Efficient Set Membership Encryption and Applications

Matthew Green
Johns Hopkins University
Baltimore, MD, USA
mgreen@cs.jhu.edu

Abhishek Jain
Johns Hopkins University
Baltimore, MD, USA
NTT Research, Inc.
Sunnyvale, CA, USA
abhishek@cs.jhu.edu

Gijs Van Laer
Johns Hopkins University
Baltimore, MD, USA
XFA.tech
Antwerp, Belgium
gijs.vanlaer@xfa.tech

ABSTRACT

The emerging area of laconic cryptography [Cho *et al.*, CRYPTO'17] involves the design of two-party protocols involving a sender and a receiver, where the receiver's input is *large*. The key efficiency requirement is that the protocol communication complexity must be independent of the receiver's input size. In recent years, many tasks have been studied under this umbrella, including laconic oblivious transfer (ℓOT).

In this work, we introduce the notion of *Set Membership Encryption* (SME) – a new member in the area of laconic cryptography. SME allows a sender to encrypt to one recipient from a universe of receivers, while using a small digest from a large subset of receivers. A recipient is only able to decrypt the message if and only if it is part of the large subset. We show that ℓOT can be derived from SME.

We provide efficient constructions of SME using bilinear groups. Our solutions achieve orders of magnitude improvements in decryption times than state-of-the-art (on ℓOT) and significant improvements overall in concrete efficiency over initial works in the area of laconic cryptography, albeit at the cost of worse asymptotics.

CCS CONCEPTS

• **Security and privacy** → **Cryptography**; **Public key (asymmetric) techniques**.

KEYWORDS

Set Membership Encryption; Laconic OT; Oblivious Transfer; Multiparty Computation; Bilinear Diffie-Hellman

ACM Reference Format:
Matthew Green, Abhishek Jain, and Gijs Van Laer. 2023. Efficient Set Membership Encryption and Applications. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3576915.3623131

1 INTRODUCTION

Recent developments in secure multiparty computation (MPC) have led to the increasing usage and deployment of the technology. This deployment has illustrated the need for further improvements in the efficiency of MPC protocols. One broad area of potential improvement has to do with the fixed cost needed for Oblivious Transfer (OT) [41]. OT is a fundamental cryptographic protocol and is foundational to the construction of MPC protocols [24, 33, 33, 35, 41, 48]. In practice, OT is a significant contributor to the overhead of MPC; moreover, the number of OT invocations typically increases with the size of the function inputs. This has motivated efficiency optimizations such as OT extension [4, 31, 38, 44]. While such improvements reduce the *computational* complexity of multiple OT interactions, they still require interactive communication that grows linearly with the number of invocations.

Recently Cho *et al.* [10] proposed a new primitive called *laconic Oblivious Transfer* (ℓOT) to address this communication cost. In laconic OT a receiver first produces a succinct digest of a vector of selector bits. The sender then uses this digest to encrypt a corresponding database of message pairs. The critical property in this scheme is that the receiver must be able to decrypt exactly one message from each pair, corresponding to its previous selections. The key efficiency requirement are as follows: the digest size must be independent of the database size, while the sender's running time and the receiver's decryption time (for each position) must be poly-logarithmic in the size of the receiver's input.

Laconic OT has many promising applications and the instantiation proposed by Cho *et al.* is elegant and relies on well-studied cryptographic hardness assumptions. Unfortunately, it is far from practical. While asymptotically efficient, the proposed scheme includes substantial concrete overhead that makes it unusable for real-world deployment. In particular, the construction makes extensive use of elliptic curve scalar multiplications that are embedded within chains of sequential garbled circuits, resulting in enormous concrete bandwidth costs. Some optimizations have been proposed to improve this and related protocols [11]; however, even the optimized realizations are challenging to implement – let alone deploy for real-world applications [11, 46].

The impracticality stems from a focus on asymptotic efficiency instead of concrete efficiency. A recent line of work [2, 27] recognized this and proposed much more concretely efficient constructions by allowing for asymptotically larger decryption times – *linear*, as opposed to poly-logarithmic. The work of Goyal *et al.* [27] presents constructions based on various number-theoretic assumptions, while Alamati *et al.* [2] presents a construction based on the ϕ-hiding assumption.

The main drawback of these works is the requirement of large decryption time, in concrete terms, for moderate to large size databases. In this work, we take a new approach towards designing ℓOT schemes. As we discuss shortly, our approach yields schemes

CCS '23, November 26–30, 2023, Copenhagen, Denmark.

ACM ISBN 979-8-4007-0050-7/23/11…$15.00
https://doi.org/10.1145/3576915.3623131

Table 1: Overview of our asymptotic and concrete efficiency in comparison with Cho *et al.* [10], Goyal *et al.* [27], and Alamati *et al.* [2] for database size $n = 2^{31}$. (DDH = Decisional Diffie-Hellman assumption, q-DBDHI = q-Decisional Bilinear Diffie-Hellman inversion assumption, sBDHE = selective Bilinear Diffie-Hellman Exponentiation assumption)

	crs size	Hash size	Send size	crs size	Hash size	Send size	Receive	Assumption
Cho *et al.* [10]	$O(1)$	$O(1)$	$O(\log n)$	4.6kB	48 bytes	1.2PB[1]	-	DDH
Goyal *et al.* [27]	$O(n)$	$O(1)$	$O(1)$	103.1GB	48 bytes	1.25kB	8.1 days	q-DBDHI
Goyal *et al.* [27] + §5.1	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(1)$	2.2MB	2.2MB	1.25kB	15.1s	q-DBDHI
Alamati *et al.* [2]	$O(1)$	$O(1)$	$O(1)$	0.8MB	3.9MB	7.7MB	85.9s	ϕ-hiding
This work §4	$O(n)$	$O(1)$	$O(1)$	412.3GB	48 bytes	1.34kB	27.7 minutes	sBDHE
This work §4 + §5.1	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(1)$	8.9MB	2.2MB	1.34kB	38.7ms	sBDHE

that achieve the same asymptotic complexity as the state-of-the-art, but achieves orders of magnitude improvements in decryption times. This brings the area of laconic cryptography to the realm of practice.

A new approach. In this work we investigate an alternative approach to building ℓOT schemes. We begin with a simple observation: namely, that ℓOT has similarities to primitives that have been studied in the literature, most notably efficient constant-size *broadcast encryption* [7, 19]. In broadcast encryption (BE), an encryptor transmits a message to a subset of recipients such that only recipients in the set can decrypt the resulting ciphertext. While this functionality is clearly different from ℓOT, the two systems share a similar structure: each can be viewed as a form of "subset" encryption in which a compact ciphertext can be decrypted by some keys and not others. Of course, broadcast encryption on its own does not obviously imply ℓOT. This motivates the following question: *can efficient broadcast encryption constructions be used as a stepping stone to construct laconic OT?*

In this work we answer the previous question in the affirmative. Our first contribution is to observe that certain pairing-based broadcast encryption schemes can be transformed into a related primitive that we name *set membership encryption* (SME). SME is a form of functional encryption [8, 43] that combines properties of broadcast encryption with those of Identity-Based Encryption (IBE) [6]. In this paradigm, a master authority generates a set of secret keys for a collection of parties. The encryptor now specifies a *single* party to be the recipient. The novel ingredient in this primitive is that the encryption algorithm *additionally* receives a succinct commitment that identifies a specific subset of possible recipients. A party can decrypt the resulting ciphertext if and only if they were identified as both the intended recipient *and* they are included within this commitment.

We show a (selectively secure) construction of set membership encryption based on the work by Boneh et al. [7].[2]

A key property of set membership encryption is that the scheme remains secure *even when all possible recipients collude*. This means that all parties' secret keys can be revealed to an adversary without compromising the security of the scheme, *i.e.* ciphertexts that are intended for a recipient that was not in the subset of recipients remain secure. This allows us to construct a laconic OT for a database of fixed size N via the simple expedient of generating $2N$ parties' public and secret keys in a trusted setup phase, and publishing the resulting key material as a structured reference string (SRS). The Receiver can then commit to its selector bits by encoding these as a commitment for the set membership encryption scheme.

Asymptotic vs Concrete Costs. An important note regarding this approach is that the resulting constructions produce the same efficiency tradeoff as in the work of Goyal *et al.* and Alamati *et al.* In comparison with the original work of Cho *et al.*, asymptotically, our construction reduces the bandwidth complexity of each ciphertext from $O(\log n)$ to constant size, but requires an increase in the size of the structured reference string (SRS) to $O(n)$ (rather than a constant) and a corresponding increase in the decryption complexity of the receiver to $O(n)$ rather than poly $(\log(n))$.

We believe that the longer size of the SRS might be acceptable for some applications since it can be re-used for multiple protocol executions, and was already introduced as useful by Goyal *et al.* We further demonstrate that other tradeoffs are possible, yielding lower decryption times. Specifically, by allowing for a larger digest (sublinear, as opposed to constant-sized), we can achieve sublinear sized SRS and sublinear decryption complexity. Nevertheless, the longer asymptotic decryption complexity of our construction compared to the original work of Cho *et al.* remains a limitation, and further improvements on this front remain an interesting avenue for future work.

Our construction performs surprisingly well when we evaluate the concrete costs. Indeed, the asymptotic complexity discussed above obscures a significant concrete improvement over the work of Cho *et al.*, due to the fact that our construction removes the need for many sequential garbled circuit evaluations, *e.g.* while prior works' ciphertexts easily grow into *petabytes* of data, ours is only a few hundred bytes. The works of Goyal *et al.* and Alamati *et al.* also achieve these better communication complexities. Our key improvement over these works is in the decryption complexity. In particular, the decryption time of our scheme (specifically the variant that achieves sublinear decryption complexity) is orders of magnitude faster than prior schemes.

We refer the reader to Table 1 and §6 for a detailed comparison of our results with prior work.

Receiver Privacy. We note that the basic definition of ℓOT as proposed by Cho *et al.* does not include *Receiver privacy*, i.e. the sender does not learn the receiver's selection bits. In practice, this property is added to a basic protocol using a two-party secure computation protocol instantiated with a garbled circuit. While a similar approach can be used with our constructions as well, we choose to add receiver privacy to the definition. Our construction

[1] This is an estimate based on circuit size for curve multiplications on secp192k1 given by Jayaraman *et al.* [34], for more details on this computation see §6.2.

[2] We also show how to build an adaptively secure scheme in the full version of our paper, but with worse efficiency. We leave it as important future work to find an adaptively secure scheme with better efficiency.

can achieve Receiver privacy without the use of garbled circuits, therefore, it can easily be added to the base construction.

