Applied Cryptography

Privacy Enhanced Access Control

Made by
Lotte Hauge, s113217
Daniel Handler, s113446
Lasse Dessau, s142980

Technical University of Denmark
DTU Compute
02232 Applied Cryptography
November 9, 2015

Abstract

German researchers have found in a study that social equality in relationships between student and teacher influences the grading of students. This report will look at some solutions for avoiding this grading bias when students use the *CampusNet*-service at the Technical University of Denmark to hand in assignments and exams.
The report models the users of *CampusNet* into four classes. For each of these classes privacy issues and constraints are detected and a threat model is identified. To find a solution we briefly discusses open and closed loop authentication, and conclude that an open loop authentication method is required.
We look at two major privacy enhanced access control frameworks: Microsoft's U-Prove and IBM's Idemix. We address these framework's advantages and disadvantages especially when it comes to computationally advantages in the protocol for accessing resources. We present a solution within our problem scope using Idemix and six use cases distributed over the classes. The report verifies the Idemix solution.
We conclude that the solution is valid under the specified threat model and requirements, and then future works that can later extend the scope of this report is mentioned.

Contents

1 Introduction

1.1 Objective

German research from 2015 indicates a bias in grading depending on the social equality in a teacher/student relationships [21]. In addition previous research by University of Georgia shows a gender bias in grading [11]. In both cases bias it not necessarily a continuous decision by teachers. While these studies only surveys students until the 8th grade, it is not a big conceptional leap to hypothesis that such bias can also be found in student/professor relationships at universities.

In this report the course platform *CampusNet* for the Technical University of Denmark (DTU) is examined with the aim of uncovering cases where bias may be introduced during assessment of work by students. Furthermore, this report will include the design of a solution that mitigates or prevent this bias, such that students will be fairly assessed solely on their course relevant knowledge and skills. In particular we will address:

1. Which privacy issues on *CampusNet* can result in unfair grading?
2. Can the privacy issues be classified and used to build a threat model for different *CampusNet* resources?
3. Can sets of privacy constraints be derived such that any solution which satisfies all the constraints will protect students from biased grading?
4. What sort of technologies can be used in a solution and which one works best protecting against the modelled threats?
5. Is open loop or closed loop authentication the best approach?
6. Is there a solution that protects against all the modelled threats?

1.2 Method

The first part of the report identifies privacy issues with the current usage of *CampusNet* as a grading platform, and model these into a threat model and verification requirements. In order to do so we introduce some terminology for this report in section 2. Then we address some privacy issues on *CampusNet* in section 3. These privacy issues will be used to build a model of the grading platform in section 4 which take the identified privacy issues as well as functionality and constraints into account. This model will focus on a small number of selected use cases defined in section 4.1. The model will then be further extended to include a threat model in section 4.2. Finally a solution verification on the requirements will be specified in section 4.3.

The second part of the report will attempt to find a solution that protects the user against the threats identified in part one. First open loop and

closed loop authentication techniques will be examined in section 5. Finally multiple techniques for privacy enhanced access control will then be studied in section 7 and 6 and discussed in section 8 in order to find a suitable security approach.

Once a good approach has been found and selected a solution will be defined and presented in section 9. This solution should satisfy the model specified in section 4.2 from the first part of this report. Using the approach specified in the verification requirements of the model from section 4.3 the solution will be verified in section 10 with focus on the use cases specified in the model description section 4.1.

In the end we conclude the report in section 11, and state what future work that needs to be done in order to use this solution in real life in section 12.

1.3 Scope

In order to deliver a solution within the given timeframe, we have scoped the report to deal solely with privacy issues between student and teacher. A consequence of this is that this report will not deal with privacy issues arising from using traceable IP addresses. Instead the report focuses on authentication and access control under the assumption of an unlinkable and untraceable anonymous channel.

Additionally, the report does not consider attacks on availability such as denial of service-attacks (DoS or DDos), nor circumvention of the authentication part of *CampusNet*. Also communication is assumed to maintain integrity, and cryptographic primitives are assumed to be unbreakable.

While bias can be found against all actors involved with DTU. The project focuses solely on *CampusNet* and bias against students. However, the approach taken by this report as well as the found solution may be applicable with minor modifications to other groups and platforms under DTU.

We do not take into account that students on other channels (e.g. oral) can tell the professor which assignment they wrote. That is, the solution will neither take receipt-freeness into account nor coercion-resistance [14]. It is assumed to be in the students interest to remain anonymous. It is also assumed to be in the DTU administrations interest that grading is done fairly.

The discussion in this report considers the two major privacy enhancing authorization systems U-Prove and Idemix [1]. A more simple token based system based on protocols like Kerberos [18] could have been developed, but since we found no studies with experience using such an approach, we have not considered it further. There are other similar frameworks [12] [10] who have been developed simultaneously with Idemix and U-Prove, but due to the scope of this report, these systems will not be taken into account.

There has to be a way of authenticating new *user*s not already in the system. This is not the focus in the report, however we will now provide a short solution. A secure way of identifying the users is to require the new students to personally show up at the DTU administration like they have done in a similar access control study in Greece [20]. Logistical this is a almost impossible solution, due to the fact that more than 1000 students are enrolled on DTU each year. Another way of establishing a secure channel is a one-time token, sent with the welcoming letter, when students are accepted at DTU. Finally, and in our opinion the best and easiest way, is to use nemID to identifying students.

2 Terminology

The following section describes the methodology used in this report. Figure 1 shows the actors defined by the ABC4Trust framework [2] [6] [13], which can be mapped to equivalent actors defined by the different privacy preserving attribute based access control systems examined in this report.

