A RISK-DRIVEN APPROACH TO DESIGNING PRIVACY-ENHANCED SECURE APPLICATIONS

Els Van Herreweghen
IBM Research Division, Zurich Laboratory
evh@zurich.ibm.com

Abstract In the context of authorization in distributed systems, security and privacy often seem at odds. Privacy goals motivate the use of privacy-enhanced forms of authorization such as attribute-based, anonymous authorization; the need to identify misbehaving users calls for either identity-based authorization or identity escrow allowing re-identification of users.

We propose a risk-driven design approach for maximizing privacy of users while satisfying security requirements of an application. In this approach, a security measure such as authentication or identity escrow is introduced only if it addresses a concrete risk. The approach helps to identify privacy-friendly solutions as well as trade-offs between privacy and cost considerations. We illustrate our approach with an example application using anonymous credentials.

Keywords: anonymity, privacy, accountability, risk analysis, anonymous credentials

INTRODUCTION

In distributed systems, the authorization of a specific request for access is often based on authentication of the requesting individual using a certificate or credential issued by a trusted entity. Attribute-based authorization is based on the individual proving possession of certified attributes rather than on his identity; attributes can be certified in conventional public-key certificates [9, 8] or in anonymous credentials [5, 4, 6, 7, 12]; proof of ownership of (the secret associated with) such a certificate or credential then proves ownership of the attributes. As attributes need not be associated with a name, attribute-based authorization can be used to increase the privacy of users while maintaining secure authorization.

Security considerations encompass more than only the verification of a user's right to perform a certain action. For many applications, it is perceived that users can misuse their rights in a way which may mandate establishing the user's identity after the fact in order to hold the user accountable for his actions. Re-identification may be achieved by means of an identity escrow entity [1] trusted with the mapping between a certificate or public key and a user's real identity; revealing this mapping may be subject to certain misuse conditions being satisfied. An issue which has received less attention than user accountability is the accountability of credential or certificate issuers towards relying parties accepting these credentials and certificates. Trust management systems (e.g., [2, 8, 11]) define what are valid chains of trust but fail to address the question of liability and verifiability of certificate issuers. When the owner of an online shop says 'I trust the customer's bank'; he probably means: 'The bank has issued certificate practice statements with liabilities for payments based on certificates it issues; the bank is endorsed by an insurance company with appropriate liabilities. Therefore, I trust that I will receive the money associated with a payment based on a certificate issued by the bank.'

Designing systems with maximal security and privacy clearly requires a way of stating security requirements in a way which allows satisfying them with privacy-friendly technologies. Also this issue seems not to have been addressed so far. In requirements engineering (e.g., [16, 15]), security requirements are often stated in terms of mechanisms such as (traditional, identity-based) authentication. Also research in the field of *security patterns* (e.g., [13, 10]) and their use in modelling and analyzing security requirements describes security problems and requirements at a similar level.

With the risk-driven design approach proposed in this paper, we attempt to address the above issues. Our goal is to correctly address security requirements while maximizing users' privacy. The focus on risks allows us to describe security needs in a mechanism-independent way and to satisfy them using privacy-friendly technologies. It also helps us in finding or avoiding hidden trust assumptions between issuers and relying parties. The approach also identifies trade-offs between privacy (or anonymity) and other considerations such as cost.

The principles of our approach are independent of the technology used for authenticating users' access requests. Of course, we can best illustrate them using an authentication mechanism supporting anonymity as well as accountability. We illustrate our approach using the anonymous credential system with conditional re-identification described in [3, 14].

The remainder of the paper is organized as follows. In Section 1, we introduce the anonymous credential system used to illustrate our approach. In Section 2, we introduce our example application and propose a first, ad-hoc, design. In Section 3, we re-design the example application using our risk-driven approach and discuss its advantages. Section 4 concludes the paper.

1. AN ANONYMOUS CREDENTIAL SYSTEM WITH CONDITIONAL RE-IDENTIFICATION

The anonymous credential system with which we illustrate our approach is the one described in [3, 14]. Here, we describe only the basic constructs; for a more in-depth discussion, we refer to [14].

