On Security of Sublinear Oblivious Transfer

***** Draft, March 2, 2006 *****

Sven Laur[1,3] and Helger Lipmaa[2,3]

[1] Helsinki University of Technology, Finland
[2] Cybernetica AS, Estonia
[3] University of Tartu, Estonia

Abstract. We study the maximal security attainable by adaptive m-out-of-n oblivious transfer protocols with sublinear communication. It is known how to construct such protocols that are private, but not known how to construct such protocols that are (fully) secure. We define the intermediate notion of coherent oblivious transfer protocols that in particular may have applications in private inference control. Coherence, like the (full) security of oblivious transfer is based on the comparison with the ideal model, with the relaxation that Sender can send the trusted third party a set of predicates so that Receiver aborts if any of the predicates hold on the queries, sent thus far. Thus, in a coherent oblivious transfer protocol, Receiver gets to know that Sender cheats but cannot complain without violating his own privacy. We then propose a simple and efficient transformation of an arbitrary 1-out-of-n private oblivious transfer protocol to a coherent protocol with only slightly larger communication. The resulting protocol is coherent, assuming that the original protocol is private and that there exists a family of collision-resistant hash functions.

Keywords: commitment scheme, hash tree, oblivious transfer, private inference control, secure two-party computation.

1 Introduction

During an $(n,m)\text{-}\mathrm{OT}^\ell$ protocol, Receiver retrieves m entries from Sender's database $S = (S[1],\dots,S[n])$, $S[i] \in \{0,1\}^\ell$, so that a computationally bounded Sender does not obtain any new information, and an unbounded Receiver does not obtain any information except these m entries. Oblivious transfer is an important subprotocol of many cryptographic protocols (e.g., for privacy-preserving data mining). Moreover, oblivious transfer is necessary and sufficient for multi-party computation [Kil88]. Thus, studying the security of oblivious transfer protocol, and in particular also studying how secure can be a communication-efficient oblivious transfer protocol, is of great importance.

The security of an oblivious transfer protocol, like the security of any two-party protocol [Gol04], is defined by comparison to the ideal model with a trusted third party. There, Sender first sends his database to the third party, and then the third party answers the queries of Receiver. In this ideal case, security of honest Sender corresponds to sender-privacy (since Sender has no private output), while for Receiver-security it is additionally required that Receiver obtains a correct output. One natural way of achieving Receiver-security is by requiring Sender to forward to the Receiver, at the beginning of the protocol, a commitment to every element of his database. Next, during each separate protocol execution, Receiver obtains information that is necessary to open a single element of the committed database. Additionally, Sender proves in zero-knowledge the correctness of his actions. Therefore, Receiver can be sure that the retrieved elements were contained in a single database snapshot of Sender, and that in particular, they did not depend anyhow on the actual queries of Receiver. However, such an implementation has total communication at least $\Omega(n) + m \cdot C(n,\ell)$, where $C(n,\ell)$ is the communication of a single execution of the underlying $(n,1)\text{-}\mathrm{OT}^\ell$ protocol.

Moreover, for $m = 1$, privacy and security are essentially equal as requirements unless one considers committed or verifiable oblivious transfer, see Sect. 3. Because of this and because of the linear communication seemingly needed by stronger security notions, it has become standard to construct *private $(n,1)\text{-}\mathrm{OT}^\ell$ protocols* where one only considers correctness in the semi-honest case and privacy of Receiver and Sender in the malicious case. Recently, Lipmaa [Lip05] and Gentry and Ramzan [GR05] proposed two-message receiver-private $(n, 1)\text{-}\mathrm{OT}^{\ell}$ protocols, with low-order polylogarithmic communication, that can be straightforwardly extended to sender-private protocols by using transformations from [NP99a,AIR01,LL05].

In the case $m > 1$, security is a much stronger requirement than privacy. For example, if Receiver retrieves adaptively up to m entries from Sender's database then it is natural and often necessary to require that Sender cannot change his input (database) during the execution of the protocol, and in particular, that he cannot use different inputs to answer different queries of Receiver.

We study how much of security can have an (adaptive) oblivious transfer protocol with sublinear communication. (This is a dual question to what cryptographic assumptions are necessary for receiver-private oblivious transfer with sublinear communication [KO00].) We introduce an intermediate security notion—that is weaker than full security and stronger than privacy—for $(n, m)\text{-}\mathrm{OT}^{\ell}$ protocols. Namely, we modify the ideal model so that together with his database, Sender also sends to the trusted third party (TTP) some randomized "aborting" predicates. After receiving the ith query from Receiver, if the ith predicate hold on the queries that Receiver has sent thus far then the TTP returns a $\perp$ message. We say that the protocol is *coherent* if any attack against honest Receiver in the real world has a counterpart in the ideal world.

While our initial motivation for this work was mainly theoretical, we want to emphasize that coherent oblivious transfer is very useful in situations like private inference control [WS04] where Sender needs to restrict the access of malicious Receivers. In this case, Sender can deny Receiver's accept to some query patterns without obtaining information about the actual queries. We think that this application is interesting enough to motivate the study of coherent oblivious transfer also from practical viewpoint.