Extensions. Finally, we consider a basic extensions to our schemes. We investigate an extension that was originally proposed by Cho *et al.*, namely, *updatable ℓOT*, i.e. adding the ability to update one selection bit without having to start the full protocol from scratch. We introduce a more general definition of this primitive and show how to realize this definition using our SME constructions. We note that the constructions from Goyal *et al.* [27] and Alamati *et al.* [2] do not seem to imply such generalization.

Concurrent work. During the review process of this paper, two concurrent works emerged, significantly contributing to this exact same problem. The first of these papers, Döttling *et al.*[16], introduces a more efficient approach, albeit based on different underlying assumptions compared to our own.

The second paper, Glaeser *et al.* [23], closely aligns with our construction but employs a distinct approach in achieving it, highlighting the robustness and versatility of our findings.

Our paper, in light of these new developments, stands as a valuable contribution that complements previous work.

1.1 Applications

We observe that SME and the laconic primitives that can be derived from it have several potential applications that motivate its study. Moreover, understanding the nature of these applications is important, as it can help to determine the appropriate efficiency requirements for an SME scheme. In previous work, many different applications have been discussed as well, most of these can be achieved without much trouble from SME or derived primitives. We briefly discuss one direct application below.

One-Time Programs. Goldwasser, Kalai and Rothblum [25] proposed the notion of *one-time programs*. These programs employ a form of secure hardware token, with multiple OT-like functionalities that "self-destruct" after use. In practice, the cost of this token functionality imposes a significant barrier to the deployment of such programs: since each token functionality holds one input label for a garbled circuit (and can be used only once), the input size (or number of program executions) is therefore bounded by the number of functionalities that can be included into a practical device. Laconic OT removes this restriction: by compressing a large selector database into a ℓOT digest, an evaluator can now evaluate a polynomial number of input wires (or program executions) using a constant number of token OT functionalities. Recently, this concept was further explored using commodity hardware by Eldridge *et al.* [17], our laconic OT scheme can be directly applied in this context.

1.2 Technical Overview

In this section we will discuss how to build set membership encryption, but first, we will provide an overview of laconic Oblivious Transfer and discuss the early construction of Cho *et al.* [10]. In the original definition of laconic OT receiver privacy is missing, however, we have added receiver privacy to the definition of laconic OT as well.

Laconic Oblivious Transfer In a traditional OT, a Sender with two messages interacts with a Receiver who possesses a selector bit. At the conclusion of the interaction, the Receiver learns one message (and nothing about the other message) and the Sender does not learn the Receiver's selection. Laconic OT generalizes this primitive to multiple interactions: the Receiver possesses a database $D \in \{0,1\}^n$ of selector bits and the Sender has n message pairs. To rule out the naive construction, laconic OT adds an efficiency restriction: the communication from Receiver to Sender must be compact, and ideally independent of the database size. An alternative view of laconic OT poses it as a type of encryption: the Receiver computes a hash of the selector bits, and the Sender uses this hash as a form of "public key" to encrypt messages for specific indices and positions. Most critically, in this formulation a single digest can be re-used to encrypt many distinct messages.

Security. Traditional OT provides privacy for both Sender and Receiver. The basic ℓOT security definition proposed by Cho *et al.* requires only privacy for the Sender's inputs. Cho *et al.* point out that Receiver privacy can be added by embedding the laconic encryption algorithm into a garbled circuit that can be evaluated by the Receiver using labels retrieved from the sender using a traditional OT protocol. However, given our very different approach, we can add receiver privacy from the start. Therefore, we have added it to the definition of laconic OT.

Efficiency. Cho *et al.* specify precise asymptotic efficiency requirements in their definition of laconic OT. Notably, they limit the size of the digest to be only polynomially-dependent on the security parameter and independent of the database size. Computationally, the digest must be computable in time $n \cdot \text{poly}(\log n)$ and encryption and decryption time are bounded by $\text{poly}(\log n)$. Unfortunately, due to the nature of our constructions we have to relax some of these bounds: specifically, we allow decryption to require $O(n)$, which is an asymptotic setback, but we can still show concrete improvements of the decryption time.

The construction of Cho *et al.* Cho *et al.* propose a construction in two parts. Given a security parameter λ, they propose a laconic OT scheme that takes as input a selector database of size 2λ and creates a digest of size λ. To achieve further compression, the second part of the construction uses this scheme in a binary Merkle tree to reduce a database of arbitrary size to a digest of size λ.

The key observation of the Cho *et al.* scheme is that the root of this tree can be used as an encryption key. This is done by constructing an efficient witness encryption scheme for a specific language involving hash functions, and then dividing the selector database into blocks of λ bits each. These blocks form the leaves of the tree and the root of the tree becomes the digest of the laconic OT scheme. Encryption proceeds by evaluating the witness encryption at each level of the tree. To make this process non-interactive, each level of encryption is embedded into a garbled circuit, resulting in ciphertexts that comprise chains of $\log N$ garbled circuits. While this design is elegant and the overall complexity assumptions are mild, practical implementations must bear the cost of evaluating elliptic curve point multiplications within many garbled circuits in order to decrypt a single laconic ciphertext: in practice this results in a substantial concrete overhead.

Our Approach. In broadcast encryption (BE) [19], an encryptor wishes to broadcast a message such that only a subset of receivers can receive the message. Intuitively one might be tempted to construct laconic OT from broadcast encryption via the following heuristic: each of 2 possible selector bits/position in the Receiver's database could be treated as a single recipient in a universe of $2n$ possible recipients. The Sender could then encrypt each of its messages to a subset of the recipients that would correspond to the appropriate selector bits in the Receiver's database. In this vision, the Receiver would be unable to decrypt messages in positions where its selector bit was not appropriately set.

Of course this intuition does not work, for several reasons. First: broadcast encryption is fundamentally the wrong primitive for this task: instead of encrypting to a specific recipient if and only if the recipient is in a chosen subset, broadcast encryption allows decryption by *any* recipient in the set. In practice this means that the recipient can open both messages at a given index. Moreover even if we ignore this fundamental issue, broadcast encryption has no notion of a succinct *digest* to encode the set of allowed recipients. Finally, the intuition above elides an important detail about the nature of the secret keys: even if keys are generated via a trusted setup procedure (or honestly by the Sender), not every broadcast encryption scheme will retain its security when all secret keys are known to an adversarial Receiver.

A key intuition of this work is that while broadcast encryption is not the right primitive, these problems can be solved by adapting specific broadcast encryption schemes to construct a new protocol that we call *set membership encryption*. This new protocol requires several features: it must incorporate a means to specify the recipient set via a compact commitment (or digest); it must allow the encryptor to encrypt to a specific recipient *as long as* they are in the recipient set; and it must provide a strong collusion resistance property. We now outline the key steps by which we construct this primitive.

Constructing a succinct digest. As a first step, we consider the problem of modifying broadcast encryption to incorporate a succinct digest of the recipient set. Our basic observation for this modification is that certain efficient broadcast encryption schemes [7, 39, 47] feature compact ciphertexts that are independent of the recipient size, and also admit homomorphic operations on the ciphertext. The nature of these protocols enables an efficient process for constructing a *digest* of the recipient set: this is done by introducing (1) a first "hashing" procedure Hash that takes as input a recipient set and outputs a succinct broadcast encryption ciphertext encrypting the identity element, and (2) a subsequent "encryption" operation Encrypt that takes the previous ciphertext and homomorphically embeds a new plaintext. Concretely, we observe that in many pairing-based BE schemes the recipient set is contained in the ciphertext as a product of certain group elements corresponding to the recipients. We can compute that product during the Hash algorithm and finish the creation of the ciphertext during the Encrypt algorithm.

Modifying the encryption functionality. Unlike broadcast encryption, set membership encryption requires two inputs to the encryption function. First: it takes as input a succinct digest of the recipient set that is produced using the approach described immediately above. Additionally it takes a specific receiver identity. The scheme must allow decryption by a specific recipient key *if and only if* the recipient was identified by the encryptor and is in the recipient set. This novel functionality combines elements of broadcast encryption with those of IBE.[3] Because of the mathematical properties of the pairing-based schemes, we can instantiate a second version of the same BE scheme, such that we can bind one recipient to the set of recipients by taking their product. Moreover, we ensure that two secret keys for the same receiver are bound as well, such that the resulting secret key can decrypt both coupled ciphertexts.

Collusion resistance. In the security game of a normal BE scheme, an adversary can receive any key they want as long as they do not appear in the challenge set, *i.e.* the set of receivers to which the challenge ciphertext is encrypted. However, we want the stronger property that given all possible secret keys, ciphertexts are still secure when they are encrypted to a recipient that was not in the recipient set. The way we achieve this is that while tying both the secret keys, we ensure that the adversary cannot use a set of keys to derive a different key. By strongly coupling the binding of both keys to the master secret key, we avoid that any malicious keys can be crafted from a combination of secret keys without knowledge of the master secret key.

Defining security for set membership encryption. We define security for SME via a game-based definition that says that an adversary cannot learn anything from a ciphertext, when that ciphertext is formulated by using the hashing algorithm on a set of receivers and the encryption algorithm for one specific receiver that is not contained in the set, even given decryption keys for all possible receivers. We will define a strong adaptively secure definition, followed by a weaker selectively secure definition where the challenger knows the challenge set of receivers and the challenge receiver before creating a common reference string.

From SME to laconic OT Having defined set membership encryption, this leaves the main question of how to create a laconic OT scheme. Figure 1 illustrates this process. As mentioned previously, the idea is to define two receivers for each location in the database: one receiver corresponds to a 0 bit and the other to a 1 bit. Now, the database can be mapped to said receivers and the Hash algorithm can be used to create a digest $\hat{D}$ of the database. Next, for every location, the sender can create two ciphertexts by encrypting the labels using both receivers at that location corresponding to 0 and 1. Given that the encoded database is used, only one of these receivers is encoded inside $\hat{D}$, therefore, due to the properties of the SME, the receiver can only decrypt one of the labels.

SME Construction We show a construction of set membership encryption that is selectively secure, using this scheme we achieve the most efficient laconic OT scheme.

This construction is based on the broadcast encryption system introduced by Boneh *et al.* [7]. This BE scheme is also only selectively secure, which explains the fact that we can only hope to prove selective security for our set membership encryption scheme. To prove this scheme secure we introduce a new assumption which we call a selective bilinear Diffie-Hellman exponentiation assumption

[3] Notably, this new scheme must use the same key for both functionalities, which rules out simple combinations of two distinct schemes.

