Limiting Data Collection in Application Forms

A real-case application of a Founding Privacy Principle

Nicolas Anciaux[1,2] Benjamin Nguyen[1,2] Michalis Vazirgiannis[3,4]

[1] INRIA
Le Chesnay, France

[2] U. de Versailles St-Quentin
Versailles, France

[3] Athens U. of Economics & Business
Athens, Greece

[4] LIX, Ecole Polytechnique
Palaiseau, France

Abstract— **Application forms are often used by companies and administrations to collect personal data about applicants and tailor services to their specific situation. For example, taxes rates, social care, or personal loans, are usually calibrated based on a set of personal data collected through application forms. In the eyes of privacy laws and directives, the set of personal data collected to achieve a service must be restricted to the minimum necessary. This reduces the impact of data breaches both in the interest of service providers and applicants. In this article, we study the problem of limiting data collection in those application forms, used to collect data and subsequently feed decision making processes. In practice, the set of data collected is far excessive because application forms are filled in without any means to know what data will really impact the decision. To overcome this problem, we propose a reverse approach, where the set of strictly required data items to fill in the application form can be computed on the user's side. We formalize the underlying NP Hard optimization problem, propose algorithms to compute a solution, and validate them with experiments. Our proposal leads to a significant reduction of the quantity of personal data filled in application forms while still reaching the same decision.**

Privacy principle; Limited collection; Automated form filling.

I. Introduction

A massive digitalization of personal information is currently underway. Individuals are receiving an ever increasing amount of important documents in digital form (financial, professional, medical, relative to insurance, administrative, linked to daily consumption, etc.), issued by their employers, banks, insurances companies, civil authorities, hospitals, schools, ISP, telcos, etc. In parallel, secured online personal stores are emerging. The domain of the personal cloud is flourishing, and a recent report forecasts a $12 billion market[1]. Alternative offers propose storage facilities on the user's side with extended privacy controls, like for example Personal Data Servers [2] or Plug Servers (e.g., FreedomBox[2]).

This thriving market attests a reality: official documents are continuously accumulated and treasured by their owner. The reason is simple: legal obligations require them to be kept (e.g., 1 year for bank statements) and these documents are used as evidence when performing subsequent administrative tasks (e.g., paying taxes) or applying to services (e.g., bank loans).

In this paper, we consider the interaction between an applicant and a service provider, where the service provider requests personal information about the applicant to select the appropriate best offer. Such interactions occur whenever services are calibrated to adapt to the particular situation of each user. For example, the characteristics of a personal loan (rate, duration, insurance fee…) are defined based on decision making processes which use personal information such as income, employment, title deeds, personal references, forms of collateral, medical records, past lines of credits, etc. To cite other examples, contracting an insurance (health, car, job protection, etc.), social assistance, tax refund, or more generally any kind of personal information describing one's specific situation, to customize the offer that is made.

The necessity of evaluating the particular situation of an applicant is unquestionable and is in the interest of both the service provider and the customer. However, the requested set of personal information must be restricted to the minimum for two main reasons. First, the privacy of the applicant must be protected. Privacy legislations worldwide [13], [19] have enacted the *Limited Data Collection* (*LDC*) principle to this end, stating that collected sets of personal data must be strictly restricted to the minimum necessary to achieve the goal the user consents to. Second, the cost of potential information leakage must be reduced. Indeed, all too often personal data ends up being disclosed by negligence or hack. In 2011, the Open Security Foundation[3] reported more that a thousand data loss incidents affecting more than a hundred millions records. This is a financial disaster for the companies in charge of the data. A recent study [20] estimates the cost of data breaches at an average $7.2million per incident. Indeed, data breach laws enacted in many countries including 46 US states and the EU, compel companies to notify data owners in the event of data breaches, assist the victims in minimizing the impact of the data leak (e.g., canceling their credit card if the number has been disclosed) and often incur financial compensations. Security companies provide online breach cost calculators[4] to draw attention to this phenomenon: the more data exposed, the greater the cost in the event of a data breach.

The target of this paper is to restrict the set of information users have to expose to service providers, in accordance with the *LDC* privacy principle, and without impacting the evaluation of the decision making processes.

[1] The Personal Cloud: Transforming Personal Computing, Mobile, And Web Markets, Frank Gillett, a Forrester report, June 2011.

[2] See http://freedomboxfoundation.org/

[3] See http://www.datalossdb.org/reports

[4] See http://databreachcalculator.com.sapin.arvixe.com/

This is a difficult problem. In practice, the data useful or useless to make the decision cannot be distinguished *a priori* (at collection time). Such an assumption only holds for very simple cases, e.g., when ordering online, the address of the customer is mandatory to deliver the purchased items. However, in a general decision making system it does not hold. What data is useful to come to the decision of lowering the rate of the loan proposed to a user? Not only does the information harvesting depend on the purpose, it also depends on the data itself. Consider a reduction of rate based on either the salary or the assets of an individual. Revealing her income of *$30.000* if her age is below *25* may be enough. But an income of *$50.000* would suffice, regardless of age. Maybe *both* income and age values are useless if sufficient assets (e.g., greater than *$100.000*) can be justified. For a user with values u_1=[*income=$35.000*, *age=21*, *assets=$10.000*] the minimum data set would be [*income*, *age*]. For a user with u_2=[*income=$40.000*, *age=35*, *assets=$250.000*] it would be [*assets*]. Hence, a bank cannot specify a minimum set of attributes needed to make its decision since this decision depends on looking at the entire attributes available. Fixing the data to be collected *a priori* inevitably leads to over-estimating the data to be collected.

The common procedure when a server has to evaluate a decision making process is thus to request the users to fill in application forms which cover all the information which *may turn out to be of use* at some point in the decision process. This obviously does not comply with the *LDC* principle, since service providers collect personal data which may *not* impact the final decision to be taken. Our study focuses on the strict compliance with the *LDC* principle in this context. Our approach is based on a reverse implementation of the traditional *LDC* strategy, where users are given enough knowledge about the underlying decision making process to determine locally the minimum set of data items to fill in to achieve the expected service with maximum benefit.