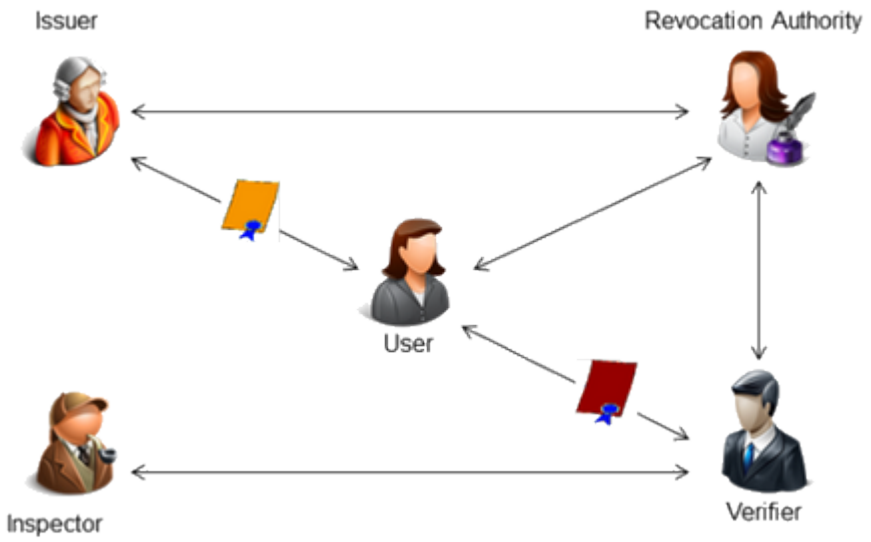

Figure 1: Actors in ABC4Trust framework
Figure is taken from D8.12 Architecture for Standardization V2 page 9

***CampusNet*:** A platform that allow students access to their mails, grades, course registration, course e-rooms (lecture notes, discussion forum and hand-ins) and course evaluations. It also allows access to other restricted resources such as download of licensed software and the platform is hosted by DTU at http://campusnet.dtu.dk. Finally, it allows professors to create

hand-ins for assignments and exams specific to a single course. The student is then expected answer the assignment by uploading a document, which is subsequently assessed by the professor.

DTU administration: The central authority responsible for managing student enrolments and collecting grades given by professors to students in different courses.

***Attribute*:** A certain property of the *user* that needs to be verified. In our case this could be the *user*'s enrolment in a course.

***user*:** The actor who's attributes must be authenticated. In our case this would be a student visiting a DTU resource on *CampusNet*.

***issuer*:** The actor issuing credentials to the *user*. The *issuer* certifies attributes about the *user* and thus must have knowledge about the user's identity. In our case the *issuer* is the system administration, who can identify and verify all students at DTU and knows the access policies for students.

***verifier*:** The actor verifying the *user* in order to authorize the *user* to access a resource. In our case the *verifier* is *CampusNet*, which should authorizes the *user* to access the resources it hosts.

***inspector*:** The actor confirming that the given credentials are valid, and can associate credentials with a certain *user*. The inspector is trusted by the *verifier*. In our case, the inspector will be a part of the DTU administration, so that they will be able e.g. to associate hand-ins with a given student.

***revocation authority*:** If tokens are long-lived it may be necessary to revoke them when circumstances change e.g. when a student is expelled from a course.

***Pseudonym*:** An anonymous identity belonging to a user. Possession of this identity is required to prove authenticity of requests to the *verifier*.

***Token*:** A set of attributes and a pseudonym which if sent to a *verifier* can grant access.

***Certificate*:** A set of attributes verified by an *issuer* for a given *user*.

***Anonymous Channel*:** Allow communication between two parties in a way where neither party can trace the other e.g. using IP-information. For many applications it is sufficient to only preserve anonymity one-way (e.g. only student are anonymous while *CampusNet* is public).

***Untraceability*:** The inability for an inspector to unmask the identity of the *user*. However, it can for some applications be necessary to be able to trace a *user*s real identity. This is not a problem if the *inspector* is a trusted party.

***Unlinkability*:** The inability to tell whether two tokens belong to the same *user*.

***Unfair grading*:** Grading that in any way *not* only is based on the students (and hence the hand-ins) academic level. E.g. grading biased by social status, physical attributes or relationships.

3 Privacy issues

In this section, identified privacy issues that may influence grading when handing in assignments or exams on *CampusNet* are discussed. The goal is not to provide maximum degree of privacy, but rather to determine which information should be kept private so that hand-ins will be graded solely on their own merits. Determining the least degree of privacy required to solve the unfair grading problem [11], will make it easier to find a valid solution that works for *CampusNet*.

3.1 Hand-ins

A professor can create a hand-in for an assignment or exam on the *CampusNet* platform. Students then currently login with a student number tied publicly to their real-life identity. This is a problem since personal knowledge about physical attributes or personal relationships can influence the grade given to a student [21] [11]. This is a privacy issue that must be solved by some degree of anonymous hand-ins. This problem has been solved before e.g. by hand-written exams using an exam number only known to the student and the university administration.

It is important to note that using the same pseudonym for multiple assignments is not good enough as each piece of work should be graded as a single entity and not influenced by the academic level of previous assignments.

3.2 Course participation statistics

When grading an exam a professor may be biased by a students general engagement in the course. We identified one *CampusNet* activity that can indicate a greater level of attentiveness of a student. This activity is downloading of course materials as they are published by a professor. Since an exam should be graded on it's own merits this activity must not influence grading and thus the students should be protected against *CampusNet* recording and associating their profile with such statistical data.

3.3 Other *CampusNet* activities

Other activities through DTU website and *CampusNet* by extension include downloading of study hand books, looking up contacts, and displaying course descriptions. Since these does not have anything to do with the course directly any bias on grading based on such statistical data should be insignificant. So we conclude that for a solution to sufficiently solve the unfair grading problem it does not need to mask such activities.

4 Classification and modelling of the problem

The purpose of this section is to create a model that in more detail considers the privacy issues identified in section 3, and to then make a classification of the accessing roles in order to model what access should be necessary for the different *CampusNet*resources. The privacy issues are described along with identified constraints that specify requirements to the solution.

4.1 Access classes of the model

We have identified the following four different access classes of the *CampusNet* resources:

World which are resources available to everyone;

DTU which are resources available to all affiliates of the Technical University of Denmark;

Course which deals with resources available to students following a particular course; and

Student which deals with personal data and hand-ins;

4.1.1 World

This group contains all users. Unless further authorization is provided these users should not be able to *CampusNet*, only the DTU homepage where

information about the university and its courses should be available. Since no authentication for these resources are required and our threat model described in 4.2 assumes an anonymous communication channel there are no privacy violations with respect to the threat model that can occur.

Use case World 1.: Looking up two course descriptions

Description A student views two course descriptions on publicly available DTU website.

Constraints None.

Privacy N/A.

4.1.2 DTU

All the students on DTU have some common things they need to access such as the phone book and the study hand book. Statistical information about contact lookups are unlikely to affect grading so no privacy properties are required.

Use case DTU 1.: Looking up two contacts

Description A student make a look up in the telephone book for two fellow students.

Constraints Student at DTU.

Privacy N/A.

4.1.3 Course

The students attending a given course must to be able to download materials uploaded by the professor. As discussed in section 3 if a download history can be associated with a student then these statistics may influence grading.