We propose a generic transformation from an arbitrary private $(n, 1)\text{-}\mathrm{OT}^{\ell'}$ protocol, for some $\ell' > \ell$, to a coherent $(n, m)\text{-}\mathrm{OT}^{\ell}$ protocol. In the resulting protocol, Sender first publishes a *short* list commitment $\mathsf{Com}(S)$ to the whole database. That is, $\mathsf{Com}(S)$ commits to the database in a way that later one can publish short certificates (partial openings) that make it possible to verify that a particular database element belongs to S. During every adaptive execution of the $(n, 1)\text{-}\mathrm{OT}^{\ell'}$ protocol, Receiver obtains a single element of the database together with a certificate that is sufficient to verify that the retrieved database element is consistent with the list commitment. The main properties of this transformation are summarized below.

Receiver-privacy follows straightforwardly from the computational-privacy of the underlying $(n, 1)\text{-}\mathrm{OT}^{\ell'}$ protocol, provided that Receiver does not let Sender know whether he aborted at any moment. Therefore, we have a trade-off between utility (Receiver would like to obtain all m items, and if Sender cheats, he would like to complain) and receiver-privacy. For computational/statistical *sender-privacy*, we have to assume that the used list commitment scheme is computationally/statistically hiding and the underlying $(n, 1)\text{-}\mathrm{OT}^{\ell'}$ protocol is computationally/statistically sender-private. For this, we also construct a statistically hiding version of the hash tree that might be of independent interest. See Sect. 2 for the construction and for a brief comparison with previous work.

Since $\mathsf{Com}(S)$ was published before any of the queries was made by Receiver then the fact that protocol is *coherent* follows from the fact that the list commitment scheme is computationally binding. The security proof is somewhat nontrivial since we need a simulator that works in time $\Theta(n^m)$. Therefore, say, simulator works in polynomial-time if (1) m is constant or (2) $m = o(n/\ln n)$ and the list commitment scheme is binding against non-uniform adversaries that work in sub-exponential time.

Assuming that the underlying $(n, 1)\text{-}\mathrm{OT}^{\ell'}$ protocol has communication complexity $C(n, \ell')$ and that there exists a binding list commitment scheme $\mathcal{LC}$ (e.g., a hash tree), the transformation results in an coherent $(n, m)\text{-}\mathrm{OT}^{\ell}$ protocol with total communication complexity $|\mathsf{pk}| + |\mathsf{Com}(S)| + m \cdot C(n, k_{\mathcal{LC}}(n) + \ell)$, where pk is the public key of the underlying list commitment scheme and $k_{\mathcal{LC}}(n)$ is the bit-length of the certificate. In particular, $\ell' = k_{\mathcal{LC}}(n) + \ell$. In the case of hash trees, $k_{\mathcal{LC}}(n)$ is equal to $(\log_2 n - 1)\lambda + \ell$, where λ is the image length of the underlying hash function, and $\mathsf{Com}(S)$ s is equal to the root hash of a modified hash tree constructed over S. In particular, when using recent polylogarithmic-communication $(n, 1)\text{-}\mathrm{OT}^{\ell'}$ protocols [CMS99,Lip05,GR05] in conjunction with communication-efficient list commitment schemes, we get an coherent $(n, m)\text{-}\mathrm{OT}^{\ell}$ protocol with communication $\Theta(m \cdot \mathrm{poly}(\log n))$.

Finally, in a coherent oblivious transfer protocol, Receiver gets to know that Sender cheats but cannot complain without violating his own privacy. Because of this, it is important to restrict the set of aborting predicates that Sender can send to the TTP. We also demonstrate a positive result in this direction.

Notation. For $t \in \mathbb{Z}^+$, let $[t] := \{1, \ldots, t\}$. For a distribution (random variable) X, let $x \leftarrow X$ denote the assignment of x according to X. We often identify sets with the uniform distributions on them, and algorithms with their output distributions, assuming that the algorithm that outputs this distribution is clear from the context or just straightforward to construct. Let k be the security parameter. Throughout this paper, we denote Sender's database size by n, the number of adaptive queries by m and the length of database elements by ℓ.

2 List Commitments

In the later generic transform we need *list commitment schemes*. Recall that an ordinary commitment scheme COM is specified by a triple of algorithms $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open})$, where: (a) initialization algorithm Gen generates public parameters pk; (b) commitment algorithm $\mathsf{Com}_{\mathsf{pk}} : \mathcal{M} \times \mathcal{R} \rightarrow \mathcal{C} \times \mathcal{D}$ is used to get a randomized pair $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m, r)$ that we usually denote as $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m)$, where the commitment c is sent to another party and d is kept for later use, and (c) an opening function $\mathsf{Open}_{\mathsf{pk}} : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{M} \cup \{\perp\}$ opens an $c \in \mathcal{C}$, given access to d, so that for all $m \in \mathcal{M}$ and $r \in \mathcal{R}$, $\mathsf{Open}_{\mathsf{pk}}(\mathsf{Com}_{\mathsf{pk}}(m; r)) = m$.

A *list commitment scheme* is a commitment scheme for lists $S = (S[1], \ldots , S[n])$, with $n \leq \mathrm{poly}(k)$, that additionally allows partial opening of the committed value. That is, a list commitment scheme is a quadruple $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Cert}, \mathsf{Open})$, with $\mathsf{Cert}_{\mathsf{pk}} : \mathcal{D} \times \mathbb{N} \rightarrow \mathcal{D}$, such that: for every $S \in \mathcal{M}^n$ and every $i \in [n]$, if $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(S)$ then $\mathsf{Cert}_{\mathsf{pk}}(d, i)$ returns a value d_i such that $\mathsf{Open}_{\mathsf{pk}}(c, d_i) = (i, S[i])$. In the case of hiding list commitment schemes (see Lem. 1), Gen can be an interactive protocol between Receiver and Sender.

