Rethinking Accountable Privacy Supporting Services

Extended Abstract

Jan Camenisch
IBM Research, Zurich
Research Laboratory
Rüschlikon, Switzerland
jca@zurich.ibm.com

Thomas Groß
IBM Research, Zurich
Research Laboratory
Rüschlikon, Switzerland
tgr@zurich.ibm.com

Thomas S.
Heydt-Benjamin
IBM Research, Zurich
Research Laboratory
Rüschlikon, Switzerland
hey@zurich.ibm.com

ABSTRACT

As privacy concerns among consumers rise, service providers will increasingly want to provide services that support privacy enhancing technologies. At the same time, providers of commercial services require the security of identifying misbehaving users. For instance, users that do not pay their bill can be held accountable for their behavior. We propose a scheme that permits privacy support while retaining accountability. In our proposed scheme an honest user may enjoy full anonymity, but dishonest users who do not pay their bill have their identity revealed. In contrast to existing revocable anonymity systems, our proposed scheme requires less trust in an external authority, while simultaneously making accountability easier (and less costly) to achieve. We contribute the concept of a time capsule, that is, a verifiable encryption with timed and revocable decryptability.

Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption—*Public key cryptosystems*

General Terms

Security

Keywords

Privacy, Accountability, Anonymous Credential Systems, Cryptographic Protocols, Time Capsule, Verifiable Encryption

1. INTRODUCTION

Services with identity management relations have a challenging set of requirements. They require accountability of identity data as well as privacy protection of their users. The first requirement is usually driven by the need to mitigate risk to the service provider, and the latter requirement driven by regulatory compliance (e.g., privacy act). We introduce new primitives for accountable and privacy-preserving identity management methods for services that provide improved support for these two requirements.

Anonymous credential systems exist that provide methods to achieve a wide range of accountability and privacy goals for services. In particular anonymous credentials systems already have the capability to selectively disclose statements about a user, e.g., proving the user's age without revealing the user's actual date of birth. With the complementary primitive of verifiable encryption [**?**], credential systems can also provide accountability without infringing on privacy requirements. For instance, a user U can encrypt her true identity to the authorities, provide this encrypted data to a service provider SP, and convince SP in a zero-knowledge proof of knowledge that this encrypted data contains a valid user identity that can be opened by the authorities.

Even though accountable and privacy-preserving identity management methods for services are already possible, we propose to rethink the corresponding primitives to better meet the actual needs of services.

We propose an improvement of such a system's flexibility and trust model. Prior systems realized accountability with verifiable encryption by encrypting the user's identity to a trusted third party (TTP), for instance, law enforcement authorities. This presents three challenges:

1. This mode of operation involves a fully trusted TTP with no graceful degradation of privacy and security should the TTP become compromised.
2. A malicious service provider holding a verifiable encryption may attempt to betray the user by opening a decryption case at the TTP without good cause.
3. Honest service providers find the traditional system encumbering because of the need to involve such highly trusted authorities for even minor dispute cases. For example, to bring a case to law enforcement in the real world is likely to have a non-trivial cost, both in the time required, and in support from legal council.

We propose what we believe to be the first verifiable encryption system to provide revocation of decryptability with end-to-end unlinkability. This simultaneously provides stronger privacy for the user, and a lower cost to the service provider for obtaining accountability when the user misbehaves.

We will show that these properties provide a system that addresses all three of the above mentioned challenges in existing privacy supporting service provision. Greater detail follows in the body of this paper, but at a high level:

1. Our revocation authority RA is a weaker TTP that cannot link user's transactions within the system. If the RA becomes compromised it is still restricted in what information it can reveal.

DIM'08, October 31, 2008, Fairfax, Virginia, USA.

- RA cannot learn any information about the user before the user's bill is past due.
- RA processes only blinded information.
  When an anonymity revocation is requested by a service provider, RA only knows that it is checking the key for a legitimate transaction, without knowing which transaction or which user. Therefore, RA cannot block requests selectively or collude against any specific user.
- Even when the bill is past due, and the user has not paid it, RA cannot link this fact to any particular contract or user.

2. Our system contains a mechanism for verifiable, yet privacy supporting, proof of fulfillment of the contracted terms of the service. The RA can easily detect an unfounded request for opening an encrypted identity, and will not service such requests.

3. Our system permits automatic identity revocation in the event that a contract is not fulfilled by the user. Because the contract satisfaction condition can be machine verified, external authorities such as law enforcement do not need to be involved.

We achieve these goals by introducing a special kind of verifiable encryption called a time capsule, represented by Θ. The time capsule has the following properties:

- Θ is issued by the user to the service provider at time α and set not to open till time event ω.
- The service provider is convinced in zero knowledge that the time capsule contains the expected piece of data (say, the user's certified identity)
- Beginning with time ω, the service provider can decrypt the time capsule (say, if after a payment grace period of a month).
- If the user fulfills a revocation condition before time event ω, the service provider cannot decrypt the time capsule (say, if the user has fulfilled her payment obligation).

Note that time may be real clock time, or any other ordered sequence of events which may be determined by the system implementation.

The time capsule improves on the general verifiable encryption primitive by becoming sensitive to external events. For time events, the verifiable encryption becomes capable of implementing statements of linear temporal logic.