Use case course 1.: Downloading of two documents

Description A student downloads slides from the first lecture and then also downloads the work sheet for the first exercise session of the course.

Constraints Enrolled in course.

Privacy Professor may not be able to unmask the students identity. Individual downloads must not be linkable.

4.1.4 Student

The individual student should be capable of handing in exams and assignments during the year. When grading assignments the professor must be unable to identify which student a particular hand-in belongs to. This anonymous grading will force professors to give grades based solely on the academical level of the hand-in itself.

However, to ensure accountability, i.e. the student cannot refuse to acknowledge authorship of a hand-in, the administration must be able to unmask the identity of the student who delivered it, in order to deliver the grade to the correct student.

Finally, exams may require that a student has passed certain number of mandatory assignments in order to qualify. In this case each assignment and exams should be graded objectively on their own merits, and in particular not be influenced by the grade given to previous assignments by the same student. Thus assignments and exams must be unlinkable with each other. This ensures that a professor cannot tell whether a student almost failed qualification or have delivered every assignment with brilliance.

We have selected to focus on the following two use cases for the student class which must pass verification for our solution to be valid. For each use case relevant constraints and privacy properties have been identified all of which must be satisfied by our solution.

Use case student 1.: Assignment hand-in

Description A student delivers an assignment on *CampusNet*. The assignment is then graded by a professor.

Constraints Enrolled in course.

Privacy Professor may not be able to unmask the students identity. But it must be possible by a trusted third party to trace the assignment back to the student after a grade has been given.

Use case student 2.: Exam hand-in

Description A student delivers an exam on *CampusNet*. The exam is then graded by a professor.

Constraints Enrolled in course and qualified for the exam by having handed in mandatory assignments.

Privacy Professor may not be able to unmask the students identity. But it must be possible by the DTU administration to trace the exam back to the student after a grade has been given. Furthermore mandatory assignments and the exam should be unlinkable.

4.1.5 Interclass interaction

To examine actions over multiple classes during the same *CampusNet*-session we look at the following use case. Ideally, the user should have the same fluid *user* experience as today even with the privacy-enhanced authentication solution.

Combined use case

Description A student checks a course description and make a look up in the phone book to find the email address of the course responsible. Then the student delivers an assignment on *CampusNet* and download the next exercise sheet. The assignment is then graded by a professor.

Constraints Enrolled in course and enrolled at DTU.

Privacy Professor may not be able to unmask the students identity. But it must be possible by a trusted third party to trace the assignment back to the student after a grade has been given.

4.1.6 Revocation of class membership

There may be instances where it would be beneficial to support revocating a user of a class membership before it naturally expires, e.g. a student cancels a course or is expelled from the institution. This could in some cases be preferable, but according to our model this will not be a requirement for a solution.

4.2 Threat model

The following subsections describe the considered actors, attacks, and summerize the threats detailed per class in section 4.1.

4.2.1 Actors

In our model, the DTU administration is a trusted party and *CampusNet*, from which professors access hand-ins and statistics, is untrusted. The third party is the student *user*, which must be allowed to hand-in assignments and exams knowing that professors cannot disclose their identity. The professors acting on *CampusNet*s behalf are thus the attackers.

4.2.2 Attacks

The model only considers identity disclosing attacks that can affect grading, and only content of exhanges are examined. All communication is assumed to preserve integrety and exchanged messages are sent using anonymous channels.

4.2.3 Threats

Table 1 sumerizes the access classes, threats, and activities that are considered in this report.

Class	Threats	Available activities	
		Actors	**Actions**
World	None	All	Lookup course descriptions
DTU	None	Student	Lookup Contacts & Study handbook
Course	Biased grading based on: • Downloads history (lecture notes, slides, etc) .	Course participating student	Download course material
		Professor	Publishes course material.
Student	Biased grading based on: • Relationships • Physical attributes • Earlier assignments	Administration	Access to graded assignments/exams
		Student	Hand in assignments and exams.
		Professor	Grade hand-ins.

Table 1: Classes, threats and activities

4.3 Verification requirements

The goal of a solution that adhere to this model is the specification of a system which allow for fair grading under the threat model while also satisfying the identified constraints required for the grading platform to work. In particular, each class of *CampusNet*s resources has at least one associated use case for which privacy and constraints should be verified.

5 Open loop and closed loop authentication

To be able to find an optimal solution to the problems, we will look at two authentication approaches, to decrease the number of possible solutions.

5.1 Closed loop

In closed loop authentication, the party that authenticates the *user*, is the party where the *user*s credentials are issued or registered [9] That is the *issuer* and the *verifier* are the same entity. Closed loop authentication is widely used. e.g. on websites with a user name/password login. In closed loop authentication both parties need to verify each other and hereby usually need to know each others identity, say *issuer* and *verifier* is the same entity. The user controls a registered end-point of communication. This communication could be email address or similar. A challenge is sent to this end point, and a response demonstrates control over end-point [17]. At *CampusNet* if a student would need to prove his enrolment at DTU, the student would need to disclose her full identity in order to prove this, since her identity is the only thing linking her to DTU.

5.1.1 Cryptography

Due to the somehow fairly simple nature of the closed loop systems, the requirements for the cryptography ensuring the integrity between the parties are equally simple. When two parties are communicating, we can handle the integrity with a standard public key encryption [9].

5.2 Open loop

In the open loop authentication, the *issuer* and the *verifier* are not the same party [24]. That is, we have a third party, that authenticates the *user*, based on some attribute(s) [9]. This is often more costly to enrol a third party, however, from at privacy point of view, the open loop authentication is a more optimal solution, if some personal attributes must be verified. If we use a third, relying party to authenticate the *user*, e.g. using Idemix or a U-prove token, the clients identity can be kept secret, and only the attribute needed for authorization is shown to the service [9]. Open loop authentication is often used when a real world identity has to be linked to a system identity. In our *CampusNet* example where a student needs to prove his enrolment at DTU, the student can prove this without disclosing her full identity. Thus the aim is to verify the real world identity so we can map it to the system identity. This can be done by two-factor authentication, which e.g. is used in the danish NemID[1] [17] [15].

5.2.1 Cryptography

The open loop authentication methods requires a more advanced approach to secure integrity and confidentiality. We have communication between several different parties, and the *issuer* and the *verifier* has to agree on how

[1] www.nemid.nu

to ensure that the certificates and attributes hasn't be tampered with, by a *user* tampering with his own certificate or a man-in-the-middle.

