Privacy Requirements: Present & Future

Pauline Anthonysamy*†1, Awais Rashid* and Ruzanna Chitchyan‡
*Security Lancaster
Lancaster University, UK
{p.anthonysamy, a.rashid}@lancaster.ac.uk
†Google, Zurich, Switzerland
{anthonysp}@google.com
‡Department of Computer Science
University of Leicester, UK
{rc256}@leicester.ac.uk

Abstract—**Software systems are increasingly open, handle large amounts of personal or other sensitive data and are intricately linked with the daily lives of individuals and communities. This poses a range of privacy requirements. Such privacy requirements are typically treated as instances of requirements pertaining to compliance, traceability, access control, verification or usability. Though important, such approaches assume that the scope for the privacy requirements can be established *a priori* and that such scope does not vary drastically once the system is deployed. User data and information, however, exists in an open, hyper-connected and potentially "unbounded" environment. Furthermore, "privacy requirements - present" and "privacy requirements - future" may differ significantly as the privacy implications are often emergent *a posteriori*. Effective treatment of privacy requirements, therefore, requires techniques and approaches that fit with the inherent openness and fluidity of the environment through which user data and information flows. This paper surveys state of the art and presents some potential directions in the way privacy requirements should be treated. We reflect on the limitations of existing approaches with regards to unbounded privacy requirements and highlight a set of key challenges for requirements engineering research with regards to managing privacy in such unbounded settings.**

I. Introduction

Contemporary software systems and services exist in a *hyper-connected* setting and regularly collect, process or disseminate massive amounts of data. Ensuring user privacy in such settings is non-trivial. This, in turn, has led to privacy concerns regarding potential individual and societal harms. The problem is, however, compounded by the fact that both the scale of this hyper-connected environment and the information flowing through it is constantly growing. Facebook alone has over 1 billion users and the volume of digital records worldwide by the year 2020 will be measured in zettabytes (10^{21}) [1]. The number of connected devices is also expected to grow to 50 billion by the year 2020 [2] driven by innovations in smart cities, IoT, body-area networks, smart grids and wearable sensors. With digital technologies becoming embedded in everyday objects and infrastructures, this effectively amounts to privacy management on an ultra-large-scale.

Requirements engineering and the wider privacy research have responded to these challenges. A number of approaches, tools and techniques have been developed for managing privacy requirements. Such approaches address privacy requirements from a particular perspective abstracted from the real-world, e.g., compliance with privacy policies and regulatory codes, traceability of privacy requirements through design and implementation, access control models to limit visibility of information to particular actors, and verifying the correctness and usability of privacy controls operationalising the privacy requirements. Though important such approaches are driven by the notion of a *bounded* system – whereby the scope of the privacy requirements is limited to the system under consideration be it a social networking platform, a wireless sensor network, a wearable device and so on.

Of course, best practice dictates that the interactions of such a system with other systems and services are taken into consideration during requirements engineering. This is nevertheless based on the assumption that the interactions and their scope can be established *a priori* and would not change drastically over the lifetime of the system. This, certainly, has been true in the past for traditional and relatively closed and static systems, like health-care, student records, or stock maintenance, where the privacy protection of the users was wholly the responsibility of such systems and their hosting organisations. Yet, the rise of the *participatory data economy* and new technologies has dramatically changed this landscape. There are hardly any "traditional" (or closed) systems left, even health-care systems are embedding third-party web-services into their daily work processes, e.g., monitoring and controlling outpatient health state via mobile applications. Moreover, now users – and not only organisations – actively exchange data with their environment. Thus, user data and information handled by such systems now exist in a potentially *unbounded* setting – in which the information is not restricted to any single domain, application, organisational, geographical, or contextual boundary – although technical mechanisms are usually in place so that they cannot be easily subverted.

While some privacy requirements can, of course, be established *a priori*, a range of unanticipated privacy requirements only emerge *a posteriori* when the system comes into contact with the hyper-connected setting. A recent example of this is that of the "eavesdropping television sets" where voice

[1] Work was done while at Security Lancaster, currently working at Google.

control features listen to all possible conversations to detect commands. Even if we could anticipate all potential privacy requirements *a priori*, "privacy requirements - present" and "privacy requirements - future" may differ considerably and are often determined by factors lying beyond the boundary and interactions anticipated during system development. For instance, privacy perspectives and concerns have shifted substantially in the post-Snowden landscape. Similarly, at its conception, WhatsApp guaranteed that user data would not be shared with third-parties, a guarantee that was recently relaxed following its takeover by Facebook[1].

This survey highlights that effective treatment of privacy requirements in such potentially unbounded settings requires techniques and approaches that fit with the inherent openness and fluidity of the environment through which user data and information flows. This, in turn, requires a fundamental shift in the way privacy requirements are treated. We categorise existing approaches to fulfilling privacy requirements into four broad classes, based on their treatment of privacy requirements, as instances of requirements pertaining to: *Compliance*, *Access Control*, *Verification* and *Usability*. We critically reflect on the strengths and limitations of approaches in each category with regards to managing privacy in an unbounded setting. Our analysis highlights that existing approaches are system-centric and largely address shallow privacy requirements, i.e., those pertaining to basic data attributes at the point of sharing. We highlight that effective treatment of privacy requirements demands several fundamental shifts: from a *system-centric* to *cross-domain view of privacy*; from *privacy management* to *user empowerment*; and from *shallow attribute-driven privacy requirements* to *deep privacy that considers derived attributes and synthetic data*. Section II, next, provides a survey of the state-of-the-art in privacy research and presents the strengths and limitations of the approaches. Section III discusses the key themes emerging from the survey and outlines a research agenda. Finally, Section IV concludes the paper.

II. State-of-the-Art: Privacy Requirements Engineering

In the following, we present a classification of existing approaches to handling privacy requirements, namely: *Compliance*, *Access Control*, *Verification* and *Usability*. These four classes have been derived by analysing existing state-of-the-art in requirements engineering and the wider privacy literature. The analysis of the state-of-the-art is by no means exhaustive yet it provides a rich view of the four classes. By classifying the approaches into these perspectives, an agreed understanding about the nature and scope of the particular perspective can be better elicited [3].