Our contributions in this paper are:

1. Propose the time capsule, and the related transactions.
2. Show that the time capsule permits identification of a user if and only if they fail to fulfill a contract by the end of the grace period.
3. Contribute an analysis of possible trust models for such a system and a realization of the time capsule with a relaxed trust model compared to the traditional verifiable encryption use.

2. PROBLEM STATEMENT

2.1 Concrete Scenario

We pursue our discussion of accountable and privacy-preserving identity management for services with a concrete example to illustrate the needs of different entities, and how those needs may be met. Let us define the following principles:

Identity Provider (IDP) issues identity credentials to a user. The IDP could be a passport authority or similar. This entity is well described by existing literature such as [**?**][**?**][**?**] and will not be discussed further here.

Service Provider (SP) provides privacy supporting services to users.

User (U) holds identity credentials issued by the IDP and attempts to use a contracted service of SP.

Time server (TS) issues a stream of time symbols t_i at regular intervals.

Satisfaction Authority (SA) the authority that determines when the satisfaction condition of a contract is fulfilled and issues a satisfaction token. For example, in the case when the satisfaction condition is payment of a bill, SA could be a bank, and the satisfaction token would be a receipt with which to prove payment.

Revocation Authority (RA) the authority that manages the keypairs used to encrypt and decrypt time capsules. U may communicates with this authority in order to encrypt a time capsule, and SP must communicate with RA in order to attempt opening of the time capsule.

Bulletin Board (BB) A public forum in which items posted are available for anyone to download. The BB in our system is used to post satisfaction tokens by which the user proves payment.

In our scenario, a user U has a relationship with an identity provider IDP from which it has obtained identity credentials. U wants to anonymously access a service provided by SP. The SP permits anonymous or pseudonymous contracts for service, yet, wants to ensure that he can hold the user accountable in cases of dispute. SP requires a dead pledge in the form of a *time capsule* Θ through which he can obtain U's true identity if U fails to pay for the service. Thus, the dead pledge Θ can be opened making U's identity accessible to SP after a well-defined grace-period. U on the other hand has the requirement that all her transactions shall be unlinkable provided that contracted obligations are fulfilled before the end of this grace period. Once U has fulfilled her obligations the service provider shall no longer be able to learn her true identity.

2.2 Trust Model

TS is trusted to:

- Release time symbols t_i sequentially and on time
- Not reveal any t_i for an i that has not yet occurred.

RA is trusted to:

- Not reveal it's own secret key sk_{RA}
- Not reveal any secret time capsule key sk_{Θ} if it detects that the time capsule's contracted revocation condition has occurred.

- Note that we do *not* need to trust that the RA will not reveal sk_Θ before the contract has expired, this is cryptographically enforced.

IDP is trusted not to reveal its secret key.

SA is trusted to honestly report when payment has occurred.

BB is trusted to report all items posted on the bulletin board (although this can also be checked by the user)

2.3 Requirements

Verifiability At the conclusion of an enrollment transaction, the service provider receives a time capsule Θ. The service provider must be able to securely determine the following before service is granted:

1. Θ contains an identity certified by IDP.
2. Θ can be opened by SP if payment has not been made by time t_ω.

End to End Unlinkability If the revocation condition occurs before t_ω then no principal will be able to link the user's transactions together nor link any transaction to a specific user. This is a stronger notion of privacy than currently offered in existing systems.

Accountability If the user does not fulfill the revocation condition before time ω, the SP must be able to open Θ and retrieve U's identity

Privacy

- Before the predetermined release time ω no principal can decrypt Θ.
- If the user fulfills the revocation condition before time ω, neither the SP nor any other principal will be able to open the sealed identity.
- If the user does not fulfill the revocation condition before time ω, only SP can open Θ.

2.4 System Definition

We define a time capsule system to be a system that supports the following transactions while obeying the requirements of §2.3

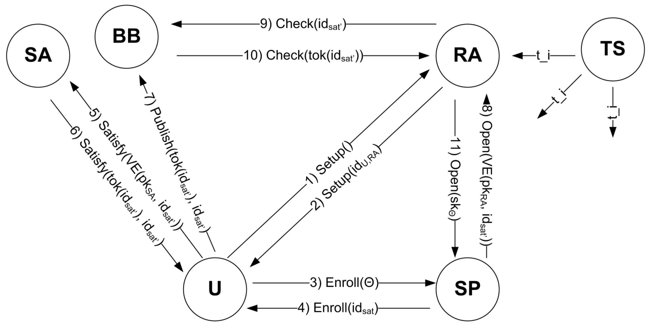
SA
BB
RA
TS
U
SP
9) Check($id_{sat'}$)
10) Check(tok($id_{sat'}$))
t_i
t_i
t_i
5) Satisfy(VE(pk_{SA}, $id_{sat'}$))
6) Satisfy(tok($id_{sat'}$), $id_{sat'}$)
7) Publish(tok($id_{sat'}$), $id_{sat'}$)
1) Setup()
2) Setup($id_{U,RA}$)
11) Open(sk_Θ)
8) Open(VE(pk_{RA}, $id_{sat'}$))
3) Enroll(Θ)
4) Enroll(id_{sat})

Figure 1: This collaboration diagram depicts the flow of information between the various principles during the transactions defined in §2.4

Setup:

$(\mathsf{U}(id_{U,RA})) \leftarrow Setup(\mathsf{U}, \mathsf{RA})$

RA issues a unique pseudonym $id_{U,RA}$ to user U, which can be used for multiple service contracts within the system. This value is used only in blinded form, thus multiple transactions with this same pseudonym are unlinkable.

