Controlling access to personal data through Accredited Symmetrically Private Information Retrieval

Mohamed Layouni

School of Computer Science, McGill University,
3480 University Street, Montreal, H3A 2A7, Quebec, Canada.

Abstract. With the digitization of society and the continuous migration of services to the electronic world, individuals have lost significant control over their data. In this paper, we consider the problem of protecting personal information according to privacy policies defined by the data subjects. More specifically, we propose a new primitive allowing a data subject to decide when, how, and by whom his data can be accessed, without the database manager learning anything about his identity, at the time the data is retrieved. The proposed solution, which we call *Accredited SPIR*, combines symmetrically private information retrieval and privacy-preserving digital credentials. We present three constructions based on the discrete logarithm and RSA problems. Despite the added privacy safeguards, the extra cost incurred by our constructions is negligeable compared to that of the underlying building blocks.

Keywords: Symmetrically private information retrieval, anonymous credentials, policy enforcement.

1 Introduction

In a transaction-based world, with continuously shrinking resources, access control has always been, and still continue to be a central issue. Oftentimes, to benefit from a service or a resource, a user is asked to show his identity, or prove possession of a set of qualifications and privileges. In many cases, this forces individuals into leaving identity trails behind them, which could be used for criminal activities such as unlawful monitoring and identity theft. The data collected from such interactions, although generally rich in personal information, is in most cases stored in databases lying outside the control of the data subject. Various techniques have been proposed in the past to strengthen users' privacy and help protect their personal information. Among these we note privacy preserving digital credentials [Cha85,Bra00,CL02], and symmetrically private information retrieval protocols [GIKM98,CMO00,KO97,AIR01,Lip05].

In a symmetrically private information retrieval (SPIR) system, there are generally two players: a Sender and a Receiver. The Sender has a database DB of records, and the Receiver submits a query Q to the Sender in order to retrieve a particular record. The main requirement in a SPIR system is privacy for both the Sender and the Receiver. That is, on the one hand the Sender should not learn any information about the index of the record the Receiver is interested in, and on the other hand, the Receiver should not learn any information about the database, beyond the content of the record defined in the query Q, and what is already publicly known. In particular, the Receiver should not be able to learn information about more than one record per query. For instance, the Receiver should not be able to learn, through one query, the value of any function on a set of more than one record. SPIR systems have many real-life applications; for instance, consider a scenario where the inventor of a new drug needs information on a number of chemical components that will constitute his final product. This information can be accessed *for a fee* at some central database. This database could be managed, however, by parties with possibly competitive interests, and the inventor fears that his intellectual property (IP) will be compromised. He would like, therefore, that his queries remain concealed from the database manager. The latter, on the other hand, wants to be paid for all information retrieved from his database. It is clear that the SPIR system described above, can be a solution to this set of conflicting requirements.

There are similar applications however, that are closely related to the IP example above, which cannot be solved by a SPIR primitive. Consider for example the following e-health scenario where three types of

participants are involved: (1) a patient, (2) a medical database containing the health records of patients, and (3) a doctor querying the medical database on patients' health records. The medical database and the doctor can be thought of as the Sender and Receiver, respectively, in a traditional SPIR setting. The requirements in the e-health application are as follows:

1. **Privacy for the Receiver**: The Receiver (doctor) wants to retrieve records from the medical database, without the Sender (DB) learning the index of those records, and thus the identity of his patient.
2. **Privacy for the Sender**: The Sender (DB) wants to be sure that, for each query, the Receiver (doctor) learns information only on one record (defined in the query) and nothing about the other records.
3. **Privacy for the data subject**: In order to comply with privacy legislation, the Sender wants to be sure that the Receiver has an valid reading authorization from the owner of the targeted record (i.e., the patient). We call the latter, *an Authorizer*. Notice that the Sender should not be able to learn the Authorizer's identity, otherwise the first requirement will be violated.

Another example where existing SPIR systems are insufficient, is that of credit history check-ups. Often when applying for a loan, or even a credit card, customers are asked by their would-be lenders for a permission to check their credit history. This credit history is generally stored in some large database managed by a government agency, or a private organization (possibly a competing lender), and is available for viewing for a fee. We call the manager of this database a Sender. We also denote the lender and customer, by Receiver and Authorizer, respectively. The requirements of the application can be stated as follows:

1. The Receiver wants to retrieve the credit history of the customer without disclosing the latter's identity to the Sender,
2. The Sender, who charges viewing fees per record retrieved, wants to be sure that the Receiver obtains information only on one record at a time, and nothing else about the other records.
3. In order to comply with privacy legislation, the Sender wants to be sure that the Receiver has obtained explicit viewing consent from the owner of the target record. This should be done without the Sender learning the identity of the target record's owner.

The two examples above show typical scenarios where plain SPIR primitives fall short of protecting the interests of the Sender, the Receiver, and the Authorizer at the same time. The solution we provide in this paper, addresses the interests of all three parties, and solves the problems described above. We call the presented solution: *Accredited SPIR*. In what follows, we sometimes refer to the latter set of requirements, namely privacy for the data-subject, the Sender, and the Receiver, as the *Accredited SPIR problem*.

Solution highlight. In the Accredited SPIR setting we have three players: a Sender, a Receiver, and an Authorizer. The Receiver submits a query Q to the Sender, who replies with a response R. The Receiver recovers the answer to his query from R. The main contribution of Accredited SPIR, is to assure the Sender, before processing the query Q, that the Receiver has obtained an explicit consent from the owner of the record defined in Q, *without* revealing the identity of this owner (i.e., the Authorizer).

The Accredited SPIR architecture we propose, combines three cryptographic primitives: privacy-preserving digital credentials, homomorphic encryption, and SPIR systems. Privacy-preserving digital credentials [Cha85,CP92,Bra94,Bra00,CL02,CL04] are cryptographic tokens issued by a certification authority CA to individuals. The CA encodes in each credential a set of attributes about the identity of its recipient. The latter is called a credential holder. A credential holder may later show his credential to a verifier in return for a service or a privilege (e.g., to receive medical treatment). Unlike traditional PKI certificates (e.g., X.509), privacy-preserving digital credentials allow their holders to selectively disclose information about their attributes [Bra00]. In particular, if a credential holder has a set of attributes $(x_1, \cdots, x_n)$, then he can prove any predicate $\mathcal{P}$ satisfied by those attributes, without the verifier learning any extra information beyond the status of $\mathcal{P}(x_1, \cdots, x_n)$. Furthermore, for specific constructions, different credential showings by the same prover are neither linkable to each other, nor to the credential issuing protocol instances that generated them (this remains true even if the credential issuer and verifier team up together.)

Assume the Authorizer has a CA-issued identity credential *Cred* containing a set of attributes (ID,Age,$\cdots$). The idea is to first make the Authorizer and Receiver jointly compute the query Q, and then have the Authorizer produce a signed proof of knowledge of the secret attributes embedded in *Cred*. Along with the latter, the Authorizer proves that the ID attribute embedded in *Cred* is the same as the one contained in the query Q. The Receiver then deposits the signed proof along with the query to the Sender. The Sender first checks the validity of the proof. If accepted it carries on with the SPIR protocol and processes the query, otherwise it rejects.

As mentioned earlier, the signed proof does not reveal any information about the credential holder, and yet guarantees that the content of the query is consistent with the secret identity attribute embedded in the credential. Furthermore, owing to the fact that it is hard for a polytime adversary to forge credentials, or to make proofs about credentials he does not own, the Sender can be sure that the Receiver has indeed obtained an explicit consent from the targeted record's owner.

This paper presents three constructions to solve the accredited SPIR problem. The first is based on a modified version of one of Brands DL-based credentials [Bra00, Section 4.5.2], the ElGamal cryptosystem, and a SPIR system proposed by Lipmaa in [Lip05]. The two additional constructions are variants of the first, and use an RSA-based version of Brands credentials [Bra00, Section 4.2.2], in combination with the ElGamal, and the Okamoto-Uchiyama [OU98] cryptosystems. In the following, we describe previous results and related work available in the literature.

2 Related work

Much research has gone into the problem of managing personal data in accordance with a user-defined privacy policy. In [GMM06], for instance, Golle *et al.* propose a mechanism by which data collectors can be caught and penalized if they violate an agreed-upon policy, and disclose sensitive data about a data-subject. The main idea there is that a data-collector would place a *bounty*, which it must forfeit if a privacy violation is uncovered. The bounty could be explicit in the form of a bond, or implicit in the form of penalties imposed if privacy is violated. This technique however is geared towards violation detection after the fact, and assumes the existence of active *bounty hunters* who seek to induce dishonest data collectors into committing unlawful disclosures.