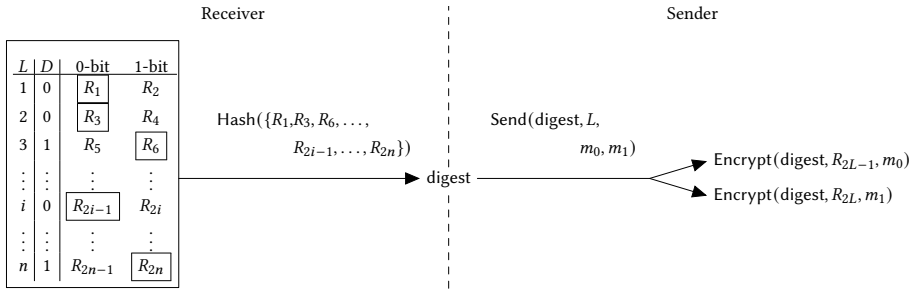

Figure 1: A schematic description of using SME to construct laconic OT based on an example database D. Note that for ease of presentation, we use simplified versions of the algorithms omitting any details such as a common reference string. Each R_i represents a receiver in the SME scheme.

(sBDHE), which is an interactive variant of a general BDHE. We can prove this assumption to be secure in the generic bilinear group model [45], by using the generic proof template of Boneh *et al.* [5]

In the full version of our paper, we show that it is possible to build an adaptively secure scheme using our technique, but crs grows to size $O\left(n^2\right)$. We leave improving the parameters of the adaptively secure scheme for future work.

Improvements of our Laconic OT Construction In the next part of this paper we will introduce a few improvements to our constructions, we will now give a brief technical overview of these improvements.

$\sqrt{n}$-Optimization Although, our construction already shows a concrete improvement over some of the previous work, the common reference string is still particularly large and the construction of Alamati *et al.* is still outperforming ours. By striking a new balance between the size of the digest and the CRS, we can achieve much better efficiency. In order to achieve this, we only initiate the underlying SME with $2\sqrt{n}$ receivers, where n is the size of the database. Instead of hashing the database as a whole, we hash it in chunks of size $\sqrt{n}$. The digest consists of $\sqrt{n}$ sub-digests, which increases the asymptotic communication efficiency of the digest to $O\left(\sqrt{n}\right)$. However, as already shown in Table 1, we achieve much better concrete efficiency overall. This leads to a very practical construction of ℓOT.

2 PRELIMINARIES

Notation. Let λ be an adjustable security parameter and $\mathsf{negl}(n)$ be a negligible function in λ. We use $\overset{c}{\approx}$ to denote computational indistinguishability. We will write $x \leftarrow \mathsf{Algo}(\cdot)$ to say that x is a specific output of running the algorithm Algo on specific inputs. We write Algo^D to say that the algorithm Algo has random read access to the set D. We denote $[n] = \{1, \ldots, n\}$ and for a bit b we write $\bar{b}$ to denote $1 - b$. We will write $x \overset{\$}{\leftarrow} S$ when x gets randomly sampled from the set S, we assume the sampling is uniformly random unless otherwise specified. For ease of presentation, in all asymptotic efficiency notations we will ignore the security parameter and will assume it appears in all of them.

2.1 Laconic Oblivious Transfer

We now give the following formal definition as presented by Cho *et al.*, we add the requirement that the size of the database is known when generating the common reference string, as well as some changes to the efficiency requirements.

Definition 2.1 (laconic OT). A laconic OT (ℓOT) scheme consists of four algorithms crsGen, Hash, Send, and Receive.

- $\mathsf{crsGen}(1^\lambda, \ell) \to \mathsf{crs}$. This algorithm takes as input the security parameter λ and the size of the database ℓ.
- $\mathsf{Hash}(\mathsf{crs}, D) \to (\mathsf{digest}, \hat{D})$. This algorithm takes as input a common reference string crs and a database $D \in \{0,1\}^\ell$ and outputs a digest digest of the database and a state $\hat{D}$.
- $\mathsf{Send}(\mathsf{crs}, \mathsf{digest}, L, m_0, m_1) \to c$. This algorithm takes as input a common reference string crs, a digest digest, a database location $L \in [\ell]$, and two labels m_0 and m_1. It outputs a ciphertext c.
- $\mathsf{Receive}^{\hat{D}}(\mathsf{crs}, c, L) \to m$. This algorithm takes as input a common reference string crs, a ciphertext c, and a database position L. Moreover, it has random read access to $\hat{D}$. It outputs a label m.

This scheme should have the following properties:

Correctness: For any database D of size $\ell = \mathsf{poly}(\lambda)$, for any polynomial function $\mathsf{poly}(\cdot)$, any database location $L \in [\ell]$, and any pair of labels $(m_0, m_1) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$, it holds that

$$\Pr\left[m = m_{D[L]} \;\middle|\; \begin{array}{ccc} \mathsf{crs} & \leftarrow & \mathsf{crsGen}\left(1^\lambda, \ell\right) \\ \left(\mathsf{digest}, \hat{D}\right) & \leftarrow & \mathsf{Hash}\left(\mathsf{crs}, D\right) \\ c & \leftarrow & \mathsf{Send}\left(\mathsf{crs}, \mathsf{digest}, L, m_0, m_1\right) \\ m & \leftarrow & \mathsf{Receive}^{\hat{D}}\left(\mathsf{crs}, c, L\right) \end{array}\right] = 1,$$

where the probability is taken over the random choices made by crsGen and Send.

Sender Privacy Against Semi-Honest Receivers: There exists a PPT simulator $\mathcal{S}$ such that for any database of size at most $\ell = \mathsf{poly}(\lambda)$ for any polynomial function $\mathsf{poly}(\cdot)$, any memory location $L \in [\ell]$, and any pair of labels $(m_0, m_1) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$, let $\mathsf{crs} \leftarrow \mathsf{crsGen}(1^\lambda, \ell)$ and $\mathsf{digest} \leftarrow \mathsf{Hash}(\mathsf{crs}, D)$, it holds that

$$\left(\mathsf{crs}, \mathsf{Send}\left(\mathsf{crs}, \mathsf{digest}, L, m_0, m_1\right)\right) \overset{c}{\approx} \left(\mathsf{crs}, \mathcal{S}\left(D, L, m_{D[L]}\right)\right).$$

Receiver Privacy: There exists a PPT simulator $\mathcal{S}$ such that for any database of size at most $\ell = \mathsf{poly}(\lambda)$ for any polynomial

function poly $(\cdot)$, let crs $\leftarrow$ crsGen$(1^\lambda, \ell)$, it holds that

$$(\text{crs}, \text{Hash}(\text{crs}, D)) \overset{c}{\approx} \left(\text{crs}, \mathcal{S}\left(1^\lambda\right)\right).$$

Efficiency: The length of digest is a fixed polynomial in λ, independent of the size of the database. Moreover, the algorithm Hash runs in time $|D| \cdot \text{poly}(\log|D|, \lambda)$, Send runs in time poly $(\log|D|, \lambda)$, and Receive runs in time $O(|D|, \lambda)$.[4]

3 SET MEMBERSHIP ENCRYPTION

In this section we introduce a new primitive called *set membership encryption* (SME). This primitive allows a first party, the hasher, to hash a subset of all receivers that can decrypt a ciphertext, but only a second party, the encryptor, adds a message to the ciphertext and defines a single recipient. Only when this recipient was included in the subset that was chosen by the hasher, the ciphertext can be decrypted. We give a game-based definition that specifies strong adaptive security as well as a second weaker selectively secure definition.

Definition 3.1 (set membership encryption). A set membership encryption scheme (SME) consists of five randomized algorithms:

- Setup$(1^\lambda, n) \to (\text{pk}, \text{msk})$. This algorithm takes as input the security parameter λ and the maximum number of receivers n. It outputs a public key pk and a master secret key msk.
- KeyGen$(k, \text{pk}, \text{msk}) \to K_k$. This algorithm takes as input a receiver $k \in [n]$, a public key pk, and a master secret key msk. It outputs a private key K_k.
- Hash$(\text{pk}, S) \to \left(\hat{S}, \text{st}\right)$. This algorithm takes as input a public key pk and a subset $S \subseteq [n]$. It outputs a digest of the set S that is denoted $\hat{S}$ and some state st.
- Encrypt$(\text{pk}, i, M, \hat{S}) \to C$. This algorithm takes as input a public key pk, a receiver $i \in [n]$, a message M, and a digest $\hat{S}$. It outputs a ciphertext C.
- Decrypt$(\text{pk}, K, S, i, C, \text{st}) \to M$. This algorithm takes as input a public key pk, a private key K, a subset $S \subseteq [n]$, a receiver $i \in [n]$, a ciphertext C, and state from Hash st. It outputs a plaintext message M.

This scheme should have the following properties:

Correctness. For all $S \subseteq [n]$ and all $i \in S$, we have

$$\Pr\left[M = \text{Decrypt}(\text{pk}, K, S, i, C, \text{st}) \;\middle|\; \begin{array}{ll} (\text{pk}, \text{msk}) & \leftarrow \text{Setup}(1^\lambda, n) \\ (\hat{S}, \text{st}) & \leftarrow \text{Hash}(\text{pk}, S) \\ C & \leftarrow \text{Encrypt}(\text{pk}, i, M, \hat{S}) \\ K & \leftarrow \text{KeyGen}(i, \text{pk}, \text{msk}) \end{array}\right] = 1.$$

Efficiency. The size of the digest $\hat{S}$ is a fixed polynomial in λ independent of the size of the original set. Moreover, the algorithm Hash runs in time $|D| \cdot \text{poly}(\log|D|, \lambda)$, Encrypt runs in time poly $(\log|D|, \lambda)$, and Decrypt runs in time poly $(|D|, \lambda)$.

(Selective) Security. For all PPT algorithms $\mathcal{A}$ we have that

$$\left|\Pr\left[b = b' \middle| (b, b') \leftarrow \textbf{Game}_{\mathcal{A},\text{SME},n}(\lambda)\right] - \frac{1}{2}\right| \leq \epsilon(\lambda),$$

with $\epsilon(\cdot)$ a negligible function and $\textbf{Game}_{\mathcal{A},\text{SME},n}$ the security game as described in Figure 2. The same definition can be stated with the selective security game $\textbf{Game}_{\mathcal{A},s\text{SME},n}(\lambda)$ as described in Figure 3.

