

Final report on mechanisms

Editors: Pierangela Samarati (UNIMI)
Reviewers: Eva Schlehahn (ULD)
Dieter Sommer (IBM)
Identifier: D2.4.1
Type: Deliverable
Version: 1.0
Class: Public
Date: May 19, 2011

Abstract

The growing amount of information daily collected and produced by users and organizations has contributed to the success and rapid evolution of novel scenarios, where the techniques for processing, storing, communicating, sharing, and disseminating information have radically changed. Such scenarios, however, make it increasingly difficult and complex to guarantee that private and sensitive data are properly protected and to provide users with control over their own data. There are, in fact, new risks and new research challenges, whose investigation is crucial for the development of future privacy-enabled technologies and applications.

Goal of Activity 2 is to investigate open problems and research challenges towards developing new technological solutions and new mechanisms providing an effective response to different aspects of the complex privacy problem. This document describes the results obtained in Activity 2 in the third year of the project. After an introductory chapter surveying the different results produced by the Activity (appeared in the papers listed in the last chapter), the document illustrates in details - in a dedicated chapter - one specific contribution for each of the four work packages composing the activity.

Members of the PrimeLife Consortium

1.	IBM Research GmbH	IBM	Switzerland
2.	Unabhängiges Landeszentrum für Datenschutz	ULD	Germany
3.	Technische Universität Dresden	TUD	Germany
4.	Karlstads Universitet	KAU	Sweden
5.	Università degli Studi di Milano	UNIMI	Italy
6.	Johann Wolfgang Goethe - Universität Frankfurt am Main	GUF	Germany
7.	Stichting Katholieke Universiteit Brabant	TILT	Netherlands
8.	GEIE ERCIM	W3C	France
9.	Katholieke Universiteit Leuven	K.U.Leuven	Belgium
10.	Università degli Studi di Bergamo	UNIBG	Italy
11.	Giesecke & Devrient GmbH	GD	Germany
12.	Center for Usability Research & Engineering	CURE	Austria
13.	Europäisches Microsoft Innovations Center GmbH	EMIC	Germany
14.	SAP AG	SAP	Germany
15.	Brown University	UBR	USA

Disclaimer: The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The below referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2009, 2010 by IBM Research GmbH, Unabhängiges Landeszentrum für Datenschutz, Technische Universität Dresden, Karlstads Universitet, Università degli Studi di Milano, Stichting Katholieke Universiteit Brabant, Katholieke Universiteit Leuven, Università degli Studi di Bergamo, SAP AG.

List of Contributors

Contributions from several PrimeLife partners are contained in this document. The following list presents the contributors for the chapters of this deliverable.

Chapter	Author(s)
Executive summary	UNIMI
<i>Chapter 1</i> : Overview of main results	KAU, K.U.Leuven, IBM, SAP, TUD, UBR, ULD, UNIBG, UNIMI
<i>Chapter 2</i> : Cryptographic mechanisms (WP2.1)	IBM , K.U.Leuven, UBR
<i>Chapter 3</i> : Mechanisms supporting users' privacy and trust (WP2.2)	TUD , KAU, TILT, ULD
<i>Chapter 4</i> : Privacy of data (WP2.3)	UNIBG , SAP, TILT, UNIMI
<i>Chapter 5</i> : Access control for the protection of user-generated data (WP2.4)	UNIMI , UNIBG
<i>Chapter 6</i> : Abstracts of research papers	KAU, K.U.Leuven, IBM, SAP, TUD, UBR, ULD, UNIBG, UNIMI

Executive Summary

The huge amount of sensitive and private information available in the digital society, together with easiness of access and computation, greatly increase the risks of privacy breaches due to improper - direct or indirect - information disclosure. Guaranteeing proper privacy in emerging novel scenarios requires tackling diverse issues and aspects; the obstacles are not only understanding and responding to the various needs raised by the different facets of the problem but also developing novel technologies successfully responding to such needs.

Aim of Activity 2 is to investigate open privacy issues, performing research, and developing novel solutions and mechanisms that can be exploited as building blocks of future privacy-enabled technologies and applications. The activity's ambitious goal is therefore twofold: *i*) investigate open challenges proposing novel technological solutions and opening new perspectives, and *ii*) develop prototypal tools realizing such technologies so to make them usable by other activities of the project or by external parties (via the open source Primelife's activities). In line with the different aspects of the privacy problem to be tackled, Activity 2 is organized in four work packages: *WP2.1 – Cryptographic mechanisms* focuses on cryptographic techniques for supporting privacy and trust; *WP2.2 – Mechanisms supporting users' privacy and trust* focuses on solutions for supporting users in checking whether their personal data are used in accordance with privacy laws and privacy constraints specified by them; *WP2.3 – Privacy of data* focuses on solutions for assessing and ensuring privacy of large collections of sensitive data; *WP2.4 – Access control for the protection of user-generated data* focuses on solutions for enabling the enforcement of access restrictions on user-generated data.

Reaching the goal above, the different work packages of the activity have produced innovative results in research and development of privacy-enabling technologies. The work performed in the activity resulted in several publications in top international journals (e.g., ACM TISSEC, ACM TODS, JCS) and conferences (e.g., CRYPTO, ESORICS, VLDB, ICDCS) as well as in several tools that have been exploited by other activities (in particular, Activities 1 and 5) and made available to external parties as open source (via Activity 3).

Since a comprehensive illustration of all the different results would inevitably remain only at a high level, following the suggestions by the EU reviewers, for presentation in this document, we have selected a representative result for each work package to be illustrated in details. The remainder of this document is then organized in six chapters. The first and last chapters aim at providing a comprehensive view of the different results produced within the activity in the third year of the project. In particular, Chapter 1 provides an overview of the main research results and Chapter 6 lists the papers on which such results have been reported. Each of the central chapters (Chapters 2–5, resp.) illustrates a specific result developed within a work package (WP2.1–2.4, resp.).

Contents

1	Overview of main results	13
1.1	Cryptographic mechanisms (WP2.1)	13
1.2	Mechanisms supporting users' privacy and trust (WP2.2)	15
1.3	Privacy of data (WP2.3)	18
1.4	Access control for the protection of user-generated data (WP2.4)	20
2	Cryptographic mechanisms (WP2.1)	23
2.1	Introduction	23
2.2	Preliminaries	26
2.2.1	Modified Boneh-Boyen signatures	27
2.2.2	Zero-knowledge proofs and Σ -protocols	27
2.2.3	Credential signature scheme	28
2.2.4	Set membership scheme	28
2.3	Oblivious transfer with access control	28
2.3.1	Security requirements	29
2.3.2	Construction	30
2.4	Extending to priced oblivious transfer	33
2.4.1	Construction overview	33
2.4.2	Additional database keys	33
2.4.3	Obtaining wallets	34
2.4.4	Recharge protocol	34
2.4.5	Modified transfer protocol	36
3	Mechanisms supporting users' privacy and trust (WP2.2)	37
3.1	Event scheduling	37
3.2	Security requirements	37
3.2.1	Confidentiality	38
3.2.2	Integrity	38
3.2.3	Involved parties	38
3.3	Existing applications	39
3.3.1	Protection against outsiders	39
3.3.2	Integrity protection against participants	40
3.3.3	Confidentiality protection against participants	40
3.3.4	Protection against the initiator	40
3.4	New schemes without server trust	41
3.4.1	Protection against outsiders	41
3.4.2	Integrity protection against participants	42

3.4.3	Confidentiality protection against participants	43
3.4.4	Protection against the initiator	44
3.4.5	Conclusions	46
4	Privacy of data (WP2.3)	49
4.1	Shuffle index data structure	50
4.2	Protection techniques	52
4.2.1	Problem statement	52
4.2.2	Cover searches	53
4.2.3	Cached searches	55
4.2.4	Shuffling	55
4.3	Performance analysis	57
4.4	Conclusions	59
5	Access control for the protection of user-generated data (WP2.4)	61
5.1	Introduction	61
5.2	Authorization and encryption policies	63
5.3	Minimum encryption policy	65
5.4	Two-layer encryption for policy outsourcing	68
5.4.1	Two-layer encryption	68
5.4.2	Protection evaluation	72
5.5	Experimental results	74
5.6	Conclusions	76
6	Abstracts of research papers	77
6.1	Cryptographic mechanisms (WP2.1)	77
6.2	Mechanisms supporting users' privacy and trust (WP2.2)	84
6.3	Privacy of data (WP2.3)	87
6.4	Access control for the protection of user-generated data (WP2.4)	91
	Bibliography	105

List of Figures

1	Privicons	16
2	Issuer setup algorithm	30
3	Database setup algorithm	30
4	Issue protocol	31
5	Transfer protocol: The proof $PK(K$ is correct) is detailed in the text	32
6	Create wallet protocol	34
7	Recharge protocol: The proof protocol PK_w is defined in the text	35
8	The necessary amount of trust one needs to have about the involved entities: The arrows indicate which role an entity can take	39
9	Construction of a URL (RFC3986 [BLFM05]) and its partitioning when doing a GET request	42
10	Compared to existing applications, the user interface does not change. The password is transmitted within the fragment part of the URL, where it can be extracted using JavaScript and be used for AES encryption and authentication	43
11	A digitally signed vote	44
12	Asymmetric encryption of a vote	44
13	Interface to save a private key at client side	45
14	An example of abstract (a) and logical (b) representation of a data structure to be outsourced, and of the corresponding view of the server (c)	51
15	Access time in a LAN as a function of the number of covers (a) and of the size of the cache (b)	58
16	Overhead in a WAN compared to the use of a plain encrypted index as a function of the number of covers	58
17	Outsourcing scenario	62
18	An example of access matrix (a) and corresponding authorization policy graph (b)	63
19	An example of encryption policy graph	65
20	An example of encryption policy graph over $\{A, B, C, D\}$	67
21	An example of initialization (a), covering (b), and factorization (c) generating an encryption policy equivalent to the authorization policy in Figure 18	69
22	An example of BEL and SEL combination with the Delta_SEL and the Full_SEL approaches	70
23	An example of grant operation	71

24	Possible views on resource r	72
25	View transitions in the Full_SEL (a) and in the Delta_SEL (b)	73
26	From locked to sel_locked views	74
27	Number of tokens for the DBLP scenario (a) and total time required for retrieving keys and resources with single layer encryption and Full_SEL over-encryption (b)	75

List of Tables

2	Overview of existing event scheduling schemes. There is no scheme, which tries to overcome the need of a trustworthy server administrator	41
3	Overview of different event scheduling schemes	46

Chapter 1

Overview of main results

The chapter surveys the different results produced in the four work packages of Activity 2. The discussion in each section is organized in paragraphs, one for each task of the corresponding work package.

1.1 Cryptographic mechanisms (WP2.1)

The objective of this work package was to perform research on new cryptographic algorithms and protocols that allow for privacy protection and trust establishment in existing and emerging applications, as well as enabling new kinds of applications. The work package was further concerned with developing means to protect cryptographic material (secret keys, certificates, and credentials) on users' devices (e.g., their laptop computers). In the third project year, this work package aimed 1) at advancing the state of the art in the theory of privacy-enhancing cryptography, 2) at implementing a trusted wallet and 3) at maintaining and extending the identity mixer implementation.

Task 2.1.1 Cryptography for privacy and trust.

Anonymous credentials. The area of anonymous credentials was one of the foci of the research performed in this Work Package, thereby leveraging core competencies of the involved people. The research caters for trust and privacy requirements throughout the PrimeLife project. As anonymous credential systems have become more mainstream in the recent past, our results can be considered substantial contributions to the literature body. First, we have shown that attribute-based credentials can be constructed from Lattice-based assumptions. While the scheme is not yet as efficient as traditional schemes, it is the first one that provides security in the presence of quantum computers and hence we consider this a very important result. Second, we have investigated and published a means to address revocation of credentials, where the issuer can publish update information for each credential that remains valid so that the user can update their credentials without any interaction with the issuer. Third, we have made a formal model of the IBM Identity Mixer anonymous credential system and then used formal methods to prove its security and anonymity properties. Finally, we have come up with the most

efficient group signature scheme to date (a group signature scheme is essentially a credential system where there is a single issuer and credentials have no attributes) and at the time of this writing we are working on extending these results towards an anonymous credential system.

Applications of anonymous credentials. We have worked on a number of use-cases for anonymous credentials. The first one is addressed by new cryptographic protocols that allows two parties to establish a shared secret key if and only if each of them owns credentials satisfying some given predicates such as being a member of the same organization. The second one was a practical demonstrator showing how access to a teenager website can be protected using anonymous credentials issued by a government authority.

Optimistic fair exchange. Fairly exchanging digital content is an everyday problem. It has been shown that fair exchange cannot be done without a trusted third party (called the Arbiter). Yet, even with a trusted party, it is still non-trivial to come up with an efficient solution, especially one that can be used in a p2p file sharing system with a high volume of data exchanged. We provide an efficient optimistic fair exchange mechanism for bartering digital files, where receiving a payment in return to a file (buying) is also considered fair. We have furthermore extended the protocols so that it works for multiple arbiters. Finally, we have come up with a protocol that allows for a fair exchange of cash and one of a list of electronic goods (all having the same price) whereby the seller does not learn which good is bought.

Oblivious trusted third parties. Many cryptographic protocols (including fair exchange) assume a trusted third party that is only involved in the protocol in case of a conflict needing to be resolved. Such parties need to be trusted to perform their role if necessary and not to abuse their powers. Thus, they need to be trusted that they behave the same in all protocol instances and do not discriminate certain users. We have developed cryptographic means (i.e., an encryption scheme and related protocols) that enforce this by ensuring that the third party can not learn which protocol instance it is being involved in and hence no discrimination is possible.

Private service access. Our results in private service access feature a method for accessing a service in an unobservable manner. Such a mechanism can prove useful in social networking scenarios for hiding user actions from the network provider or for the access to sensitive data such as DNA sequences. The goal here is to allow users accessing the service without the services provider learning which service a user accesses or who the user is. We continued to develop protocols that allow a user to obliviously access records of a database while the data base server can ensure that only users who have the necessary access credentials for the requested record can read (decrypt) the records. We will describe our results in this area in detail in Section 2.

Cryptography for selective access control in social networks. The results in this area allow for strengthening the trust model for social networking. Particularly, users are given better control of who may view their data and their data may be hidden from the social networking provider. These results are an input to WP1.2 and interesting for the long-term development of the area of social networking at large. Our demonstrator makes use of existing key management mechanism based on OpenPGP and is available for download as open source component.

Task 2.1.2 Trusted wallet. This task is about the realization of means to securely store and use (anonymous) credentials. The idea is that credentials should be treated similarly to paper credentials: they are stored in the wallet and each time one is required to produce a credential, one opens the wallet, looks at the available credentials, and then selects one. For an electronic wallet, this could be quite similar – however, in this case one also needs to worry about securing the wallet, e.g., from viruses etc. Apart from progressing the architecture, we have in the reporting period focused on having credentials issued to users which can be protected against unauthorized access and use (for instance by viruses). To this end we have studied the use of smart cards for storing anonymous credentials and executing the protocols using the credentials on the card. We were able to demonstrate that with standard cards it is indeed possible to execute the protocols on the card by just using the standard API that the card offers (Accessing the cryptographic processor on the card is not possible without modification of the card's operating system which in turn would require the rather costly recertification of the card). Thereby we have proved that such cards can be used to protect credentials from unauthorized access. On top of that, we implemented a trusted wallet prototype by extending the identity selector of the Higgins identity framework. We have also extended other identity services, such as relying parties in order to accept anonymous credentials. In this way, we are able to prove that by the usage of a smart card that stores credentials, it is possible for users to perform anonymous authentication for websites. The trusted wallet prototype is an extended version of an identity selector that is activated whenever the user wants to present his credentials anonymously. The activation is done using a Firefox extension that bridges the browser and the trusted wallet, and presents the required claims. According to the required claims the credential selector presents to the user the anonymous credentials that match the claims. The relying party website can then verify the credential and thus allowing and denying authentication.

1.2 Mechanisms supporting users' privacy and trust (WP2.2)

The objective of Work Package 2.2 is to research mechanisms to support users in preserving and controlling their privacy (Task 2.2.1 Transparency support tools, Task 2.2.2 Privacy measurement) while enabling interaction and collaboration of group/community members (Task 2.2.3 Privacy-respecting establishment of collaborative groups, Task 2.2.4 Trust management by interoperable reputation systems, Task 2.2.5 Awareness). A special focus of our work in the third project year was on the development of a distributed privacy-preserving secure logging system, on conducting experiments related to privacy awareness tools and on advancing the solution for privacy-preserving event scheduling.

Task 2.2.1 Transparency support tools. Improving transparency via icons has been discussed especially for websites and their privacy statements so that users can better understand how their personal data are being processed. In this context, the so-called “Privicons” approach has been elaborated where e-mail senders can attach information to their messages how they want others to handle them. Six icons have been introduced that exist in a graphical as well as in pure ASCII form. These icons are shown in Fig. 1.

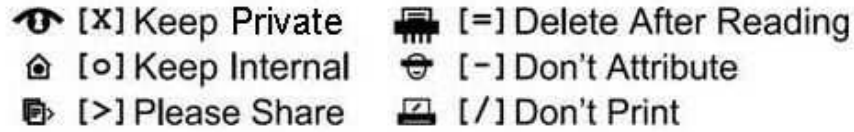


Figure 1: Privicons

The ASCII icons can syntactically be integrated in the first line of the body, in the subject line and/or in a dedicated header of any e-mail message. Note that the Privicons do not enforce the sender’s preferences, but only provide a way to express them.

The Privicon approach is being developed and implemented by researchers of Stanford University, the PrimeLife project and interested individuals¹. Further details are described in the proposed Internet-Draft “Privacy Preferences for E-Mail Messages” submitted to the IETF². Meanwhile, a first browser extension of the Privicons is available via the project’s website <http://privicons.org/>.

With respect to enhancing transparency of data processing, a privacy-preserving secure log has been developed. This enables users to see the events on a server relating to their data. The current goal is to create a distributed version of the privacy-preserving secure log that functions in a distributed (or cloud-based) environment while still preserving the requirements and functionality of the original log system. The enhancement of the privacy-preserving secure log will handle what we have called cascading logs, which is in essence a privacy-preserving referral mechanism between logs run by different parties. The system will make it possible for users to follow how his/her data is distributed, shared and used between different parties provided that the logging system is used and actions are logged. In collaboration with WP2.1, a design for such a distributed privacy-preserving secure logging system has been created, which allows data subjects to reconstruct a log trail of how his/her data has been processed across multiple parties.

Task 2.2.2 Privacy measurement. In this period, research has been done regarding the development of a privacy-respecting reputation protocol. However, besides *privacy*, reputation protocols should as well have other important properties, which are *liveliness* and *fairness*. Liveliness means that the reputation system does not reach a final state. With respect to fairness, both interaction partners need to trust in the other one’s correct behavior during rating. According to Camerer et al. and Dasgupta [CW88, Das00] this is a trust game. This trust game is fair if every user has equal possibilities for rating interaction partners. While reputation systems providing various notions of privacy (e.g., [Ste08, PRT04, Del00, ACBM08]) as well as privacy and liveliness [SCS10] have been proposed before, none of these systems has all three properties.

In our work, we have proposed a reputation system enabling privacy of reputation system users by information-theoretic relationship anonymity with respect to users and the reputation provider. At the same time the protection goals liveliness and fairness of the reputation system get fulfilled. For enforcing fairness, mutual ratings are forced to

¹<http://privicons.org/>

²<http://tools.ietf.org/html/draft-koenig-privicons>

be simultaneous and self rating is prevented. For the anonymity properties, the protocol bases on a DC-Net-like [Cha88] approach. Fairness is enforced by giving interaction partners the ability to rate only in cases both have successfully completed a registration phase beforehand. Thereby, the protocol enables to balance the security and privacy requirements of all users involved in the reputation system. The work has been published in [SCS11].

Task 2.2.3 Privacy-respecting establishment of collaborative groups. An overall investigation of privacy-enhanced event scheduling was performed and published [Kel10]. Furthermore, an extension to the proposed privacy-enhanced event scheduling algorithm was developed and published [Kel11]. This extension drops the restriction of unanimous agreement. Several usability tests were made, automatic tests and automatic bug-reporting were implemented. Consequently, the implementation is quite stable and the code was released as Open Source. A more detailed overview of the research result of this task is documented in Chapter 3.

Also, we did some analyzing work related to defining privacy in more complex scenarios (such as collaborative environments) than the ones usually considered in privacy research. This work has been published in [BBP11].

Task 2.2.4 Trust management by interoperable reputation systems. Our focus in the current working period was on investigating requirements for interoperability of reputation systems with other systems. This work is published in [Ste09]. In addition, we implemented a privacy-respecting reputation extension to the well known MediaWiki software, and published its design [KPS11].

Task 2.2.5 Awareness. In order to understand and support privacy awareness especially in a Web 2.0 context, we continued research on the influence of privacy-awareness-cues on users' perceived privacy and their disclosure behavior. The results of our studies show that forums users feel less private when privacy-awareness-cues are presented and that numerical cues have a slightly stronger impact than textual cues [Pöt10, PWG10]. Using the scenario of a wiki with reputation system, we found that – besides name, age and place of residence – users also regard their individual reputation value as personal data [KPS11].

In addition, we also continued the development of a privacy-awareness-tool for the popular community-software phpBB. The tool, which we called *Personal Data MOD*, could be integrated in the header of each phpBB-based forum to provide information about visibility of personal data to users and thereby supporting users' privacy awareness. More information and a download link for the tool can be found on the open source section of the project website [Pri].

Another aspect covered was the research area of privacy and trust in online social lending. We measured how much and which kinds of personal data members of the German-based social lending platform Smava.de disclosed [PB10, BP10, BP11].

1.3 Privacy of data (WP2.3)

The focus of Work Package 2.3 is represented by the consideration of large data collections that contain sensitive information on citizens. The overall goal is the definition of novel metrics and techniques able to support the management of privacy requirements, at the same time offering a significant degree of utility in access to the data. The investigation of these topics in PrimeLife had two roles: on one hand, it produced concrete techniques for the protection of personal information, identifying the amount of exposure deriving from access to the data and proposing approaches able to satisfy the stringent privacy requirements that the project wants to support; on the other hand, the identification of metrics and techniques provided input on the definition of components in the policy language able to express user preferences on the processing of their data. The availability of relational database technology is assumed for many of the scenarios considered in the Work Package. Work Package 2.3 is organized in three tasks, focusing on specific aspects of the privacy problem.

Task 2.3.1 Privacy assessment and privacy metrics. The goal of the task is the consideration of current privacy metrics for data collections and the design of new metrics or new techniques supporting privacy requirements in large data collections. An interesting research direction has been pursued which has led to the definition of a novel approach for the protection of privacy constraints. The approach is based on the consideration of novel solutions presented in the scientific literature and in their adaptation to a more general framework. The work in this task produced results with an impact on Task 2.3.2. The model has been presented in publication [DFJ⁺10c], by UNIMI and UNIBG.

Another contribution is related to the study of existing state-of-the-art protection mechanisms that addresses different aspects of privacy in mobile applications. This work produced a proposal ensuring private communication in the context of hybrid mobile networks, that is, networks that integrate wired, wireless, and cellular technologies [AJSS10].

In addition, in this task we investigated how to estimate the inference risk in a scenario where large datasets are released. In this context, the scope is not protecting single records, but minimizing the risk that statistical inference on the released data may result in confidential information leakage. We proposed a new metric to estimate this risk, based on information theory. The work has been described in a publication by SAP and UNIMI [BDLS10].

Lastly, SAP continued to investigate the relationship between different privacy metrics (k -anonymity, t -closeness, ℓ -diversity), deriving a new framework to enable a direct comparison between these risk indicators. This work has been described in the publication [Bez10].

Task 2.3.2 Techniques for enforcing data privacy. The goal of this task is the design of techniques supporting the management of privacy constraints in relational databases. The work in this task started from the abstract specification of constraints as sets of attributes in a relational schema that must not be accessible together, unless the user has adequate privileges. A first approach assumes that data have to be stored on an outsourced server and designs a fragmentation of the schema, where each fragment

only stores attribute in cleartext, satisfying the confidentiality constraints. The design of the fragmentation considers the profile of queries on the data, in order to identify an efficient solution. The work has been described in publication [CDF⁺10] by UNIMI and UNIBG.

Taking into account the results of the above work and extending the model considered in Task 2.3.1 for the definition of constraints and views, the work [DFJ⁺10c] also contributes to this task, presenting a technique for the protection of private data that relies on the use of fragmentation and the subsequent addition of loose associations. Another line of research started from the same definition of confidentiality constraints and considered a scenario where, together with outsourced data, the trusted user is also able to store data on a local (light-weight) trusted device. Then, a fragmentation is designed which aims at minimizing the amount of local storage used, satisfying all the confidentiality constraints defined on the data. This work has produced a journal publication [CDF⁺11a], by UNIMI and UNIBG.

The enforcement of privacy when dealing with large data collections has also been considered at a legal-theoretical level by TILT, as applied to several concrete topical issues and developments. First, the threats to privacy of data in big databases have been illustrated by analyzing a) Facebook's use of the 'Like' button, which also tracks and traces non-Facebook members, and b) the tagging of pictures in Social Network Sites. This shows that privacy threats arise from data shadows (information others generate about users) at least as much as from digital footprints (information generated by users themselves). Second, the implications of this for protecting user privacy have been analyzed, in particular by exploring novel approaches to legal protection that put less emphasis on individual user control (which is difficult with data shadows) than the current revision of the Data Protection Directive envisions. A dual approach has been explored that alters the relationship between data processors and data subjects in two ways: diminishing upwards transparency (i.e., subjects becoming less visible to data processors) through data obfuscation, and enhancing downwards transparency (i.e., data processors becoming more visible to data subjects) through legal and technical measures. Third, the relationship between legislation and technical enforcement measures ("techno-regulation") has been studied to determine how effective protection of data subjects can best be achieved. An analysis of the three steps involved in techno-regulation (identifying the legal norm, moving from legal norm to techno-rule, and deploying the techno-rule in practice) showed that a trade-off exists between the plasticity and flexibility of technology in techno-regulation and the usefulness and adoption of techno-regulation. The feasibility and limitations of techno-regulation have been studied in relation to three specific issues, with particular attention to the rise of data shadows alongside digital footprints: the legal norm of purpose-binding, the proposed right to be forgotten, and the transparency of tagging. The analysis suggests that embedding legal norms in technical design has considerable potential to enhance compliance with data-protection rules, but also involves trade-offs with feasibility and side-effects that decision-makers must carefully take into account when choosing a mix of regulatory approaches to ensure privacy in big databases.

Task 2.3.3 Efficient organisation and access to privacy-preserving data collections. The goal of this task is the consideration of efficiency issues in access to

protected data collections. Part of the work realized in Task 2.3.2 has an impact on this task. In general, the consideration and design of novel index structures for protected data represents the issue at the center of this research. Effort has been spent in this task for the analysis of recent interesting proposals that have appeared in the literature, with the goal of adapting and improving them, taking into account the requirements of the scenario considered in the project. The result of this work is represented by the definition of the shuffle index, an encrypted B+-tree for the outsourced storage and indexing of data that at every access requires the execution of a shuffling operation. The goal of the shuffling is to destroy the possibility for the server to monitor the sequence of accesses realized by the user. The shuffle index has been described in publication [DFP⁺11], by UNIMI and UNIBG. The shuffle index will be the subject of Chapter 4.

1.4 Access control for the protection of user-generated data (WP2.4)

Today's digital infrastructure provides unprecedented opportunities for the collection and sharing of sensitive information from and about users. As a matter of fact, information about users is continually collected as they complete e-commerce transactions, create accounts, query search engines, and, more in general, use any kind of online services. With the availability of these huge collections of personal data stored at external parties, the potential for disastrous leaks of personal information is tremendously increased. Privacy of the data is then becoming an issue that most people are concerned about and that has captured the attention of many researchers [DFL11, DFS11, DFSL11].

The main goal of this Work Package is to define new models and methods for the definition and enforcement of access control restrictions on user-generated data. These solutions should enhance the user awareness and empowerment, granting users the ability to participate in (and be aware of) the management and dissemination of their data and resources. This is a fundamental aspect for enabling users to live in an electronic society and to enjoy electronic services in the full respect of their privacy.

The advancements in the research activity of the third year of the project resulted in several publications that have appeared in international journals and conferences. The Work Package also continued the work on a tool demonstrating the techniques for the realization of access control policies with encryption. For each task composing the work package, we now provide a brief description of the main research contributions.

Task 2.4.1 Dissemination control and secondary use restrictions. The work in this task focused on an analysis of the solutions for regulating the access and dissemination of different types of personal information (e.g., location information). This problem is particularly critical in several emerging scenarios, such as mobile scenarios where users stay virtually connected anywhere anytime, and where their personal information is easily available and often needed to provide enhanced services. Furthermore, this task focused on the development of a prototype for managing user-generated data (in the specific case, curriculum data), for expressing their usage constraints, and for transferring them to third parties. The prototype has been shown at ICT conference in Brussels.

Task 2.4.2 Access control to confidential data stored at external services. The work in this task addressed the problem of protecting data stored at external servers. Traditional access control solutions [SD01] are not directly applicable since they are based on the assumption that the service provider is fully trusted and can therefore access resource content. In fact, in most practical contexts, the server storing the data is *honest-but-curious* (i.e., it is fully trusted for guaranteeing data availability, but it is not trusted for accessing the sensitive content of outsourced data) [SD10]. Such a scenario introduces new privacy and security concerns that have been analyzed and studied from different perspectives, resulting in different contributions.