Another related approach is that of policy-based encryption by Bagga *et al.* [BM05,BM06]. Policy-based encryption allows a user to encrypt a message with respect to a credential-based policy, formalized as a monotone boolean expression. The encryption is such that only a user having access to a qualified set of credentials, complying with the policy, is able to successfully decrypt the message. The context in [BM05,BM06], however, is different from the one in this paper, since the goal there is to allow the user to send a secret message to a designated set of players defined by a policy. In our context, the user's data is already stored in a database, and the goal is to allow user-authorized parties to retrieve the user's data, without the database manager learning which data has been retrieved or the identity of the data subject. It is also not clear how revocability can be implemented in the context of [BM05,BM06].

In [SWP00], Song *et al.* present a scheme for the problem of searching keywords on encrypted data. The setting there consists of a user, and a server storing encrypted data owned by the user. The server can process search queries on the user's stored ciphertext, only if given proper authorization from the user. The proposed scheme also supports hidden user queries, where the server conducts the search without learning anything about the content of the query. Although somewhat related to our context, it is not clear how the work in [SWP00] can be applied to the problem we describe in this paper, since delegating querying capabilities to a third party, may require the user to reveal his encryption key, and thus share all of his past and future secrets. Besides, it is not clear how the identity of the data-owner can be hidden from the server, or how to impose restrictions (e.g., time or usage restrictions) on the search capabilities delegated to a third party.

Finally, in [AIR01] Aiello *et al.* consider a scenario where a database contains a set of priced data items, and users privately retrieve data from it. The proposed protocol is called priced oblivious transfer, and allows a user U, who made an initial deposit, to buy different data items, without the database manager learning which items U is buying, subject to the condition that U's balance contains sufficient funds. We believe

the construction in [AIR01] is the first to consider imposing additional requirements on oblivious transfer protocols. While interesting in their own right, the added requirements do not address the identity of the data owners.

3 Summary of Contribution and paper organization

We propose a solution to the accredited SPIR problem, and three constructions to implement it. The solution we present allows a user to issue authorizations to a Receiver to privately retrieve his records, without the database manager learning anything about the retrieved data, or the data subject's identity. The authorizations contain computationally non-modifiable, unforgeable, user-defined policies and limitations, governing their use. The authorizations can also be anonymously revoked by their issuer if needed.

To the best of our knowledge, this work is the first to give a solution to the accredited SPIR problem, and to address the more general issue of enforcing user-defined privacy policies, by combining SPIR protocols and privacy-preserving digital credentials.

The remainder of this paper is organized as follows. In Section 4, we describe the main building blocks used in the first construction and throughout the paper. In Section 5, we present a DL-based accredited SPIR construction. In Sections 6 and 7, we discuss the security, privacy features, and performance of the first construction. In Section 8.2, we present a second construction based on a RSA version of Brands credentials. In Section 8.4, we give a third variant based on the Okamoto-Uchiyama cryptosystem. We conclude in Section 9.

4 Building Blocks for the DL-based construction

4.1 Brands-CP credentials

In [Bra00], Brands proposes various credential systems based on the hardness of the discrete logarithm problem in groups of prime order, and the RSA problem in groups of composite order (RSA groups). Brands has also proposed other variants of the above systems, based on DSA and the Chaum-Pedersen signatures [Bra00, Section 4.5.2]. For the purpose of our first construction, we will use the latter variant, and will refer to it as the Brands-CP system. The security of the Brands-CP system is based on the hardness of the discrete logarithm problem in groups of prime order. The Brands-CP system allows a certification authority CA to issue to a user U a set of credentials encoding attributes about U's identity. The credential itself consists of (1) a public key h embedding the user's attributes and (2) a special CA-supplied digital signature on it, denoted $\sigma_{\mathrm{CA}}(h)$. At the end of the issuing protocol, the credential that user U has obtained is perfectly hidden from the CA, and perfectly indistinguishable from any other credential the CA has previously issued. Later, user U can show his credential to individuals and organizations in return for a service. Showing a credential does not necessarily require the revealing of the attributes encoded in it. A credential holder can selectively and verifiably disclose any information he wishes about his attributes, which may include revealing the actual values of the attributes, or just proving a predicate about them. In [Bra00, Section 3.6], Brands shows how to prove a class of linear predicates about the attributes. At a later stage, and depending on the application, the verifying individual or organization may want to deposit the credential showing transcript to the certification authority. This deposit can be thought of as a cheque deposit in the context of e-banking or as a ballot submission in the context of e-voting. The deposited transcript is unlinkable to the instance of the issuing protocol that generated the credential. For the sake of completeness, we give in the following a brief description of the issuing, and showing protocols of the Brands-CP system, as well as an overview of the parameters and setting.

System setting. On input the security parameter κ, the CA chooses κ-sized primes p and q such that $2q|p-1$. Let G_q be the unique subgroup of $\mathbb{Z}_p^*$ of order q, and let g_0 be one of its generators. The security parameter κ is chosen large enough so that computing discrete logarithms in G_q is hard. The CA also chooses $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, a public collision-resistant hash function. In the setup phase, the certification authority

randomly chooses $y_1, y_2, \cdots, y_\ell$ and $x_0 \in_R \mathbb{Z}_q^*$, and computes $(g_1, g_2, \cdots, g_\ell, h_0) := (g_0^{y_1}, g_0^{y_2}, \cdots, g_0^{y_\ell}, g_0^{x_0})$ mod p. The parameters $(g_1, g_2, \cdots, g_\ell, h_0)$ are then made public along with g_0, q, and G_q.

The discrete representation problem : Let $(\varepsilon_1, \cdots, \varepsilon_\ell)$ be a set of elements from $\mathbb{Z}_q$, and let $\gamma := g_1^{\varepsilon_1} \cdots, g_\ell^{\varepsilon_\ell}$ mod p. We say that $(\varepsilon_1, \cdots, \varepsilon_\ell)$ is a discrete representation of γ with respect to the basis $(g_1, g_2, \cdots, g_\ell)$. In [Bra00, Section 2.3.2], Brands shows that given a random $\gamma \in G_q \setminus \{1\}$, and a basis $(g_1, g_2, \cdots, g_\ell)$, finding a discrete representation of γ with respect to $(g_1, g_2, \cdots, g_\ell)$ is at least as hard as breaking the discrete logarithm problem in G_q. Also, given γ and a representation $(\varepsilon_1, \cdots, \varepsilon_\ell)$ of γ with respect to basis $(g_1, g_2, \cdots, g_\ell)$, finding another representation $(\mu_1, \cdots, \mu_\ell)$ of γ with respect to the same basis is at least as hard as breaking the discrete log problem in G_q.

Credential issuing. To obtain a credential, a user first convinces the certification authority that he fulfills a set of application-specific requirements necessary to receive that credential. The certification authority then encodes a set of ℓ of user attributes in the credential. It is also possible for the user to encode a subset of the attributes which will remain hidden from the certification authority. Let $x_1, \cdots, x_\ell$ denote the attributes to be encoded. The credential's public key is then computed as $h := (g_1^{x_1} \cdots g_\ell^{x_\ell} h_0)^\alpha$, where α is a secret blinding factor randomly chosen in $\mathbb{Z}_q^*$ by the user.

The certification authority's signature on the credential is a triplet $(c_0', r_0', z') \in \mathbb{Z}_q^2 \times G_q$, satisfying the relation $c_0' = H(h, z', g_0^{r_0'} h_0^{-c_0'}, h^{r_0'} z'^{-c_0'})$. At the end of the issuing protocol, the certification authority knows neither h nor the signature (c_0', r_0', z'). Figure 1 summarizes the issuing protocol.

User U	Public Info	CA
	$(g_i)_{0\le i\le \ell}, h_0, (g_i^{x_0})_{0\le i\le \ell}, h_0^{x_0}, q, G_q, H$	
	$\xleftrightarrow{\text{agree on attributes } (x_1,\cdots,x_\ell)}$	$w_0 \in_R \mathbb{Z}_q$
		$a_0 := g_0^{w_0}$
	$\xleftarrow{a_0, b_0}$	$b_0 := (g_1^{x_1} \cdots g_\ell^{x_\ell} h_0)^{w_0}$
$\alpha_1 \in_R \mathbb{Z}_q^*$,		
$\alpha_2, \alpha_3 \in_R \mathbb{Z}_q$,		
$h := (g_1^{x_1} \cdots g_\ell^{x_\ell} h_0)^{\alpha_1}$		
$z := (g_1^{x_1} \cdots g_\ell^{x_\ell} h_0)^{x_0}$		
$z' := z^{\alpha_1}$		
$a_0' := h_0^{\alpha_2} g_0^{\alpha_3} a_0$		
$b_0' := (z')^{\alpha_2} h^{\alpha_3} b_0^{\alpha_1}$		
$c_0' := H(h, z', a_0', b_0')$		
$c_0 := c_0' + \alpha_2 \mod q$	$\xrightarrow{c_0}$	
	$\xleftarrow{r_0}$	$r_0 := c_0 x_0 + w_0 \mod q$
$r_0' := r_0 + \alpha_3 \mod q$		
accept iff $a_0' b_0' = (g_0 h)^{r_0'} (h_0 z')^{-c_0'}$		
store $h, \sigma_{\mathrm{CA}}(h) := (z', r_0', c_0')$		

Fig. 1. Brands-CP Credential Issuing protocol

Credential showing. In order to have access to a service, user U can show his credential without the verifying party being able (1) to learn information about the encoded attributes beyond what U willingly discloses, or (2) to link the credential to the user's identity even if it colludes with the certification authority. If desired, the setting can be made such that the user's attributes can be uncovered if the credential is shown more than a pre-defined threshold number of times [Bra00, Section 5.4]. In practice, to show his credential to a verifying party, user U reveals (1) the credential's public key h along with a signature $\sigma_{\mathrm{CA}(h)} := (z', c_0', r_0')$,

and (2) a signed proof of knowledge of a representation of h, with respect to basis $(g_1, g_2, \cdots, g_\ell, h_0)$. This signature is performed on a verifier-chosen challenge m. The verifier checks the validity of the credential by verifying if the relation $c_0' \stackrel{?}{=} H(h, z', g_0^{r_0'} h_0^{-c_0'}, h^{r_0'} z'^{-c_0'})$ holds. If the credential is valid, the verifier moves on to check the validity of the signed proof of knowledge. Figure 2 sketches the basic showing protocol of the Brands-CP system.