Privacy: There exists a PPT simulator $\mathcal{S}$ such that for any set S of size at most $n = \text{poly}(\lambda)$ for any polynomial function poly $(\cdot)$, let $(\text{pk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, n)$, $\forall i \in [n] : K_i = \text{KeyGen}(i, \text{pk}, \text{msk})$, and $(\hat{S}, \text{st}) \leftarrow \text{Hash}(\text{pk}, S)$, it holds that

$$\left(\text{pk}, \{K_i\}_{i\in[n]}, \hat{S}\right) \overset{c}{\approx} \left(\text{pk}, \{K_i\}_{i\in[n]}, \mathcal{S}\left(1^\lambda\right)\right).$$

Security game for set membership encryption $\textbf{Game}_{\mathcal{A},\text{SME},n}(\lambda)$.

Setup. The challenger runs Setup $\left(1^\lambda, n\right)$ to obtain a public key pk and master secret key msk, it hands the public key to the adversary $\mathcal{A}$ as well as private keys $K_k \leftarrow \text{KeyGen}(k, \text{pk}, \text{msk})$, for all $k \in [n]$.

Challenge. $\mathcal{A}$ picks a message M and sends this message together with a subset $S \subseteq [n]$ and a receiver $i \notin S$. The challenger computes a digest $(\hat{S}, \text{st}) \leftarrow \text{Hash}(\text{pk}, S)$ and picks $b \overset{\$}{\leftarrow} \{0, 1\}$ and sets $C \leftarrow \text{Encrypt}\left(\text{pk}, i, M, \hat{S}\right)$ if $b = 0$ and picks M' at random and sets $C \leftarrow \text{Encrypt}\left(\text{pk}, i, M', \hat{S}\right)$ otherwise. The challenger gives this digest, the state st, and the ciphertext to $\mathcal{A}$.

Guess. The adversary $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$ for b and wins the game if $b = b'$.

Figure 2: Security game for set membership encryption.

3.1 Laconic OT from Set Membership Encryption

We now prove that our newly introduced set membership encryption primitive implies ℓOT, by constructing a ℓOT scheme based on a secure SME scheme.

Theorem 1. *Given a selectively secure set membership encryption scheme*

$$\mathit{SME} = (\text{Setup}, \text{KeyGen}, \text{Hash}, \text{Encrypt}, \text{Decrypt})$$

there exists a secure laconic OT scheme ℓOT.

Proof. We build the following laconic OT scheme ℓOT, with $\ell = |D|$

crsGen$(1^\lambda, \ell)$: Set $n = 2\ell$, run $(\text{pk}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, n)$, and run $K_i \leftarrow \text{KeyGen}(i, \text{pk}, \text{msk})$, for all $i \in [n]$. Output crs $= (\text{pk}, \{K_i\}_{i\in[n]})$.

Hash(crs, D) : Set $E = \left\{2i - \overline{D[i]} \middle| \forall i \in [\ell]\right\}$ and

$$(\text{digest}, \text{st}) \leftarrow \text{Hash}(\text{pk}, E).$$

Output $\left(\text{digest}, \hat{D} = (E, \text{digest}, \text{st})\right)$.

[4] We slightly relax the efficiency in comparison with the definition given in Cho et al. [10] This slightly worse asymptotic receiver time results in constructions with much better concrete efficiency.

Selective security game for set membership encryption $\mathbf{Game}_{\mathcal{A},s\mathsf{SME},n}(\lambda)$.

Setup. The adversary $\mathcal{A}$ sends a set $S \subseteq [n]$ and a receiver $i \notin S$. The challenger runs Setup $\left(1^\lambda, n\right)$ to obtain a public key pk and master secret key msk, it hands out the public key to the adversary $\mathcal{A}$ as well as private keys $K_k \leftarrow \mathsf{KeyGen}(k, \mathsf{pk}, \mathsf{msk})$, for all $k \in [n]$.

Challenge. $\mathcal{A}$ picks a message M and sends this message. The challenger computes a digest $(\hat{S}, \mathsf{st}) \leftarrow \mathsf{Hash}(\mathsf{pk}, S)$ and picks $b \xleftarrow{\$} \{0,1\}$ and sets $C \leftarrow \mathsf{Encrypt}\left(\mathsf{pk}, i, M, \hat{S}\right)$ if $b = 0$ and picks M' at random and sets $C \leftarrow \mathsf{Encrypt}\left(\mathsf{pk}, i, M', \hat{S}\right)$ otherwise. The challenger gives this ciphertext, the digest, and the state st to $\mathcal{A}$.

Guess. The adversary $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ for b and wins the game if $b = b'$.

Figure 3: Selective security game for set membership encryption.

$\mathsf{Send}(\mathsf{crs}, \mathsf{digest}, L, m_0, m_1)$: create

$$c_0 = \mathsf{Encrypt}(\mathsf{pk}, 2L-1, m_0, \mathsf{digest})$$

and $c_1 = \mathsf{Encrypt}(\mathsf{pk}, 2L, m_1, \mathsf{digest})$. Output $c = (c_0, c_1)$.

$\mathsf{Receive}^{\hat{D}}(\mathsf{crs}, c, L)$: Parse c as (c_0, c_1). If $D[L] = 0$, set

$$m = \mathsf{Decrypt}(\mathsf{pk}, K_{2L-1}, E, 2L-1, c_0, \mathsf{st}).$$

If $D[L] = 1$, set

$$m = \mathsf{Decrypt}(\mathsf{pk}, K_{2L}, E, 2L, c_1, \mathsf{st}).$$

Correctness follows by inspection and because of the correctness of the underlying SME scheme.

To achieve receiver privacy, the simulator can internally use the simulator of the privacy property of the underlying SME scheme. Indistinguishability of both views follows immediately.

To show sender privacy, we construct the following simulator $\mathcal{S}(D, L, m_{D[L]})$:

- digest $\leftarrow \mathsf{Hash}(\mathsf{crs}, D)$
- $c_0 = \mathsf{Encrypt}(\mathsf{pk}, 2L-1, m_{D[L]}, \mathsf{digest})$ and $c_1 = \mathsf{Encrypt}(\mathsf{pk}, 2L, m_{D[L]}, \mathsf{digest})$ and output $c = (c_0, c_1)$.

Assume we have a distinguisher $\mathcal{A}$ that can distinguish between the normal Send algorithm and this simulator, then we build an adversary $\mathcal{B}$ that can break the SME security game. $\mathcal{B}$ receives a public key pk and all private keys for every $i \in [n]$ from the challenger in the SME security game. It passes this information as the crs to $\mathcal{A}$. Next, $\mathcal{B}$ computes digest honestly using Hash, picks a message $m_{\overline{D[L]}}$ at random, and sends $m_{D[L]}$ and $m_{\overline{D[L]}}$ as well as E (from Hash), $2L - \overline{D[L]}$, and digest to the challenger in the SME security game. $\mathcal{B}$ receives a ciphertext c. Now, $\mathcal{B}$ sets $c_{\overline{D[L]}} = c$ and $c_{D[L]} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, 2L - \overline{D[L]}, m_{D[L]}, \mathsf{digest})$. Now, $\mathcal{A}$ answers with either 0, *i.e.* we are using the real Send algorithm, in which case $\mathcal{B}$ answers with $D[L]$, or $\mathcal{A}$ answers with 1, *i.e.* we are using the simulator. In which case $\mathcal{B}$ sends $\overline{D[L]}$ to the challenger.

The efficiency of the laconic OT scheme follows directly from the efficiency of the set membership encryption scheme. □

Malicious Sender and Receiver. Similar to previous work, the constructions that we give are only secure against semi-honest adversaries. Standard techniques can be used to upgrade to malicious adversaries, *e.g.* by the use of Non-Interactive Zero Knowledge proofs (NIZKs) [18]. Luckily, given the use of bilinear maps in our constructions it is possible to use quite optimal NIZKs such as the ones introduced by Groth and Sahai [29], Groth [28], and Lai *et al.* [37]. We suspect that these proofs can even be further optimized leading to a very practical construction, we leave these optimizations of the NIZKs for future work.

Other possibilities to improve the construction against malicious adversaries are the techniques of Ishai *et al.* [32], or using interactive zero-knowledge proofs, but this introduces more interactivity. Given the optimal setting for NIZKs, we rather prefer these former techniques.

4 SME CONSTRUCTION

We show a selectively secure construction of the set membership encryption protocol. In the full version of our paper, we introduce an adaptively secure scheme, but decryption keys grow to $O(n)$ in the process. We leave it to future work to find an adaptively secure scheme with constant sized decryption keys. This selectively secure scheme is based upon the broadcast encryption scheme by Boneh, Gentry, and Waters [7]. The original scheme was selectively CPA secure, therefore intuitively, we can only hope to distill a selectively secure SME from this construction.

For ease of presentation we are showing our construction with type-I pairings, but our constructions can easily be adapted to type-III pairings.

$\mathsf{Setup}\left(1^\lambda, n\right)$: Let $\mathbb{G}$ be a bilinear group of prime order p. Choose random generator $g \in \mathbb{G}$ and random $\alpha, \{\beta_i\}_{i\in[n]} \in \mathbb{Z}_p$. Compute $g_i = g^{(\alpha^i)} \in \mathbb{G}$ for $i \in [2n] \setminus \{n+1\}$, and $g^{\beta_i} \in \mathbb{G}$ for $i \in [n]$. Also, choose $\gamma \in \mathbb{Z}_p$ and set $v = g^\gamma \in \mathbb{G}$. Output $\mathsf{pk} = \left(g, \{g_i\}_{i\in[2n]\setminus\{n+1\}}, \left\{g^{\beta_i}\right\}_{i\in[n]}, v\right)$, and $\mathsf{msk} = \left(\gamma, \{\beta_i\}_{i\in[n]}\right)$

$\mathsf{KeyGen}(k, \mathsf{pk}, \mathsf{msk})$: This algorithm takes as input a receiver $k \in [n]$, a public key pk, and a master secret key msk. Parse pk as $\left(g, \{g_i\}_{i\in[2n]\setminus\{n+1\}}, \left\{g^{\beta_i}\right\}_{i\in[n]}, v\right)$ and msk as $\left(\gamma, \{\beta_i\}_{i\in[n]}\right)$. Output $d_k = g_k^\gamma g^{\beta_k}$.