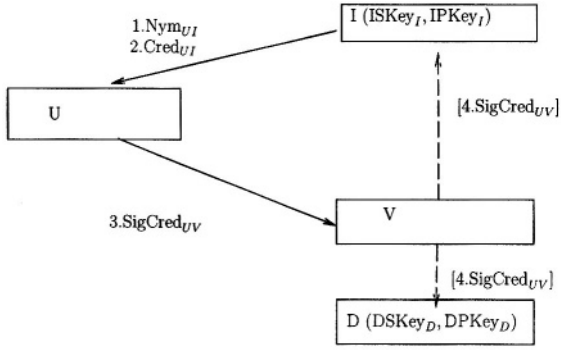

Figure 1. Basic Credential System Protocols

Figure 1 shows a user U (1) establishing a nym (pseudonym) Nym_{UI} with credential issuer I, (2) obtaining a credential Cred_{UI} from I, and (3) proving ownership of that credential to a verifier V by producing a credential-based signature $\mathrm{SigCred}_{UV}$. Such a credential-based signature proves that the signer possesses a credential issued by I. However, it does not contain Cred_{UI} or Nym_{UI} (optional mechanisms allowing to derive Nym_{UI} from $\mathrm{SigCred}_{UV}$ under specific conditions are discussed below). Obtaining a nym or credential and signing with a credential are interactive protocols; representing them as single messages is appropriate for the present discussion.

A credential Cred_{UI} is represented as follows:

$$\mathrm{Cred}_{UI} = \mathrm{Cred}(\mathrm{Nym}_{UI}, \mathrm{IPKey}_I, \mathrm{MultiSig}, \mathrm{Exp})$$

- Nym_{UI} is the nym (pseudonym) under which I knows U. Unless establishing Nym_{UI} involved identity escrow (see below), I cannot link Nym_{UI} to the identity of U.
- IPKey_I is the issuing public key of the issuer; the credential is signed with the corresponding secret key ISKey_I.
- MultiSig being `true` or `false` indicates whether the credential can be used multiple times or only once. When U signs twice with the same one-use credential, the resulting signatures allow I to extract Nym_{UI}. Figure 1 shows V sending SigCred_{UV} to I for double-spending detection. This can be done online, during verification of SigCred_{UV}; offline double-spending detection is less expensive to realize but only allows for after-the fact detection.
- Exp represents a credential expiration time.

U's credential-based signature reflects the parameters and options with which U invoked the signing.

$$\text{SigCred}_{UV} = \text{SigCred}(\text{Cred}_{UI}, \text{Msg}, \text{DeAnCond}, \text{DPKey}_D)$$

represents a signature on Msg (may be `null`, if the goal is authentication without signing a particular message) using Cred_{UI}. The zero-knowledge realization of SigCred_{UV} ensures that SigCred_{UV} cannot be linked to another signature $\text{SigCred}_{UV'}$ with the same Cred_{UI}; SigCred_{UV} only proves (and allows V to prove) that Msg was signed with a (non-expired) credential issued by I, and with parameters DeAnCond and DPKey_D as described in the following.

DPKey_D and DeAnCond are optional parameters related to deanonymization. If DPKey_D is `null`, there is no way of linking SigCred_{UV} back to Nym_{UI}, even with cooperation from I. If DPKey_D is non-`null`, it represents the public key of a deanonymization organization D; with the associated secret key DSKey_D, D can deanonymize the transaction, i.e., extract Nym_{UI} from SigCred_{UV}; Figure 1 shows V sending SigCred_{UV} to D for deanonymization. DeAnCond expresses the condition (signed by U) under which deanonymization by D is allowed. (Of course, fairness and correctness of D's operation are fundamental to users trusting the system. In [14], ways are discussed to motivate such fairness and correctness by making D's actions verifiable.) From SigCred_{UV}, V has a proof that D can deanonymize the signature; when D actually deanonymizes it, D can also prove the correctness of this deanonymization, i.e., D can prove that SigCred_{UV} indeed resulted from showing a credential issued on Nym_{UI}.