Enrollment:

$(\mathsf{U}(id_{sat}), \mathsf{SP}(\Theta)) \leftarrow Enroll(\mathsf{U}(id_{U,RA}, id_{U,IDP}), \mathsf{SP})$

During enrollment the user makes a contract with the service provider, receives a token that permits use of the service, and leaves the user with a mechanism to prove fulfillment of the contract (payment). It leaves the service provider with a time capsule Θ that can be opened later to identify the user if the contract is not fulfilled.

Satisfaction of Revocation Condition:

$(\mathsf{U}(\mathsf{tok}(id_{sat}'), id_{sat}'), \mathsf{SA}(id_{sat}')) \leftarrow Satisfy(\mathsf{U}(id_{sat}'), \mathsf{SA}(sk_{SA}))$

The user invokes this protocol to fulfill her contract (pay her service bill). The satisfaction authority (the bank) provides a secure receipt that the user later publishes to prove payment of the bill. The receipt is blinded such that it can prove payment for the transaction without revealing the identity of the user.

Publish:

$() \leftarrow Publish(\mathsf{U}(\mathsf{tok}(id_{sat}'), id_{sat}'), \mathsf{BB})$

By publishing the receipt of payment the user prevents the service provider from being able to reveal her identity.

Check:

$(\mathsf{RA}(\mathsf{tok}(id_{sat}'))) \leftarrow Check(\mathsf{RA}(id_{sat}'), \mathsf{BB}(id_{sat}'))$

The revocation authority RA checks for revocation of Θ by searching the BB for the secure receipt, and verifying that it is a correct payment for the correct bill. If a receipt is found and verified then the check succeeds, indicating that revocation has occurred and the contract is complete. In other words, if check succeeds then Θ should never be opened.

Open:

$(\mathsf{SP}(sk_\Theta)) \leftarrow Open(\mathsf{SP}(\Theta), \mathsf{RA}(sk_{RA}), \mathsf{TS}(T_i))$

In the open transaction the service provider attempts to gain knowledge of the user's real identity. The SP requests this knowledge by providing the revocation authority with a designated portion of the time capsule. If the user has not payed their bill, and the bill is overdue, the revocation authority will be able to extract the secret key that unlocks the rest of the time capsule. If the user has honestly paid her bill, however, this transaction will not succeed, and the service provider learns nothing.

3. IMPOSSIBILITY RESULT

Given the trust model in §2.2, we perceive that an external trusted entity is required. We consider three arguments for necessary conditions for a timed and revocable verifiable encryption. We analyze the requirement of a trusted clock, a revocation authority, and the nature of their information flow.

First, let us analyze the need for a trusted clock. As we do not trust the service provider, it may manipulate its local clock. Thus, the system requires at least a trusted clock publishing user-independent immutable time events.[1] This result accounts for the involvement of the time server TS and the value $\mathsf{TS}(t_i)$ in the $Open()$ transaction. It is a *necessary* condition for any system with the given privacy requirements (§2.3). For a time capsule system that only permits decryption after time event ω has passed, this is *sufficient*.

[1] This may be either realized by an explicit time server TS as defined in the system interface of §2.4 or by a distributed system of trusted clock components at each principal.

Second, we turn to the requirement of a partially trusted revocation authority. A trusted clock solely publishing user-independent time events is not sufficient for a system that accepts a revocation condition. Let us assume there exist two systems with identical state apart from the fact that a user's verifiable encryption is revoked in one system and valid in the other. Thus, the information obtained by the service provider in the $Open()$ transaction must convey at least one bit of information indicating whether a user's verifiable encryption is revoked or not. If the service provider SP interacts solely with the user-independent time server TS, the system cannot fulfill this *necessary* condition. By this information flow argument, we realize the need to assume an additional principal RA that

- holds a revocation state, and
- creates a decryption event that is impacted by user-specific information flow.

Third, we can also determine the nature of the information flow to the service provider SP by a generic argument. Because we do not trust the service provider SP, we assume it may ignore arbitrary pieces of information or rewind its internal state. The piece of information provided by the principal RA in the $Open()$ is therefore *necessary* for the decryption of the verifiable encryption. That is, it must contain a secret piece of information without which the verifiable encryption cannot be decrypted up to the security of the underlying encryption method. The service provider SP could otherwise neglect this piece of information and decrypt the verifiable encryption anyway.

It therefore is a *necessary* condition that a trusted entity governs the revocation process and holds imperative decryption information confidential.