To the best of our knowledge, all existing techniques addressing *LDC* principle fix the data to be collected a priori, leading to collect too much data. A few recent works in the domain of credential based access control can be viewed as vanguards in the application of the *LDC* principle. However, the underlying techniques cannot be used to solve our problem, mainly because of scalability issues. We give details about the positioning of our work in Section VI.

The contribution we make in this paper is threefold:

(i) we formalize the *Minimum Exposure* approach in the case of a decision making process;

(ii) we state the underlying optimization problem and study its complexity; and

(iii) since the problem is NP-hard we propose approximation algorithms and validate them with experiments.

The paper is organized as follows. Section II gives the general scenario, and presents the running example. In Section III, we state the *Minimum Exposure* optimization problem and study its complexity. Several algorithms are introduced in Section IV, and validated in Section V. Section VI discusses related works and Section VII concludes.

II. SCENARIO FOR MINIMUM EXPOSURE

A. *General Scenario*

We consider the general scenario depicted in Figure 1 which involves three main parties: Data Producers, Users, and Service Providers. **Data Producers** act as data sources. They include for example banks, employers, hospitals, or administrations. The information they deliver to users can be signed to prove integrity and origin (e.g., salary forms, bank records history, tax receipts, etc.). **Users** store the documents they receive in their personal spaces. We make no hypothesis on users' personal space, which could be their own PC, cloud storage, or secure devices, etc. **Service Providers** may include banks or insurances companies, but also public welfares or administrations. They propose services, which may include bank loans, health insurance or social benefits, which require users' personal information to evaluate the decision making processes and calibrate the offer made to the applicant.

In practice, Service providers issue application forms to collect the data which may impact their decision. Huge amounts of data items may be requested, depending on the context. For example, loan applications may include mortgage application forms, which commonly collect hundreds of personal data items[5], and social care applications require equivalently large forms to be filled in.

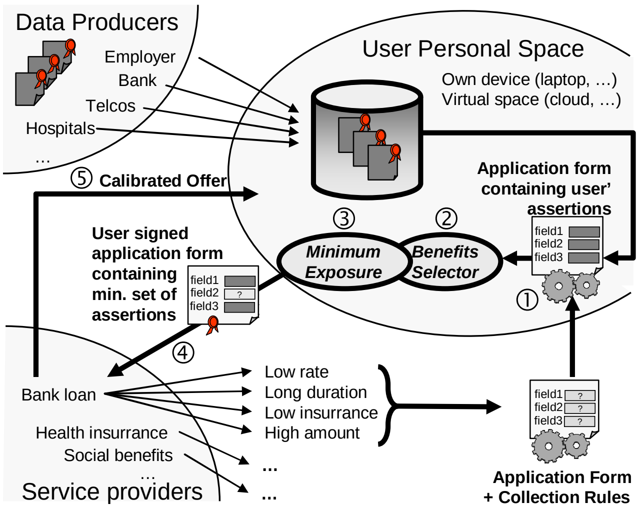

Figure 1. General architecture enabling Minimum Exposure.

We promote in this paper a new approach where the server has to provide both the application form and a set of data *collection rules*. Those collection rules enable the users to select among the data items requested in the application form the minimum required set to be filled in. We call *Minimum Exposure* (*ME*) the process which identifies the minimum subset of assertions to be exposed by a User to a Service Provider to trigger the desired service with the set of advantages she can (and wants) to obtain. *ME* requires

[5] See for example the mortgage application form of Nationwide Building Society (the largest building society in the world) as a good representative: http://www.nationwide.co.uk/nr/rdonlyres/a48ffc87-7e29-4ea6-b24d-2720746c5d9e/0/m1inov06.pdf

confronting the set of assertions that can be made by the User, with the advantages associated with the set of collection rules describing the information requested by the Service Provider.

The execution of *ME* must take place on the user's side or on any trusted third party, to comply with the *LDC* principle. Indeed, the system in charge of running *ME* needs to collect more data than the minimum subset computed by *ME*.

The general scenario pictured on Figure 1 is as follows: when a user wants to apply to a service, she ① downloads the application forms and the collection rules provided by the service provider and fills in the form given the documents she owns, ② uses the *Benefit Selector* to locally compute the advantages she can obtain based on the completed application form and the collection rules, and selects among these advantages the ones she desires to obtain, ③ runs a *Minimum Exposure* process to compute the minimum set of useful assertions (i.e., data items provided) in the application form to obtain the service with the selected advantages, ④ validates and signs the application form with this minimum set of assertions and sends it to the service provider. The service provider can ⑤ run its decision processes based on the content of the form and calibrate the offer made to the applicant. The declaration of the applicant is then stored by the service provider (the delay depends on the context). After point ④ the applicant can be asked to provide certified documents to prove the assertions provided in her form. Remark that in many cases the verification phase only occurs unlikely, e.g, medical declarations linked to car or loan insurance are often only checked in the event of a claim, income taxe returns in the case of a tax audit, etc.

B. *Setting*

1) *Collection Rules*

The collection rules describe the information required by the service provider and the advantages associated with it.

Decision making processes are generally based on *white box* mechanisms (i.e., comprehensible by humans and justifiable) and are thus public. This is the case of administrative applications (e.g., tax services, social and health care where the decision process is either publicly documented or considered as common knowledge), and law abiding commercial applications. Indeed, as shown in [15], a white box requirement is imposed by law and/or for user acceptance of the process in many domains. For example, laws such as the US "Equal Credit Opportunity Act" impose white box for credit scoring[6]. This is the same for many medical systems. Recent studies like [7] even transform "black box" decision models like SVM or neural networks into white box ones.