Deanonymization thus allows for a conditional linking between a transaction (SigCred_{UV}) and a user's nym (Nym_{UI}). Another feature necessary for realizing re-identification is identity escrow ('signed nym registration' in [14]). Identity escrow is realized by U providing I with a proof of linking between Nym_{UI} and U's real identity with a signature

$$\text{SigNym}_{UI} = S_U(\text{Nym}_{UI}, \text{Msg})$$

where $S_U()$ denotes a signature with a signature private key SSKey_U associated with a public key SPKey_U certified in an 'external' certificate Cert_{CA-U}. This certificate is issued by certification authority CA, who is trusted to either include the real name of the user in Cert_{CA-U} or to reveal it whenever asked. Msg is an optional message; it can, e.g., contain information about U's liability for Nym_{UI}.

When used without identity escrow or deanonymization, the anonymous credential system described allows for fully unlinkable and anonymous attribute-based authorization. The optional features of deanonymization and identity escrow allow for a conditional re-identification with separation of duties between de deanonymizer (D) conditionally mapping a transaction back to a nym and the entity enforcing identity escrow (I) mapping the nym back to an identity. In the following sections, we now use this credential system in the design of secure and privacy-enhanced applications.

2. THE AdsOL ADVERTISEMENT SERVICE

2.1 High-Level Description

AdsOL is an advertisement service accessible through the PrivacyPortal web portal. PrivacyPortal promotes the use of anonymous credentials as the recommended mechanism for any type of authentication. PrivacyPortal offers a Kiosk service to all its members; Kiosk is able to accept an 'external' payment on behalf of AdsOL (or another portal service) and converts it into a payment proof in the form of an anonymous credential.

AdsOL charges users posting advertisements, e.g., by a monthly fee; retrieving and reading advertisements is free of charge. Users posting advertisements can enter pseudonymous contact information such as a temporary or pseudonymous e-mail address obtained from PrivacyPortal's pseudonym email address server. This allows the user who posted an ad to remain anonymous even when contacted by an interested party.

AdsOL does not limit the range of items that can be advertised for. However, if a user posts an illegal (e.g., drugs-related) advertisement, law enforcement (LE) requires a re-identification of the transaction.

U obtaining posting credential from K:

$K \rightarrow U$: Nym_{UK}

$U \rightarrow K$: payment, Cert_{CA-U}, $\text{SigNym}_{UK} = S_U(\text{Nym}_{UK}, \ldots)$

$K \rightarrow U$: $\text{Cred}_{UK} = \text{Cred}(\text{Nym}_{UK}, \text{IPKey}_K, \texttt{true}, \text{Exp}_{UK})$

U posting advertisement to A:

$U \rightarrow A$: $\text{SigCred}_{UA} = \text{SigCred}(\text{Cred}_{UK}, \texttt{Advert}, \texttt{MisuseCond}, \text{DPKey}_{LE})$

Re-identification:

$LE \rightarrow A$: need re-identification of `Advert` posting

$A \rightarrow LE$: SigCred_{UA}

LE: deanonymizes SigCred_{UA} and obtains Nym_{UK}

$LE \rightarrow A$: need U related to Nym_{UK}

$A \rightarrow K$: need U related to Nym_{UK}

$K \rightarrow A$: U and proof of mapping between U and Nym_{UK}

$A \rightarrow LE$: U and proof of mapping between U and Nym_{UK}

Figure 2. AdsOL Design

2.2 AdsOL Design

We now propose a first privacy-friendly design satisfying above requirements based on attribute-based authorization using our anonymous credential system; in the next section, we will then show how a risk-driven design can identify more secure and privacy-friendly solutions.

The idea of our solution is the following.

- K (Kiosk) issues a posting credential (a paid subscription to AdsOL) to a user U on condition of having received payment, and having applied identity escrow to the credential (nym);
- A (AdsOL) accepts an advertisement on condition of having received a deanonymizable signature with a posting credential issued by K.

The design is summarized in Figure 2. In a first step, U obtains the posting credential from K as follows. U establishes a nym Nym_{UK} with K, performs the payment and provides K with the identity escrow information $(\text{Cert}_{CA-U}, \text{SigNym}_{UK})$. K then issues the posting credential Cred_{UK} valid for a certain period (Exp_{UK}).