User U	Public Info $(g_i)_{0\le i\le \ell}, h_0, (g_i^{x_0})_{0\le i\le \ell}, h_0^{x_0}, q, G_q, H$	Verifier
$w_1, \cdots, w_{\ell+1} \in_R \mathbb{Z}_q$ $a := (g_1)^{w_1} \cdots g_\ell^{w_\ell} h_0^{w_{\ell+1}}$ $c := H(h, a, m)$ $r_1 := \alpha c x_1 + w_1$ $\vdots$ $r_\ell := \alpha c x_\ell + w_\ell$	$\xleftarrow{m}$	$m := nonce\|\|..$
$r_{\ell+1} := \alpha c + w_{\ell+1}$	$\xrightarrow{h, \sigma_{\mathrm{CA}}(h):=(z', c_0', r_0'), (a, r_1, \cdots, r_{\ell+1})}$	accept iff $c_0' = H(h, z', g_0^{r_0'} h_0^{-c_0'}, h^{r_0'} z'^{-c_0'})$, and for $c := H(h, a, m)$, $h^c a = g_1^{r_1} \cdots g_\ell^{r_\ell} h_0^{r_{\ell+1}}$ holds.

Fig. 2. Brands-CP basic showing protocol – signed proof of knowledge

The signed proof of Figure 2 can be computed, for example, with respect to a predicate $\mathcal{P}$ of the form "$x_i = x_j$" for some $(i, j) \in [1, \ell]$. In this case, we have $h = (g_1^{x_1} \cdots (g_i g_j)^{x_i} \cdots g_\ell^{x_\ell} h_0)^{\alpha_1}$, and the proof of knowledge is carried out as follows. First a description of the predicate $\mathcal{P}$ is hashed along in the challenge $c := H(h, a, \mathcal{P}, m)$, and the user proves knowledge of a representation of h with respect to basis $(g_1, \cdots, g_{i-1}, g_i g_j, g_{i+1}, \cdots, g_{j-1}, g_{j+1}, \cdots, g_\ell, h_0)$. User U can also prove a much wider class of predicates in a similar fashion [Bra00, Section 3.6].

Main Security and Privacy properties of the Brands-CP credentials.

- Unforgeability of credentials: assuming the discrete logarithm problem is hard, forging credentials in probabilistic polynomial time with a non-negligeable probability is infeasible.
- Unconditional privacy with respect to Issuer: At the end of an issuing protocol, the credential obtained by the user is perfectly indistinguishable to the Issuer, regardless of its power.
- Unconditional privacy with respect to Verifier: At the end of a showing protocol, the Verifier does not learn any information about the attributes embedded in the credential, beyond what the user willfully discloses.
- One-time Unlinkability: the showings, by the same user, of any two credentials that have not been previously shown, are perfectly unlinkable. Furthermore, the showings of a credential are perfectly unlinkable to the issuing protocol instance that generated it.
- Consistency of proofs: Assuming the discrete logarithm problem is hard, it is infeasible for a user to prove, with a non-negligeable probability, a predicate that is not satisfied by the attributes initially embedded in his credential.
- Unforgeability and Non-modifiability of showing transcripts: Assuming the discrete logarithm problem is hard, it is infeasible for an adversary to:
  1. produce, with a non-negligeable probability, a showing transcript of a credential he does not own,
  2. alter a existing transcript, for example by modifying the predicate of the original proof.

4.2 ElGamal homomorphic encryption

Our DL-based accredited SPIR construction relies on the ElGamal encryption scheme because of its homomorphic properties and because it fits well the setting of the Brands-CP credentials. In the following we recall the settings of the ElGamal cryptosystem.

Settings. Let p, q, and G_q, be the public parameters chosen by the CA in the setup of the Brands-CP credential system. User U randomly chooses g_{EIG}, a generator of G_q, and $x_u \in_R \mathbb{Z}_q^*$, and computes $y_{\mathsf{EIG}} := g^{x_u} \bmod p$. User U then publishes his ElGamal public key $(G_q, g_{\mathsf{EIG}}, y_{\mathsf{EIG}})$, and keeps his private key x_u secret. A message $m \in G_q$, can be encrypted by choosing a random $r \in_R \mathbb{Z}_q^*$, and computing $c = (g_{\mathsf{EIG}}^r, y_{\mathsf{EIG}}^r m) = (c_1, c_2)$. Using U's private key, the plaintext can be recovered as $m = c_2/c_1^{x_u}$. Given a constant α, and encryptions of m and m', it is easy to compute randomized encryptions of $m \times m'$ and m^α. An added feature of the ElGamal scheme, is that the validity of the public key and the ciphertext is easily verifiable. For the latter, it suffices to check that (c_1, c_2) are both elements of G_q, while for the former, one needs to check that p and q are both primes, that $q|p-1$, and that $(g_{\mathsf{EIG}}, y_{\mathsf{EIG}})$ are both generators of G_q.

4.3 AIR-based Lipmaa $\binom{1}{n}$-OT

In [Lip05], Lipmaa proposes a SPIR scheme based on ideas from a construction by Aiello et al. [AIR01]. Lipmaa's SPIR scheme is computationally private for the Receiver and perfectly private for the Sender. Its security relies on the hardness of the *decisional Diffie-Hellman* and the *decisional composite residuosity* problems [Lip05]. The SPIR scheme in [Lip05] has a log-squared communication complexity (in the size of the database).

Main idea. Let DB denote the Sender's private database, and let s be the index of the record the Receiver is interested in. The receiver computes $\mathsf{c} := \mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(s)$, a homomorphic encryption of s, and sends it to the Sender. Using the homomorphic properties of the encryption, the Sender computes for each record $\mathrm{DB}[j]$ in the database, $\mathrm{DB}'[j] := \mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(\delta_j(s-j) + \mathrm{DB}[j])$, where δ_j is a random blinding factor chosen by the Sender. The encrypted records $\mathrm{DB}'[j]$ are then sent to the Receiver, who will be able to retrieve something meaningful only from $\mathrm{DB}'[s] := \mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(\mathrm{DB}[s])$; everything else will decrypt to randomness. The construction in [Lip05] follows a similar methodology to the above, except that the Sender uses an extra loop of superposed encryptions that leads to a randomized ciphertext of $\mathrm{DB}'[s]$. Only the latter is sent back to the Receiver. This is done as follows. The database $(\mathrm{DB}[1], \cdots, \mathrm{DB}[n])$ is arranged in an α-dimensional $\lambda_1 \times \cdots \times \lambda_\alpha$ hyper-rectangle for some pre-defined positive integers λ_j, such that $n = \prod_{j=1}^{\alpha} \lambda_j$. Each record $\mathrm{DB}[i]$ is indexed by a tuple $(i_1, \cdots, i_\alpha)$ on this hyper-rectangle, where $i_j \in \mathbb{Z}_{\lambda_j}$. To retrieve a particular record $(s_1, \cdots, s_\alpha)$, the Receiver submits to the Sender a homomorphic encryption $\beta_{jt} := HomEnc_{pk}(b_{jt})$, for $1 \le j \le \alpha, 0 \le t < \lambda_j$, where $b_{jt} = 1$ if $t = s_j$, and $b_{jt} = 0$ otherwise. The Sender exploits the homomorphic properties of the encryption scheme $HomEnc_{pk}(.)$ to create a new $(\alpha-1)$-dimensional database DB_1, such that $\forall (i_2, \cdots, i_\alpha) \in \mathbb{Z}_{\lambda_2} \times \cdots \times \mathbb{Z}_{\lambda_\alpha}$, $\mathrm{DB}_1(i_2, \cdots, i_\alpha)$ is equal to an encryption of $\mathrm{DB}_0(s_1, i_2, \cdots, i_\alpha)$, where DB_0 is the the Sender's original database DB. The same procedure is repeated, and at the j^{th} iteration, an $(\alpha-j)$-dimensional database DB_j is obtained by the Sender, such that $\mathrm{DB}_j(i_{j+1}, \cdots, i_\alpha)$ is equal to a j-times encryption of $\mathrm{DB}_0(s_1, \cdots, s_{j-1}, i_j, \cdots, i_\alpha)$. After α iterations, the Sender obtains DB_α, an α-times encryption of $\mathrm{DB}_0(s_1, \cdots, s_\alpha)$. The Sender returns DB_α to the Receiver, who needs to decrypt it α times to recover $\mathrm{DB}(s)$. In [Lip05], Lipmaa uses a special *Length Flexible Additively Homomorphic* encryption scheme to implement $HomEnc_{pk}(.)$. Notice that in the hyper-rectangle construction above, the Receiver can cheat by maliciously sending $\beta_{jt} := HomEnc_{pk}(1)$, for $t \neq s_j$. To stop such attacks, the Sender performs the repeated encryptions above, on $\mathrm{DB}_0 = \mathrm{DB}'$ rather than on the Sender's original DB. Let s' be the index corresponding to the β_{jt}'s. At the end of the protocol, the Receiver obtains DB_α, which he decrypts α times to recover $\mathrm{DB}'[s'] = \mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(\delta_{s'}(s-s') + \mathrm{DB}[s'])$. Next, the Receiver decrypts $\mathrm{DB}'[s']$ once again, and recovers something meaningful ($\mathrm{DB}[s]$) only if $s' = s$. A summary of the the whole protocol is given in Figure 3.