In addition, decision making rules may be complex in practice, e.g., loans are granted based on decision trees, SVM or neural networks [12]. The collection rules must be expressive enough to successfully reflect the decision making process of the service provider. In this paper, we consider sets of collection rules, each one being modeled using disjunctions of conjunctions of constraints on attribute-value pairs. This is a comprehensible rule-based model, which is very expressive since it covers the widely used decision tree model [17], as well as forests of decision trees (i.e., covering multi-dimensional decisions). For example, an organization offering loans may include a dimension in its decision making rules to favor families and young students by subsidizing a part of the loan as a *Non Interest Loan* (*NILo*), expressed by the following rule:

$$NILo\text{: } (married=true \wedge children>0) \vee (age<30 \wedge Edu='Univ')$$

We assume that no-one can force users to transmit assertions. The only penalty is to prevent them from obtaining advantages. Therefore, rules must be *positive*, in the sense that it is beneficial for a user to trigger them. This is not a limitation of the model since rules leading to constraints that prevent the grant of services (called *negative* rules) can be constructed by integrating the negation of the rule into the collection rule set. For example, if the *NILo* mentioned above is *not* granted to people with a police record ($police\_record='YES' \Rightarrow \neg NILo$), the rule can be written:

$$NILo\text{: } (married=true \wedge children>0 \wedge police\_record='NO') \\ \vee (age<30 \wedge Edu='Univ' \wedge police\_record='NO')$$

2) *Users' Assertions and Documents*

We distinguish two types of documents: assertion documents (simply called assertions) used to fill in application forms; and official documents signed by data producers and kept by users to be shown if service providers request proving assertion validity. Application forms are signed as a whole by applicants and official documents are signed by data producers at various granularities. In case of verification of an assertion, e.g., *income = $50.000*, the user may disclose an official document containing this value, e.g., the income tax receipt.

For the sake of simplicity and without lack of generality, we consider in this paper that each assertion is an inseparable (*attribute*, *value*). This is the format in which application forms generally require personal information. We show in [3] that our model can also be extended to *attribute* θ *value* assertions, with θ the comparator <, ≤, =, ≠, ≥, or >, leading to expose even less information with respect to collection rules. Regarding official documents, most are currently signed as a whole, but there is no technical difficulty to sign (*attribute*, *value*) pairs or *attribute* θ *value* separately as show in [8]. Thus, we also consider that each assertion *d* can be proven by a given official documents *d'* without leaking more information than *d*. We provide in [3] a discussion when this hypothesis is not fulfilled.

3) *Metrics to Evaluate the Degree of Exposure*

The minimization of the set of assertions resulting from the application of the *ME* algorithm can be appreciated in terms of reduction of the data (assertions) exposed, harmful to both user (in terms of privacy) and service provider (in terms of financial cost). We consider on the one hand that the privacy harm associated with a dataset is proportional to the usefulness of that dataset, and on the other hand that the financial cost of a data breach for service providers is directly proportional to the quantity of exposed data. The financial cost for service providers is determined by two dominant factors [20]. First, the *ex-post response* represents 20% of the cost. It includes the actions taken by the company to provide assistance to the

[6] See http://www.ftc.gov/bcp/edu/pubs/consumer/credit/cre15.shtm

victims in the necessary procedures conducted to minimize the harm: the greater the exposure, the greater the harm. Second, *lost business* (50% of data breach cost) is the direct consequence of the negative publicity associated with the data breach incident headings: the greater the exposure, the worse the publicity.

These components of the breach costs are thus tightly linked with information loss, both for users and service providers. Many information loss metrics exist (e.g., *minimal distortion* [21], [22] or *ILoss* [23]). They all associate an exposure value to each dataset item independently; our approach can be used with any such metric.

C. Running Example

We introduce here a loan scenario, used as a running example (see Table I) throughout the paper. The example has deliberately been simplified, since real loan application forms may include mortgage/medical forms that collect hundreds of personal data items.

An institution proposes, to any applicant, personal loans of *$5.000* at *10%* rate with *1* year duration and a *$50* per month insurance cost for job loss protection. But, a higher loan of *$10.000* can be offered to wealthy customers fulfilling the following requirement:

$$(\mathit{income}>\$30.000 \wedge \mathit{assets}>\$100.000)$$
$$\vee\ (\mathit{collateral}>\$50.000 \wedge \mathit{life\_insurance}=\text{'}\mathit{yes}\text{'})$$

This leads to the first collection rule r_1 given in Table I. Collection rule r_2 enables obtaining a loan granted at only *5%* rate for families and low risk factor young people; collection rule r_3 expresses that loans can be granted for an extended duration of *2* years to high revenues families and to low risk people; and rule r_4 states that the insurance cost for job loss protection can be proposed with a *30%* discount to rich families and promising young workers. Those collection rules are made of a disjunction of conjunction of predicates p_i of the form *attribute* θ *value* with θ the comparator <, ≤, =, ≠, ≥, or >. They are given with the corresponding application form in table I.

The user can assert that she is married, 25 years old, with one child, a $35.000 year income, a university degree, $5.000 goods as collateral, an income tax at 11.5% rate, a life insurance. She also claimed only $250 last year. This information is summarized by a set of *attribute = value* assertion termed as_1 to as_{10} in Table I such that $as_i \Rightarrow p_i$. This user could then activate the complete set of advantages c_1 to c_4. The *ME* algorithm has to identify the minimum set of assertions allowing this.

III. THE MINIMUM EXPOSURE PROBLEM

This section first states the *Minimum Exposure* problem more formally and studies its complexity.

A. Problem Statement

We denote by |*S*| the cardinality of a set *S*. We introduce below the other required definitions, and then state the problem. We illustrate the notions using the example in Table I.

TABLE I. FORM, RULES AND ASSERTIONS FOR THE LOAN SCENARIO.