The simplest list commitment consists of the list of ordinary commitment values $(c_1, \ldots , c_n)$ to $(S[1], \ldots S[n])$, however it has commitment length $|\mathsf{Com}_{\mathsf{pk}}(S)|$ that is linear in n. Another famous example of a binding (though not hiding) list commitment is hash tree [Mer80], where Gen generates a random h from a collision-resistant hash function family, $\mathsf{Com}_h(S)$ returns the hash tree root on S as a commitment value c, and sets $d \leftarrow S$. A partial opening $\mathsf{Cert}_h(S, i)$ is the hash certificate for the ith node, i.e., the minimum amount of information required to verify that $S[i]$ belonged to the database that was committed by $\mathsf{Com}_h(S)$.

Like commitments schemes, a list commitment scheme is required to be binding and hiding. More precisely, for a non-uniform adversary $A = \{A_k\}$, define

$$\mathsf{Adv}^{\mathsf{bind}}_{\mathsf{COM}}(A, k) := \Pr\left[\begin{array}{l}\mathsf{pk} \leftarrow \mathsf{Gen}(1^k), (c, d_0, d_1) \leftarrow A_k(\mathsf{pk}),\\ (i_0, m_0) \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_0), (i_1, m_1) \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_1) :\\ i_0 \neq i_1 \wedge \perp \neq m_0 \neq m_1 \neq \perp\end{array}\right]$$

and

$$\mathsf{Adv}^{\mathsf{hide}}_{\mathsf{COM}}(A, k) := 2 \cdot \left|\Pr\left[\begin{array}{l}\mathsf{pk} \leftarrow \mathsf{Gen}(1^k), (S_0, S_1) \leftarrow A_k(\mathsf{pk}), b \leftarrow \{0, 1\},\\ (c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(S_b) : S_0, S_1 \in \{0, 1\}^{\ell n} \wedge\\ A_k^{\mathsf{Cert}_{\mathsf{pk}}(d, \cdot)}(\mathsf{pk}, S_0, S_1, c) = b\end{array}\right] - \frac{1}{2}\right| ,$$

where probability is taken over coin tosses of all relevant algorithms, and A can query $\mathsf{Cert}(d, i)$ on inputs i where $S_0[i] = S_1[i]$. A list commitment scheme is *computationally hiding* if for all family $A = \{A_k\}$ of polynomial-size circuits, $\mathsf{Adv}^{\mathsf{hide}}_{\mathsf{COM}}(A, k) \leq k^{-\omega(1)}$. A commitment scheme is *computationally binding* if for any family $A = \{A_k\}$ of polynomial-size circuits, $\mathsf{Adv}^{\mathsf{bind}}_{\mathsf{COM}}(A) \leq k^{-\omega(1)}$. In the case of unbounded adversaries, we speak respectively about *statistical hiding* and *statistical binding*.

Note that the hiding property for list commitments is defined exactly the same way as for ordinary commitments, except that the adversary has access to an additional oracle. On the other hand, the binding property has now a different interpretation—it should be infeasible to open same part of commitment differently. It might even be infeasible to fully open the commitment under successful attack.

Going back to the example of hash trees, the certificate (partial opening) d_i is valid when it is consistent with the root hash c, otherwise $\mathsf{Open}_h(c, d_i) = \perp$. Clearly, the binding property follows from the non-uniform collision-resistance of the function family $\mathcal{H}$: if an adversary can find a double opening

$(j, x_j) \neq (j, \widehat{x}_j)$ then there must be a collision in the root path. Thus, collision resistance of $\mathcal{H}$ implies that polynomial-size hash trees are binding. On the other hand, a pure hash tree is not hiding since the hash certificate $\mathsf{Cert}_h(S, i)$ reveals not only $S[\sigma]$ but also $S[\sigma \oplus 1]$, the sibling of $S[\sigma]$ in the tree. However, one can straightforwardly make the hash trees hiding as follows. See, e.g., [MRK03] for a similar construction. Since we do not need to provide the zero-knowledge property then our construction is somewhat simpler.

Lemma 1. *Assume the existence of a computationally/statistically hiding commitment scheme. There exists a computationally/statistically hiding list commitment scheme with comparable efficiency to the hash tree.*

Proof (Sketch). Construction of this list commitment scheme is straightforward: given a database S, first compute a database S', where with $S'[i] = \mathsf{Com}_{\mathsf{pk}}(S[i]; r_i)$ is an ordinary computationally/statistically hiding commitment to $S[i]$. Then use the non-hiding hash-tree list commitment scheme on S', but include also the pair $(S[i], r_i)$ to the new $\mathsf{Cert}_{\mathsf{pk}}(d, i)$. Here, pk is defined jointly by Receiver and Sender, and therefore the resulting list commitment scheme has two keys, h defined by Receiver and pk defined jointly by Receiver and Sender. (If the commitment scheme is perfectly hiding then Receiver can also generate pk by himself.) $\square$

3 Preliminaries on Oblivious Transfer

The participants of an m-out-of-n oblivious transfer protocol for ℓ-bit strings, $(n, m)\text{-}\mathrm{OT}^{\ell}$, are Receiver and Sender. Sender holds a database S of length n, with every element being an ℓ-bit string, and Chooser holds an index $i \in [n]$. Intuitively, the protocol enables Chooser to read m locations $\sigma_1, \ldots \sigma_m$ of S without a computationally-bounded Sender learning anything about any σ_j. Moreover, an unbounded Receiver should not learn anything about other elements of the database even if he can choose the subsequent queries adaptively.

