# Laconic Private Set Intersection and Applications

Navid Alamati[1], Pedro Branco[2] Nico Döttling[3], Sanjam Garg[1,4], Mohammad Hajiabadi[5], and Sihang Pu[3]

[1] UC Berkeley
[2] IT, IST - University of Lisbon
[3] Helmholtz Center for Information Security (CISPA)
[4] NTT Research
[5] University of Waterloo

**Abstract.** Consider a server with a *large* set $S$ of strings $\{x_1, x_2 \ldots, x_N\}$ that would like to publish a *small* hash $h$ of its set $S$ such that any client with a string $y$ can send the server a *short* message allowing it to learn $y$ if $y \in S$ and nothing otherwise. In this work, we study this problem of two-round private set intersection (PSI) with low (asymptotically optimal) communication cost, or what we call *laconic* private set intersection ($\ell$PSI) and its extensions. This problem is inspired by the recent general frameworks for laconic cryptography [Cho et al. CRYPTO 2017, Quach et al. FOCS'18].
We start by showing the first feasibility result for realizing $\ell$PSI based on the CDH assumption, or LWE with polynomial noise-to-modulus ratio. However, these feasibility results use expensive non-black-box cryptographic techniques leading to significant inefficiency. Next, with the goal of avoiding these inefficient techniques, we give a construction of $\ell$PSI schemes making only black-box use of cryptographic functions. Our construction is secure against semi-honest receivers, malicious senders and reusable in the sense that the receiver's message can be reused across any number of executions of the protocol. The scheme is secure under the $\phi$-hiding, decisional composite residuosity and subgroup decision assumptions.
Finally, we show natural applications of $\ell$PSI to realizing a semantically-secure encryption scheme that supports detection of encrypted messages belonging to a set of "illegal" messages (e.g., an illegal video) circulating online. Over the past few years, significant effort has gone into realizing laconic cryptographic protocols. Nonetheless, our work provides the first black-box constructions of such protocols for a natural application setting.

## 1 Introduction

Laconic cryptography [13,40,20,18] is an emerging paradigm which enables realizing cryptographic tasks with asymptotically-optimal communication in just two messages. In this setting, the receiver has a potentially large input, and the size of her protocol message only depends on the security parameter, and not her input size. The second message, sent by the sender, may grow with the size of the sender's input, but should be independent of the receiver's input size.

The pioneering work of [13] introduced the notion of laconic oblivious transfer (laconic OT), which allows a receiver with a large input $D \in \{0,1\}^n$ to send a short hash digest $h$ of her input $D$. Next, a sender with an input $(i \in [n], m_0, m_1)$, sends a short message $\mathsf{ots}$ to the receiver, enabling the receiver to learn $m_{D[i]}$, and nothing more. We require (a) the sizes of $h$ and $\mathsf{ots}$ be $\mathsf{poly}(\log(n), \lambda)$, where $\lambda$ is the security parameter; (b) the sender's computation time be $\mathsf{poly}(\log(n), \lambda)$ and (c) and receiver's second-phase computation time be $\mathsf{poly}(\log(n), \lambda)$.

The notion of laconic OT, and the techniques built around it, have led to breakthrough results in the last few years, which, among others, include the first construction of identity-based encryption from CDH [17,16,8,19], and two-round MPC protocols from minimal assumptions [24,25,4].

*Laconism beyond OT?* Motivated by the developments enabled by laconic OT, it is natural to ask whether we can push the boundary further, realizing laconism for richer functionalities. Laconic OT by itself does not seem to be sufficient for this task (at least generically). Specifically, the general laconic OT+garbled circuit based approach for a function $f(\cdot, \cdot)$ results in protocols in which the size of the sender's protocol message grows with the receiver's input size.

The work of Quach, Wee and Wichs [40] shows how to realize laconic cryptography for general functionalities using LWE. However, two significant issues remain. Firstly, it is not clear whether we can achieve laconism from other assumptions, for functionalities beyond OT. As mentioned above, research in laconic OT has led to several breakthrough feasibility results, motivating the need for developing techniques that can be realized using wider assumptions and for richer functionalities. Secondly, existing constructions of laconic primitives are non-black-box, leading to inefficient constructions. Addressing the above shortcomings, our goals are twofold: (1) Feasibility: Can we realize laconic primitives beyond OT from assumptions other than LWE? and (2) Black-boxness: Can we make the constructions black-box?

*Black-box techniques.* We use the notion "black-box" techniques in the sense that the construction should not use an explicit circuit-level description of cryptographic primitives. In this sense, we think of constructions which e.g., compute cryptographic primitives inside garbled circuits (as previous laconic OT constructions) or use general purpose NIZK proofs (which express statements in terms of NP-complete languages) as "non-black-box" techniques.

*Laconic PSI.* We make the first progress toward the above two goals with respect to a non-trivial functionality: Laconic Private Set Intersection ($\ell$PSI) and its family. Private set intersection (PSI) is a cryptographic primitive that allows two parties to learn the intersection of their input sets and nothing else. Because of its usefulness and versatility, this cryptographic primitive has been extensively studied in numerous settings throughout the years (see e.g., [35,39,31,42,36,38] and references therein).

Laconic PSI allows a receiver to send a short digest of its large data set, which in turn can be used by a sender to compute a PSI second round message. We require that the total communication complexity as well as the sender's running time to be independent of the receiver's input size.

## 1.1 Our Results

As our first result, we give a generic construction of laconic PSI from a primitive called anonymous hash encryption, which in turn can be realized from CDH/LWE [17,16,8]. Our construction builds on the Merkle-tree garbled circuit based approach of [17,16,8,22,23,26], showing how to use garbled circuits to perform binary search on a set of sorted values. Prior to our work there did not exist any construction of a laconic primitive from CDH beyond OT. We also obtain an LWE instantiation with polynomial modulus to noise ration, improving the subexponential ratio of [40].

The above construction is non-black-box caused by the use of garbled circuits. As our second contribution, we achieve a black-box construction of laconic PSI from the $\phi$-hiding assumption.

Both constructions above are only semi-honest secure, and can be made malicious (UC) secure by using Non-Interactive Zero Knowledge (NIZK).[6] However, the eventual protocol will be non-black-box. To enhance applicability, we show how to make our second construction secure against malicious senders, and semi-honest receivers in the CRS model, by additionally assuming decisional composite residuosity (DCR) and subgroup decision assumptions. We term this notion *reusable malicious* laconic PSI, meaning the receiver's message may be re-used.[7]

*Applications.* We show an application of laconic PSI in realizing a primitive that we dub self-detecting encryption. A self-detecting encryption acts like a normal public-key encryption with a key difference that it is possible to detect whether the underlying message of a given ciphertext belongs to a database of special (e.g., "illegal") messages. This can be determined just by knowing the database values, as opposed to the system's secret key. Such encryption systems provide a feature for detecting the presence of illegal contents, without compromising the privacy of legal messages. There has only been a limited number of proposals for this task so far, and all of them use heavy tools (e.g., FHE) for this purpose (see [28] for more details). We formally define this notion, and show how to realize it using laconic PSI.

In a self-detecting encryption, an authority (e.g., a government entity or a delegated NGO) publishes a small hash value of a (possibly large) database of

[6] Note that in the laconic setting we cannot prove malicious security against a receiver since it is information-theoretically impossible to extract its input. Thus, since the NIZK will only be computed by the sender, the protocol will remain laconic.

[7] We use the word reusability only in conjunction with malicious security, since in the semi-honest setting, reusability is satisfied by default.

special messages such that a user can encrypt a message using the system's public key and the hash value. If the message belongs to the database, then the authority can detect it; else, the message remains hidden to the authority. We require that the size of the hash and the encryption running time to be independent of the database size.

We note that attribute-based encryption does not provide a solution to the above problem, because either the authority should reveal its database to a master-key generator, or it should be the master-key generator itself – both of which defeat our security purposes.

