WEDNESDAY APRIL 7th

7:30	8:45		BREAKFAST
9:00	9:05	Organizers	Welcome
9:05	9:50	Gerome Miklau, Univ of Massachusetts, USA	Securely Managing Historical Retention in Database Systems
9:50	10:35	Kristen LeFevre, Univ of Michigan, USA	Privacy Wizards for Social Networking Sites
10:35	10:50		COFFEE
10:50	11:35	David Chadwick, Univ of Kent, UK	The distributed enforcement of sticky policies
11:35	12:20	Felix Klaedtke, ETH Zürich, CH	Monitoring Security Policies with Metric First-order Temporal Logic
12:20	14:00		LUNCH
14:00	14:45	Lujo Bauer, CMU, USA	Taking Authorization Logics Beyond Access Control
14:45	15:30	Daniel Trivellato, Univ. of Eindhoven, NL	Privacy-Aware Distributed Goal Evaluation
15:30	16:00		COFFEE
16:00	16:45	Frank McSherry, Microsoft Research, USA	Differential Privacy: Theory and Practice
16:45	17:15	Renato Iannella, NICTA, AU	Adding Privacy Support to a Rights Language (short talk)
17:15	18:00	Organizers	DISCUSSION / WRAP-UP DAY 1
18:00			DINNER

THURSDAY APRIL 8th

7:30	8:45		BREAKFAST
9:00	9:45	Ravi Sandhu, Univ. of Texas at St. Antonio, USA	Group-Centric Models for Secure and Agile Information Sharing
9:45	10:30	Fabio Massacci, Univ. of Trento, I	Infringement Management: or why Schneider, Hamlen, Bauer, and many others are all looking in the wrong place
10:30	10:45		COFFEE
10:45	11:30	Fabio Martinelli, CNRS Pisa, I	Risk-aware usage control
11:30	12:15	Alexander Pretschner, Fraunhofer IESE, D	Usage Control meets Information Flow
12:15	14:00		LUNCH
14:00	14:45	Ricardo Neisse, Fraunhofer IESE, D	Trust Management Models for Distributed Usage Control
14:45	15:30	Will Winsborough, Univ. of Texas at St. Antonio, USA	Methodological considerations for the design of group-based secure information sharing systems
15:30	16:00		COFFEE
16:00	16:45	Stephan Micklitz, Google, D	Privacy as a product - Transparency, Control and the Google Dashboard
16:45	17:15	Rafael Accorsi, Univ. of Freiburg, D	FORTES: Forensic Information Flow Analysis of Business Processes (short talk)
17:15	18:00	Organizers	DISCUSSION / WRAP-UP DAY 2
18:00			DINNER

FRIDAY APRIL 9TH

7:30	8:45		BREAKFAST
9:00	9:45	Christian Schaefer, Docomo Euro-Labs, D	Privacy in Ubiquitous Mobile Services
9:45	10:30	Frank Piessens, KU Leuven, B	Noninterference Through Secure Multi-Execution
10:30	11:15	Bruno Pontes Soares Rocha, Univ. of Eindhoven, NL	Towards Static Flow-based Declassification for Legacy and Untrusted Programs
11:15	12:15	Organizers	DISCUSSION / WRAP-UP DAY 1-3
12:15	14:00		LUNCH

Wednesday

Gerome Miklau: Securely Managing Historical Retention in Database Systems

Databases that preserve a historical record of their operations and data offer the important benefit of system accountability: past events can be analyzed to detect breaches and maintain data quality. But the retention of history can also pose a threat to privacy. System designers must carefully balance the need for privacy and accountability by controlling how and when data is retained by the system, and who will be able to recover it.

This talk addresses two challenges to securely managing historical data. First, databases don't easily "forget". Even when users intend to remove sensitive information, it may persist against their will. We report experimental results on this unintended retention and propose solutions. Second, while databases can "remember", they don't remember safely. Few tools exists for managing an intended audit record in a manner consistent with retention policies. We describe a framework in which audit logs can be protected and audit queries are answered approximately.