Remark. Both [Lip05] and [AIR01] propose the ElGamal cryptosystem to implement $\mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}$. It is worth noting however, that using plain ElGamal, it is not possible to compute $\mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(\delta_j(s-j)+\mathrm{DB}[j])$ given $\mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(s)$, since ElGamal is *only multiplicatively* homomorphic. We fix this problem in the next section.

Receiver (index s)	Public Info	Sender (Database DB)
	$n=\lvert\mathrm{DB}\rvert, k, \mathcal{R}, \mathsf{pk}_{\mathsf{hom}}, pk$	
$(\mathsf{sk}_{\mathsf{hom}}, \mathsf{pk}_{\mathsf{hom}}) \leftarrow \mathsf{Gen}_{\mathsf{hom}}(1^k)$ $\mathsf{c} := \mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(s)$		
$(sk, pk) \leftarrow Gen_{Hom}(1^k)$ For $j := 1$ to α do : For $t := 0$ to $\lambda_j - 1$ do : $r_{jt} \in_R \mathcal{R}$ $\beta_{jt} := HomEnc_{pk}(b_{jt}, r_{jt})$, where $b_{jt} := 1$ if $t = s_j$, and $b_{jt} := 0$ otherwise.	$\xrightarrow{\mathsf{c},\ \{\beta_{jt}\}_{1\le j\le\alpha, 0\le t<\lambda_j}}$	Check validity of $\mathsf{pk}_{\mathsf{hom}}$ and c. For $j := 1$ to n do : $\delta_j \in_R [1, n]$ $\mathrm{DB}_0[j] := \mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}(\delta_j(s-j)+\mathrm{DB}[j])$ For $j := 1$ to $\alpha - 1$ do :, For $i_{j+1} := 0$ to $\lambda_{j+1} - 1, \cdots,$ $i_\alpha := 0$ to $\lambda_\alpha - 1$ do : $\mathrm{DB}_j(i_{j+1}, \cdots, i_\alpha) :=$ $\prod_{t\in\mathbb{Z}_{\lambda_j}} (\beta_{jt})^{\mathrm{DB}_{j-1}(t, i_{j+1}, \cdots, i_\alpha)}$
$\mathrm{DB}'_\alpha := \mathrm{DB}_\alpha$ For $j := \alpha$ downto 1 do :, $\mathrm{DB}'_{j-1} := HomDec_{sk}(\mathrm{DB}'_j)$ Output $\mathrm{DB}[s] := \mathsf{D}_{\mathsf{sk}_{\mathsf{hom}}}(\mathrm{DB}'_0)$	$\xleftarrow{\mathrm{DB}_\alpha}$	$\mathrm{DB}_\alpha := \prod_{t\in\mathbb{Z}_{\lambda_\alpha}} (\beta_{\alpha t})^{\mathrm{DB}_{(\alpha-1)}(t)}$

Fig. 3. Lipmaa $\binom{1}{n}$-OT (also based on [AIR01])

5 Accredited SPIR based on the DL problem

The DL-based accredited SPIR scheme we propose, is achieved by combining the three building blocks above, modulo few adaptations. We first give a high-level overview of the construction, before getting into the details. We assume the public parameters of the three building blocks are already known to all parties. Let $\mathrm{ID}_{\mathcal{A}}$ be an attribute, that uniquely identifies the Authorizer (e.g., an SSN). This $\mathrm{ID}_{\mathcal{A}}$ will determine the index by which the Receiver will query the Sender's database. Let us first assume that the Authorizer possesses a Brands-CP credential of the form $(h, \sigma_{\mathrm{CA}}(h))$, where $h = (g_1^{\mathrm{ID}_{\mathcal{A}}} g_2^{x_2} \cdots g_\ell^{x_\ell} h_0)^\alpha$. The Authorizer computes $\mathsf{c} := \mathsf{E}_{\mathsf{pk}_{\mathsf{hom}}}((g_{db})^{\mathrm{ID}_{\mathcal{A}}}) := \mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}((g_{db})^{\mathrm{ID}_{\mathcal{A}}}) := (\mathsf{c}_1, \mathsf{c}_2)$, where $\mathsf{pk}_{\mathsf{ElG}}$ is the Receiver's ElGamal public key, and g_{db} is a public generator of G_q chosen by the Sender. Next, the Authorizer produces a signed proof of knowledge asserting that the logarithm, to the base g_{db}, of the plaintext encoded in c, is the same as the first attribute embedded in credential h. We call this last assertion an ID-consistency proof.

Notice that this latter proof cannot be done in a straightforward way using the original Brands-CP credentials, because h has the form $h := g_1^{\beta_1} \cdots g_\ell^{\beta_\ell} h_0^\alpha$, where $\beta_i = \alpha x_i$ for some random blinding fac-

tor α. Establishing ID-consistency in this case requires proving a non-linear predicate on secret exponents $(\alpha, \beta_1, \mathrm{ID}_{\mathcal{A}})$, defined by $\mathcal{P} \equiv$ "$\beta_1 = \alpha \times \mathrm{ID}_{\mathcal{A}}$", which cannot be done efficiently. To fix this problem we propose a modified version of the Brands-CP credentials, with exactly the same security and privacy properties. In the modified version, the credential's public key h is computed as $h := (g_1^{x_1} \cdots g_{\ell-1}^{x_{\ell-1}} g_\ell^{\alpha} h_0)$, where $x_1, \cdots, x_{\ell-1}$ are identity attributes, and α is a secret random blinding factor chosen by the credential recipient. This modification is of general interest, and can be used in other contexts as well. A summary of the modified credential system is outlined in Figure 4.

In what follows, we assume the Authorizer possesses a credential of the new type. Let us denote the public key of the Authorizer's new credential by $h := (g_1^{\mathrm{ID}_{\mathcal{A}}} \cdots g_{\ell-1}^{x_{\ell-1}} g_\ell^{\alpha_1} h_0)$. To prove ID-consistency between h and the SPIR query $(\mathsf{c_1}, \mathsf{c_2}) := \mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}((g_{db})^{\mathrm{ID}_{\mathcal{A}}})$, it suffices for the Authorizer to produce a signed proof of knowledge of a DL-representation of $(h/\mathsf{c_2} \mod p)$ with respect to basis $((g_1 g_{db}^{-1}), g_2, \cdots, g_\ell, h_0, g_{\mathsf{ElG}}, y_{\mathsf{ElG}})$. This is done using the same method of Figure 2. We denote the latter signed proof by $\mathrm{SPK}\{(\varepsilon_1, \cdots, \varepsilon_\ell, \mu, \nu) : h = g_1^{\varepsilon_1} \cdots g_\ell^{\varepsilon_\ell} h_0 \wedge \mathsf{c_2} = y_{\mathsf{ElG}}^{\mu} g_{db}^{\nu} \wedge \varepsilon_1 = \nu\}(m)$. As shown in Figure 2, the message m to be signed, can be a concatenation of several fields, including a fresh nonce. In addition, it may contain the identity of the Receiver, which will allow the Authorizer to exclusively tie the authorization to the Receiver, and discourage him from sharing it with a third party. The Authorizer may also include an expiry date in m to make sure his authorization remains valid only for the appropriate amount of time. More generally, the Authorizer may encode in m any application-specific policy he wants the Receiver to follow.