When posting an advertisement, U sends SigCred_{UA} to A. The advertisement is included as the signed message in SigCred_{UA}; SigCred_{UA} is deanonymizable by LE on condition `MisuseCond`. Choosing LE as deanonymizing organization may make sense under the assumption that LE anyway has the last word in judging `MisuseCond`; we could also have chosen an independent organization D to deanonymize transactions.

If re-identification is necessary (`MisuseCond` fulfilled), *LE* asks *A* for SigCred_{UA} related to the specific `Advert`, deanonymizes it, and asks for the identity of *U* related to Nym_{UK} (we assume that *LE* deals with *A* for obtaining the necessary information). *A* obtains the provable mapping between *U* and Nym_{UK} from *K* and provides it to *LE.*

2.3 Analysis

In the above design, we directly derived authorization and re-identification requirements from a high-level description of the system. Authorization and (re-)identification ensure that security requirements are fulfilled; the attribute-based and anonymous authorization together with the conditions for re-identification were meant to ensure that these requirements were dealt with in the most privacy-friendly way.

By requiring that every posting transaction be re-identifiable, we assume having excluded *A*'s risk of not being able to re-identify when needed, and having provided *LE* with a secure means to trace illegal advertisements. Has our design fulfilled these goals?

We have assumed *A* to be liable towards *LE* for not being able to re-identify an illegal advertisement. In our design, *A* trusts *K* to invest in an identity escrow infrastructure (relationship with *CA*, verifying SigNym) and to be able to identify a real user associated with a nym. Neither our high-level description nor our design, though, has provided a motivation for *K* to do so in in the form of a contractual guarantee or liability towards *A* or *LE.* Thus, we have reduced the privacy of users (systematic identity escrow and re-identifiability of transactions) without correctly addressing the risk! Even if such liabilities exist, they may not compensate *A*'s potential loss. E.g., if *A*'s operating license gets revoked by *LE* if an illegal advertisement posting cannot be re-identified, *A* is likely to decide to act as escrow agent himself. Or, maybe, *A* will look for ways to prevent illegal advertisements altogether!

On the other hand, we can also stipulate concrete liabilities and exclude *A*'s risk without solving *LE*'s security problem. Assume that *A*'s liability for not being able to re-identify an illegal advertisement is a fine of $1000 to be paid to *LE;* and *K*'s liability for not being able to map a nym to an identity is a fine of $1000 to be paid to *A. A* has excluded his risk. But, a bribe of $2000 paid by the misbehaving user may convince *K* to claim he 'is sorry he lost the user's identity mapping record (including SigNym_{UK}) and is willing to pay the $1000 to *A*'. We now provide expensive but unconditional (assuming *K* is an 'honest' bribee) anonymity to misbehaving users.

The above reasoning presents several issues we did not consider in our first design:

- Risks and liabilities should be made explicit without (misplaced) trust assumptions between organizations: *A* should not have to (blindly) trust *K* for addressing *A*'s risks and liabilities.
- When addressing a risk, the cost of the risk and its liabilities has to be weighed against the cost of addressing the risk. A fine provides a different motivation for addressing the risk than having to suspend operation. From the point of view of the party (*LE*) imposing this liability, this of course means that a liability has to motivate the intended result (correct re-identification).
- We have decided to address *LE*'s requirement for re-identification without considering whether it was possible to prevent misuse altogether.

Our risk-driven design in the next section will address these issues.

3. A RISK-DRIVEN DESIGN OF THE AdsOL SERVICE

In this section, we propose a risk-driven approach for satisfying security requirements while maximizing privacy. In this approach, an initial risk analysis is followed by an iterative process of design option analysis, design decisions and residual risk analysis; during this process, we gradually refine the protocols used. By focusing on concrete risks for a specific party, we assure that trust and liability requirements among organizations are made explicit. By identifying potential design options regardless of cost, we can identify the most privacy-friendly solution as well as compromises beween privacy and cost.

