pi, i) \cap ful(opt(\pi) \cap req_i) \\ &= vault(\pi, i) \cap ful(req_i \cap adm(pol_0)) \\ &= vault(\pi, i) \cap ful(opt_{-i}(\pi)). \end{aligned}$$

Now, if an element r of $vault(\pi, i)$ is included in an element r' of $opt_{-i}(\pi)$, it must be $r = r'$, because $r' \in vault(\pi, i)$ (by Lemma 6) and $vault(\pi, i)$ is a threshold. Therefore, we have $vault(\pi, i) \cap ful(opt_{-i}(\pi)) = opt_{-i}(\pi)$, and the thesis. □

Theorem 7. *If all providers have the same policy and provider i is truthful, then for all $j \neq i$ it holds $res(\pi, j) \subseteq pol_j$.*

Proof. Let $j \neq i$. Since server i is truthful, it holds $opt_{-j}(\pi) = opt(\pi) = pol_i \cap adm(pol_0)$. By definition, for all $r \in res(\pi, j)$ there exists $r' \in req_j \cap opt(\pi)$ such that $r' \subseteq r$. We prove that in this case it also holds $r' = r$. Assume by contradiction that r' is strictly contained in r. We have that $r \in vault(\pi, j)$ and hence $r' \notin vault(\pi, j)$, because $vault(\pi, i)$ is a threshold. By Lemma 6, from $r' \in opt(\pi) = opt_{-j}(\pi)$ it follows $r' \in vault(\pi, j)$, which is a contradiction. Hence, $r \in opt(\pi) \subseteq pol_i = pol_j$, and the thesis. □