Kristen LeFevre: Privacy Wizards for Social Networking Sites

Privacy is an enormous problem in online social networking sites.

While sites such as Facebook allow users fine-grained control over who can see their profiles, it is difficult for average users to specify this kind of detailed policy. For this reason, we propose a template for the design of a social networking privacy wizard. The intuition for the design comes from the observation that real users conceive their privacy preferences (which friends should be able to see which information) based on an implicit set of rules. Thus, with a limited amount of user input, it is usually possible to build a machine learning model that concisely describes a particular user's preferences, and then use this model to configure the user's privacy settings automatically. As an instance of this general framework, we have built a wizard based on an active learning paradigm called uncertainty sampling. The wizard iteratively asks the user to assign privacy ``labels" to selected (``informative") friends, and it uses this input to construct a classifier, which can in turn be used to automatically assign privileges to the rest of the user's (unlabeled) friends.

To evaluate our approach, we collected detailed privacy preference data from 45 real Facebook users. Our study revealed two important things. First, real users tend to conceive their privacy preferences in terms of {communities, which can easily be extracted from a social network graph using existing techniques. Second, our active learning wizard, using communities as features, is able to recommend high- accuracy privacy settings using less user input than existing policy-specification tools.

David Chadwick: The distributed enforcement of sticky policies

We propose a system in which a user's personal policy is submitted along with their personal identifying information (PII), and this policy is thereafter stuck to their PII whilst it is used by the original organisation and any other organisations that it is subsequently transferred to. The handling of the sticky policy and the policy enforcement is taken care of by an application independent authorisation infrastructure, so that existing applications need to make very few changes to their existing code in order to support distributed sticky policy enforcement.

Felix Klaedtke: Monitoring Security Policies with Metric First-order Temporal Logic

We show the practical feasibility of monitoring complex security properties using a runtime monitoring approach for metric first-order temporal logic. In particular, we show how a wide variety of security policies can be naturally formalized in this expressive logic, ranging from traditional policies like Chinese wall and separation of duty to more specialized usage-control and compliance

requirements. We also explain how these formalizations can be directly used for monitoring and experimentally evaluate the performance of the resulting monitors.
Joint work with David Basin and Samuel Mueller.

Lujo Bauer: Taking Authorization Logics Beyond Access Control
Authorization logics have proven to be a useful tool for modeling and building distributed access-control systems. In this talk, I will discuss some advances in authorization logics that move this approach towards being able to reason about and implement usage-control scenarios.
First, I will present a method for implementing consumable credentials using authorization logic. Such credentials convey use-limited authority (e.g., to open a door once) or authority to utilize resources that are themselves limited (e.g., concert tickets). I will briefly describe design and implementation of mechanisms to enforce the consumption of credentials in a distributed system, and to protect credentials from nonproductive consumption as might result from misbehavior or failure.
Second, I will briefly survey several other recent advancements in authorization logics that make them more suitable for use in real-world usage-control scenarios. These include a method for significantly reducing the latencies and overhead typically associated with formal proofs of access, and extensions to authorization logics to support DRM-style control over the usage of digital content.

Daniel Trivellato: Privacy-Aware Distributed Goal Evaluation
In this talk, we present a distributed goal evaluation algorithm based on a foundation language for modern Trust Management (TM) systems. The algorithm solves two problems faced in distributed implementations: first, it detects when the computation has terminated (all answers are collected) in a completely distributed way. Second, compared to other distributed goal evaluation methods, it reduces network traffic by allowing principals to delay the response until a “maximal” set of answers has been computed, without running the risk of causing a deadlock.
We also show that existing TM systems require principals to exchange much more information than actually needed, either because they rely on a centralized goal evaluation algorithm or because they require principals to reveal part of their intensional policy. The proposed algorithm seeks to avoid unnecessary disclosure of policy information, forming the basis of a privacy-friendly TM system.