Figure 3 illustrates the risk-driven design process for *A*. 'R' stands for risk, 'O' for option. Ri represents the i^{th} first-level risk; RiOj represents design option j addressing risk i; RiOjRk represents residual risk k within design option RiOj, etc.

3.1 Initial Risk Analysis for AdsOL (*A*)

What are the initial risks *A* is exposed to?

- **R1. Not being paid.** *A* can lose money and even go out of business because it is not paid correctly for advertisements.
- **R2. Fine for illegal advertisement.** *A* may have to pay `IllegalFine` for an illegal advertisement.

R1. Not being paid
R1O1. *A* accepts external payments
option discarded: infrastructure, cost
R1O2. *K* accepts payment for *A*
R1O2R1. *K* not paying *A*
R1O2R1O1. *A* can prove issuing of a posting credential
option discarded: cannot be realized
R1O2R1O2. *A* can prove use of 'new' posting credential
option discarded: privacy impact too high

← **Change of assumption**: pay-per-subscription → pay-per-posting

R1O2R1O3. *A* can prove use of posting credential
***K*'s analysis** ← *K* requires one-use posting credential with double-spending risk for *A*
R1O2R1O3R1. Not being paid for double-spent posting credential
R1O2R1O3R1O1. Online double-spending detection
R1O2R1O3R1O2. Offline double-spending detection

R2. Fine for illegal advertisement
R2O1. No check
option discarded: cost (fines)
R2O2. Check by *A*
option discarded: cost (residual risk of fines)
R2O3. Check by *LE*
R2O3O1. Requested by user
R2O3O2. Requested by *A*

R3. Suspending operation
risk excluded by R2O3O2

Figure 3. Risk/Options Analysis for *A*

A also needs to determine the consequences of not being able to re-identify an illegal posting. In a real-world system, this may require an interaction with *LE*. We assume that the consequence is suspension of its operation:

- **R3. Suspending operation.**

For the various risks, we now describe the process of design options analysis, design decisions and residual risk analysis which will gradually refine the protocols.

3.2 Design Options Analysis

3.2.1 R1: Not being paid. *A* addresses this risk by not publishing an advertisement without having received a (guarantee of) payment for it. We consider the following design options:

- **R1O1. *A* accepts external payments.** *A* accepts bank transfers, credit card payments etc. from users.
- **R1O2. *K* accepts payment for *A*.** *K* accepts payments from users and issues posting credentials with which the users can post advertisements to *A*. *K,* of course, has to forward the payment for every new posting credential to *A*.

Assuming *A* prefers not to deal with external payments, we discard R1O1 and further explore R1O2. This decision is represented by underlining design option R1O2 in Figure 3.

Previously, we stated that *A* and *K* are different business entities. *A* thus has to take into account a remaining risk of *K* not paying *A* for every posting credential it issues:

- **R1O2R1. *K* not paying *A***

In order to address this risk, *A* wants a statement from *K* specifying *K*'s liabilities (e.g., payment to *A* upon issuing a posting credential); in addition, *A* wants to be able to prove when a new credential has been issued or is used:

- **R1O2R1O1. *A* can prove issuing of a posting credential**
- **R1O2R1O2. *A* can prove use of 'new' posting credential**

R1O2R1O1 cannot be realized as *A* cannot control *K*'s interactions with users. R1O2R1O2 requires that *A* can distinguish between multi-use credentials when they are shown. This can only be realized by introducing linkabilities between credential issuing and credential use; that would mean reducing users' privacy in order to solve a trust and liability problem between *K* and *A*!

Getting appropriate (proof of due) payment while respecting users' anonymity can only be realized in a model where *A* charges *K* for every advertisement posted; as *K* is only a payment intermediary, this naturally translates into a model where also users pay *K* per posting as opposed to per subscription.

At this point, we can either backtrack by reconsidering R1O1 which was previously discarded, or by changing our initial assumptions and investigating a pay-per-posting model. In such a model, it suffices that:

- **R1O2R1O3. *A* can prove use of posting credential**

We assume the pay-per-posting model is acceptable to both *A* and *K* and thus choose this option. As the change of assumption does not influence any of the earlier considerations or decisions, there is no need for backtracking.