5.3 Combined use

It is important to note that there is a difference in how systems are used. If a browser uses a third party to check certificates the browser is in practise using open loop authentication. This is commonly done to establish a secure channel to do closed loop authentication. We will in this report address the authentication used with the end point as the used terminology. As an example, when an open loop authentication is used to establish a secure channel to make a closed loop authentication, this will be addressed as a closed loop authentication.

5.4 Example protocols

As stated above, most of the common used protocols today is closed loop. Due to the fact that open loop authentication still is a rather new method, the number of existing systems are limited. However, Microsoft and IBM have each developed a system that uses open loop methods. Microsofts system is called U-Prove[19] and IBM has developed Idemix[3]. These systems will be analysed futher, in the following sections.

6 U-Prove

The U-Prove presentation protocol allows *user*s (in U-Prove provers) to confirm personal attributes to a *verifier* without disclosing their identity. Attributes for a *user*s are stored in tokens which are requested by the user from an *issuer* during the U-Prove insuance protocol. To prevent forgery and to ensure authenticity and integrity, tokens are signed by the *issuer* when they are generated[19].

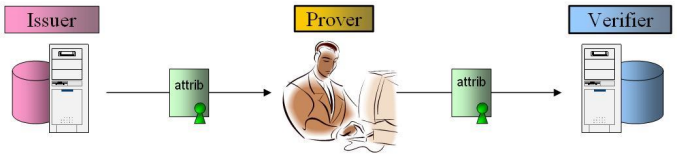

Figure 2: U-Prove framework
Figure is taken from U-Prove Technology Overview V1.1 Revision 2 page 14[19]

6.1 Privacy properties

During the U-Prove insuance protocol the *user* generates a fresh asymmetric key pair, and the *issuer* then blindly encodes the *user*'s public key part into the issued token. This embedded user-key allow *verifier*s to issue unique presentation challenges to *user*s thus confirming the *user*s' identities while denying replay attacks [19].

At the end of this process *user*'s public token-specific key is still only known to the user and the *verifier*, since the *issuer* had to blindly encode it into the token [19]. This means that the *issuer* and the *verifier* cannot later use this information even if colluding to unmask the identity of the owner of the token - a concept refered to as untraceability. Since a unique key pair is generated by the *user* for each token, uses of different tokens can never be linked to the same user - a concept refered to as unlinkability [19].

6.2 On-demand and long-lived tokens

Tokens can be requested on demand from the *issuer* while interacting with a *verifier*, or be long-lived and work for multiple *verifier*s [19]. The required type depend on the application, e.g. whether the service should be able to recognize returning *user*s over multiple sessions. In general, tokens may be costly or time consuming to generate depending on the type of attributes that the *issuer* must vouch. So while on-demand are guaranteed to be fresh, long-lived tokens may be more feasible for depending on the circumstances.

6.3 Granularity

U-Prove tokens contains an attribute field in which a number of attributes can be stored. This field has the property that the *user* can pick and choose exactly which of these properties to disclose [19]. This means that while an issued token can contain many attributes only the part strictly required by the *verifier* need to be revealed.

6.4 Generation of new pheudonyms

In U-Prove *verifier*s can compute a unique token identifier [19]. This means that the *verifier* will be able to recognize a *user* that uses the same token multiple times. This can be useful if the *verifier* is a service that relies on a *user* profile (e.g. a forum). However, in some cases it is beneficial that multiple visits to a *verifier* be unlinkable. In this case a new token must first be generated at the *issuer* and sent to the *user* before communicating with the *verifier*.

6.5 Revocation

U-Prove tokens can be revoked globally at the *issuer* or locally at a *verifier*. Since a unique identifier can be computed for each token, a *verifier* can simply blacklist that identifier for local revocation. Global revocation requires encoding an identifier into the tokens information field. Note though, that a token cannot have both untraceability and global revocability since these properties conflict [19].

7 Idemix

Another approach to privacy preserving authentication is given by IBM. They have developed a system called Identity Mixer, in short Idemix, which also protects the identity of the user like U-prove, but in a different way.

Idemix is a very complex system that contains a lot of different protocols, that can ensure different aspects of the authentication. [5]

I this section we describe how Idemix works, and which opportunities for authentication, revocation and identification it gives. The description is taken mainly from Bichsel et al. *Research Report Cryptographic Protocols of the Identity Mixer Library* [3], Camenisch and Van Herreweghen *Design and Implementation of the idemix Anonymous Credential System* [5], Camenisch *ABC4Trust & PrimeLife Toturial - Part IV: Use Cases and 2'nd Demo* [4], and Camenisch and Lysyanskaya *An efficient system for non-transferable anonymous credentials with optional anonymity revocation* [8]. Figures in this section are taken from Camenisch, Lehmann, and Neven *Privacy-enhancing Attribute-based Authentication* [7].

7.1 Outline

In Idemix the *user* is equipped with a private master key K. From this master key the *user* can generate several pseudonyms. These pseudonyms can be sent to an *issuer* along with some identification, and the *user* can receive a certificate C for the given pseudonym [5]. This is illustrated in fig 3. A variation of this is sending the *user*s actual identity instead of a pseudonym and the receive a certificate on this identity. [4]

The certificate consist of some attributes relevant for the *user*. When the *user* wants to get into contact with a *verifier* O_v, the user generates a pseudonym denoted N and pulls out the relevant attributes from her certificate. Then the *user* sends the pseudonym and the special certificate to the *verifier*. [5]

The *verifier* can now verify the credentials, and grant access. This is illustrated in fig. 4.

We see that in Idemix the *user* gets certified for all its attributes, and can afterwards by herself distribute what attributes to send to who. This

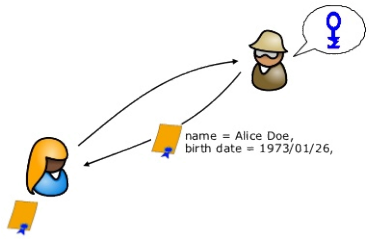

Figure 3: Idemix certificate issuing.
Figure is taken from the slides *Privacy-enhancing Attribute-based Authentication* page 16 [7]

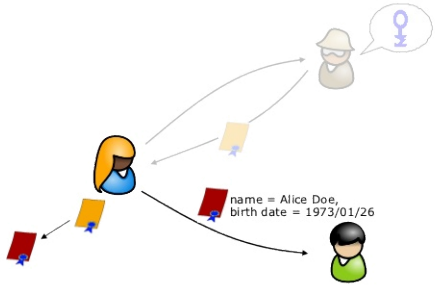

Figure 4: Idemix certificate distribution
Figure is taken from the slides *Privacy-enhancing Attribute-based Authentication* page 18 [7]

minimize the communication needed with the *issuer*.