Frank McSherry: Differential Privacy: Theory and Practice
We present an introduction to the recent concept of ``Differential Privacy", a privacy criterion requiring that a computation not reveal the presence or absence of individual records in an input data set. After developing the mathematical foundation, we proceed to describe the ``Privacy Integrated Queries" platform, an analysis language and system providing differential privacy guarantees even for users without privacy experience. The platform requires some new mathematics, tasteful language restriction, and careful implementation, but enables a large set of new computations that would otherwise require ad-hoc expert analysis before execution against sensitive data.

Renato Iannella: Adding Privacy Support to a Rights Language
Abstract TBA – short talk

Thursday

Ravi Sandhu: Group-Centric Models for Secure and Agile Information Sharing

To share information and retain control (share-but-protect) is a classic cyber security problem for which effective solutions continue to be elusive.

Where the patterns of sharing are well defined and slow to change it is reasonable to apply the traditional access control models of lattice-based, role-based and attribute-based access control, along with discretionary authorization for further fine-grained control as required. Proprietary and standard rights markup languages have been developed to control what a legitimate recipient can do with the received information including control over its further discretionary dissemination. This dissemination-centric approach offers considerable flexibility in terms of controlling a particular information object with respect to already defined attributes of users, subjects and objects. However, it has many of the same or similar problems that discretionary access control manifests relative to role-based access control. In particular specifying information sharing patterns beyond those supported by currently defined authorization attributes is cumbersome or infeasible. Recently a novel mode of information sharing called group-centric was introduced by me and my colleagues. Group-centric secure information sharing (g-SIS) is designed to be agile and accommodate ad hoc patterns of information sharing. In this lecture we review g-SIS models, discuss their relationship with traditional access control models and demonstrate their agility relative to these.

Fabio Massacci: Infringement Management: or why Schneider, Hamlen, Bauer, Ligatti, Walker, and many others are all looking in the wrong place.

There are many papers around on how to enforce security policies by means of security, insertion, suppression, edit, [a-z]*-automata. They are based on two simple ideas: the enforcement mechanism should leave good executions unchanged (transparency) and Bad execution should be made good (soundness).

Practical systems deal with bad and good traces in *some* ways but the theoretical models have so far concentrated only on the characterization of good traces for this or that enforcement mechanism.

Unfortunately, from a practical perspective, these are the most uninteresting properties ever. What makes a difference in practice is not what the enforcement mechanism does when the doctors follows the hospital's policy (because nothing should be done!) but when they start to misbehave. But we have no theory for it!

This problem is a source of huge paradoxes. A black-hole system (silently ignoring the doctor's requests until he remember to insert the clinical trial number in a past prescription) or the heads-up system (notify the head nurse that they forgot to insert it) cannot be formally distinguished. Even the very running example of Bauer, Ligatti and Walker's IJIS paper cannot be geneerated following the theoretical construction (of IJIS or TISSEC papers).

I will try in this talk provide some foundation for more practical enforcement theories based on the notions of infringement management with veenial and amendable errors.

Fabio Martinelli: Risk-aware usage control

We started a research line devoted to consider the notion of risk in the Usage Control (UCON). This can be considered from several perspectives.

In particular, UCON is based on the idea that attributes required for decision-making can be changed over a period of usage. It is not always possible to get a fresh and trustworthy value of attributes, and a decision has to be done with some uncertainties in mind. Our study concerns the risks associated with imperfect mechanisms collecting information about an authorization context. We use Continuous-time Markov Chains to solve the problem.

The other direction is to help a client to assess the riskiness of data processing while client's data is under control of a service provider in Service Oriented Architecture (SOA). We propose a method to empower usage control decision making process for more efficient and flexible control of access to data. We show how to select a service provider using risk, re-evaluate the risk level when some changes have happened and how to improve an infrastructure in order to reduce the risk level. Qualitative risk assessment is used to tackle the problem.