4. CRYPTOGRAPHIC BUILDING BLOCKS

The properties of our system our built on top of several cryptographic primitives. We present them here in overview so that it can be seen how their properties combine in our protocols to fulfill the requirements of our problem statement. We specify them in two forms of notations: (a) as abstract method interfaces for higher cryptographic primitives, such as

$$Com(v_1, \ldots, v_m; r)$$

for an integer commitment; and (b) as abstract definition of a Zero-knowledge Proof of Knowledge introduced by Camenisch and Stadler [?], e.g.,

$$PK\{(\alpha) : y = g^\alpha\}$$

for a zero-knowledge proof of an integer α that has the property $y = g^\alpha$.

First, we lay the foundations with the Strong RSA assumption, integer commitments, and Zero-Knowledge Proofs of Knowledge. Second, we introduce the Camenisch-Lysyanskaya signature scheme, Committed Blind Anonymous IBE, and Verifiable Encryption as high-level cryptographic primitives.

4.1 Assumptions

Strong RSA Assumption [?, ?]: Given an RSA modulus n and a random element $g \in \mathbb{Z}_n^*$, it is hard to compute $h \in \mathbb{Z}_n^*$ and integer $e > 1$ such that $h^e \equiv g \bmod n$. The modulus n is of a special form pq, where $p = 2p' + 1$ and $q = 2q' + 1$ are safe primes.

4.2 Integer Commitments

Recall the Pedersen commitment scheme [?], in which the public parameters are a group G of prime order q, and generators $(g_0, \ldots, g_m)$. In order to commit to the values $(v_1, \ldots, v_m) \in \mathbb{Z}_q^m$, pick a random $r \in \mathbb{Z}_q$ and set $C = \mathsf{Com}(v_1, \ldots, v_m; r) = g_0^r \prod_{i=1}^m g_i^{v_i}$.

Damgård and Fujisaki [?] show that if the group G is an RSA group and the committer is not privy of the factorization of the modulus, then in fact the Pedersen commitment scheme can be used to commit to *integers* of arbitrary size.

We specify an abstract interface for integer commitments:

$Com(v_1, \ldots, v_m; r)$ for values v_i and a random number r produces an integer commitment C on the values v_i

4.3 Known Discrete-Logarithm-Based, Zero-Knowledge Proofs

In the common parameters model, we use several previously known results for proving statements about discrete logarithms, such as (1) proof of knowledge of a discrete logarithm modulo a prime [?] or a composite [?, ?], (2) proof of knowledge of equality of representation modulo two (possibly different) prime [?] or composite [?] moduli, (3) proof that a commitment opens to the product of two other committed values [?, ?, ?], (4) proof that a committed value lies in a given integer interval [?, ?, ?, ?], and also (5) proof of the disjunction or conjunction of any two of the previous [?]. These protocols modulo a composite are secure under the strong RSA assumption and modulo a prime under the discrete logarithm assumption.

When referring to the proofs above, we will follow the notation introduced by Camenisch and Stadler [?] for various proofs of knowledge of discrete logarithms and proofs of the validity of statements about discrete logarithms. For instance,

$$PK\{(\alpha, \beta, \delta) : y = g^\alpha h^\beta \wedge \tilde{y} = \tilde{g}^\alpha \tilde{h}^\delta \wedge (u \le \alpha \le v)\}$$

denotes a "*zero-knowledge Proof of Knowledge of integers α, β, and δ such that $y = g^\alpha h^\beta$ and $\tilde{y} = \tilde{g}^\alpha \tilde{h}^\delta$ holds, where $u \le \alpha \le v$,*1" where $y, g, h, \tilde{y}, \tilde{g}$, and $\tilde{h}$ are elements of some groups $G = \langle g \rangle = \langle h \rangle$ and $\tilde{G} = \langle \tilde{g} \rangle = \langle \tilde{h} \rangle$. The convention is that Greek letters denote quantities of which knowledge is being proven, while all other values are known to the verifier. We apply the Fiat-Shamir heuristic [?] to turn such proofs of knowledge into signatures on some message m; denoted as, e.g., $SPK\{(\alpha) : y = g^\alpha\}(m)$.

Given a protocol in this notation, it is mostly straightforward to derive actual protocol implementing the proof. Indeed, the computational complexities of the proof protocol can be easily derived from this notation: basically for each term $y = g^\alpha h^\beta$, the prover and the verifier have to perform an equivalent computation, and to transmit one group element and one response value for each exponent. With statement such as $(u \le \alpha \le v)$ we denote only these interval check that are basically for free [?, ?, ?] and are not tight (but good enough if the non-tightness can be accounted for as in our application). We note that this exclude the interval proof protocol as the one by [?] that are tight but computationally costly, i.e., they require the prover to provide a number of so-called integer commitments and to prove relations among them. If we require such a proof, we will give the full protocol specification.

4.4 CL Signatures

Let us recall the Camenisch-Lysyanskaya signature scheme [?]. We introduce their high-level principles in this section and refer to the Appendix §A for an rigorous definition. For this papers contribution we do not depend on a deep understanding of the internals of CL-Signatures, but leverage them as high-level building block.

It consists of two secure two protocols:

(1) An efficient protocol between a user and a signer with keys $(pk_{\mathsf{IDP}}, sk_{\mathsf{IDP}})$. The common input consists of pk_{IDP} and C, a Damgård and Fujisaki commitment as introduced in §4.2. The user's secret input is the set of values $(v_1, \ldots, v_\ell, r)$ such that $C = Com(v_1, \ldots, v_\ell; r) \pmod{n}$. As a result of the protocol, the user obtains a signature $\sigma_{pk_{\mathsf{IDP}}}(v_1, \ldots, v_\ell)$ on his committed values, while the signer does not learn anything about them.