7.2 Variations

When the *user* wants to get into contact with a *verifier*, O_v, the *verifier* can demand that the user generates a pseudonym specific for O_v, here called N_v that she tie the relevant attributes from her certificate to N_v before sending them. [4] An illustration of this can be seen in fig 5. This is done by generating N_v with a part from the *verifier* and the *user*s master key K. [8] This ensures that one user can only achieve one pseudonym pr. *verifier*.

Another variation is when the *verifier* demands a traceable pseudonym. Here the *user*s pseudonym is an encryption of her identity, which can be decrypted by an *inspector*. In this case the *verifier* can tie the *user*s certificate to her pseudonym and verify this but only the *inspector* can decrypt the pseudonym to an identity. [5] [4]

Yet another variation is were you collide certificates, and generate an

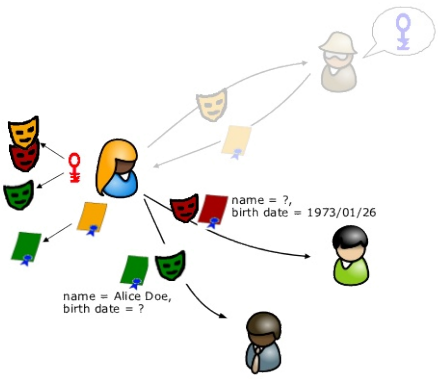

Figure 5: Idemix verifying with spesific pseudonym
Figure is taken from the slides *Privacy-enhancing Attribute-based Authentication* page 29 [7]

attribute set which origins from several certificates. [8]

7.3 Security

The system is build on public key cryptography, where the *issuer* sign the attributes with his private key. When the *user* uses some of the attributes the *verifier* can verify that they were stated by the *issuer* and then trust the authenticity of the certificate. [8] [3] In the case where an attribute set consists of attributes from different *issuer*s the *verifier* would be able to verify which attribute is issued by which *issuer*. [4]

Idemix has a lot of variations. This makes a lot of security combinations possible, but it also enhance the complexity of the system. A complex system is often more difficult to maintain. [22]

8 Discussion

We have now identified two privacy-enhanced authentication solutions: Idemix and U-Prove. This section discusses the advantages and disadvantages of choosing one of the solution over the other. First we discuss the different characteristics of the two solutions in section 8.1. Then we discuss how the two solutions applies on the different classes from our problem in section 8.2

8.1 Idemix and U-Prove

Both of these two solutions provides a way to authorize a *user* without revealing her identity. We call this a zero-knowledge proof, since we prove

some attributes about a *user* without revealing any knowledge about who she is, other than the necessary attributes.[3]

U-Prove is simple in the way that the *user* gets pseudonym and attribute from an *issuer* and uses that for verification. This makes the security comprehensible but every time the *user* needs access with a new pseudonym to a *verifier*, she needs to contact the *issuer*. This causes some additional traffic, and if there are many *user*s in a system and few *issuer*s, this setup can have high impact on how fast a transaction is. The *issuer* is omitted in every access protocol in Idemix, where the *user* receives all her certificates and then by herself extract attributes, create pseudonyms, and sends these to a *verifier*. In this way you move the *issuer* contact to preprocessing, shorten the protocol with the *verifier*, and you do not need the *issuer* on demand, such that you can control when the *issuer* is active instead of it always being so. Additional you have the computation of the pseudonym at the *user* instead of at the *issuer*, which limits the computation power needed for the *issuer*. This trick cost a lot of complexity in order to ensure authenticity and privacy for the *user*, but the amount of authentication to the *issuer* is decreased.

One of the great things about U-Prove and Idemix are untraceability and unlinkability. The privacy of the *user* is always ensured, even if the *verifier* and the *issuer* collude. But this is not always possible to ensure. If someone in Idemix get their hands on the *user*s master key, they can unwrap everything[3] so we assume this master key is kept safe. Further more in Idemix if a *verifier* wants to ensure that one *user* only has one pseudonym the *verifier* can state some numbers the pseudonym should contain and the method of creating pseudonyms from the master key makes it possible only to have one pseudonym for this *verifier*. This is much more difficult to ensure with U-Prove. Here the *issuer* would have to ensure that a *user* only has one attribute set per *verifier* and thus keep a log of all *user*s and where they have pseudonyms to, which can be quite comprehensive.[19] Likewise traceability can be necessary and the *verifier* can demand this property in both solutions. The *verifier* himself is not able to trace back the *user*, but the *user*s identity is encrypted such that an *inspector* can trace it back. This feature is used when revocation is needed, since the *verifier* needs to know which pseudonyms the revoke.

This issue is one of the issues in privacy-enhanced authentication systems that you can not have untraceability, unlinkability, and revocability simultaneously.[19] You can have a weak form of revocability, if you have linkability, where a *verifier* can exclude a *user* using the same pseudonym. But the *user* can get a new pseudonym and thus grant access again.[19]

If you have revocability, unlinkability, and traceability the *revocation authority* must have traceability access like the *inspector* but then the *revocation authority* can identify which pseudonyms to revoke and can revoke the*user* with several *verifier*s.

We have summarised the above mentioned features in table 2.

Table 2: Comparison between U-Prove and Idemix

Properties	U-Prove	Idemix
Zero-knowledge proof	✓	✓
Simple system	✓	
Divisible pseudonym		✓
Divisible attributes	✓	✓
Untraceability	✓*	✓*
Unlinkability	✓*	✓*
Revocable	✓*	✓*
Attribute generation	*issuer*	*user*

8.2 Authentication system for *CampusNet*

In the authentication system for *CampusNet* we have defined four classes that have different authorization needs. We will look at these classes and discuss which solution is best suited for that class. The solutions concerned is nothing, U-Prove, and Idemix.

8.2.1 World

The **World** class resources does not need to be protected with any security and thus authorization is omissible. Using either authentication solution will only grant unnecessary complexity.

8.2.2 DTU

The **DTU** class resources needs to authorize that people are in fact from DTU. Both U-Prove and Idemix ensures privacy enhanced authentication. In security it is often best to chose the simplest solution that fits, since more complexity results in fewer people understanding the system and can lead to flaws.[22] U-Prove is the simplest system. If we choose this we will have some additional traffic with the *issuer* in the verifying process compared to Idemix. The *issuer* is the DTU administration, the *verifier* is Campusnet. Since both *issuer* and *verifier* is run in close proximity at DTU, the communication can be done with little delay thus U-Prove's additional contact with the *issuer* does not consume much time.