The first contribution is the definition of a novel access control technique allowing selective access to outsourced data. The proposed technique maintains sensitive information not intelligible to the storing servers themselves, and supports policy updates in dynamic scenarios [DFJ⁺10b] (see Chapter 5).

The second contribution is a solution that supports users in the specification of access restrictions to resources they wish to share, via an external storage service, with a desired group of other users [DFJ⁺10a]. The proposed solution exploits encryption to attach the access control restrictions to the resources and relies on key agreement and key derivation techniques to ensure manageability and scalability of key management. The proposed approach leverages on solutions proposed for the data outsourcing scenarios, extending them to the consideration of the presence of many users exchanging resources, each having both the role of data owner and data consumer.

The third contribution addressed the problem of computing data releases in the form of *fragments* (vertical views) over a relational table. The fragments satisfy both confidentiality and visibility constraints, expressing needs for information protection and release, respectively [CDF⁺11b]. A new modeling of the fragmentation problem is proposed that exploits the representation of confidentiality and visibility constraints as Boolean formulas, and of fragments as truth assignments over Boolean variables corresponding to attributes in the original relation. In this way, the computation of a fragmentation that satisfies the given constraints greatly depends on the efficiency with which Boolean formulas are manipulated and represented. Since the classical methods for operating on Boolean formulas are impractical for large-scale problems, the proposed solution exploited reduced Ordered Binary Decision Diagrams (OBDDs). OBDDs are a canonical form for Boolean formulas that can be manipulated efficiently, thus being suitable for compactly representing large Boolean formulas [MT98]. The size of an OBDD does not directly depend on the size of the corresponding formula and therefore the complexity of the Boolean operators depends on the OBDD size only. Although the size of an OBDD could be, in the worst case, exponential in the number of variables appearing in the formula, the majority of Boolean formulas can be represented by very compact OBDDs. The proposed approach then consists in transforming all the inputs of the fragmentation problem into Boolean formulas, and in exploiting their representation through OBDDs to process different constraints simultaneously, and to easily check whether a fragmentation reflects the given confidentiality and visibility constraints.

The fourth contribution is a solution for the specification and enforcement of authorizations regulating data release among data holders collaborating in a distributed computation. The proposal ensures that query processing discloses only data whose release has been explicitly authorized [DFJ⁺11]. Authorizations regulate not only the data

on which parties have explicit visibility, but also the visibility of possible associations that such data convey. The simple authorization form essentially corresponds to generic view patterns, thus nicely meeting both expressiveness and simplicity requirements. A novel aspect of the proposed model is the definition of distinct authorization profiles for different parties in the system and the explicit support for cooperative query evaluation. This is an important feature in distributed settings, where the minimization of data exchanges and the execution of a query step in locations where it can be less costly is a crucial factor in the identification of an execution strategy characterized by good performance.

Finally, the last contribution is a framework for protecting the privacy of biometric data [BBC⁺10]. The protection of biometric data is becoming increasingly important in the context of distributed biometric systems, where biometric data are transmitted through a network infrastructure, thus reducing the direct control that users usually have on their biometric information. In these distributed systems, the server that processes the biometric matching should also not learn anything on the database and should not be able to exploit the resulting matching values to extract any knowledge about the user presence or behavior. To this purpose, the implemented system computes the matching task in the encrypted domain by exploiting homomorphic encryption and using the fingerprint template, called Fingercode.

Chapter 2

Cryptographic mechanisms (WP2.1)

One of the main research results of the third year are a number of protocols that allow a user to anonymously access database records such that the database owner is ensured that the accessing user holds the (anonymous) credentials as required by the records access control policy. Nevertheless, the database provider does not learn which record a user accesses.

We have come up with several solutions for this scenario. In the most simple one, the access control policy for each record is known to the user and, when accessing the record the user, the user proves to the database provider that she holds the credentials required by the policy for the record she wants to access. The latter step can be achieved in such a way the the database provider does not learn the policy nor the record the user wishes to access. We then extend this basic protocol to allow the database providers to charge for the different records, where each record can have a different price. In the following we describe these two solutions in more detail.

2.1 Introduction

More and more transactions in our daily life are performed electronically. People enter their credentials online and into various databases and disclose their personal information to different organisations with the belief that small amounts of information cannot reveal enough about them to impact them in a negative way. When using the Internet extensively however, they can give away much more information about themselves than they may care to admit.

Also to protect sensitive information such as medical or financial data, we need to provide strong access control to be sure that only those people who have the necessary permissions can access it. But statistics about what sort of data people query also reveals a lot of information about them.

It is possible to build a complete picture of someone's movements, transactions, locations and relationships from the trail left from interaction with websites and various data-bases. So personal security has become a serious issue.

To protect the users' privacy, it is important that all electronic transactions can be performed without revealing more personal information than is absolutely necessary. Here we consider the case of access to a database where the different records in the database have different access control conditions. These conditions could be certain attributes, roles, or rights that a user needs to have to access the records. The assigning of attributes to users is done by a separate entity called the issuer, external to the database. To provide the maximal amount of privacy, a protocol is required such that:

- Only users satisfying the access conditions for a record can access that record;
- The service (database) provider does not learn which record a user accesses;
- The service (database) provider shall not learn which attributes, roles, etc. a user has when she accesses a record, i.e., access shall be completely anonymous, nor shall it learn which attributes the user was required to have to access the record.

One real-life example where such a protocol is important are DNA databases, containing information about the purpose of each gene. Such databases are extremely valuable and thus there are not sold on a whole, but rather users are charged per access to the database. On the other hand, the particular DNA sequences accessed by a user reveal a lot of information about her interests, e.g., for which disease she is developing medication. Moreover, it is quite likely that subscription prices vary with the different species. Using our protocol, the database can charge different rates for the DNA sequences of mice and apes, without forcing its users to reveal which species they're interested in.

Other examples of databases where users have an interest to keep their queries hidden are stock quotes, since they can reveal information about their investment strategy, and patent search, since they can reveal sensitive business information. Our protocol directly addresses these problems and provides a practical solution for it.

Oblivious transfer with access control. To fulfill all these requirements, we construct an Oblivious Transfer with Access Control (AC-OT) protocol [CDN09a], which is based on the oblivious transfer protocol by Camenisch et al. [CNS07a] and works as follows. Each record in the database has an access control list (ACL). The ACL is a set of categories. We note that the name "category" is inspired by the different data categories that a user is allowed to access. However, the category could just as well encode the right, role, or attribute that a user needs to have in order to access a record.

The database server first encrypts each record with a unique key and publishes these encryptions. The encryption key is derived from the index of the record, the ACL of the record, and a secret of the database server. Although the secret of the database is the same for all record keys, it is not possible to derive the encryption key for one record from that of another record. Thus, to decrypt a record the user needs to retrieve the corresponding key from the server.

To be able to do this, the user has to obtain the necessary credentials from the issuer. Each anonymous credential [Cha85, LRSW99, CL01], issued to a user, certifies a *category* of records the user is allowed to access. Recall that anonymous credentials allow the user to later prove that she possesses a credential without revealing any other information whatsoever. Also, anonymous credential systems provide different revocation mechanisms. Note that if a record has several categories attached to it, then the

user must have a credential for *all* of these categories, basically implementing an AND condition. If one would want to specify an OR condition, one could duplicate the record in the database with a second set of categories.

To obviously access a record for which the user has the necessary credentials, she engages in a transfer protocol with the database and while retrieving a key, gives a zero-knowledge proof of knowledge that she possess credentials on all the categories that are encoded into the key that she wants to retrieve. If she succeeds, she can decrypt that record, otherwise, she cannot. The database learns nothing about the index of the record that is being accessed, nor about the categories associated to the record.

Priced oblivious transfer with rechargeable wallets. Now consider a database where each record may have a different price, for example, DNA or patent database, as described above. In this setting, it is necessary to prevent the database from gathering information about a user's shopping behavior, while still allowing it to correctly charge users for the purchased items.

To solve this problem, we propose the first truly anonymous priced oblivious transfer protocol (POT) [CDN10a], where users load money into their pre-paid accounts, and can then start downloading records so that:

- The database does not learn which record is being purchased, nor the price of the record that is being purchased;
- The user can only obtain a single record per purchase, and cannot spend more than his account balance;
- The database does not learn the user's remaining balance; and
- The database does not learn any information about who purchases a record.

We note that previous POT protocols ([AIR01a, Tob03, RKP09]) do not provide full anonymity (the last requirement) for the users: the database can link transactions of the same user. Furthermore, they also lack a recharge functionality: once a user's balance does not contain enough credit to buy a record, but is still positive, the user cannot use up the balance, but will have to open a new account for further purchases. Even if the protocol can be extended so that the user can reveal and reclaim any remaining credit, he will leak information about his purchases by doing so. In our protocol, users can recharge their balances anonymously at any time.

In addition, we provide an enhanced protocol where records are transferred using an optimistic fair exchange protocol [ASW97, ASW00], thereby preventing a cheating database from decreasing a user's wallet without sending the desired record.

Here, in Priced Oblivious Transfer with Rechargeable Wallets protocol, as with the AC-OT protocol, the database provider encrypts and publishes the entire encrypted database. Each record is encrypted with a unique key that is derived from its index and its price.

To be able to access records, a user first contacts the provider to create a new, empty wallet. Users can load more money into their wallet at any time, using an anonymous e-cash scheme, for example.

When a user wants to purchase a record with index i and price p_i from the database, the provider and the user essentially run a two-party protocol, at the end of which

the user will have obtained the decryption key for the record i as well as an updated wallet with a balance of p_i units less. This is done in such a way that the provider does not learn anything about i or p_i . More precisely, we model wallets as one-time-use anonymous credentials with the balance of the wallet being encoded as an attribute. When the user buys a record (or recharges her wallet), she basically uses the credential and gets in exchange a new credential with the updated balance as an attribute, without the provider learning anything about the wallet's balance. The properties of one-time-use credentials ensure that a user cannot buy records worth more than what she has (pre-)paid to the provider.

Related work. There is of course a large body of works on oblivious transfer which per se offers users access to a database without the server learning the contents of the query. In its basic form, oblivious transfer puts no restrictions on which records a particular user can access. There are a couple of papers that consider oblivious transfer with access control, each of them, however, aiming at a goal different from ours.

Aiello, Ishai, and Reingold [AIR01b] present *priced* oblivious transfer. Here, each record has attached a (possibly different) price. The user holds a (homomorphically) encrypted balance which is reduced with each transfer. Thus, the user can only retrieve records as long as her balance is positive. Another related flavor is *conditional* oblivious transfer, proposed by Di Crescenzo, Ostrovsky, and Rajagopalan [DOR99], where access to a record is only granted if the user's secret satisfies some given predicate. However, none of these protocols offer anonymity to the users.

Herranz [Her08] proposes *restricted* oblivious transfer, which also protects each record with an access control policy. In his case the policy consists of a list saying which user has access to which record, and the user authenticates to the server openly. In contrast, our protocol employs a more powerful attribute-based access control paradigm, and guarantees user anonymity.

To the best of our knowledge, the only paper considering oblivious transfer with access control is the recent work by Coull, Green, and Hohenberger [CGH08]. They propose a scheme for controlling access to records using state graphs. With each access a user transitions from one state to another, where the transition is defined by the index of the record the user has accessed. By restricting the possible transitions between states, a user being in a particular state can only access the records corresponding to the possible transitions.

An exact comparison between our protocols and that of [CGH08] depends on the particular access structure of the database and on how the AC-OT primitive is translated into a graph structure. In general however, our protocol is more efficient because it avoids re-issuing user credentials at each transfer. We discuss two ways of implementing AC-OT using Coull et al.'s protocol below.

2.2 Preliminaries

Let $\text{Pg}(1^\kappa)$ be a pairing group generator that on input 1^κ outputs descriptions of multiplicative groups $\mathbb{G}_1, \mathbb{G}_T$ of prime order p where $|p| > \kappa$. Let $\text{Pg}(p)$ be a pairing group generator that on input p outputs descriptions of multiplicative groups $\mathbb{G}_1, \mathbb{G}_T$ of prime

order p .

Let $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}$ and let $g \in \mathbb{G}_1^*$. The generated groups are such that there exists an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, meaning that (1) for all $a, b \in \mathbb{Z}_p$ it holds that $e(g^a, g^b) = e(g, g)^{ab}$; (2) $e(g, g) \neq 1$; and (3) the bilinear map is efficiently computable.

2.2.1 Modified Boneh-Boyen signatures

We use the following modification of the weakly-secure signature scheme by Boneh and Boyen [BB04]. The scheme uses a pairing generator Pg as defined above.

The signer's secret key is $(x_m, x_1, \dots, x_l) \xleftarrow{\$} \mathbb{Z}_p$, the corresponding public key is $(g, y_m = g^{x_m}, y_1 = g^{x_1}, \dots, y_l = g^{x_l})$ where g is a random generator of \mathbb{G}_1 . The signature on the tuple of messages (m, c_1, \dots, c_l) is the following $s \leftarrow g^{1/(x_m + m + x_1 c_1 + \dots + x_l c_l)}$; verification is done by checking whether $e(s, y_m \cdot g^m \cdot y_1^{c_1} \cdot \dots \cdot y_l^{c_l}) = e(g, g)$ is true.

Security against *weak* chosen-message attacks is defined through the following game: The adversary begins by outputting N tuples of messages $((m_1, c_{1,1}, \dots, c_{1,l}), \dots, (m_N, c_{N,1}, \dots, c_{N,l}))$. The challenger then generates the key pair and gives the public key to the adversary, together with signatures s_1, \dots, s_N on the message tuples. The adversary wins if it succeeds in outputting a valid signature s on a tuple $(m, c_1, \dots, c_l) \notin \{(m_1, c_{1,1}, \dots, c_{1,l}), \dots, (m_N, c_{N,1}, \dots, c_{N,l})\}$.

The scheme is said to be unforgeable under weak chosen-message attack if no PPT adversary has non-negligible probability of winning this game. An adaptation of the proof of [BB04] can be used to show that this scheme is unforgeable under weak chosen-message attack if the $(N + 1)$ -SDH assumption holds.

2.2.2 Zero-knowledge proofs and Σ -protocols

We use various zero-knowledge proofs of knowledge [BG93, CDM00] protocols to prove knowledge of and statement about discrete logarithms. These include

- proof of knowledge of a discrete logarithm modulo a prime [Sch91],
- proof of knowledge of equality of (elements of) representations [CP93],
- proof that a commitment opens to the product of two other committed values [Bra97, CM99, Cam98], and also
- proof of the disjunction or conjunction of any two of the previous [CDS94].

When referring to the proofs above, we will follow the notation introduced by Camenisch and Stadler [CS97] and formally defined by Camenisch, Kiayias, and Yung [CKY09]. For instance, $PK\{(a, b, c) : y = g^a h^b \wedge \tilde{y} = \tilde{g}^a \tilde{h}^c\}$ denotes a “*zero-knowledge Proof of Knowledge of integers a, b, c such that $y = g^a h^b$ and $\tilde{y} = \tilde{g}^a \tilde{h}^c$ holds,*” where $y, g, h, \tilde{y}, \tilde{g}$, and \tilde{h} are elements of some groups $G = \langle g \rangle = \langle h \rangle$ and $\tilde{G} = \langle \tilde{g} \rangle = \langle \tilde{h} \rangle$. The convention is that the letters in the parenthesis (a, b, c) denote quantities of which knowledge is being proven, while all other values are known to the verifier.

Given a protocol in this notation, it is straightforward to derive actual protocol implementing the proof. Indeed, the computational complexities of the proof protocol

can be easily derived from this notation: basically for each term $y = g^a h^b$, the prover and the verifier have to perform an equivalent computation, and to transmit one group element and one response value for each exponent. We refer to, e.g., Camenisch, Kiayias, and Yung [CKY09] for details on this.

2.2.3 Credential signature scheme

We use the signature scheme proposed and proved secure by Au et al. [ASM06], which is based on the schemes of Camenisch and Lysyankaya [CL04] and of Boneh et al. [BBS04].

It assumes cyclic groups \mathbb{G} and \mathbb{G}_T of order p and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The signer's secret key is a random element $x \xleftarrow{\$} \mathbb{Z}_q$. The public key contains a number of random bases $g_1, h_0, \dots, h_\ell, h_{\ell+1} \xleftarrow{\$} \mathbb{G}$, where $\ell \in \mathbb{N}$ is a parameter, and $y \leftarrow g_1^x$.

A signature on messages $m_0, \dots, m_\ell \in \mathbb{Z}_p$ is a tuple (A, r, s) where $r, s \xleftarrow{\$} \mathbb{Z}_p$ are values chosen at random by the signer and $A = (g_1 h_0^{m_0} \dots h_\ell^{m_\ell} h_{\ell+1}^r)^{1/(x+s)}$. Such a signature can be verified by checking whether

$$e(A, g_1^s y) = e(g_1 h_0^{m_0} \dots h_\ell^{m_\ell} h_{\ell+1}^r, g_1) .$$

Now assume that we are given a signature (A, r, s) on messages $m_0 \dots, m_\ell \in \mathbb{Z}_p$ and want to prove that we indeed possess such a signature. To this end, we need to augment the public key with values $u, v \in \mathbb{G}$ such that $\log_{g_1} u$ and $\log_{g_1} v$ are not known. This can be done by choosing random values $t, t' \xleftarrow{\$} \mathbb{Z}_p$, computing $\tilde{A} = Au^t$, $B = v^t u^{t'}$ and executing the following proof of knowledge

$$PK\{(\alpha, \beta, s, t, t', m_0, \dots, m_\ell, r) : B = v^t u^{t'} \wedge 1 = B^{-s} v^\alpha u^\beta \wedge \frac{e(\tilde{A}, y)}{e(g_1, g_1)} = e(\tilde{A}, g_1)^{-s} \cdot e(u, y)^t e(u, g_1)^\alpha e(h_{\ell+1}, g_1)^r \prod_{i=0}^{\ell} e(h_i, g_1)^{m_i} \} ,$$

where $\alpha = st$ and $\beta = st'$.

2.2.4 Set membership scheme

To prove that the user's new balance after buying a record remains positive and is not more than the maximum balance, we use a signature-based set membership protocol suggested by Camenisch, Chaabouni and shelat [CCS08].

They consider a zero-knowledge protocol which allows a prover to convince a verifier that a digitally committed value is an element of a given public set. The verifier signs the individual elements and sends the signatures to the prover. The prover shows that he knows a valid signature (by the verifier) on the element that he holds. The scheme of [CCS08] employs the weak signature scheme by Boneh and Boyen [BB04]. They prove that their protocol is a zero-knowledge argument of set membership for a set Φ , if the $|\Phi|$ -SDH assumption holds.

2.3 Oblivious transfer with access control

An oblivious transfer protocol with access control (AC-OT) is run between the following parties:

- users (U_1, \dots, U_M) known by pseudonyms;
- an issuer I providing access credentials to users for the data categories that they are entitled to access;
- a database DB hosting the list of records and giving users access to those records that they are entitled to access.

In a nutshell, an oblivious transfer protocol with access control works as follows.

1. The issuer I generates his key pair for issuing credentials and publishes the public key as a system-wide parameter.
2. The database server initiates a database containing records protected by access control lists: It generates the encrypted database and makes it available to all users, e.g., by posting it on a website.
3. Users contact the issuer to obtain credentials for the data categories that they want or are entitled to access.
4. When a user wants to access a record in the database, she proves to the database, in a zero-knowledge way, that she possesses credentials for all categories associated with this record. If she succeeds, she can decrypt the record, otherwise, she cannot. The database learns nothing about the index of the record which is being accessed, nor about the categories associated to the record.

2.3.1 Security requirements

We here informally state the security properties which we require from our protocol. For a formal definition of security, we refer to Camenisch et al. [CDN10b].

User Privacy: The database cannot tell which user makes a query, nor can it tell which record is being accessed. That is, the database only gets to know that some user accesses some record for which the user priorly obtained the necessary credentials. If the database colludes with the issuer and potentially with other users, then they can only try to identify the user or her selection based on which credentials were issued to whom, and which credentials are necessary to successfully access which record.

Database Security: A cheating user alone cannot access a record for which she does not have the necessary credentials. Colluding users cannot pool their credentials, meaning that they cannot access any records that none of them would have been able to obtain individually. If the issuer colludes with one or more users, they can only obtain as many records from the database as the number of transfer queries that were performed.

2.3.2 Construction

We now describe our scheme in detail. We model access control lists as tuples of exactly ℓ categories $ACL_i = (c_{i1}, \dots, c_{i\ell}) \in \mathcal{C}^\ell$. A record can therefore be associated with at most ℓ categories; unused entries are filled with a dummy category $c_{ij} = \text{dummy}$ for which we assume every user is given a credential for free. To issue anonymous credentials, we employ the signature scheme presented in Section 2.2.3. and to implement the oblivious access control we extend the protocol by Camenisch et al. [CNS07b]. We will also use a number of proof protocols about discrete logarithms as described in Section 2.2.2

Initial setup. We now describe the setup procedures of the issuer and the database provider. Users do not have their own setup procedure.

ISetup(C):

```

 $(\mathbb{G}, \mathbb{G}_T, p) \xleftarrow{\$} \text{Pg}(1^\kappa)$ 
 $g_t, h_t \xleftarrow{\$} \mathbb{G}_T^*$ ;  $g_1, h_0, h_1, h_2, u, v \xleftarrow{\$} \mathbb{G}^*$ ,  $x_I \xleftarrow{\$} \mathbb{Z}_p$ ;  $y_I \leftarrow g_1^{x_I}$ 
 $sk_I \leftarrow x_I$ ;  $pk_I \leftarrow (g_1, h_0, h_1, h_2, u, v, w, g_t, h_t, y_I)$ 
Return  $(sk_I, pk_I)$ 

```

Figure 2: Issuer setup algorithm

To set up its keys, the issuer runs the randomized ISetup algorithm displayed in Figure 2. This will generate groups of prime order p , a public key pk_I and a corresponding secret key sk_I for security parameter κ and category universe \mathcal{C} . He publishes the public key as a system-wide parameter.

DBSetup($pk_I, DB = (R_i, ACL_i)_{i=1, \dots, N}$):

```

 $(\overline{\mathbb{G}}, \overline{\mathbb{G}}_T) \xleftarrow{\$} \text{Pg}(p)$ ;  $g, h \xleftarrow{\$} \overline{\mathbb{G}}^*$ ;  $H \leftarrow \overline{e}(g, h)$ 
 $x_{\text{DB}} \xleftarrow{\$} \mathbb{Z}_p$ ;  $y_{\text{DB}} \leftarrow g^{x_{\text{DB}}}$ 
For  $i = 1, \dots, \ell$  do  $x_i \xleftarrow{\$} \mathbb{Z}_p$ ;  $y_i \leftarrow g^{x_i}$ 
 $sk_{\text{DB}} \leftarrow (h, x_{\text{DB}}, x_1, \dots, x_\ell)$ ,  $pk_{\text{DB}} \leftarrow (g, H, y_{\text{DB}}, y_1, \dots, y_\ell)$ 
For  $i = 1, \dots, N$  do
  Parse  $ACL_i$  as  $(c_{i1}, \dots, c_{i\ell})$ 
   $E_i \leftarrow g^{\frac{1}{x_{\text{DB}} + \sum_{j=1}^{\ell} x_j \cdot c_{ij}}}$ ,  $F_i \leftarrow e(h, E_i) \cdot R_i$ ,  $ER_i \leftarrow (E_i, F_i)$ 
Return  $((pk_{\text{DB}}, ER_1, \dots, ER_N), sk_{\text{DB}})$ 

```

Figure 3: Database setup algorithm

To set up the database, the database provider runs the algorithm shown in Figure 3. That is, it uses the issuer's public key and a pairing group generator to create groups of the same order p and generate keys for encrypting records. First, the database provider chooses its secret key x_{DB} . Next he encrypts each record R_i as (E_i, F_i) , each with its own key. These keys do not only depend on the database provider's secret key (x_{DB}), but also on the index of the record (i) and the categories defined in the access control policy for the record ($\{x_c\}_{c \in \bigcup_{i=1}^N ACL_i}$). The pairs (E_i, F_i) can be seen as an ElGamal

encryption [ElG85] in $\overline{\mathbb{G}}_T$ of R_i under the public key H . But instead of using random elements from $\overline{\mathbb{G}}_T$ as the first component, our protocol uses verifiably random [DY05] values $E_i = g^{\frac{1}{x_{DB} + i + \sum_{j=1}^{\ell} x_j \cdot c_{ij}}}$. It is this verifiability that during the transfer phase allows the database to check that the user is indeed asking for the decryption key for one particular records with a particular access control policy for which user has appropriate credentials.

Issuing credentials. To be able to make database queries, a user needs to obtain the credentials for the categories she is allowed to access. To this end, the user runs the **Issue** protocol with the issuer as depicted in Figure 4. We leave open how the issuer determines which user has access to which categories, but we do assume that the communication links are authenticated, so the issuer knows which user it is talking to.

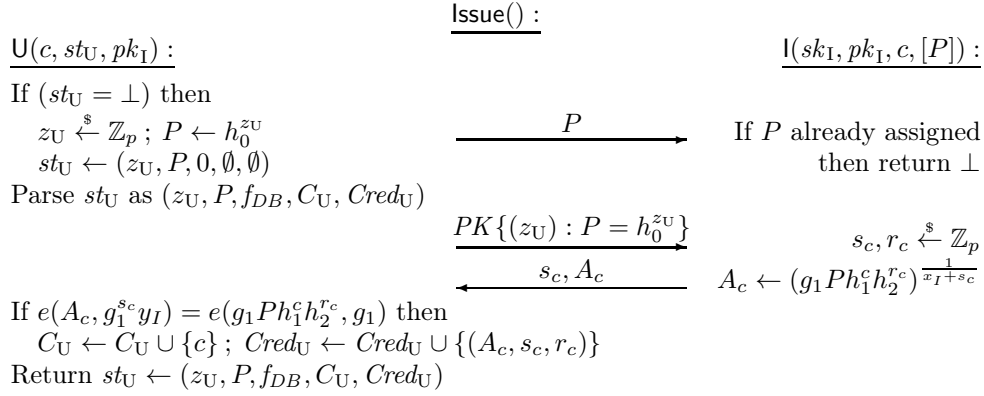


Figure 4: Issue protocol

Apart from the issuer's public key, the user's input also includes her state $st_U = (z_U, P, f_{DB}, C_U, Cred_U)$, which is a tuple containing her master secret, her pseudonym, a bit f_{DB} indicating whether she already accessed the database, the set of categories C_U to which she currently has access, and the corresponding credentials $Cred_U$. The input of the issuer contains his secret and public key, the category c for which the user wants a credential, and the pseudonym P of the user, if she registered one before.

If the user runs the issuing protocol for the first time, her input will contain the empty state ($st_U = \perp$). In this case, the user first generates her master secret z_U and calculates her pseudonym $P = h_0^{z_U}$, sends P to the issuer, and then initializes her state as $st_U = (z_U, P, 0, \emptyset, \emptyset)$.

As a result of the issuing protocol, the user will obtain an access credential for the category $c \in \mathcal{C}$. This credential is a tuple $cred_c = (A_c, s_c, r_c)$ which can be verified by checking $e(A_c, g_1^{s_c} y_I) = e(g_1 P h_1^c h_2^{r_c}, g_1)$. We assume that the user and the issuer run the issuing protocol for each category for which the user is allowed to obtain a credential individually. It is not hard to see how to issue the credentials for all of the user's categories at once.

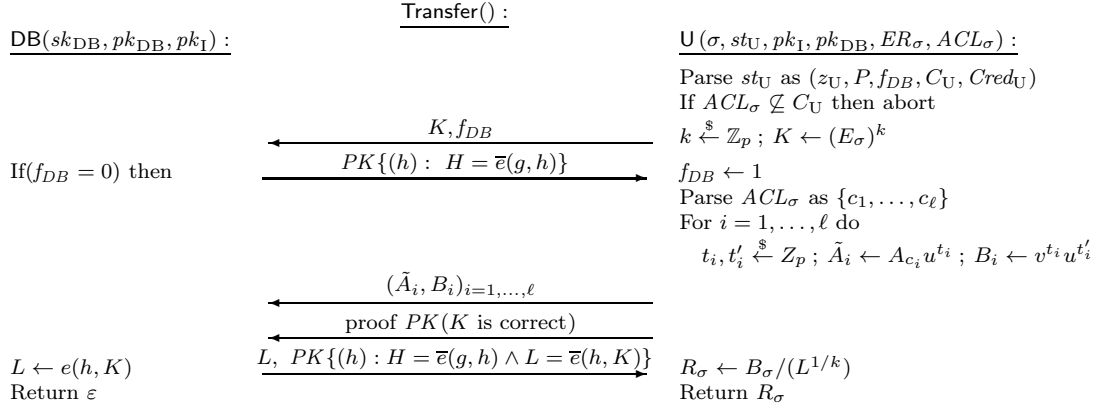


Figure 5: Transfer protocol: The proof $PK(K \text{ is correct})$ is detailed in the text

We note that credential (A_c, s_c, r_c) is a signature as defined in Section 2.2.3 on the set of messages (z_U, c) , where z_U is the user's master secret.