Collection rules:		
	r_1: $(p_1 \wedge p_2) \vee (p_3 \wedge p_4)$	$\Rightarrow c_1$
	r_2: $(p_5 \wedge p_6 \wedge p_7) \vee (p_4 \wedge p_8 \wedge p_9)$	$\Rightarrow c_2$
	r_3: $(p_1 \wedge p_6 \wedge p_7) \vee (p_2 \wedge p_4 \wedge p_{10})$	$\Rightarrow c_3$
	r_4: $(p_2 \wedge p_5 \wedge p_6 \wedge p_7) \vee (p_1 \wedge p_4 \wedge p_8 \wedge p_9)$	$\Rightarrow c_4$
with	p_1: *year_inc>$30.000,*	p_2: *assets>$100.000,*
	p_3: *collateral>$1.000,*	p_4: *life_insurance='yes',*
	p_5: *tax_rate>10%,*	p_6: *married=true,*
	p_7: *children>0,*	p_8: *edu='university',*
	p_9: *age<30,*	p_{10}: *insurance_claims<$5.000.*
and	c_1*=high_loan,*	c_2*=5%_rate,*
	c_3*=long_loan,*	c_4*=low_insurance.*
Application form: year_inc?, collateral?, tax_rate?, children?, age?, assets?, life_insurance?, married?, edu?, insurance_claims?		
User's assertions:		
	as_1: *year_inc=$35.000,*	as_2: *assets=$150.000,*
	as_3: *collateral=$5.000,*	as_4: *life_insurance='yes',*
	as_5: *tax_rate=11.5%,*	as_6: *married=true,*
	as_7: *children=1,*	as_8: *edu='univ',*
	as_9: *age=25,*	as_{10}: *insurance_claims=$250.*

1) Definitions

Attributes. Let $A = \{a_i\}$ represent a finite set of attributes. Each attribute a_i has an associated domain $dom(a_i)$.

Classes. Let $C = \{c_j\}$ represent a finite set of Boolean variables, interpreted as *positive* classes to which users can belong. If c_j*=true* for a given user, this means she can obtain the advantage associated with c_j.

Predicates. We call *predicate over A* any expression of the form $a\theta v$ where $a \in A$, $v \in dom(a)$ and $\theta \in \{=, <, >, \leq, \geq, \neq\}$.

Example: p_1*: year_inc>$30.000* is a predicate.

Assertions. Let as_i represent an assertion composed of a single equality predicate over *A*. We denote by $Data_u=\{as_i\}$ the set of assertions a given user *u* can state truthfully (i.e. she owns a signed document proving this assertion). We say that an assertion as$_i$ *proves* a predicate *p* if $as_i \Rightarrow p$.

Atomic Rules. An atomic rule leading to class $c_{j,}$, denoted by $atom_j$ is a conjunction of predicates such that $atom_j=true \Rightarrow c_j=true$. Since there are usually several atomic rules leading to a class c_j we write $atom_{j,k}$ using *k* to distinguish them.

Example: $atom_{1,1}$*:* (*year_inc>$30.000* ∧ *assets>$100.000*) and $atom_{1,2}$*:* (*collateral>$50.000* ∧ *life_insurance='yes'*) are two atomic rules leading to class c_1.

We say that a set of assertions $Data_u=\{as_i\}$ proves an atomic rule $atom_{j,k} = \bigwedge_m q_{j,k,m}$ where $q_{j,k,m}$ is a predicate over *A*, if and only if $\forall j,k,m\ \exists\ i : as_i \Rightarrow q_{j,k,m}$ and *uniquely proves* $atom_{j,k}$ if and only if $\forall j,k,m\ \exists!\ i : as_i \Rightarrow q_{j,k,m}$.

Example: $data_u=\{as_1, as_2, as_3, as_4\}$ uniquely proves atomic rules $atom_{1,1}$ and $atom_{1,2}$.

Collection Rules. A collection rule r_j is a disjunction of atomic rules leading to class c_j. More formally: r_j: $\bigvee_k atom_{j,k}$. If a signed set of assertions $Data_u$ proves an atomic rule $atom_{j,k}$ then we say that $Data_u$ *proves* r_j, which means that user *u* can benefit from the advantage associated with c_j (obviously, $r_j=true \Rightarrow c_j=true$).

Example: r_1: (*year_inc>\$30.000* $\wedge$ *assets>\$100.000*) $\vee$ (*collateral>\$50.000* $\wedge$ *life_insurance='yes'*) is a collection rule leading to class *high_loan*.

In what follows, we write $r_j = \vee_k (\wedge_m \; q_{j,k,m})$ where $q_{j,k,m}$ is a predicate over A. Considering r_1 in the previous example, we have $q_{1,1,1}$: *year_inc>\$30.000*, $q_{1,1,2}$: *assets>\$100.000*, $q_{1,2,1}$: *collateral>\$50.000* and $q_{1,2,2}$: *life_insurance='yes'*.

Rule Set. Let $R = \{r_j\}$ represent a set of $|C|$ *collection rules*, one for each class c_j. If $Data_u$ (uniquely) proves all the rules of R then we say that $Data_u$ (uniquely) proves R.

Rule Set Boolean Formula. Since only one (verifiable) assertion uniquely proves a given predicate used in the rules, deciding whether $Data_u$ proves the rule set R is *equivalent* to testing the truth-value of a Boolean formula $E_R = \wedge_j (\vee_k (\wedge_m b_{f(j,k,m)}))$ called Rule Set Boolean Formula associated to R, where $f(j,k,m)$ is a function of domain $[1;|A|]$ defined by $[f(j,k,m)=i$ such that $as_i \Rightarrow q_{j,k,m}]$ and $b_{f(j,k,m)}$ is a Boolean variable which is *true* if $as_{f(j,k,m)}$ is exposed and *false* otherwise. Note that if we consider the truth assignment that sets all values $b_{f(j,k,m)}$ to *true*, then $E_R=true \Leftrightarrow Data_u$ *proves* R.

Example: Table II illustrates a Rule Set Boolean Formula based on R defined in Table I.

Exposure metric. Let $B=\{b_i\}$ represent a set of Boolean variables. Let T_B represent a truth assignment of these variables such that $b_i=true \Leftrightarrow$ as$_i$ is disclosed to the service provider. We note $\mathbf{EX}(T_B)$ a function representing the exposure of the associated assertions set disclosed. Exposure is proportional to financial cost for service providers, and privacy harm for users.