Since our threat model does not consider the *issuer* and the *verifier* colluding, we do not need unlinkability and untraceability. Revocability, however, could be an issue if a student is expelled from DTU. This can be ensured using both Idemix and U-Prove. However if the attribute for enrolment at DTU only is valid one year at a time, the impact of not having

revocabilty is not severe since the protected files are not that sensitive. E.g if a *user* is expelled from DTU, the granted access to a the DTU phone book for the rest of the year does not have much impact, assuming the phone book does not change rapidly, since she could have downloaded a fairly recent copy anyway.

8.2.3 Course

The **course** class deals with the same issues as the **DTU** class. The protected files are a bit more important since there are copyright issues with non-participating people having access to course material. When this is said users that are kicked out have already had access to the material. So even if the users access is immediately revoked the denial of access may not be significant compared to the naturally expiry date at the end of the course. One impact is the expelled students ability to still follow the course by accessing new resources added in the course folder. Another impact could be expelled students posting excessively in the course conference room, which may have an negative social impact. These impacts are solved if revocation is possible since the expelled can be banned immediately.

A difference to the **course** class is that the professor can make a biased grade to students who download the course material and students who not. In order to prevent this we have pseudonyms, but we need a new pseudonym for every download in order to prevent linkability between who has downloaded course material. This pseudonym generation is more expensive in U-Prove than Idemix, since in U-prove attributes are generated to a specific pseudonym. In order to make a new pseudonym one must contact the *issuer* and get a whole new attribute set based on the new pseudonym.[19] In Idemix the *user* can generate a new pseudonym from her master key, and use a certificate with the correct attributes and generate a token to the new pseudonym. The great difference is whether the attributes and the pseudonym are generated at the *user* or at the *issuer*, but when generating at the *user* the *user* omit authenticating herself to the *issuer*.

8.2.4 Student

The **student** class resources have the issue that a student must be held accountable for his work without disclosing the identity to the professor. This means that the professor anonymously must grade assignments, but at the same time DTU administration must be able to trace the assignment and deliver the grade back to the student. In order to enforce this we need traceability. When it comes to the exam, we have two cases. One where the student must be enrolled in the course in order to go to the exam and a more restrictive one where the student need to pass a certain amount of assignments and be enrolled in the course in order to go to the exam.

Since the assignments are delivered with different pseudonyms the *verifier* cannot ensure this by himself, because he cannot link the assignments to a *user* or pseudonym. This means we need the *revocation authority* to tell the *issuer* how many assignments a *user* has passed. If this knowledge shall be linked to a pseudonym, the *user* must ask for an exam attribute telling whether or not she is allowed to go to the exam. This new attribute must be given by the *issuer* so both Idemix and U-Prove must contact the *issuer* again not just U-Prove as in the other cases. Since the exams is placed in certain periods of the year the *user* could achieve all exam attributes at ones. Idemix could receive an exam certificate, allowing all exams for the exam period, and U-Prove could achieve all pseudonyms and attributes at once as well. When an exam attribute is needed revocation becomes superfluous, since the attribute would be given just before the exam start and have the new knowledge anyway. When handing in assignments we only need the attribute of being enrolled in the course, and then revocability would have an effect. The privacy for the *user* would be the same with or without revocability, since traceability is necessary and the *inspector* and the *revocation authority* is the same, DTU administration, in our solution. The professor might have some additional work correcting expelled students assignments if revocability is not ensured, and thus revocability is preferable.

8.2.5 All classes

When looking at the classes individually there are some advantages and disadvantages for choosing one privacy-enhanced authentication solution over the other. The major discussion of whether to chose one solution over the other is whether to chose system complexity over computationally advantages in the access protocol. Idemix is more complex, but does not depend on the *issuer* in the access protocol, where U-Prove are simpler but depends on the *issuer* in the access protocol.

One could argue that we could use the solution best suitable for every class and thus ending up with using both Idemix and U-Prove. This would, however, have large impact on the complexity and usability, and would need separate authentication for every class such that being logged in at a course folder not necessarily gives access to resources from the **DTU** class such as the study handbook. This would result worsening the usability and the administrating two different systems would be immense.

9 Solution

In section 8 advantages and disadvantages with Idemix and U-Prove solutions are discussed. We found that any solution requires constant generation of pseudonyms. This is a problem in U-Prove since each new pseudonym requires a new token to be generated by the issuer. Under the assumption that

generation is a computationally taxing process and there are limited computer power available for this power at DTU, it is better to move pseudonym generation to the users own system. This is particular important during peak load (e.g. semester start and exam periods). Furthermore authentication will be needed for every contact with the *issuer*.

As a result of this Idemix is chosen for our solution.

9.0.6 Actors

The *verifier* in the solution is the *CampusNet*-service. The *user* is in our case the student. The DTU administration hold the roles of the *issuer*, the *revocation authority* and the *inspector*. The following subsections describe our solution in further detail.

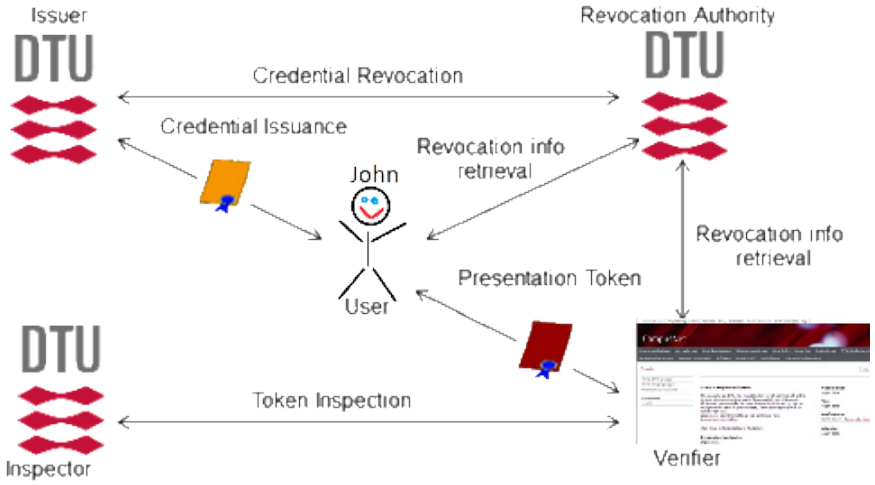

Figure 6: Solution using Idemix

9.1 Trust relationship

As described in the section 8 traceability is required in order to ensure that authentication tokens can be revoked when the administration no longer consider them valid.