The standard security definition of an oblivious transfer protocol is given via comparison with its ideal model implementation [Gol04]. The ideal model implementation is defined as follows. Assume (n, m, ℓ) is a system parameter. The TTP receives from Sender $S = (S[1], \ldots, S[n])$, where $S[i] \in \{0, 1\}^{\ell}$. The TTP now receives from Receiver m adaptive queries $\sigma_1, \ldots, \sigma_m$. For every query σ_j, the TTP sends $\perp$ to Receiver if Sender issues an abort command or if $j > m$. Otherwise, the TTP forwards the element $S[\sigma_j]$ to Receiver.

Following [Gol04], we say that an $(m, n)\text{-}\mathrm{OT}^{\ell}$ protocol is (1) *correct*, if in the case of honest Receiver and Sender, Receiver always gets the queried elements; (2) computationally *receiver-private*, if—assuming that Receiver is honest—the views of a polynomial-size non-uniform malicious Sender, corresponding to any two inputs of an honest Receiver, are computationally indistinguishable; (3) computationally *receiver-secure*, if—assuming that Receiver is honest—any attack by a polynomial-size non-uniform malicious Sender in the real world can be converted, by a polynomial-size non-uniform simulator, to an attack in the ideal world such that the joint output distributions of Receiver and Sender are computationally indistinguishable; (4) statistically *sender-private*, if—assuming that Sender is honest—any attack by a malicious Receiver in the real world can be converted, by a simulator, to an attack in the ideal world such that the output distributions of Receiver are statistically indistinguishable. An $(m, n)\text{-}\mathrm{OT}^{\ell}$ protocol is *secure* if it is receiver-secure and sender-private, and *private* if it is correct, receiver-private and sender-private.

In *committed oblivious transfer* [CvdGT95,GMY04], Receiver is committed to σ, Sender is committed separately to every $S[i]$, and at the end Receiver obtains a commitment to $S[\sigma]$. Committed oblivious transfer can be used to perform tasks like oblivious circuit evaluation, mental games and distributed computation [CvdGT95]. According to the definition of [CC00, Sect. 4], a 1-*out-of-*n *verifiable oblivious transfer protocol* for ℓ-bit strings is an oblivious transfer protocol on committed values, where Sender has made n commitments $C_{S[i]}$, containing n values $S[i]$, and Receiver has made a commitment C_{σ}, containing $\sigma \in [n]$. The requirements are that Receiver outputs $S[\sigma]$ without learning anything about values $S[\sigma']$, for $\sigma' \neq \sigma$, and that Sender does not learn anything about σ. Verifiable oblivious transfer has been studied in quite a few previous papers [CvdGT95,CD97,NP99b,CC00,AJL04,Lip03,GMY04]. (Though at least the protocols [NP99b,AJL04,Lip03] are not verifiable but coherent, according to the definitions of the next section.) Differently from verifiable/committed oblivious transfer, we prefer not to explicitly require any element to be committed.

In any implementation of a secure $(n, m)\text{-}\,\mathrm{OT}^{\ell}$ protocol, $m > 1$, also Receiver's aborting probability must be almost the same for all possible queries, even if a malicious Server changes database between queries. Current state of the art methods use one of the two following alternatives to achieve this. First, Receiver and Server execute the private $(n, m)\text{-}\,\mathrm{OT}^{\ell}$ protocol and then Server proves in zero-knowledge that all his steps are correct—this results in communication $\Omega(nm)$. Alternatively [OK04], Server first sends an element-wise encrypted database such that the decryption keys are independent of the data, and later Receiver and Sender securely compute the necessary decryption key. Although, the latter method might use on more communication-efficient zero-knowledge proofs that can be even polylogarithmic in security parameter, it still has total communication $\Omega(nm\ell)$.

There exist several techniques [NP99a,AIR01,LL05] to transform *any* $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocol into a sender-private $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocol with comparable communication. Some polylogarithmic-communication $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocols were proposed in [CMS99,Lip05,GR05], and some sublinear-but-superpolylogarithmic $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocols were proposed in [KO97,Ste98]. Any private $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocol can be, by repetition, straightforwardly transformed into a private $(n, m)\text{-}\,\mathrm{OT}^{\ell}$ protocol.

For $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocols, privacy and security are often equivalent since there one does not have any problems with the consistency of Sender's inputs unless one is required to commit the inputs. Therefore, in the case $m = 1$ one is interested either in private $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocols or in secure committed $(n, 1)\text{-}\,\mathrm{OT}^{\ell}$ protocols.

4 Coherent Oblivious Transfer: Security Definitions

As mentioned in the previous section, secure oblivious transfer seems to need linear communication. On the other hand, there exist several private protocols with polylogarithmic communication. As explained in introduction, privacy might not be sufficient in all applications. The main goal of this paper is to show that privacy is not the strongest relevant security notion for oblivious transfer protocols that still allows sublinear communication.