*Additional new results: Labeled laconic PSI and malicious laconic OT (LOT).* We extend our laconic PSI techniques to build a reusable *labeled* laconic PSI. Labeled PSI [34,11] is a flavor of PSI, where the sender holds a *label* $\ell_i$ associated with each set element $x_i$, and the receiver will learn the labels corresponding to the intersection elements. Labeled PSI has several practical applications (e.g., private web service queries [11]).

Moreover, we show how to use our techniques to realize the first construction of a reusable LOT secure against malicious senders and semi-honest receivers.

*DV-NIZK range proofs for DJ ciphertexts.* As a building block for our laconic PSI protocol, we propose a Designated-Verifier Non-Interactive Zero-Knowledge (DV-NIZK[8]) scheme for range proof with Damgård Jurik (DJ) ciphertexts, which may be of independent interest. Our DV-NIZK has statistical simulation soundness and computational zero-knowledge given that the subgroup decision (SD) assumption holds [5,29].

Such range proofs can also be constructed in the random oracle model (ROM) via the Fiat-Shamir transform (e.g., [15,3,9,43]), which might yield the best efficiency. As our LPSI construction is modular, this can be done independently of the remaining results in the paper. The goal of our DV-NIZK is to provide an efficient standard model construction which we see as a reasonable middle ground between feasibility from the weakest assumption (at the cost of unrealistic efficiency) and practical efficiency (at the cost of relying on strong heuristic assumptions such as the ROM).

### 1.2 Previous Work

Laconic PSI can be seen as a particular case of unbalanced PSI. Protocols for unbalanced PSI were presented in [2,41,12,11]. The protocol of [41] achieves linear communication complexity on the receiver's set size in the *pre-processing* model. The protocols of [12,11] rely on somewhat homomorphic encryption (SWHE) and proceed in two rounds. However, the communication complexity scales with the size of the receiver's set (and logarithmic with the size of the sender's set), in contrast with our protocol whose communication complexity scales with the sender's set size.

[8] DV-NIZK *only* allows the designated prover to prove that it holds a witness for a certain $\mathsf{NP}$ statement to a verifier in just one message

*Comparison with [2].* Ateniese et al. in [2] proposed a semi-honest size-hiding PSI protocol[9] inspired by RSA accumulators that achieves communication complexity independent of the receiver's set size. However, we emphasize that their scheme does not fit the framework of laconic cryptography since it requires the sender to know the factorization of a CRS modulus N. Thus, either it requires pre-processing (giving a designated secret key to the sender), or it requires three rounds in the CRS model. In contrast, laconic cryptography requires (a) two rounds and (b) no pre-processing (i.e., neither party receives a secret key correlated with the CRS). Both (a) and (b) are crucially used in applications of laconic cryptography. Specifically, these restrictions prevent use of [2] in settings with multiple senders, an aspect that has been critical for laconic cryptography applications. Finally, we remark that the security of [2] relies on random oracles, whereas we prove security in the standard model and achieve a substantially stronger security notion without resorting to heavy generic tools.

All of above constructions are just secure against semi-honest adversaries, except for [11] which achieves security against a malicious receiver.

### 1.3 Open Problems

The main open question is to realize laconic cryptography for functionalities richer than PSI. A second question is to build laconic PSI in a black-box way from assumptions not involving $\phi$-hiding (e.g., pairings alone).

In this work, we build DV-NIZK for proving equality of plaintexts across different encryption schemes, namely between the DJ [15] and the BGN [5,29] encryption schemes. This scheme opens the door to new applications since it allows us to extend the capabilities of GS/GOS proof systems [29,30] to non-pairing-based primitives with additional properties (in our case to the DJ cryptosystem). We believe that these ideas will have applications beyond range proofs, e.g., one can think of further uses of structure preserving cryptography, so we leave this as an open problem for future works.

## 2 Technical Overview

### 2.1 Semi-Honest PSI from CDH/LWE

Our protocol uses hash encryption and garbled circuits, building on [17,8,22], while introducing new techniques. A hash-encryption scheme allows one to encrypt a message $m$ to the output $h$ of a hash function by specifying an index/bit $(i, b)$ (denoted $\mathsf{HEnc}(h, m, (i, b))$), so that knowledge of a consistent pre-image value $z$ allows for decryption ($\mathsf{Hash}(z) = h$ and $z_i = b$) while having semantic security against inconsistent pre-image values (i.e., against $z$ where $\mathsf{Hash}(z) = h$ but $z_i = \bar{b}$).[10]

In all discussion below we assume the sender's and receiver's elements are in $\{0,1\}^\lambda$ and that the output of $\mathsf{Hash}$ also has $\lambda$ bits.

[9] Such schemes were also studied in [33,37,32].

[10] $\mathsf{Enc}$ also takes as input a public parameter $\mathsf{pp}$, which we ignore here.

*Receiver's set size is 2.* We first assume the receiver has only two elements $S_{\mathsf{R}} = \{\mathsf{id}_1, \mathsf{id}_2\}$ and the sender has a single element $\mathsf{id}$. The receiver sends $\mathsf{hr}_{\mathsf{root}} := \mathsf{Hash}(\mathsf{id}_1, \mathsf{id}_2)$. Consider a circuit $\mathsf{F}[\mathsf{id}]$, with $\mathsf{id}$ hardwired, which on input $(\mathsf{id}', \mathsf{id}'')$ outputs $\mathsf{id}$ if $\mathsf{id} \in \{\mathsf{id}', \mathsf{id}''\}$; else, $\perp$. The sender garbles $\mathsf{F}[\mathsf{id}]$ to get $(\widetilde{\mathsf{C}}_0, \{\mathsf{lb}_{i,b}\})$[11] and sends $\mathsf{psi}_2 := (\widetilde{\mathsf{C}}_0, \{\mathsf{ct}_{i,b}\})$, where $\mathsf{ct}_{i,b} := \mathsf{HEnc}(\mathsf{hr}_{\mathsf{root}}, \mathsf{lb}_{i,b}, (i, b)))$. The receiver who has the pre-image $z := (\mathsf{id}_1, \mathsf{id}_2)$ can retrieve only the labels $\mathsf{lb}_{i,z_i}$, and the rest will be hidden. Thus, by garbled circuit security the receiver will only learn the output of $\mathsf{F}[\mathsf{id}](\mathsf{id}_1, \mathsf{id}_2)$, as desired.

*Moving beyond $|S_{\mathsf{R}}| = 2$.* Suppose the receiver has four elements $S_{\mathsf{R}} = \{\mathsf{id}_1, \mathsf{id}_2, \mathsf{id}_3, \mathsf{id}_4\}$ in ascending order. The receiver Merkle-hashes all these values and sends $\mathsf{hr}_{\mathsf{root}}$, the root hash. Let $h_1$ and $h_2$ be the two hash values at level one (i.e., $h_1 = \mathsf{Hash}(\mathsf{id}_1, \mathsf{id}_2)$). If the sender knows the value of, say, $h_1$, he may hash-encrypt $\{\mathsf{lb}_{i,b}\}$ (defined in the previous paragraph) under $h_1$, so that the receiver can only open the labels that correspond to the bits of $z = (\mathsf{id}_1, \mathsf{id}_2)$, revealing the value of $\mathsf{F}[\mathsf{id}](\mathsf{id}_1, \mathsf{id}_2)$. However, $h_1$ is statistically hidden given $\mathsf{hr}_{\mathsf{root}}$. Thus, we use the idea of deferred evaluation [17,13,16,8], delegating the task of hash-encrypting $\{\mathsf{lb}_{i,b}\}$ to the receiver herself, via garbled circuits.