Example: The function $\mathbf{EX}(T_B) = |\{ b_x \in B\text{: } T_B(b_x)=true \}|$ that counts the number of assertions disclosed can be used as an exposure metric. Note that any metric that is invariant over time, given a truth assignment T_B can be used. In particular, this includes information loss metrics, which can be assumed proportional to **EX**. Henceforth, if an assertion is disclosed, we say that it is *exposed*.

Boolean Minimum Exposure Problem. We can now define the Minimum Exposure decision problem of a set of assertions $Data_u$ with regards to a rule set R and an exposure metric **EX**. Note that we suppose that $Data_u$ proves R. Should this not be the case, we would simply use R' the subset of rules of R proven by $Data_u$. Our goal is to find a truth assignment T_B of the Boolean variables associated to the disclosure of the assertions minimizing their exposure computed using the above exposure metric.

The *Boolean ME* decision problem:
Given a rule set R, $Data_u=\{as_x\}$ a set of q assertions that uniquely prove R, B a set of Boolean variables $B=\{b_1,..,b_q\}$ such that $b_x=true \Leftrightarrow as_x$ is exposed, $E_R=\wedge_j (\vee_k (\wedge_m \; b_{j,k,m}))$ where $\forall j,k,m \; b_{j,k,m} \in B$ the rule set formula associated to R, and the exposure function **EX,** $Data_u$ is *n-exposable* with regards to R if and only if there exists a truth assignment T_B of B such that $\mathbf{EX}(T_B) \leq n$ and E_R is *true*.

We study the related optimization problem, whose goal is to minimize n.

TABLE II. RULE SET BOOLEAN FORMULA FOR THE LOAN SCENARIO

$B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}\}$ such that:
$\forall i \in [1;10]$, $b_i=true \Leftrightarrow as_i$ *is exposed.*
The Rule Set Boolean Formula E_R is as follows:
$E_R = \;\; ((b_1 \wedge b_2) \vee (b_3 \wedge b_4))$
$\wedge ((b_5 \wedge b_6 \wedge b_7) \vee (b_4 \wedge b_8 \wedge b_9))$
$\wedge ((b_1 \wedge b_6 \wedge b_7) \vee (b_2 \wedge b_4 \wedge b_{10}))$
$\wedge \;\; ((b_2 \wedge b_5 \wedge b_6 \wedge b_7) \vee (b_1 \wedge b_4 \wedge b_8 \wedge b_9))$
Suppose that the user can only truthfully state assertions 1-9 we prune out all the classes and atomic rules that can not be proven:
$E_R = \;\; ((b_1 \wedge b_2) \vee (b_3 \wedge b_4))$
$\wedge ((b_5 \wedge b_6 \wedge b_7) \vee (b_4 \wedge b_8 \wedge b_9))$
$\wedge ((b_1 \wedge b_6 \wedge b_7)$
$\wedge ((b_2 \wedge b_5 \wedge b_6 \wedge b_7) \vee (b_1 \wedge b_4 \wedge b_8 \wedge b_9))$

TABLE III. ALGORITHM NOTATIONS USING THE LOAN SCENARIO

$D=|Data_u|=10$; $C=4$;
B[] is an array of Booleans of size D such that:
$\forall i \in [1;10]$, $B[i]=true \Leftrightarrow as_i$ is *exposed*
R[] is an array of C collection rules;
R[j].atom[] for $j \in [1;4]$ are arrays of *2* atomic rules; *R[j].atom[k].b[]* with $j \in [1;4]$, $k \in [1;2]$ are arrays of references to *B[i]* elements. We denote by **B[i]* a reference to *B[i]*. *R[j].atom[k].b[m]*are set as follows:
*R[1].atom[1].b[1]←*B[1]*; *R[1].atom[1].b[2]←*B[2]*;
(…) *R[2].atom[2].b[2]←*B[8]*; *R[2].atom[2].b[3]←*B[9]*; (…)
*R[4].atom[2].b[3]←*B[8]*; *R[4].atom[2].b[4]←*B[9]*;

B. *Complexity Results*

The *ME* problem defined above is NP-Hard (proof is a reduction to the min weighted SAT problem omitted due to lack of space, and can be found in [3]). In addition, we show in [3] that the *ME* optimization problem is not in APX[7], and has a differential approximation[8] ratio of 0-DAPX[9]. This is a negative complexity result in the sense that it shows that the problem is difficult and that polynomial approximation algorithms will provide bad approximation guarantees in the worst case. In Section IV, we examine the problem by (experimentally) exploring the domain where it is possible to provide an exact resolution using a state of the art solver. When such a resolution is too long to compute, we rely on polynomial approximation algorithms.

IV. SOLUTIONS OF THE *ME* PROBLEM

In this section, we provide exact and approximation algorithms to compute a solution of the *ME* problem. For the exact resolution, we use a *Binary Integer Programming* (BIP) state of the art solver. For the approximate resolution, we propose a naïve random algorithm, a simulated annealing based meta-heuristics algorithm, and a specific heuristic algorithm.

[7] The APX class is the set of NP optimization problems that allow polynomial-time approximation algorithms with an *approximation ratio* bounded by a constant.

[8] Given an instance I of an optimization problem, and a feasible solution S of I, we denote $m(I,S)$ the value of solution S, $opt(I)$ the value of an optimal solution of I and $W(I)$ the value of a worst solution of I. The *differential approximation ratio* of S is defined by $DR(I,S)=abs((m(I,S)–W(S))/(opt(I)–W(I))$. The traditional approximation ratio for a minimization problem is simply defined by $m(I,S)/opt(I)$.

[9] 0-DAPX is the class of NP optimisation problems for which all polynomial approximation algorithms have a *differential approximation ratio* of 0.