$\mathsf{Hash}(\mathsf{pk}, S \subseteq [n])$: This algorithm takes as input a public key pk and a subset $S \subseteq [n]$. It outputs a digest $\hat{S}$ and a state st. Parse pk as

$$\left(g, \{g_i\}_{i\in[2n]\setminus\{n+1\}}, \left\{g^{\beta_i}\right\}_{i\in[n]}, v\right).$$

Pick uniformly random $z \xleftarrow{\$} \mathbb{Z}_p$ and compute

$$\hat{S} = g^z v \prod_{j \in S} g_{n+1-j}.$$

Output $(\hat{S}, \mathsf{st} = z)$

Encrypt $\left(\mathsf{pk}, i, M, \hat{S}\right)$: This algorithm takes as input a public key pk, a receiver $i \in [n]$, a message M, and a digest $\hat{S}$. It outputs a ciphertext $C^* = (\mathsf{Hdr}, C)$. Parse pk as $(g, \{g_i\}_{i \in [2n] \setminus \{n+1\}}, \left\{g^{\beta_i}\right\}_{i \in [n]}, v)$. Pick $t \xleftarrow{\$} \mathbb{Z}_p$ and set

$$\mathsf{Hdr} = \left(g^t, \left(g^{\beta_i} \hat{S}\right)^t\right) = \left(g^t, \left(g^z v g^{\beta_i} \prod_{j \in S} g_{n+1-j}\right)^t\right).$$

Compute $K = e\left(g, g_{n+1}\right)^t$, note that you can compute

$$e\left(g, g_{n+1}\right) = e\left(g_1, g_n\right).$$

Set $C = M \cdot K$ and output $C^* = (\mathsf{Hdr}, C)$.

Decrypt $(\mathsf{pk}, d_i, S, i, C, \mathsf{st})$:

$$K = \frac{e\left(g_i, c_2\right)}{e\left(d_i \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}, c_1\right)}. \tag{1}$$

Output $M \leftarrow C \cdot K^{-1}$.

Correctness We show that Equation (1) indeed recovers K, which then can be used to recover the plaintext message M

$$\frac{e\left(g_i, c_2\right)}{e\left(g_i^z d_i \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}, c_1\right)} = \frac{e\left(g^{(\alpha^i)}, \left(g^z v g^{\beta_i} \prod_{j \in S} g_{n+1-j}\right)^t\right)}{e\left(g_i^z \left(g^{(\alpha^i)}\right)^\gamma \left(g^{(\alpha^i)}\right)^{\beta_i} \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}, g^t\right)}$$

$$= \frac{e\left(g^{(\alpha^i)}, g_{n+1-i}\right)^t e\left(g^{(\alpha^i)}, g^z v \prod_{j \in S} g_{n+1-j}\right)^t e\left(g^{(\alpha^i)}, g^{\beta_i}\right)^t}{e\left(g_i^z \left(g^{(\alpha^i)}\right)^\gamma \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}, g^t\right) e\left(\left(g^{(\alpha^i)}\right)^{\beta_i}, g^t\right)}$$

$$= e\left(g, g_{n+1}\right)^t \frac{e\left(g, g_i^z v^{(\alpha^i)} \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}\right)^t}{e\left(g_i^z \left(g^{(\alpha^i)}\right)^\gamma \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}, g\right)^t} = e\left(g, g_{n+1}\right)^t = K$$

Efficiency In this construction it can be seen that the size of the public key is $3n + 1$ elements in $\mathbb{G}$. The msk is $n + 1$ elements in $\mathbb{Z}_p$. The size of the secret keys is just 1 group element.

Hash can be computed by doing $|S|$ multiplications within $\mathbb{G}$, therefore, this algorithm runs in time $O\left(|S|\right)$. The output of the algorithm is just 1 group element. On the other hand, Encrypt runs in constant time, because it only computes 1 multiplication and 2 exponentiations in $\mathbb{G}$, and 1 pairing, 1 exponentiation, and 1 multiplication in $\mathbb{G}_T$. The output of this algorithm is 2 elements in $\mathbb{G}$ and 1 element in $\mathbb{G}_T$.

Finally decryption runs in $O\left(|S|\right)$ because it needs to do $|S| - 1$ multiplications, 2 pairings, and 1 multiplication and 1 division in $\mathbb{G}_T$.

Privacy To show privacy we need to create a simulator such that for any set S of size at most $n = \mathrm{poly}\left(\lambda\right)$ for any polynomial function poly $(\cdot)$, let $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, n)$, $\forall i \in [n] : K_i = \mathsf{KeyGen}(i, \mathsf{pk}, \mathsf{msk})$, and $(\hat{S}, \mathsf{st}) \leftarrow \mathsf{Hash}(\mathsf{pk}, S)$, it holds that

$$\left(\mathsf{pk}, \{K_i\}_{i \in [n]}, \hat{S}\right) \overset{c}{\approx} \left(\mathsf{pk}, \{K_i\}_{i \in [n]}, \mathcal{S}\left(1^\lambda\right)\right).$$

A very straight forward simulator is just sampling a uniform random element in $\mathbb{G}$, because of the randomly chosen blinding factor z during hashing, it is clear that both outputs are perfectly indistinguishable.

Security We introduce a selective BDHE assumption. Although this is a new assumption, we formulate it as a variation of a GBDHE, except that some of the parameters can be chosen adversarially in advance. This approach fits within a framework as used in many papers using GBDHE-style assumptions. Therefore, similar to these previous works, we can prove our assumption to be generically secure (*i.e.* in the generic bilinear group model [45]), by using the generic proof template of Boneh, Boyen, and Goh [5].

Assumption 1. *After given $S \subset [n]$ and $i \in [n]$, with $i \notin S$ by an adversary, a challenger outputs*

$$\left(g, h = g^t, \left\{g_i = g^{(\alpha^i)}\right\}_{i \in [2n] \setminus \{n+1\}}, \left\{g^{\beta_i}\right\}_{i \in [n]}, \right.$$
$$\left. v = g^\gamma, \left\{g_i^\gamma g_i^{\beta_i}\right\}_{i \in [n]}, \left(v g^{\beta_i} \prod_{j \in S} g_{n+1-j}\right)^t\right),$$

and a value T that is either $e\left(h, g_{n+1}\right)$ or a random element in $\mathbb{G}_T$. The assumption states that the adversary has negligible advantage in distinguishing between the two possibilities of T.

We introduce the following lemma to prove above assumption to be secure in the generic group model. We use notation as introduced by Boneh, Boyen, and Goh [5].

Lemma 4.1. *Given $S \subset [n]$ and $i \in [n]$, the above assumption is a selective BDHE assumption, ie: for the following polynomials, f is independent of (P, Q), if $i \notin S$.*

$$P = \left(1, t, \left\{\alpha^k\right\}_{k \in [2n] \setminus \{n+1\}}, \{\beta_k\}_{k \in [n]}, \gamma, \left\{\gamma \alpha^k + \beta_k \alpha^k\right\}_{k \in [n]}, \right.$$
$$\left. \kappa, \gamma t + \beta_i t + \kappa t + t \sum_{j \in S} \alpha^{n+1-j}\right)$$

$$Q = (1)$$

$$f = t\alpha^{n+1}$$

Proof. To show that f is independent of (P, Q) we have to show that f cannot be constructed in the following way: $f = \sum_i \sum_j p_i p_j + \sum_k q_k$, where p_i, p_j are polynomials in P and q_k polynomials in Q.

To create the term $t\alpha^{n+1}$, we need to look at the term $\gamma t + \beta_i t + \kappa t + t \sum_{j \in S} \alpha^{n+1-j}$, because none of the other terms contain α^{n+1}, or any α^k in combination with t. To achieve the term we multiply this with α^k for $k \in S$. This results in $\gamma t \alpha^k + \beta_i t \alpha^k + \kappa t + t \sum_{j \in S} \alpha^{n+1-j+k}$. Now, to cancel out the term $\beta_i t \alpha^k$, we note that $k \neq i$ because $i \notin S$, therefore we cannot hope to use any of the $\gamma \alpha^k + \beta_k \alpha^k$. The only option to try and cancel out that term is by using $\gamma t + \beta_i t + \kappa t + t \sum_{j \in S} \alpha^{n+1-j}$ again if $k = n + 1 - j$ for some $j \in S$ and multiply it with β_i. However, now we introduced the term $\beta_i^2 t$, the only way

this term can be created is in the same way we did, which will lead us in circles. This concludes the proof. □

Theorem 2. *If Assumption 1 holds, then our SME construction above is a selectively secure SME.*

Proof. Given an adversary $\mathcal{A}$ for our SME construction, we build an adversary $\mathcal{B}$ to break Assumption 1. First $\mathcal{B}$ receives a set $S \subset [n]$ and an index $i \in [n]$, but $i \notin S$. It forwards these elements to the challenger of the assumption. $\mathcal{B}$ receives

$$\left(g, h = g^t, \left\{g_i = g^{(\alpha^i)}\right\}_{i\in[2n]\setminus\{n+1\}}, \left\{g^{\beta_i}\right\}_{i\in[n]}, \right.$$
$$\left. v = g^\gamma, \left\{g_i^\gamma g_i^{\beta_i}\right\}_{i\in[n]}, g^z, \left(g^z v g^{\beta_i} \prod_{j\in S} g_{n+1-j}\right)^t\right)$$

from the challenger. It sends

$$pk = \left(g, \{g_i\}_{i\in[2n]\setminus\{n+1\}}, \left\{g^{\beta_i}\right\}_{i\in[n]}, v\right)$$

to the adversary $\mathcal{A}$. $\mathcal{A}$ will respond with a message M. Now, $\mathcal{B}$ can respond to $\mathcal{A}$ with a correctly constructed digest

$$\hat{S} = g^z v \prod_{j\in S} g_{n+1-j}, \mathsf{Hdr} = \left(h, \left(g^z v g^{\beta_i} \prod_{j\in S} g_{n+1-j}\right)^t\right),$$

and $C = M \cdot T$ which is a correctly formed ciphertext that encrypts M when $T = e(h, g_{n+1})$, and an encryption of a random message when T is random. If $\mathcal{A}$ says this is an encryption of the message M then $\mathcal{B}$ responds to the challenger that $T = e(h, g_{n+1})$. On the other hand, if $\mathcal{A}$ says this is an encryption of a random message, then $\mathcal{B}$ responds to the challenger by saying T is random. If $\mathcal{A}$ has a non-negligible advantage in breaking this SME, then $\mathcal{B}$ has a non-negligible advantage in breaking Assumption 1. □

5 EXTENSIONS AND APPLICATIONS

In this section we introduce several extensions and improvements upon our construction of laconic OT, as well as a specific application for our new primitive SME. First we show how to decrease the CRS size by increasing the size of the digest. Next, we present how to get updatable laconic OT for our schemes, which is an extension of normal laconic OT that was introduced by Cho *et al.* [10] and that is important for the applications presented in their paper.