In essence, we want the receiver to be able to compute the hash encryption of $\{\mathsf{lb}_{i,b}\}$ wrt either $h_1$ or $h_2$ (depending on whether $\mathsf{id} \le \mathsf{id}_2$ or not), but not both; because obtaining both hash encryptions will allow the receiver to open both labels $\mathsf{lb}_{i,0}$ and $\mathsf{lb}_{i,1}$ for some indices $i$ (because $(\mathsf{id}_1, \mathsf{id}_2) \neq (\mathsf{id}_3, \mathsf{id}_4)$), destroying garbled circuit security. Thus, the sender has to make sure that the receiver will be able to obtain only either of the above hash encryptions, the one whose sub-tree contains $\mathsf{id}$. To enable this, we perform binary search.

*Performing binary search.* We handle the above difficulty by performing *binary search* using ideas developed in the context of registration-based encryption [22]. The hash of each node is now computed as the hash of the concatenation of its left child's hash, right child's hash, and the largest identity under its left child. For example, the hash root is $\mathsf{hr}_{\mathsf{root}} = \mathsf{Hash}(h_1, h_2, \mathsf{id}_2)$, where $h_1$ and $h_2$ are the hash values of the two nodes in the first level, and in turn $h_1 = \mathsf{Hash}(\mathsf{id}_1, \mathsf{id}_2, \mathsf{id}_1)$. Now let $\mathsf{id}$ be the sender's element, and change $\mathsf{F}[\mathsf{id}]$ to be a circuit that on input $(\mathsf{id}', \mathsf{id}'', *)$ outputs $\mathsf{id}$ if $\mathsf{id} \in \{\mathsf{id}', \mathsf{id}''\}$, else $\perp$. Letting $(\widetilde{\mathsf{C}}_0, \{\mathsf{lb}_{i,b}\})$ be the garbling of $\mathsf{F}[\mathsf{id}]$, consider a circuit $\mathsf{G}[\mathsf{id}, \{\mathsf{lb}_{i,b}\}]$ which on input $(h, h', \mathsf{id}')$ outputs a hash-encryption of $\mathsf{lb}_{i,b}$ either under $h$ or under $h'$, depending on whether $\mathsf{id} \le \mathsf{id}'$ or $\mathsf{id} > \mathsf{id}'$. Let $(\widetilde{\mathsf{C}}', \{\mathsf{lb}'\}_{i,b})$ be the garbling of $\mathsf{G}[\mathsf{id}, \{\mathsf{lb}_{i,b}\}]$, let $\{\mathsf{ct}_{i,b}\}$ be the hash encryption of $\{\mathsf{lb}'_{i,b}\}$ wrt $\mathsf{hr}_{\mathsf{root}}$, and return $\mathsf{psi}_2 := (\widetilde{\mathsf{C}}_0, \widetilde{\mathsf{C}}', \{\mathsf{ct}_{i,b}\})$. Using the pre-image $z := (h_1, h_2, \mathsf{id}_2)$ of $\mathsf{hr}_{\mathsf{root}}$, the receiver can retrieve the labels $\{\mathsf{lb}'_{i,z[i]}\}$, allowing to compute $\mathsf{G}[\mathsf{id}, \{\mathsf{lb}_{i,b}\}](h_1, h_2, \mathsf{id}_2)$, which will produce a hash encryption $\{\mathsf{ct}'_{i,b}\}$ of $\{\mathsf{lb}_{i,b}\}$ under either $h_1$ or $h_2$, depending on whether $\mathsf{id} \le \mathsf{id}_2$, or not. For concreteness, suppose $\mathsf{id} \le \mathsf{id}_2$, meaning that $\{\mathsf{ct}'_{i,b}\}$ are formed under $h_1$, and so the pre-image $z' = (\mathsf{id}_1, \mathsf{id}_2, \mathsf{id}_1)$ of $h_1$ will lead to $\{\mathsf{lb}_{i,z'_i}\}$, which along

[11] $\widetilde{\mathsf{C}}_0$ stands for the garbled circuit and $\{\mathsf{lb}_{i,b}\}_i$ are the corresponding labels of inputs.

with $\widetilde{\mathsf{C}}_0$ will reveal the value of $\mathsf{F}[\mathsf{id}](\mathsf{id}_1, \mathsf{id}_2, \mathsf{id}_1)$. Of course, the receiver *a priori* does not know whether $\{\mathsf{ct}'_{i,b}\}$ are encryptions under $h_1$ or $h_2$, so the receiver should try decrypting wrt both, and see which one succeeds.

*Are we done?* Unfortunately, when arguing security, a subtle issue emerges. Suppose a hash-encryption ciphertext reveals its hash value (e.g., the hash is appended to the ciphertext). Then, the ciphertexts $\{\mathsf{ct}'_{i,b}\}$ will reveal whether they were encrypted under $h_1$ or $h_2$; equivalently, whether $\mathsf{id} \leq \mathsf{id}_2$ or $\mathsf{id} > \mathsf{id}_2$. We cannot allow this information to be leaked if $\mathsf{id} \notin S_{\mathsf{R}}$. To fix this issue we assume the hash-encryption scheme is *anonymous*, meaning that, roughly, a random ciphertext leaks no information about the underlying hash value. This property was defined in [8] for achieving anonymous IBE. The use of anonymous hash encryption does not resolve the issue completely yet. For concreteness, suppose $\mathsf{id} < \mathsf{id}_1$. This means that $\{\mathsf{ct}'_{i,b}\}$ is encrypted under $h_1$, and so by decrypting $\{\mathsf{ct}'_{i,b}\}$ using $z' = (\mathsf{id}_1, \mathsf{id}_2, \mathsf{id}_1)$, the receiver will obtain meaningful labels, evaluating the garbled circuit $\widetilde{\mathsf{C}}_0$ to $\perp$ (rightly so, because $\mathsf{id} \notin S_{\mathsf{R}}$). On the other hand, if the receiver tries decrypting $\{\mathsf{ct}'_{i,b}\}$ using $z'' = (\mathsf{id}_3, \mathsf{id}_4, \mathsf{id}_3)$ which is not a pre-image of $h_1$, then the resulting labels will be meaningless, evaluating $\widetilde{\mathsf{C}}_0$ to junk. This leaks which path is the right binary search path, giving information about $\mathsf{id}$. To fix this issue, we change the circuit $\mathsf{F}$ so that if $\mathsf{id} \notin S_{\mathsf{R}}$, then decryption along any path will result in a random value. Specifically, sample two random values $r$ and $r'$, let $\mathsf{F}[\mathsf{id}, r, r'](\mathsf{id}', \mathsf{id}'', *)$ return $r$ if $\mathsf{id} \notin \{\mathsf{id}', \mathsf{id}''\}$ and $r'$ otherwise. We will also include $r$ in the clear in $\mathsf{psi}_2$. Now the receiver can check decryption along which path (if any) yields $r$; in which case, the receiver can determine the intersection identity. To argue security, if we use anonymous garbled circuits [8], then we can argue if $\mathsf{id} \notin S_{\mathsf{R}}$, then $\mathsf{psi}_2$ is pseudorandom to the receiver. Arguing this formally (especially for the general case) is non-trivial, requiring a delicate formulation of hybrids.

*Receiver's security?* The receiver's hash $\mathsf{hr}_{\mathsf{root}}$ is computed deterministically from $S_{\mathsf{R}}$, so it cannot be secure. But this is easy to fix: On the leaf level we append the identities with random values, and only then will perform the Merkle hash.

## 2.2 Reusable Laconic PSI

We now outline our techniques for obtaining laconic PSI in a black-box way, for both semi-honest and malicious cases.

*A semi-honestly secure protocol* Our starting point is a recent construction of a *one-way function with encryption* from the $\phi$-hiding assumption due to Goyal, Vusirikala and Waters [27], and we remark that similar *accumulator-style* ideas were used before to construct PSI [2]. Since the protocol of [27] is "almost" a PSI protocol, we will directly describe the underlying semi-honestly secure PSI based. Assume for a moment that both the receiver's input $S_{\mathsf{R}}$ and the sender's input $S_{\mathsf{S}}$ are subsets of a polynomially-sized universe $\mathcal{U} = \{1, \ldots, \ell\}$. We will later remove this size-restriction on $\mathcal{U}$. We have a common reference string $\mathsf{crs}$ which is composed of an RSA modulus $N = PQ$, a uniformly random generator $g \in \mathbb{Z}_N^*$ and pairwise distinct primes $p_1, \dots, p_\ell$.