In a protocol with sublinear communication, we most probably cannot neither commit to/encrypt all separate elements of the database nor provide a full zero-knowledge proof that the probability of aborting would be exactly the same for all Receiver's queries. Without zero-knowledge proofs, even if the protocol is otherwise secure, it becomes possible that Sender obtains additional information in the case Receiver acknowledges that he aborts in the middle of the protocol. This happens if Sender can make Receiver to abort depending on Receiver's queries.

Thus, we have arrived to the next security definition, that, as we will show in Sect. 5, can be implemented by using sublinear communication. Consider first the next ideal implementation. Assume (n, m, ℓ) is a system parameter. For $i \in [m]$, let $\mathcal{P}_i$ be a set of predicates on $[n]^i$. The TTP receives from Sender $S = (S[1], \ldots, S[n])$, where $S[i] \in \{0, 1\}^{\ell}$, and a set of possibly randomly picked predicates $\{p_i\}$, $p_i \in \mathcal{P}_i$ for $i \in [m]$. Here, p_i is given by its truth table. The TTP then receives from Receiver m adaptive queries $\sigma_1, \ldots, \sigma_m$. For every query σ_j, the TTP sends $\perp$ to Receiver if Sender issues an abort command, $j > m$ or if $p_j(\sigma_1, \ldots, \sigma_j) = 1$. Otherwise, the TTP forwards the element $S[\sigma_j]$ to Receiver.

The definitions of correctness and sender-privacy stay unmodified. Similarly to the previous definitions, we say that an $(m, n)\text{-}\,\mathrm{OT}^{\ell}$ protocol is *coherent* for $\{\mathcal{P}_i\}$, if—assuming that Receiver is honest—any attack by a non-uniform malicious Sender in the real world can be converted, by a polynomial-size non-uniform simulator, to an attack in the ideal world such that the joint output distributions of Receiver and Sender are computationally indistinguishable. We say that an oblivious transfer protocol is *coherent* if it is coherent for the set of all (efficiently computable) predicates on $[n]^i$.

Intuitively, a protocol is coherent if and only if that simulator outputs the same answer whenever Receiver does not abort, and aborts whenever Receiver aborts—except with a negligible probability. The only possibility to define this in the ideal model seems to be the way taken by us, letting Sender to send query-dependent abort commands to TTP. The adaptive case, where the predicates depend on the previous queries, is clearly equivalent from the security viewpoint.

COMMON INPUT: Both parties hold k, n, ℓ, m and $\ell' := \ell + k_{\mathcal{LC}}(n)$.
CHOOSER'S INPUT: $(\sigma_1, \dots, \sigma_m) \in [n]^m$ (possibly chosen adaptively).
SENDER'S INPUT: $S = (S[1], \dots, S[n]) \in \{0,1\}^{n\ell}$.
UNDERLYING PROTOCOLS:
A list commitment scheme $\mathcal{LC} = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Cert})$, an $(n,1)$- $\mathrm{OT}^{\ell'}$ protocol $\mathcal{OT}$.

Initialization phase:

1. Receiver and Sender generate jointly a $\mathsf{pk} \leftarrow \mathsf{Gen}(1^k)$.
2. Sender does: Set $(c,d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(S)$. Store a database π, where $\pi[j] \leftarrow \mathsf{Cert}_{\mathsf{pk}}(d,j)$ for $j \in [n]$. Send c to Receiver. Store d and $\mathsf{counter} \leftarrow 1$.
3. Receiver stores c.

Protocol execution on a single query σ:

1. Sender aborts if $\mathsf{counter} > m$. Otherwise, Sender increments $\mathsf{counter}$.
2. Receiver and Sender execute a single run of $\mathcal{OT}$ on database π.
3. Let $d^{(\sigma)}$ be Receiver's private output in $\mathcal{OT}$.
4. Receiver outputs s if $\mathsf{Open}_{\mathsf{pk}}(c, d^{(\sigma)}) = (\sigma, s)$ for some s, and $\perp$, otherwise.

Protocol 1: A new coherent (n,m)- OT^{ℓ} protocol

Utility versus privacy. As mentioned before, it is unknown how to construct a secure (n,m)- OT^{ℓ} protocol with sublinear construction. In a coherent but not secure protocol, the event that Receiver outputs $\perp$ depends on the queries he has made thus far. Therefore, if Receiver complains that Sender cheated then Sender gets to know something about Receiver's queries. Therefore, in a receiver-private and coherent protocol, an honest Receiver must not send any feedback to Sender on whether he aborted or not. Thus, there is a strict tradeoff between utility and privacy. (Though, as we point out in Sect. 6, coherent but not fully secure oblivious transfer has applications in private inference control.) In practice, one can improve upon it by allowing Receiver to occasionally ask the Sender, *after* obtaining the output of a coherent protocol, to prove in zero-knowledge the correctness of his actions, so that a caught Sender would be imposed a heavy fine.

Because of the privacy concerns, it is clearly desirable that a protocol is coherent for as small set of predicates as possible. An important intermediate goal is the case where p_i is just a predicate on the Receiver's ith query; we say that a (m,n)- OT^{ℓ} protocol is *memoryless-coherent* if this holds. In the case of a memoryless-coherent (m,n)- OT^{ℓ} protocols, if Sender gets to know that Receiver did obtain $\perp$ as a response to his ith query then some information about the ith query only will be leaked.