A. Privacy from the Perspective of Compliance

Approaches in this perspective operate on the basis of deriving privacy requirements from data protection legislation. The focus of these approaches is on eliciting and analysing requirements that are necessary to make systems, such as health-care, hotel management, etc., data protection legislation compliant [4], [5], [6]. These methods make use of the theoretical frameworks provided by legal scholars [7], [8] and security/privacy standards frameworks [9] to elicit the privacy requirements [5], [10] and model privacy expectations and practices [11]. Other approaches have focused on expressing traceability relationships between various software entities such as legal documents, requirements and, source code [12], [13], [14] and identifying inconsistencies in natural language software requirements for a successful software system development [15], [16].

1) Policies and Requirements: Antón et al. [17] present a technique for aligning security and privacy policies with system requirements of e-commerce websites via inspections. The inspections were conducted using heuristics to compare requirements, privacy and security policies to identify and resolve conflicts and inconsistencies across the documents early in the software development process. The heuristics for the comparisons were inherited from the Evolutionary Prototyping with Risk Analysis and Mitigation Model (EPRAM) [18] which includes risk and compliance assessment activities to verify proposed requirements for compliance with security and privacy policies. The approach uses goal models to express requirements and policies. These models are then cross-examined with each other to first identify what they call 'critical conflicts' (e.g., a policy exists but the corresponding requirement is missing) and then re-examined for terminology conflicts. Relationship indicators are defined to assess the degree of compliance between requirements and policy statements. Antón et al.'s work is among the first literature on assessing compliance between requirements and policy documents. However, the approach is highly manual. Although the goal extraction activity is automated, significant effort is required from analysts.

There has been a significant amount of work to ensure that software requirements comply with governing legal texts and privacy policies [6], [19]. Initial efforts in this area involved extracting requirements from privacy policies. Young et al. in [6], [19] introduced a systematic method - Commitment Analysis - for obtaining requirements from privacy policies of health-care organisations by extracting *commitments*, *privileges*, and *rights*. A *commitment* reflects an actor's pledge; a *privilege* reflects an action that an actor is entitled to perform; and, a *right* reflects an action that an actor is entitled to perform while imposing an action on another party. The method is composed of three steps: (1) the policy document is parsed into individual statements; (2) policy statements are classified based on a set of classifications; (3) classified statements are operationalised into requirements using templates. In Step 2, the statements are classified based on scope, actors, and concepts (commitments, privileges, and rights). This method was aimed at aiding requirement engineers to analyse the natural language text in privacy policies rather than using an intermediate representation i.e. goals [17]. The method focuses on the initial derivation of privacy requirements that

[1] https://www.nytimes.com/2016/08/26/technology/relaxing-privacy-vow-whatsapp-to-share-some-data-with-facebook.html?_r=0

are compliant with policies. This is a severe limitation as policies and requirements tend to evolve independently and this means a completely new set of requirements will have to be produced every time there is a change (even a minor one). Additionally, tracking policies and maintaining compliance comes at a cost when *commitments* need to change, and in turn software system requiring subsequent (manual) re-evaluation against those evolving requirements. The cumulative cost of minor changes rippling through can become unmanageable.

Breaux et al. [4] developed a method using Semantic Parametrisation to extract rights and obligations from legal documents, for example, the HIPAA Privacy Rule. The method uses Goal-Based Requirements Analysis Method (GBRAM) and Semantic Parametrisation (a process in which domain descriptions are represented in first-order predicate logic) to construct formal models from natural language texts. The method is composed of two parts: (1) policy goal mining, in which policy documents are translated into goal models; (2) semantic parametrisation, in which the generated goal models are transformed into restricted natural language statements (RNLS) and then parametrised to achieve semantic models. The semantic models can be leveraged to identify missing information or clarify ambiguities and aid in conflict analysis. Similar to the approaches described above, this work also primarily focuses on deriving requirements from privacy policies and legal documents (of health-care domains).

Hassan and Logrippo [20] present an approach to validate compliance and consistency between legal and organisational policies (or requirements). Requirements are extracted from plain legal texts using a Unified Modeling Language (UML) class model called Governance Extraction Model (GEM). The authors classify legal statements into three types, namely procedural, declarative and ontology. A procedural statement takes the form *if–then*, while declarative statements reveal system properties that can be translated into procedural statements. Ontology statements can be either organisational structure statements or process ontology statements. Compliance checking is then performed by translating the requirements into a formal language – based on first-order logic – called Governance Analyst Language (GAL). Finally, an implementation and validation checking is proposed to ensure that the software meets organisational requirements. Although, the authors propose this final validation step, details on how this might be achieved are not discussed.

Robinson developed a framework, REQMON, to monitor software requirements at runtime [21]. The framework is composed of (1) a language for requirements and monitor definitions; and, (2) a method for defining requirements, identifying obstacles and analysing monitor feedback. REQMON uses the KAOS language, patterns and methodology for its requirements and monitor definitions. REQMON automatically links raw event data on system usage to measurements on specified requirements. Monitoring determines if specified properties, which 'ought' to occur, do in fact occur as desired; it determines requirements satisfaction from the observed behaviour. This approach focuses on runtime compliance with system requirements but lacks in handling requirements that may emerge *a posteriori*, that is, when the system comes into contact other systems in a hyper-connected setting.

Gervasi et al. [15] propose an approach to automatically discover inconsistencies in the requirements from multiple stakeholders, using both theorem proving and model-checking techniques. Inconsistency occurs when a specification contains conflicting, contradictory descriptions of the expected behaviour of the system to be built or of its domain [16]. The method presents (1) a formal framework for identifying, analysing, and managing inconsistency in requirements specifications; (2) a parsing technique and a translation schema that allow requirements expressed as simple (controlled) natural language sentences to be automatically transformed into propositional logic formulae and the reverse i.e., propositional logic to natural language sentences. Such a formalisation can be leveraged to partially validate natural language requirements and/or policy documents.

The alignments of policy with commitments and regulations to software requirements is definitely necessary to support accountability as these techniques make auditing explicit and possible. Nevertheless, the above approaches emphasise an organisational viewpoint specifically for single systems (in a domain with limited volatility e.g., health-care). In contrast, modern software systems are required to exist in a more open and volatile environment, often underpinned and driven by the *participatory data economy*. As much as these formal and practical approaches are considered state of the art, they may not be adequate to adapt to the changing nature of software systems that are more open, and where information flows from one data controller to another in various contexts.