For the sake of simplicity, we will assume in this outline that the sender's input set $S_\mathsf{S}$ is a singleton set $\{w\} \subseteq \mathcal{U}$. The actual protocol will be obtained by running the protocol we will now sketch for every element in the sender's input set. The protocol commences as follows: The receiver first *hashes* its input set into

$$h = g^{r \prod_{i \in S_\mathsf{R}} p_i} \mod N,$$

where $r$ is chosen a uniformly chosen random from $[N]$ (and thus $r \mathrm{mod}\ \phi(N)$ is statistically close to uniform). The receiver then sends $h$ to the sender.

The sender, whose input is $S_\mathsf{S} = \{w\}$, chooses a uniformly random value $\rho \leftarrow_\$ [N]$ and a uniformly random seed $s$ for a suitable randomness extractor $\mathsf{Ext}$, and computes the values $f \leftarrow g^{\rho p_w}$ and $R \leftarrow \mathsf{Ext}(s, h^\rho)$. It sends $s$, $f$ and $R$ to the receiver.

The receiver, upon receiving $f$ and $R$, will check for all elements $i \in S_\mathsf{R}$ whether it holds that $R_i \overset{?}{=} R$, for $R_i \leftarrow \mathsf{Ext}(s, f^{r \cdot \prod_{j \in S_\mathsf{R} \setminus \{i\}} p_j})$. If it finds such an $i$, it outputs $\{i\}$ as the intersection of $S_\mathsf{R}$ and $S_\mathsf{S}$. Correctness of this protocol follows routinely[12]. by noting that if $w \in S_\mathsf{R}$ then

$$f^{r \cdot \prod_{j \in S_\mathsf{R} \setminus \{w\}} p_j} = g^{\rho \cdot r \cdot \prod_{j \in S_\mathsf{R}} p_j} = h^\rho.$$

Also, note that this scheme is laconic, as the size of the messages exchanged by the parties is independent of the size of the set $S_\mathsf{R}$.

Arguing security against a semi-honest sender is also routine, as $h$ is in fact statistically close to a uniformly random group element in $\mathbb{Z}_N^*$. Proving security against a semi-honest receiver is a bit more involved and proceeds via the following hybrid modifications. Let $S_\mathsf{S} = \{w\}$ be the sender's input such that $w \notin S_\mathsf{R}$. In the first hybrid, we will choose the modulus $N$ such that $p_w$ divides $\phi(N)$; under the $\phi$-hiding assumption this change will go unnoticed. Now, via a standard lossiness-argument, we have that $f = g^{\rho p_w}$ loses information about $g^\rho$, i.e., $g^\rho$ has high min-entropy given $f$. This means that $h^\rho = g^{\rho r \cdot \prod_{i \in S_\mathsf{R}} p_i}$ has also high min-entropy as $w \notin S_\mathsf{R}$ and thus $p_w$ does not divide $r \cdot \prod_{i \in S_\mathsf{R}} p_i$ (w.o.p). Consequently, as $h^\rho$ has high min-entropy conditioned on $f$, in the next hybrid change we can replace $R = \mathsf{Ext}(s, h^\rho)$ with a uniformly random value, incurring only a negligible statistical distance via the extraction property of $\mathsf{Ext}$. In the next hybrid change, we can switch the modulus $N$ back to normal mode, i.e., such that $p_w$ does not divide $\phi(N)$. But now $f = g^{\rho p_w}$ is statistically close to uniform in $\mathbb{Z}_N^*$. Thus, in the last hybrid change we can replace $f$ with a uniformly random value in $\mathbb{Z}_N^*$ and get that the view of the receiver is independent of $w$, as required.

For the case that the sender's input $S_\mathsf{S}$ contains more than a single element, we mount a hybrid argument repeating the above modifications for each element of $S_\mathsf{S}$ not in the receiver's set $S_\mathsf{R}$.

[12] We will not further discuss the small correctness-error of this protocol as our final protocol will not suffer from this defect

*Large universes* The above protocol has the drawback that the size of the common reference string $\mathsf{crs}$ depends linearly on the size of the universe $\mathcal{U}$, which is highly undesirable. There is a standard way of overcoming this issue: Instead of explicitly listing all the primes $p_i$ in $\mathsf{crs}$, we will describe them implicitly via a pseudorandom function (PRF).[13] For this purpose, we need a PRF which maps into the set of primes of a certain size. This can e.g. be achieved by using rejection sampling: we first sample $y \leftarrow F_k(x|i)$ (starting with $i = 1$) and check if $y$ is a prime number. If it is, we output $y$; else, we increment $i$ until a prime is hit. Under standard number-theoretic assumptions, this process finds a prime after a logarithmic number of steps. One small issue is that, in the above security proof, we need to replace one of the primes with a prime provided by the $\phi$-hiding experiment. We resolve this issue by making the PRF programmable in one point, e.g., by setting $F_{k,k'}(x|i) = F'_k(x|i) \oplus k_i$ for a PRF $F'$, $k' = (k_1, \ldots, k_\xi)$ and a suitable choice of $\xi$.

*A first attempt at malicious sender security* Our protocol thus far, however, offers no security against a malicious sender. The main issue is that a corrupted sender may choose the values $f$ and $R$ arbitrarily, and further, there is no mechanism for a simulator against a malicious sender to extract the senders input $w$. Of course, this protocol can be made secure against malicious senders by letting the sender prove via a general purpose NIZK proof that it follows the semi-honest protocol correctly. This however would necessitate to make non-black-box use of our semi-honest laconic PSI protocol, contrary to our goal of achieving a fully black-box protocol.

Re-inspecting the above protocol, we have not made full use of the fact that the extracted string $R$ is uniformly random. Our first idea to make the sender extractable is to make better use of $R$. Instead of sending $R$ in the plain, we will use $R$ as random coins for a public key encryption (PKE) scheme to encrypt the sender's input $w$. More concretely, we will modify the above protocol as follows. We include a public key $\mathsf{pk}$ of a PKE scheme in the common reference string $\mathsf{crs}$ and, instead of having the sender include $R$ in the plain in its message to the receiver, it will include a ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, i; R)$. We also need to modify the procedure of the receiver. The receiver will recover $R_i$ as before, but will now use $R_i$ to *re-encrypt* the index $i$, that is, for each $i \in S_{\mathsf{R}}$ it will compute $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, i; R_i)$.

First notice that, as a side bonus, this modification makes our laconic PSI scheme perfectly correct, given that the PKE scheme is perfectly correct, as now $\mathsf{ct}_i$ uniquely specifies the element $i$.

In terms of security, we first observe that this modification does not harm security against a semi-honest receiver given that the PKE scheme is IND-CPA secure. In the above sketch of a security proof, we have argued that, if $w$ is not in the set $S_{\mathsf{R}}$, then $R$ is uniformly random from the view of the receiver. This means now that $\mathsf{ct}$ is a freshly encrypted ciphertext, using fresh random

[13] We remark that we use a PRF, not because we want uniform outputs, but to implicitly define the set of primes. A similar trick was used in [6].

coins (independent of $\rho$). Moreover, we can use IND-CPA security of the PKE to replace $\mathsf{ct}$ with an encryption of 0, and then continue as above to argue security against a semi-honest receiver.