At this point, we have to interleave our discussion with *K*'s independent risk and options analysis. *K* agrees to *A*'s conditions but of course decides to issue only one-use posting credentials and to not be liable for postings with a double-spent credential.

We now return to *A*'s analysis. *A*'s remaining risk is

- **R1O2R1O3R1. Not being paid for double-spent posting credential**

A now considers either on- or offline double-spending detection:

- **R1O2R1O3R1O1. Online double-spending detection**
- **R1O2R1O3R1O2. Offline double-spending detection**

An online double-spending check is expensive but reduces *A*'s risk to null. (Strictly speaking, the remaining risk for *A* is *K* not paying *A* even if *A* has proof of *K*'s liability to pay; this should be covered by *K*'s liabilities as already assumed when discussing R1O2R1.) In this case, however, an offline double-spending check may suffice. If a user double-spends a posting credential, he loses anonymity for both advertisements, as *K* would be able to link the nym revealed from double-spending detection to the user's identity through the identity escrow data. In addition, as soon as double-spending is detected, *A* can immediately cancel the publication of both advertisements paid for with the credential. Thus, a user can lose more than he can gain by double-spending a posting credential.

Because of the possibility to withdraw the advertisements of a double-spending user, *A* decides to go for the latter option. Then, for every posting credential correctly verified by *A, K* needs to either pay *A* or prove it was double-spent.

So far, we derived a partial solution to the design of *A,* taking into account the risk R1 of *A* not being paid. Of the two options for addressing this risk, the first was discarded because of cost reasons. Within the remaining option, we identified the remaining risk by examining trust assumptions between *A* and *K.* We chose to change to a pay-per-posting model as it allows to fulfill *A*'s security requirements while allowing more user privacy. *K*'s risk and options analysis then made *K* to accept being charged for one-use posting credentials without being

liable for their double-spending. Within the pay-per-posting model, we investigated the remaining risk (double-spending) and the various ways of addressing it. Between online and offline double-spending detection, the former is more expensive but excludes remaining risk. We chose, however, to implement the latter because it is less expensive while *A* has good reasons for accepting the residual risk.

We now apply a similar analysis to the remaining initial risk factors. R2 and R3 share a causal event, an illegal advertisement; R3's materialization depends in addition on a second event, *A*'s inability to provide the user's name.

In order to avoid unnecessary backtracking, we want to start with the risk analysis (R2 or R3) which results in the strongest measures against (or: strongest measures preventing) illegal advertisements. If providing the user's name were deemed impossible, R3, being the more severe risk, would lead us to implement the stronger measures; if providing the user's name were trivial or could be realized with no cost to *A,* R2 would provide us with the stronger measures. In the absence of any of these assumptions, we start with the analysis of R2.

3.2.2 R2: Fine for illegal advertisement (`IllegalFine`). *A* can address this risk in various ways:

- **R2O1. No check.** *A* can choose not to address this risk; any message is publicized without screening. This option is discarded because it is deemed too exensive (`IllegalFine`).

- **R2O2. Check by *A*.** *A* can prevent illegal advertisements by screening and approving every message before it is posted. This is expensive and can prevent most illegal advertisements; however, *A* has no guarantee that an advertisement which it considers legal is not considered illegal by *LE*; also this residual risk is deemed too high to accept.

- **R2O3. Check by *LE*.** Illegal advertisements can be excluded if every message, before being published, is approved by *LE* himself. This is a very expensive solution but completely excludes *A*'s risk if *LE*'s approval is provable (e.g., by a signature); also, given that excluding illegal messages also excludes any re-identification requirements in the system and thus is attractive to users, *LE* may consider charging the extra cost for this solution to the users.