5.1 Optimization of the Laconic OT Construction

Given the fact that the CRS in our laconic OT construction is still linear in the size of the database, we introduce the following optimization by striking a new balance between the sizes of the different components. We decrease the size of the CRS by increasing the size of the digest. By doing this we break the efficiency definition of laconic OT that states that the digest can only depend on the security parameter, however, as we show in Section 6.2, we get a much more practical result by doing so.

We start with the laconic OT construction based on an SME scheme as shown in Section 3.1, remember that we set

$$E = \left\{2i - \overline{D[i]} \middle| \forall i \in [|D|]\right\},$$

with D being the original database. Now we initiate the SME with size $2\sqrt{|D|}$, instead of $n = |E|$. We distribute all positions $L \in D$ to their respective bucket of size $2\sqrt{|D|}$. We can do this by computing $y = \left\lfloor \frac{i}{\sqrt{|D|}} \right\rfloor$, next we compute

$$\forall y, E_y = \left\{x - y\sqrt{n} \mid x \in E \cap \left]y\sqrt{n}, (y+1)\sqrt{n}\right]\right\}.$$

The size of the CRS clearly decreases to $O\left(\sqrt{|D|}\right)$, however, the digest will grow to size $O\left(\sqrt{|D|}\right)$.

5.2 Updatable Laconic OT

For the applications in the original paper by Cho *et al.* [10] we need a slightly different version of laconic OT which the authors of [10] called *updatable laconic OT*. They introduced two more algorithms SendWrite and ReceiveWrite, and required some form of sender privacy where the receiver would not learn the original digest. This definition was very tailored to their specific construction and application. Instead we propose a new version of updatable laconic OT by introducing an algorithm called UpdateDigest, an algorithm that the sender can run on their own, followed by sending the necessary information to the receiver, sender privacy can then be achieved by performing this operation in a garbled circuit.

Definition 5.1 (updatable laconic OT). An updatable laconic OT scheme exists of the algorithms (crsGen, Hash, Send, Receive) as defined in Definition 2.1, additionally the algorithm UpdateDigest is added with the following syntax.

UpdateDigest(crs, digest, L, b) $\to$ digest*. It takes as input a common reference string crs, a digest digest, a location $L \in \mathbb{N}$, a bit $b \in \{0,1\}$ to be written. It outputs a new digest digest*.

On top of the properties of the normal laconic OT scheme we require the following properties:

Correctness with regards to writes: For any database of size $\ell = \text{poly}(\lambda)$ for any polynomial function poly$(\cdot)$, any memory location L $in[\ell]$, any bit $b \in \{0,1\}$ the following holds. Let D^* be identical to D except that $D^*[L] = b$,

$$\Pr\left[\mathsf{digest}^* = \mathsf{digest}' \middle| \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{crsGen}(1^\lambda, \ell) \\ (\mathsf{digest}, \hat{D}) \leftarrow \mathsf{Hash}(\mathsf{crs}, D) \\ (\mathsf{digest}^*, \hat{D}^*) \leftarrow \mathsf{Hash}(\mathsf{crs}, D^*) \\ \mathsf{digest}' \leftarrow \mathsf{UpdateDigest}(\mathsf{crs}, \mathsf{digest}, L, b) \end{array}\right] = 1,$$

where the probability is taken over the randomness of crsGen and UpdateDigest.

Note that the definition is quite similar to the definition in the original paper with respect to correctness, as described above, we leave sender privacy to the addition of a garbled circuit. Moreover, this algorithm outputs the updated digest in a normal representation while Cho *et al.* represent it by using a label corresponding to every bit, however, this is very tailored to the applications and setting they were working in, we believe there is major benefit in defining this more generally the way we do.

Updatable Laconic OT Construction Based on SME of Section 4 Similarly, we show the construction of this new algorithm UpdateDigest in the context of the laconic OT construction that is derived from the SME scheme in Section 4.

We define UpdateDigest as follows:

$$\mathsf{UpdateDigest}(\mathsf{crs}, \mathsf{digest}, L, b) : \text{Output } \frac{\mathsf{digest} \cdot g_{n+1-2L+b}}{g_{n+1-2L+b}}.$$

Note on Achieving Cho *et al.*'s Applications When running the above instantiations of updatable laconic OT inside a garbled circuit we have all the same components as presented in Cho *et al.*, therefore, we can use our version of ℓOT in their applications. However, in terms of efficiency, we have to look at the increased CRS size in our construction in comparison with the construction of Cho *et al.* We note that in the application, the CRS is hard coded inside a garbled circuit, leading to a large garbled circuit, but given our much more efficient overall construction, this is still better than using Cho *et al.*'s construction. The other difference we have to take into account is receiver time, but given that this algorithm is running outside of garbled circuits, we refer the reader to our comparison in Section 6 to note that the overall efficiency of our scheme will easily outweigh Cho *et al.*'s construction. One downside remains the trusted setup that our construction needs, which Cho *et al.* does not.

6 EVALUATION AND COMPARISON

In this section we will evaluate our laconic OT scheme as derived from the SME construction in Section 4. How to derive a laconic OT scheme can be found in Section 3.1. More specifically we will look at the concrete efficiency and compare this with previous work. We recall that we compare the derived laconic OT schemes instead of another primitive for ease of presentation and such that we can compare with the initial work of Cho *et al.* [10].

A difficulty in comparing with previous work is the plethora of papers that build upon the initial work, improving the construction step by step, but usually focusing on the underlying techniques and not necessarily laconic OT itself. Understanding how these all fit together is no easy task. Therefore, when we talk about Cho *et al.*, we actually mean a combination of different papers. The basic construction is taken from the work by Cho *et al.*, but quite immediately after that paper, some improvements were made by Dottling and Garg [12]. The last improvement on which we base ourselves was made by Cong *et al.* [11], which in turn based their improvements on Goyal and Vusirikala [26].

Furthermore, we compare with the work from Goyal *et al.* [27] that introduces constructions for One-Way Functions with Encryption (OWFE). It is easy to see that OWFE implies laconic OT when the one-way function has the necessary compression to act as the hash function in the laconic OT scheme. Goyal *et al.* have such construction and we compare with their most efficient construction based on the q-DBDHI assumption.

Finally, we also compare with the work of Alamati *et al.* [2] that shows a construction for laconic OT from the ϕ-hiding assumption based on another OWFE construction by Goyal *et al.* [27]. Although, this construction seems to have great asymptotic efficiency, because of the ϕ-hiding assumption the construction happens in a rather large RSA group leading to concrete inefficiencies.

6.1 Asymptotic Efficiency

First we compare the asymptotic efficiency between previous work and this work. We compare the laconic OT scheme that is constructed from the SME constructions in Section 4, and with the optimization described in Section 5.1. In Table 2 we show the computational and communication efficiency of the different algorithms. In terms of computation time, we make a $\log n$ improvement for both Hash and Send in comparison with Cho *et al.*, but we go from polylogarithmic to linear decryption time.

In terms of communication complexity, the common reference string in our scheme is much larger compared to Cho *et al.* Nevertheless, the size of the encryption generated by Send decreases from logarithmic to constant size.

All our asymptotic efficiencies are similar to the ones in Goyal *et al.*, but Alamati *et al.* achieve better asymptotic efficiency by reducing the common reference string to constant size, hence, achieving overall better efficiency than Cho *et al.* with the exception of receiving time.

6.2 Concrete Efficiency

Now we will look into concrete efficiency of our scheme in comparison with previous work.

To the best of our knowledge, Cong *et al.* [11] are the only ones to make an estimate of the Send size in Cho *et al.* [10], they estimate that size to be 11TB, this is based on *one* curve multiplication at every level in their tree based construction.[5] However, when looking closely at the circuits that get garbled at every stage of the tree, they contain $\sim 2\lambda$ curve multiplications, leading to a much bigger circuit. Because in the work of Cong *et al.* they use a database of size 2^{31}, we will be using the same number throughout our comparisons.

Computing Efficiency for Cho *et al.* To compute the concrete efficiency of Cho *et al.* we will use secp192k1 [42], because this seems to be the only curve for which someone actually computed the number of gates a circuit would contain when doing a curve multiplication [34]. Because this curve only has 96 bits of security, we do not hope to achieve any better security, and will use this security parameter for all other computations as well.

Moreover, we take the 30% optimization of Cong *et al.* into account as well, leading to a size of 1.2 petabytes. We compute the size using the following formula:

$$\text{size} = ((2\lambda \times d + 1) \times 19\,200\,000\,000 \times 160 \text{ bits}) \times 0.7,$$

where λ is the security parameter, d is the number of levels in the tree, the number 19.2 billion corresponds to the amount of non-XOR gates for doing a curve multiplication [34], and finally, by using half gates [49] for each non-XOR gate [36] we need two ciphertexts for each non-XOR gate which we will assume are 80 bits each. We also use the 0.7 factor to account for the 30% optimization by Cong *et al.* Also note that d is $O(\log n)$, but not exactly $\log n$ because each leaf can contain 2λ bits, we compute d accordingly.

Computing Efficiency for Goyal *et al.* We compute the concrete efficiency of Goyal *et al.* over the BLS12-381 curve, similar to how we compute our efficiency. Given the fact that both constructions are very similar we see very similar results. Only in receiver time, we see

[5] Remember that the construction in Cho *et al.* [10] contains a path from root to leaf on a tree, while at each level a garbled circuit is computed.

Table 2: Comparison of asymptotic computation and communication efficiency.

	crsGen	Hash	Send	Receive	crs size	Hash size	Send size
Cho *et al.* [10]	$\mathcal{O}(1)$	$n\text{poly}(\log n)$	$\text{poly}(\log n)$	$\text{poly}(\log n)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\log n)$
Goyal *et al.* [27]	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Goyal *et al.* [27] + §5.1	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(1)$
Alamati *et al.* [2]	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
This work §4	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
This work §4 + §5.1	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(1)$

that our construction really shines in comparison with Goyal *et al.* We give some more details about computing this concrete receiver time below.

Computing Efficiency for Alamati *et al.* To compute the concrete efficiency for Alamati *et al.* we have to compute a few of the parameters first. We note that the authors mention a security parameter λ, but next take an RSA composite number N with λ bits. If we want to achieve about 128 bit security, we need to take $\lambda = 2,048$. Next, for their PPRF trick that reduces the crs to constant size, we need to compute the value ξ. They write that this value needs to be $\mathcal{O}\left((\log 2^{\kappa})^2\right)$, where κ is the size of the different primes they use as exponents and $5\kappa \leq \lambda$. Therefore, κ needs to be around 409 bits, which leads to $\xi \approx (409 \cdot \log 2)^2 \approx 15,129$. Finally, we work in the group $\mathbb{Z}_{N^{\xi+1}}$, i.e. a group with numbers of size up to $2,048 \cdot 15,130 = 30,986,240$. Although, this is all still practically doable as the numbers in Table 3 show, it is not ideal.