To establish security against a malicious sender, we would like to argue as follows. The simulator can now generate the public key $\mathsf{pk}$ in $\mathsf{crs}$ together with a secret key $\mathsf{sk}$. Given a message $(s, f, \mathsf{ct})$ by a malicious sender, the simulator can recover the set element $w$ by decrypting the ciphertext $\mathsf{ct}$ using $\mathsf{sk}$. At a first glance this seems to provide us with security against malicious senders. And indeed, the simulator will recover all elements for which the receiver would have declared to be in the intersection. There is a grave issue however: The simulator has no means of detecting whether the honest receiver would actually have succeeded in re-encrypting the index $i$. In other words, the malicious sender can make the simulator *false positives*, such that the simulator declares an element $i$ to be in the intersection, whereas an honest receiver would not have.

*Switch groups, extract everything!* We briefly recall some facts about the Damgård-Jurik cryptosystem [15]. The group $\mathbb{Z}^*_{N^{\xi+1}}$ contains a cyclic subgroup $\mathbb{NR}_N$ of order $\phi(N)$[14]. Now let $g_0 \in \mathbb{NR}_N$ be a generator of $\mathbb{NR}_N$. Then we can generate the entire group $\mathbb{Z}^*_{N^{\xi+1}}$ by $g_0$ and $1 + N$, i.e. we can write every $h \in \mathbb{Z}^*_{N^{\xi+1}}$ as $h = g_0^t \cdot (1 + N)^m$ for some $t \in \mathbb{Z}_{\phi(N)}$ and $m \in \mathbb{Z}_{N^\xi}$. Furthermore, we can efficiently compute discrete logarithms relative to $1+N$, i.e. if $h = (1+N)^m$ for an $m \in \mathbb{Z}_{N^\xi}$, then we can efficiently compute $m$ from $h$. Finally, the decisional composite residue (DCR) assumption in $\mathbb{Z}^*_{N^{\xi+1}}$ states that a random element in $\mathbb{NR}_N$ is indistinguishable from a random element in $\mathbb{Z}^*_{N^{\xi+1}}$. It follows that $g_1 = g_0^{t_1}$ and $g_2 = g_0^{t_2} \cdot (1 + N)$ (for uniformly random $t_1, t_2 \leftarrow_\$ \mathbb{Z}_{\phi(N)}$) are computationally indistinguishable. Moreover, if $h = g_2^t$ for a $t < N^{\xi-1}$, we can efficiently compute $t$ from $h$ using $\phi(N)$ as a trapdoor by first computing

$$h^{\phi(N)} = g_2^{t\cdot\phi(N)} = \underbrace{g_0^{t\phi(N)}}_{=1} \cdot (1+N)^{t\cdot\phi(N)} = (1+N)^{t\cdot\phi(N)} \mod N^{\xi+1},$$

from which we can efficiently compute $t \cdot \phi(N)$ (as $t \cdot \phi(N) < N^\xi$) and thus $t$.

Given this, we will now make the following additional modification to our PSI protocol. Instead of choosing the element $g$ in the common reference string $\mathsf{crs}$ to be a random generator of $\mathbb{Z}^*_N$, we choose $g$ to be a random generator of $\mathbb{NR}_N$, where $\mathbb{NR}_N$ is the subgroup of order $\phi(N)$ in $\mathbb{Z}^*_{N^{\xi+1}}$ (for a sufficiently large but constant $\xi$). Our first observation is that this does not affect the security proof in the case of a semi-honest receiver, since $\mathbb{NR}_N$ is still a cyclic group of order $\phi(N)$ and the above argument using the $\phi$-hiding assumption works analogously in this group.

[14] Note that $\mathbb{NR}_N$ is not a cyclic group and we only assume this here for simplicity. Actually, if we choose $N$ as a product of two safe primes, then we could find a cyclic subgroup $\mathbb{J}_N$ which is the group of elements with Jacobi symbol 1, and its subgroup $\mathbb{T}_N$ composing of $N^\xi$-th powers of $\mathbb{J}_N$ has order $\phi(N)/2$. Namely, just replace the group pair $(\mathbb{Z}^*_{N^{\xi+1}}, \mathbb{NR}_N)$ with $(\mathbb{J}_N, \mathbb{T}_N)$ to fix this issue. Please refer to Section 3.1 and Section 6 for details.

Assume for a moment we had a mechanism which ensures that the group element $f$ in the sender's message is of the form $f = g^a$ for an $a < N^{\xi-1}$. We can then argue security against a malicious sender as follows: First we make a hybrid change and choose the element $g$ in the common reference string like $g_2$ above, i.e. we choose $g = g_0^t(1+N)$; under the DCR assumption this change goes unnoticed. Now, given that $f = g^a$ for an $a < N^{\xi-1}$ and using $\phi(N)$ as a trapdoor, the simulator can efficiently compute $a$ from $f$ as described above. Since it can also recover the index $w$ from the ciphertext $\mathsf{ct}$ as described above, it can now check if $a$ is of the form $a = \rho \cdot p_w$. If so, it recovers $\rho$ and performs the same re-encryption test for $\mathsf{ct}$ which the real receiver would perform. This makes the simulation indistinguishable from the real experiment.

## 2.3 DV-NIZK Range Proofs for DJ Ciphertexts

The final component which is missing to make the above argument succeed is a mechanism which ensures that the group element $f$ is indeed of the form $f = g^a$ for a *small* $a$. For the sake of generality, we will make the following discussion for general DJ-ciphertexts, that is, ciphertexts of the form $c = h^t \cdot (1+N)^a$ (where $h = g_1^z$ is the public key). If we can show that such a ciphertext encrypts a small value $a$, proving that $f = g^a$ and $c = h^t \cdot (1+N)^a$ for the same $a$ can be efficiently proven via a standard hash-proof system (HPS) [14].

First, we observe that, to show that $c = h^t \cdot (1+N)^a$ encrypts a value $a < 2^k$ for some parameter $k$, it suffices to prove that some ciphertexts $c_0, \ldots, c_{k-1}$ encrypt *bits* $b_1, \ldots, b_{k-1}$. Assume for now we had a DV-NIZK protocol $\Pi$ to prove that the ciphertexts $c_0, \ldots, c_{k-1}$ all just encrypt bits. The prover can convince the verifier as follows that $c$ encrypts a value $a < 2^k$. First the prover encrypts bit $b_i$ in a ciphertext $c_i$ and sets $c' = \prod_{i=0}^{k-1} c_i^{2^i}$ (it is not hard to see that $c'$ encrypts $a$). Now, the prover uses $\Pi$ to to convince the verifier that $c_0, \ldots, c_{k-1}$ indeed encrypt bits. Furthermore, it can use a standard HPS to prove that $c$ and $c'$ indeed encrypt the same value. Zero-knowledge follows routinely. To see that this protocol is sound, observe that if the $c_i$ indeed encrypt bits, then $c'$ must encrypt a value bounded by $2^k$.

*A DV-NIZK proof system for ciphertext equality across different encryption schemes* Alas, we do not know of a black-box DV-NIZK which proves that DJ ciphertexts encrypt bits. However, for the pairing-based Boneh-Goh-Nissim (BGN) cryptosystem [5], such a proof system was constructed by Groth, Ostrovsky and Sahai [29]. Consequently, if we could prove in a black box way that a BGN ciphertext encrypts the same value as DJ ciphertext we would be done.

Recall that, in the BGN cryptosystem, public keys are of the form $(G, H)$, where $G$ and $H$ a generators of subgroups of a composite-order pairing group $\mathbb{G}$. BGN ciphertexts are of the form $C = G^m H^r$, where $m$ is the encrypted message and $r$ are random coins.

Our final contribution is a DV-NIZK proof system which allows us to prove that a DJ ciphertext and a BGN ciphertext encrypt the same value.

To simplify the description of our prove system, assume we have BGN public keys $(G, H_1), \ldots, (G, H_\ell)$, i.e. each key sharing the same $G$ but having fresh and random $H_i$, and an element $H_0$. Furthermore, assume that we have DJ public keys $h_1, \ldots, h_\ell$, and an element $h_0$. We will assume that both sequences of keys are in a public setup, together with the elements $H_0, h_0$.