5 New Coherent Oblivious Transfer Protocol

Assume that we are given a list commitment scheme $\mathcal{LC}$, with $|\mathsf{Cert}_{\mathsf{pk}}(\cdot,\cdot)| = k_{\mathcal{LC}}(n)$ for any S with $\sharp S = n$ and for any $i \in [n]$, and an $(n,1)$- $\mathrm{OT}^{\ell + k_{\mathcal{LC}}(n)}$ protocol $\mathcal{OT}$ with communication $C(n, \ell + k_{\mathcal{LC}}(n))$. A candidate coherent (n,m)- OT^{ℓ} protocol is depicted by Prot. 1. It is easy to see that Prot. 1 is correct. As for communication-efficiency, Prot. 1 has total communication $|\mathsf{pk}| + |\mathsf{Com}_{\mathsf{pk}}(\pi)| + m \cdot C(n, \ell + k_{\mathcal{LC}}(n))$. By using the same techniques as in [IKO05], one can build a list commitment scheme based on every (adaptive) coherent oblivious transfer protocol. Thus, in particular, a coherent oblivious transfer protocol exists iff there exists a list commitment scheme.

Computational receiver-privacy of Prot. 1 follows directly from the receiver-privacy of the underlying $(n,1)$- OT^{ℓ} protocol $\mathcal{OT}$. More precisely:

Lemma 2 (Computational receiver-privacy). *Assume that $\mathcal{OT}$ is receiver-private and that $\mathcal{LC}$ is any list commitment scheme. Then Prot. 1 is receiver-private.*

Proof. Clear, since adversary's view consists of a random public key pk and of m queries of $\mathcal{LC}$. □

Thus for receiver-privacy, $\mathcal{LC}$ does not have to be hiding or binding. Since for the protocol to be coherent, $\mathcal{LC}$ has to be binding, we can use the "plain" hash trees in the case where $\mathcal{LC}$ has to be receiver-private and coherent but not sender-private.

For computational/statistical sender-privacy of Prot. 1, it is needed that $\mathcal{OT}$ is computationally/statistically sender-private and that $\mathcal{LC}$ is a computationally/statistically hiding list commitment scheme (see Lem. 1).

Lemma 3. *Assume that $\mathcal{LC}$ is statistically hiding and that $\mathcal{OT}$ is statistically sender-private. Then Prot. 1 is statistically sender-private.*

Proof. Standard hybrid argument. Let $\varepsilon_{\mathcal{OT}}$ quantify how close the simulator's output is to Sender's view in the case of the proof of sender-privacy of $(n,1)\text{-}\mathrm{OT}^{\ell'}$. In the game G_0 (that corresponds to a real execution), Receiver sees (pk, c) and m views corresponding to executions of the $(n,1)\text{-}\mathrm{OT}^{\ell'}$ protocol. In game G_1, the $(n,1)\text{-}\mathrm{OT}^{\ell'}$ protocol $\mathcal{OT}$ is replaced by an ideal $(n,1)\text{-}\mathrm{OT}^{\ell'}$ protocol. Therefore, in this game, Receiver's view is (pk, c) together with m elements $\mathsf{Cert}(d, \sigma_j)$. Thus, attackers advantages in games G_0 and G_1 differ at most by $m \cdot \varepsilon_{\mathcal{OT}}$. In game G_2, we replace $\mathcal{LC}$ with an ideal commitment scheme. In this game, attacker's advantage changes by $\mathsf{Adv}^{\mathsf{hide}}_{\mathcal{LC}}(A, k)$. Thus, in game G_0, attacker's advantage is not larger than $\mathsf{Adv}^{\mathsf{hide}}_{\mathcal{LC}}(A, k) + m \cdot \varepsilon_{\mathcal{OT}}$. □

Lemma 4. *Assume that $\mathcal{LC}$ is binding and that n^m is polynomial in the security parameter. Then Prot. 1 is coherent.*

Proof. We construct a simulator that, given black-box access to malicious Sender A, does the following. It feeds A random coins according to their distribution in the real execution. Recall that predicates are given by their truth tables. Then

1. Generate a random pk and send it to Verifier. Get back "commitment" c. Send an abort command to TTP if Sender aborts. Construct a database S with initial values $S[j] \leftarrow 0^\ell$ and a database H with initial values $H[j] \leftarrow \perp$, for all $j \in [n]$.
2. For all $j \in \{1, \dots, m\}$ and for all $(\sigma_1, \dots, \sigma_j) \in \{0,1\}^{\ell j}$ do:
   (a) Let $\sigma := (\sigma_1, \dots, \sigma_j)$ be the current node.
   (b) Rewind A back to the state, corresponding to the node $(\sigma_1, \dots, \sigma_{j-1})$ (or to the initial state, if $j = 1$).
   (c) Execute the $(n,1)\text{-}\mathrm{OT}^{\ell'}$ protocol with A with Receiver's private input σ_j, let π be Receiver's private output in this protocol run.
   (d) If $\pi = \perp$ then label σ by "(Sender's state,$\perp$)" and set $p_j(\sigma) = 0$, else
      i. If $H[d] \neq \perp$ and $S[d] \neq a$ then abort ($\mathcal{LC}$ is broken), else set $S[d] \leftarrow a$ and $H[d] \leftarrow \top$.
      ii. Label σ by "(Sender's state,$(a, d) := \mathsf{Open}(c, \pi)$)," set $p_j(\sigma) = 1$.
3. Send $(S, \{p_i\})$ to TTP and output A's output in this protocol, given (say) Receiver's input $(1, \dots, 1)$.