Accessing a record. When the user wants to access a record in the database, she engages in a **Transfer** protocol (Figure 5) with the database server.

The input of the database server is her secret and public key as well as the public key of the issuer. The input of the user is the index σ of the record that she wants to access, the encryption $ER_\sigma = (E_\sigma, F_\sigma)$ of that record, the access control policy of the records, her state (containing all her credential), and the public keys of the issuer and the database.

If this is the first transfer protocol she executes with this database (i.e., $f_{DB} = 0$), then the user asks the database to execute a proof of knowledge of the database secret key h . This zero-knowledge proof will enable to decrypt the contents of the database in the security proof.

Then the user randomizes E_σ and sends this randomized version K to the database. Note that E_σ is derived from the database provider's secret key, the index of the records, and, most importantly all the categories of the record.

Next the user proves that K is correctly formed as a randomization of some E_i for which she possesses all necessary credentials. She uses the following proof $PK(K \text{ is correct})$ for this:

$$PK\left\{(\sigma, k, z_U, (c_i, s_{c_i}, r_{c_i}, t_i, t'_i, \alpha_i, \beta_i)_{i=1, \dots, \ell}) : \right.$$

$$\bar{e}(K, y_{DB}) \bar{e}(K, g)^\sigma \prod_{i=1}^{\ell} \bar{e}(K, y_i)^{c_i} = \bar{e}(g, g)^k \wedge \bigwedge_{i=1}^{\ell} (B_i = v^{t_i} u^{t'_i} \wedge 1 = B_i^{-s_{c_i}} v^{\alpha_i} u^{\beta_i} \wedge$$

$$\left. \frac{e(\tilde{A}_i, y_I)}{e(g_1, g_1)} = e(\tilde{A}_i, g_1)^{-s_{c_i}} e(u, y_I)^{t_i} e(u, g_1)^{\alpha_i} e(h_2, g_1)^{r_{c_i}} e(h_0, g_1)^{z_U} e(h_1, g_1)^{c_i} \right\}$$

If the database provider accepts the proof, it computes L from h and K , sends L to the user, and proves that L was computed correctly.

The protocol is easily seen to be correct by observing that $L = e(h, E_\sigma)^k$, so therefore $F_\sigma/L^{1/k} = R_\sigma$.

2.4 Extending to priced oblivious transfer

We now explain how to extend the construction from the previous sections so the database provider can assign to each records a price to be paid by the user.

2.4.1 Construction overview

The high-level idea is as follows. The price of a record is treated as a special category. Thus, the database provider encrypts each record with a key that is derived from not only its index and the access control list but also from its price. It then publishes the entire encrypted database.

Before being able to access records, a user first contacts the provider to create a new, empty wallet. users can then load (more) money into their wallet at any time. When a user wants to purchase a record with index σ and price p from the database, the provider and the user essentially employ the same protocols as in the previous section except that now the user also proves to the database that her wallet contains more than p monetary units. After the protocol, the user will have obtained the decryption key for the record σ as well as an updated wallet with a balance of p units less. This is done in such a way that the provider does not learn anything about σ or p . More precisely, we model wallets as one-time-use anonymous credentials with the balance of the wallet being encoded as an attribute. When the user buys a record (or recharges her wallet), she basically uses the credential and gets in exchange a new credential with the updated balance as an attribute, without the provider learning anything about the wallet's balance. The properties of one-time-use credentials ensure that a user cannot buy records worth more than what she has (pre-)paid to the provider. We prove our protocol secure in the standard model (i.e., without random oracles).

2.4.2 Additional database keys

To prove that the customer's new balance after buying a record remains positive and is not more than the maximum balance we use a signature-based set membership protocol suggested by Camenisch, Chaabouni and Shelat [CCS08].

They consider a zero-knowledge protocol which allows a prover to convince a verifier that a digitally committed value is an element of a given public set. The verifier signs the individual elements and sends the signatures to the prover. The prover shows that he knows a valid signature (by the verifier) on the element that he holds. The scheme of [CCS08] employs the weak signature scheme by Boneh and Boyen [BB04]. They prove that their protocol is a zero-knowledge argument of set membership for a set Φ , if the $|\Phi|$ -SDH assumption holds.

Let $p_{max} \leq b_{max} < 2^{\kappa-1} < q/2$ be the maximal balance that can be stored in a customer's wallet. To prove that the customer's new balance after buying a record

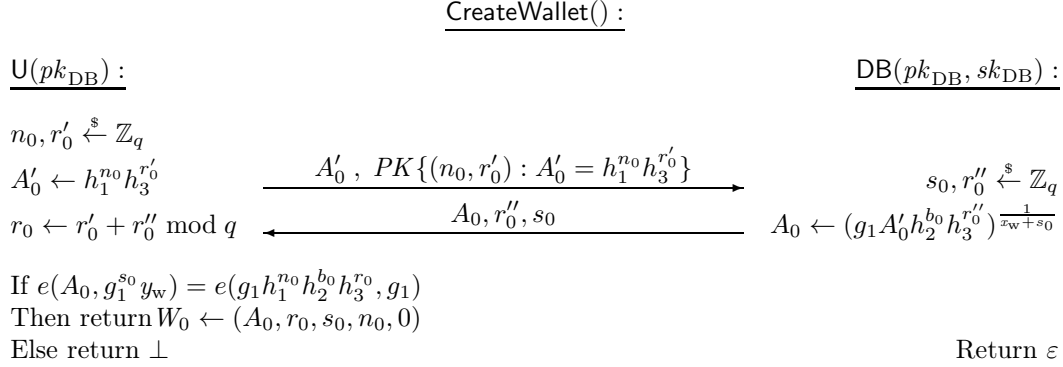


Figure 6: Create wallet protocol

remains positive and is not more than the maximum balance, we use a signature-based set membership protocol of Section 2.2.4. Thus, to authenticates all possible balances, the database chooses a secret key x_b and computes the public key $y_b = g^{x_b}$. Here the set contains all possible balances from the customer's wallet $\{0, \dots, b_{\max}\}$. So for each possible balance $0 \leq i \leq b_{\max}$ the database provider uses x_b to compute a signature $\{y_b^{(i)} = g^{1/(x_b+i)}\}$. These values are included in the database's public key; they will be used by the customer to prove that her balance remains positive after subtracting the price of the purchased record.

To issue wallets, the database uses the same parameters for issuing credentials as the issuer, except that it of course has its own secret key x_w and public key $y_w = g^{x_w}$.

These elements x_b, x_w and y_b, y_w are added to the overall secret and public key of the database.

2.4.3 Obtaining wallets

Before purchasing any records, users first need to create an empty wallet and then charge it with money. To create a wallet, the user runs the **CreateWallet** protocol with the database provider.

The database provider's public key pk_{DB} is a common input. The database provider has his secret key sk_{DB} as a private input. At the end of the protocol, the user obtains a wallet $W_0 = (A_0, r_0, s_0, n_0, b_0 = 0)$ signed by the database provider. Here, (A_0, r_0, s_0) is essentially a signature as per the scheme of Section 2.2.3 of a serial number n_0 chosen by the user and the initial balance of the wallet $b_0 = 0$. Next, the user verifies the wallet's signature and outputs W_0 if the check is successful.

2.4.4 Recharge protocol

Users can recharge the balance of their wallets by engaging in a **Recharge** protocol (Figure 7) with the database server. Doing so does not reveal the remaining balance in the wallet, nor whether this is a freshly created wallet or an updated wallet obtained after purchasing a record. The common inputs are the database provider's public key pk_{DB}

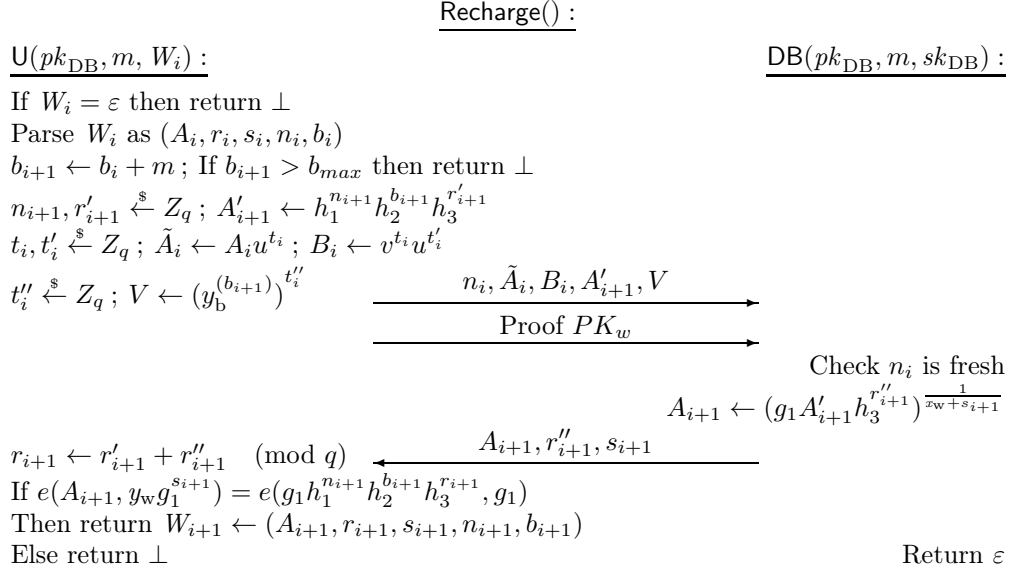


Figure 7: Recharge protocol: The proof protocol PK_w is defined in the text

and the amount of money m that the user wants to add to her balance. The database's secret key sk_{DB} and the user's current wallet W_i are private inputs to the database and the user, respectively.

If the user already obtained a wallet earlier (her state is not empty), she updates her balance to $b_{i+1} = b_i + m$ and generates a fresh serial number n_{i+1} and a randomizer r'_{i+1} for the new wallet. Then she chooses from the set of database signatures $y_b^{(0)}, \dots, y_b^{(b_{max})}$ of possible balances the signature corresponding to her new balance and blinds it as $V = (y_b^{(b_{i+1})})^{t''_i}$. This allows her to next prove that her new balance b_{i+1} is positive and less than b_{max} with the set membership scheme from [CCS08]. The user further proves that she correctly increased her balance by the amount m being deposited. This proof protocol PK_w is defined as follows

$$PK\{(r_i, s_i, n_{i+1}, b_{i+1}, r'_{i+1}, t_i, t'_i, t''_i, \alpha_i, \beta_i) : A'_{i+1} = h_1^{n_{i+1}} h_2^{b_{i+1}} h_3^{r'_{i+1}} \wedge e(V, y_b) = e(V, g)^{-b_{i+1}} e(g, g)^{t''_i} \wedge B_i = v^{t_i} u^{t'_i} \wedge 1 = B_i^{-s_i} v^{\alpha_i} u^{\beta_i} \wedge \frac{e(\tilde{A}_i, y_w)}{e(g_1 h_1^{n_i} h_2^{-m}, g_1)} = e(u, y_w)^{t_i} e(\tilde{A}_i^{-s_i} u^{\alpha_i} h_3^{r_i} h_2^{b_{i+1}}, g_1)\} .$$

The database provider checks whether the proof is valid and whether the serial number n_i is fresh, i.e., whether it previously saw the number n_i . If not, then the database decides that the user is trying to overspend and aborts. Otherwise, if the database provider accepts the proof, it signs the user's new wallet with updated balance and sends it to the user. The user checks the validity of the signature on her new wallet, and if it verifies correctly, outputs an updated state containing the new wallet W_{i+1} .

2.4.5 Modified transfer protocol

Now to access a record, the user and the database run a combination of the recharge protocol (Figure 7) and the transfer protocol (Figure 5). That is, the user has to first prove that 1) she possesses all necessary credentials to access the record, 2) her wallets contains a balance that is larger than the price of the record. If these conditions are met, the users exchanges her wallet for a new wallet with the reduced balance and a key allowing her to decrypt the record. We leave the details of this as an exercise to the interested reader.

Chapter 3

Mechanisms supporting users' privacy and trust (WP2.2)

The area of *mechanisms supporting users' privacy and trust* is very broad. It ranges from transparency and awareness considerations related to privacy over trust/reputation and peculiarities of privacy in and of collaborative groups to privacy measurement. An overview of the individual results of this research field is given in Section 1.2. In this chapter, we will focus on one particular research outcome specifically addressing privacy-preserving event scheduling.

3.1 Event scheduling

Event scheduling represents one of the most popular evolved Web 2.0 functionalities. There is a huge list of applications which allow users to create web polls (e.g., Doodle, Moreganize, Meet-O-Matic).¹ Scheduling events typically comprises three steps:

1. *Poll initialization*: An initiator who wants to schedule an event creates a poll and sends a link to the poll to potential participants.
2. *Vote casting*: Each participant has to fill in his so-called *availability pattern*.
3. *Result publication*: By analyzing the availability patterns of all participants, a meeting date can be scheduled that fits best.

3.2 Security requirements

Casting a vote and *analyzing the result* are the two operations a user of a Web 2.0 event scheduling application wants to perform. Translating these two operations to classical access control terminology would mean to *write* to a poll and to *read* from it.

¹doodle.com, moreganize.ch, meetomatic.com

Besides these functional requirements, event scheduling requires to consider also security requirements. Basically, these can be divided into the two classical protection goals *confidentiality* and *integrity*.²

3.2.1 Confidentiality

Availability patterns often contain sensitive information in at least two respects. First, it is possible to read information directly out of the pattern (e.g., “Does my boss work after 3pm?”, “Will my husband vote for the date of our wedding anniversary?”). Second, indirect inference arises from the fact that availability patterns contain much entropy and thus often allow to (re-)identify individuals who would otherwise remain pseudonymous (e.g., “The participant stating this particular availability pattern goes to lunch every day at 11:30. It therefore has to be Peter.”, “The availability pattern of user bunny23 looks suspiciously like the one of my employee John Doe!”).

Users of web applications have particular privacy requirements and, depending on the context, the amount and the kind of data they are willing to disclose is different. As we do not know in advance how sensible the data is that a user will disclose within a certain web poll, only necessary data should be disclosed to the smallest possible group of people.

3.2.2 Integrity

Apart from privacy threats, users want to be sure that the results of the votes are correct and complete. Every participant has to be convinced that the availability patterns of the participants displayed at the poll belong to the names they claim to belong to and are not forged by somebody.

This protection goal stands in contrast to confidentiality. In case of a poll allowing every participant to act pseudonymously, each of them should be sure that only the authorized participants state their votes and that every participant may vote only once.

3.2.3 Involved parties

After discussing *what* to protect, we shortly take a look on the question against *whom* data have to be protected. Involved parties in event scheduling are the following.

- *Initiator*. The person who sets up a poll in order to organize an event.
- *Participant*. Somebody who should participate in the event and therefore should state his availability pattern.
- *Server administrator*. Somebody who has physical access to the poll server, at which the poll is conducted.
- *Network provider*. Somebody who has physical access to the network between the user and the poll server.

²We want to skip the discussion of the third classical protection goal – *availability* – here, as this discussion would not be specific to event scheduling, but it is a general discussion to every web application.

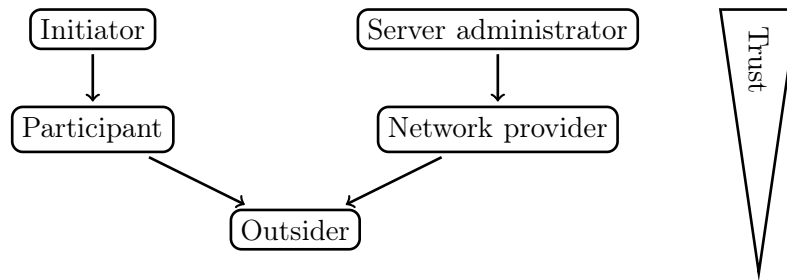


Figure 8: The necessary amount of trust one needs to have about the involved entities: The arrows indicate which role an entity can take

- *Outsider.* Every other person.

Note that some of these persons may act in different roles at the same time. To give an example, the initiator of a poll usually is also a participant and, therefore, he may act in both roles. Thus, the trust one has to assume about a participant is at least as high as the trust one has to assume about the initiator of the poll. Figure 8 shows these dependencies in a diagram. Thereby, the arrows indicate which role an entity can take. The amount of trust which one needs to have in an entity decreases within the direction of the arrows.

3.3 Existing applications

As already stated, a huge amount of Web2.0 event scheduling applications is already available. Within this section, we discuss how the security requirements introduced above (cf. Section 3.2) are fulfilled by these applications. The section is organized according to the respective approach of access control within the existing applications (i.e., what users have to do to write and read within the application). Note that we will skip the discussion on protecting the poll against the network provider as in the context of a web application, a simple SSL encryption already offers enough protection to make the network provider no more powerful than outsiders.

The simplest case which all applications support is a scenario, where no credential is needed to interact with the application. A concrete scenario would be that some initiator posts a link on a public mailing list, encouraging all potential readers to state their preferences. In such a case, one has to trust all involved entities with respect to confidentiality and integrity.

3.3.1 Protection against outsiders

Password protection is a simple protection against outsiders. Password protection of write accesses protects the integrity of the poll whereas password protection of read access protects the confidentiality of the poll data.

In fact, all of the surveyed applications offer a password protection functionality. This is due to the fact that the URL of the polls typically is generated at random and contains enough entropy (e.g., <http://www.moreganize.ch/bzcGgZx4vpG>). However, as every participant uses the same password, the protection helps against outsiders only and not against other participants.

Table 2 on the facing page shows an overview of all existing schemes. The credentials a user has to show to participate in a poll is shown in the second and third column. The last four columns show, if the protection goals confidentiality and integrity are fulfilled with respect to the given entities. For this scheme (the line for Section 3.3.1) one can see that confidentiality and integrity is achieved against outsiders only.

3.3.2 Integrity protection against participants

Instead of using the *same* password for every participant, a *different*, user-specific password can be used. This ensures that no participant may change the votes of others and therefore it offers integrity protection. The passwords may either be chosen by the participants themselves or generated randomly for each poll and for each participant from the server. The advantage of generating passwords at poll initialization is that users do not have to register in order to protect their vote against manipulation.

















Implementations of both variants can be found in existing applications. User registration is supported, for example, by Doodle, Moreganize, and agreeAdate (agreeAdate.com). Diarised (diarised.com), Moreganize, and agreeAdate may generate personalized links at poll initialization. Assuming nobody reads the password e-mails, every participant knows that the availability pattern really belongs to the person owning a particular e-mail address.

3.3.3 Confidentiality protection against participants

To fulfill privacy needs of participants, existing applications implement so-called hidden polls. Within these polls, the result is not shown to every participant. There, it is available at a different (non-guessable) URL only, which is given to the initiator during the poll initialization. If one trusts the initiator with respect to confidentiality *and* integrity, the information contained in the availability patterns is seen as being protected.

3.3.4 Protection against the initiator

The most confidential scheme found within the existing applications is a scheme, where confidentiality protection against all entities but the server administrator is achieved. In this case, the data is aggregated in a way that the entropy of the result is low enough to guarantee confidentiality protection. The most commonly used aggregation is to sum up all available participants per date (e.g., found at Moreganize and the survey applications LimeSurvey - limesurvey.com - and SurveyMonkey - surveymonkey.com). In theory, user passwords might protect the integrity in such a case, but to the best of our knowledge there is no application offering sum-only schemes and user specific passwords.

Section	cast vote (write)	analyze result (read)	initiator	participant	outsider	server admin
3.3.1	poll password	poll password			 	
3.3.2	user password	poll password			 	
3.3.3	user password	initiator password		 	 	
3.3.4	user password	display sum only	 	 	 	



 ... confidentiality protection,  ... integrity protection

Table 2: Overview of existing event scheduling schemes. There is no scheme, which tries to overcome the need of a trustworthy server administrator

3.4 New schemes without server trust

Our small survey has shown that there are lots of applications offering very different solutions for protecting confidentiality and integrity. However, none of these solutions overcome the trust assumptions needed in the poll server. This can be easily seen in Table 2, where the last column is empty. We therefore describe an approach to enhance the existing schemes with protection against the server administrator.

The section is organized very similar to the former one where we presented the schemes with decreasing trust assumptions. Within this section, we will present one scheme per subsection each related to a subsection of Section 3.3. Each scheme in Section 3.4.x has the same trust assumptions as its corresponding one in Section 3.3.x, except that it overcomes server trust additionally. In each subsection, we first explain the scheme and then discuss how we implemented the scheme in our Web 2.0 application.

3.4.1 Protection against outsiders

The simple password protection in Section 3.3.1 prevents unauthorized modification and unauthorized reading of data. If we use the password as key for symmetric authentication, the same protection can be achieved without the need of trusting the server. If the password is used additionally for encryption, the poll is secured against unauthorized read access and therefore confidentiality is achieved with respect to others *and* the server administrator.

Implementation. To leave the user experience of an implementation of such a scheme untouched, we used JavaScript to encrypt and authenticate the availability patterns. Therefore, all data is encrypted with AES-CCM, which guarantees encryption and authentication at the same time. To derive a key from a password, we use PBKDF2 from

$$\underbrace{\text{https://}}_{\text{scheme}} \underbrace{\text{dudle.inf.tu-dresden.de}}_{\text{host}} \underbrace{\text{/somepoll/}}_{\text{path}} \underbrace{\text{\#passwd=rvoj4pej}}_{\text{fragment}}$$

```

1 $ telnet -z ssl dudle.inf.tu-dresden.de 443
2 GET /somepoll/ HTTP/1.0
3 Host: dudle.inf.tu-dresden.de

```

Figure 9: Construction of a URL (RFC3986 [BLFM05]) and its partitioning when doing a GET request

RFC2898 [Kal00].³ All cryptographic operations are done using the Stanford JavaScript Crypto Library [SHB09].

To enhance the usability with respect to the password entry, the user can access the poll through a special URL. Here, we use the fragment part of the URL, which is usually used to address a document anchor. A document anchor is a reference to a certain element within an HTML document. The anchor permits to jump to the corresponding element after the page load is complete. The document anchor is specified within the fragment part of the URL.

Figure 9 shows an example of a URL with fragment part. The host part as well as the path of this URL is transmitted to the server when performing a GET request. Everything after the hash (#) is the fragment part of the URL. The fragment part is *not* transmitted inside the GET request. Therefore it can be used for sensible data like the password, which must stay local.

Figure 10 shows a screenshot of the vote casting user interface. It demonstrates that the interface does not look significantly different from the applications without cryptography. As with the schemes with weaker trust assumptions (cp. Section 3.3.1), a user has to click on a link and enter his preferences. Encryption, decryption, and authentication is realised transparently in the background using JavaScript.

3.4.2 Integrity protection against participants

To protect the integrity of the votes against other participants, existing applications use different passwords for each participant (cp. Section 3.3.2). If one chooses a digital signature scheme instead of user passwords, no trust assumptions against the server administrator regarding the integrity of a vote are needed.

Implementation. To validate that the signatures of participants are correct, one has to assume that the public keys are exchanged correctly. In our implementation we used PGP as an existing public key infrastructure to make this verification easier. Therefore a participant may sign his vote with his PGP-key before sending it to the server. Note that we did *not* implement signing in JavaScript as, in this case, a user would have to enter his decrypted secret key into the browser. This would mean that the user has

³Key stretching [KSHW97] of 10Bit is done with the salted password.

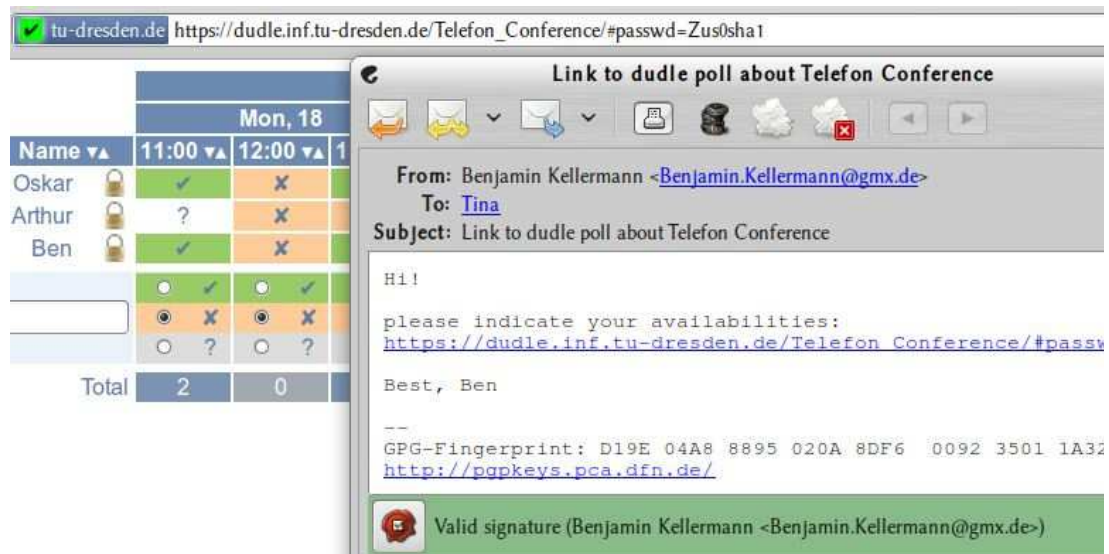


Figure 10: Compared to existing applications, the user interface does not change. The password is transmitted within the fragment part of the URL, where it can be extracted using JavaScript and be used for AES encryption and authentication

to trust that neither JavaScript nor some browser extension would leak the secret key which one may use for e-mail encryption additionally. Instead of using JavaScript-based signing, a user can make use of one of the various PGP/GPG-Applets (e.g., Seahorse, KGpg, GPA, FireGPG, Cryptophane, GnuPG-Shell).

In contrast to signing, verification can be done using JavaScript since only public keys are involved. Figure 11 shows a screenshot of the implementation. The public key of the signed votes is downloaded from a key server and the signature of the signed availability pattern is checked. If the signature is correct, a seal-icon is displayed beneath the name of the participant. Hovering the icon displays the PGP fingerprint of the corresponding key.

3.4.3 Confidentiality protection against participants

The way to achieve confidentiality within existing applications is to require the initiator password for displaying the result of the votes (cp. Section 3.3.3). To reduce trust in the server, the votes can be encrypted asymmetrically to the poll initiator. The initiator therefore specifies a public key at poll creation, to which the votes are encrypted.

Implementation. As the encryption uses the public key only, it can be realised using JavaScript⁴. Therefore, the only difference of the user interface compared to those of the existing applications is that the user is informed by a small note indicating the encryption

⁴The library of Herbert Hanewinkel is used here [Han05].

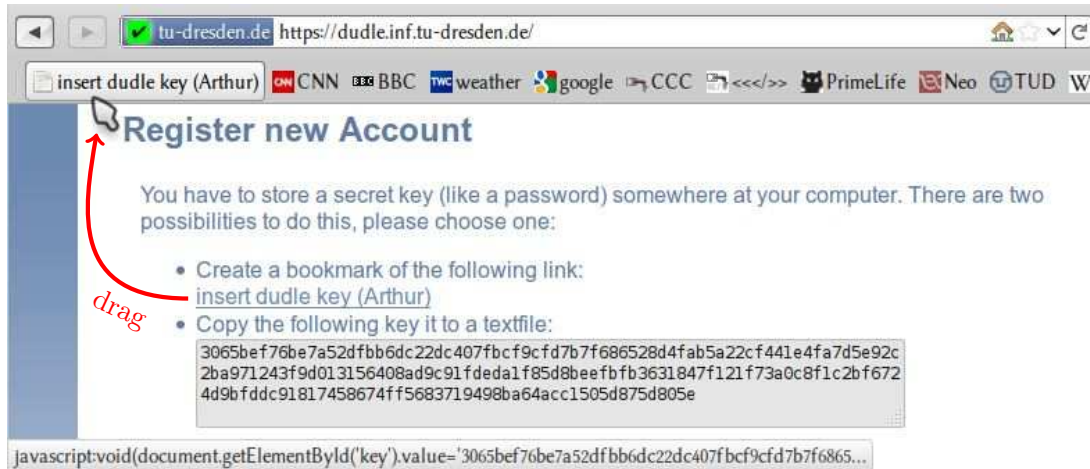


Figure 13: Interface to save a private key at client side

We therefore propose a completely new scheme which guarantees that after performing a poll:

- only the previously determined participants have participated maximal once in the poll and
- nobody is able to see single votes of the participants in clear text.









































These two properties are achieved if at least one participant does not cooperate with the attacker. The whole scheme has a complexity of $\mathcal{O}(n)$, so it needs only one asymmetric operation per participant.

On a more technical level: Each participant encrypts his vote with a homomorphic encryption (like in a DC-Net [Cha88]). This encryption is implemented in JavaScript within the user's browser at the client side. Each participant will see the encrypted votes only and is able (due to the homomorphic property) to calculate the result of the poll without decrypting the other participants' votes. However, the whole scheme is too complex to be described here, interested readers should refer to [KB09, Kel11].

Implementation. To exchange keys between participants, Diffie–Hellman key agreement [DH76] is used. The user has to generate a key pair and store the secret key at his client computer. To make this key storage easier, a so-called *Bookmarklet*⁵ is used. A screenshot of this registration step is shown in Figure 13. There, a link is displayed to the user together with a request to store it as a bookmark. The user can drag the link and drop it at his bookmark bar.