User U	Public Info	CA
	$(g_i)_{0\le i\le \ell}, h_0, (g_i^{x_0})_{0\le i\le \ell}, h_0^{x_0}, q, G_q, H$	
	$\xleftarrow{\text{agree on attributes } (x_1,\cdots,x_{\ell-1})}$	
		$w_0 \in_R \mathbb{Z}_q$
		$a_0 := g_0^{w_0}, a_\ell := g_\ell^{w_0}$
	$\xleftarrow{a_0, b_0, a_\ell}$	$b_0 := (g_1^{x_1} \cdots g_\ell^{x_\ell} h_0)^{w_0}$
$\alpha_1 \in_R \mathbb{Z}_q^*$,		
$\alpha_2, \alpha_3 \in_R \mathbb{Z}_q$,		
$h := g_1^{x_1} \cdots g_{\ell-1}^{x_{\ell-1}} h_0 g_\ell^{\alpha_1}$		
$z' := (g_1^{x_1} \cdots g_{\ell-1}^{x_{\ell-1}} h_0 g_\ell^{\alpha_1})^{x_0}$		
$a_0' := h_0^{\alpha_2} g_0^{\alpha_3} a_0$		
$b_0' := (z')^{\alpha_2} h^{\alpha_3} (b_0 a_\ell^{\alpha_1})$		
$c_0' := H(h, z', a_0', b_0')$		
$c_0 := c_0' + \alpha_2 \mod q$	$\xrightarrow{c_0}$	
	$\xleftarrow{r_0}$	$r_0 := c_0 x_0 + w_0 \mod q$
$r_0' := r_0 + \alpha_3 \mod q$		
accept iff $a_0' b_0' = (g_0 h)^{r_0'} (h_0 z')^{-c_0'}$		
store $h, \sigma_{\mathrm{CA}}(h) := (z', r_0', c_0')$		

Fig. 4. Modified version of the Brands-CP Credential Issuing protocol

Running the SPIR. Now let us assume the signed proof above was accepted. The next step, would be for the Receiver to compute the query messages as shown in the OT scheme of Figure 3. First let $(\mathrm{ID}_{\mathcal{A}}{}^{(1)}, \cdots, \mathrm{ID}_{\mathcal{A}}{}^{(\alpha)})$ be the representation of the Authorizer's $\mathrm{ID}_{\mathcal{A}}$ in the α-dimensional hyper-rectangle $\lambda_1 \times \cdots \times \lambda_\alpha$ used by the Sender's database. The Receiver then computes, for $1 \le j \le \alpha, 0 \le t < \lambda_j$, the homomorphic encryptions $\beta_{jt} := HomEnc_{pk}(b_{jt})$, where $b_{jt} = 1$ if $t = \mathrm{ID}_{\mathcal{A}}{}^{(j)}$, and $b_{jt} = 0$ otherwise. Next, the Receiver submits to Sender:

- the credential $(h, \sigma_{\mathrm{CA}}(h))$,
- the first part of the query $(\mathsf{c_1}, \mathsf{c_2}) := \mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}((g_{db})^{\mathrm{ID}_{\mathcal{A}}})$,

– an ID-consistency proof $\mathrm{SPK}\{(\varepsilon_1, \cdots, \varepsilon_\ell, \mu, \nu) : h = g_1^{\varepsilon_1} \cdots g_\ell^{\varepsilon_\ell} h_0 \wedge \mathsf{c}_2 = y_{\mathsf{ElG}}^{\mu} g_{db}^{\nu} \wedge \varepsilon_1 = \nu\}(m)$,
– the second part of the query consisting of the β_{jt}'s for $1 \leq j \leq \alpha, 0 \leq t < \lambda_j$.

Note that there is no need for the Receiver to prove consistency between the β_{jt}'s and $(\mathsf{c}_1, \mathsf{c}_2)$. As we will show later, any attempt by the Receiver to incorrectly compute the β_{jt}'s, will prevent him from learning anything meaningful at the end of the SPIR protocol.

Once the ID-consistency check succeeds, the Sender starts processing the Receiver's query as explained in the following. But first, we make few practical assumptions. We assume that n, the size of the Sender's database, is bounded above by q, the order of G_q. In practice, q is chosen to be at least 160-bit long, which means the Sender's database could have up to 2^{160} different records. Although we think this should be sufficient in practice, the size of q can always be increased if needed. Moreover, we assume that each record $\mathrm{DB}[i]$ contains a field for storing $(g_{db}^{-i} \mod p)$, in addition to a large field containing application-specific data (e.g., health, financial data).

Now the query is processed as follows. Using the first part of the query, and the *multiplicatively* homomorphic properties of ElGamal, the Sender computes for $j \in [1, n]$, $\mathrm{DB}_0[j] = \mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}((g_{db})^{\delta_j(\mathrm{ID}_{\mathcal{A}} - j)} \times \mathrm{DB}[j]) = ((\mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}(g_{db}^{\mathrm{ID}_{\mathcal{A}}}) \times g_{db}^{-j})^{\delta_j} \times \mathrm{DB}[j])$, where δ_j is a random blinding factor chosen by the Sender. More precisely, the sender computes $\mathrm{DB}_0[j] := \left({\mathsf{c}_1}^{\delta_j}, (\mathsf{c}_2 \times g_{db}^{-j})^{\delta_j} \times \mathrm{DB}[j]\right)$, where $(\mathsf{c}_1, \mathsf{c}_2) := \mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}(g_{db}^{ID_{\mathcal{A}}})$. Note that g_{db}^{-j} has already been precomputed and stored in with DB's j^{th} record. The Sender then proceeds with computing DB_α by repeated encryptions of the records of DB_0 as indicated in Figure 3. Upon receiving DB_α, the Receiver recovers $\mathrm{DB}_0[\mathrm{ID}_{\mathcal{A}}]$ by repeated decryption using his secret homomorphic key sk. Next the sender obtains the desired record $\mathrm{DB}[\mathrm{ID}_{\mathcal{A}}]$ by decrypting $\mathrm{DB}_0[\mathrm{ID}_{\mathcal{A}}]$ using his ElGamal private key. A summary of the protocol is given in Figure 5.

6 Security and privacy properties

Theorem 1 *Assuming the DL problem is hard, the ElGamal and $HomEnc(.)$ cryptosystems IND-CPA secure, and that the none of the three parties colludes with the other, the protocol of Figure 5 solves the Accredited SPIR problem in the random oracle model, and provides computational privacy for both the Authorizer and Receiver, and perfect privacy for the Sender.*

Proof Sketch.[1] *Correctness.* easy to check and relies on the homomorphic properties of the ElGamal and the $HomEnc(.)$ cryptosystems.

Soundness. Assume the Receiver does not follow the protocol, and maliciously uses an index $s \neq \mathrm{ID}_{\mathcal{A}}$ in the second part of the query. This will lead DB_α computed by the Sender to be an α-time encryption under $HomEnc_{pk}(.)$ of $DB_0[s] = \mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}((g_{db})^{\delta_s(\mathrm{ID}_{\mathcal{A}} - s)} \times \mathrm{DB}[s])$, where δ_s is a secret blinding factor. $DB_0[s]$ will clearly decrypt to random, which results in perfect privacy for the Sender.

Privacy for the Receiver. The Sender's view consists of a credential, an ElGamal encryption, a signed proof of knowledge, and a $HomEnc(.)$ encryption of an α-dimensional coordinate. Because the proof of knowledge is honest-verifier zero-knowledge for any distribution of the attributes (cf. [Bra00, Prop. 3.3.4]), seeing the signed proof of knowledge, together with the credential, does not leak any information to the Sender about the content of the Receiver's query. The Sender could only hope to extract information from the ElGamal ciphertext c and the homomorphic encryptions β_{jt}'s. But this should not be possible, because it implies breaking the security of ElGamal, or the $HomEnc(.)$ cryptosystems, which contradicts our assumption.

Privacy for the Authorizer. Assuming the Receiver and the Sender do not collude, there are two ways to violate the Authorizer's privacy. Either by learning information from the Receiver's query, or by forging a

[1] Complete details will be given in the full version of the paper.