Due to this issued tokens must be trusted by the *CampusNet*-service and the DTU administrations ability to revoke tokens needs to be trusted by the *user* and *CampusNet*. Indeed, we cannot build a solution without a trusted party[16], but since DTU administration are independent and it is in their best interests to be considered impartial, students and professors should be able to trust their discretion.

Because of this trust relationship the DTU administration will be able to create, modify and delete *user*s, and beside this the administration will

be able to link the correct grade with the correct student when a assignment or an exam is graded by a professor.

9.2 Authentication with issuer

When a new student has to be enrolled in the *CampusNet*-services, the administration needs to authenticate the student before issuing a certificate.

As stated in section 1.3 the student is required to login using their nemID account. During this session a *Idemix master key* will be generated as described in section 7. For further interactions with the *issuer* possession of this master key can prove the *user*s identity. This master key will be used by the *user* to create pseudonyms and compose tokens for these pseudonyms. New attributes will have to be certified from the *issuer* before they can be used.

9.3 Attributes

The attributes that the certificates can hold are

- DTU enrolment
- Enrolled courses
- Qualified exams

Note, that as explained in section 7 certain attributes can be traceable. That is they are kept secret from the *verifier* while being disclosable to the *inspector*.

9.4 Revocation

Revocation is needed in the **student** class in order to revoke the number of exam giving assignments passed to the *issuer*. As discussed section 8 we do not loose any privacy on ensuring revocation from *issuer* to *verifier* as well, but gain some unnecessary time spend from the professor. Revocation would also be preferable in the **course** class in order to avoid spam in the conference room. For the **DTU** and the **world** class revocation is found unnecessary, and will not be upheld.

9.5 Use case World 1: Looking up a two course descriptions

The *user* will log into the course database, and will be able to search and read about the two courses.

1. The *user* requests to lookup the course description. In this process the request is always granted.

9.6 Use case DTU 1: Looking up two contacts

The *user* will log on the *CampusNet*, open the phone book and look up two names or telephone numbers.

1. The *user* is logged in using a pseudonym that includes an non-traceable attribute certifying the *user* as a student of DTU.
2. The *user* requests to lookup contact details of a DTU affiliate in the phone book. In this process the request is granted if the *user* is a student at DTU.

9.7 Use case Course 1: Downloading of course material

The *user* will log on to *CampusNet*, find the needed material and download it.

1. The *user* is logged in using a traceable one-time pseudonym that includes a non-traceable attribute certifying the students enrolment in the course.
2. The *user* requests to download a single hand-in on *CampusNet*. In this process the request is granted if the *user* is enrolled in the course, and their enrolment status has not been revoked.

9.8 Use case Student 1: Assignment hand-in

When a student is logged on to *CampusNet*, he will be able to hand-in the assignment, similar to the way it is done now. When the assignment is graded, the student will receive the grade.

1. The *user* hand in the assignment with the assignment specific traceable pseudonym on *CampusNet*, which will verify that that it is traceable, unique to a student, that that student is enrolled in the course, and that the student enrolment status in the course has not been revoked.
2. The professor will then be capable of viewing the hand-in, grade it, and then upload the assessment *CampusNet*.
3. The administration will then be able to view the grade, unmask the pseudonym and link the grade to the real life identity of the student.

9.9 Use case Student 2: Exam hand-in

When a student is logged on to *CampusNet*, he will be able to hand-in the assignment if the exam requirements are fulfilled. When the exam is graded, the student will receive the grade.

1. The *user* is logged in using a traceable one-time exam-specific pseudonym which includes an attribute certifying the students entitlement to participate in the exam.

2. The *user* hands in the exam to *CampusNet*. In this process the *verifier CampusNet* will verify that that the hand-in is traceable, unique to a student, that the student is entitled to the course exam, that the pseudonym has not already answered exam, and that neither course enrolment nor exam-entitlement has been revoked.

3. The professor will then be capable of viewing the exam hand-in, grade it, and then upload the assessment to *CampusNet*.

4. The administration will then be able to view the grade, unmask the pseudonym and link the grade to the real life identity of the student.

9.10 Combined use case

As the *user* is performing different activities, their computer communicates with *CampusNet* in order negotiate which kind of pseudonyms and attributes are required to facilitate the different part of the combined use case. As long as the *user* only is allowed to perform specified activities, the computer can sequence any of the described actions in order to achieve the *user*s goal and the *user* will only have to authenticate one time per session.

10 Verification

To verify our solution, we will check that the use cases from section 4 are satisfied under the threat model as defined in section 4.2, to confirm that our solution in section 9 solves the problems for each of the four classes, described in section 4.1.

World The current state of *CampusNet* has no privacy issues concerning the world class. Due to this our solution handles the world class correct.

Use case 1 Since Use Case World(4.1.1) has neither constraints nor privacy issues, the use case is trivially satisfied.

DTU According to the threat model, the thread against *user*s enrolled on DTU, is concerning the phone book and study handbook. Our solution handles this by using the Idemix pseudonym to verify that the *user* is indeed enrolled at DTU. This ensures that you cannot access the phone book if you are not enrolled at DTU, and that you cannot trace lookups in the phone book back to the user.

Use case 1 Since the pseudonym holds a DTU-enrolment attribute, it is ensured that the *user* is a student at DTU, which satisfies the constraints of Use Case DTU(4.1.2).

Course Our threat model states that the student can be graded unfairly based on download history. Our solution handles this by using a one-time pseudonym, with a course attribute. Due to the one-time use of this untraceable pseudonym, the professor will not be able to log the students history and grade unfairly. Due to the course attribute of the one-time pseudonym, only *user*s enrolled in the course will be able to download the material, hence copyright issues are avoided.

Use case 1 Since the professor is unable to unmask the students identity, and that a unique, unlinkable pseudonym is used for each download then the privacy properties of Use Case Course(4.1.3) are satisfied. The professor is ensured that the student is enrolled in the course, which satisfies the constraints of Use Case Course(4.1.3).

Student The threat model states that the issues concerning the student is unfair grading based on personal relationships with professors or physical attributes. Our solution handles this by using a hand-in specific pseudonym with a course attribute or an exam attribute each time the *user* hand in an assignment or an exam. This set-up ensures that the professor do not know which students hands in which assignment, but is guaranteed that the student only hand in once and is enrolled in the course or allowed to hand in the exam. When the professor is finished grading, the pseudonym and the grade are linked by the DTU administration, who will be able to revoke the pseudonym and link the correct real-life student with the grade.