(2) An efficient proof of knowledge of a signature protocol between a user and a verifier. The common inputs are pk_{IDP} and a commitment C. The user's private inputs are the values $(v_1, \ldots, v_\ell, r)$, and $\sigma_{pk_{\mathsf{IDP}}}(v_1, \ldots, v_\ell)$ such that $C = Com(v_1, \ldots, v_\ell; r)$. These signatures are secure under the strong RSA assumption (§4.1).

Camenisch-Lysyanskaya use the integer commitment primitive $Com()$ introduced in §4.2 and be accessed by zero-knowledge proofs of knowledge in the Camenisch-Stadler notation (§4.3). We provide an additional interface to the scheme:

$\mathit{CLSetup}(1^k)$ outputs parameters parCL.

$\mathit{CLKeyGen}(\mathit{parCL})$ outputs a public key/private key keypair $(pk_{\mathsf{IDP}}, sk_{\mathsf{IDP}})$.

$\mathit{CLSign}(\mathsf{U}(pk_{\mathsf{IDP}}, C, (v_1, \ldots, v_\ell, r)), \mathsf{IDP}(pk_{\mathsf{IDP}}, sk_{\mathsf{IDP}}, C))$ outputs a CL signature on the user's values $(v_1, \ldots, v_\ell)$ to the user U and nothing to the IDP, iff $C = Com(v_1, \ldots, v_\ell, r)$.

$\mathit{CLVerify}(\mathsf{U}(pk_{\mathsf{IDP}}, C, (v_1, \ldots, v_\ell, r), \sigma_{pk_{\mathsf{IDP}}}(v_1, \ldots, v_\ell)), \mathsf{V}(pk_{\mathsf{IDP}}, C))$ outputs valid to the verifier V and nothing to the user U iff $C = Com(v_1, \ldots, v_\ell, r)$. Otherwise, it outputs $\perp$ to both principals.

Note that the later transaction models a high-level view of a simple proof of possession for a CL-Signature between a user U and a verifier V. There is an equivalence between this view and the corresponding Camenisch-Stadler notation of a CL signature proof as shown in Appendix §A:

$$\mathit{CLVerify}(\mathsf{U}(pk_{\mathsf{IDP}}, C, (v_1, \ldots, v_\ell, r),$$

$$\sigma_{pk_{\mathsf{IDP}}}(v_1, \ldots, v_\ell)), \mathsf{V}(pk_{\mathsf{IDP}}, C))$$

$$\Longleftrightarrow$$

1.Compute blinded CL-signature (A', e, v'):

Choose random $r_A$of sufficient length.

$$A' = S^{r_A}; v' = v - er_A.$$

$$2.\mathit{PK}\{(\varepsilon, \nu', \mu) : \quad Z \equiv \pm R_0^{\mu_0} \cdots R_\ell^{\mu_\ell} A'^{\varepsilon} S^{\nu'} \pmod{n} \wedge$$
$$\mu_i \in \pm\{0,1\}^{\ell_m} \;\wedge\; \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}$$

4.5 Committed Blind Anonymous IBE

An *Identity-based Encryption* (IBE) scheme is an encryption scheme that allows the public key of a recipient to be an identity string id. Examples could be the recipient's e-mail address, yet, also the identifier of a service transaction. IBE schemes rely on a trusted third party, the *Key Generation Center* (KGC), to derive the user's secret key for an identity sk_{id}. The function to generate this secret key from the identity string is called $\mathit{IBEExtract}()$.

We leverage an IBE scheme with two additional properties: anonymity and blind extraction. Anonymous IBE as introduced by Abdalla et al. [?] ensures that it is infeasible to derive the identity string id to which the message was encrypted from the ciphertext. Green and Hohenberger [?] introduced blind extract IBE, which provides the capability to generate the secret key to an identity sk_{id} in a blinded fashion. Where normal IBE schemes transfer the identity string in plain text to the Key Generation Center, Blind Extract IBE keeps the identity id confidential from it.

Camenisch, Kohlweiss, Durán, and Sheedy [?] propose an IBE primitive that combines both properties. Their *Committed Blind Anonymous IBE* allows for a blind extraction of the secret key sk_{id} by giving the Key Generation Center a commitment on the identity string. The user of the system may either disclose partial information about the identity string or prove statements with efficient Zero-Knowledge Proofs of Knowledge about the commitment. This may, for instance, convey statements about linear relations [?] or range proofs [?]. Clearly, such proofs can be easily combined with proofs over attributes certified by a Camenisch-Lysyanskaya signature as introduced in §4.4.

We refer to the system interface of Camenisch, Kohlweiss, Durán, and Sheedy [?]:

$\mathit{IBESetup}(1^k)$ outputs parameters parIBE and master secret msk.

$\mathit{IBEBlindExtract}(\mathsf{A}(\mathit{parIBE}, id, \mathit{open}), \mathsf{KGC}(\mathit{msk}, C))$ generates the secret decryption key sk_{id} for a user A's identity id in an interactive key issuing protocol between A and the KGC. If $C = Com(id, \mathit{open})$, then the user's output is a decryption key sk_{id} and the output of the KGC is empty. Otherwise both parties output $\perp$.