2) Traceability: Traditionally, traceability is achieved through a matrix [22] which correlates any two baseline artefacts e.g., requirements and architectural models, that may have one-to-many or many-to-many relationships between them. However, constructing and maintaining such matrices are deemed complex and laborious tasks, especially since software artefacts evolve over time, the traces tend to erode into an inaccurate state. Newer traceability mechanisms have been proposed over the last decade [23], [13] which include automated trace retrievals through machine learning techniques [12] and mapping of requirements to legal documents [5]. These techniques have contributed and greatly eased program comprehension and software reuse.

Massey et al. [5] evaluated the security and privacy requirements of an existing software system – iTrust, an open source electronic health record system – for regulatory compliance (HIPAA). The evaluation method comprises of four key activities: (1) terminology mapping, in which the terms used in the software requirements and legal text (with which the corresponding requirements must comply) are mapped based on actors, data objects and actions primitives; (2) requirements identification and disambiguation, in which, each requirement (from the previous step) is annotated with answers to a set of questions (Inquiry Cycle model) and then disambiguated; (3) requirements elaboration, in which disambiguated requirements are documented for priority and origin (provenance); (4) tracing requirements to legal texts, in which traceability links are established for each requirement from the set of requirements produced by step (3) and (4) to the relevant statements in the legal text with which each requirement must comply. This work mainly focuses on establishing trace links between software requirements and legal texts which is an important initial step in legal compliance and establishing accountability. However, the approach is proposed for bounded systems, in this case health-care systems. Here an important question to consider is, what if the system becomes more open, for instance, integration with a monitoring device that sends back patient data? Such devices cause a range of privacy requirements to emerge *a posteriori* when they *intersect* with extant health-care systems and user behaviours.

Cleland-Huang et al. [12] proposed two machine learning (ML) methods to automatically generate traceability links between regulatory codes – a subset of HIPAA – and product level requirements. The first ML approach uses a manually created traceability matrix to train a classifier to trace regulatory codes. The training set includes regulatory codes, requirements and their associated traces. Requirements are classified based on a score that reflects the degree to which a term (in the requirement) represents a specific regulatory code. To evaluate the effectiveness of the regulatory classifier, the authors used a leave-one-out cross validation experimental design for tracing the HIPAA rules against the patient health-care systems. In the second ML approach, a novel information retrieval method to mine terms and phrases from domain specific documents is proposed. This approach is based on the idea that when a training set is not available, a relevant set of indicator terms can be learned from domain specific documents from the Internet in order to replace or augment original trace queries. The steps involved in this approach include: (a) identify domain specific documents; (b) analyse identified documents to extract domain specific terms; (c) compose the terms into a new query which is used to execute the trace. Both methods were evaluated by tracing security regulations in the HIPAA document against the requirements of ten health-care systems. A similar method with a different emphasis is presented by Antoniol et al. [14], [13] in which traceability links are established and maintained between source code and text documents such as requirements, design documents and user manuals. The approach was evaluated by tracing the Java classes of Albergate (a hotel management system) to its functional requirements. Once again these approaches are developed based on and for bounded systems, namely health-care and hotel management. The strength and weaknesses of the work related to compliance are summarised in Table I.

B. Privacy from the Perspective of Access Control

Access control management is known to be a difficult problem for end-users in diverse areas such as authentication, authorisation, etc. [24]. The approaches presented in this section focus on the realisation of access control mechanisms with respect to disclosed user information.

TABLE I
STRENGTHS AND WEAKNESSES OF COMPLIANCE SOLUTIONS.

Strengths
1) Consideration of security and privacy requirements at the early stages of software development and not as an afterthought.
2) Development of formal languages for specifying policies and regulatory texts.
3) Implementation of traceability during software development life-cycle allows engineers to navigate between and browse different software artefacts.
4) The alignments of policy with commitments and regulations to software requirements supports accountability as these techniques make auditing explicit and possible.
Weaknesses
1) Methods for demonstrating traceability relationships primarily focus on terms of compliance with governing legal documents – *a posteriori* system implementation.
2) Inadequate for data intensive applications and hyper-connected systems like social networks, smart grids, etc. – most solutions are based on less volatile and single domain systems such as health-care.
3) No support for the integration of privacy constraints that emerge *a posteriori* and from external entities like third-party organisations.
4) Lack in providing continuous conformance to changing privacy requirements and functionalities (implementation) – limited evolution support.

1) Policy Languages: The goal of policy language research is to develop platform independent technical privacy policy languages that allow users and organisations to express the privacy controls that they desire. In this case policies are treated as a set of requirements that are operationalised as controls. Using such policy languages service providers can encode their data collection and usage practices in a machine-readable format.

Platform for Privacy Preferences (P3P) [25] was the pioneering work in the policy languages domain. P3P documents are based on an XML schema that allow service providers to publish their privacy policies in a machine-readable format. P3P-enabled browsers can read policies published in this XML format and compare them with user specified privacy settings. Users are able to rely on these browsers to read and evaluate privacy policies on their behalf avoiding the overhead of having to read through the natural language privacy policies' legal jargon. A P3P statement comprises of: purpose – how collected data is used and whether individuals can opt-in or out of any of these uses; data – the types of data; recipients – whether and under what conditions data can be shared and whether there is an opt-in or out; and, consequence – human readable explanation of a site's data practices. Although significant effort was invested in the development of P3P, it was not widely implemented [2] and ceased operation in part due to businesses not wanting to allow users to negotiate their policies on an individual basis.

In contrast to P3P which is a web privacy policy language, the Enterprise Privacy Authorisation Language (EPAL) [26] and eXtensible Access Control Markup Language

[2] https://www.epic.org/reports/prettypoorprivacy.html

(XACML) [27] are enterprise policy languages. Both EPAL and XACML are XML based languages (as P3P) and primarily used to represent the internal policies of an enterprise which would help organisations to perform the actions as stated in their privacy policies. While the two are very similar in structure and concept, the differences between the languages are significant, and greatly affect their usability and their ability to meet the requirements of an enterprise privacy policy language [28]. One notable difference between the two is that EPAL is mainly targeted at privacy policies and not access control policies in general.

Research in policy languages have considered extensively users' requirements for expressing privacy preference at the time of design, and provided vocabularies to cover key aspects including purposes, obligations, and data retentions. Although this is a positive step in terms of increasing understandability of privacy policies these languages ended up being too complicated for users to comprehend and parse. Follow-up studies like [29] have shown that it is impractical to expect extensive inputs from users for setting up their preferences, creating an avoidable barrier for adoption. Alternate proposals were made by Sadeh et al. [30] to use "privacy personas" for representing user privacy preferences wherein, clusters of people can be identified based on the similarities in the type and amount of information they chose to disclose about themselves. Nevertheless, an important drawback of the proposed solutions are the fact that they focus on handling privacy issues relating to the *first hop* of personal data flow from the user. This means that they do not cover all the data processors who may end up receiving users' personal data especially in a hyper-connected and potentially unbounded setting.