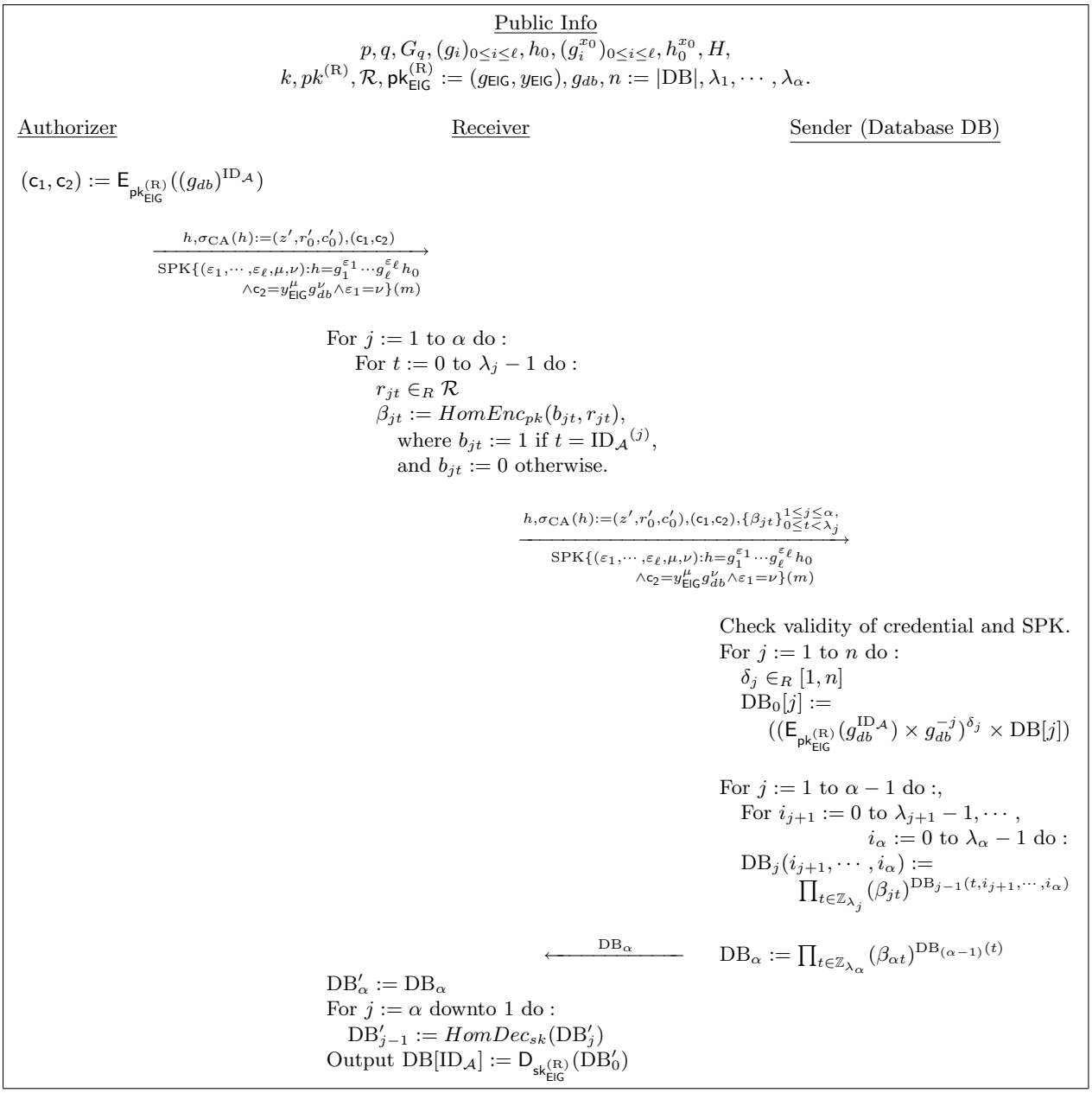

Fig. 5. Accredited SPIR architecture (DL-based construction)

signed proof of knowledge on behalf of the Authorizer. The former attack is computationally impossible and follows from the Receiver's privacy. To achieve the second however, one could either (1) forge a new credential from scratch with the Authorizer's identity embedded in it, (2) forge a signed proof on a legitimately-obtained credential that was not issued to the Authorizer, and thus does not initially contain his identity, or (3) forge a signed proof on behalf of the Authorizer on a credential owned by the Authorizer. Attack (1) can be ruled out based on the computational unforgeability of the Brands-CP credentials in the random oracle model (cf. [Bra00, Prop. 4.3.7]). Similarly, attack (2) is impossible because of the *non-modifiability* property of Brands-CP credentials [Bra00, Prop. 3.3.8]. The non-modifiability feature states that, assuming the DL problem

is hard, it is infeasible, in the random oracle model, to construct a signed proof for a formula that does not in fact apply to the prover's representation. Finally, attack (3) can be ruled out as well, owing to the unforgeability of signed proofs on Brands-CP credentials in the random oracle model [Bra00, Prop. 3.3.6].

6.1 Additional privacy for the Authorizer

User-centricity and policy enforcement. When issuing an authorization, the credential holder could specify in the message of the signed proof of knowledge, a set of rules that he wishes the authorization recipient to comply with. For instance, he could specify an expiry date, an upper bound on the number of times the authorization is used, or any other usage policy. The Sender is supposed to refuse processing queries from a Receiver who does not satisfy the usage policy specified in the signature.

Revocability. The Authorizer may decide to revoke a previously issued authorization. This can be done anonymously as follows. The Authorizer first needs to prove knowledge, over an physically anonymized channel (e.g., a MixNet[Cha81]), of a representation of the credential used in the authorization to be revoked. Once the proof of knowledge is accepted, the Authorizer requires the Sender (DB manager) to add the credential in question to a black list of revoked authorizations. Later, it is easy for the Sender to check whether the credential contained in a submitted query is on the black list or not. This can be done efficiently using hash tables for instance. Note that an authorization can be revoked only by its issuer, since an polytime adversary cannot find a discrete-log representation with non-negligeable probability.

Authenticated personal information retrieval. In the special case, where the Authorizer and Receiver are the same entity, the construction we propose provides the data-subject with a mechanism to retrieve *his own* personal data anonymously. Our construction also ensures that the stored data can be retrieved only by its owner. The channel between the Receiver and Sender in this case has to be physically anonymized.

7 Performance analysis

The accredited SPIR construction of Figure 5 does not lead to a significant increase in computation and communication complexity, compared to the underlying SPIR scheme[Lip05]. If we assume the Authorizer has a credential with $(\ell-1)$ attributes, then the added computation complexity is as follows. The Authorizer needs to make $(\ell+6)$ offline exponentiations (all precomputable), while the Receiver and Sender, both need to make $(\ell+8)$ online exponentiations. This is negligeable compared to the complexity of the underlying SPIR scheme which is linear in n, the size of the database. In practice, ℓ is in the range of 20, while $n \approx 2^{160}$. In terms of communication complexity, both the Authorizer and Receiver need to send $(\ell+8)\log(n)+5$ extra bits to the Receiver and Sender respectively. Again this does not change the overall $\mathcal{O}\big(\log^2(n)\big)$ asymptotic communication complexity of the underlying SPIR scheme [Lip05].

8 Accredited SPIR based on RSA

The constructions we present in this section are based on a RSA-version of Brands' credentials [Bra00, Section 4.2.2]. For the sake of completeness, we briefly introduce them in the following.

8.1 Brands-RSA credentials

Settings. On input the security parameter κ, the credential issuer chooses:

- κ-sized primes P and Q, and computes $N := PQ$.
- a prime v smaller than N, and co-prime to $\phi(N)$.
- random elements $(g_1, \cdots, g_\ell) \in_R (\mathbb{Z}_N^*)^\ell$
- a one-way hash function $\mathcal{H}(.) = \mathcal{H}_{N,v}(.) : \{0,1\}^* \to \mathbb{Z}_s$, for some s superpolynomial in κ.

The credential issuer makes the parameters $N, v, (g_1, \cdots, g_\ell), \mathcal{H}$ public, and keeps P and Q secret. In addition, the issuer chooses $x_0 \in_R \mathbb{Z}_v^*$, such that given $h_0 := x_0^v \mod N$, computing the v^{th} root of h_0 is hard. The issuer then publishes h_0 and keeps x_0 secret.

The RSA representation problem : Let $(\varepsilon_1, \cdots, \varepsilon_\ell)$ be a set of elements in $\mathbb{Z}_v$, and $\varepsilon_{\ell+1}$ be an element in $\mathbb{Z}_N^*$. Let $\gamma := g_1^{\varepsilon_1} \cdots, g_\ell^{\varepsilon_\ell} \varepsilon_{\ell+1}^v \mod N$. We say that $(\varepsilon_1, \cdots, \varepsilon_\ell, \varepsilon_{\ell+1})$ is an RSA representation of γ with respect to the basis $(g_1, g_2, \cdots, g_\ell, v)$. In [Bra00, Section 2.3.3], Brands shows that given a random $\gamma \in \mathbb{Z}_N^* \setminus \{1\}$, and a basis $(g_1, g_2, \cdots, g_\ell, v)$, finding an RSA representation of γ with respect to $(g_1, g_2, \cdots, g_\ell, v)$ is at least as hard as breaking the RSA problem in $\mathbb{Z}_N^*$. Also, given γ and a representation $(\varepsilon_1, \cdots, \varepsilon_\ell, \varepsilon_{\ell+1})$ of γ with respect to basis $(g_1, g_2, \cdots, g_\ell, v)$, finding another representation $(\mu_1, \cdots, \mu_\ell, \mu_{\ell+1})$ of γ with respect to the same basis is at least as hard as breaking the RSA problem in $\mathbb{Z}_N^*$.