  - **R2O3O1. Requested by user.** *A* can require that users only post messages if approved (with a signature) by *LE*;

U obtaining posting credential from K:
$K \rightarrow U$: Nym_{UK}
$U \rightarrow K$: payment (linked to Nym_{UK})
$K \rightarrow U$: $\text{Cred}_{UK} = \text{Cred}(\text{Nym}_{UK}, \text{IPKey}_K, \texttt{false}, \text{Exp})$

U obtaining approval from LE:
$U \rightarrow LE$: `Advert`, possibly payment
$LE \rightarrow U$: $S_{LE}(\texttt{Advert})$

U posting advertisement to A:
$U \rightarrow A$: $\text{SigCred}_{UA} = \text{SigCred}(\text{Cred}_{UK}, S_{LE}(\texttt{Advert}), \texttt{null}, \texttt{null})$

Contracts and liabilities:
K: 'I will pay A every time a Cred_{UK} is shown unless I can prove it is double-spent'
A: 'For any advertisement which is considered illegal but not approved by LE, I will pay `IllegalFine` and suspend operation if I cannot provide a mapping with a user'

Figure 4. AdsOL Risk-Driven Design: Protocols and Liabilities

– **R2O3O2. Requested by *A*.** *A* himself can contact *LE* for every advertisement posted and charges the extra cost to the user in the form of a higher price per advertisement.

We now assume that users are willing to accept the burden of contacting *LE* before posting a message to *A*, as this guarantees them unconditional anonymity of the posting. (Of course, users require the exchange with *LE* to be anonymous as well!) By choosing R2O3O1, we have completely excluded risk R2 as well as R3.

3.2.3 R3: Suspending Operation. The risk of suspending operation has thus been excluded by excluding illegal message postings.

3.3 Resulting Design

The resulting design is depicted in Figure 4. In order to obtain a one-use (MultiSig=`false`) posting credential, *U* establishes a nym Nym_{UK} with *K* and performs a payment. The implementation should ensure that the payment is linked to Nym_{UK}, e.g., by including Nym_{UK} in the signed payment message. Before posting `Advert` to *A*, *U* requests a signed approval on `Advert` from *LE*. *LE* may request a payment for this service as well. *U* then posts the approved $S_{LE}(\texttt{Advert})$ to *A* in SigCred_{UA}, now without deanonymization.

The contracts and liabilities represent the observations we made during the above analysis. *K*'s liability expresses his financial obligation

towards *A* related to a posting with a non-double-spent Cred_{UK}. *A*'s liability towards *LE* expresses payment of `IllegalFine` for an illegal advertisement, and *A's* obligation to suspend operation if the corresponding transaction cannot be re-identified. By not taking liability for illegal advertisements approved by *LE* and by not accepting non-approved advertisements, *A* can however exclude these risks.

3.4 Generalization of the Risk-Driven Design Approach

Our risk-driven design of the AdsOL application allowed us to identify a more privacy-friendly solution, although at a potentially higher cost. It also allowed us to identify hidden trust assumptions (between *A* and *K)* in our previous design which could have led to undefined liabilities in the event of failed re-identification. We now capture the principles of the approach.

In the above risk-driven design, we have started out with a risk analysis for the entity to be designed *(A).*

A risk which does not involve liabilities towards another party is an **internal risk.** *A* not being paid for the service it provides is an internal risk.

An **external risk** is related to a liability towards another party. *A*'s risk to have to pay `IllegalFine` is an external risk. Such risks are typically captured in contracts or liabilities as shown in Figure 4.

A risk can be dealt with in different ways, leading to different design options. Of these options, we can assess trust assumptions, cost and privacy features; these are taken into account when choosing a design option.

Depending on the option chosen, a risk can then be:

- accepted: *A* accepts the risk of revenue loss due to double-spending.
- transformed (delegated) into a risk for another party: *A* transforms R1 into a payment liability by *K* towards *A*.
- prevented or excluded: a loss of revenue because of a double-spent posting credential can be prevented by an online double-spending check. Note that, what hat seems as prevention, may often be a transformation: with or without online double-spending detection, the initial risk of not being paid (by the user) is only transformed into the risk of not being paid by *K*; the protocols chosen merely ensure that *A* can hold *K* liable in case *K* refuses to pay *A* (as expressed with *K*'s liability statement).

The risk/options analysis is an iterative process as a residual risk analysis leads to a new analysis of design options.