2) Browser–Based Privacy Solutions: Fredrikson and Livshits [31] present an in-browser approach, RePriv, that aims to perform personalisation while preserving user privacy. RePriv achieves this goal by requiring explicit user consent in any transfer of sensitive user information. The prototype includes: (1) a mining algorithm that observes users' browsing behaviour and automatically updates a profile of user interests; (2) a communication protocol on top of HTTP that allows web sites to utilise the information maintained in the browser; (3) an extension framework that allows third-party extensions to mine and utilise the information maintained by RePriv, and interact programatically with web sites. The authors also show how, with the help of static software verification, third-party code can be incorporated into the system, and given access to sensitive user information, without sacrificing control and user consent.

Guha et al. [32] propose a framework, Ibex, for authoring, analysing, and deploying secure browser extensions. The framework provides an API that exposes core browser functionality to extensions. These APIs are designed for the static verification of extension security and therefore mediate access to features that can be abused by malicious extensions. To describe an extension's privilege over browser resources, the framework includes a policy language based on Datalog. The policy language allows the specification of fine-grained authorisation and data flow policies on web content and browser state which are accessible by extensions. The framework also includes a policy visualisation tool that helps an administrator to estimate an extension's access rights on specific web pages, formal semantics of policies in terms of a safety property on the execution of extensions, and a verification method that allows static verification of extensions for policy compliance. This work highlights the strength of static checkers and how it can be utilised in tracking information flow and extracting code identifiers, e.g., function names and variables.

TABLE II
STRENGTHS AND WEAKNESSES OF ACCESS CONTROL SOLUTIONS.

Strengths
1) Provide individuals with the ability to control the information they reveal to others at a higher level of granularity. 2) Enhances economical gains of businesses, namely social networks – enable them to carefully use data mining technologies to obtain hidden information with out any potential intrusion to users' privacy.
Weaknesses
1) Addresses the privacy management problem purely from an access control perspective within a single stakeholder/entity (single-hop). 2) Correctness of access controls are mainly verified independently of policies (which are presented to users) – privacy policies and controls are treated as separate entities. 3) Browser-based solutions focus on simple mechanisms to check for users' privacy preferences, at the time of data disclosure.

FaceCloak [33] which is implemented as a browser extension protects user privacy on social networks by shielding one's personal information from the social media site and other users who does not have authorised access. Privacy Bucket [34] measures the extent to which third-party trackers can discover demographic information about its users. For example, visiting online shopping sites like Monsoon to purchase evening wear may suggest that the user is female within a certain age group. The goal of Privacy Bucket is to provide an overview of information that can be discovered about them through their browsing habits.

The ease of deployment of browser-based solutions has led to their rise as a privacy protection mechanism. However, browser-based protections are merely discrete steps in the absence of a comprehensive privacy solution. Similar to policy languages these solutions address issues related to a single hop of personal data flow from the user. Ultra-large-scale hyper-connected settings bring a bigger challenge in terms of data flow, i.e., multi-hop, and pose a wicked problem for privacy requirements engineering. The strength and weaknesses of the work related to access control are summarised in Table II.

C. Privacy from the Perspective of Verification

Approaches in this perspective focus on the verification and correctness of software systems using formal methods [35], [36], [37], [38]. They innovate by applying formal methods for verification of security and privacy requirements which enhances software reliability of systems that employ them.

TABLE III
STRENGTHS AND WEAKNESSES OF VERIFICATION SOLUTIONS.

Strengths
1) Fully automated technique for verifying behavioural properties of a model of a system by exhaustively enumerating its states – that is checking every possible execution of a system.
2) Have the ability to provide a *counter-example*, i.e., when a program fails to satisfy a property, model checking always demonstrates an execution of the system which renders the property violated.
Weaknesses
1) Not very scalable to very large hyper-connected systems unless the model is very abstract – as the size of software increases, it becomes very hard to build and test e.g., state explosion problem.
2) Traditional approaches to model checking have been to build a model of the system and verify it; the actual implementation is done after the model has been verified which contradicts current web application development practices.
3) Deriving a model from the source code is a key problem especially when the system is open and connected with other systems; wherein a coarse abstraction may not be precise enough to prove the property, and the analysis of a detailed abstraction time consuming.

Model checking is an automatic technique for verifying finite state systems [35]. This technique for validation of software has gained increasing appeal in systems that handle sensitive data, mainly because it can perform exhaustive checking. Fisler et al. [36] attempt a model-checking based verification system, Margrave, for analysing role-based access control policies. The verification system consumes a policy (requirement), represented using XACML, and a property and determines whether the policy satisfies this property. The verifier translates the XACML policies into a form of decision diagram i.e., multi-terminal binary decision diagrams (MTBDD), to answer queries that verify a system property.

May et al. [37] present a framework that formalises regulatory rules, namely HIPAA, and exploits this formalisation to automatically analyse the rules' conformance in a healthcare system. Their formalisation is known as the 'privacy APIs' – modelled using Promela, a C-like language – and is an extension of the basic set of access control matrix operations. The key advantage of this formalism is its ability to preserve the subtleties of the law during modelling and analysis of regulatory texts. The framework details a method that translates natural language text to this formal language. Upon translation, the rules are converted into a format suitable for input to a standard model checker. Finally, the framework also allows for policy evaluation.

Basin et al. [38] show how policies can be formalised using metric first-order temporal logic (MFOTL) and be used for monitoring system compliance to those policies. MFOTL is an expressive first-order language with metric temporal operators. The first-order fragment is used for formalising relations on system data, while the metric temporal operator is used to specify properties depending on the times i.e., past, present, and future system events. The approach illustrates the formalisation of a variety of security policies including Chinese Wall, compliance and history-based access-control policies, which are important for many enterprises and which govern the access and the usage of sensitive data. Formalised policies are then used to monitor and evaluate the conformance of system behaviour to the policies.

Model checkers and model-based approaches are prominent practices in software verification for compliance with security and privacy requirements. Nevertheless, model-checking tools are primarily applied on static software systems and not on systems that inherently change, for instance, data intensive applications like online social networks. The profile information that users provide to social networks, along with their social graphs, interactions with internal and external applications, and linkable actions outside the network open up such systems and lead to privacy implications that are often emergent *a posteriori*. The strength and weaknesses of the work related to verification are summarised in Table III.