We tested the interface of Figure 13 and found out that the majority of the test persons clicked at the link instead of creating a bookmark from it. We cannot change

⁵Bookmarklet is a portmanteau of *bookmark* and *applet*. It denotes a browser bookmark, which contains JavaScript code.

Section	cast vote (write)	analyze result (read)	initiator	participant	outsider	server admin
3.3.1	poll password	poll password			 	
3.4.1	symmetric authentication	symmetric encryption			   	
3.3.2	user password	poll password			 	
3.4.2	digital signature	symmetric encryption			   	
3.3.3	user password	initiator password		 	 	
3.4.3	digital signature	asym. enc. to initiator		 	   	
3.3.4	user password	display sum only	 	 	 	
3.4.4	digital signature	sum only computable	 	 	   	



 ... confidentiality protection,  ... integrity protection

Table 3: Overview of different event scheduling schemes

the target of the link (`href`) to display some help as this would destroy the semantics of it and would result in a wrong bookmark.⁶ Therefore, we added an `onClick`-Handler, which displays a browser-specific help page explaining how to create a bookmark. The source code of the link looks similar to the following:

```

1 <a href="javascript:void(document.getElementById('key')
2 wp.value='0xDEADBEEF') " onClick="help()">insert key</a>

```

When participating in a poll, the user is asked to click on this bookmark. This inserts the key into the corresponding field, which is necessary to perform the encryption. Apart from this necessary key handling procedure, the user interface behaves like the interface in non-privacy-enhanced polls.

3.4.5 Conclusions

A structured view on all schemes discussed in this chapter, is provided in Table 3. Note that it is not possible to derive “the best” scheme from this table as there are further constraints not shown there.

For example, it might be the case that some of the participants are more important to attend a meeting than others and the whole decision function, which would determine a winning time slot needs more information than only the sum of all available participants and is too complex to be specified from the beginning. Therefore, it might be the wish of

⁶In Figure 13 one can see the sourcecode of the bookmarklet at the bottom of the screenshot, which necessarily has to be the `href` of the link.

all participants to know which participant is available at which time. In such a scenario one would have to stick to the scheme described in Section 3.4.1 and it would not be possible to increase confidentiality further.

All presented schemes are available with the prototype and allow users to use the application with the amount of confidentiality and integrity, they are willing to have.

Privacy of data (WP2.3)

The research and industrial communities have been recently showing considerable interest in the outsourcing of data and computation. The motivations for this trend come from the economics of system administration, which present large scale economies, and by the evolution of ICT, which offers universal network connectivity that makes it convenient for users owning multiple devices to store personal data into an external server. A major obstacle toward the large adoption of outsourcing, otherwise particularly attractive to individuals and to small/medium organizations, is the perception of insecurity and potential loss of control on sensitive data and the exposure to privacy breaches. Guaranteeing privacy in a context where data are externally outsourced entails protecting the confidentiality of the data as well as of the accesses to them. In particular, it requires to maintain confidentiality on: the data being outsourced (*content confidentiality*), the fact that an access aims at a specific data (*access confidentiality*), the fact that two accesses aim at the same data (*pattern confidentiality*).

Several solutions have been proposed in the past few years, both in the theoretical and in the system communities, for protecting the confidentiality of the outsourced data. Typically, such solutions consider a honest-but-curious server (i.e., a server trusted to provide the required storage and management service but not authorized to read the actual data content) and resort to encryption to protect the outsourced data. Since the server is not allowed to decrypt the data for access execution, these solutions provide different techniques for elaborating queries on encrypted data. Furthermore, they aim at content confidentiality but do not address the problem of access and pattern confidentiality.

Access and pattern confidentiality have been traditionally addressed within a different line of works by *Private Information Retrieval* (PIR) proposals, which provide protocols for querying a database that prevent the storage server from inferring which data are being accessed. PIR approaches typically work on a different problem setting. As a matter of fact, in most proposals, the external database being accessed is in plaintext (i.e., content confidentiality is not an issue). Regardless of whether the external database is plaintext or encrypted, PIR solutions have high computational complexity

and are therefore not applicable to real systems. It has been proved that the execution of information-theoretic PIR protocols require more resources than those required for a complete transfer of the database from the server to the client.

In the work in WP2.3, we aimed at providing a novel efficient approach addressing the different aspects of the privacy problem [DFP⁺11]. We consider a reference scenario where a *data owner* outsources data to an external honest-but-curious server, and accesses her data by submitting requests to a *client* that directly interacts with the server. Our goal is to enable the owner to efficiently access the outsourced data while guaranteeing content, access, and pattern confidentiality from any observer, including the server itself.

We proposed a novel data structure, called *shuffle index*, with which the data to be outsourced are organized (Section 4.1). Our shuffle index assumes an unchained $B+$ -tree organization of data and applies node-level encryption to hide actual data from the external storage. In the working of the system, the client can hide the actual request within cover (fake) requests, cache nodes, and shuffle the content among blocks stored at the server. In this way, no observer, including the server itself, can reconstruct the association between blocks read and actual accessed data (Section 4.2). Our solution combines cover, caching, and shuffling techniques in an effective way to provide confidentiality

while maintaining a limited performance overhead (Section 4.3). Our approach is the only solution known to us that delivers content, access, and pattern confidentiality at the same time, offering a performance profile adequate for real applications.

4.1 Shuffle index data structure

For outsourcing, we assume data to be indexed over a candidate key K defined for the data collection and organized as an *unchained $B+$ -tree*, with data stored in the leaves in association with their index values, and where there are no links from a leaf to the next, representing a chain. Accesses to the data (searches) are based on the value of the index. The reason for not representing the links between the leaves is that following such links, when accessing data, would leak to the server (to which the content of the nodes is not known) *i*) the fact that the query being executed is a range query, and *ii*) the order relationship among index values in different nodes.¹ Our data structure is therefore characterized by a fan out \mathcal{F} , meaning that each node (except the root) has $q \geq \lceil F/2 \rceil$ children and stores $q - 1$ values v_1, \dots, v_{q-1} , ordered from the smallest to the greatest. The i -th child of any internal node in the unchained $B+$ -tree is the root of a subtree containing the values v with: $v < v_1$; $v_{i-1} \leq v < v_i$, $i = 2, \dots, q - 2$; $v \geq v_{q-1}$. Figure 14(a) illustrates a graphical representation of our data structure. Pointers between nodes of the abstract data structure correspond, at the logical level, to *node identifiers*, which can then be easily translated at the physical level into physical addresses. At the logical level, our data structure can be seen as a set of nodes, where each node is a pair $\langle id, n \rangle$, with id the node identifier and n the node content. Note that the possible

¹Range queries are supported with only the additional cost of accessing the next leaf, starting the access from the root. With a collection of shuffle indexes, the overhead due to restarting the access from the root, rather than going directly to the next leaf, causes an increase of only a few percentage points in the overall access times.

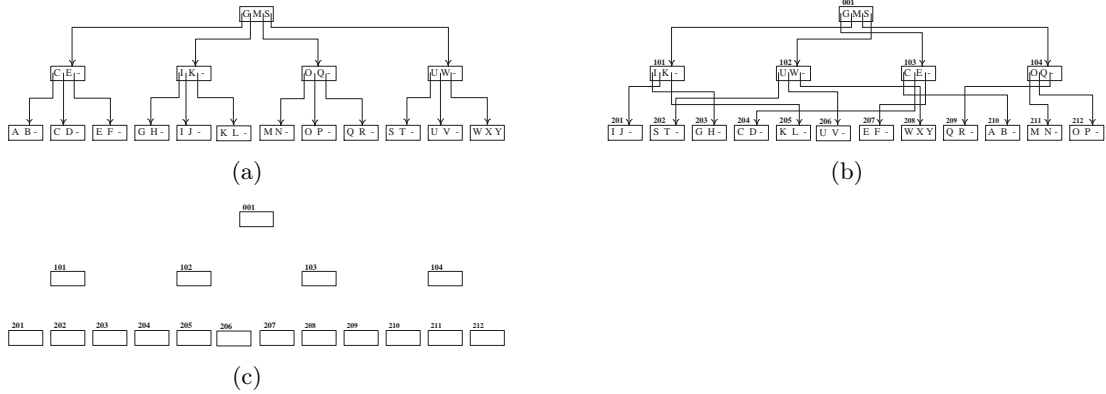


Figure 14: An example of abstract (a) and logical (b) representation of a data structure to be outsourced, and of the corresponding view of the server (c)

order between identifiers does not necessarily correspond to the order in which nodes appear in the value-ordered abstract representation. Figure 14(b) illustrates a possible representation of the data structure in Figure 14(a), where nodes appear ordered (left to right) according to their identifiers, which are reported on the top of each node. For simplicity and easy reference, in our example, the first digit of the node identifier denotes the level of the node in the tree. The reason why we distinguish between node identifier and node content is that, as we will see later on, our approach is based on shuffling content among nodes. In other words, a given content may be associated with different identifiers at different times. In the following, when clear from the context, we will use the term *node* to refer to either the content of a node or to the content together with the identifier.

As typical in emerging outsourcing solutions, we use *encryption* to preserve *content* confidentiality. We assume encryption to be applied at the node level (i.e., each node is individually encrypted). To destroy plaintext distinguishability, the encryption function adopts a random salt. Also, the encrypted node is concatenated with the result of a MAC function applied on the encrypted node and its identifier. In this way, the client can assess the authenticity of the node returned by the server. Note that, since nodes contain pointers to children, the ability to establish authenticity of a node (starting from the root) implies the ability to establish authenticity, and therefore integrity, of the whole data structure.

In the realization of physical accesses, for efficiency reasons, the size of the node to be stored (i.e., its encrypted version together with the result of the MAC function) should be a multiple of the size of the disk block. For simplicity, we assume the size of each encrypted node to be equal to the size of one disk block of the server, and the identifier of the block to be the same as the identifier of the node. We refer to an encrypted node as a *block*. Blocks are formally defined as follows.

Definition 4.1.1 (Block). *Let $\langle id, n \rangle$ be a node of an unchained B^+ -tree. The encrypted version of $\langle id, n \rangle$, called block, is a pair $\langle id, b \rangle$, with $b = \mathcal{C} || \mathcal{T}$, $\mathcal{C} = E_k(\text{salt} || n)$,*

$\mathcal{T} = \text{MAC}_k(\text{id}||\mathcal{C})$, with E a symmetric encryption function, k the encryption key, salt a value chosen at random during each encryption, and MAC a strongly un-forgable keyed cryptographic hash function.

We refer to the encrypted version of the logical data structure outsourced to the server and on which accesses are executed as *shuffle index*. The reason for the term *shuffle* is due to the way the structure is dynamically modified at each access, shuffling content among blocks (see Section 4.2). Our shuffle index is defined as follows.

Definition 4.1.2 (Shuffle index). *Let $\{\langle \text{id}_0, n_0 \rangle, \dots, \langle \text{id}_m, n_m \rangle\}$ be a set of nodes of an unchained B+-tree. The shuffle index is the set $\{\langle \text{id}_0, b_0 \rangle, \dots, \langle \text{id}_m, b_m \rangle\}$ of corresponding blocks (Definition 4.1.1).*

According to the definition of shuffle index, the server just sees a collection of blocks, each with a given identifier but whose content is encrypted. Access to the data requires an iterative process between the client and the server. The client performs an iteration for each level of the shuffle index starting from the root. At each iteration it determines the node to be read (i.e., the block to be retrieved from the server) at the next level. The process ends when a leaf block is retrieved, which is the block that contains the index value searched (or where it would have appeared, if the index value does not belong to the database).

4.2 Protection techniques

We first describe the different aspects of confidentiality we want to guarantee against unauthorized observers. We then illustrate our protection techniques complementing encryption for ensuring confidentiality.

4.2.1 Problem statement

Our goal is to protect the confidentiality of the outsourced data against any possible observer. Since, among all possible observers, the server is the party which has the highest potential for observations (all accesses are executed by it), without loss of generality in the following we assume the server as our observer.

The server receives from the data owner a set of blocks to store and receives requests to access such blocks with the iterative process described in Section 4.1. The server has therefore knowledge of the number m of blocks (nodes) and their identifiers, and the height h of the shuffle index (because the iterative process requires the retrieval of a block for each level of the shuffle index). Also, by observing a long enough history of accesses, the server can easily establish the level associated with each block. Note instead that the topology of the shuffle index (i.e., the pointers between parent and children) is not known to the server. Figure 14(c) illustrates the view of the server on the shuffle index in Figure 14(b).

Before defining the confidentiality we want to guarantee, we note that the server can only monitor accesses at the granularity of a block (node). The basic protection granted by encryption already ensures uncertainty on the actual index value (and therefore on the specific data) requested by an access, since any of the index values stored in the returned

node could potentially be the target. Such a basic protection cannot be considered sufficient, also because index values stored in the same node will all be close within a given range. Given this observation, in the following, we consider confidentiality breaches at the granularity of nodes.

In the working of the system, every access request translates into an *observation* o_i of the server corresponding to a sequence of blocks $\{b_{i1}, \dots, b_{ih}\}$ accessed. At any point in time, given a sequence of observations o_1, \dots, o_z corresponding to all the accesses performed, the server should not be able to infer: *i*) the data stored in the shuffle index (*content confidentiality*); *ii*) the data to which access requests are aimed, that is, $\forall i = 1, \dots, z$, the server should not infer that o_i aims at a specific node (*access confidentiality*); and *iii*) that o_i aims at accessing the same node as o_j , $\forall i, j = 1, \dots, z, i \neq j$ (*pattern confidentiality*). Intuitively content confidentiality refers to the data stored in the leaves of the unchained $B+$ -tree, access confidentiality to the data targeted by a request, and pattern confidentiality to the relationship between the data targeted by different requests. It is easy to see that encryption provides content confidentiality for data at rest and access confidentiality of individual requests. It is however not sufficient for providing pattern confidentiality of a set of observations. To illustrate, suppose that a shuffle index never changes. By observing that two accesses retrieve the same blocks, an observer could easily determine that the accesses refer to the same node, thus breaching pattern confidentiality. An observer can then exploit the possible information on the frequencies with which different values can be accessed and a set of observations to reconstruct the correspondence between plaintext values and blocks and infer (or restrict its uncertainty on) the specific node to which a specific access refers, thus breaching access confidentiality.

Since the information that the server can exploit in the working of the system is the comparison between the frequencies with which blocks are accessed and the frequencies of accesses to different values, the key aspect for guaranteeing all forms of confidentiality above is to destroy such a correspondence. Our approach to protect confidentiality is based on the combination of three basic strategies: *1) cover searches*, *2) cached searches*, and *3) shuffling*.

4.2.2 Cover searches

As noted above, the execution of an access over the shuffle index can trivially leak information on the fact that two accesses aim, or do not aim, at the same node. Also, combined with the possible knowledge of the server on frequencies of accesses to node contents, it can help the server to establish the correspondence between node contents and blocks where they are stored (frequently accessed data will correspond to frequently accessed blocks). For instance, consider the logical representation of a shuffle index in Figure 14(b), and two consecutive requests for index value ‘F’ translating into accesses to blocks $\{(001); (103); (207)\}$ and $\{(001); (103); (207)\}$, respectively. By observing these sequences of accessed blocks, the server can infer that the two requests refer to the same data (i.e., the content of block 207). Our first protection technique aims at introducing confusion on the target of an access request by hiding it within a group of other requests that work as covers.

Cover searches are fake searches the client executes in conjunction with the actual *target* search of the index value it aims to access. The number of cover searches is a protection parameter of our approach.

Since, as noted in Section 4.2.1, the granularity of protection is the block (node), cover searches must provide block diversity, which means they must translate into accesses to different blocks at each level of the shuffle index, but the root. As a matter of fact, covers translating to the same block would not provide any additional protection than that offered by encryption. For instance, ‘E’ cannot be chosen as a cover for ‘F’ as both would translate into accesses to block 207, thus disclosing that the access requests refer to the content of block 207. Given a shuffle index built over a candidate key with domain \mathcal{D} and a value $v \in \mathcal{D}$, $path(v)$ denotes the set of blocks in the unique path of the shuffle index that starts at the root and ends in the leaf block where v is possibly stored, if v is in the database. Cover searches are formally defined as follows.

Definition 4.2.1 (Cover searches). *Let $\{\langle id_0, b_0 \rangle, \dots, \langle id_m, b_m \rangle\}$ be a set of blocks forming a shuffle index built over a candidate key with domain \mathcal{D} , and let v_0 be a value in \mathcal{D} . A set $\{v_1, \dots, v_n\}$ of values in \mathcal{D} is a set of cover searches for v_0 if $\forall v_i, v_j \in \{v_0, v_1, \dots, v_n\} : v_i \neq v_j \implies path(v_i) \cap path(v_j) = \langle id_0, b_0 \rangle$, that is, contains only the root of the shuffle index.*

Basically, assuming num_cover searches are adopted in the execution of an access, instead of asking the server to retrieve, for each level in the shuffle index, the block in the path from the root to the target, the client asks the server to retrieve $num_cover + 1$ blocks: one corresponds to the block on the path to the target, and each of the others corresponds to the block on the path to one cover.

Intuitively, cover searches hide the actual search within a set of searches, since any of the $num_cover + 1$ leaf blocks could equivalently contain the actual target. This requires cover searches to be indistinguishable from actual searches. We guarantee this cover/target indistinguishability property by ensuring that the frequency distribution with which values in the candidate key domain \mathcal{D} are used as cover searches is the same as the frequency distribution with which values are searched upon client’s request. For instance, consider again the two searches above for index value ‘F’ (block 207), and assume the first uses cover ‘I’ while the second one uses cover ‘M’. The sequences of accesses to blocks observed by the server would now be $\{(001); (101,103); (201,207)\}$ and $\{(001); (103,104); (207,211)\}$, respectively. While without cover the server was able to detect that the two requests aimed at the same block (node), with one cover the server can assess this only with probability $0.5 \cdot 0.5 = 0.25$.

The fact that searches are all executed in parallel (i.e., all the $num_cover + 1$ blocks at each level of the shuffle index are retrieved before proceeding at the next level), confuses the parent-child relationship of the different blocks. In fact, at each level any of the $num_cover + 1$ parents could be associated with any of the $num_cover + 1$ children, producing therefore $(num_cover + 1)^h$ potential paths. For instance, with reference to the example above, 201 could be child of either 101 or 103. Of course, parent-child information (like actual targets) can be disclosed by intersection attacks, observing the same set of blocks in different accesses (103 and 207 in the example above). Intersection attacks are counteracted by caching and shuffling, as explained in the remainder of this section.

4.2.3 Cached searches

Our second protection technique aims at counteracting *intersection* attacks in the short term and consists in maintaining at the trusted client side a local copy, called *cache*, of nodes in the path to the target. Being client side, we maintain the cache in plaintext (i.e., the cache stores plaintext nodes and not their encrypted version).

Definition 4.2.2 (Cache). *Let $\{\langle id_0, n_0 \rangle, \dots, \langle id_m, n_m \rangle\}$ be a set of nodes forming an unchained $B+$ -tree of height h . A cache \mathcal{C} of size num_cache for the unchained $B+$ -tree is a layered structure of $h+1$ sets $Cache_0, \dots, Cache_h$, where:*

- *Cache₀ contains the root node $\langle id_0, n_0 \rangle$;*
- *Cache_l, $l = 1, \dots, h$, contains num_cache nodes belonging to the l -th level of the unchained $B+$ -tree;*
- *$\forall n \in Cache_l, l = 1, \dots, h$, the parent of n in the unchained $B+$ -tree belongs to $Cache_{l-1}$ (path continuity property).*

Path continuity guarantees that the parent of any node in the cache belongs to the cache. As a consequence, the path connecting the root of the unchained $B+$ -tree to every node in the cache completely belongs to the cache itself. We assume the cache to be properly initialized by the data owner at the time of outsourcing, by locally storing nodes in num_cache disjoint paths (i.e., with only the root in common) of the unchained $B+$ -tree.

In the working of the system, the cache will be updated and will keep track only of actual (and not of cover) searches, since it is intended to work as an actual cache. We assume the cache at each level to be managed according to the LRU policy: when a new node is added to $Cache_l$, the node least recently used is pushed out from $Cache_l$. The application of the LRU policy guarantees the satisfaction of the path continuity property.

The cache helps in counteracting short term intersection attacks since it avoids the client to search for a repeated target of two close access requests. For instance, with reference to the two consecutive requests for index value ‘F’ in Section 4.2.2, the second request would find ‘F’ in cache. Since the number of blocks requested to the server has always to be the same (i.e., $num_cover + 1$), the client would generate, for the second request, two cover searches (e.g., ‘M’ and ‘W’). Consequently, the observations of the server on the two requests would be $\{(001); (101,103); (201,207)\}$ and $\{(001); (102,104); (208,211)\}$, respectively. The server would not be able to determine whether the two requests aim at the same target. The reader may wonder why we perform $num_cover+1$ fake cover searches when the target node is already in cache. First, if the observer knows that an access was to be executed, not performing it would leak information on the fact that the target node is in the cache. Second, the protection given by the cache does not work only as an independent technique, but plays a role together with the other protection techniques.

4.2.4 Shuffling

Caching does not prevent intersection attacks on observations that go beyond the size of the cache. As an example, suppose that no cache is used (i.e., $num_cache=0$), and

with reference to Figure 14(b) consider three consecutive requests all for index value ‘F’, using one cover search for each request (e.g., ‘I’, ‘M’, and ‘W’, respectively). These access requests will translate into the following sequences of accesses to blocks $\{(001); (101,103); (201,207)\}$, $\{(001); (103,104); (207,211)\}$, and $\{(001); (102,103); (207,208)\}$, respectively. Assuming the indistinguishability of targets and covers, by the observation of these sequences of accesses the server can infer with probability $0.5 \cdot 0.5 \cdot 0.5 = 0.125$ that the three access requests refer to the same data (i.e., the content of block 207). Also, accesses leak to the server the parent-child relationship between blocks. While the information on the parent-child relationship by itself might seem to not compromise confidentiality, it can easily open the door to privacy breaches and should then remain confidential. Given a long enough history of observations, the server will be able to reconstruct the topology of the shuffle index and therefore gain knowledge on the similarity between values stored in the blocks.

Our third protection technique starts from the observation that inferences such as the one mentioned above are possible to the server by exploiting the one-to-one correspondence between a block and the node stored in it: accesses to the same block trivially correspond to accesses to the same node. *Node shuffling* breaks this one-to-one correspondence by exchanging the content among nodes (and therefore blocks). Since a block depends on the content of the corresponding node and on the node identifier (Definition 4.1.1), shuffling clearly requires the re-computation of the blocks associated with shuffled nodes and then requires node decryption and re-encryption. Note how the re-encryption of a node, applied to the node content concatenated with a different random salt, produces a different encrypted text (block). This aspect is particularly important since encrypted text corresponding to a given node automatically changes at each access, making it impossible to track the shuffling executed and to determine if the node content stored in a block has been changed or has remained the same. Node shuffling is formally defined as follows.

Definition 4.2.3 (Shuffling). *Let $\mathcal{N} = \{\langle id_1, n_1 \rangle, \dots, \langle id_m, n_m \rangle\}$ be a set of nodes at the same level of an unchained $B+$ -tree and π be a permutation of id_1, \dots, id_m . The node shuffling of \mathcal{N} with respect to π is the set $\{\langle id_1, n'_1 \rangle, \dots, \langle id_m, n'_m \rangle\}$ of nodes, where $id_i = \pi(id_j)$ and $n'_i = n_j$, with $i, j = 1, \dots, m$.*

Intuitively, our approach exploits shuffling by exchanging the contents of all blocks read in the execution of an access and the nodes in cache (so that their contents are shuffled), and rewriting all of them back on the server. In this way, the correspondence existing between block identifiers and the content of the nodes they store is destroyed. For instance, assume that shuffling is used and that the server observes the following sequence of accesses to blocks $\{(001); (101,103); (201,207)\}$; $\{(001); (103,104); (207,211)\}$; and $\{(001); (102,103); (207,208)\}$. The server can only note that the three sequences have a leaf block in common (i.e., 207). The three requests aim at accessing the same node only if: the second and third requests are for the content of block 207 (the probability is $0.5 \cdot 0.5 = 0.25$); the data target of the first request coincides with the content of block 207 after the first shuffling operation (the probability is 0.5); and the content of block 207 is not moved by the second shuffling operation (the probability is 0.5). As a consequence, 0.0625 is the probability that the three requests aim at the same node.

Note that shuffling among nodes at a given level requires to update the parents of

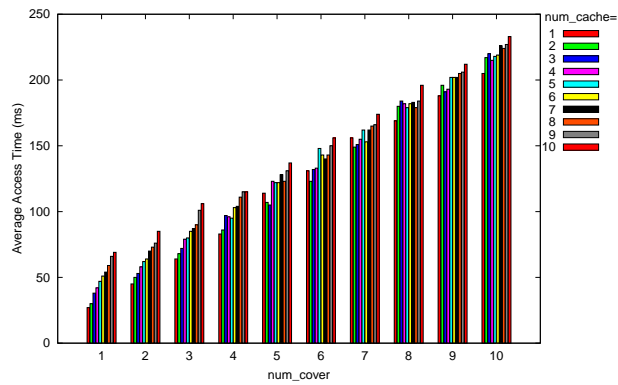
the nodes so that the pointers in them properly reflect the shuffling. For instance, consider Figure 14(b) and assume nodes (103,104) are shuffled so that $\pi(103)=104$ and $\pi(104)=103$, (i.e., their contents are swapped). As a consequence, root node $[_{103}G_{101}M_{104}S_{102}]$ must be updated to be $[_{104}G_{101}M_{103}S_{102}]$.

4.3 Performance analysis

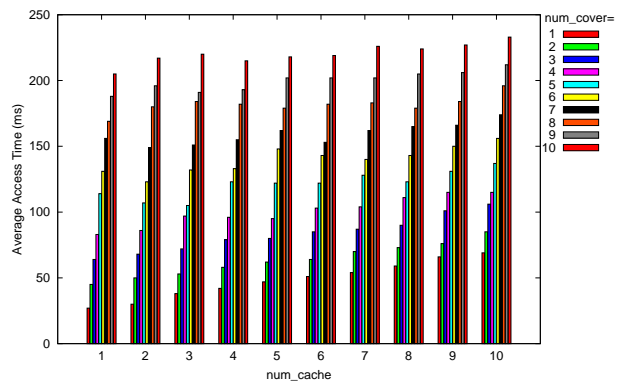
We implemented the shuffle index with a Java program. To assess its performance, we used a data set of 1 TB stored in the leaves of a shuffle index with 4 levels, built on a numerical candidate key K of fixed-length, fan out 512, and representing 2^{32} (over 4 billion) different index values. The size of the nodes of the shuffle index was 8 KB. The hardware used in the experiments included a server machine with 2 Intel Xeon Quad 2.0 GHz L3-4 MB, 12 GB RAM, four 1 TB disks, 7200 RPM, 32 MB cache, and Linux operating system with the ext4 file system. The client machine was running an Intel Core 2 Duo CPU at 2.4 GHz, with 4 GB RAM. The index was stored on all 4 drives of the server. The performance analysis started after the system had processed a significant number of accesses, to be in a steady state. To evaluate the performance of the shuffle index we took into consideration the cost of: CPU, disk, and network.

CPU. The computational load required for the management of the shuffle index is quite limited. The algorithm uses only symmetric encryption and a MAC; the execution times we measured on an 8 KB block for both cryptographic functions are under 100 μ s, a negligible fraction of the time required by network and disk accesses. The performance of the shuffle index is then driven by disk and network performance.

Disk. We analyzed the performance of the shuffle index when client and server operate in a local area network (we used a 100 Mbps Ethernet network). In this configuration, disk performance becomes the limiting factor. These experiments then permit to identify the maximum rate of queries that a server can support. Figures 15(a) and 15(b) report observed times in milliseconds. The values are grouped by the same value of *num_cover* and for the same value of *num_cache*, both varying from 1 to 10. As expected, the access time grows linearly with the number of cover searches, since every additional cover requires the traversing of an additional path in the shuffle index. Although an increase in *num_cache* causes a growth in the number of blocks written for each level of the shuffle index, the number of cached nodes has a smaller impact on the access time. This is justified by the fact that the disk operations caused by the increase in *num_cache* greatly benefit from buffering and cache mechanisms at the operating system and disk controller level. We claim that, as it is typical for database index structures, the bottleneck in the performance of the shuffle index in a LAN is the number and profile (e.g., random, sustained/repeated) of read and write operations on the hard disks. The access times show that the system is able, when retrieving randomly chosen 8 KB blocks over a 1 TB collection, to manage up to 40 requests per second. The best performance is obtained when using a single cover and a single cache; increasing the number of covers there is an impact on performance, but in every tested configuration the access time was below 250 ms. We also note that no solution providing support for access and pattern confidentiality offers comparable performance.