Alexander Pretschner: Usage Control meets Information Flow

Major technologies for usage control enforcement include runtime verification and complex event processing. The universe of discourse of respective policies is that of events. If usage control is defined as access control that extends to the future and, in a distributed setting, is defined to concern the usage of data after giving it away, there is a mismatch: Policies primarily pertain to events while usage control requirements primarily pertain to data. Our work aims at making usage control more data-centric: rather than stipulating that file song1.mp3 must not be played after 31st of December, want to specify (and enforce) that the song (the data) that corresponds to (the file) song1.mp3 must not be played after 12-31. In particular, all copies of file song1.mp3 must not be played after that date. In order to enforce this kind of requirements, we hence need to track copies of file song1.mp3 at and across different layers of abstraction (e.g., operating system, runtime system, window system, sound hardware). This can be achieved by data flow tracking systems.
In this talk, we show how to marry usage control to information flow and present a set of fundamental technical and conceptual challenges.
Joint work with Enrico Lovat, Matus Harvan, Christian Schaefer, Thomas Walter

Ricardo Neisse: Trust Management Models for Distributed Usage Control

Distributed usage control is concerned with the governance of who is authorized to access a piece of data and rights and duties that must be enforced after data is released. When enforcing distributed usage control policies, different trust aspects should be considered, for example, the integrity of system components that enforce usage control policies and the identities of the parties that exchange data. In this talk I will introduce our work on trust management for distributed usage control and the future research directions we are pursuing. Our work on trust management builds up on trust guarantees realized using Trusted Computing Platform and integrates with other trust issues considering the goals and dependencies of the different parties that interact and exchange data. The initial focus of our work is on trust issues related to distributed usage control in service oriented and cloud computing architectures. Our long term objective is to specify an abstract trust management model and identify trust management patterns to support decision and risk analysis considering data usage control in a distributed computing scenario.

Will Winsborough: Methodological considerations for the design of group-based secure information sharing systems

This talk considers methodological issues with application to the design of group-based secure information sharing models. Our approach combines several design steps, each intended to focus the attention of the designer on different aspects of the larger system. The basic approach is to use temporal logic to specify system behavior. This enables us to focus on basic group operations--a user joining a group, user leaving, object being added to a group, object removal from a group, group creation and destruction, subject creation and destruction. Administrative policies will be needed to govern some of these operations. The fundamental principle is that users gain access to objects by virture of belonging to groups in which the object is present. Several variations of join, leave, add, and remove have been studied, and various inter-group relationships are currently under investigation. The key system behavior is that of authorization for a user or his subject to be

granted access to an object. Behavior of users and administrators is given by the join, leave, add and remove events.
Methodologically, the first steps involve the simultaneous design of core properties that focus on one aspect of system behavior at a time. This provides a sanity check that enables the designer to consider each facet of the system in isolation. The second codifies these properties into a single specification of the circumstances in which subject creation, object read and write are authorized, based directly on past join, leave, add, remove, object and subject creation events. Core properties are verified to be satisfied by this (stateless) specification. Working on a refinement basis, the third develops (stateful) data structures that verifiably support identical authorization decisions as the stateless specification. The fourth considers issues arising due to distributed implementation and non-instantaneous communication. These considerations prevent developing a specification that is a pure refinement of the stateful specification. Notions of stale-safety are introduced, and alternative enforcement-level policy options are discussed.

Stephan Micklitz: Privacy as a product - Transparency, Control and the Google Dashboard
Abstract TBA

Rafael Accorsi: FORTES: Forensic Information Flow Analysis of Business Processes
Nearly 70% of all business processes in use today rely on automated workflow systems for their execution. Despite the growing expenses in the design of advanced tools for secure and compliant deployment of workflows, an exponential growth of dependability incidents persists.
Concepts beyond access control focusing on information flow control offer new paradigms to design security mechanisms for reliable and secure IT-based workflows.
This talk presents FORTES, an approach for the forensic analysis of information flow properties. FORTES claims that information flow control can be made usable as a core of an audit-control system. For this purpose, it reconstructs workflow models from secure log files (i.e. execution traces) and, applying security policies, analyzes the information flows to distinguish security relevant from security irrelevant information flows. FORTES thus cannot prevent security policy violations, but by detecting them with well-founded analysis, improve the precision of audit controls and the generated certificates.
Short talk