D. Privacy from the Perspective of Usability

Usability researchers focus on the evaluation and understanding of user behaviours, needs, and motivations through observation techniques, and analysis of usability problems of existing privacy solutions. This perspective covers a wide spectrum which includes user studies on privacy perceptions [39], [40], [41], privacy breaches in social media [42], and improvement of user awareness [43], [44], [45].

A pioneering study on Facebook users' privacy settings was conducted in 2006 by Acquisti et al. [45] during which Facebook was still a social network for colleges and high schools. This study measured the accuracy of users' perceptions of their level of disclosure by questioning them on the visibility of their profiles and comparing their answers against the amount of data available to all members of the users' university network. The results indicated that 8% of the users were sharing more than they expected and 11% were sharing less than they expected, but overall most users, 70%, were fully aware of what they were sharing [45]. This study was a follow-up from the author's previous study [44] that passively measured information disclosure on Facebook in which the majority of users shared large amounts of personal information, but only a minority of them chose to limit access to their profile to just friends (0.06%).

Bonneau et al. [43] conducted a comprehensive study on privacy of 45 online social networks (OSNs) using criteria such as the diversity of data collected by the sites, the types of privacy controls, promotional methods etc. Their study included a general analysis of legal privacy policies in which attributes like the accessibility of privacy policies, their length, data claims and so on were examined. The authors then present a privacy communication game model, in which they conclude that OSNs may have evolved to communicate differently to users with different levels of privacy concerns. Another empirical evaluation shows that OSN privacy settings do not match sharing intentions [41]. There is a mismatch between OSN users' beliefs and their information sharing practices. Their results indicate that every participant (N=260) had at least one incorrect privacy setting. Krasnova et. al [46] held a focus group with 210 university students about their concerns on the use of Facebook. The most frequently reported theme was concern over unwanted audiences such as supervisors, subordinates and parents, viewing shared content. Participants also reported concerns over the collection and use of their data by the OSN provider and third-parties.

Gurses et al. [42] distinguish between four categories of OSN-specific privacy breaches encompassing issues of indeterminate visibility of user profile information, separation of identities and aggregation as well as misappropriation and contested ownership of user data. Indeterminate visibility is the problem of a user's profile being visible to others without the user's explicit knowledge or approval; separation of identities refers to the problem of a user's internal and/or external identity, which selectively reveal information, being exposed through data aggregation; contested ownership describes the problem of explicit and implicit definitions of data ownership that lead to privacy breaches; and, misappropriation is the problem when users' OSN data is re-purposed for a different context from its original collection purposes. These categories are interdependent but highlight different aspects of privacy breaches. Additionally, the work also presents privacy design heuristics to overcome these breach categories.

Luders et al. [39] performed a user study on the experience and attitudes of the general public, primarily Norwegian users, with regard to personal and consumer protection in social media. This survey showed that users' knowledge on how social media functions in regards to use, disclosure and transfer of their personal data is largely inadequate. The authors reported that users found the privacy controls to be difficult to configure and comprehend. In contrast, Majeski et al. [40] investigated users' sharing intentions and actual privacy controls in search of infringement. The study found that every one of their participants (N=65) had at least one sharing violation based on their stated sharing intentions. Both studies share a similar motivation that despite continuous efforts from OSN providers, users are still dissatisfied and concerned about the consequences of sharing their personal data.

The lack of usability and the complexity of configuring privacy controls have been identified as one of the main causes for unintended data disclosure in OSNs by the approaches above. However, usability and complexity are only part of the problem. Unlike "traditional" systems i.e., health-care, or student records, in which the responsibility of managing one's privacy is upheld by the concerned organisation, "newer" systems, i.e., OSNs, require users to take an active role in protecting their privacy online. Herein, there is a shift in responsibility which demands for new privacy requirements and software engineering models that handle both usability and process of engineering privacy into these newer systems. The strengths and weaknesses of the work related to usability are summarised in Table IV.

TABLE IV
STRENGTHS AND WEAKNESSES OF USABILITY SOLUTIONS.

Strengths
1) Presents to users the inherent problems they face in configuring privacy controls.
2) Enable the development of tools that increases comprehension and contributes to the minimisation of cognitive overhead for users.
Weaknesses
1) Targets only GUI complexities.
2) Lacks mechanisms and models that will allow the mapping of privacy requirements, articulated in the social context (user-perspective) of the system-to-be, to abstract properties, i.e., properties that are modelled and guaranteed by privacy solutions.
3) No support for traceability of privacy controls to policies displayed on a website, which is critical to establish users' trust.

III. Research Agenda

Tables I-IV in Section II highlight the strengths and weaknesses of existing approaches to complying with privacy requirements in each of the four categories. The limitations of the work reviewed in the four above discussed categories are summarised in Table V under three headings:

- The *System-Centric* heading refers to the issue that treatment of privacy remains focused around one (possibly distributed) system and does not effectively tackle the complexity of understanding, analysing, managing and operationalising privacy requirements in highly open and interconnected settings. As discussed before, existing work has not addressed privacy issues arising from a system's edges bleeding into several others (e.g., with health data getting mixed up with fitness applications and the integration of social networks through social-plugin).
- The *Syntactical* heading refers to the problem of "first hop" in personal data flow. Existing privacy solutions have focused on issues relating to the first hop of the personal data flow from the user, i.e., they focus on protecting privacy-related information (such as birth date, search terms, and relationship status) from being shared with the party with whom the user is currently communicating. They do not address protection and/or use of the once shared data across subsequent communications or further sharing with third-parties. Based on the reviewed perspectives only work on usability has looked at post-first-hop issues, albeit from the viewpoint of users' perceptions.
- The *Attribute-Driven* heading refers to the fact that existing approaches focus on protecting privacy by limiting the disclosure of a subset of data attributes such as personally identifiable information (PII) and not information that can be inferred from other shared information.

Treating privacy as an unbounded concern is, extremely challenging. However, significant advances can be made by addressing the three fundamental limitations above and broad privacy requirements that lie at the core of such unbounded concern. We outline these broad requirements next, highlight

TABLE V
Limitations of Existing State-Of-The-Art

	System-Centric	Syntactical	Attribute-Driven
Compliance	✓	✓	✓
Access-Control	✓	✓	✓
Verification	✓	✓	✓
Usability	✓		✓

current work that tackles aspects of these requirements and identify open problems as a research agenda for requirements engineering research and the wider software engineering and privacy research communities. We note that these broad requirements are not orthogonal.