(a)



(b)

Figure 15: Access time in a LAN as a function of the number of covers (a) and of the size of the cache (b)

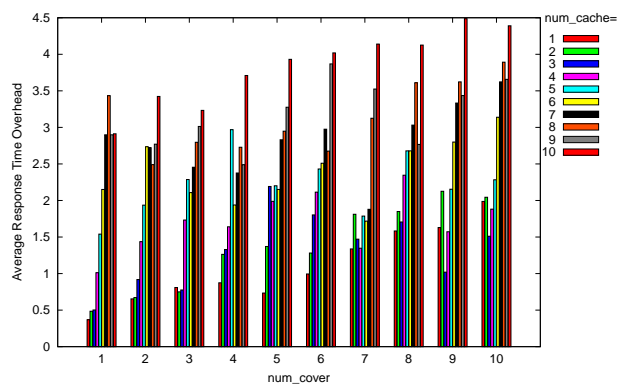


Figure 16: Overhead in a WAN compared to the use of a plain encrypted index as a function of the number of covers

Network. We analyzed the performance of the shuffle index when client and server operate in a wide area network. The server was in a University lab and the client was located in the same city, accessing the server using a 10 Mbps residential Internet connection. This scenario, where a client uses a remote untrusted party for the private access to data, is the most interesting and natural for the shuffle index. In this configuration, network performance becomes the limiting factor. Rather than focusing on absolute numbers that strongly depend on network configuration parameters that are not under control, we were especially interested in comparing the performance of the shuffle index and the performance offered by a plain encrypted index. A plain encrypted index has the same static structure of the shuffle index, but it does not use covers, caching, and shuffling to provide access and pattern confidentiality. The plain encrypted index still requires the client to visit the nodes in the tree level-by-level. Figure 16 reports the overhead compared to the use of the plain encrypted index. The reported measures were obtained by averaging over 100 experiments for each data point. We built a statistical model to analyze the results of experiments. From the model, we derive that each increase in the number of covers or cache searches adds respectively 30% and 10% of the plain encrypted access time. The difference between the impact of covers and caches is due to the different disk costs discussed above. Again, even in a WAN configuration, our solution enjoys considerably better performance with respect to approaches providing comparable protection. Also, we note that configurations with *num_cover*=1 and *num_cache* between 1 and 2 already provide a strong degree of access and pattern confidentiality with a performance overhead factor below 50% (the measured values were 170 ms for the plain encrypted index and less than 240 ms for the shuffle index). Hence, we believe our approach to be particularly appealing to many application scenarios, providing adequate access and pattern confidentiality at an affordable overhead.

4.4 Conclusions

We presented an indexing technique for data outsourcing that proves to be efficient while ensuring content, access, and pattern confidentiality. To our knowledge, this is the first work providing such a guarantee of protection while enjoying actual applicability. The shuffle index presents additional advantages. First, the underlying structure is that of $B+$ -trees, which are commonly used in relational DBMSs to support the efficient execution of queries. This similarity can facilitate the integration between shuffle indexes and traditional query processing. A second advantage is the possibility for the use of multiple indexes, defined on distinct search keys, over the same collection of data. Commercial DBMSs often support multiple $B+$ -trees over the same table, choosing one index as primary (the one with the tuples in the $B+$ -tree leaves) and putting into the leaves of the secondary indexes the key value with respect to the primary index. Every search over the secondary index first retrieves the primary key value and then executes a subsequent search on the primary index. The same approach can be applied to shuffle indexes, obtaining the immediate support of multiple privacy-compliant access paths.

Chapter 5

Access control for the protection of user-generated data (WP2.4)

Current access control models typically assume that resources are under the strict custody of a trusted party, which monitors each access request to verify if it is compliant with the access control policy. There are many scenarios where this approach is becoming no longer adequate. Many trends in Web technology are creating a need for owners of sensitive information to manage access to it by legitimate users using the services of *honest but curious* third parties. In this scenario, the data owner encrypts the data before outsourcing and stores them at the server. Possible access authorizations are to be enforced by the owner. This chapter addresses the problem of enforcing selective access on outsourced data without need of involving the owner in the access control process [DFJ⁺10b]. The solution puts forward a novel approach that enforces access control via *selective encryption*. The chapter also describes a two-layer encryption approach that allows the data owner to outsource, besides the data, the complete management of the authorization policy itself, thus providing efficiency and scalability in dealing with policy updates.

5.1 Introduction

Contrary to the vision of a few years ago, where many predicted that Internet users would have in a short time exploited the availability of pervasive high-bandwidth network connections to activate their own servers, users are today, with increasing frequency, resorting to service providers for disseminating and sharing resources they want to make available to others. This trend supports the view that service providers will be more and more requested to be responsible for the storage and the efficient and reliable distribution of content produced by others, realizing a “data outsourcing” architecture on a wide scale. The situation is particularly clear when we look at the success of services like YouTube, Flickr, Blogger, MySpace. These services typically assume that the server has complete access to the stored resources and therefore have limited use for all those

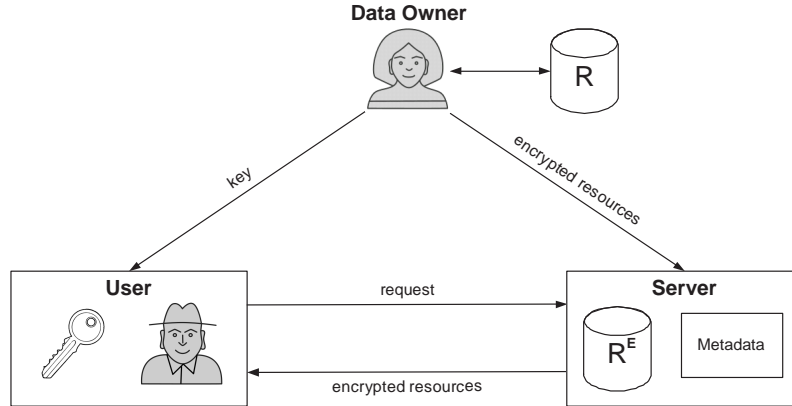


Figure 17: Outsourcing scenario

scenarios where the server cannot be granted such an access. In many applications, in fact, the server is considered *honest but curious*, meaning the server is relied upon for the availability of outsourced data but it is not authorized to see the actual data content. The most convincing and emerging solutions for these scenarios assume that the *data owner* encrypts data before sending them to the *server* for storage and gives the corresponding key to *users* authorized to access the data (see Figure 17). In this way, the confidentiality of information does not rely on an implicit assumption of trust on the server or on the legal protection offered by specific service contracts, but instead relies on the technical guarantees provided by encryption techniques. Typically, these solutions [CDD⁺05, HIM02, HIML02] focus on the problem of executing queries directly on the encrypted data by exploiting associated metadata and do not explicitly address the problem of supporting different keys or different access privileges (authorizations) for different users.

This chapter presents an approach that allows to enforce selective access to encrypted outsourced data. The basic idea is to integrate access control and encryption, thus encrypting the data to be outsourced with different keys depending on the authorizations to be enforced on the data. Although it is usually advisable to leave authorization-based access control and cryptographic protections separate, as encryption is traditionally considered a mechanism and should not be adopted in model definition [SD01], such a combination proves successful and powerful in the data outsourcing scenario.

The goal of the solution illustrated in this chapter is to translate an authorization policy to be enforced in an equivalent encryption policy regulating which data are encrypted with which key and regulating key release to users. The approach is based on the requirements of releasing at most one key to each user, and encrypting each resource at most once. These desiderata are achieved by exploiting a hierarchical organization of keys allowing the derivation of keys from other keys and public tokens [AFB05]. The goal is then to minimize the number of tokens to be generated and maintained. The problem of enforcing updates to the authorization policy, while limiting the cost in terms of bandwidth and computational power, is also addressed (providing a two layer approach that avoids the need for the owner to download the affected resources, decrypt and re-encrypt

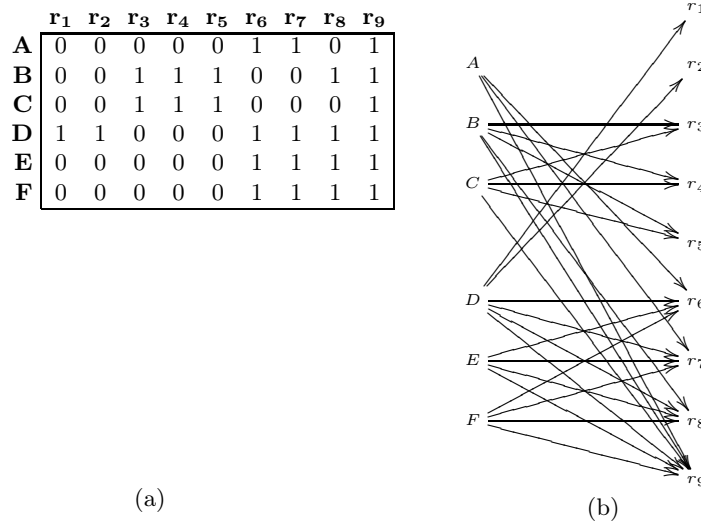


Figure 18: An example of access matrix (a) and corresponding authorization policy graph (b)

them, and reload their new versions). The solution proposed is particularly appealing as it allows delegating to the server the complete management, not only the enforcement, of the authorization policy.

5.2 Authorization and encryption policies

The data owner defines a discretionary authorization policy to regulate access to the outsourced resources, where a resource could be a file, a relational table, or even a tuple within a relation. Access by users to the outsourced resources is assumed to be read-only, while write operations are to be performed at the owner's site (typically by the owner itself). Given a set \mathcal{U} of users and a set \mathcal{R} of resources, an *authorization policy* over \mathcal{U} and \mathcal{R} is a triple $\mathcal{A} = \langle \mathcal{U}, \mathcal{R}, \mathcal{P} \rangle$, where \mathcal{P} is a set of permissions of the form $\langle u, r \rangle$, with $u \in \mathcal{U}$ and $r \in \mathcal{R}$, stating the accesses to be allowed. The set of permissions can be represented through an access matrix $\mathcal{M}_{\mathcal{A}}$, with a row for each user $u \in \mathcal{U}$ and a column for each resource $r \in \mathcal{R}$ [SD01]. Each entry $\mathcal{M}_{\mathcal{A}}[u, r]$ is set to 1 if u can access r ; it is set to 0 otherwise. Given an access matrix $\mathcal{M}_{\mathcal{A}}$ over sets \mathcal{U} and \mathcal{R} , $acl(r)$ denotes the *access control list* of r (i.e., the set of users that can access r).

The authorization policy is modeled as a directed and bipartite graph $\mathcal{G}_{\mathcal{A}} = \langle V_{\mathcal{A}}, E_{\mathcal{A}} \rangle$, having a vertex for each user $u \in \mathcal{U}$ and for each resource $r \in \mathcal{R}$, and an edge from u to r for each permission $\langle u, r \rangle \in \mathcal{P}$ to be enforced. In the following, the reachability of vertices in graph $\mathcal{G}_{\mathcal{A}}$ will be denoted by $\xrightarrow{\mathcal{A}}$. Figure 18 illustrates an example of authorization policy with 6 users, 9 resources, and 26 permissions, reporting the access matrix and the corresponding authorization policy graph.

The goal of the proposed modeling is to represent the authorization policy by means

of proper resource encryption and key distribution. For efficiency reasons, resources are protected through symmetric encryption. A naive solution to our goal would consist in encrypting each resource with a different key and assigning to each user the set of keys used to encrypt the resources she can access. Such a solution is clearly unacceptable, since it would require each user to manage as many keys as the number of resources she is authorized to access.

A *key derivation method* will be used to avoid users having to store and manage a huge number of (secret) keys. Basically, a key derivation method allows the computation of a key starting from another key and some public information. Among all the key derivation methods (e.g., [AT83, AFB05, ADFM06, CMW06]), the proposal in [AFB05] minimizes the amount of re-encrypting and re-keying that must be done to enforce changes to the authorization policy. The method is based on the definition and computation of *public tokens*. Let \mathcal{K} be the set of symmetric encryption keys in the system. Given two keys k_i and k_j in \mathcal{K} , a token $t_{i,j}$ is defined as $t_{i,j} = k_j \oplus h(k_i, l_j)$, where l_j is a publicly available label associated with k_j , \oplus is the bitwise xor operator, and h is a deterministic cryptographic function. The existence of a public token $t_{i,j}$ allows a user knowing k_i to derive key k_j through token $t_{i,j}$ and public label l_j . Since keys need to remain secret, while tokens are public, the use of tokens greatly simplifies key management. Key derivation via tokens can be applied in chains: a *chain of tokens* is a sequence $t_{i,l} \dots t_{n,j}$ of tokens such that $t_{c,d}$ directly follows $t_{a,b}$ in the chain only if $b = c$.

Since the set \mathcal{T} of tokens defined in the system and the set \mathcal{L} of labels associated with the keys in \mathcal{K} are public information, they are stored on the remote server (just like the encrypted data), so any user can access them. The relationships between keys are modeled through tokens allowing derivation of one key from another, via a *key and token graph* $\mathcal{G}_{\mathcal{K},\mathcal{T}} = \langle V_{\mathcal{K},\mathcal{T}}, E_{\mathcal{K},\mathcal{T}} \rangle$. The graph has a vertex for each pair $\langle k, l \rangle$ of key k and corresponding label l . There is an edge from a vertex $\langle k_i, l_i \rangle$ to a vertex $\langle k_j, l_j \rangle$ if there exists a token $t_{i,j}$ allowing the derivation of k_j from k_i .

The definition of tokens permits to easily support the assumption that each user can be released only a single key and that each resource can be encrypted by using a single key. Note that these are not simplifying or limiting requirements, rather they are desiderata that the proposed solution should satisfy. A *key assignment and encryption schema* over $\mathcal{U}, \mathcal{R}, \mathcal{K}, \mathcal{L}$ is a function $\phi : \mathcal{U} \cup \mathcal{R} \mapsto \mathcal{L}$ that returns for each user $u \in \mathcal{U}$ the label $l \in \mathcal{L}$ associated with the (single) key k in \mathcal{K} released to the user and for each resource $r \in \mathcal{R}$ the label $l \in \mathcal{L}$ associated with the (single) key k in \mathcal{K} with which the resource is encrypted. Note that each key k is uniquely identified through the label l associated with it.

The *encryption policy* can therefore be defined as follows.

Definition 5.2.1 (Encryption policy). *Let \mathcal{U} and \mathcal{R} be the set of users and resources in the system, respectively. An encryption policy over \mathcal{U} and \mathcal{R} , denoted \mathcal{E} , is a 6-tuple $\langle \mathcal{U}, \mathcal{R}, \mathcal{K}, \mathcal{L}, \phi, \mathcal{T} \rangle$, where \mathcal{K} is the set of keys defined in the system, \mathcal{L} is the set of corresponding labels, ϕ is a key assignment and encryption schema, and \mathcal{T} is a set of tokens defined on \mathcal{K} and \mathcal{L} .*

The encryption policy can be conveniently represented via an *encryption policy graph* $\mathcal{G}_{\mathcal{E}} = \langle V_{\mathcal{E}}, E_{\mathcal{E}} \rangle$, by extending the key and token graph to include a vertex for each user

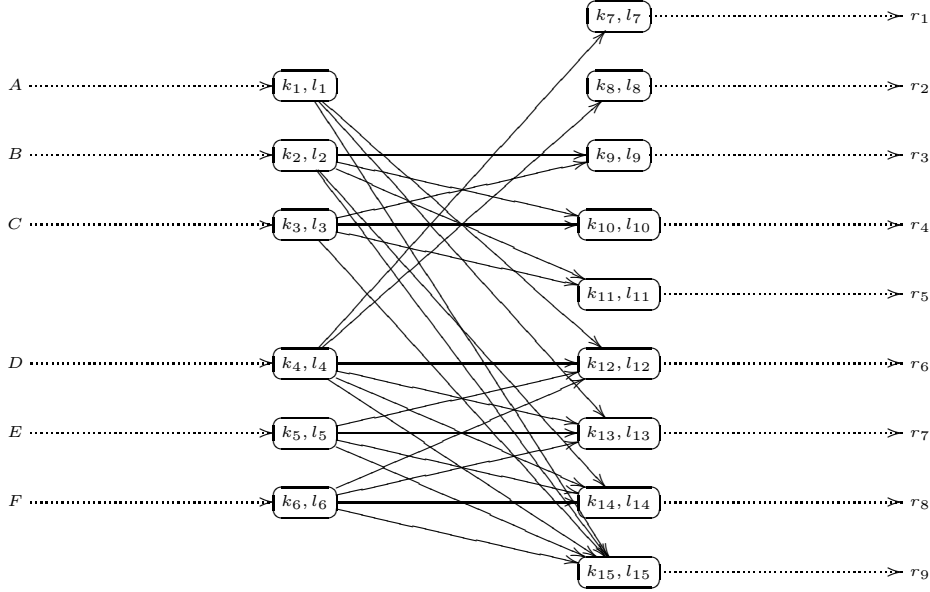


Figure 19: An example of encryption policy graph

and each resource, and adding an edge from each user vertex u to the vertex $\langle k, l \rangle$ such that $\phi(u)=l$ and from each vertex $\langle k, l \rangle$ to each resource vertex r such that $\phi(r)=l$.

Figure 19 illustrates an example of encryption policy graph, where dotted edges represent the key assignment and encryption schema (function ϕ) and solid edges represent the tokens. In the following, the reachability of vertices in graph $\mathcal{G}_{\mathcal{E}}$ will be denoted by $\xrightarrow{\mathcal{E}}$ (e.g., $A \xrightarrow{\mathcal{E}} r_6$). A user u can then retrieve (via her own key and the set of public tokens) all the keys of the vertices reachable from vertex whose label l is equal to $\phi(u)$. The resources accessible to a user according to an encryption policy are therefore all and only those reachable from u in the encryption policy graph $\mathcal{G}_{\mathcal{E}}$. The goal is then to translate an authorization policy \mathcal{A} into an equivalent encryption policy \mathcal{E} , meaning that \mathcal{A} and \mathcal{E} allow exactly the same accesses, as formally defined in the following.

Definition 5.2.2 (Policy equivalence). *Let $\mathcal{A} = \langle \mathcal{U}, \mathcal{R}, \mathcal{P} \rangle$ be an authorization policy and $\mathcal{E} = \langle \mathcal{U}, \mathcal{R}, \mathcal{K}, \mathcal{L}, \phi, \mathcal{T} \rangle$ be an encryption policy. \mathcal{A} and \mathcal{E} are equivalent, denoted $\mathcal{A} \equiv \mathcal{E}$, iff the following conditions hold:*

- $\forall u \in \mathcal{U}, r \in \mathcal{R} : u \xrightarrow{\mathcal{E}} r \implies u \xrightarrow{\mathcal{A}} r$
- $\forall u \in \mathcal{U}, r \in \mathcal{R} : u \xrightarrow{\mathcal{A}} r \implies u \xrightarrow{\mathcal{E}} r$

For instance, it is easy to see that the authorization policy in Figure 18 and the encryption policy represented by the encryption policy graph in Figure 19 are equivalent.

5.3 Minimum encryption policy

A straightforward approach for translating an authorization policy \mathcal{A} into an equivalent encryption policy \mathcal{E} consists in associating with each user a different key, encrypting

each resource with a different key, and producing and publishing a token $t_{u,r}$ for each permission $\langle u, r \rangle \in \mathcal{P}$. The encryption policy graph in Figure 19 corresponds to an encryption policy that has been generated by translating the authorization policy in Figure 18 with this approach. While simple, this translation generates as many keys as the number of users and resources and as many tokens as the number of permissions in the system. Even if tokens, being public, need not to be remembered or stored by users, producing and managing a token for each single permission can be unfeasible in practice. Indeed, each access to an encrypted resource requires a search across the catalog and therefore the total number of tokens is a critical factor for the efficiency of access to remotely stored data [DFJ⁺08].

This simple solution can be improved by grouping users with the same access privileges and by encrypting each resource with the key associated with the set of users that can access it. The advantage is that a key can be possibly used to encrypt more than one resource. Since there is a one-to-one mapping between an encryption policy \mathcal{E} and the encryption policy graph $\mathcal{G}_{\mathcal{E}}$ over \mathcal{E} , the hierarchy among sets of users induced by the partial order relationship based on set containment (\subseteq) can be exploited to define a proper encryption policy as follows. The encryption policy graph $\mathcal{G}_{\mathcal{E}} = \langle V_{\mathcal{E}}, E_{\mathcal{E}} \rangle$ is characterized by a set $V_{\mathcal{E}}$ of vertices, with $V_{\mathcal{E}} = V_{\mathcal{K},\mathcal{T}} \cup \mathcal{U} \cup \mathcal{R}$ and where $V_{\mathcal{K},\mathcal{T}}$ includes a vertex for each possible subset U of \mathcal{U} , and by a set $E_{\mathcal{E}}$ of edges including:

- an edge (v_i, v_j) for each possible pair of vertices $v_i, v_j \in V_{\mathcal{K},\mathcal{T}}$ such that the set U_i of users represented by v_i is a subset of the set U_j of users represented by v_j and the set containment relationship is direct;
- an edge (u_i, v_i) for each user $u_i \in \mathcal{U}$ such that $v_i \in V_{\mathcal{K},\mathcal{T}}$ and the set of users represented by v_i is $\{u_i\}$;
- an edge (v_j, r_j) for each resource $r_j \in \mathcal{R}$ such that $v_j \in V_{\mathcal{K},\mathcal{T}}$ and the set of users represented by v_j is $acl(r_j)$.

As an example, consider the portion of the authorization policy in Figure 18 that is defined on the set $\{A, B, C, D\}$ of users. Figure 20 illustrates the encryption policy graph over $\{A, B, C, D\}$ defined as described above. In the figure, each vertex v_i also reports, between square brackets, the set of users, denoted $v_i.acl$, represented by v_i .

By assigning to each vertex $v \in V_{\mathcal{K},\mathcal{T}}$ of the encryption policy graph a pair $\langle v.key, v.label \rangle$, corresponding to a key and label, the authorization policy can be enforced by: *i*) encrypting each resource with the key of the vertex corresponding to its access control list (e.g., resource r_5 should be encrypted with the key associated with the vertex representing $\{B, C\}$), and *ii*) assigning to each user the key associated with the vertex representing the user in the graph.

Although this solution is simple and easy to implement, it defines more keys than actually needed and requires the publication of a great amount of information on the remote server, thus causing an expensive key derivation process at the user-side. It is therefore important to find a *minimum encryption policy*, equivalent to a given authorization policy and minimizing the number of tokens to be maintained by the server. Unfortunately, the problem of minimizing the number of tokens in the encryption policy \mathcal{E} , while guaranteeing equivalence with the access control policy \mathcal{A} , is NP-hard (it can be reduced to the set cover problem [DFJ⁺10b]).

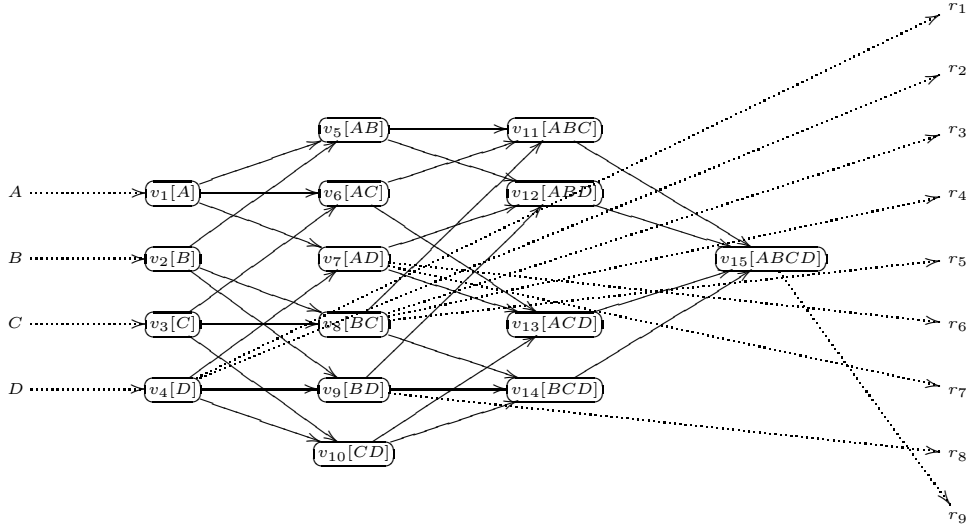


Figure 20: An example of encryption policy graph over $\{A, B, C, D\}$

It is therefore necessary to adopt a heuristic approach that reduces the user's overhead in deriving keys through a simplification of the encryption policy graph created according to the process previously described. The proposed heuristic approach is based on two basic observations. First, the encryption policy graph has to include only the vertices that are needed to enforce a given authorization policy, connecting them to ensure a correct key derivability. Second, beside the vertices needed for the enforcement of the authorization policy, other vertices can be included if they are useful for reducing the number of tokens in the public catalog. The proposed heuristic algorithm for computing a minimal encryption policy takes an authorization policy $\mathcal{A} = \langle \mathcal{U}, \mathcal{R}, \mathcal{P} \rangle$ as input and returns an encryption policy \mathcal{E} equivalent to \mathcal{A} . The algorithm is logically partitioned in the following four phases.

- *Initialization.* The algorithm identifies the vertices necessary to enforce \mathcal{A} , called *material* vertices. Material vertices represent: *i*) singleton sets of users, whose keys are communicated to the users and that allow them to derive the keys of the resources they are entitled to access; and *ii*) the *acls* of the resources, whose keys are used for encryption.
- *Covering.* According to the definition of policy equivalence (Definition 5.2.2), the material vertices must be connected in the graph in such a way that each user $u \in \mathcal{U}$ is able to derive keys allowing access to all and only the resources she is authorized to read. To this purpose, for each material vertex v corresponding to a non-singleton set of users, the algorithm finds a set of material vertices that form a *non-redundant set covering* for v , which become direct ancestors of v . A set V of vertices is a set covering for v if for each u in v , there is at least a vertex v_i in V such that u appears in v_i . It is non redundant if the removal of any vertex from V produces a set that does not cover v .

- *Factorization.* Whenever there is a set $\{v_1, \dots, v_m\}$ of vertices that have $n > 2$ common ancestors v'_1, \dots, v'_n , it is convenient to insert an intermediate vertex v representing all the users in v'_1, \dots, v'_n and to connect each v'_i , $i = 1, \dots, n$, with v , and v with each v_j , $j = 1, \dots, m$. In this way, the encryption policy includes $n + m$, instead of $n \cdot m$ tokens in the catalog.
- *Encryption policy generation.* The algorithm finally generates the encryption policy corresponding to the key and token graph generated by the previous phases. To this aim, it generates a key and a label for each vertex in the graph, it computes the key derivation token corresponding to each edge in the graph, and defines the key assignment and encryption schema ϕ .

Example 5.3.1. Consider the access control policy in Figure 18. During the initialization phase, the algorithm identifies the material vertices represented in Figure 21(a). Vertices v_1, \dots, v_6 represent the encryption keys communicated to users, and vertices v_7, \dots, v_{10} represent the encryption keys used to protect resources. Figure 21(b) illustrates the key and token graph resulting from the covering phase of the algorithm, which correctly enforces the access control policy in Figure 18. It is easy to see that this graph does not contain redundant edges. Figure 21(c) represents the key and token graph resulting from the factorization of vertices v_8 and v_9 that have three common direct ancestors (i.e., v_4 , v_5 , and v_6). To this purpose, the algorithm inserts non material vertex v_{11} , representing the set DEF of users, in the key and token graph. It then removes the 6 edges connecting v_4 , v_5 , and v_6 to v_9 and v_9 , and inserts 3 edges connecting v_4 , v_5 , and v_6 to v_{11} and 2 edges connecting v_{11} to v_8 and v_9 . Note that the graph in Figure 21(b) has 12 edges, while the graph in Figure 21(c) has 11 edges, thus saving one token.

5.4 Two-layer encryption for policy outsourcing

The model described in the previous sections assumes that keys and tokens are computed, on the basis of the existing authorization policy, prior to sending the encrypted resources to the server. When permissions are updated, the data owner interacts with the service provider for modifying the token catalog and for re-encrypting the resources involved in the update. Even if the computation and communication overheads caused by policy updates are limited, the data owner may not have the computational or bandwidth resource availability for managing policy changes. To further reduce the data owner's overhead, besides the resource storage, the authorization policy management is outsourced to the server, as described in the following.

5.4.1 Two-layer encryption

To delegate policy changes enforcement to the server, avoiding re-encryption for the data owner, a two-layer encryption approach can be adopted. The proposed mode is characterized by the following two encryption layers.