Suppose further that we have BGN ciphertexts $C_1, \ldots, C_\ell$, where $C_i = G^{m_i} H_i^r$, i.e., all ciphertexts use the same random coins $r$ but encrypt possibly different bits $m_i$.[15] As mentioned above, using the NIZK scheme from [29], we can prove that the ciphertexts $C_i = G^{m_i} H_i^r$ are indeed well-formed and that $m_i \in \{0, 1\}$. Moreover, we have $C_0 = H_0^r$, which can be proven well-formed using a standard hash proof system (HPS) [14].

Assume further that we are given DJ ciphertexts $c_1, \ldots, c_\ell$, where $c_i = h_i^t \cdot (1+N)^{m'_i}$, i.e., again the ciphertexts share the same random coins $t$.[16] Moreover, assume that we have a value $h_0^r$ exactly as above. We want to prove that it holds for all $i \in [\ell]$ that $m_i = m'_i$. Our DV-NIZK proof system for equality of BGN and DJ ciphertexts now proceeds roughly as follows:

– The verifier starts by sampling a uniformly random binary string $\sigma \leftarrow_{\$} \{0, 1\}^\ell$ and computes $F = H_0^A \prod H_i^{\sigma_i} \in \mathbb{G}$ and $f = h_0^\alpha \prod h_i^{\sigma_i} \in Z^*_{N^{\xi+1}}$, for uniformly random values $A, \alpha$. It sends $\mathsf{crs} = (F, f)$ to the prover and keeps $\sigma$ as the designated-verifier key.
– The prover is given ciphertexts $C_1, \ldots, C_\ell$ and $c_1, \ldots, c_\ell$ with $C_i = G^{m^i} H_i^r$ and $c_i = h_i^t (1 + N)^{m_i}$, and the values $C_0 = H_0^r$ and $c_0 = h_0^t$. It computes $K = F^r G^\tau$ and $k = f^t (1+N)^\tau$ where $\tau$ is sampled according to a distribution which is wide enough to drown the $m_i$, but short enough such that it is bounded by $N$. The proof $\pi$ is consists of $(K, k)$.
– The verifier, given the proof $\pi = (K, k)$, computes the discrete log $y$ (in base $(1 + N)$) of $k^{-1} c_0^\alpha \prod_{i=1}^\ell c_i^{\sigma_i}$ and checks if $G^y = K^{-1} C_0^A \prod_{i=1}^\ell C_i^{\sigma_i}$.

For completeness, note that

$$\begin{aligned} k^{-1} c_0^\alpha \prod c_i^{\sigma_i} &= \left(h_0^\alpha \prod h_i^{\sigma_i}\right)^{-t} (1 + N)^{-\tau} \left(h_0^t\right)^\alpha \prod \left(h_i^t (1 + N)^{m_i}\right)^{\sigma_i} \\ &= (1 + N)^{\sum \sigma_i m_i - \tau}, \end{aligned}$$

from which the verifier can recover $y = \sum \sigma_i m_i - \tau$. Moreover

$$L = K^{-1} C_0^A \prod C_i^{\sigma_i} = \left(H_0^A \prod H_i^{\sigma_i}\right)^{-r} G^{-\tau} \left(H_0^r\right)^A \prod (H_i^r G^{m_i})^{\sigma_i} = G^{\sum \sigma_i m_i - \tau}$$

and thus $G^y = L$.

The zero-knowledge property can be established by noting that the term $\tau$ statistically drowns $\sum_i \sigma_i m_i$.

To prove *reusable statistical soundness* (or simulation soundness), we argue as follows. First note that $\sigma$ is statistically hidden, given $F = H_0^A \prod H_i^{\sigma_i}$ and $f =$

[15] Via a standard rerandomization argument we can show that reusing the same random coins across different keys does not harm CPA security.
[16] Same as above.

$h_0^\alpha \prod h_i^{\sigma_i}$, by the uniform values $A, \alpha$. We need to show that if there is an index $i$ for which $m_i \neq m_i'$, then the verifier will reject with high probability, irrespective of the (adversarial) choices of $\tau, \tau'$ (which are not necessarily short)[17]. It follows from the above description that the verifier accepts a proof if the condition

$$\sum \sigma_{i,j} m_i - \tau_j \mod n = \left(\sum \sigma_{i,j} m_i' - \tau_j' \bmod N^\xi\right) \bmod n$$

is satisfied, where $n$ is the order of the subgroup of $\mathbb{G}$ generated by $G$. In the main body we will show that, given that $n > N^\xi$, this condition will be violated with probability $\approx 1/2$ if the there exists an index $i$ for which $m_i \neq m_i'$. By repeating the protocol $\lambda$ times, we achieve negligible soundness error.

### 2.4 Labeled Laconic PSI and Laconic OT

Our laconic PSI construction can be easily extended into a labeled laconic PSI, in which the receiver also learns labels associated with set elements in the intersection. To achieve this, we simply use an extractor with an output size twice as large: the first half is used as above to perform the re-encryption step; the other half is used as an one-time pad to encrypt the corresponding label. It is easy to see that the receiver can only recover the labels for the elements within the intersection, since the security proof follows the same blueprint as before.

We also build an LOT using the same ideas as above. The receiver commits to a database $D \in \{0,1\}^\Gamma$ by computing $h = g_0^{r \prod_{i=1}^{\Gamma} e_{i,D_i}} \bmod N^{\xi+1}$, where each prime $e_{i,b}$ is the output of a PRF (just as before). The sender computes $f_j = g_0^{\rho_j e_{L,j}}, F_j = g_1^{\rho_j e_{L,j}}(1+N)^{\rho_j e_{L,j}}$ for each $j \in \{0,1\}$, together with a range proof. Moreover, he encrypts each message as $\mathsf{ct}_j = k_j \oplus m_j$ where $k_j \leftarrow \mathsf{Ext}(s_j, h^{\rho_j})$. Again, security follows the same reasoning as above. Our LOT protocol is the first one to provide security against a malicious sender while incurring in communication complexity independent of the size of $D$.

## 3 Preliminaries

We use $\lambda$ to denote the security parameter. By $\mathsf{negl}(\lambda)$, we denote a negligible function in $\lambda$. For an integer $n$, $[n]$ denotes $\{1, \ldots, n\}$. If $\mathcal{A}$ is an algorithm, we denote by $y \leftarrow \mathcal{A}(x)$ the output $y$ after running $\mathcal{A}$ on input $x$. For a $S$, $x \leftarrow_\$ S$ denotes sampling $x$ uniformly at random from $S$. If $D$ is a distribution, then $x \leftarrow_\$ D$ denotes sampling $x$ according to $D$, and $y \in D$ indicates $y$ is in the support of $D$. We say that $D$ is $B$-bounded if for every $x \leftarrow_\$ D$, we have $|x| < B$, except with negligible probability. If $D_0, D_1$ are two distributions, we use $D_0 \approx_\varepsilon D_1$ to indicate that $D_0$ and $D_1$ are statistically indistinguishable. Throughout this work, $\phi$ will denote the Euler's totient function.

[17] We assume that the verifier rejects if it fails to compute the discrete logarithm of $k^{-1} \prod d_i^{\sigma_i}$.

In terms of cryptographic primitives we need public-key encryption (PKE), designated-verifier zero-knowledge (DV-NIZK) proof systems, programmable pseudorandom functions (PPRF) [36] and strong randomness extractors. The hardness assumptions used in this work are the $\phi$-hiding, decisional composite residuosity (DCR), subgroup decision (SD), computational Diffie-Hellman (CDH) and learning with errors (LWE). Apart from $\phi$-hiding and DCR assumptions we described below, other primitives/assumptions are reviewed in the full version of this paper [1].