A. System-Centric to Cross-Domain Privacy Requirements

As highlighted in Section II, current research typically takes an organisation- or network-centric view of privacy. Similar to policy enforcement in distributed system settings, this translates to either an obligation-driven or authorisation-driven [47] approach. In the former case, actions are enforced in response to particular events or stimuli within a system while, in the latter, access control rules specify whether a particular subject can legitimately access (or not) a particular object. Such approaches assume that the system, whether distributed or not, is within a single administrative control and even where platform or geographical boundaries are crossed, this happens within the control of a single organisation or a federated identity management framework [48]. In contemporary hyper-connected settings, data and information regularly crosses a range of platform, administrative, organisational and even geographical boundaries. This can be exemplified by a typical scenario whereby an individual uses his/her own device (e.g., a mobile phone or a laptop), a third-party network (e.g., an Internet cafe) and cloud services (e.g., Dropbox, Google Drive) to access, manage and utilise his/her personal data or information. One may argue that such cross-domain settings can be managed through service-level agreements (SLAs) between various providers in the above scenario. However, violation of such SLAs is often only detected post-hoc. Furthermore, in a large set of scenarios, for instance, those involving untrusted or partially-trusted third-party networks, specification, agreement and enforcement of an SLA is impossible.

What is required is a shift in perspective – from that of system-centric and data-origin-centric view of privacy to understanding and addressing privacy requirements arising from the cross-domain nature of contemporary settings. One way of potentially achieving this is by instilling accountability and transparency. Information accountability means the use of information should be transparent so it is possible to determine whether a particular use is appropriate under a given set of rules and that the system enables individuals and institutions to be held accountable for misuse [49]. Some research work is already heading in this direction, for instance, there is a body of work on audit-based solutions for data access in healthcare systems [50], or suggesting means of deterrence for information that requires more than traditional access control enforcement [51], or others offering *a posteriori* control for systems that require flexibility with accesses that are unanticipated under special circumstances [52]. However, there is a lack of such work in the context of hyper-connected systems like social networks, where the end user is the major active player, i.e. they have to actively manage their personal data as their own data controller across domains, instead of the system administrator.

Therefore, we call out to the research community to create new or adapt existing privacy requirement models to facilitate the notion of "accountability" and support the integration of privacy requirements that may emerge *a posteriori*. Such models will enable institutions and individuals to be called or identified to account for data misuses.

B. Privacy Management to User Empowerment

In many ways the basic premise of "privacy management" is flawed. In a potentially unbounded, cross-domain, information flow such privacy management is quite impossible except in the case of the most elementary privacy requirements. This is reflected in the focus on the "first hop" and privacy management at the point of sharing. Privacy, however, is highly contextual. Privacy trade-offs can often be essential to empower otherwise marginalised groups. For instance, [53] highlights scenarios where location privacy must be diminished to enable victims of domestic violence to access digital support services in physical locations away from their abuser.

What is required is a shift in perspective – from that of privacy management to empowering users with regards to their data and information, how it is utilised and how privacy is preserved as it flows through a hyper-connected environment. Some initial work in this direction has recently emerged. For instance, Bilogrevic et al. use machine learning techniques to enable optimised information sharing on user's context and schedule availability through the SPISM system [54]. Each sharing request is considered with respect to 18 different features that help to align this request with the user's preferences and sharing history. If SPISM can then, with high certainty, class a given request as acceptable or not, it will resolve the request accordingly, otherwise it will request an explicit decision from the user. While this approach shows encouraging results (90% correct decisions as per user feedback), it is not able to support more than 9 simultaneous policies per user, and yet again, focuses on the first-hop of data sharing only. Whereas in [55], Omoronyia et al. discuss how privacy disclosure can be regulated through the so-called privacy awareness requirements and demonstrate that such requirements are useful for establishing trade-offs between the need to disclose information (e.g., for social interaction, such as sporting activities) and wish to minimise threats to own privacy. Here both static attributes and inferred data and behaviour are considered and data flow history is accounted for in disclosure decision making. However, this work too assumes "complete" knowledge of the participating actors, context, and the privacy parameters. Yet, privacy must exist in the real world under incomplete knowledge, uncertainty, and inevitable error recovery. *A fundamental shift is also required in participatory data economy models with regards to user empowerment*, as exemplified in [56], wherein a trade-off framework incorporates a *utility function* of data sharing as the benefit from interaction with other users and potential privacy violations as the *cost function* which has to be balanced against gains. Using these two functions, this work adapts the intended sharing circle each time to maximise users' overall utility. Unfortunately, truly meaningful utility and cost functions are not easy to construct.

Such a shift in perspective, therefore, requires approaches in Section II to incorporate truly adaptive decision making techniques (such as those facilitated via machine learning), preserve history of the data exchange, and to review the utility and cost of each data sharing transaction. The notion of requirements@runtime [57] can potentially play an important role in realising such requirements.

C. Shallow Privacy to Deep Privacy Requirements

As discussed previously, one of the key issues across the various approaches examined in Section II is that they are often focused on an attribute-driven view of privacy. This is normally driven by regulatory requirements that call for protection of personally identifiable information (PII). While this is important, this nevertheless represents a shallow view of privacy requirements in contemporary settings. Research has already shown that "identifiable" information can be inferred from data beyond PII [58], [59], [60] and that it is possible to link seemingly disparate pieces of information about individuals online [61].

What is required is a shift in perspective – from that of a shallow view of privacy to a deeper view of privacy requirements that goes beyond basic attributes and considers derived attributes and synthetic data within the scope of privacy requirements. From a general software engineering perspective, existing work on Privacy Enhancing Technologies (PETs) [62] deals with measuring and protecting informational privacy by eliminating or minimising personal data using privacy preserving techniques like *k-anonymity* [63] and *differential privacy* [64], [65]. The primary objective of these solutions is to minimise identity leaks, promote transparency and to provide controls over data post-collection. However, there is limited research in this area from a requirements engineering perspective. The most relevant work is by Gürses et al. [66], which proposes a privacy requirements ontology to provide a set of concepts to reconcile the different privacy notions, solutions (that are often abstracted away from a specific context) and their interpretation in a given context by different stakeholders (multilaterality) during requirements engineering. The ontology extends the Zave and Jackson [67] model to better integrate stakeholder preferences with web-based systems and privacy. Although, this work is a step towards highlighting the need for existing requirements engineering methods to address a "deep" view of privacy, significant advances are needed. This requires requirements engineering approaches to, at least, start incorporating simple derived attributes, e.g., addressing issue of location privacy inferred based on known average speed and a previously known location [54], or that of probabilistically derived attributes through history of sharing and interaction. Addressing such issues can act as a stepping stone towards tackling the hard challenge of *deep privacy*.