Moreover, receiver time is still linear in the database size and it is not possible to use the same $\sqrt{}$-optimization as described in Section 5.1 because this would grow the digest size to around 180GB, which is clearly undesirable in a laconic OT scheme.

Computing Concrete Receiver Computation Time Finally we show how we achieve concretely better receiver time in comparison with all previous work. First, we note that computing the receiver time in the work by Cho *et al.* is nearly impossible, but given the Send size of 1.2PB it would already take 11 days just to transfer this amount of data over a 10Gbps line, let alone handle this amount of data on a reasonably sized computer.

Therefore, we focus our efforts on comparing receiver time with the work of Goyal *et al.* and Alamati *et al.* To get a fair estimate of the receiving time in all constructions we benchmarked all operations on an Apple M1 Max. For Goyal *et al.* and our work we use: 775ns for a multiplication, 325μs for an exponentiation, 1393μs for a pairing, and 4.5μs for a multiplication in the target group. In Goyal *et al.*, we ignored the symbolic evaluation of a degree-n polynomial. For Alamati *et al.* we benchmarked a multiplication in the respective group to take 40ns, we ignored the single exponentiation.

In Table 3, we show the full comparison of concrete efficiency.

7 RELATED WORK

Laconic Oblivious Transfer (OT) was first introduced by Cho *et al.* [10] in comparison with regular OT, laconic OT requires the receiver's outgoing message to be small, more specifically, it shouldn't grow with the size of the database. They prove their system to be secure based on the Decisional Diffie-Hellman (DDH) assumption in the common reference string (CRS) model. To achieve this they use somewhere statistically binding (SSB) hash functions [30] combined with hash proof systems. This specific technique has been refined in several subsequent papers changing names to Chameleon Hashing [12], Hash Encryption [14], or Hash Garbling [9]. Most recently this technique was further optimized in the context of Registration Based Encryption (RBE) [11, 20, 21, 26]. Even with all these improvements the construction remains merely theoretical, although the asymptotic efficiency seems quite optimal, implementing the scheme would require giant garbled circuits impossible to create and evaluate on any normal sized computer.

Goyal *et al.* [27] introduced a construction for One-Way Functions with Encryption (OWFE) from which you can easily build laconic OT in the special case where the one-way function also has a compression property. This is the case for their particular construction, both the common reference string and the receiver time is linear in the database size, similar to our constructions. However, receiver time in our scheme is orders of magnitude faster. In 2021, Alamati *et al.* [2] improved one of the schemes from Goyal *et al.*, achieving nearly optimal asymptotic efficiency, but receiver time is still linear. Moreover, it is not possible to apply the square root-optimization to their scheme, because the digest size would grow to several gigabytes. The scheme is also based on the ϕ-hiding assumption, which is not desirable. Recently, Aranha *et al.* [3] introduced a laconic private set intersection scheme based on pairings. They reduce their scheme to the security of the Strong Bilinear Decisional Diffie-Hellman Problem, which can also be proven secure in the generic group model.

The application that was presented in the original paper that introduced laconic OT [10], has been further developed in work by Garg *et al.* [22] Other laconic primitives such as laconic function evaluation were achieved by Döttling *et al.* [13], Quach *et al.* [40], and Agrawal and Roşie [1]. In the work of Döttling *et al.* [15], they introduce the slightly stricter definition of private laconic OT.

We create the new primitive Set Membership Encryption inspired by Broadcast Encryption, a primitive that was first introduced by Fiat and Naor [19]. We are interested in the instantiations that have a short ciphertext as introduced by Boneh *et al.* [7], which has very good efficiency, but we can only prove a derivative of this scheme to be selectively secure. Therefore, we look to adaptively secure broadcast encryption schemes that are proven secure using the Dual System Encryption technique of Waters [47], but we distill a broadcast encryption system of the follow-up work by Lewko and Waters [39] in a composite order bilinear group, unfortunately, the private keys grow to $\mathcal{O}(n)$.

Table 3: Comparison of concrete efficiency, with $n = 2^{31}$. Cho *et al.* is estimated over elliptic curve secp192k1 with $\lambda = 96$, Goyal *et al.* and our work are estimated on the BLS12-381 curve with security parameter roughly 120 bits, and Alamati *et al.* is estimated using an RSA group of 2048 bits to achieve around 128 bit security. (kB = 1000 bytes, MB = 1 million bytes, GB = 1 billion bytes, PB = 1 quadrillion bytes)

	λ	crs size	Hash size	Send size	Receive
Cho *et al.* [10]	96 bits	4.6kB	48 bytes	1.2PB	-
Goyal *et al.* [27]	~ 120 bits	103.1GB	48 bytes	1.25kB	8.1 days
Goyal *et al.* [27] + §5.1	~ 120 bits	2.2MB	2.2MB	1.25kB	15.1s
Alamati *et al.* [2]	~ 128 bits	0.8MB	3.9MB	7.7MB	85.9s
This work §4	~ 120 bits	412.3GB	48 bytes	1.34kB	27.7 minutes
This work §4 + §5.1	~ 120 bits	8.9MB	2.2MB	1.34kB	38.7ms

8 CONCLUSION

We introduced a new primitive which we call set membership encryption, where one party can define a set of receivers and a second party can encrypt to one specific receiver if and only if that receiver was part of the original set of receivers. We show how to build laconic OT from this primitive. Next, we show a construction of this new primitive, which has very practical concrete efficiency.

Finally, we evaluate the efficiency of the laconic OT schemes that can be derived from said set membership encryption constructions. We compare with previous work on laconic OT and show that ours is several orders of magnitude more efficient.

Future Work Improving the size of the public parameters is important future work to further increase the efficiency of this primitive. Given work on improving the efficiency of broadcast encryption we could hope to similarly improve this new primitive. Studying if this could happen under the stronger adaptive security should be part of that future work. Another interesting question is if we can create a private or anonymous version of set membership encryption similar to what has been studied for private/anonymous broadcast encryption.

We give a few pointers of what can be investigated in future work:

Using different broadcast encryption schemes Given the extensive literature on broadcast encryption, it seems likely that other schemes might have the same type of property, where two keys for a different set of receivers can somehow be bound together. This could lead to even more efficient laconic OT constructions from bilinear maps or other assumptions.

Decreasing CRS size Given the nature of how we use broadcast encryption, it seems inherent that the CRS is always going to be linear in the size of the database. However, we can hope to create constructions that can generate both public key and private keys on the fly. Note that for the private keys, this would mean that the master secret key or some derivation probably needs to be part of the CRS, in which case it is important that the binding between both parts of the key is happening inside the msk.

Decreasing decryption time Although the linear decryption time follows directly from the BE schemes that we use, there is a lot of research on anonymous BE schemes. These schemes do not take the receiver set as input during decryption, therefore, we could hope that they will not be linear in the size of that set of receivers and therefore, not linear in the size of the database during decryption.

Acknowledgments. Matthew Green and Gijs Van Laer were supported by NSF under awards CNS-1653110, CNS-1801479, and by DARPA under Contract No. HR001120C0084. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Government or DARPA.

Abhishek Jain is supported in part by NSF CNS-1814919, NSF CAREER 1942789, Johns Hopkins University Catalyst award, JP Morgan Faculty Award, and research gifts from Ethereum, Stellar and Cisco.

REFERENCES

[1] Shweta Agrawal and Razvan Roşie. 2021. Adaptively Secure Laconic Function Evaluation for NC^1. (2021).

[2] Navid Alamati, Pedro Branco, Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, and Sihang Pu. 2021. Laconic Private Set Intersection and Applications. Cryptology ePrint Archive, Report 2021/728. https://eprint.iacr.org/2021/728.

[3] Diego Aranha, Chuanwei Lin, Claudio Orlandi, and Mark Simkin. 2022. Laconic Private Set-Intersection From Pairings. Cryptology ePrint Archive, Report 2022/529. https://eprint.iacr.org/2022/529.

[4] Donald Beaver. 1996. Correlated Pseudorandomness and the Complexity of Private Computations. In *28th ACM STOC*. ACM Press, 479–488. https://doi.org/10.1145/237814.237996

[5] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. 2005. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In *EUROCRYPT 2005 (LNCS, Vol. 3494)*, Ronald Cramer (Ed.). Springer, Heidelberg, 440–456. https://doi.org/10.1007/11426639_26

[6] Dan Boneh and Matthew K. Franklin. 2001. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001 (LNCS, Vol. 2139)*, Joe Kilian (Ed.). Springer, Heidelberg, 213–229. https://doi.org/10.1007/3-540-44647-8_13

[7] Dan Boneh, Craig Gentry, and Brent Waters. 2005. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *CRYPTO 2005 (LNCS, Vol. 3621)*, Victor Shoup (Ed.). Springer, Heidelberg, 258–275. https://doi.org/10.1007/11535218_16

[8] Dan Boneh, Amit Sahai, and Brent Waters. 2011. Functional Encryption: Definitions and Challenges. In *TCC 2011 (LNCS, Vol. 6597)*, Yuval Ishai (Ed.). Springer, Heidelberg, 253–273. https://doi.org/10.1007/978-3-642-19571-6_16

[9] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. 2018. Anonymous IBE, Leakage Resilience and Circular Security from New Assumptions. In *EUROCRYPT 2018, Part I (LNCS, Vol. 10820)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer, Heidelberg, 535–564. https://doi.org/10.1007/978-3-319-78381-9_20

[10] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. 2017. Laconic Oblivious Transfer and Its Applications. In *CRYPTO 2017, Part II (LNCS, Vol. 10402)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, Heidelberg, 33–65. https://doi.org/10.1007/978-3-319-63715-0_2

[11] Kelong Cong, Karim Eldefrawy, and Nigel P. Smart. 2021. Optimizing Registration Based Encryption. Cryptology ePrint Archive, Report 2021/499. https://ia.cr/2021/499.