- **Base Encryption Layer (BEL)**, performed by the data owner before transmitting the resources to the server. It enforces encryption on the resources according to

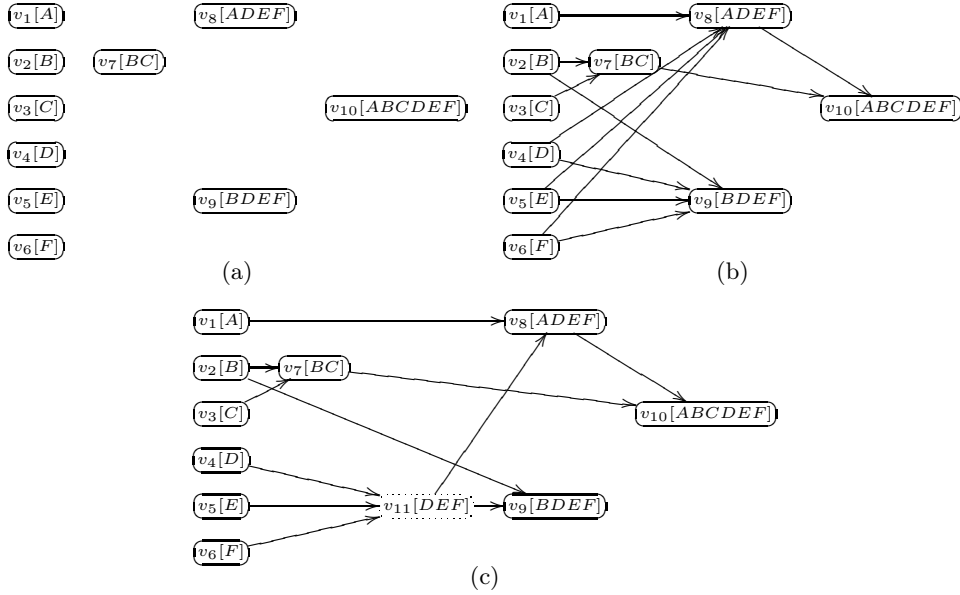


Figure 21: An example of initialization (a), covering (b), and factorization (c) generating an encryption policy equivalent to the authorization policy in Figure 18

the policy existing at initialization time. At this layer, there are two kinds of keys: *derivation keys* and *access keys*. Access keys are used to encrypt resources, while derivation keys are used to provide the derivation capability via tokens. Each derivation key k is always associated with an access key k_a , obtained by applying a secure hash function to k , that is, $k_a = h(k)$. The BEL is characterized by an encryption policy $\mathcal{E}_b = \langle \mathcal{U}, \mathcal{R}, \mathcal{K}_b, \mathcal{L}_b, \phi_b, \mathcal{T}_b \rangle$.

- **Surface Encryption Layer (SEL)**, performed by the server over the resources already encrypted by the data owner. It enforces the dynamic changes over the policy. The SEL is characterized by an encryption policy $\mathcal{E}_s = \langle \mathcal{U}, \mathcal{R}, \mathcal{K}_s, \mathcal{L}_s, \phi_s, \mathcal{T}_s \rangle$.

Each resource can then be encrypted twice: at the BEL first, and then at the SEL. Each user u receives two keys: one to access the BEL and the other to access the SEL, and will be able to access resources for which she knows both the keys (BEL and SEL) used for encryption.

In principle, the encryption policies at the BEL and at the SEL can be arbitrarily defined, as long as their combination is equivalent to the authorization policy. Given the encryption policy at the BEL, which is equivalent to \mathcal{A} , two approaches can be followed in the construction of the two layers: *i*) the SEL encryption policy is initialized to reflect exactly the BEL encryption policy (Full_SEL); or *ii*) the SEL policy is initialized to not carry out any over-encryption (Delta_SEL). Figure 22(a) illustrates an example of the BEL key and token graph and of the key assignment and encryption schema enforcing

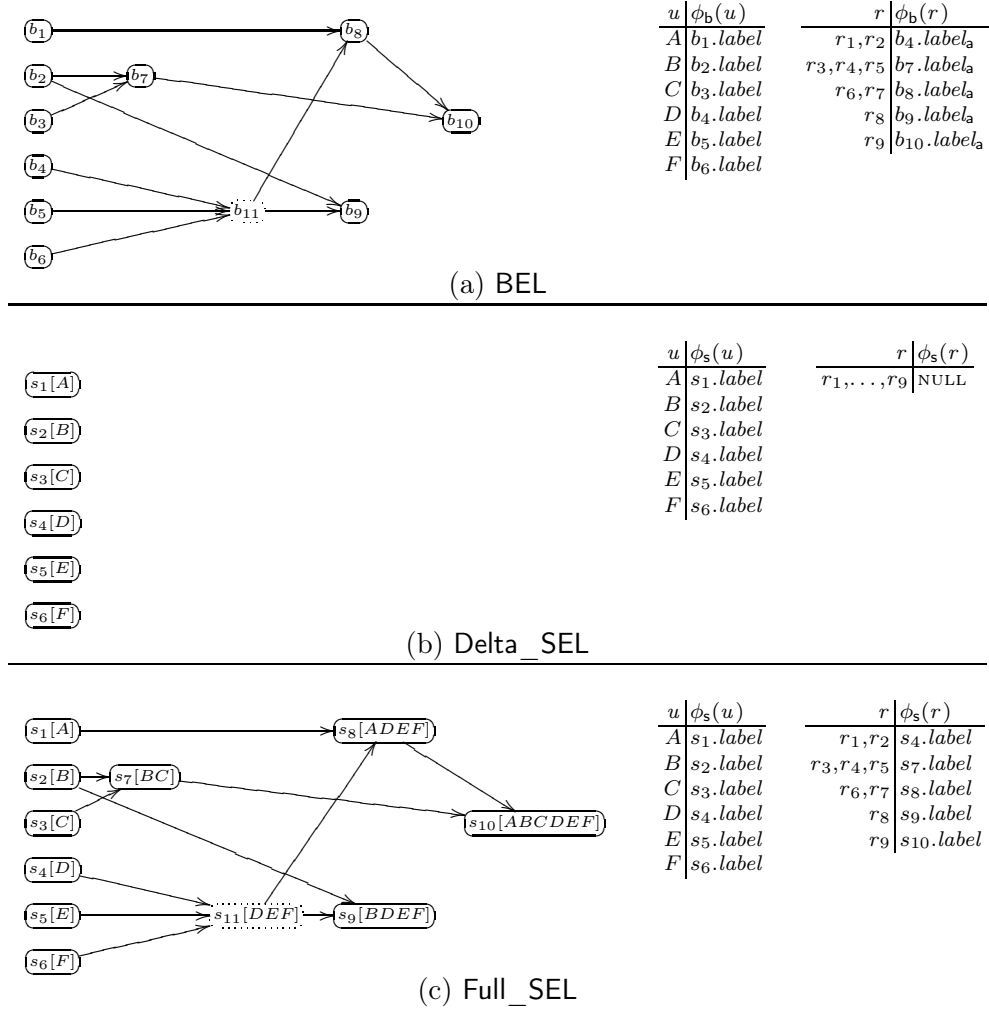


Figure 22: An example of BEL and SEL combination with the Delta_SEL and the Full_SEL approaches

the authorization policy in Figure 18. Figures 22(b) and (c) illustrate an example of the SEL key and token graph and of the key assignment and encryption schema, considering the Delta_SEL and Full_SEL approaches, respectively, that combined with the BEL encryption policy in Figure 22(a) correctly enforce the authorization policy in Figure 18. The reason for considering both the Full_SEL and Delta_SEL approaches is the different performance and protection guarantees that they enjoy. In particular, Full_SEL always requires double encryption to be enforced (even when permissions remain unvaried), thus doubling the decryption load of users for each access. By contrast, the Delta_SEL approach requires double encryption only when actually needed to enforce a change in the permissions. However, as described in the following, the Delta_SEL is characterized by greater information exposure than the Full_SEL approach. The choice between one or the other can then be a trade-off between costs and resilience to attacks.

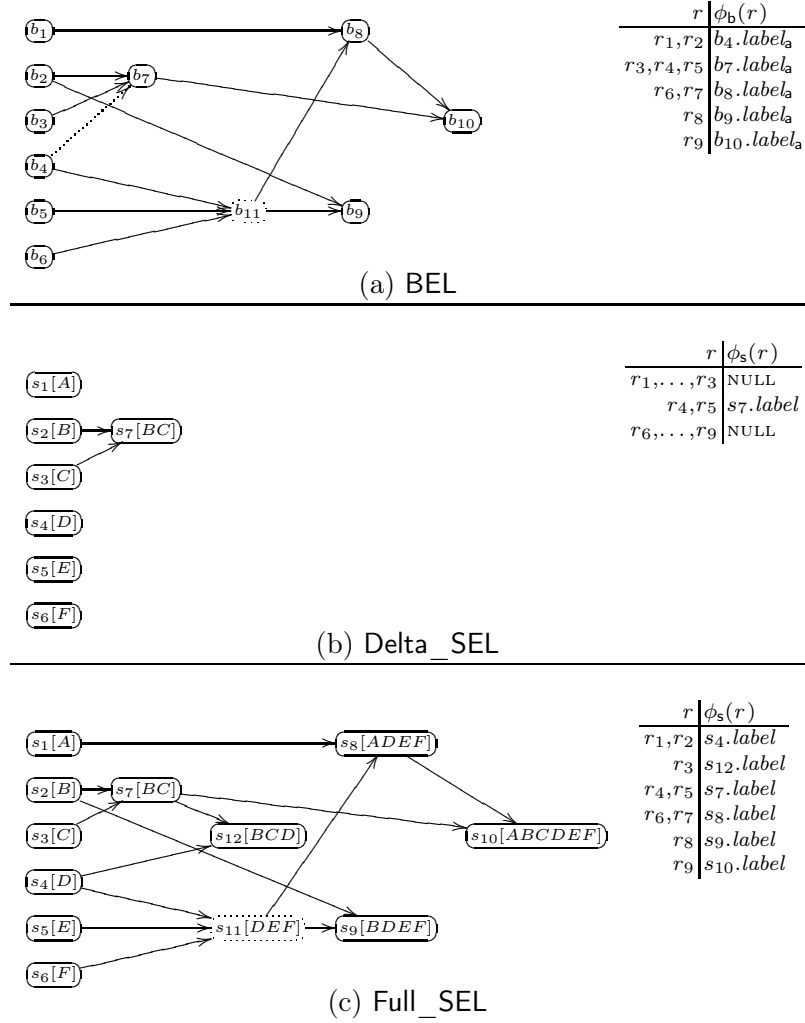


Figure 23: An example of grant operation

The two-layer approach enables the enforcement of policy updates without the need for the owner to re-encrypt and resend resources to the server. Indeed, the owner just adds (if necessary) some tokens at the BEL and delegates policy changes to the SEL by possibly requesting the server to over-encrypt some resources. The SEL (enacted by the server) receives *over-encryption* requests by the BEL (under the control of the data owner) and operates accordingly, adjusting tokens and possibly encrypting (and/or decrypting) resources.

Example 5.4.1. Consider the two layer encryption policy in Figure 22. Figure 23 illustrates the evolution of the corresponding key and token graphs and of $\phi_b(r)$ and $\phi_s(r)$ for resources in \mathcal{R} , when user D is granted access to resource r_3 . Note that $\phi_b(u)$ and $\phi_s(u)$ for users in \mathcal{U} never change upon grant/revoke operations.

Since access key $b_7.\text{key}_a$ used to encrypt r_3 cannot be derived from the derivation

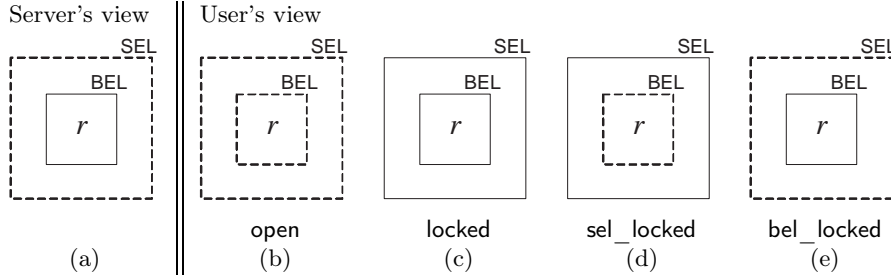


Figure 24: Possible views on resource r

key of vertex b_4 corresponding to $\phi_b(D)$, a token allowing computation of $b_7.key_a$ from $KeyDerb_4$ is added to BEL. $b_7.key_a$ is also used to encrypt resources r_4 and r_5 , which D is not authorized to view. As a consequence, these resources have to be over-encrypted so to make them accessible only to users B and C . In the `Delta_SEL` scenario, a new vertex s_7 is inserted in the key and token graph, with $s_7.acl=\{BC\}$, for resources r_4 and r_5 . The protection of resource r_3 at BEL level is instead sufficient and no over-encryption is needed. In the `Full_SEL` scenario, resources r_4 and r_5 are already correctly protected, r_3 is instead over-encrypted with the key of vertex s_{12} , which is created and inserted in the graph.

5.4.2 Protection evaluation

Although effective and efficient, the two-layer approach may be vulnerable to attacks from users who access and store all information offered by the server, or from *collusion* attacks, where different users (or a user and the server) combine their knowledge to access resources they would not otherwise be able to access. Note that for collusion to exist, both parties should gain in the exchange (as otherwise they will not have any incentive in colluding).

To model exposure, the different views that one can have on a resource r are graphically denoted with resource r in the center and with fences around r , denoting the barriers to the access imposed by the knowledge of the keys used for r 's encryption at the BEL (inner fence) and at the SEL (outer fence). The fence is continuous if there is no knowledge of the corresponding key (the barrier cannot be passed); it is discontinuous otherwise (the barrier can be passed). Figure 24 illustrates the different views that can exist on the resource. On the left-hand side of Figure 24(a), there is the view of the server itself, which knows the key at the SEL but does not have access to the key at the BEL. On the right-hand side of Figure 24(a), there are the different possible views of users, for whom the resource can be:

- **open**: the user knows the key at the BEL as well as the key at the SEL (Figure 24(b)). This view corresponds to the view of authorized users;
- **locked**: the user knows neither the key at the BEL nor the key at the SEL (Figure 24(c));

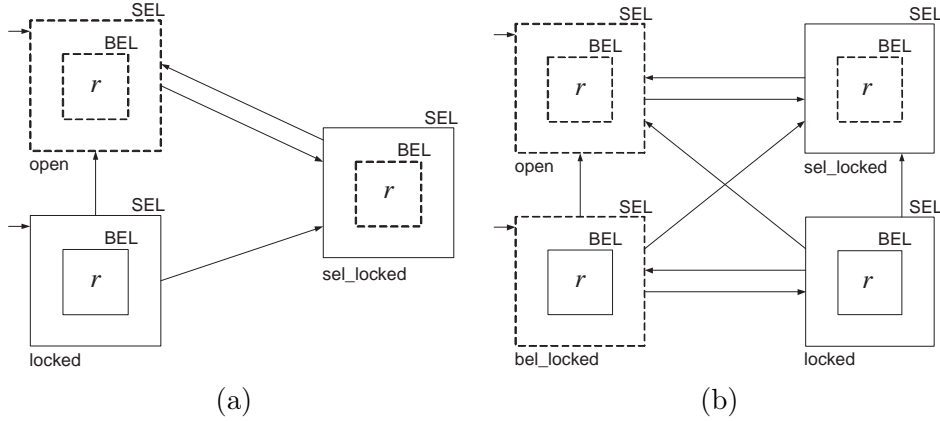


Figure 25: View transitions in the Full_SEL (a) and in the Delta_SEL (b)

- **sel_locked**: the user knows only the key at the BEL but does not know the key at the SEL (Figure 24(d));
- **bel_locked**: the user knows only the key at the SEL but does not know the key at the BEL (Figure 24(e)). Note that this latter view corresponds to the view of the server itself.

Figure 25(a) summarizes the possible view transitions in the Full_SEL approach. Since nobody (but the server) can have a **bel_locked** view, it is only necessary to consider users having the **sel_locked** view. Since users having the same views will not gain anything in colluding, the only possible collusion can happen between the server (who has a **bel_locked** view) and a user who has a **sel_locked** view. In this situation, the knowledge of the server allows lowering the outer fence, while the knowledge of the user allows lowering the inner fence: merging their knowledge, they would then be able to bring down both fences and enjoy the **open** view on the resource. The risk of collusion then arises on resources for which a user holds a **sel_locked** view and the user never had the permission to access the resource (i.e., the user never belonged to the *acl* of the resource). Indeed, if a user would get access to a resource she previously had permission for, the user has no gain in colluding with the server. This situation can happen if the release of the key at the BEL is necessary to make accessible to the user another resource r' that is, at the BEL, encrypted with the same key as r . To illustrate, suppose that at initialization time resources r and r' are both encrypted with the same key and they are not accessible by user u (see the leftmost view in Figure 26). Suppose then that u is granted the permission for r' . To make r' accessible at the BEL, a token is added to make the key corresponding to label $\phi_b(r)$ derivable by u , where however $\phi_b(r) = \phi_b(r')$. Hence, r' will be over-encrypted at the SEL and the key corresponding to label $\phi_s(r')$ made derivable by u . The resulting situation is illustrated in Figure 26, where r' is **open** and r results **sel_locked**. Thus, colluding with the server, the user would gain access to r .

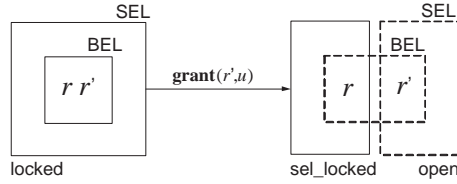


Figure 26: From locked to sel_locked views

Figure 25(b) summarizes the possible view transitions in the Delta_SEL approach. It is easy to see that, in this case, a single user by herself can hold, at different points in time, the two different views: sel_locked and bel_locked. In other words a (planning-ahead) user could retrieve the resource at initialization time, when she is not authorized, getting and storing at her side r 's bel_locked view. If, at a later point in time the user is released the key corresponding to label $\phi_b(r)$ to make accessible to her another resource r' , she will acquire the sel_locked view on r . Merging this with the past bel_locked view, she can enjoy the open view on r . Note that the set of resources potentially exposed to a user coincides with the resources exposed to collusion between that user and the server in the Full_SEL approach.

It is important to note that in both cases (Full_SEL and Delta_SEL), this exposure only impacts resources that have been involved in a policy split to make other resources, encrypted with the same BEL key, available to the user. Exposure is therefore limited and well identifiable. This allows the owner to counteract it, when the owner feels specific risks have to be minimized, via explicit selective re-encryption or by proper design (as discussed in the next section).

5.5 Experimental results

An important issue for the success of the techniques illustrated in this chapter is their scalability. The potential for their adoption would be greatly compromised if they were not applicable in large-scale scenarios. The two series of experiments illustrated in the following evaluate: *i*) the number of tokens needed for representing an authorization policy; and *ii*) the performance of over-encryption, in terms of the time required for deriving keys and for downloading and decrypting resources, proving the feasibility of the proposed approach.

Evaluation of the number of tokens. Since there is no large scale access control system available today, the experiments use a description of the structure of a large social network (i.e., the DBLP bibliography) to derive a number of resource dissemination requests. The assumption at the basis of this series of experiments is that each paper represents a resource that must be accessible by all its authors. The graph in Figure 27(a) illustrates how the number of tokens increases with the number of users. We observe that the growth is linear and that the number of tokens remains low (with 2000 authors, we have 3369 tokens).

Another important metric was the one evaluating the impact of the vertices factorization process on the number of tokens. The optimization presented a very limited benefit in

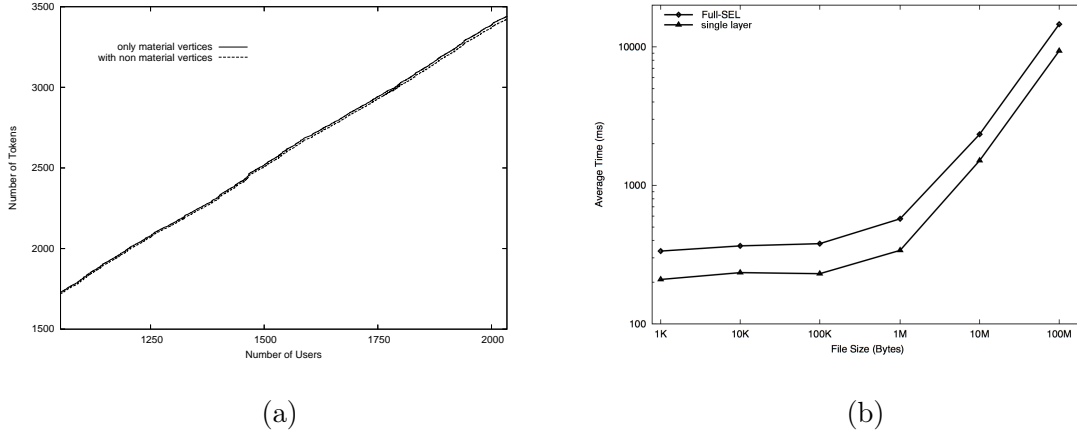


Figure 27: Number of tokens for the DBLP scenario (a) and total time required for retrieving keys and resources with single layer encryption and Full_SEL over-encryption (b)

the DBLP scenario, as visible from Figure 27(a) (18 tokens gained out of 3369, thanks to the introduction of 12 non material vertices). The rationale is that the structure of the social network is relatively sparse. However, in more complex scenarios we evaluated, the optimization resulted highly effective and the number of tokens after the application of the optimization techniques increases linearly with the increase in the number of users, with no sign of divergence for extremely large configurations [DFJ⁺10b].

Evaluation of the performance of over-encryption. The implemented prototype is a Web-based file sharing application, with a Java server answering requests originated in the client by a Firefox plugin. The extension was integrated with the XUL model underlying the Firefox interface, uses JavaScript to control the interaction with the user, and invokes the services offered by a binary library (originally written in C++) to realize the encryption functions. Open source implementations of the SHA-1 hash function and of the AES algorithm have been used. The extension is multi-platform (Windows, MacOS X, Linux). The experiments have been executed using two distinct machines as server and client. The two computers were common PCs running Linux on the server and Windows XP SP2 on the client. The two PCs were connected by a local 100Mb/s Ethernet connection. The experiments have considered requests on resources varying in size from 1KB to 100MB, with a 10X increase at each step. Figure 27(b) shows the time required to complete the retrieval of a resource. The graph compares the time required to complete the request with a system using only BEL protection and a system using over-encryption with the Full_SEL approach. As the graph shows, for small resources the time required is doubled, whereas for large resources there is a 36% increase. The motivation is that for small resources the dominant factor is the retrieval of tokens and key derivation, which is executed twice when using over-encryption. For large resources, the difference is due to the prototype writing on disk the result of the SEL decryption before applying the BEL decryption.

5.6 Conclusions

This chapter addressed the problem of enforcing access control in a scenario where data are outsourced to external servers that, while trusted for data management, are not authorized to read the data content. The solution puts forward a novel approach combining authorizations and encryption. The chapter illustrated an approach for translating authorization policies into equivalent encryption policies, while minimizing the storage and computational overheads. The chapter also described a novel solution that allows the data owner to outsource the complete management of the authorization policy by providing two layers of encryption.

The research on this topic will continue after the end of the project, investigating the open security issues that still need to be addressed, such as the management of selective write privileges and the definition of techniques able to provide integrity guarantees to outsourced data.

Abstracts of research papers

6.1 Cryptographic mechanisms (WP2.1)

1. F. Beato, M. Kohlweiss, K. Wouters, “Scramble! Your Social Network Data,” in *Proc. of the International Symposium on Privacy Enhancing Technologies (PETS 2011)* [BKW11].

Abstract. Social network sites (SNS) allow users to share information with friends, family, and other contacts. However, current SNS sites such as Facebook or Twitter assume that users trust SNS providers with the access control of their data. In this paper we propose Scramble, the implementation of a SNS-independent Firefox extension that allows users to enforce access control over their data. Scramble lets users define access control lists (ACL) of authorised users for each piece of data, based on their preferences. The definition of ACL is facilitated through the possibility of dynamically defining contact groups. In turn, the confidentiality and integrity of one data item is enforced using cryptographic techniques. When accessing a SNS that contains data encrypted using Scramble, the plugin transparently decrypts and checks integrity of the encrypted content.

2. P. Bichsel, J. Camenisch, G. Neven, N.P. Smart, B. Warinschi, “Get Shorty via Group Signatures without Encryption,” in *Proc. of the 7th International Conference on Security and Cryptography for Networks (SCN 2010)* [BCN⁺10].

Abstract. Group signatures allow group members to anonymously sign messages in the name of a group such that only a dedicated opening authority can reveal the exact signer behind a signature. In many of the target applications, for example in sensor networks or in vehicular communication networks, bandwidth and computation time are scarce resources and many of the existent constructions simply cannot be used. Moreover, some of the most efficient schemes only guarantee anonymity as long as no signatures are opened, rendering the opening functionality virtually useless. In this paper, we propose a group signature scheme with the shortest known signature size and favorably comparing computation time, whilst still offering a strong and practically relevant security level that guarantees secure opening of sig-

natures, protection against a cheating authority, and support for dynamic groups. Our construction departs from the popular sign-and-encrypt-and-prove paradigm, which we identify as one source of inefficiency. In particular, our proposal does not use standard encryption and relies on re-randomizable signature schemes that hide the signed message so as to preserve the anonymity of signers. Security is proved in the random oracle model assuming the XDDH, LRSW and SDLP assumptions and the security of an underlying digital signature scheme. Finally, we demonstrate how our scheme yields a group signature scheme with verifier-local revocation.

3. J. Camenisch, N. Casati, T. Groß, V. Shoup, “Credential Authenticated Identification and Key Exchange,” in *Proc. of 30th Annual Cryptology Conference (CRYPTO 2010)* [CCGS10].

Abstract. Secure two-party authentication and key exchange are fundamental problems. Traditionally, the parties authenticate each other by means of their identities, using a public-key infrastructure (PKI). However, this is not always feasible or desirable: an appropriate PKI may not be available, or the parties may want to remain anonymous, and not reveal their identities. To address these needs, we introduce the notions of credential-authenticated identification (CAID) and key exchange (CAKE), where the compatibility of the parties’ credentials is the criteria for authentication, rather than the parties’ identities relative to some PKI. We formalize CAID and CAKE in the universal composability (UC) framework, with natural ideal functionalities, and we give practical, modularly designed protocol realizations. We prove all our protocols UC-secure in the adaptive corruption model with erasures, assuming a common reference string (CRS). The proofs are based on standard cryptographic assumptions and do not rely on random oracles. CAKE includes password-authenticated key exchange (PAKE) as a special case, and we present two new PAKE protocols. The first one is interesting in that it uses completely different techniques than known practical PAKE protocols, and also achieves UC-security in the adaptive corruption model with erasures; the second one is the first practical PAKE protocol that provides a meaningful form of resilience against server compromise without relying on random oracles.

4. J. Camenisch, S. Mödersheim, D. Sommer, “A Formal Model of Identity Mixer,” in *Proc. of the 15th International Workshop on Formal Methods for Industrial Critical Systems (FMICS 2010)* [CMS10].

Abstract. Identity Mixer is an anonymous credential system developed at IBM that allows users for instance to prove that they are over 18 years old without revealing their name or birthdate. This privacy-friendly technology is realized using zero-knowledge proofs. We describe a formal model of Identity Mixer that is well-suited for automated protocol verification tools in the spirit of black-box cryptography models.

5. J. Camenisch, M. Dubovitskaya, G. Neven, “Unlinkable Priced Oblivious Transfer with Rechargeable Wallets,” in *Proc. of the 14th International Conference on Financial Cryptography and Data Security (FC 2010)* [CDN10b].

Abstract. We present the first truly unlinkable priced oblivious transfer protocol. Our protocol allows customers to buy database records while remaining fully

anonymous, i.e., (1) the database does not learn who purchases a record, and cannot link purchases by the same customer; (2) the database does not learn which record is being purchased, nor the price of the record that is being purchased; (3) the customer can only obtain a single record per purchase, and cannot spend more than his account balance; (4) the database does not learn the customer's remaining balance. In our protocol customers keep track of their own balances, rather than leaving this to the database as done in previous protocols. Our priced oblivious transfer protocol is also the first to allow customers to (anonymously) recharge their balances. Finally, we prove our protocol secure in the standard model (i.e., without random oracles).

6. J. Camenisch, M. Kohlweiss, C. Soriente, "Solving Revocation with Efficient Update of Anonymous Credentials," in *Proc. of the 7th International Conference on Security and Cryptography for Networks (SCN 2010)* [CKS10].

Abstract. Anonymous credential systems promise efficient, ubiquitous access to digital services while preserving user privacy. However, their diffusion is impaired by the lack of efficient revocation techniques. Traditional credential revocation measures based on certificate revocation lists or online certification authorities do not provide privacy and cannot be used in privacy-sensitive contexts. Existing revocation techniques specifically geared towards anonymous credential systems are more involved for the credential issuer, users, as well as credential consumers as users have to prove that their credential is still valid, e.g., not included in a revocation list. We introduce a novel, non-interactive technique to update issuer-controlled attributes of anonymous credentials. Revocation is implemented by encoding the validity time of a credential into one of these attributes. With the proposed protocol, credential issuers can periodically update valid credentials off-line and publish a small per-credential update value on a public bulletin-board. Users can later download their values and re-validate their credentials to prove possession of a valid credential for the current time period. Our solution outperforms all prior solutions for credential revocation in terms of communication and computational costs for the users and credential consumers and the issuer's effort is comparable to the best prior proposals.

7. J. Camenisch, M. Dubovitskaya, G. Neven, G.M. Zaverucha, "Oblivious Transfer with Hidden Access Control Policies," in *Proc. of the 14th International Conference on Practice and Theory in Public Key Cryptography (PKC 2011)* [CDNZ11]

Abstract. Consider a database where each record has different access control policies. These policies could be attributes, roles, or rights that the user needs to have in order to access the record. Here we provide a protocol that allows the users to access the database record while: (1) the database does not learn who queries a record; (2) the database does not learn which record is being queried, nor the access control policy of that record; (3) the database does not learn whether a user's attempt to access a record was successful or not; (4) the user can only obtain a single record per query; (5) the user can only access those records for which she has the correct permissions; (6) the user does not learn any other information about the database structure and the access control policies other than whether he was granted access to the queried record, and if so, the content of the record;

and (7) the users' credentials can be revoked. Our scheme builds on the one by Camenisch, Dubovitskaya and Neven (CCS'09), who consider oblivious transfer with access control when the access control policies are public.