$\mathit{IBEEnc}(\mathit{parIBE}, id, m)$ outputs ciphertext ct encrypting m under id.

$\mathit{IBEDec}(\mathit{parIBE}, sk_{id}, ct)$ outputs message m encrypted in ct.

4.6 Verifiable Encryption

Verifiable encryption schemes allow for proving properties about encrypted data. In their current form, they were introduced by Camenisch and Shoup [?], who provided the first efficient construction without resorting to cut-and-choose proofs. Their contribution focuses on discrete-log problems. It can be combined with Pedersen's as well as Damgård and Fujisaki's commitment schemes [?, ?] and the Camenisch-Lysyanskaya signature scheme introduced in §4.4.

Suppose a principal T owns a public key/secret key pair $(pk_{\mathsf{T}}, sk_{\mathsf{T}})$. Suppose further that A encrypts a message m under the public key pk_{T}, derives a ciphertext ct and sends it to B. A can prove a statement about the encrypted m in an efficient Zero-Knowledge Proof of Knowledge to B, while B cannot decrypt the ciphertext ct on its own. B may however gain confidence by A's proof that the ciphertext ct contains a valuable piece of information that principal T is able to retrieve. For instance, A may encrypt its own true identity id_{A} under the public key pk_{T}. It may prove in zero-knowledge that the ciphertext contains the very same identity string as certified in a credential A possesses. Then a receiving service B can be confident that it can have T decrypt the true identity of A if a dispute occurs.

We use the following interface for Camenisch and Shoup's construction:

$\mathit{VESetup}(1^k)$ outputs parameters parVE and a public key/secret key pair $(pk_{\mathsf{T}}, sk_{\mathsf{T}})$.

$\mathit{VEEnc}(pk_{\mathsf{T}}, m)$ Encrypts the message m to the public key pk_{T}.

$\mathit{VEDec}(sk_{\mathsf{T}}, ct)$ Decrypts the ciphertext ct with secret key sk_{T}.

The proof statements about properties of encrypted messages are orthogonal to the verifiable encryption primitive. We can use the same efficient Zero-Knowledge Proof of Knowledge mechanisms as for proofs over commitments, Camenisch-Lysyanskaya signatures (§4.4), or Committed Blind Anonymous IBE identity strings (§4.5).

5. SOLUTION OVERVIEW

We present an example cryptographic implementation of our system using building blocks from anonymous credentials systems such as ([**?**][**?**][**?**]) and the relatively new *committed blind anonymous identity based encryption* [**?**]. We describe here at a high level how these building blocks can implement a system with the requirements and definitions listed in §2. More details about the properties of the building blocks are explained in §4.

System Setup: RA acting as an IBE KGC and a participant in a verifiable cryptosystem performs $\mathit{IBESetup}(1^k)$ and $\mathit{VESetup}(1^k)$. RA publishes the resultant public information $(\mathit{parIBE}, \mathit{parVE}, pk_{SA})$. Hereafter we will implicitly assume that these public parameters are available to all principals.

Setup: $\mathit{Setup}(\mathsf{U}, \mathsf{RA}) \rightarrow id_{U,RA}$
U is issued a unique pseudonym $id_{U,RA}$ by RA. This pseudonym will form a part of the time capsule's key. By using committed blind anonymous IBE the pseudonym will be later used only in blinded form, preventing linkability between transactions.

Enrollment: $\mathit{Enroll}(\mathsf{U}(id_{U,RA}, id_{U,IDP}), \mathsf{SP})$

1. U and SP jointly compute a random transaction identifier id_{sat} which is then known to both parties, and a blinded form of the same identifier known to neither U or SP, but rather encrypted to SA and RA respectively: $\mathit{VEEnc}(pk_{SA}, id_{sat}')$, $\mathit{VEEnc}(pk_{RA}, id_{sat}')$

    - The Joint computation of id_{sat} which is then known to both parties can be realized using standard techniques. Joint computation of the blinded id_{sat}' that is known to neither computing party, but is verifiably encrypted to other parties can be realized using standard homomorphic encryption techniques with the addition of blind anonymous verifiable encryption of the jointly computed value.

2. U computes the public portion of a keypair (pk_Θ, sk_Θ) which is generated s.t. the keypair depends on $(id_{U,RA}, t_\omega, id_{sat})$.

    - The creation of a keypair that depends on values $(id_{U,RA}, t_\omega, id_{sat})$ is possible by using those values as the identity in the generation of a committed blind anonymous IBE. Using this form of IBE pk_Θ can be released, and an encryption against this key can be verified, without revealing the identity or corresponding sk_Θ. Additionally, the later extraction of sk_Θ is a blind anonymous extraction that will not reveal these values.

3. U computes the time capsule as follows: $\Theta = (\mathit{VEEnc}(pk_\Theta, id_{U,IDP}), \mathit{VEEnc}(pk_{RA}, id_{sat}'))$ and sends Θ to SP

4. SP verifies in zero knowledge that Θ can be decrypted by sk_Θ and that RA can derive sk_Θ.

    - The verifiable encryption primitive of [**?**] permits the properties that we require of the time capsule Θ. In particular it is suitable for the encryption key pk_Θ such that the contents of the encryption and the decryptability under the secret values $(id_{U,RA}, t_\omega, id_{sat})$ is possible with further zero knowledge proofs.