Credential issuing. Assume after making the necessary identity checks, the certification authority accepts to issue a credential to the user. Let $(x_1, \cdots, x_\ell) \in (\mathbb{Z}_v^*)^\ell$ be the attributes the CA wants to encode in the credential, and let $h := g_1^{x_1} \cdots g_\ell^{x_\ell} \mod N$. The x_i's are known to both the user and the CA. The user then chooses a random blinding factor $\alpha_1 \in \mathbb{Z}_N^*$ and computes the credential's public key $h' := h\alpha_1^v$. The certification authority's digital signature on the credential is a pair $(c_0', r_0') \in \mathbb{Z}_s \times \mathbb{Z}_N^*$, satisfying the relation $c_0' = \mathcal{H}(h', {r_0'}^v (h_0 h')^{-c_0'})$. At the end of the issuing protocol, the certification authority knows neither h' nor the signature (c_0', r_0'). Figure 6 shows how a user may obtain a RSA-based credential.

User U	Public Info	CA
	$N, v, h_0, (g_i)_{1 \le i \le \ell}, \mathcal{H}, s$	
$h := g_1^{x_1} \cdots g_\ell^{x_\ell}$	$\xleftrightarrow{\text{agree on attributes } (x_1, \cdots, x_\ell)}$	$h := g_1^{x_1} \cdots g_\ell^{x_\ell}$
$\alpha_1, \alpha_2 \in_R \mathbb{Z}_N^*$,	$\xleftarrow{a_0}$	$a_0 \in_R \mathbb{Z}_N^*$
$\alpha_3 \in_R \mathbb{Z}_v$,		
$h' := h\alpha_1{}^v$		
$c_0' := \mathcal{H}(h', \alpha_2{}^v (h_0 h)^{\alpha_3} a_0)$		
$c_0 := c_0' + \alpha_3 \mod v$	$\xrightarrow{c_0}$	
$r_0^v (h_0 h)^{-c_0} \stackrel{?}{=} a_0$	$\xleftarrow{r_0}$	$r_0 := ((h_0 h)^{c_0} a_0)^{1/v}$
$r_0' := r_0 \alpha_2 \alpha_1^{c_0'} (h_0 h)^{(c_0' + \alpha_3) \text{ div } v}$		
store $h', \sigma_{\text{CA}}(h') := (c_0', r_0')$		

Fig. 6. Brands-RSA Credential Issuing protocol

Credential showing. Similar to the Brands-CP system, a user can show his credential to a verifying party, by first revealing the credential's public key h' and CA-signature (c_0', r_0'). The verifier checks if the validity relation $c_0' \stackrel{?}{=} \mathcal{H}(h', {r_0'}^v (h_0 h')^{-c_0'})$ holds. Once the validity check succeeds, the user produces a signed proof of knowledge of a RSA representation $(x_1, \cdots, x_\ell, \alpha_1)$ of h' with respect to basis the $(g_1, \cdots, g_\ell, v)$. The signed proof can also be computed with respect to a predicate $\mathcal{P}$ on exponents $(x_1, \cdots, x_\ell)$, agreed-upon by the user and the verifier at the time of the showing. Figure 7 sketches the basic Brands-RSA credential showing protocol.

8.2 Combining ElGamal encryption with Brands RSA-based Credentials

We assume the Authorizer possesses a Brands-RSA credential h', and certificate $\sigma_{\text{CA}}(h') := (c_0', r_0')$, with h' of the form $h' = (g_1^{\text{ID}_\mathcal{A}} g_2^{x_2} \cdots g_\ell^{x_\ell} \alpha^v) \mod N$. Recall that $\text{ID}_\mathcal{A}, x_2 \cdots x_\ell$ are elements of $\mathbb{Z}_v$, and that $\text{ID}_\mathcal{A}$ represents a record index in the Sender's DB. To accommodate all possible DB indexes $\text{ID}_\mathcal{J}$ (ranging over

User U	Public Info	Verifier
	$N, v, h_0, (g_i)_{1\leq i\leq \ell}, \mathcal{H}, H, s$	
$w_1, \cdots, w_\ell \in_R \mathbb{Z}_v$ $w_{\ell+1} \in_R \mathbb{Z}_N^*$ $a := g_1^{w_1} \cdots g_\ell^{w_\ell} w_{\ell+1}^v$ $c := H(h', a, m)$ $\forall\ 1 \leq i \leq \ell, r_i := cx_i + w_i$ $r_{\ell+1} := \prod_{j=1}^{\ell} g_j^{(cx_j + w_j \text{ div } v)} x_{\ell+1}^c w_{\ell+1}$	$\xleftarrow{m}$	$m := nonce\|\|...$
	$\xrightarrow{h', \sigma_{\mathrm{CA}}(h') := (c_0', r_0'), a, r_1, \cdots, r_{\ell+1}}$	accept iff $c_0' = \mathcal{H}(h', {r_0'}^v (h_0 h')^{-c_0'})$, and for $c := H(h', a, m)$, $\prod_{i=1}^{\ell} g_i^{r_i} r_{\ell+1}^v (h')^{-c} = a$ holds.

Fig. 7. Brands-RSA basic Credential Showing protocol

$[1, n] = [1, q]$), the prime v is chosen to be greater than q, the order of G_q in the ElGamal setting. Also, for reasons that will become clear shortly, the prime v is chosen to be coprime to $p - 1$, where p denotes the public prime parameter in the ElGamal setting. Finally, the RSA factors P and Q are chosen to be greater than p.

As in the first construction based on Brands-CP credentials, the Authorizer (data subject) and Receiver use the ElGamal cryptosystem to compute the SPIR query. Assuming the same setting for the Sender and Receiver as in Section 5, the Authorizer computes $(\mathsf{c}_1, \mathsf{c}_2) := \mathsf{E}_{\mathsf{pk}_{\mathsf{ElG}}}((g_{db})^{\mathrm{ID}_{\mathcal{A}}})$. To prove ID-consistency between h' and the SPIR query, it suffices for the Authorizer to produce a signed proof of knowledge of a RSA-representation of $(h'/\mathsf{c}_2 \mod Np)$ with respect to basis $((g_1 g_{db}^{-1}), g_2, \cdots, g_\ell, y_{\mathsf{ElG}}^{-1}, v)$. First we make the following observations:

1. By construction $\mathsf{c}_2 \in \mathbb{Z}_p^*$, and thus $gcd(\mathsf{c}_2, p) = 1$. Moreover, primes P and Q are greater than p, which implies that $gcd(\mathsf{c}_2, P) = gcd(\mathsf{c}_2, Q) = 1$. As a result, $gcd(\mathsf{c}_2, Np) = 1$, and $({\mathsf{c}_2}^{-1} \mod Np)$ exists.
2. The existence of inverses modulo Np of g_{db}, g_{ElG}, and y_{ElG} can be shown in a similar fashion.
3. Finally, note that $gcd(v, \phi(Np)) = gcd(v, \phi(N)(p-1)) = 1$, since v is coprime to both $\phi(N)$ and $(p-1)$ owing to the parameter choices above.

Following these observations, it is clear that we are again in the settings of the Brands-RSA credential system, except that the RSA modulus ($N' = Np = PQp$) this time has a non-standard form in the sense that it contains more than 2 primes. The security of cryptosystems based on non-standard multi-prime RSA moduli has already been studied in [Tak98,HLT02]. Both studies suggest that cryptosystems based on RSA moduli with more than two factors are less vulnerable to known attacks on standard RSA. In particular, Hinek *et al.* show in [HLT02] that as the numbers of prime factors in the modulus increases, extending standard RSA attacks to the multi-prime case becomes more complex, which results in the attacks applying in fewer instances, or becoming totally ineffective. Moreover, it is shown that, using Chinese remaindering, the decryption process in the multi-prime case is more efficient than in the standard RSA setting.

Putting the pieces together. As in the Brands-CP case, the Authorizer proves ID-consistency between his credential and the query by sending to the Receiver a signed proof of knowledge of the form $\mathrm{SPK}\{(\varepsilon_1, \cdots, \varepsilon_{\ell+1}, \mu) : h'/\mathsf{c}_2 = (g_1 g_{db}^{-1})^{\varepsilon_1} g_2^{\varepsilon_2} \cdots g_\ell^{\varepsilon_\ell} (y_{\mathsf{ElG}}^{-1})^{\varepsilon_{\ell+1}} \mu^v \mod Np\}(m)$. This is done using the protocol in Figure 7. The Authorizer can use the message m to encode any usage policy he wants the Receiver to follow. For instance, the Authorizer may include in m the identity of the Receiver to exclusively tie the authorization to the latter, or an expiry date to make sure the authorization remains valid only for the desired amount of time.