Use case 1 Since the professor is unable to unmask the students identity, and it is possible for the DTU administration to the link the assignment back to the student then the privacy properties of Use Case Student 1(4.1.4 are satisfied. The professor is ensured that the student is enrolled in the course, which satisfies the constraints of Use Case Student 1(4.1.4).

Use case 2 Since the professor is unable to unmask the students identity, it is possible for the DTU administration to the link the assignment back to the student, and that any qualifying assignments are unlinkable due to the hand-in specific pseudonym, then the privacy properties of Use Case Student 2(4.1.4) are satisfied. The professor is ensured that the student is qualified to participate in the exam, which satisfies the constraints of Use Case Student 2(4.1.4).

Since privacy properties and the constraints are satisfied under the threat model for all the use cases the verification requirements stated in section 4.3, thus the solution is valid.

11 Conclusion

We have examined bias in grading when students hand in an assignment using *CampusNet*. Several privacy issues were identified, including disclosure of students identity when acting on *CampusNet*. We created a modelling of the user groups on *CampusNet* using the classes World, DTU, Course, and Student. On the basis of this and the privacy issues, we have developed six use cases and a threat model which details the threats, actors, and attacks that we consider for the solution. To find a suitable solution, we have examined open and closed loop authentication. We found that closed loop is unsuitable for our privacy issues, due to the fact that you have to trust the service, in contrast to open loop authentication, where it suffices to rely on a commonly trusted third party. After examining the two major contenders in privacy enhanced access control, Microsoft's U-Prove and IBM's Idemix, we have decided to use Idemix, due to the increased performance in anonymously accessing resources in an unlinkable manner. Finally we were able to describe a solution which solves the privacy issues previously identified under the specified threat model. The solution uses unlinkable pseudonyms with given attributes, to ensure the modelled constraints and privacy properties. Finally the six use cases were verified against the solution.

12 Future work

This report is scoped quite narrowly in order to deliver a ready to implement proof of concept on a privacy-enhanced access control solution to *CampusNet* within the timeframe set aside for this project. In order to apply this solution in real life some future work is needed which includes implementing the solution and extending the solution to cover other *CampusNet* features such as

- Re-exams
- Re-submission of assignments
- Modifying hand-ins after delivery to *CampusNet* but before deadline
- Viewing exam grades

Defining the use of *CampusNet* for professors and the DTU administration also lies within the scope of future work necessary in order to apply this solution on *CampusNet*.

13 References

References

[1] ABC4Trust. Project description, 2009.

[2] ABC4Trust. Abc4trust architecture for privacy-abcs, 2015.

[3] Bichsel, P., Binding, C., Camenisch, J., Gross, T., Heydt-Benjamin, T., Sommer, D., and Zaverucha, G. Research report cryptographic protocols of the identity mixer library. http://paedi.biche.ch/work/pub/rz3730.pdf.

[4] Camenisch, J. Abc4trust & primelife toturial - part iv: Use cases and 2'nd demo. https://abc4trust.eu/webmedia/2011-06-10-abc4trust_tutorial/Session4.html.

[5] Camenisch, J., and Herreweghen, E. V. Design and implementation of the idemix anonymous credential system. http://freehaven.net/anonbib/cache/idemix.pdf.

[6] Camenisch, J., Krontiris, I., Lehmann, A., Neven, G., Paquin, C., Rannenberg, K., and Zwingelberg, H. D2.1 architecture for attribute-based credential technologies, version 1, 2011.

[7] Camenisch, J., Lehmann, A., and Neven, G. Privacy-enhancing attribute-based authentication. http://www.slideshare.net/thomasgross/attributebased-authentication.

[8] Camenish, J., and Lysyanskaya, A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. https://eprint.iacr.org/2001/019.pdf.

[9] Corella, F., and Lewison, K. A comprehensive approach to cryptographic and biometric authentication from a mobile perspective. http://pomcor.com/whitepapers/CryptographicAuthentication.pdf.

[10] Corella, F., and Lewison, K. Privacy postures of authentication technologies, 2013.

[11] Cornwell, C., Mustard, D. B., and Van Parys, J. Noncognitive skills and the gender disparities in test scores and teacher assessments: Evidence from primary school. *Journal of Human Resources 48*, 1 (2013), 236–264.

[12] Coull, S. E., Green, M., and Hohenberger, S. Access controls for oblivious and anonymous systems. In *ACM Transactions on Information and System Security* (2011), vol. 14.

[13] de Gramatica, M., and Massacci, F. Abc4trust. In *FP4 ICT & Security Projects Handbook* (2015), University of Trento, pp. 10–12.

[14] Delaune, S., Delaune, S., and Ryan, M. Coercion-resistance and receipt-freeness in electronic voting, 2006. https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/2002/assignments/stho.pdf.

[15] Jensen, C. D. Open loop authentication. Oral: Lecture about Multi-factor authentication in DTU course 02234 Current Topics In System Security.

[16] Jensen, C. D. The importance of trust in computer security. In *Trust Management VIII* (2014), vol. 430, Springer Berlin Heidelberg.

[17] Jensen, C. D. Multi factor authentication, 2015. Lecture slides from DTU course 02234 Current Topics In System Security.

[18] Miller, S. P., Neuman, B. C., Schiller, J. I., and Saltzer, J. H. Kerberos authentication and authorization system. In *Project Athena Technical Plan* (1987).

[19] Paquin, C. U-prove technology overview v1.1 (revision 2), April 2013. http://research.microsoft.com/apps/pubs/default.aspx?id=166980.

[20] Rannenberg, K. Privacy-abcs and anonymous course evaluations at universities Ű an eparticipation vehicle in education, 2015.

[21] Rausch, T., Karing, C., Dörfler, T., and Artelt, C. Personality similarity between teachers and their students influences teacher judgement of student achievement. *Educational Psychology* (2015).

[22] Thorén, S. Applying common software security guidelines to improve program robustness. https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/2002/assignments/stho.pdf.

[23] Veseli, F., Sommer, D. M., Schallaöck, J., and Krontiris, I. D8.12 architecture for standardization v2. https://abc4trust.eu/download/D8.12%20-%20Architecture%20for%20Standardization%20V2.pdf.

[24] Wang, J., Floerkemeier, C., and Sarma, S. E. Session-based security enhancement of rfid systems for emerging open-loop applications. In *Pers Ubiquit Comput* (2014), Springer.