5. RA verifies in zero knowledge that Θ contains the user's identity with respect to the system: $id_{U,IDP}$.

6. SP sends to U a token permitting access to the service.

Satisfaction of Revocation Condition:
$\mathit{Satisfy}(\mathsf{U}(\mathit{VEEnc}(pk_{SA}, id_{sat}')), \mathsf{SA}(sk_{SA}))$
At the end of this protocol U has proven to SA that the revocation condition has occurred with respect to the blinded encrypted form of the satisfaction token $\mathit{VEEnc}(pk_{SA}, id_{sat}')$. The user receives from RA a revocation token $tok(id_{sat}')$, and the now decrypted satisfaction token id_{sat}' which together prove that the contract with respect to the blinded satisfaction token has been fulfilled. This proof can be accomplished with standard anonymous credential techniques:

- The satisfaction token minted by SP with respect to id_{sat}' can be accomplished by encoding the satisfaction conditions (such as an amount of money to be paid, and the destination of that money) as attributes in an anonymous credential that has id_{sat}' encoded as a committed attribute.
- When SA decrypts its copy of id_{sat}' it can verify that the correct amount of money is being paid to the correct entity, and that the transaction is securely linked to the blinded satisfaction token.
- SA can then sign a statement that the transaction id_{sat}' has been fulfilled using any secure signature scheme.

Revocation: $\mathit{Revoke}(\mathsf{U}, \mathsf{BB})$ The user publishes to the bulletin board BB $tok(id_{sat}')$ and id_{sat}'.

- The user can revoke the ability of SP to open the time capsule by broadcasting proof of payment on BB. Since only id_{sat}' and a signature on id_{sat}' are broadcast, and since id_{sat}' cannot be linked with the user, this satisfies the privacy requirements of the system.

Check: $\mathit{Check}(\mathsf{RA}(id_{sat}'), \mathsf{BB}(id_{sat}'))$ The RA checks for revocation by searching the BB for id_{sat}', and verifying $tok(id_{sat}')$. If they are found and verified then the check succeeds, indicating that revocation has occurred and the contract is complete.

- During the *Check* operation id_{sat}' is revealed to RA by decryption under sk_{RA}. This suffices for RA to determine whether the contract has been satisfied, but RA cannot link the *Enroll* event for this transaction with this check, which provides the end to end unlinkability required for honest users.

Open: $\mathit{Open}(\mathsf{SP}(\Theta), \mathsf{RA}(sk_{RA}), \mathsf{TS}(T_i))$

1. SP sends $\mathit{VEEnc}(pk_{RA}, {id_{sat}}'))$ to RA and requests sk_{Θ}.
   - In the first step of opening, RA must perform an IBE key extraction of sk_{Θ} in order to retrieve ${id_{sat}}'$. Here we use the special anonymity properties of anonymous blind committed IBE to perform the key extraction without RA gaining knowledge of the user's true identity. Note that key extraction will only succeed if time symbol t_{ω} is now available from TS. This satisfied the temporal properties required by the system.
2. RA decrypts $\mathit{VEEnc}(pk_{RA}, {id_{sat}}')$ to retrieve ${id_{sat}}'$ (which is not linkable by RA with any other data).
3. RA performs *Check* $({id_{sat}}')$. If *Check* succeeds, RA detects revocation and halts without returning sk_{Θ}.
4. if $i <= \omega$, RA halts without returning sk_{Θ}.
5. RA generates and returns sk_{Θ}.
6. SP decrypts and reveals $id_{U,IDP}$
   - Only if the revocation is not detected by RA and key extraction is possible (it is after time ω) will sk_{Θ} be revealed to SP. If the key is revealed to SP then the capsule can be opened and the user's true identity revealed. This satisfies the privacy properties and the accountability properties of the system.

6. OTHER RELATED WORKS

Chaum pioneered privacy-preserving protocols that minimize the amount of personal data disclosed. His work put forth the principles of anonymous credentials [**?**, **?**, **?**], group signatures [**?**], and electronic cash [**?**]. Subsequently, a number of authors provided more efficient implementations of these primitives, e.g., group signatures [**?**, **?**, **?**], e-cash [**?**, **?**, **?**], anonymous credentials [**?**, **?**, **?**, **?**, **?**], traceable signatures [**?**], anonymous auctions [**?**], and electronic voting based on blind-signatures [**?**].

All these primitives have in common that some party issues a user some form of certificate that often contains information about the user encoded as attributes. Typically, these attributes are encoded as a discrete logarithm or, more generally, as an element (exponent) of a representation of a group element.

There are also some works [**?**, **?**, **?**, **?**, **?**, **?**] that these authors employ to prove AND, OR and NOT statement about attributes, e.g., "a user has attribute a OR b," basically by showing that some committed value equals a given value OR some other given value.