In the AdsOL example, we avoided backtracking through design options in order to keep the simplicity of the example. In general, however, the design process tries to optimize a function of cost, privacy and risk. A systematic risk/options analysis is thus likely to involve backtracking; options should be discarded only if they cannot lead to an optimized solution.

The description of the approach presented here is informal. Clearly, development of a real methodology for the risk-driven design needs a formalization of the risks, of the options addressing those risks and of the measures for evaluating them:

- Risks have to be expressed in terms of the occurrence of events; if a risk is external, provability of events plays an important role. E.g., risk R2 can be stated as the existence of a proof of existence, on *A*'s web site, of an illegal message posting for which *A* cannot prove approval by *LE*.

- Options addressing a risk can be stated in terms of minimizing or excluding the occurrence of these events. E.g., *A*'s addressing R2 (and R3) consists of preventing anyone to be able to prove the above.

- The analysis of the various design options has to be done based on measurable criteria, e.g., cost estimates of risks and design options.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented and illustrated a risk-driven approach towards designing privacy-friendly and secure applications. Fundamental principles of the approach are the identification of risks without hidden trust assumptions; and a thorough analysis of design options for addressing risks with a focus on prevention.

Our risk-driven design approach provides for a conceptual framework facilitating the correct addressing of security requirements resulting from real risks while enabling identification of the most privacy-friendly solution. However, the development of a real methodology needs a formalization of the risks, of the options addressing those risks and of the measures for evaluating them. These are topics for further research.

References

[1] T. Aura and C. Ellison. Privacy and accountability in certificate systems. Research Report HUT-TCS-A61, Helsinki University of Technology Laboratory for Theoretical Computer Science, 2000.

[2] M. Blaze, J. Feigenbaum, and A. D. Keromytis. Keynote: Trust management for public-key infrastructures (position paper). In *Proc. 1998 Security Protocols Workshop,* volume 1550 of *Lecture Notes in Computer Science,* pages 59–63. Springer-Verlag, 1998.

[3] J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proc. 2002 ACM Conference on Computer and Communications Security.* ACM Press, 2002.

[4] D. Chaum and J.-H. Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. In *Advances in Cryptology – CRYPTO '86,* volume 263 of *Lecture Notes in Computer Science,* pages 118–167. Springer-Verlag, 1987.

[5] D. L. Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM,* 28(10):1030–1044, 1985.

[6] L. Chen. Access with pseudonyms. In *Cryptography: Policy and Algorithms,* volume 1029 of *Lecture Notes in Computer Science,* pages 232–243. Springer Verlag, 1995.

[7] I. B. Damgård. Payment systems and credential mechanism with provable security against abuse by individuals. In *Advances in Cryptology – CRYPTO '88,* volume 403 of *Lecture Notes in Computer Science,* pages 328–335, 1990.

[8] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B.Thomas, and T. Ylonen. SPKI certificate theory. RFC 2693, Sept. 1999.

[9] International Telecommunication Union. ITU-T recommendation x.509 - the directory: Authentication framework, Aug. 1997.

[10] S. Konrad, B. Cheng, L. Campbell, and R. Wassermann. Using security patterns to model and analyze security requirements. In *International Workshop on Requirements for High Assurance Systems (RHAS),* 2003.

[11] N. Li, B. Grosof, and J. Feigenbaum. A practically implementable and tractable delegation logic. In *Proc. 2000 IEEE Symposium on Research in Security and Privacy,* pages 27–42. IEEE Computer Society Press, 2000.

[12] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In *Selected Areas in Cryptography,* volume 1758 of *Lecture Notes in Computer Science.* Springer-Verlag, 1999.

[13] The Open Group. Technical guide: Security design patterns, Apr. 2004.

[14] E. Van Herreweghen. Designing anonymous applications with accountability using anonymous credentials. Research Report RZ 3526, IBM Research Division, Jan. 2004.

[15] A. van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Proc. IEEE International Symposium on Requirements Engineering,* pages 249–262, 2001.

[16] E. Yu and L. Cysneiros. Designing for privacy and other competing requirements. In *2nd Symposium on Requirements Engineering for Information Security (SREIS'02),* Raleigh, North Carolina, Oct. 2002.