IV. Conclusion

In this paper we have presented a review of the current privacy-related research, categorising it into coherent groups and identifying the strengths and weaknesses of addressing privacy issues from each of these perspectives. We have also outlined three broad core requirements critical to developing the research and practical capacity to address privacy requirements in potentially unbounded settings where:

- personal data no longer belongs to any single domain, application, organisational, geographical, or contextual boundary;
- individual users (and not only organisations) are the potential managers, beneficiaries, and victims in terms of gains and losses from their privacy disclosure and protection, and
- privacy protection depends not only on direct privacy attributes, but also on the way that one's shared data continues to flow across the various boundaries, and unexpected privacy breaches can occur through aggregation and analysis of seemingly unrelated, publicly available data.

We must observe that the environment where software and related privacy concerns reside has already morphed into such an unbounded setting. The privacy challenges stemming from this unbounded nature of privacy have also already started to materialise and will become more and more pressing. Yet, the software engineering research in general and requirements engineering in particular have not yet progressed sufficiently in addressing these challenges. Requirements engineering researchers can, and should, lead the way.

References

[1] (2014, February) International data corporation. http://uk.idc.com/. [Last Accessed 21 August 2014].

[2] (2011, February) More than 50 billion connected devices. ERICSSON White paper, http://www.ericsson.com/openarticle/mwc-connected-devices_1686565587_c. [Last Accessed 08 March 2015].

[3] C. Bennett and C. Raab, *The Governance of Privacy: Policy Instruments in Global Perspective*. Ashgate, 2003.

[4] T. Breaux and A. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE Trans. Softw. Eng.*, January 2008.

[5] A. Massey, P. Otto, L. Hayward, and A. Anton, "Evaluating existing security and privacy requirements for legal compliance," *Proc. RE*, 2010.

[6] J. Young, "Commitment analysis to operationalize software requirements from privacy policies," *Requirements Engineering*, 2011.

[7] H. Nissenbaum, "Privacy as contextual integrity," *Wash. L. Rev.*, 2004.

[8] D. J. Solove, "A Taxonomy of Privacy," *University of Pennsylvania Law Review*, January 2006.

[9] "Iso 27000 series." [Online]. Available: \url{https://en.wikipedia.org/wiki/ISO/IEC_27000-series}

[10] A. I. Antón, E. Bertino, N. Li, and T. Yu, "A roadmap for comprehensive online privacy policy management," *Commun. ACM*, 2007.

[11] A. Barth, A. Datta, J. C. Mitchell, and H. Nissenbaum, "Privacy and contextual integrity: Framework and applications," in *Proc. 2006 IEEE Symposium on Security & Privacy*, Washington, DC, USA.

[12] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A machine learning approach for tracing regulatory codes to product specific requirements," in *ICSE*, 2010.

[13] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo, "Tracing object-oriented code into functional requirements," in *8th International Workshop on Program Comprehension, 2000.*

[14] G. Antoniol, G. Canfora, A. de Lucia, and G. Casazza, "Information retrieval models for recovering traceability links between code and documentation," in *Proceedings of the International Conference on Software Maintenance*. Washington, DC, USA: IEEE Computer Society, 2000.

[15] V. Gervasi and D. Zowghi, "Reasoning about inconsistencies in natural language requirements," *ACM Trans. Softw. Eng. Methodol.*, 2005.

[16] C. Ghezzi and B. Nuseibeh, "Guest editorial: Introduction to the special section," *IEEE Transactions on Software Engineering*, 1999.

[17] A. I. Antón, J. B. Earp, and R. A. Carter, "Precluding incongruous behavior by aligning software requirements with security and privacy policies," *Information & Software Technology*, 2003.

[18] A. I. Anton, R. A. Carter, J. B. Earp, and L. A. Williams, "Epram: Evolutionary prototyping risk analysis & mitigation," Raleigh, NC, USA, Tech. Rep., 2001.

[19] J. Young and A. Anton, "A method for identifying software requirements based on policy commitments," in *18th IEEE International Requirements Engineering Conference (RE)*, Oct 2010.

[20] W. Hassan and L. Logrippo, "Governance requirements extraction model for legal compliance validation," in *2nd International Workshop on Requirements Engineering and Law*, Sept 2009.

[21] W. N. Robinson, "Implementing rule-based monitors within a framework for continuous requirements monitoring," *2014 47th Hawaii International Conference on System Sciences*, vol. 7, p. 188a, 2005.

[22] M. Hazewinkel. (2001) Matrix. Encyclopedia of Mathematics. Springer.

[23] J. Cleland-Huang, R. Settimi, E. Romanova, B. Berenbach, and S. Clark, "Best practices for automated traceability," *Computer*, June 2007.

[24] J. Saltzer and M. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, Sept 1975.

[25] L. Cranor, M. Langheinrich, and M. Marchiori, "A p3p preference exchange language 1.0 (appel 1.0)," WWW Consortium, April 2002.

[26] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, "Enterprise Privacy Authorization Language (EPAL)," Rschlikon, Tech. Rep., 2003.

[27] T. Moses. (2005, Feb.) eXtensible Access Control Markup Language TC v2.0 (XACML). OASIS.

[28] A. Anderson, "A comparison of two privacy policy languages: Epal and xacml," in *In SWS06: Proceedings of the 3rd ACM workshop on Secure web services*. ACM Press, 2006.

[29] L. F. Cranor, "Necessary but not sufficient: Standardized mechanisms for privacy notice and choice," *JTHTL*, 2012.

[30] N. M. Sadeh, J. I. Hong, L. F. Cranor, I. Fette, P. G. Kelley, M. K. Prabaker, and J. Rao, "Understanding and capturing people's privacy policies in a mobile social networking application," *Personal and Ubiquitous Computing*, 2009.

[31] M. Fredrikson and B. Livshits, "Repriv: Re-imagining content personalization and in-browser privacy," Los Alamitos, CA, USA, 2011.