Friday
Christian Schaefer: Privacy in Ubiquitous Mobile Services
The scenario considered is future mobile networks where it is envisioned that many more enhanced services will be available for mobile users than today and in a functionally much wider scope. The services are composed by service enablers which provide some specific functionality. Service enablers by third parties might also be used to provide some specific service. This means that the third party might need some private data to provide the required results. A user giving away his private data might wish to have some means to control how his private data will be used.
The goal of the proposed work is to provide an environment where users can distribute private data and be sure that these data is handled according to some privacy policy they have specified. The presentation will cover the necessary areas of work one has to deal with to solve this problem and concentrate on the planned work items.
To provide an environment where private data is protected it is envisioned to use former work on distributed usage control as basis. Thus it is necessary to develop an enforcement infrastructure that is able to provide the desired protection for the user's private data. Some of the topics that need to be considered are the enforcement architecture, identity management, data/policy management,

protocols for data/policy exchange (also within the enforcement architecture if you consider a distributed infrastructure), some kind of "specification language" for enforcement mechanisms and other topics.

Frank Piessens: Noninterference Through Secure Multi-Execution
A program is defined to be noninterferent if its outputs cannot be influenced by inputs at a higher security level than their own. Various researchers have demonstrated how this property (or closely related properties) can be achieved through information flow analysis, using either a static analysis (with a type system or otherwise), or using a dynamic monitoring system. We propose an alternative approach, based on a technique we call secure multi-execution. The main idea is to execute a program multiple times, once for each security level, using special rules for I/O operations. Outputs are only produced in the execution linked to their security level. Inputs are replaced by default inputs except in executions linked to their security level or higher. Input side effects are supported by making higher-security-level executions reuse inputs obtained in lower-security-level threads.

We show that this approach is interesting from both a theoretical and practical viewpoint. Theoretically, we prove for a simple deterministic language with I/O operations, that this approach guarantees complete soundness (even for the timing and termination covert channels), as well as good precision (identical I/O for terminating runs of termination-sensitively noninterferent programs). On the practical side, we present an experiment implementing secure multi-execution in the mainstream Spidermonkey Javascript engine, exploiting parallelism on a current multi-core computer. Benchmark results of execution time and memory for the Google Chrome v8 Benchmark suite show that the approach is practical for a mainstream browser setting. Certain programs are even executed faster under secure multi-execution than under the standard execution.

We discuss challenges and propose possible solutions for implementing the technique in a real browser, in particular handling the DOM tree and browser callback functions. Finally, we discuss how secure multi-execution can be extended to handle language features like exceptions, concurrency or nondeterminism.

Bruno Pontes Soares Rocha: Towards Static Flow-based Declassification for Legacy and Untrusted Programs
Simple non-interference is too restrictive for specifying and enforcing information flow policies in most programs. Exceptions to non-interference are provided using declassification policies. Several approaches for enforcing declassification have been proposed in the literature. In most of these approaches, the declassification policies are embedded in the program itself or heavily tied to the variables in the program being analyzed, thereby providing little separation between the code and the policy.

Consequently, the previous approaches essentially require that the code be trusted, since to trust that the correct policy is being enforced, we need to trust the source code.

In this paper, we propose a novel framework in which declassification policies are related to the source code being analyzed via its I/O channels. The framework supports many of the of declassification policies identified in the literature. Based on flow-based static analysis, it represents a first step towards a new approach that can be applied to untrusted and legacy source code to automatically verify that the analyzed program complies with the specified declassification policies. The analysis works by constructing a conservative approximation of expressions over input channel values that could be output by the program, and by determining whether all such expressions satisfy the declassification requirements stated in the policy.

We introduce a representation of such expressions that resembles tree automata. We prove that if a program is considered safe according to our analysis then it satisfies a property we call Policy Controlled Release, which formalizes information-flow correctness according to our notion of declassification policy. We demonstrate, through examples, that our approach works for several interesting and useful declassification policies, including one involving declassification of the average of several confidential values.