In all algorithms, we consider a Boolean formula E_R constructed as explained in Section III using a rule set R composed of a set of C collection rules associated with classes (or benefits) that the user can (and wants) to claim, and where each atomic rule can be proven using her assertions. Atomic rules that cannot be proven are removed via *Benefits Selector* (see Figure 1, step ②) before constructing R. The size of $Data_u$, i.e., the set of assertions related to the rule set, is noted D.

T_B is a truth assignment function to $Data_u$ that we implement as an array of Booleans with the semantics *B[i]=true* ⇔ *as_i is exposed*. The rule set is represented as an array *R[]* of C collection rules, each collection rule *R[i]* being an array *atom[]* of atomic rules, each atomic rule *R[i].atom[j]* being an array *b[]* of references to the elements of B (see example in Table III). Note that E_R is *true* when each collection rule *R[i]* has at least one atomic rule where all referenced Boolean elements are *true*.

A. Exact Resolution (BIP model)

We propose to use a state of the art BIP solver, generally termed as *Mixed Integer Non-Linear Program* (MINLP) solver, to produce an exact result. We have chosen the popular and open source *COUENNE* solver [9] to this respect.

In order to use a MINLP solver, an instance of the problem must be written as a MINLP program. This is a direct transformation where each assertion corresponds to a Boolean variable, where the objective function is simply the sum of all the variables, and in which we express one *non-linear* constraint per collection rule r_j: $\Sigma_k \Pi_m as_{j,k,m} \geqslant 1$

The running example presented in Section II.C can be expressed by the following program, written in *AMPL* [14].

```
var b1 binary; ... var b10 binary;
minimize EX:
b1+b2+b3+b4+b5+b6+b7+b8+b9+b10;
subject to
r1: b1*b2 + b3*b4 >= 1;
r2: b5*b6*b7 + b4*b8*b9 >= 1;
r3: b1*b6*b7 + b2*b4*b10 >= 1;
r4: b2*b5*b6*b7 + b1*b4*b8*b9 >= 1;
```

The program is then fed to the BIP solver. As shown in Section V, the range of parameters for which the BIP solver computes the solution in an acceptable time (under 2h) is small.

B. Approximate Solutions (Polynomial Time)

We need to revert to a polynomial time approximation in order to compute results for the instances of the problem that cannot be tackled within reasonable time by the solver. We propose three algorithms: a naïve fully random algorithm called *RAND**, a simulated annealing meta-heuristics based algorithm called *SA**, and an algorithm called *HME* using a heuristic specially designed for the *ME* problem. These algorithms are non deterministic, therefore they can be run many times and the best solution is kept. However, they produce their first result in linear or polynomial time, depending on the algorithm. We discuss the complexity of the algorithms on a single run, to compare their speed. To compare their quality, we run the longest algorithm (*HME*) once, and we execute the other algorithms (*RAND** and *SA**) as many times as necessary, until they run out of processor time.

1) Fully random algorithm (RAND*)

The fully random algorithm *RAND** is based on a random choice of rules, and serves as a baseline. *RAND** randomly chooses one atomic rule for each collection rule and sets to *true* the value of each Boolean in B that this atomic rules refers to. Since each class is covered, the corresponding set of assertions determined by the truth assignment T_B is a solution to the *ME* problem instance. The result is the solution found within the allocated time limit for which **EX** is minimum (best result).

2) Simulated Annealing Algorithm (SA*)

Meta-heuristics are used in optimization problems in order to guide the algorithm towards better solutions, instead of simply randomly selecting them. We consider here simulated annealing [16] and introduce the *SA** algorithm to serve as a representative for meta-heuristic guided algorithms. Before each run, *RAND** is executed once to provide a starting solution, to feed *SA**. This solution is improved by *SA**, and if given enough processor time, *SA** can restart. The pseudo code of *SA** can be found in [3]. *Both RAND** and *SA** algorithm provide a solution in polynomial (linear) time of complexity.

3) The HME Algorithm

The Heuristic for Minimum Exposure (*HME*) algorithm that we propose uses a specific heuristic for the *ME* problem. The heuristic lies in the computation of *score[i]* the score of the i^{th} Boolean entry in B, using the function *fix*(B). This function computes a lower bound of the value of **EX**, by computing the number of predicates that can no longer be set to *false* for the given B. For instance, suppose that *B[i]=false* (i.e., as_i is not exposed). All the atomic rules referring to *B[i]* cannot be proven anymore. This leads to the fact that **EX** will be greater (or equal) to the value of the cardinality of the set of predicates in the atomic rules that are the only ones left to prove a given class. Using the running example (see Section II.C), we illustrate the computation of *fix(B)* in Table IV for each Boolean entry at each step of the algorithm. Let us briefly see how *score[1]* and *score[3]* are computed for the first step. If *B[1]=false*, then we have to prove collection rules *R[1]*, *R[3]*, *R[4]* using respectively atomic rules *R[1].atom[2]*, *R[3].atom[2]*, *R[4].atom[1]* (i.e., which means setting to *true* the *7* Booleans *B[2]*, *B[3]*, *B[4]*, *B[5]*, *B[6]*, *B[7]*, *B[10]*), leading to *score[1]=7*. If *B[3]=false*, this means proving *R[1]* using *R[1].atom[1]* (i.e., set to *true* the *2* Booleans *B[1]*, *B[2]*), therefore *score[3]=2*. We show in grey the lowest score, which means a truth assignment set to *false* in next steps, indicated by the symbol –. Assertions for which the score is denoted by ∞ are those for which the final truth assignment is set to true. The final result is here B=[*B[1]=true*, *B[2]=true*, *B[3]=false*, *B[4]=false*, *B[5]=true*, *B[6]=true*, *B[7]=true*, *B[8]=false*, *B[9]=false*, *B[10]=false]* which happens to be the minimal value of **EX** on this instance of the problem.