Thus, for any query profile $(\sigma_1, \dots, \sigma_m)$ of Receiver, the ideal and real world "behave" the same, except when the simulator breaks the binding property of $\mathcal{LC}$. Simulator's working time is dominated by the execution of $(n^m - 1)/(n - 1) = \Theta(n^m)$ oblivious transfer protocols. □

In particular, the simulator runs in polynomial time—and thus it suffices to assume that $\mathcal{LC}$ is binding—if m is an arbitrary constant. Alternatively, we can make a stronger assumption that $\mathcal{LC}$ is binding against non-uniform adversaries of subexponential size; such an assumption is quite common [Pas03,BP04]. Then, Prot. 1 is computationally coherent for $m = o(n/\log n)$.

On memoryless-coherent protocols. If we also consider stateful protocols then there is no black-box construction from an arbitrary $(n,m)\text{-}\mathrm{OT}^{\ell}$ protocol to a memoryless-coherent $(n,m)\text{-}\mathrm{OT}^{\ell}$. For example, assume that the $(n,1)\text{-}\mathrm{OT}^{\ell'}$ protocol is any of the protocols based on homomorphic encryption [Ste98,AIR01,Lip05], and modify it to a stateful protocol where in the jth execution of $(n,1)\text{-}\mathrm{OT}^{\ell'}$, Receiver sends queries, corresponding to $(\sigma_1, \dots, \sigma_j, \sigma'_{j+1}, \dots, \sigma'_m)$ for randomly chosen $\sigma'_{j'}$, and encrypted under the *same* public key. For example, if we modify the oblivious transfer protocol from [AIR01], then in the second query, Receiver sends $(E_K(\sigma_1), E_K(\sigma_2), \dots)$. When replying to the second query, by using the properties of homomorphic encryption, Sender can make his answer depend on the fact whether $\sigma_1 = \sigma_2$. Therefore, the new $(n,1)\text{-}\mathrm{OT}^{\ell'}$ protocol is clearly private, but the resulting $(n,m)\text{-}\mathrm{OT}^{\ell}$ protocol is not memoryless-coherent. However, we can show that Prot. 1 is memoryless-coherent if Receiver in $\mathcal{OT}$ is stateless, i.e., when the subsequent executions of $\mathcal{OT}$ with the same Receiver are completely independent.

Lemma 5. *Assume that $\mathcal{LC}$ is binding, $\mathcal{OT}$ is receiver-private and stateless and that n^m is polynomial in the security parameter. Then Prot. 1 is memoryless-coherent.*

Proof (Sketch). Assume that we have an adversary A that breaks the memoryless-coherent property of Prot. 1. That is, he can force Receiver to abort as a predicate p_i on Receiver's queries $(\sigma_1, \dots, \sigma_i)$ thus far, such that p_i is a non-trivial function of least on two different values σ_a and σ_b for $a, b \leq i$. Since the protocol is stateless then the adversary can play the role of Receiver in round $b > a$ to breach the privacy of Receiver in round a: given his knowledge of whether he aborted in round b, he will have some advantage in guessing σ_a, given the value $p_i(\sigma_a, \sigma_b)$. □

Essentially all proposed oblivious transfer protocols, including the sublinear ones, are stateless. However, protocols based on homomorphic encryption can be easily made stateful by not generating a new encryption key at every protocol execution.

6 Applications And Open Problems

A coherent oblivious transfer protocol makes it possible for Receiver to obtain, in a privacy-preserving manner, a database element and then verify whether it is consistent with a previously published database digest. This has applications in privacy-preserving time-stamping, or in the case of secure communication when a party wants to retrieve a correct public key from a PKI directory, without revealing which party he wants to communicate with.

Moreover, coherent oblivious transfer is very useful in situations like private inference control [WS04] where Sender needs to restrict the access of malicious Receivers. In this case, Sender can deny Receiver's accept to some query patterns without obtaining information about the actual queries. We think that this application is interesting enough to motivate the study of coherent oblivious transfer also from practical viewpoint.

We used both list commitment schemes and $(n,1)\text{-}\mathrm{OT}^{\ell'}$ protocols. One can construct commitment schemes from any oblivious transfer protocols [IKO05]. Therefore, one can also construct (linear-communication) list commitment schemes from any oblivious transfer protocol. However, it seems that for sublinear communication, something extra (akin to collision-resistancy) is needed. Another interesting open question is to find a generic transformation from any non-sender-private but receiver-secure $(n,1)\text{-}\mathrm{OT}^{\ell}$ protocol to a sender-private and receiver-secure $(n,1)\text{-}\mathrm{OT}^{\ell}$ protocol, as the already known transformations [NP99a,AIR01,LL05] do not preserve receiver-security.

Acknowledgements. We would like to thank Benny Pinkas for useful comments. The first author was partially supported by the Finnish Academy of Sciences. The second author was supported by the Estonian Science Foundation, grant 6848, and by the base funding from the University of Tartu.

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced Oblivious Transfer: How to Sell Digital Goods. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135, Innsbruck, Austria, May 6–10, 2001. Springer-Verlag.