[32] A. Guha, M. Fredrikson, B. Livshits, and N. Swamy, "Verified security for browser extensions," in *2011 IEEE Symposium on Security and Privacy*, May 2011.

[33] W. Luo, Q. Xie, and U. Hengartner, "Facecloak: An architecture for user privacy on social networking sites," in *Computational Science and Engineering, 2009. CSE '09. International Conference on*, Aug 2009.

[34] (2012, March) Privacy bucket. https://github.com/mfredrik/Privacy-Bucket/wiki. [Last Accessed 17 February 2014].

[35] E. M. Clarke, Jr., O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.

[36] K. Fisler, S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz, "Verification and change-impact analysis of access-control policies," in *Proc. Int'l Conf. SW Eng. (ICSE)*, New York, NY, USA, 2005.

[37] M. J. May, C. A. Gunter, and I. Lee, "Privacy apis: Access control techniques to analyze and verify legal privacy policies," in *Proc. 19th IEEE WS on Computer Security Foundations*, 2006.

[38] D. Basin, F. Klaedtke, and S. Müller, "Monitoring security policies with metric first-order temporal logic," in *Proc. 15th ACM Symposium on Access Control Models and Technologies*. NY, USA: ACM, 2010.

[39] P. B. Brandtzaeg and M. Lüders, "Privacy 2.0: Personal and consumer protection in new media reality," Tech. Rep., Nov'09 2009.

[40] M. Majeski, M. Johnson, and S. M. Bellovin, "The failure of online social network privacy settings," Tech. Rep. CUCS-010-11, Feb. 2011.

[41] M. L. Johnson, S. Egelman, and S. M. Bellovin, "Facebook and privacy: it's complicated," in *SOUPS*, 2012.

[42] S. Gurses, R. Rizk, and O. Gunther, "Privacy design in online social networks: Learning from privacy breaches and community feedback," in *ICIS 2008 Proceedings*. New York, USA: ACM, 2008.

[43] J. Bonneau and S. Preibusch, "The privacy jungle: On the market for data protection in social networks," in *Economics of Information Security and Privacy*. Springer US, 2010.

[44] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks (the facebook case)," in *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, 2005.

[45] A. Acquisti and R. Gross, "Imagined communities: Awareness, information sharing, and privacy on the facebook," in *Privacy Enhancing Technologies*. Springer Berlin Heidelberg, 2006.

[46] H. Krasnova, O. Gnther, S. Spiekermann, and K. Koroleva, "Privacy concerns and identity in online social networks," *Identity in the Information Society*, 2009.

[47] M. Sloman, "Policy driven management for distributed systems," *J. Network Syst. Manage.*, 1994.

[48] A. Squicciarini, A. Bhargav-Spantzel, A. Czeskis, and E. Bertino, "Traceable and automatic compliance of privacy policies in federated digital identity management," in *Proc. 6th Int'l Conf. Privacy Enhancing Technologies*. Berlin, Heidelberg: Springer-Verlag, 2006.

[49] (2014, October) Data sharing needs accountability. http://www.computerweekly.com/news/2240232292/Tim-Berners-Lee-Data-sharing-needs-accountability.

[50] M. A. C. Dekker and S. Etalle, "Audit-based access control for electronic health records," *Electron. Notes Theor. Comput. Sci.*, Feb 2007.

[51] K. Padayachee and J. H. P. Eloff, "Adapting usage control as a deterrent to address the inadequacies of access controls," *Comput. Secur.*, 2009.

[52] S. Etalle and W. H. Winsborough, "A posteriori compliance control," in *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*, New York, NY, USA, 2007.

[53] M. Emms, B. Arief, and A. P. A. van Moorsel, "Electronic footprints in the sand: Technologies for assisting domestic violence survivors," in *Privacy Technologies and Policy - First Annual Privacy Forum, APF 2012, Limassol, Cyprus*, 2012.

[54] I. Bilogrevic, K. Huguenin, B. Agir, M. Jadliwala, and J. Hubaux, "Adaptive information-sharing for privacy-aware mobile social networks," in *Proc. Int'l Conf. Pervasive & Ubiquitous Computing, UbiComp*, 2013.

[55] I. Omoronyia, L. Cavallaro, M. Salehie, L. Pasquale, and B. Nuseibeh, "Engineering adaptive privacy: on the role of privacy awareness requirements," in *35th Intl. Conference on Software Engineering, ICSE '13*.

[56] M. Yang, Y. Yu, A. K. Bandara, and B. Nuseibeh, "Adaptive sharing for online social networks: A trade-off between privacy risk and social benefit," in *Proc. 13th IEEE Int'l TrustCom Conf., 2014*.

[57] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. Finkelstein, "Requirements-aware systems: A research agenda for re for self-adaptive systems," in *Proc. 18th IEEE Int'l Requirements Eng. Conf.*, 2010.

[58] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *IEEE Symposium on Security & Privacy, San Francisco, California*, 2012.

[59] S. Afroz, A. C. Islam, A. Stolerman, R. Greenstadt, and D. McCoy, "Doppelgänger finder: Taking stylometry to the underground," in *2014 IEEE Symposium on Security and Privacy, Berkeley, USA*, 2014.

[60] A. Rashid, A. Baron, P. Rayson, C. May-Chahal, P. Greenwood, and J. Walkerdine, "Who am I? Analyzing digital personas in cybercrime investigations," *Computer*, 2013.

[61] M. J. Edwards, A. Rashid, and P. Rayson, "A service-indepenent model for linking online user profile information," in *IEEE Joint Intelligence and Security Informatics Conference, JISIC*, 2014.

[62] G. W. van Blarkom, J. J. Borking, and J. G. E. Olk, "Handbook of Privacy and Privacy-Enhancing Technologies: The case of Intelligent Software Agents," Privacy Incorporated Software Agent Consortium, Den Haag, Tech. Rep., 2003.

[63] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information (abstract)," in *Proc. 17th ACM Symposium on Principles of Database Systems*, New York, NY, USA, 1998.

[64] C. Dwork, "Differential privacy," in *ICALP*. Springer, 2006.

[65] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *In Proceedings of the 3rd Theory of Cryptography Conference*. Springer, 2006.

[66] S. Gurses, "Multilateral privacy requirements analysis in online social network services." Ph.D. dissertation, University of Leuven, May 2010.

[67] P. Zave and M. Jackson, "Four dark corners of requirements engineering," 1997.