We see that the cost of *HME* algorithm is proportional to $O(\mathrm{COST}_{\mathrm{FIX}} \times D^2)$, where $\mathrm{COST}_{\mathrm{FIX}}$ is the cost of computing the *fix* function. More precisely, in our implementation, $COST_{FIX} = O(|R| \times d_C \times d_{QD})$, where $|R|$ is the number of collection rules, d_C is the number of atomic rules per collection rule and d_{QD} is the number of predicates per atomic rule.

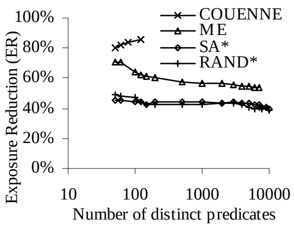

Figure 2. *ER* varying the number of distinct predicates.

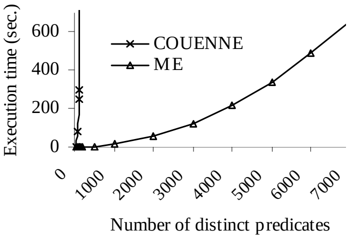

Figure 3. Execution time varying the number of distinct predicates.

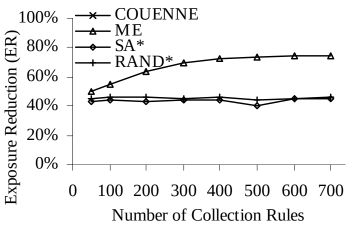

Figure 4. *ER* varying the number of collection rules.

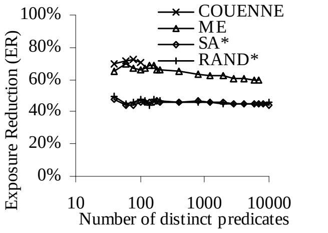

Figure 5. *ER* varying the number of predicates & collection rules

***HME* algorithm**

```
Input:  E_R a Rule Set Boolean Formula
Ouput:  B a truth assignment of the assertions that
        proves R
```

```
for i = 1 to D do
  B[i] ← true
endfor
while (exists i such that: B[i]=true and
          if B[i] is set to false then E_R (B) remains true)
  for i = 1 to D do
    score[i] ← ∞
  endfor
  forall i such that B[i] = true do
    B[i] ← false
    if E_R (B)=true then // E_R (B) is true iff B proves R
      score[i] ← fix(B)
    endif
    B[i] ← true
  endforall
  m ← i such that score[i] is minimum
  B[m] ← false
endwhile
return B
```

The intuition behind the heuristic is to successively get rid of the assertions which require keeping the least number of other assertions (such that all benefits are preserved) among the remaining ones. This heuristic is particularly relevant when the number of atoms per collection rule is small. Our performance evaluation confirms this scope. Note that if this number increases then *HME* tends towards *RAND**.

We show in Section V that the *HME* algorithm provides very good results in terms of quality of approximation, while maintaining reasonable computational complexity. We show in [3] that the *HME* algorithm can be extended to assertions with *attribute* θ *value* predicates, with θ ∈ {<, ≤, =, ≠, ≥, >}.

TABLE IV. EXECUTION OF THE *HME* ALGORITHM

Steps	*B[1]*	*B[2]*	*B[3]*	*B[4]*	*B[5]*	*B[6]*	*B[7]*	*B[8]*	*B[9]*	*B[10]*
1: score[i]	7	7	**2**	5	4	6	6	4	4	3
2: score [i]	∞	∞	–	5	5	6	6	5	5	**4**
3: score [i]	∞	∞	–	**5**	7	∞	∞	5	5	–
4: score [i]	∞	∞	–	–	∞	∞	∞	**5**	5	–
5: score [i]	∞	∞	–	–	∞	∞	∞	–	**5**	–
Final B[i]	*true*	*true*	*false*	*false*	*true*	*true*	*true*	*false*	*false*	*false*

V. EXPERIMENTS

In this section, due to lack of space we only briefly present an experimental validation of our approach on synthetic data. Experiments were conducted on a HP workstation with 3.1GHz Intel CPU and 8GB RAM running Java1.6 (x64). The *COUENNE* solver was run on the same machine. Algorithms, data and BIP model generator code are available at http://project.inria.fr/minexp/

Results concern both scalability and quality. The quality is measured by computing the reduction of the set of exposed assertions in the application form, as follows:

Exposure Reduction: $\boldsymbol{ER}(T_B)=1-\mathbf{EX}(T_B)/|B|$,.

There are many parameters to the problem (number of collection rules, atomic rules per collection rule, number of distinct predicates/assertions, etc.), but many are linked. We fix the number of predicates per atomic rule (assuming that a user can prove all predicates through assertions, this is equivalent to fixing $Data_u$) and the number of atomic rules per collection rule to 4. This corresponds to the values of real decision trees extracted from neural networks used for credit scoring process [7]. To analyze exposure reduction and scalability, we generate rule sets varying the number of rules $|R|$ and the number of distinct predicates D in these rules (which is equivalent to the number of assertions related to the rule set). The number of atomic rules $|Q|$ varies accordingly since $|Q|=|R|\times4$. We set a time limit of 2 hours for the exact solver and of 10 minutes for the approximation algorithms. The results are presented in Figures 2 to 5.

We draw three main conclusions from these experiments. First of all, the exposure reduction is important even with very simple algorithms (see RAND*), ranging from 30% to 80%, and is on average of 70% in the area of applicability of the exact solver. This means that on average only 30% of a user's data items is sent when using *ME* compared to the traditional case. Second, the scope of the exact solution is limited, and therefore the use of approximation algorithms is unavoidable. Third, *HME* provides the best results of the approximation algorithms, outperforming them by about 10%, and scales in polynomial time with regards to *D*.