8. J. Camenisch, M. Dubovitskaya, R. Enderlein, G. Neven, "Oblivious Transfer with Hidden Access Control from Attribute-Based Encryption," *in submission* [CDEN11].

Abstract. We present a new instantiation of oblivious transfer with hidden access control policies (HACOT) as recently proposed by Camenisch et al. (Public-Key Cryptography 2011). This primitive allows a user to anonymously query a database where each record is protected by a hidden access control policy. At each query, the user either learns the value of a single record if the user's attributes satisfy the policy, or the mere fact that its attributes do not satisfy the policy. The database, even when colluding with the attribute issuer, learns nothing about the identity of the user, the index or the access policy of the record, or whether access was granted or denied. Our construction is based on attribute-based encryption with hidden ciphertext policies and, in comparison to the protocol of Camenisch et al., offers more expressive policies, improved efficiency, and correct detection by users when access is denied. We prove our construction secure in the common reference string model (without random oracles) under existing assumptions about groups with bilinear maps. Timing results of a prototype implementation of our scheme show that the scheme is scalable and sufficiently performant to be used in practical settings.

9. J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, "Structure Preserving CCA2 Encryption and Its Application to Oblivious Third Parties," *in submission* [CHKL11].

Abstract. In this paper we present the first public key encryption scheme that is structure preserving, i.e., our encryption scheme uses only algebraic operations. In particular it does not use hash-functions or interpret group elements as bit-strings. This makes our scheme a perfect building block for cryptographic protocols where parties for instance want to prove, to each other, properties about ciphertexts or jointly compute ciphertexts. Our scheme is also very efficient and is secure against chosen ciphertext attacks. We also provide a few example protocols for our scheme, for instance a joint computation of a ciphertext of a plaintext shared by two parties, where in the end, only one of the parties learns the ciphertext. This latter protocol serves as a building block for our second contribution which is a set of protocols that implement the concept of oblivious trusted third parties. This concept has been proposed before, but no concrete realization was known before.

10. J. Camenisch, K. Haralambiev, A. Lysyanskaya, G. Neven, V. Shoup, "Memento in Cryptography," *in submission* [CHL⁺11]

Abstract. We all have to manage with an increasing number of passwords and cryptographic keys to authenticate to other parties or to encrypt and decrypt files. One popular solution involves storing all these secrets encrypted on a personal device such as a smart phone or laptop computer, encrypted under a single strong (but still human-memorizable) password. As soon as this device and/or the en-

encrypted passwords fall in the wrong hands (e.g., due to loss, theft, or intrusion), the user is vulnerable to an off-line dictionary attack. In this paper we propose a better and very practical solution relying on two different devices or hosts in such a way that no single host stores or receives any data that allows it to perform an off-line attack. These hosts could be the user's own devices, services in the cloud, or a mix of these. Thereby only on-line dictionary attacks are possible, the effectiveness of which can be limited by slowing down or refusing access altogether after too many failed attempts. Our protocol is very practical: in its most efficient form (under the DDH assumption in the random oracle model) it requires the user's device to do only perform 8 modular exponentiations (e.g., in an elliptic curve group). Prior similar solutions were either much more expensive or provide much weaker security guarantees. We achieve our results by careful protocol design and focusing on a two-server setting only.

11. J. Camenisch, G. Neven, M. Rückert, "Credentials from Lattices," *in submission* [CNR11].

Abstract. One of the most important tools to protect privacy in electronic communication systems are anonymous authentication schemes such as group signatures and anonymous credentials. The only scheme that remains secure in a post-quantum world is the recent one from Gordon et al. (ASIACRYPT 2010). Unfortunately, their scheme provides only very weak anonymity, i.e., it protects the users' privacy only as long as *no* user's anonymity is revoked – a severe drawback. Moreover, it requires that the group manager is fully trusted as it knows all the users' secret keys and could frame them. In this paper we overcome both of these limitations. We achieve even more: we define and construct two *anonymous attribute token* schemes (one without and one with anonymity revocation) where users are issued attribute-credentials, which they can use to generate anonymous authentication tokens. An example of an anonymous attribute token scheme without anonymity revocation based on the discrete logarithm problem is Microsoft's U-Prove scheme. A group signature scheme is the special case of an anonymous attribute token scheme with anonymity revocation where the issuer asserts to all users a single fixed attribute, e.g., the group identifier. Our new group signature scheme resulting from this provides the strongest form of anonymity: it is guaranteed even under adaptive chosen signature (ciphertext) attacks. We construct our schemes from new lattice-based cryptographic tools, e.g., for aggregating signatures and verifiable CCA2-secure encryption.

12. Y. De Mulder, K. Wouters, B. Preneel, "A Privacy-Preserving ID-Based Group Key Agreement Scheme Applied in VPAN," in *Proc. of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2011)* [MWP11]

Abstract. In 2008, Wan et al. presented an anonymous ID-based group key agreement scheme for wireless networks, for which they claim that it ensures anonymity and unlinkability of the group members, as well as forward and backward secrecy of the group session key. In this paper, we show that forward and backward secrecy do not hold for the protocol. We propose a correction that introduces a shielding factor that protects each member's input to the group key. We also introduce a new

feature that assures the correctness of the key as computed by all group members. This results in an increased computation cost, due to extra public key operations, and a similar communication cost. We also show in which practical setting the protocol can be deployed.

13. M. Deng, K. Wuyts, R. Scandariato, B. Preneel, W. Joosen, "A Privacy Threat Analysis Framework: Supporting the Elicitation and Fulfillment of Privacy Requirements," in *Requirements Engineering* [DWS⁺11].

Abstract. Ready or not, the digitalization of information has come and privacy is standing out there, possibly at stake. Although digital privacy is an identified priority in our society, few systematic, effective methodologies exist that deal with privacy threats thoroughly. This paper presents a comprehensive framework to model privacy threats in software-based systems. First, this work provides a systematic methodology to model privacy-specific threats. Analogous to STRIDE, an information flow oriented model of the system is leveraged to guide the analysis and to provide broad coverage. The methodology instructs the analyst on what issues should be investigated, and where in the model those issues could emerge. This is achieved by (i) defining a list of privacy threat types and (ii) providing the mappings between threat types and the elements in the system model. Second, this work provides an extensive catalogue of privacy-specific threat tree patterns that can be used to detail the threat analysis outlined above. Finally, this work provides the means to map the existing privacy-enhancing technologies (PETs) to the identified privacy threats. Therefore, the selection of sound privacy countermeasures is simplified.

14. A. Küpçü, A. Lysyanskaya, "Usable Optimistic Fair Exchange," in *The Cryptographer's Track at RSA Conference (CT-RSA 2010)* [KL10b].

Abstract. Fairly exchanging digital content is an everyday problem. It has been shown that fair exchange cannot be done without a trusted third party (called the Arbiter). Yet, even with a trusted party, it is still non-trivial to come up with an efficient solution, especially one that can be used in a p2p file sharing system with a high volume of data exchanged. We provide an efficient optimistic fair exchange mechanism for bartering digital files, where receiving a payment in return to a file (buying) is also considered fair. The exchange is optimistic, removing the need for the Arbiter's involvement unless a dispute occurs. While the previous solutions employ costly cryptographic primitives for every file or block exchanged, our protocol employs them only once per peer, therefore achieving $O(n)$ efficiency improvement when n blocks are exchanged between two peers. The rest of our protocol uses very efficient cryptography, making it perfectly suitable for a p2p file sharing system where tens of peers exchange thousands of blocks and they do not know beforehand which ones they will end up exchanging. Therefore, our system yields to one-two orders of magnitude improvement in terms of both computation and communication (80 seconds vs. 84 minutes, 1.6MB vs. 100MB). Thus, for the first time, a provably secure (and privacy respecting when payments are made using e-cash) fair exchange protocol is being used in real bartering applications (e.g., BitTorrent) [14] without sacrificing performance.

15. A. K upc u, A. Lysyanskaya, “Optimistic Fair Exchange with Multiple Arbiters,” in *Proc. of the 15th European Symposium on Research in Computer Security (ESORICS 2010)* [KL10a].

Abstract. Fair exchange is one of the most fundamental problems in secure distributed computation. Alice has something that Bob wants, and Bob has something that Alice wants. A fair exchange protocol would guarantee that, even if one of them maliciously deviates from the protocol, either both of them get the desired content, or neither of them do. It is known that no two-party protocol can guarantee fairness in general; therefore the presence of a trusted arbiter is necessary. In optimistic fair exchange, the arbiter only gets involved in case of faults, but needs to be trusted. To reduce the trust put in the arbiter, it is natural to consider employing multiple arbiters. Expensive techniques like byzantine agreement or secure multi-party computation with $\Omega(n^2)$ communication can be applied to distribute arbiters in a non-autonomous way. Yet we are interested in efficient protocols that can be achieved by keeping the arbiters autonomous (non-communicating), especially for p2p settings in which the arbiters do not even know each other. Avoine and Vaudenay employ multiple autonomous arbiters in their optimistic fair exchange protocol which uses global timeout mechanisms; all arbiters have access to -loosely- synchronized clocks. They left two open questions regarding the use of distributed autonomous arbiters: (1) Can an optimistic fair exchange protocol without timeouts provide fairness (since it is hard to achieve synchronization in a p2p setting) when employing multiple autonomous arbiters? (2) Can any other optimistic fair exchange protocol with timeouts achieve better bounds on the number of honest arbiters required? In this paper, we answer both questions negatively. To answer these questions, we define a general class of optimistic fair exchange protocols with multiple arbiters, called “distributed arbiter fair exchange” (DAFE) protocols. Informally, in a DAFE protocol, if a participant fails to send a correctly formed message, the other party must contact some subset of the arbiters and get correctly formed responses from them. The arbiters do not communicate with each other, but only to Alice and Bob. We prove that no DAFE protocol can meaningfully exist.

16. A. Rial, B. Preneel, “Optimistic Fair Priced Oblivious Transfer,” in *Proc. of the 3rd International Conference on Cryptology in Africa (AFRICACRYPT 2010)* [RP10].

Abstract. Priced oblivious transfer (POT) is a two-party protocol between a vendor and a buyer in which the buyer purchases digital goods without the vendor learning what is bought. Although privacy properties are guaranteed, current schemes do not offer fair exchange. A malicious vendor can, e.g., prevent the buyer from retrieving the goods after receiving the payment, and a malicious buyer can also accuse an honest vendor of misbehavior without the vendor being able to prove this untrue. In order to address these problems, we define the concept of optimistic fair priced oblivious transfer and propose a generic construction that extends secure POT schemes to realize this functionality. Our construction, based on verifiably encrypted signatures, employs a neutral adjudicator that is only involved in case of dispute, and shows that disputes can be resolved without the buyer losing her privacy, i.e., the buyer does not need to disclose which digital

goods she is interested in. We show that our construction can be instantiated with an existing universally composable POT scheme, and furthermore we propose a novel full-simulation secure POT scheme that is much more efficient.

6.2 Mechanisms supporting users' privacy and trust (WP2.2)

1. B. Kellermann. "Open Research Questions of Privacy-enhanced Event Scheduling," in *Proc. of iNetSec 2010: Open Research Problems in Network Security* [Kel10].
Abstract. Event-scheduling applications like Doodle have the problem of privacy relevant information leakage. A simple idea to prevent this would be to use an e-voting scheme instead. However, this solution is not sufficient as we will show within this paper. Additionally we come up with requirements and several research questions related to privacy-enhanced event scheduling. These address privacy, security as well as usability of privacy-enhanced event scheduling.
2. B. Kellermann, "Privacy-enhanced Web-based Event Scheduling with Majority Agreement," in *Proc. of the 26th IFIP TC-11 International Information Security Conference (SEC 2011)* [Kel11].
Abstract. Applications which help users to schedule events are becoming more and more important. A drawback of most existing applications is, that the preferences of all participants are revealed to the others. Previously proposed privacy-friendly solutions could only schedule meetings if all participants were available at the same time slot. We propose a new scheme, which overcomes this limitation, i.e., the meeting can be scheduled at the time slot, where just the majority of participants is available. Duddle (<http://dudle.inf.tu-dresden.de>), a web-application which implements the protocol is presented. We measured its performance in order to show that the protocol is practical and feasible.
3. M. Berg, K. Borcea-Pfitzmann, "Implementability of the Identity Management Part in Pfitzmann/Hansen's Terminology for a Complex Digital World," in *Proc. of PrimeLife / IFIP Summerschool on Privacy and Identity Management for Life* [BBP11].
Abstract. Based on a widely cited terminology, this paper provides different interpretations of concepts introduced in the terminology asking for an implementable privacy model for computer-mediated interactions between individuals. A separation of the digital world and the physical world is proposed, as well as a linkage of the two worlds. The digital world contains digital representations of individuals and it consists of pure data. The physical world contains individuals and it consists of information (produced by individuals) and data. Moreover, a refined definition of privacy is being elaborated that serves as justification for identity management of individuals interested in a sophisticated perspective of privacy.
4. S. Schiffner, S. Clauß, S. Steinbrecher, "Privacy, Liveliness and Fairness für Reputation," in *Proc. of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2011)* [SCS11].

Abstract. In various Internet applications, reputation systems are typical means to collect experiences users make with each other. We present a reputation system that balances the security and privacy requirements of all users involved. Our system provides *privacy* in the form of information theoretic relationship anonymity w.r.t. users and the reputation provider. Furthermore, it preserves *liveliness*, i.e., all past ratings can influence the current reputation profile of a user. In addition, mutual ratings are forced to be simultaneous and self rating is prevented, which enforces *fairness*. What is more, without performing mock interactions—even if all users are colluding—users cannot forge ratings. As far as we know, this is the first protocol proposed that fulfills all these properties simultaneously.

5. S. Pötzsch, “Einfluss Wahrgenommener Privatsphäre und Anonymität auf Forennutzer,” in *Proc. of Mensch & Computer* [Pöt10].

Abstract. User-generated content becomes more and more important on the Internet. This implies that also the ability of users to make an informed and well-aware decision whether and to which detail they like to disclose personal data, gains in importance. Using data from an empirical study, this paper researches the influence of privacy-awareness information a) on users' perceived privacy and anonymity and b) on their actual disclosure behavior.
6. S. Pötzsch, P. Wolkerstorfer, C. Graf, “Privacy-awareness Information for Web Forums: Results from an Empirical Study,” in *Proc. of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* [PWG10].

Abstract. While interacting with others on the Internet, users share a lot of personal data with a potentially large but *invisible* audience. An important issue is maintaining control over personal data and therefore, in the first place, users need to be aware to whom they are disclosing which data. Based on the cues-filtered-out theory we introduce a new feature to support the privacy-awareness of forum users and tested it with 313 users. The results of our empirical study show that the presentation of privacy-related context cues indeed increases forum users' privacy-awareness. This is an important precondition for users' willingness to modify privacy settings or to use privacy-enhancing technologies.
7. B. Kellermann, S. Pötzsch, S. Steinbrecher, “Privacy-respecting Reputation for Wiki Users,” in *Proc. of the 5th IFIP WG 11.11 International Conference on Trust Management (IFIPTM 2011)* [KPS11].

Abstract. Wikis are popular tools for creation and sharing of content. Integrated reputation systems allow to assess expertise and reliability of authors and thus to support trust in the wiki content. Yet, results from our empirical study indicate that the disclosure of user reputation evokes privacy issues. As a solution for this conflict between the need to evaluate trustworthiness of users and protecting their privacy, we present a privacy-respecting reputation system for wikis that we realized as OpenSource-Extension for the wiki software MediaWiki.
8. S. Pötzsch, R. Böhme, “The Role of Soft Information in Trust Building: Evidence from Online Social Lending,” in *Proc. of the 3rd International Conference on Trust and Trustworthy Computing (TRUST 2010)* [PB10].

Abstract. We analyze empirical data of Germany's largest online social lending platform Smava.de to exemplarily study the contribution of unstructured, ambiguous, or unverified information to trust building in online communities. After controlling for the influence of hard information, we find that textual statements that appeal to social behavior actually affect trust building. However, the evidence is less clear for voluntarily disclosed personal data. Lenders generally seem to give more weight to hard information so that disclosing personal data promises little benefit while potentially exposing borrowers to privacy risks.

9. R. Böhme, S. Pöttsch, "Social Lending aus der Perspektive des Datenschutzes," in *SICHERHEIT 2010 - Sicherheit, Schutz und Zuverlässigkeit* [BP10].

Abstract. Online social lending refers to the idea of loan origination among private persons. Borrowers publish credit applications on websites which match them with private investors. We point to a conflict between economic interests and privacy goals in online social lending, empirically analyze the effect of data disclosure on credit conditions using empirical data from the popular German platform Smava.de. Results suggest that it does not pay off for borrowers to disclose more personal data than absolutely necessary.

10. R. Böhme, S. Pöttsch, "Collective Exposure: Peer Effects in Voluntary Disclosure of Personal Data," in *Proc. of the 15th International Conference on Financial Cryptography and Data Security (FC 2011)* [BP11].

Abstract. This paper reports empirical evidence for peer effects in privacy behavior using field data from online social lending. Our content analysis and regression models show that individuals copy observable behavior of others in decisions on a) how much to write about oneself, b) whether to share custom pictures, c) what personal data to disclose, and d) how identifiable to present oneself. We frame this finding in the theory of descriptive social norms and analyze moderating effects, such as similarity of context, social proximity, and mimicry of success factors. The presence of peer effects in disclosure behavior can explain the formation and change of apparent social norms and attitudes towards privacy.

11. S. Steinbrecher, "The Need for Interoperable Reputation Systems," in *Open Research Problems in Network Security* [Ste09].

Abstract. Nowadays more and more Internet applications install reputation systems to collect opinions users have about some reputation objects. The opinions are usually formalized in the form of ratings the reputation system can use to build overall reputation profiles of the reputation objects. Reputation objects might be other users, products, web content and anything else that can be rated. Users may investigate the reputation object's reputation profile to estimate its quality resp. trustworthiness. As there are currently many providers of reputation systems it would be desirable to make reputation information in different systems interoperable or to establish meta reputation systems that collect information from various applications resp. their reputation systems. This process should consider both interoperability of reputation systems themselves and their interoperability with applications, trust and identity management systems as we will discuss in this paper.

6.3 Privacy of data (WP2.3)

1. C.A. Ardagna, S. Jajodia, P. Samarati, A. Stavrou, “Providing Mobile Users’ Anonymity in Hybrid Networks,” in *Proc. of the 15th European Symposium on Research in Computer Security (ESORICS 2010)* [AJSS10].

Abstract. We present a novel hybrid communication protocol that guarantees mobile users’ k-anonymity against a wide-range of adversaries by exploiting the capability of handheld devices to connect to both WiFi and cellular networks. Unlike existing anonymity schemes, we consider all parties that can intercept communications between the mobile user and a server as potential privacy threats. We formally quantify the privacy exposure and the protection of our system in the presence of malicious neighboring peers, global WiFi eavesdroppers, and omniscient mobile network operators. We show how our system provides an automatic incentive for users to collaborate, since by forwarding packets for other peers users gain anonymity for their own traffic.

2. M. Bezzi, S. De Capitani di Vimercati, G. Livraga, P. Samarati, “Protecting Privacy of Sensitive Value Distributions in Data Release,” in *Proc. of the 6th Workshop on Security and Trust Management (STM 2010)* [BDLS10].

Abstract. In today’s electronic society, data sharing and dissemination are more and more increasing, leading to concerns about the proper protection of privacy. In this paper, we address a novel privacy problem that arises when non sensitive information is incrementally released and sensitive information can be inferred exploiting dependencies of sensitive information on the released data. We propose a model capturing this inference problem where sensitive information is characterized by peculiar distributions of non sensitive released data. We also discuss possible approaches for run time enforcement of safe releases.

3. M. Bezzi, “An Information Theoretic Approach for Privacy Metrics,” in *Transactions on Data Privacy* [Bez10].

Abstract. Organizations often need to release microdata without revealing sensitive information. To this scope, data are anonymized and, to assess the quality of the process, various privacy metrics have been proposed, such as k-anonymity, l-diversity, and t-closeness. These metrics are able to capture different aspects of the disclosure risk, imposing minimal requirements on the association of an individual with the sensitive attributes. If we want to combine them in a optimization problem, we need a common framework able to express all these privacy conditions. Previous studies proposed the notion of mutual information to measure the different kinds of disclosure risks and the utility, but, since mutual information is an average quantity, it is not able to completely express these conditions on single records. We introduce here the notion of one-symbol information (i.e., the contribution to mutual information by a single record) that allows to express and compare the disclosure risk metrics. In addition, we obtain a relation between the risk values t and l, which can be used for parameter setting. We also show, by numerical experiments, how l-diversity and t-closeness can be represented in terms of two different, but equally acceptable, conditions on the information gain.

4. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Combining Fragmentation and Encryption to Protect Privacy in Data Storage,” in *ACM Transactions on Information and System Security* [CDF⁺10].
Abstract. The impact of privacy requirements in the development of modern applications is increasing very quickly. Many commercial and legal regulations are driving the need to develop reliable solutions for protecting sensitive information whenever it is stored, processed, or communicated to external parties. To this purpose, encryption techniques are currently used in many scenarios where data protection is required since they provide a layer of protection against the disclosure of personal information, which safeguards companies from the costs that may arise from exposing their data to privacy breaches. However, dealing with encrypted data may make query processing more expensive. In this paper, we address these issues by proposing a solution to enforce privacy of data collections that combines data fragmentation with encryption. We model privacy requirements as confidentiality constraints expressing the sensitivity of attributes and their associations. We then use encryption as an underlying (conveniently available) measure for making data unintelligible, while exploiting fragmentation as a way to break sensitive associations among attributes. We formalize the problem of minimizing the impact of fragmentation in terms of number of fragments and their affinity and present two heuristic algorithms for solving such problems. We also discuss experimental results comparing the solutions returned by our heuristics with respect to optimal solutions, which show that the heuristics, while guaranteeing a polynomial-time computation cost are able to retrieve solutions close to optimum.
5. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Selective Data Outsourcing for Enforcing Privacy,” in *Journal of Computer Security* [CDF⁺11a]
Abstract. Existing approaches for protecting sensitive information outsourced at external “honest-but-curious” servers are typically based on an overlying layer of encryption applied on the whole information, or use a combination of fragmentation and encryption. In this paper, we put forward a novel paradigm for preserving privacy in data outsourcing, which departs from encryption. The basic idea is to involve the owner in storing a limited portion of the data, while storing the remaining information in the clear at the external server. We analyze the problem of computing a fragmentation that minimizes the workload of the owner, which is represented using different metrics and corresponding weight functions, and prove that this minimization problem is NP-hard. We then introduce the definition of locally minimal fragmentation that is used to efficiently compute a fragmentation via a heuristic algorithm. The algorithm works on a modelization of the problem of finding a locally minimal fragmentation as a hypergraph 2-coloring problem. Finally, we illustrate the execution of queries on fragments and provide experimental results comparing the fragmentations returned by our heuristics with respect to optimal fragmentations. The experiments show that the heuristics guarantees a low computation cost and is able to compute a fragmentation close to optimum.
6. S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Fragments and Loose Associations: Respecting Privacy in Data Publishing,” in

Proc. of the VLDB Endowment [DFJ⁺10c].

Abstract. We propose a modeling of the problem of privacy-compliant data publishing that captures confidentiality constraints on one side and visibility requirements on the other side. Confidentiality constraints express the fact that some attributes, or associations among them, are sensitive and cannot be released. Visibility requirements express requests for views over data that should be provided. We propose a solution based on data fragmentation to split sensitive associations while ensuring visibility. In addition, we show how sensitive associations broken by fragmentation can be released in a sanitized form as loose associations formed in a way to guarantee a specified degree of privacy.

7. S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, P. Samarati, “Efficient and Private Access to Outsourced Data,” in *Proc. of the 31st International Conference on Distributed Computing Systems (ICDCS 2011)* [DFP⁺11].

Abstract. As the use of external storage and data processing services for storing and managing sensitive data becomes more and more common, there is an increasing need for novel techniques that support not only data confidentiality, but also confidentiality of the accesses that users make on such data. In this paper, we propose a technique for guaranteeing content, access, and pattern confidentiality in the data outsourcing scenario. The proposed technique introduces a *shuffle index* structure, which adapts traditional *B+*-trees. We show that our solution exhibits a limited performance cost, thus resulting effectively usable in practice.

8. B.J. Koops, “Een Herdenking van Dataprotectie: Datamist en Beslissingstransparantie als Alternatief,” in *Jaarboek ICT and Samenleving: Privacy Revisited, Delft: TNO* [Koo11a].

Abstract. In this Dutch contribution to a renowned ICT yearbook, I explore alternatives to data protection as we know it. It observes that a) European data-protection law faces fundamental challenges in the database age (due to data explosion, risk culture and function creep), b) the existing pillars of the Directive (such as data minimisation, purpose-binding, and user control) are not well-equipped to address these challenges, and c) that the current focus of revising the Directive (simplification, more enforcement, PETs, and more transparency in data collection and processing) clings to the old pillars and does not promise much to fundamentally address data protection in the database age. Therefore, the chapter argues for a dual alternative approach. First, data subjects can employ the strategy of ‘data fog’, i.e., confuse data processors through a combination of data obfuscation and data exhibitionism. Second, decision transparency should be fostered, i.e., organisations that make decisions about individuals should be forced to be transparent about which data are used in which ways in their decisions. This downward transparency can be effected through increased legal (FOIA-type) obligations as well as through *sousveillance*. This dual alternative approach needs much more reflection and analysis, but it may be better equipped to really protect individuals in the database age than a (revised) Directive will be able to if it clings to a paper exercise in data minimisation and user control.

9. B.J. Koops, “Effecting Decision Transparency to Achieve Real Data Protection af-

ter the Computational Turn,” in *Privacy and Due Process after the Computational Turn* [Koo11b].

Abstract. This chapter puts forth and expands the ideas of the Dutch chapter [Koo11a] on decision transparency to effectively protect individuals in the database age. After briefly analysing the limitations of the Data Protection Directive and of the current plans for revision, it provides a theoretical and practical perspective on effecting decision transparency. The first, theoretical, part applies the conceptual framework of Heald on transparency relations to explain data-processing relationships. This shows that privacy and due process can be fostered by altering the relationship between data processors and data subjects in two ways: diminishing upwards transparency (i.e., subjects becoming less visible to the organisations hierarchically above them) through data obfuscation, and enhancing downwards transparency (i.e., organisations being more visible to the subjects below them) through legal and technical measures. The second part of the chapter provides an analysis of ways in which the theoretical approach could be effected in practice. It describes illustrative cases of downwards transparency and analyses how these could be applied to data protection. This yields directions in which data protection can be effected through increased downwards transparency. This is subsequently compared with the approach of data-minimised, PET-enabled user control underlying the Directive revision. Finally, an assessment is made whether the two approaches (decision transparency versus PET-enabled user control) should be seen as mutually exclusive or complementary.

10. B.J. Koops, “The (In)flexibility of Techno-regulation and the Case of Purpose-Binding,” in *Legisprudence* (submitted, to appear in special issue on techno-regulation) [Koo11d].

Abstract. Current literature on techno-regulation - the conscious deployment of technology to regulate people’s behaviour - briefly touches upon the issue of flexibility. On the one hand, it is suggested that technology-embedded rules tend to be rigid, whereas legal norms are flexible and open to interpretation. On the other hand, technology in principle allows for flexibility through open configurations and the plasticity of software, while law is relatively static. <purpose> This paper analyses the role of open norms and software plasticity in techno-regulation in order to shed more light on the (in)flexibility of techno-regulation.</purpose> This is done through a case study of the legal norm of purpose-binding in data-protection legislation. The issue of flexibility and interpretation plays at the level of both the legal norm (purpose-binding in the Data Protection Directive) and its application in practice (defining purposes for concrete data processing). Regulating data protection with purpose-binding is trying to control a moving target - purposes for data processing that shift in the database age - with an instrument that is itself far from fixed, since it is an open, procedural norm. <other-purpose>The case study provides a heuristic for looking at the feasibility of techno-regulation</other-purpose>, in determining three steps involved in techno-regulation: identifying the legal norm, moving from legal norm to techno-rule, and deploying the techno-rule in practice. The core step is the second, where the transition from law to technology is primarily made, in which three levels are distinguished: a) development of

technical frameworks (e.g., privacy markup languages such as P3P and EPAL), b) filling in the frameworks for concrete cases, and c) enforcing links between actions and rules within the framework.

The case of purpose-binding adds to our understanding of techno-regulation and flexible rules in showing that a trade-off exists between the plasticity of technology in techno-regulation and the usefulness and adoption of techno-regulation. The more plastic the techno-regulation, the less it adds to legal regulation; and the more it can add, the more rigid it will have to be. Techno-regulation may be a realistic venture for simple rules that are well suited to be represented computationally, which may help organisations in compliance assurance, but it has little added value in terms of enhancing precision or enforceability, and is therefore not particularly interesting from a regulatory and theoretical perspective. Techno-regulation as enforcement of a legal norm is problematic if the norm itself is complex due to openness, fuzziness, contextual complexity, or regulatory turbulence. Since much cyberlaw and privacy law is complex and in flux, perhaps paradoxically, techno-regulation does not seem particularly suited to regulate cyberspace itself or to enhance privacy. The outlook for techno-regulation may therefore be limited. Rules need breathing space, and it still takes a human being to make a rule come to life.