[12] Nico Döttling and Sanjam Garg. 2017. Identity-Based Encryption from the Diffie-Hellman Assumption. In *CRYPTO 2017, Part I (LNCS, Vol. 10401)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, Heidelberg, 537–569. https://doi.org/10.1007/978-3-319-63688-7_18

[13] Nico Döttling, Sanjam Garg, Vipul Goyal, and Giulio Malavolta. 2019. Laconic Conditional Disclosure of Secrets and Applications. In *60th FOCS*, David Zuckerman (Ed.). IEEE Computer Society Press, 661–685. https://doi.org/10.1109/FOCS.2019.00046

[14] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, and Daniel Masny. 2018. New Constructions of Identity-Based and Key-Dependent Message Secure Encryption Schemes. In *PKC 2018, Part I (LNCS, Vol. 10769)*, Michel Abdalla and Ricardo Dahab (Eds.). Springer, Heidelberg, 3–31. https://doi.org/10.1007/978-3-319-76578-5_1

[15] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. 2019. Trapdoor Hash Functions and Their Applications. In *CRYPTO 2019, Part III (LNCS, Vol. 11694)*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer, Heidelberg, 3–32. https://doi.org/10.1007/978-3-030-26954-8_1

[16] Nico Döttling, Dimitris Kolonelos, Russell W. F. Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. 2023. Efficient Laconic Cryptography from Learning with Errors. In *EUROCRYPT 2023, Part III (LNCS, Vol. 14006)*, Carmit Hazay and Martijn Stam (Eds.). Springer, Heidelberg, 417–446. https://doi.org/10.1007/978-3-031-30620-4_14

[17] Harry Eldridge, Aarushi Goel, Matthew Green, Abhishek Jain, and Maximilian Zinkus. 2022. One-Time Programs from Commodity Hardware. Cryptology ePrint Archive, Report 2022/1257. https://eprint.iacr.org/2022/1257.

[18] Uriel Feige, Dror Lapidot, and Adi Shamir. 1990. Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract). In *31st FOCS*. IEEE Computer Society Press, 308–317. https://doi.org/10.1109/FSCS.1990.89549

[19] Amos Fiat and Moni Naor. 1994. Broadcast Encryption. In *CRYPTO'93 (LNCS, Vol. 773)*, Douglas R. Stinson (Ed.). Springer, Heidelberg, 480–491. https://doi.org/10.1007/3-540-48329-2_40

[20] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. 2018. Registration-Based Encryption: Removing Private-Key Generator from IBE. In *TCC 2018, Part I (LNCS, Vol. 11239)*, Amos Beimel and Stefan Dziembowski (Eds.). Springer, Heidelberg, 689–718. https://doi.org/10.1007/978-3-030-03807-6_25

[21] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, Ahmadreza Rahimi, and Sruthi Sekar. 2019. Registration-Based Encryption from Standard Assumptions. In *PKC 2019, Part II (LNCS, Vol. 11443)*, Dongdai Lin and Kazue Sako (Eds.). Springer, Heidelberg, 63–93. https://doi.org/10.1007/978-3-030-17259-6_3

[22] Sanjam Garg, Rafail Ostrovsky, and Akshayaram Srinivasan. 2018. Adaptive Garbled RAM from Laconic Oblivious Transfer. In *CRYPTO 2018, Part III (LNCS, Vol. 10993)*, Hovav Shacham and Alexandra Boldyreva (Eds.). Springer, Heidelberg, 515–544. https://doi.org/10.1007/978-3-319-96878-0_18

[23] Noemi Glaeser, Dimitris Kolonelos, Giulio Malavolta, and Ahmadreza Rahimi. 2022. Efficient Registration-Based Encryption. Cryptology ePrint Archive, Report 2022/1505. https://eprint.iacr.org/2022/1505.

[24] Oded Goldreich, Silvio Micali, and Avi Wigderson. 1987. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *19th ACM STOC*, Alfred Aho (Ed.). ACM Press, 218–229. https://doi.org/10.1145/28395.28420

[25] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. 2008. One-Time Programs. In *CRYPTO 2008 (LNCS, Vol. 5157)*, David Wagner (Ed.). Springer, Heidelberg, 39–56. https://doi.org/10.1007/978-3-540-85174-5_3

[26] Rishab Goyal and Satyanarayana Vusirikala. 2020. Verifiable Registration-Based Encryption. In *CRYPTO 2020, Part I (LNCS, Vol. 12170)*, Daniele Micciancio and Thomas Ristenpart (Eds.). Springer, Heidelberg, 621–651. https://doi.org/10.1007/978-3-030-56784-2_21

[27] Rishab Goyal, Satyanarayana Vusirikala, and Brent Waters. 2020. New Constructions of Hinting PRGs, OWFs with Encryption, and More. In *CRYPTO 2020, Part I (LNCS, Vol. 12170)*, Daniele Micciancio and Thomas Ristenpart (Eds.). Springer, Heidelberg, 527–558. https://doi.org/10.1007/978-3-030-56784-2_18

[28] Jens Groth. 2016. On the Size of Pairing-Based Non-interactive Arguments. In *EUROCRYPT 2016, Part II (LNCS, Vol. 9666)*, Marc Fischlin and Jean-Sébastien Coron (Eds.). Springer, Heidelberg, 305–326. https://doi.org/10.1007/978-3-662-49896-5_11

[29] Jens Groth and Amit Sahai. 2008. Efficient Non-interactive Proof Systems for Bilinear Groups. In *EUROCRYPT 2008 (LNCS, Vol. 4965)*, Nigel P. Smart (Ed.). Springer, Heidelberg, 415–432. https://doi.org/10.1007/978-3-540-78967-3_24

[30] Pavel Hubacek and Daniel Wichs. 2015. On the Communication Complexity of Secure Function Evaluation with Long Output. In *ITCS 2015*, Tim Roughgarden (Ed.). ACM, 163–172. https://doi.org/10.1145/2688073.2688105

[31] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. 2003. Extending Oblivious Transfers Efficiently. In *CRYPTO 2003 (LNCS, Vol. 2729)*, Dan Boneh (Ed.). Springer, Heidelberg, 145–161. https://doi.org/10.1007/978-3-540-45146-4_9

[32] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. 2011. Efficient Non-interactive Secure Computation. In *EUROCRYPT 2011 (LNCS, Vol. 6632)*, Kenneth G. Paterson (Ed.). Springer, Heidelberg, 406–425. https://doi.org/10.1007/978-3-642-20465-4_23

[33] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. 2008. Founding Cryptography on Oblivious Transfer - Efficiently. In *CRYPTO 2008 (LNCS, Vol. 5157)*, David Wagner (Ed.). Springer, Heidelberg, 572–591. https://doi.org/10.1007/978-3-540-85174-5_32

[34] Bargav Jayaraman, Hannah Li, and David Evans. 2017. Decentralized Certificate Authorities. *CoRR* abs/1706.03370 (2017). arXiv:1706.03370 http://arxiv.org/abs/1706.03370

[35] Joe Kilian. 1988. Founding Cryptography on Oblivious Transfer. In *20th ACM STOC*. ACM Press, 20–31. https://doi.org/10.1145/62212.62215

[36] Vladimir Kolesnikov and Thomas Schneider. 2008. Improved Garbled Circuit: Free XOR Gates and Applications. In *ICALP 2008, Part II (LNCS, Vol. 5126)*, Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz (Eds.). Springer, Heidelberg, 486–498. https://doi.org/10.1007/978-3-540-70583-3_40

[37] Russell W. F. Lai, Giulio Malavolta, and Viktoria Ronge. 2019. Succinct Arguments for Bilinear Group Arithmetic: Practical Structure-Preserving Cryptography. In *ACM CCS 2019*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, 2057–2074. https://doi.org/10.1145/3319535.3354262

[38] Enrique Larraia. 2015. Extending Oblivious Transfer Efficiently - or - How to Get Active Security with Constant Cryptographic Overhead. In *LATINCRYPT 2014 (LNCS, Vol. 8895)*, Diego F. Aranha and Alfred Menezes (Eds.). Springer, Heidelberg, 368–386. https://doi.org/10.1007/978-3-319-16295-9_20

[39] Allison B. Lewko and Brent Waters. 2010. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010 (LNCS, Vol. 5978)*, Daniele Micciancio (Ed.). Springer, Heidelberg, 455–479. https://doi.org/10.1007/978-3-642-11799-2_27

[40] Willy Quach, Hoeteck Wee, and Daniel Wichs. 2018. Laconic Function Evaluation and Applications. In *59th FOCS*, Mikkel Thorup (Ed.). IEEE Computer Society Press, 859–870. https://doi.org/10.1109/FOCS.2018.00086

[41] Michael O. Rabin. 2005. How To Exchange Secrets with Oblivious Transfer. Cryptology ePrint Archive, Report 2005/187. https://eprint.iacr.org/2005/187.

[42] Certicom Research. 2010. SEC 2: Recommended elliptic curve domain parameters. https://www.secg.org/sec2-v2.pdf [Online; Accessed on October 27, 2021].

[43] Amit Sahai and Brent R. Waters. 2005. Fuzzy Identity-Based Encryption. In *EUROCRYPT 2005 (LNCS, Vol. 3494)*, Ronald Cramer (Ed.). Springer, Heidelberg, 457–473. https://doi.org/10.1007/11426639_27

[44] Peter Scholl. 2018. Extending Oblivious Transfer with Low Communication via Key-Homomorphic PRFs. In *PKC 2018, Part I (LNCS, Vol. 10769)*, Michel Abdalla and Ricardo Dahab (Eds.). Springer, Heidelberg, 554–583. https://doi.org/10.1007/978-3-319-76578-5_19

[45] Victor Shoup. 1997. Lower Bounds for Discrete Logarithms and Related Problems. In *EUROCRYPT'97 (LNCS, Vol. 1233)*, Walter Fumy (Ed.). Springer, Heidelberg, 256–266. https://doi.org/10.1007/3-540-69053-0_18

[46] Nigel Smart. 2020. Twitter thread: How many AND gates would there be in a combinatorial circuit for an elliptic curve point multiplication? https://twitter.com/SmartCryptology/status/1327280495978278914 [Online; @SmartCryptology].

[47] Brent Waters. 2009. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *CRYPTO 2009 (LNCS, Vol. 5677)*, Shai Halevi (Ed.). Springer, Heidelberg, 619–636. https://doi.org/10.1007/978-3-642-03356-8_36

[48] Andrew Chi-Chih Yao. 1982. Protocols for Secure Computations (Extended Abstract). In *23rd FOCS*. IEEE Computer Society Press, 160–164. https://doi.org/10.1109/SFCS.1982.38

[49] Samee Zahur, Mike Rosulek, and David Evans. 2015. Two Halves Make a Whole - Reducing Data Transfer in Garbled Circuits Using Half Gates. In *EUROCRYPT 2015, Part II (LNCS, Vol. 9057)*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer, Heidelberg, 220–250. https://doi.org/10.1007/978-3-662-46803-6_8