VI. RELATED WORK

The transposition of legal privacy principles into privacy aware systems has fostered many studies. Emblematic examples include the P3P Platform for Privacy Preferences [11], privacy policy languages like EPAL [6] and Hippocratic databases [1]. P3P highlights conflicting policies, but it offers no means to calibrate the data exposed by a user and achieve *Limited Data Collection* (LDC). Many other policy languages have been proposed for different application scenarios, like EPAL [6], XACML [18] or WSPL [4], but to the best of our knowledge, no language has been introduced with *LDC* in mind. Another emblematic study deals with Hippocratic databases [1]. The architecture of a Hippocratic database is based on ten guiding privacy principles including *LDC*. It addresses *LDC* by maintaining the set of attributes that are required for achieving each declared purpose. However, this solution assumes useful and useless data for a given purpose can be distinguished at the time of the data collection. As mentioned in the introduction, this assumption only holds for simple cases, but not in general decision making processes.

Existing works closer to our study can be found in the area of automated trust negotiation and credential based access control, where access decisions are based on the gradual confrontation of an access control policy with a set of credentials. A few number of works including [5, 10, 24] can be considered as following a *minimum exposure* approach. All those works minimize the privacy leak of a set of personal data items (credentials) while enabling a given decision to be made (the grant or deny access decision). However, the problem and solutions are different from ours for two founding reasons. First, the decision making processes that we consider are more complex than access control. The collection rules in *ME* can model sets of decision trees classifiers: several dimensions can be considered (e.g., lower credit rate, longer duration, lower cost of insurance, larger portion of 0% loan, etc.) each one potentially impacting the final offer made to the applicant. Second, in our context, the decision making process requires by nature a huge amounts of personal data (e.g., to obtain a loan offer customers are asked to fill in forms with hundreds to thousands fields), while in access control only a few credentials are considered (e.g., up to 35 in [5]). The results of these works can therefore not be used in our context, because they fall short on both expressivity and scalability requirement.

VII. CONCLUSION

In this article, we have introduced the *Minimum Exposure* approach and the related *ME* problem. We have shown how it can be expressed in the form of a Boolean minimum weighted satisfiability problem. We have studied the scope of applicability of general operational research solutions, using a state of the art MINLP solver. For cases where an exact resolution was not applicable, we have proposed several algorithms to compute an approximation of the solution. In all cases, we have shown that the exposure reduction that can be achieved compared with traditional implementations of limited data collection is around 50% in the average. These benefits are not only interesting for the user, whose privacy is less exposed, but also for the service providers who can limit their losses in the event of a data breach. Our hope is to open a new avenue for interesting applications of the minimum exposure principle introduced in this paper.

VIII. ACKNOWLEDGMENTS

This work was supported by the KISS grant ANR-11-INSE-0005, by the DIGITEO LeTeVoNe grant, and by the INRIA Collaborative Action on the Protection of Privacy in the Information Society (CAPPRIS) grant.

REFERENCES

[1] Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. Hippocratic databases. In *Proceedings of VLDB*, 2002.

[2] Allard, T., Anciaux, N., Bouganim, L., Guo, Y., Le Folgoc, L., Nguyen, B., Pucheral, P., Ray, I., Ray, I., and Yin, S. Secure Personal Data Servers: a Vision Paper. In *VLDB Endowment, 3*(1), 2010.

[3] Anciaux, N., Nguyen, B., and Vazirgiannis, M. Minimum Exposure in classification scenarios. INRIA Research Report, 2012. Available at http://www-smis.inria.fr/~anciaux/MinExp/

[4] Anderson, A.H. An Introduction to the Web Services Policy Language (WSPL). In *Proceedings of the POLICY Workshop*, 2004.

[5] Ardagna, C.A., De Capitani di Vimercati, S., Foresti, S., Paraboschi, S., and Samarati, P. Minimising Disclosure of Client Information in Credential-Based Interactions. *Int. Journal of Information Privacy, Security and Integrity*, *1*(2/3), to appear in 2012.

[6] Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. Enterprise privacy authorization language 1.2 (EPAL 1.2). W3C Member Submission, 2003.

[7] Baesens, B., Setiono, R., Mues, C. and Vanthienen, J. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, *49*(3), 2003.

[8] Bauer D, Blough D, and Cash D. Minimal information disclosure with efficiently verifiable credentials. *Digital Identity Management*, 2008.

[9] Belotti, P., Lee, J., Liberti, L., Margot, F., and Wachter, A. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, *24*(4-5), 2009.

[10] Chen, W., Clarke, L., Kurose, J., and Towsley, D. Optimizing cost-sensitive trust-negotiation protocols. *IEEE Computer and Communications Societies* (INFOCOM), 2005.

[11] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, 2002.

[12] Crook, J.N., Edelman, D.B., and Thomas, L.C. Recent developments in consumer credit risk assessment. *Euro. J. of Op. Research*, *183*(3), 2007.

[13] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data. *Official Journal of the EC*, *23*, 1995.

[14] Fourer, R., Gay, D.M., and Kernighan, B.W. A Modeling Language for Mathematical Programming. *Management Science*, *36*, 1990.

[15] Huysmans, J., Baesens, B., and Vanthienen, J. Using rule extraction to improve the comprehensibility of predictive models. Open Access publications from Katholieke Universiteit Leuven, 2007.

[16] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. Optimization by Simulated Annealing. *Science*, *220*(4598), 1983.

[17] Mitchell, T. *Machine Learning*. McGraw-Hill, 1997.

[18] Moses, T. Extensible access control markup language (xacml) version 2.0. Oasis Standard, 2005.

[19] OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, 23rd Sept. 1980.

[20] Ponemon Institute, LLC. 2010 Annual Study: U.S. Cost of a Data Breach. 2011.

[21] Samarati, P. Protecting respondents' identities in microdata release. *IEEE TKDE*, *13*(6), 2001.

[22] Sweeney, L. k-Anonymity: a model for protecting privacy. *Int. Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, *10*, 2002.

[23] Xiao, X., and Tao, Y. Personalized privacy preservation. In *Proceedings of ACM SIGMOD*, 2006.

[24] Yao, D., Frikken, K.B., Atallah, M.J., and Tamassia, R. Private information: To reveal or not to reveal. In *ACM TISSEC*, *12*(1), 2008.