After accepting the signed proof, the Receiver proceeds with the SPIR protocol of Figure 5 without any further changes.

8.3 Security and privacy properties

Theorem 2 *Assuming the (multi-prime) RSA problem is hard, the ElGamal and HomEnc(.) cryptosystems IND-CPA secure, and that none of the three parties colludes with the other, the protocol of Section 8.2 solves the Accredited SPIR problem in the random oracle model, and provides computational privacy for both the Authorizer and Receiver, and perfect privacy for the Sender.*

The proof of Theorem 2 relies on roughly the same arguments as in the case of Theorem 1, except that we are dealing here with the RSA representation problem. Complete details of the proof will be given in the full version of the paper.

In addition to Theorem 2, the properties of *user-centricity*, *revocability*, and *authenticated PIR* described in Section 6.1, do apply for the new scheme as well.

8.4 Variant based on the Okamoto-Uchiyama cryptosystem

The construction of Section 8.2, can be modified by using the Okamoto-Uchiyama cryptosystem [OU98] instead of ElGamal. The Okamoto-Uchiyama cryptosystem is a probabilistic public key cryptosystem whose security is equivalent to the problem of factoring moduli of the form $n = p^2q$, for p and q prime. The Okamoto-Uchiyama cryptosystem is additively homomorphic.

Setting of the Okamoto-Uchiyama cryptosystem. Given security parameter κ, choose κ-sized primes p and q, and let $n = p^2q$. The choice of p and q should be such that $gcd(p, q-1) = gcd(q, p-1) = 1$. Choose random $g \in \mathbb{Z}_n^*$, such that $g_p = g^{p-1} \mod p^2$ has order p. Let $h = g^n \mod n$. The tuple (n, g, h, κ) is published as the public key, while (p, q) are kept secret. To encrypt a message $0 < m < 2^{\kappa-1}$, select random $r \in \mathbb{Z}_n$, and compute $\mathsf{E}_{\mathsf{pk_{OU}}}(m, r) := g^m h^r \mod n$. The decryption function uses a special "logarithmic function"[OU98].

Putting the pieces together. The Authorizer uses the Receiver's public key (in this case the Okamoto-Uchiyama public key) to produce the SPIR query and prove ID-consistency between the latter and an Authorizer's credential. Let $\mathsf{c} := \mathsf{E}_{\mathsf{pk_{OU}}}(\mathrm{ID}_{\mathcal{A}}, r) = g^{\mathrm{ID}_{\mathcal{A}}} h^r \mod n$ be a randomized encryption of the Authorizer's ID. Moreover, let $(h', \sigma_{\mathrm{CA}}(h'))$ be the Authorizer's Brands-RSA credential, with $h' = (g_1^{\mathrm{ID}_{\mathcal{A}}} g_2^{x_2} \cdots g_\ell^{x_\ell} \alpha^v) \mod N$. The Authorizer computes $h'/\mathsf{c} = ((g_1 g^{-1})^{\mathrm{ID}_{\mathcal{A}}} g_2^{x_2} \cdots g_\ell^{x_\ell} h^{-r} \alpha^v) \mod Nn$, and produces a signed proof of knowledge of a RSA representation of h'/c with respect to basis $((g_1 g^{-1}), g_2, \cdots, g_\ell, h^{-1}, v)$. With very high probability, parameters g and h are coprime with N, otherwise they can be used to factor N and break the security of the Brands-RSA credential system. Therefore, with very high probability, g^{-1} and h^{-1} exist modulo Nn. Similarly, with very high probability $gcd(v, \phi(Nn)) = 1$, otherwise v can be used to factor n and break the Okamoto-Uchiyama cryptosystem. Therefore, the RSA representation above is well defined. Once the signed proof is accepted, the Receiver deposits the SPIR query together with the signed proof to the Sender. The Sender in turn checks the validity of the credential, and signed proof, and proceeds with the remaining steps of the original SPIR scheme of Figure 3.

9 Conclusion

The paper describes a new access control scheme, where access policies are defined by the data subjects. More specifically, the proposed scheme allows database managers to be convinced that each of their stored data is being retrieved according to the policies of the data subjects, without learning which data has been retrieved or the identity of its owner. We present three constructions based on the discrete logarithm and RSA problems. The constructions we propose rely on anonymous authorizations, and combine SPIR systems and privacy-preserving digital credentials. The authorizations contain non-modifiable, unforgeable, user-defined policies governing their use. Moreover, authorizations can be anonymously revoked by their issuers whenever needed. The proposed solutions yield only a negligeable increase in complexity compared to that of the underlying SPIR scheme.

This work can be extended in a number of ways. For example it would be interesting to add a mechanism to support the "authorized and anonymous editing of records". One could also try to improve efficiency, and propose additional constructions based on other building blocks and assumptions.

Acknowledgement. The author is grateful to Stefan Brands for interesting discussions and for pointing out related literature. This work is supported by the IWT SBO ADAPID project (Advanced Applications for e-ID cards in Flanders). The author was previously funded in part by the Universitary Mission of Tunisia in North America.

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold, *Priced oblivious transfer: How to sell digital goods.*, Advances in Cryptology – EUROCRYPT'01, LNCS, vol. 2045, Springer-Verlag, 2001, pp. 119–135.

[BM05] Walid Bagga and Refik Molva, *Policy-based cryptography and applications.*, Financial Cryptography, LNCS, vol. 3570, Springer-Verlag, 2005, pp. 72–87.

[BM06] ———, *Collusion-free policy-based encryption.*, Proceedings of the 9th International Information Security Conference, LNCS, vol. 4176, Springer-Verlag, 2006, pp. 233–245.

[Bra94] Stefan Brands, *Untraceable off-line cash in wallet with observers*, Advances in cryptology – Crypto'93, LNCS, vol. 773, Springer-Verlag, 1994, pp. 302–318.

[Bra00] ———, *Rethinking public key infrastructures and digital certificates: Building in privacy*, The MIT Press, 2000, Available for download at http://www.credentica.com/the_mit_pressbook.html.

[Cha81] David Chaum, *Untraceable electronic mail, return addresses, and digital pseudonyms*, Communications of the ACM **24** (1981), no. 2, 84–90.

[Cha85] ———, *Security without identification: Transaction systems to make big brother obsolete.*, Communications of the ACM **28** (1985), no. 10, 1030–1044.

[CL02] Jan Camenisch and Anna Lysyanskaya, *Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation*, Advances in Cryptology – EuroCrypt'01, LNCS, vol. 2045, Springer Verlag, 2002, pp. 93–118.

[CL04] ———, *Signature schemes and anonymous credentials from bilinear maps.*, Advances in cryptology – Crypto'04, LNCS, vol. 3152, Springer-Verlag, 2004, pp. 56–72.

[CMO00] Giovanni Di Crescenzo, Tal Malkin, and Rafail Ostrovsky, *Single database private information retrieval implies oblivious transfer*, Advances in Cryptology – EUROCRYPT'00, LNCS, vol. 1807, Springer-Verlag, 2000, pp. 122–138.

[CP92] David Chaum and Torben P. Pedersen, *Wallet databases with observers*, Advances in cryptology – Crypto'92, LNCS, vol. 740, Springer-Verlag, 1992, pp. 89–105.

[GIKM98] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin, *Protecting data privacy in private information retrieval schemes*, STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing, ACM Press, 1998, pp. 151–160.

[GMM06] Philippe Golle, Frank McSherry, and Ilya Mironov, *Data collection with self-enforcing privacy*, ACM Conference on Computer and Communications Security—ACM CCS 2006, ACM, 2006, pp. 69–78.

[HLT02] M. Jason Hinek, Mo King Low, and Edlyn Teske, *On some attacks on multi-prime rsa.*, Selected Areas in Cryptography, LNCS, vol. 2595, Springer-Verlag, 2002, pp. 385–404.

[KO97] E. Kushilevitz and R. Ostrovsky, *Replication is not needed: single database, computationally-private information retrieval*, Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97), IEEE Computer Society, 1997, pp. 364–373.

[Lip05] Helger Lipmaa, *An oblivious transfer protocol with log-squared communication.*, Proceedings of the 8th International Information Security Conference, LNCS, vol. 3650, Springer-Verlag, 2005, pp. 314–328.

[OU98] Tatsuaki Okamoto and Shigenori Uchiyama, *A new public-key cryptosystem as secure as factoring*, Advances in Cryptology – EUROCRYPT'98, LNCS, vol. 1403, Springer-Verlag, 1998, pp. 308–318.

[SWP00] Dawn Xiaodong Song, David Wagner, and Adrian Perrig, *Practical techniques for searches on encrypted data*, Proceedings of the 2000 IEEE Symposium on Security and Privacy, IEEE Computer Society, 2000, pp. 44–55.

[Tak98] Tsuyoshi Takagi, *Fast RSA-type cryptosystem modulo $p^k q$*, Advances in Cryptology – CRYPTO'98, vol. 1462, 1998, pp. 318–326.