7. CONCLUSION

In this paper we presented a novel scheme to better fill the needs of service providers. We did this by introducing the concept of the time capsule: a cryptographic device that contains the user's real identity, and can only be opened if the user does not fulfill the terms of a predetermined service contract such as a contract to pay a bill on time.

By leveraging the recently developed anonymous blind committed identity based encryption scheme we are able to form an encryption key for the time capsule that depends on the user's pseudonym within the system, as well as conditions such as time and payment status. We have shown that due to the properties of this and other blinding mechanisms, the secret key for decrypting the time capsule can only be derived if the user misbehaves.

By contrast with existing anonymity revocation schemes, we do not require a fully trusted third party, such as a law enforcement agency, or judicial process. Due to the weaker trust requirements for the TTP it is easier for service providers to perform the anonymity revocation when a user misbehaves, relieving what may be a prohibitive expense for routine transactions.

While we have shown that a system with our desired properties can be constructed from known cryptographic primitives, it remains for future work to describe an exact implementation. Furthermore we believe that there may be more than one way to implement a system matching our requirements. Future work should investigate which implementations might offer the best efficiency, or the weakest assumptions.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nř 216483.

APPENDIX

A. CAMENISCH-LYSYANSKAYA

Let us recall Camenisch-Lysyanskaya signatures [**?**] (we present a slight and straightforward variant which allows messages to be negative integers as well). Let ℓ_m, ℓ_e, ℓ_n, ℓ_r and L be system parameters (ℓ_r is a security parameter, the meanings of the others will become apparent soon).

Key generation. On input ℓ_n, choose an ℓ_n-bit RSA modulus n such that $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$, where p, q, p', and q' are primes. Choose, uniformly at random, $R_0, \ldots, R_{L-1}, S, Z \in \mathrm{QR}_n$. Output the public key $(n, R_0, \ldots, R_{L-1}, S, Z)$ and the secret key p.

Message space is the set $\{(m_0, \ldots, m_{L-1}) : m_i \in \pm\{0,1\}^{\ell_m}\}$.

Signing algorithm. On input $m_0, \ldots, m_{L-1}$, choose a random prime number e of length $\ell_e > \ell_m + 2$, and a random number v of length $\ell_v = \ell_n + \ell_m + \ell_r$, where ℓ_r is a security parameter. Compute $A = (\frac{Z}{R_0^{m_0} \ldots R_{L-1}^{m_{L-1}} S^v})^{1/e} \bmod n$. The signature consists of (e, A, v).

Verification algorithm. To verify that the tuple (e, A, v) is a signature on message $(m_0, \ldots, m_{L-1})$, check that $Z \equiv A^e R_0^{m_0} \ldots R_{L-1}^{m_{L-1}} S^v \pmod{n}$, $m_i \in \pm\{0,1\}^{\ell_m}$, and $2^{\ell_e} > e > 2^{\ell_e - 1}$ holds.

THEOREM A.1. *[**?**] The signature scheme is secure against adaptive chosen message attacks [**?**] under the strong RSA assumption.*

Proving Knowledge of a Signature. Let us further recall how a prover can prove that she possesses a CL signature without revealing any other information about the signature.

Of course we want to use the protocols described in §4.3. Now, if A was a public value, we could do so by proving knowledge representation of Z w.r.t. $R_0, \ldots, R_{L-1}, S$, and A. Obviously making A public would destroy privacy as that would make all transaction linkable. Luckily, one can randomize A: Given a signature (A, e, v), the tuple $(A' := AS^{-r} \bmod n, e, v' := v + er)$ is also a valid signature as well. Now, provided that $A \in \langle S \rangle$ and that r is chosen uniformly at random from $\{0,1\}^{\ell_n+\ell_\varnothing}$, the value A' is distributed statistically close to uniform over $\mathbb{Z}_n^*$. Thus, the user could compute a fresh A' each time, reveal it and then run the protocols

$$\mathit{PK}\{(\varepsilon,\nu',\mu):\ Z \equiv \pm R_0^{\mu_0}\cdots R_{L-1}^{\mu_{L-1}} A'^{\varepsilon} S^{\nu'} \pmod n \wedge \\ \mu_i \in \pm\{0,1\}^{\ell_m} \ \wedge\ \varepsilon \in [2^{\ell_e-1}+1, 2^{\ell_e}-1]\}$$

Now, there is a technical consequence from this proof protocol regarding the the statements $\mu_i \in \pm\{0,1\}^{\ell_m} \quad \wedge \quad \varepsilon \in [2^{\ell_e-1} + 1, 2^{\ell_e}-1]$. While these can be implemented virtually for free, they requires that the actually secret lie in a smaller interval, i.e., the signer needs to choose e from $[2^{\ell_e-1} - 2^{\ell'_e} + 1, 2^{\ell_e-1} + 2^{\ell'_e} - 1]$ with $\ell'_e < \ell_e - \ell_\varnothing - \ell_\mathcal{H} - 3$, where $\ell_\varnothing$ and $\ell\mathcal{H}$ are security parameters (the first controlling statistical zero-knowledge and the second one being the size of the challenge message in the *PK* protocol). Similarly, we require $m_i \in \pm\{0,1\}^{\ell_m-\ell_\varnothing-\ell_\mathcal{H}-2}$ when input to the signature scheme (cf. [**?**]).