[AJL04] Andris Ambainis, Markus Jakobsson, and Helger Lipmaa. Cryptographic Randomized Response Techniques. In Feng Bao, Robert H. Deng, and Jianying Zhou, editors, *Public Key Cryptography 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 425–438, Singapore, March 1–4, 2004. Springer-Verlag.

[BP04] Boaz Barak and Rafael Pass. On the Possibility of One-Message Weak Zero-Knowledge. In Moni Naor, editor, *First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 121–132, Cambridge, MA, USA, February 19–21, 2004. Springer Verlag.

[CC00] Christian Cachin and Jan Camenisch. Optimistic Fair Secure Computation. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 93–111, Santa Barbara, USA, August 20–24 2000. Springer-Verlag.

[CD97] Ronald Cramer and Ivan Damgård. Linear zero-knowledge – a note on efficient zero-knowledge proofs and arguments. In *Proceedings of the Twenty-Nineth Annual ACM Symposium on the Theory of Computing*, pages 436–445, 1997.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computational Private Information Retrieval with Polylogarithmic Communication. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414, Prague, Czech Republic, May 2–6, 1999. Springer-Verlag.

[CvdGT95] Claude Crépeau, Jeroen van de Graaf, and Alain Tapp. Committed Oblivious Transfer and Private Multi-Party Computation. In Don Coppersmith, editor, *Advances in Cryptology — CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 110–123, Santa Barbara, USA, August 27–31 1995. Springer-Verlag.

[GMY04] Juan A. Garay, Philip MacKenzie, and Ke Yang. Efficient and Universally Composable Committed Oblivious Transfer and Applications. In Moni Naor, editor, *First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 297–316, Cambridge, MA, USA, February 19–21, 2004. Springer Verlag.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[GR05] Craig Gentry and Zulfikar Ramzan. Single-Database Private Information Retrieval with Constant Communication Rate. In Luis Caires, Guiseppe F. Italiano, Luis Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *The 32nd International Colloquium on Automata, Languages and Programming, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 803–815, Lisboa, Portugal, 2005. Springer-Verlag.

[IKO05] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Sufficient Conditions for Collision-Resistant Hashing. In Joe Kilian, editor, *The Second Theory of Cryptography Conference, TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 445–456, Cambridge, MA, USA, February 10–12, 2005. Springer Verlag.

[Kil88] Joe Kilian. Founding Cryptography on Oblivious Transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 20–31, Chicago, Illinois, USA, 2–4 May 1988. ACM Press.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is Not Needed: Single Database, Computationally-Private Information Retrieval. In *38th Annual Symposium on Foundations of Computer Science*, pages 364–373, Miami Beach, Florida, October 20–22, 1997. IEEE Computer Society.

[KO00] Eyal Kushilevitz and Rafail Ostrovsky. One-Way Trapdoor Permutations Are Sufficient for Non-trivial Single-Server Private Information Retrieval. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 104–121, Bruges, Belgium, 14–18 May 2000. Springer-Verlag.

[Lip03] Helger Lipmaa. Verifiable Homomorphic Oblivious Transfer and Private Equality Test. In Chi Sung Laih, editor, *Advances on Cryptology — ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 416–433, Taipei, Taiwan, November 30–December 4, 2003. Springer-Verlag.

[Lip05] Helger Lipmaa. An Oblivious Transfer Protocol with Log-Squared Communication. In Jianying Zhou and Javier Lopez, editors, *The 8th Information Security Conference (ISC'05)*, volume 3650 of *Lecture Notes in Computer Science*, pages 314–328, Singapore, September 20–23, 2005. Springer-Verlag.

[LL05] Sven Laur and Helger Lipmaa. Additive Conditional Disclosure of Secrets And Applications. Technical Report 2005/378, IACR, November 21 2005.

[Mer80] Ralph Charles Merkle. Protocols for Public Key Cryptosystems. In *Proceedings of the 1980 Symposium on Security and Privacy*, Oakland, California, USA, April 14–16, 1980. IEEE Computer Society Press.

[MRK03] Silvio Micali, Michael O. Rabin, and Joe Kilian. Zero-Knowledge Sets. In *44th Annual Symposium on Foundations of Computer Science*, pages 80–91, Cambridge, MA, USA, October, 11–14 2003. IEEE, IEEE Computer Society Press.

[NP99a] Moni Naor and Benny Pinkas. Oblivious Transfer and Polynomial Evaluation. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 245–254, Atlanta, Georgia, USA, May 1–4, 1999. ACM Press.

[NP99b] Moni Naor and Benny Pinkas. Oblivious Transfer with Adaptive Queries. In Donald Beaver, editor, *Advances in Cryptology — CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590, Santa Barbara, USA, 15–19 August 1999. Springer-Verlag.

[OK04] Wakaha Ogata and Kaoru Kurosawa. Oblivious Keyword Search. *Journal of Complexity*, 20(2–3):356–371, 2004.

[Pas03] Rafael Pass. Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 160–176, Warsaw, Poland, May 4–8, 2003. Springer-Verlag.

[Ste98] Julien P. Stern. A New and Efficient All or Nothing Disclosure of Secrets Protocol. In Kazuo Ohta and Dingyi Pei, editors, *Advances on Cryptology — ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 357–371, Beijing, China, October 18–22, 1998. Springer-Verlag.

[WS04] David P. Woodruff and Jessica Staddon. Private Inference Control. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 188–197, Washington, DC, USA, October 25–29, 2004.