11. B.J. Koops, “Forgetting Footprints, Seizing Shadows. A Critical Analysis of a ‘Right to be Forgotten’ in Big Data Practice,” *in submission* [Koo11c].

Abstract. The so-called “right to be forgotten” has been put firmly on the agenda, both of academia and of policy. Although the idea is intuitive and appealing, the practical implications and form of a right to be forgotten have hardly yet been analysed. This contribution aims to critically assess what a right to be forgotten could or should entail in actual practice. It outlines the current socio-technical context as one of Big Data, in which massive data collections are mined and used for many, sometimes unforeseen, purposes. Individuals’ digital shadow (information about them generated by others) has by now outgrown individuals’ digital footprint (data they leave behind themselves). This provides particular challenges for the right to be forgotten, that will be discussed in the form of three key questions. Against whom can the right be invoked? When and why can the right be invoked? And how can the right be invoked or effected?

6.4 Access control for the protection of user-generated data (WP2.4)

1. M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzaretti, V. Piuri, F. Scotti, “A Privacy-Compliant Fingerprint Recognition System Based on Homomorphic Encryption and Fingerprint Template,” in *Proc. of the 4th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2010)* [BBC⁺10].

Abstract. The privacy protection of the biometric data is an important research topic, especially in the case of distributed biometric systems. In this scenario, it is very important to guarantee that biometric data cannot be stealed by anyone,

and that the biometric clients are unable to gather any information different from the single user verification/identification. In a biometric system with high level of privacy compliance, also the server that processes the biometric matching should not learn anything on the database and it should be impossible for the server to exploit the resulting matching values in order to extract any knowledge about the user presence or behavior. Within this conceptual framework, in this paper we propose a novel complete demonstrator based on a distributed biometric system that is capable to protect the privacy of the individuals by exploiting cryptosystems. The implemented system computes the matching task in the encrypted domain by exploiting homomorphic encryption and using Fingerprint templates. The paper describes the design methodology of the demonstrator and the obtained results. The demonstrator has been fully implemented and tested in real applicative conditions. Experimental results show that this method is feasible in the cases where the privacy of the data is more important than the accuracy of the system and the obtained computational time is satisfactory.

2. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, G. Livraga, P. Samarati, "Enforcing Confidentiality and Data Visibility Constraints: An OBDD Approach," in *Proc. of the 25th Annual WG 11.3 Conference on Data and Applications Security and Privacy (DBSec 2011)* [CDF⁺11b].

Abstract. The problem of enabling privacy-preserving data releases has become more and more important in the last years thanks to the increasing needs of sharing and disseminating information. In this paper we address the problem of computing data releases in the form of *fragments* (vertical views) over a relational table, which satisfy both confidentiality and visibility constraints, expressing needs for information protection and release, respectively. We propose a modeling of constraints and of the data fragmentation problem based on Boolean formulas and Ordered Binary Decision Diagrams (OBDDs). Exploiting OBDDs, we efficiently manipulate Boolean formulas, thus easily computing data fragments that satisfy the constraints.

3. S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, "Authorization Enforcement in Distributed Query Evaluation," in *Journal of Computer Security* [DFJ⁺11].

Abstract. We present a simple, yet powerful, approach for the specification and enforcement of authorizations regulating data release among data holders collaborating in a distributed computation, to ensure that query processing discloses only data whose release has been explicitly authorized. Data disclosure is captured by means of profiles, associated with each data computation, that describe the information carried by a base or a derived (i.e., computed by a query) relation. We present an algorithm that, given a query plan, determines whether it can be safely executed and produces a safe execution strategy for it. For each operation in a safe query plan, the algorithm determines the server(s) responsible for the execution, based on the entailed information flows, considering different strategies for the execution of joins. Finally, we discuss the architecture of a distributed database system based on the proposed model, illustrating possible design choices and their impact.

4. S. De Capitani di Vimercati, S. Foresti, G. Livraga, P. Samarati, “Anonymization of Statistical Data,” in *IT - Information Technology* [DFLS11].
Abstract. In the modern digital society, personal information about individuals can be collected, stored, shared and disseminated much more easily and freely. Such data can be released in *macrodata* form, reporting aggregated information, or in *microdata* form, reporting specific information on individual respondents. To ensure proper privacy of individuals as well of public and private organizations, it is then important to protect possible sensitive information in the original dataset from either direct or indirect disclosure. In this paper, we characterize macrodata and microdata releases and then focus on microdata protection. We provide a characterization of the main microdata protection techniques and describe recent solutions for protecting microdata against identity and attribute disclosure, discussing some open issues that need to be investigated.
5. S. De Capitani di Vimercati, S. Foresti, P. Samarati, “Protecting Information Privacy in the Electronic Society,” in *e-Business and Telecommunications International Conference (ICETE 2009)* [DFS11].
Abstract. The privacy of users, the confidentiality of organizations, and the protection of huge collections of sensitive information, possibly related to data that might be released publicly or semi-publicly for various purposes, are essential requirements for the today’s Electronic Society. In this chapter, we discuss the main privacy concerns that arise when releasing information to third parties. In particular, we focus on the data publication and data outsourcing scenarios, illustrating the emerging trends in terms of privacy and data protection and identifying some research directions to be investigated.
6. S. De Capitani di Vimercati, S. Foresti, G. Livraga, “Privacy in Data Publishing,” in *Data Privacy Management and Autonomous Spontaneous Security* [DFL11].
Abstract. In modern digital society, personal information about individuals can be easily collected, shared, and disseminated. These data collections often contain sensitive information, which should not be released in association with respondents’ identities. Removing explicit identifiers before data release does not offer any guarantee of anonymity, since de-identified datasets usually contain information that can be exploited for linking the released data with publicly available collections that include respondents’ identities. To overcome these problems, new proposals have been developed to guarantee privacy in data release. In this chapter, we analyze the risk of disclosure caused by public or semi-public microdata release and we illustrate the main approaches focusing on protection against unintended disclosure. We conclude with a discussion on some open issues that need further investigation.
7. S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Encryption Policies for Regulating Access to Outsourced Data,” in *ACM Transactions on Database Systems (TODS)* [DFJ⁺10b].
Abstract. Current access control models typically assume that resources are under the strict custody of a trusted party, which monitors each access request to verify if it is compliant with the specified access control policy. There are many

scenarios where this approach is becoming no longer adequate. Many clear trends in Web technology are creating a need for owners of sensitive information to manage access to it by legitimate users using the services of *honest but curious* third parties, that is, parties trusted with providing the required service but not authorized to read the actual data content. In this scenario, the data owner encrypts the data before outsourcing and stores them at the server. Only the data owner and users with knowledge of the key will be able to decrypt the data. Possible access authorizations are to be enforced by the owner. In this paper, we address the problem of enforcing selective access on outsourced data without need of involving the owner in the access control process. The solution puts forward a novel approach that combines cryptography with authorizations, thus enforcing access control via *selective encryption*. The paper presents a formal model for access control management and illustrates how an authorization policy can be translated into an equivalent encryption policy while minimizing the amount of keys and cryptographic tokens to be managed. The paper also introduces a two-layer encryption approach that allows the data owner to outsource, besides the data, the complete management of the authorization policy itself, thus providing efficiency and scalability in dealing with policy updates. We also discuss experimental results showing that our approach is able to efficiently manage complex scenarios.

8. S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, G. Pelosi, P. Samarati, "Encryption-based Policy Enforcement for Cloud Storage," in *Proc. of the 1st ICDCS Workshop on Security and Privacy in Cloud Computing (SPCC 2010)* [DFJ⁺10a].

Abstract. Nowadays, users are more and more exploiting external storage and connectivity for sharing and disseminating user-generated content. To this aim, they can benefit of the services offered by Internet companies, which however assume that the service provider is entitled to access the resources. To overcome this limitation, we present an approach that does not require complete trust in the external service w.r.t. both resource content and authorization management, while at the same time allowing users to delegate to the provider the enforcement of the access control policy on their resources. Our solution relies on the translation of the access control policy into an equivalent encryption policy on resources and on a hierarchical key structure that limits both the number of keys to be maintained and the amount of encryption to be enforced.

9. P. Samarati, S. De Capitani di Vimercati, "Data Protection in Outsourcing Scenarios: Issues and Directions," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2010)* [SD10].

Abstract. Data outsourcing is an emerging paradigm that allows users and companies to give their (potentially sensitive) data to external servers that then become responsible for their storage, management, and dissemination. Although data outsourcing provides many benefits, especially for parties with limited resources for managing an ever more increasing amount of data, it introduces new privacy and security concerns. In this paper we discuss the main privacy issues to be addressed in data outsourcing, ranging from data confidentiality to data utility. We then illustrate the main research directions being investigated for providing effective data

protection to data externally stored and for enabling their querying.

10. S. De Capitani di Vimercati, S. Foresti, “Privacy of Outsourced Data,” in *Privacy and Identity Management for Life* [DF10].

Abstract. Data outsourced to an external storage server are usually encrypted since there is the common assumption that all data are equally sensitive. The encrypted data however cannot be efficiently queried and their selective release is not possible or require the application of specific solutions. To overcome these problems, new proposals have been recently developed, which are based on a fragmentation technique possibly combined with encryption. The main advantage of these proposals is that they limit the use of encryption, thus improving query execution efficiency. In this paper, we describe such fragmentation-based approaches focusing in particular on the different data fragmentation models proposed in the literature. We then conclude the paper with a discussion on some research directions.

Bibliography

- [ACBM08] E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin. Reputation systems for anonymous networks. In *Proc. of the 8th International Symposium on Privacy Enhancing Technologies (PETS 2008)*, Leuven, Belgium, July 2008.
- [ADFM06] G. Ateniese, A. De Santis, A.L. Ferrara, and B. Masucci. Provably-secure time-bound hierarchical key assignment schemes. In *Proc. of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, Alexandria, VA, USA, November 2006.
- [AFB05] M.J. Atallah, K.B. Frikken, and M. Blanton. Dynamic and efficient key management for access hierarchies. In *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS 2005)*, Alexandria, VA, USA, November 2005.
- [AIR01a] W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045, pages 119–135. Springer, 2001.
- [AIR01b] W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045, pages 119–135. Springer, 2001.
- [AJSS10] C.A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou. Providing mobile users’ anonymity in hybrid networks. In *Proc. of the 15th European Symposium on Research in Computer Security (ESORICS 2010)*, Athens, Greece, September 2010.
- [ASM06] M.H. Au, W. Susilo, and Y. Mu. Constant-size dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *SCN 06: 5th Security in Communication Networks*, volume 4116, pages 111–125. Springer, 2006.
- [ASW97] N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *Proc. of the 4th ACM Conference on Computer and Communications Security (CCS 1997)*, Zurich, Switzerland, April 1997.
- [ASW00] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):591–610, April 2000.

- [AT83] S. Akl and P. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer System (TOCS)*, 1(3):239–248, August 1983.
- [BB04] D. Boneh and X. Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027, pages 56–73. Springer, 2004.
- [BBC⁺10] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva. A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingerprint templates. In *Proc. of the 4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*, pages 1–7, September 2010. 978-1-4244-7580-3.
- [BBP11] M. Berg and K. Borcea-Pfitzmann. Implementability of the Identity Management Part in Pfitzmann/Hansen’s Terminology for a Complex Digital World. In *Proc. of PrimeLife / IFIP Summerschool on Privacy and Identity Management for Life*, Trento, Italy, September 2011.
- [BBS04] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In Matthew K. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer Verlag, 2004.
- [BCN⁺10] P. Bichsel, J. Camenisch, G. Neven, N.P. Smart, and B. Warinschi. Get shorty via group signatures without encryption. In Juan A. Garay and Roberto De Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 381–398. Springer, 2010.
- [BDLS10] M. Bezzi, S. De Capitani di Vimercati, G. Livraga, and P. Samarati. Protecting privacy of sensitive value distributions in data release. In *Proc. of the 6th Workshop on Security and Trust Management (STM 2010)*, Athens, Greece, September 2010.
- [Bez10] M. Bezzi. An information theoretic approach for privacy metrics. *Transactions on Data Privacy*, 3(3):199–215, 2010.
- [BG93] M. Bellare and O. Goldreich. On defining proofs of knowledge. In E.F. Brickell, editor, *Advances in Cryptology – CRYPTO ’92*, volume 740, pages 390–420. Springer, 1993.
- [BKW11] F. Beato, M. Kohlweiss, and K. Wouters. Scramble! your social network data. In Simone Fischer-Huebner and Nicholas J. Hopper, editors, *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011*, Lecture Notes in Computer Science, page 15, Waterloo,CA, 2011. Springer-Verlag.
- [BLFM05] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (Standard), January 2005.

- [BP10] R. Böhme and S. Pötzsch. Social lending aus der perspektive des datenschutztes. In *Proc. of the 5th Issue of the Sicherheit, Schutz und Zuverlässigkeit Conference (SICHERHEIT 2010)*, Berlin, Germany, October 2010.
- [BP11] R. Böhme and S. Pötzsch. Collective exposure: Peer effects in voluntary disclosure of personal data. In *Proc. of the 15th International Conference on Financial Cryptography and Data Security (FC 2011)*, St. Lucia, February 2011.
- [Bra97] S. Brands. Rapid demonstration of linear relations connected by boolean operators. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 318–333. Springer Verlag, 1997.
- [Cam98] J.L. Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem*. PhD thesis, ETH Zürich, 1998. Diss. ETH No. 12520, Hartung Gorre Verlag, Konstanz.
- [CCGS10] J. Camenisch, N. Casati, T. Groß, and V. Shoup. Credential authenticated identification and key exchange. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *LNCS*, pages 255–276, Santa Barbara, CA, USA, August 2010. Springer.
- [CCS08] J. Camenisch, R. Chaabouni, and A. Shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *Advances in Cryptology – ASIACRYPT 2008*, volume 5350, pages 234–252. Springer, 2008.
- [CDD⁺05] A. Ceselli, E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Modeling and assessing inference exposure in encrypted databases. *ACM Transactions on Information and System Security (TISSEC)*, 8(1):119–152, February 2005.
- [CDEN11] J. Camenisch, M. Dubovitskaya, R. Enderlein, and G. Neven. Oblivious transfer with hidden access control from attribute-based encryption, 2011.
- [CDF⁺10] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Combining fragmentation and encryption to protect privacy in data storage. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):22:1–22:33, July 2010.
- [CDF⁺11a] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Selective data outsourcing for enforcing privacy. *Journal of Computer Security*, 2011.
- [CDF⁺11b] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, G. Livraga, and P. Samarati. Enforcing confidentiality and data visibility constraints: An obdd approach. In *Proc. of the 25th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy (DBSec 2011)*, Richmond, VA, USA, July 2011.

- [CDM00] R. Cramer, I. Damgård, and P.D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000: 3rd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1751, pages 354–372. Springer, 2000.
- [CDN09a] J. Camenisch, M. Dubovitskaya, and G. Neven. Oblivious transfer with access control. In *Proc. of the 16th ACM Conference on Computer and Communications Security (CCS 2009)*, pages 131–140. ACM Press, 2009.
- [CDN09b] J. Camenisch, M. Dubovitskaya, and G. Neven. Oblivious transfer with access control. In *Proc. of the 16th ACM Conference on Computer and Communications Security (CCS 2009)*. ACM Press, 2009.
- [CDN10a] J. Camenisch, M. Dubovitskaya, and G. Neven. Unlinkable priced oblivious transfer with rechargeable wallets. In Radu Sion, editor, *Financial Cryptography*, volume 6052 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 2010.
- [CDN10b] J. Camenisch, M. Dubovitskaya, and G. Neven. Unlinkable priced oblivious transfer with rechargeable wallets. In Radu Sion, editor, *Financial Cryptography*, volume 6052 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 2010.
- [CDNZ11] J. Camenisch, M. Dubovitskaya, G. Neven, and G.M. Zaverucha. Oblivious transfer with hidden access control policies. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 192–209. Springer, 2011.
- [CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer Verlag, 1994.
- [CGH08] S. Coull, M. Green, and S. Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. Cryptology ePrint Archive, Report 2008/474, 2008. <http://eprint.iacr.org/>.
- [Cha85] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.
- [Cha88] D. L. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, January 1988.
- [CHKL11] J. Camenisch, K. Haralambiev, M. Kohlweiss, and J. Lapon. Structure preserving cca2 encryption and its application to oblivious third parties, 2011.

- [CHL⁺11] J. Camenisch, K. Haralambiev, A. Lysyanskaya, G. Neven, and V. Shoup. Memento in cryptography, 2011.
- [CKS10] J. Camenisch, M. Kohlweiss, and C. Soriente. Solving revocation with efficient update of anonymous credentials. In Juan A. Garay and Roberto De Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 454–471. Springer, 2010.
- [CKY09] J. Camenisch, A. Kiayias, and M. Yung. On the portability of generalized schnorr proofs. In Antoine Joux, editor, *EUROCRYPT*, pages 425–442, 2009.
- [CL01] J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer Verlag, 2001.
- [CL04] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew K. Franklin, editor, *Advances in Cryptology — CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer Verlag, 2004.
- [CM99] J. Camenisch and M. Michels. Proving in zero-knowledge that a number n is the product of two safe primes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 107–122. Springer Verlag, 1999.
- [CMS10] J. Camenisch, S. Mödersheim, and D. Sommer. A formal model of identity mixer. In Stefan Kowalewski and Marco Roveri, editors, *FMICS*, volume 6371 of *Lecture Notes in Computer Science*, pages 198–214. Springer, 2010.
- [CMW06] J. Crampton, K. Martin, and P. Wild. On key assignment for hierarchical access control. In *In Proc. of the 19th IEEE Computer Security Foundations Workshop (CSFW 2006)*, Venice, Italy, July 2006.
- [CNR11] J. Camenisch, G. Neven, and M. Rückert. Credentials from lattices, 2011.
- [CNS07a] J. Camenisch, G. Neven, and A. Shelat. Simulatable adaptive oblivious transfer. In Moni Naor, editor, *Eurocrypt 2007*, *Lecture Notes in Computer Science*, pages 573–590, 2007.
- [CNS07b] J. Camenisch, G. Neven, and A. Shelat. Simulatable adaptive oblivious transfer. In Moni Naor, editor, *Advances in Cryptology — EUROCRYPT 2007*, volume 4515, pages 573–590. Springer, 2007.
- [CP93] D. Chaum and T.P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer-Verlag, 1993.

- [CS97] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In Burt Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, volume 1296 of *Lecture Notes in Computer Science*, pages 410–424. Springer Verlag, 1997.
- [CS03] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144, 2003.
- [CW88] C. Camerer and K. Weigelt. Experimental tests of a sequential equilibrium reputation model. *Econometrica*, 56(1):1–36, 1988.
- [Das00] P. Dasgupta. Trust as a commodity. In *Trust: Making and Breaking Cooperative Relations*, 2000.
- [Del00] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proc. of the 2nd ACM Conference on Electronic Commerce (EC 2000)*, Minneapolis, MN, USA, October 2000.
- [DF10] S. De Capitani di Vimercati and S. Foresti. Privacy of outsourced data. In M. Bezzi, P. Duquenoy, S. Fischer-Huebner, M. Hansen, and G. Zhang, editors, *Privacy and Identity Management for Life*. Springer, 2010.
- [DFJ⁺08] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, G. Pelosi, and P. Samarati. Preserving confidentiality of security policies in data outsourcing. In *Proc. of the Workshop on Privacy in the Electronic Society (WPES 2008)*, Alexandria, VA, USA, October 2008.
- [DFJ⁺10a] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, G. Pelosi, and P. Samarati. Encryption-based policy enforcement for cloud storage. In *Proc. of the 1st ICDCS Workshop on Security and Privacy in Cloud Computing (SPCC 2010)*, Genova, Italy, June 2010.
- [DFJ⁺10b] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Encryption policies for regulating access to outsourced data. *ACM Transactions on Database Systems (TODS)*, 35(2):12:1–12:46, April 2010.
- [DFJ⁺10c] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragments and loose associations: Respecting privacy in data publishing. *Proc. of the VLDB Endowment*, 3(1):1370–1381, September 2010.
- [DFJ⁺11] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Authorization enforcement in distributed query evaluation. *Journal of Computer Security*, 2011.

- [DFL11] S. De Capitani di Vimercati, S. Foresti, and G. Livraga. Privacy in data publishing. In J. Garcia-Alfaro, G. Navarro-Arribas, A. Cavalli, and J. Leneutre, editors, *Data Privacy Management and Autonomous Spontaneous Security*, LNCS 6514. Springer, 2011. invited.
- [DFLS11] S. De Capitani di Vimercati, S. Foresti, G. Livraga, and P. Samarati. Anonymization of statistical data. *IT - Information Technology*, 53(1):18–25, January 2011.
- [DFP⁺11] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati. Efficient and private access to outsourced data. In *Proc. of the 31st International Conference on Distributed Computing Systems (ICDCS 2011)*, Minneapolis, Minnesota, USA, June 2011.
- [DFS11] S. De Capitani di Vimercati, S. Foresti, and P. Samarati. Protecting information privacy in the electronic society. In J. Filipe and M.S. Obaidat, editors, *e-Business and Telecommunications International Conference (ICETE 2009)*, volume 130. Springer, 2011.
- [DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.
- [DOR99] G. Di Crescenzo, R. Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional oblivious transfer and timed-release encryption. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT ’99*, volume 1592, pages 74–89. Springer, 1999.
- [DWS⁺11] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering Journal*, 16(1):3–32, 2011.
- [DY05] Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *Public Key Cryptography – PKC 2005*, volume 3386, pages 416–431. Springer, 2005.
- [ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO ’84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer Verlag, 1985.
- [Han05] H. Hanewinkel. PGP / GnuPG / OpenPGP message encryption in JavaScript, February 2005. <http://www.hanewin.net/encrypt/>.
- [Her08] J. Herranz. Restricted adaptive oblivious transfer. Cryptology ePrint Archive, Report 2008/182, 2008. <http://eprint.iacr.org/>.
- [HIM02] H. Hacigümüs, B. Iyer, and S. Mehrotra. Providing database as a service. In *Proc. of 18th IEEE International Conference on Data Engineering (ICDE 2002)*, San Jose, CA, USA, February 2002.

- [HIML02] H. Hacigümüs, B. Iyer, S. Mehrotra, and C. Li. Executing SQL over encrypted data in the database-service-provider model. In *Proc. of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2002)*, Madison, WI, USA, June 2002.
- [Jos06] S. Josefsson. The Base16, Base32, and Base64 Data Encodings. RFC 4648 (Proposed Standard), October 2006.
- [Kal00] B. Kaliski. PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898 (Informational), September 2000.
- [KB09] B. Kellermann and R. Böhme. Privacy-enhanced event scheduling. In *Proc. of the 2009 International Conference on Information Privacy, Security, Risk and Trust (PASSAT 2009)*, Vancouver, Canada, August 2009.
- [Kel10] B. Kellermann. Open research questions of privacy-enhanced event scheduling. In *Proc. of the 2010 IFIP WG 11.4 International Workshop (iNetSec 2010)*, Sofia, Bulgaria, March 2010.
- [Kel11] B. Kellermann. Privacy-enhanced web-based event scheduling with majority agreement. In *Proc. of 26th IFIP International Information Security Conference (IFIP SEC 2011)*, Lucerne, Switzerland, June 2011.
- [KL10a] A. Küpçü and A. Lysyanskaya. Optimistic fair exchange with multiple arbiters. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *ESORICS*, volume 6345 of *Lecture Notes in Computer Science*, pages 488–507. Springer, 2010.
- [KL10b] A. Küpçü and A. Lysyanskaya. Usable optimistic fair exchange. In Josef Pieprzyk, editor, *CT-RSA*, volume 5985 of *Lecture Notes in Computer Science*, pages 252–267. Springer, 2010.
- [Koo11a] B.J. Koops. Een herdenking van dataprotectie: datamist en beslissingstransparantie als alternatief. V. Frissen, L. Kool and M. van Lieshout (eds), *Jaarboek ICT and Samenleving: Privacy Revisited*, Delft, 2011.
- [Koo11b] B.J. Koops. Effecting decision transparency to achieve real data protection after the computational turn. *Privacy and Due Process after the Computational Turn*, 2011.
- [Koo11c] B.J. Koops. Forgetting footprints, seizing shadows. a critical analysis of a “right to be forgotten” in big data practice. SCRIPT-ed (submitted), 2011.
- [Koo11d] B.J. Koops. The (in)flexibility of techno-regulation and the case of purpose-binding. *Legisprudence*, special issue on techno-regulation (submitted), 2011.
- [KPS11] B. Kellermann, S. Pötzsch, and S. Steinbrecher. Privacy-respecting reputation for wiki users. In *Proc. of the 5th IFIP WG 11.11 International Conference on Trust Management (IFIPTM 2011)*, Copenhagen, Denmark, June 2011.

- [KSHW97] J. Kelsey, B. Schneier, C. Hall, and D. Wagner. Secure applications of low-entropy keys. In *Proc. of the 1st International Workshop on Information Security (ISW 1997)*, Ishikawa, Japan, September 1997.
- [LRSW99] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In Howard Heys and Carlisle Adams, editors, *Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*. Springer Verlag, 1999.
- [MT98] C. Meinel and T. Theobald. *Algorithms and Data Structures in VLSI Design*. Springer-Verlag, 1998.
- [MWP11] Y. De Mulder, K. Wouters, and B. Preneel. A Privacy-preserving ID-based Group Key Agreement Scheme applied in VPAN. In Juraž Hromkovic and Rastislav Královic, editors, *SOFSEM 2011: 37th Conference on Current Trends in Theory and Practice of Informatics*, volume 6543 of *Lecture Notes in Computer Science*, pages 214–222. Springer-Verlag, 2011.
- [PB10] S. Pötzsch and R. Böhme. The role of soft information in trust building: Evidence from online social lending. In *Proc. of the 3rd International Conference on Trust and Trustworthy Computing (TRUST 2010)*, Berlin, Germany, June 2010.
- [Pöt10] S. Pötzsch. Einfluss wahrgenommener Privatsphäre und Anonymität auf Forennutzer. In *Proc. of Mensch & Computer 2010*, Duisburg, Germany, September 2010.
- [Pri] PrimeLife. phpbb privacy-awareness support: Personal data mod. <http://www.primelife.eu/results/opensource/59-phpbb-pam>.
- [PRT04] E. Pavlov, J. S. Rosenschein, and Z. Topol. Supporting privacy in decentralized additive reputation systems. In *Proc. of the 2nd International Conference on Trust Management*, Oxford, United Kingdom, March 2004.
- [PWG10] S. Pötzsch, P. Wolkerstorfer, and C. Graf. Privacy-awareness information for web forums: results from an empirical study. In *Proc. of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI 2010)*, Reykjavik, Iceland, October 2010.
- [Rab81] M.O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.
- [RKP09] A. Rial, M. Kohlweiss, and B. Preneel. Universally composable adaptive priced oblivious transfer. In Hovav Shacham and Brent Waters, editors, *Proc. of the 3rd International Conference on Pairing-Based Cryptography*, page 24. Springer, 2009.
- [RP10] A. Rial and B. Preneel. Optimistic Fair Priced Oblivious Transfer. In *Progress in Cryptology - AFRICACRYPT 2010*, volume 6055 of *Lecture Notes in Computer Science*, pages 131–147. Springer-Verlag, 2010.

- [Sch91] C.P. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):239–252, 1991.
- [SCS10] S. Schiffner, S. Clauß, and S. Steinbrecher. Privacy and liveness for reputation systems. In *Proc. of the 6th European Workshop on Public Key Services, Applications and Infrastructures (EuroPKI 2009)*, Pisa, Italy, September 2010.
- [SCS11] S. Schiffner, S. Clauß, and S. Steinbrecher. Privacy, liveness and fairness for reputation. In *Proc. of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2011)*, Nový Smokovec, Slovakia, January 2011.
- [SD01] P. Samarati and S. De Capitani di Vimercati. Access control: Policies, models, and mechanisms. In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design*, pages 137–196. Springer-Verlag, London, 2001.
- [SD10] P. Samarati and S. De Capitani di Vimercati. Data protection in outsourcing scenarios: Issues and directions. In *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2010)*, Beijing, China, April 2010. invited paper.
- [SHB09] E. Stark, M. Hamburg, and D. Boneh. Symmetric cryptography in javascript. In *Proc. of the 2009 Computer Security Applications Conference (ACSAC 2009)*, Honolulu, HI, December 2009.
- [Ste08] S. Steinbrecher. Enhancing multilateral security in and by reputation systems. In *Proc. of the IFIP/FIDIS Internet Security and Privacy Summer School*, Brno, Czech Republic, September 2008.
- [Ste09] S. Steinbrecher. The need for interoperable reputation systems. In *Proc. of the 2009 IFIP WG 11.4 International Workshop (iNetSec 2009)*, Zurich, Switzerland, April 2009.
- [Tob03] C. Tobias. Practical oblivious transfer protocols. In Fabien A. P. Petitcolas, editor, *Proc. of the 5th International Workshop on Information Hiding (IH 2002)*, volume 2578, pages 415–426. Springer, 2003.