## 3.1 Hardness Assumptions

We start by introducing some notation. Let $\mathsf{Primes}(\kappa)$ denote the set of prime numbers of bit-length $\kappa$. Let

$$\mathsf{RSA}(\lambda) = \{N : N = PQ \text{ and } P, Q \in \mathsf{Primes}(\lambda/2) \text{ and } \gcd(P-1, Q-1) = 2\}$$

and

$$\mathsf{RSA}_e(\lambda) = \{N : e|\phi(N)\}$$

for any $e \leq 2^\lambda$.

**Definition 1 (Phi-Hiding)** *The* phi-hiding *assumption, denoted as $\phi$-hiding, states that for all $\varepsilon > 0$ and $3 < e < 2^{\lambda/4-\varepsilon}$ and all PPT adversaries $\mathcal{A}$, we have that*

$$|\Pr[1 \leftarrow \mathcal{A}(N,e) : N \leftarrow_\$ \mathsf{RSA}(\lambda)] - \Pr[1 \leftarrow \mathcal{A}(N,e) : N \leftarrow_\$ \mathsf{RSA}_e(\lambda)]| \leq \mathsf{negl}(\lambda)\,.$$

Let $N = PQ$ where $P, Q$ are safe primes (that is, $P = 2p' + 1$ and $Q = 2q' + 1$ for primes $p', q'$) and consider the multiplicative group $\mathbb{Z}^*_{N^{\xi+1}}$ where $\xi$ is a fixed non-negative integer. Recall that $\mathbb{Z}^*_{N^{\xi+1}}$ can be written as the product of two subgroups $\mathbb{H}_N \times \mathbb{NR}_N$ where $\mathbb{H}_N = \{(1+N)^i : i \in [N^\xi]\}$ and $\mathbb{NR}_N = \{x^{N^\xi} : x \in \mathbb{Z}^*_{N^{\xi+1}}\}$ (the subgroup of $N^\xi$-residues) which has order $\phi(N)$. Given $(1+N)^m \bmod N^{\xi+1}$, there is a polynomial-time algorithm that allows to recover $m$ [15].

Furthermore, let $\mathbb{J}_N$ be the group of elements with Jacobi symbol 1, i.e., $\mathbb{J}_N = \{x : (x|N) = 1, x \in \mathbb{Z}^*_{N^{\xi+1}}\}$. Note that $\mathbb{J}_N$ can be written as the direct product of two cyclic groups $\mathbb{H}_N \times \mathbb{T}_N$ with order $N^\xi$ and $\phi(N)/2$, respectively. Also, the subset membership problem for $(\mathbb{J}_N, \mathbb{T}_N)$ is still hard if DCR assumption holds as shown in Section 8.2 of [14].

The following lemma is straightforwardly adapted from [27].

**Lemma 1 ([27])** *Assume that the $\phi$-hiding assumption holds. Let $\mathsf{Ext}$ be a $(\kappa - 1, \mathsf{negl}(\lambda))$-strong extractor. For every admissible stateful PPT adversary $\mathcal{A}$ and for all $\lambda, \kappa$ such that $\lambda \geq 5\kappa$, we have that*

$$\left| \Pr\left[ b \leftarrow \mathcal{A}(y_b) : \begin{array}{c} N \leftarrow_\$ \mathsf{RSA}(\lambda);\ s \leftarrow_\$ \{0,1\}^\lambda \\ e \leftarrow_\$ \mathsf{Primes}(\kappa);\ g \leftarrow \mathbb{T}_N \\ G \leftarrow \mathcal{A}(N, s, e, g);\ b \leftarrow_\$ \{0,1\} \\ y_0 \leftarrow \mathsf{Ext}(s, g^{Ge^{-1}} \bmod N^{\xi+1});\ y_1 \leftarrow_\$ \mathcal{Y} \end{array} \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)$$

*where an admissible adversary is one that outputs $G$ such that $e$ does not divide $G$.*

In this work, we also make use of the DCR assumption which we define in the following. We present the DCR assumption as a subgroup indistinguishability assumption [7].

**Definition 2 (Decisional Composite Residuosity)** *Let $N = \mathsf{RSA}(\lambda)$ and let $\xi \geq 0$ be a fixed integer. The* decisional composite residuosity *(DCR) assumption states that for all PPT adversaries $\mathcal{A}$,*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}(N,x) : x \leftarrow_{\$} \mathbb{Z}^*_{N^{\xi+1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}(N,x) : x \leftarrow_{\$} \mathbb{NR}_N\right]\right| \leq \mathsf{negl}(\lambda)\,.$$

**Lemma 2 ([14])** *$N = \mathsf{RSA}(\lambda)$ and let $\xi \geq 0$ be a fixed integer. Assume that the DCR assumption holds. Then for all PPT adversaries $\mathcal{A}$,*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}(N,x) : x \leftarrow_{\$} \mathbb{J}_N\right] - \Pr\left[1 \leftarrow \mathcal{A}(N,x) : x \leftarrow_{\$} \mathbb{T}_N\right]\right| \leq \mathsf{negl}(\lambda)\,.$$

*Proof (Proof (sketch)).*

The proof follows from the following observation: The map $x \to x^2(-1)^b$ where $b \leftarrow_{\$} \{0,1\}$ sends the uniform distribution on $\mathbb{NR}_N$ to the uniform distribution on $\mathbb{T}_N$, and the uniform distribution on $\mathbb{H}_N \times \mathbb{NR}_N$ to the uniform distribution on $\mathbb{H}_N \times \mathbb{T}_N$.

### 3.2 Laconic Private Set Intersection

*Laconic Private Set Intersection.* An $\ell$PSI is a two-round protocol that implements a PSI functionality and has special compactness properties.

**Definition 3** *A $\ell$PSI scheme $\mathsf{LPSI} = (\mathsf{GenCRS}, \mathsf{R}_1, \mathsf{S}, \mathsf{R}_2)$ is defined as follows:*

- $\mathsf{GenCRS}(1^\lambda)$*: Takes as input a security parameter $1^\lambda$, and outputs a common reference string* $\mathsf{crs}$.
- $\mathsf{R}_1(\mathsf{crs}, S_R)$*: Takes as input a* $\mathsf{crs}$ *and a set $S_R$. It outputs a first PSI message* $\mathsf{psi}_1$ *and a state* $\mathsf{st}$.
- $\mathsf{S}(\mathsf{crs}, S_S, \mathsf{psi}_1)$*: Takes as input a* $\mathsf{crs}$*, a set $S_S$ and a first PSI message* $\mathsf{psi}_1$*. It outputs a second PSI message* $\mathsf{psi}_2$.
- $\mathsf{R}_2(\mathsf{crs}, \mathsf{st}, \mathsf{psi}_2)$*: Takes as input a* $\mathsf{crs}$*, a state* $\mathsf{st}$ *and a second message* $\mathsf{psi}_2$*. It outputs a set $\mathcal{I}$.*

*We require the following properties.*

- ***Correctness****: The protocol satisfies PSI correctness in the standard sense.*
- ***Efficiency Requirements.*** *There exists a fixed polynomial* $\mathsf{poly}$ *such that the length of* $\mathsf{psi}_1$ *and the running time of* $\mathsf{S}$ *are at most* $\mathsf{poly}(\lambda, \log|S_R|)$.

For malicious security, we work in the standard UC-framework [10] that allows us to prove security of protocols under arbitrary composition with other protocols. Let $\mathcal{F}$ be a functionality, $\pi$ a protocol that implements $\mathcal{F}$ and $\mathcal{E}$ be a environment, an entity that oversees the execution of the protocol in both the real and the ideal worlds. Let $\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{E}}$ be a random variable that represents the output of $\mathcal{E}$ after the execution of $\mathcal{F}$ with adversary $\mathsf{Sim}$. Similarly, let $\mathsf{REAL}^{\mathcal{G}}_{\pi,\mathcal{A},\mathcal{E}}$ be a random variable that represents the output of $\mathcal{E}$ after the execution of $\pi$ with adversary $\mathcal{A}$ and with access to the functionality $\mathcal{G}$.

**Definition 4** *A protocol* $\pi$ UC-realizes $\mathcal{F}$ in the $\mathcal{G}$-hybrid model *if for every PPT adversary* $\mathcal{A}$ *there is a PPT simulator* $\mathsf{Sim}$ *such that for all PPT environments* $\mathcal{E}$*, the distributions* $\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{E}}$ *and* $\mathsf{REAL}^{\mathcal{G}}_{\pi,\mathcal{A},\mathcal{E}}$ *are computationally indistinguishable.*

We present the (reusable) PSI ideal functionality.

*Reusable PSI functionality.* The functionality $\mathcal{F}_{\mathsf{rPSI}}$ is parametrized by a universe $\mathcal{U}$ and works as follows:

– **Setup phase.** $\mathsf{R}$ sends $(\mathsf{sid}, S_{\mathsf{R}})$ to $\mathcal{F}_{\mathsf{rPSI}}$ where $S_{\mathsf{R}} \subseteq \mathcal{U}$. It ignores future messages from $\mathsf{R}$ with the same $\mathsf{sid}$.
– **Send phase.** $\mathsf{S}$ sends $(\mathsf{sid}, i, S_{\mathsf{S}} \subseteq \mathcal{U})$ to $\mathcal{F}_{\mathsf{rPSI}}$. $\mathcal{F}_{\mathsf{rPSI}}$ sends $(\mathsf{sid}, i, S_{\mathsf{R}} \cap S_{\mathsf{S}})$ to $\mathsf{R}$. It ignores future messages from $\mathsf{S}$ with the same $\mathsf{sid}$ and $i \in \mathbb{N}$.

## 4 Semi-Honest Laconic PSI from CDH/LWE

We show how to realize semi-honest $\ell$PSI from CDH/LWE. Our construction is non-black-box, making use of garbled circuits. This leads to the first feasibility result based on CDH, and an alternative LWE construction to that of [40].

Our construction makes use of hash encryption schemes in conjunction with garbled circuits, which we review below.

**Definition 5 (Hash Encryption [17,8])** *A hash encryption scheme* $\mathsf{HE} = (\mathsf{HGen}, \mathsf{Hash}, \mathsf{HEnc}, \mathsf{HDec})$ *is defined as follows.*

– $\mathsf{HGen}(1^\lambda, n)$*: Takes as input a security parameter* $1^\lambda$ *and an input size* $n$ *and outputs a hash key* $\mathsf{pp}$*.*
– $\mathsf{Hash}(\mathsf{pp}, z)$*: Takes as input a hash key* $\mathsf{pp}$ *and* $z \in \{0,1\}^n$*, and deterministically outputs* $h \in \{0,1\}^\lambda$*.*
– $\mathsf{HEnc}(\mathsf{pp}, h, \{m_{i,b}\}_{i\in[n],b\in\{0,1\}}; \{r_{i,b}\})$*: Takes as input a hash key* $\mathsf{pp}$*, a hash output* $h$*, messages* $\{m_{i,b}\}$ *and randomness* $\{r_{i,b}\}$*, and outputs* $\{\mathsf{cth}_{i,b}\}_{i\in[n],b\in\{0,1\}}$*. We write it shortly as* $\{\mathsf{cth}_{i,b}\}$*. Overloading notation, each ciphertext* $\mathsf{cth}_{i,b}$ *is computed as* $\mathsf{cth}_{i,b} = \mathsf{HEnc}(\mathsf{pp}, h, m_{i,b}, (i,b); r_{i,b})$*.*
– $\mathsf{HDec}(z, \{\mathsf{cth}_{i,b}\})$*: Takes as input a hash input* $z$ *and* $\{\mathsf{cth}_{i,b}\}$ *and outputs* $n$ *messages* $(m_1, \ldots, m_n)$*.*

*We require correctness meaning that for the variables above,* $(m_1, \ldots, m_n) = (m_{1,z[1]}, \ldots, m_{n,z[n]})$*. We define two notions of security.*

– ***Semantic Security***: *Given $z \in \{0,1\}^n$, no adversary can distinguish between encryptions of messages made to indices $(i, \bar{z}_i)$. For any PPT $\mathcal{A}$, sampling* $\mathsf{pp} \leftarrow_{\$} \mathsf{HGen}(1^\lambda, n)$, *if* $(z, \{m_{i,b}\}, \{m'_{i,b}\}) \leftarrow_{\$} \mathcal{A}(\mathsf{pp})$ *and if* $m_{i,z[i]} = m'_{i,z[i]}$ *for all* $i \in [n]$, *then* $\mathcal{A}$ *cannot distinguish between* $\mathsf{HEnc}(\mathsf{pp}, h, \{m_{i,b}\})$ *and* $\mathsf{HEnc}(\mathsf{pp}, y, \{m'_{i,b}\})$, *where* $h := \mathsf{Hash}(\mathsf{pp}, z)$.
– ***Anonymous Semantic Security***: *For a random* $\{m_{i,b}\}$ *with equal rows (i.e.,* $m_{i,0} = m_{i,1}$*), the output of* $\mathsf{HEnc}(\mathsf{pp}, h, \{m_{i,b}\})$ *is pseudorandom even in the presence of the hash input. Formally, for any* $z \in \{0,1\}^n$, *sampling* $\mathsf{pp} \leftarrow_{\$} \mathsf{HGen}(1^\lambda, n)$, $h := \mathsf{Hash}(\mathsf{pp}, z)$, *and sampling* $\{m_{i,b}\}$ *uniformly at random with the same rows, then* $\boldsymbol{v} := (\mathsf{pp}, z, \mathsf{HEnc}(\mathsf{pp}, h, \{m_{i,b}\}))$ *is indistinguishable from another tuple in which we replace the hash-encryption component of* $\boldsymbol{v}$ *with a random string.*

We have the following results from [8,21].

**Lemma 3** *Assuming CDH/LWE there exists anonymous hash encryption schemes, where* $n = 3\lambda$ *(i.e.,* $\mathsf{Hash}(\mathsf{pp}, \cdot)\colon \{0,1\}^{3\lambda} \mapsto \{0,1\}^\lambda$*).*[18] *Moreover, the hash function* $\mathsf{Hash}$ *satisfies robustness in the following sense: for any input distribution on* $z$ *which samples at least* $2\lambda$ *bits of* $z$ *uniformly at random,* $(\mathsf{pp}, \mathsf{Hash}(\mathsf{pp}, z))$ *and* $(\mathsf{pp}, u)$ *are statistically close, where* $\mathsf{pp} \leftarrow_{\$} \mathsf{HGen}(1^\lambda, 3\lambda)$ *and* $u \leftarrow_{\$} \{0,1\}^\lambda$.

We also review the notion of garbled circuits and the anonymous property, as defined in [8].

**Definition 6 (Garbled Circuits)** *A garbling scheme for a class of circuits* $\{\mathsf{C}\colon \{0,1\}^n \mapsto \{0,1\}^m\}$ *consists of* $(\mathsf{Garb}, \mathsf{Eval}, \mathsf{Sim})$ *satisfying the following.*

– ***Correctness***: *for all* $\mathsf{C} \in \mathcal{C}$, $\mathsf{m} \in \{0,1\}^n$, $\Pr[\mathsf{Eval}(\tilde{\mathsf{C}}, \{\mathsf{lb}_{i,\mathsf{m}[i]}\}) = \mathsf{C}(\mathsf{m})] = 1$, *where* $(\widetilde{\mathsf{C}}, \{\mathsf{lb}_{i,b}\}) \leftarrow_{\$} \mathsf{Garb}(1^\lambda, \mathsf{C})$.
– ***Simulation Security***: *For any* $\mathsf{C} \in \mathcal{C}$ *and* $\mathsf{m} \in \{0,1\}^n$: $(\tilde{\mathsf{C}}, \{\mathsf{lb}_{i,\mathsf{m}[i]}\}) \overset{c}{\equiv} \mathsf{Sim}(1^\lambda, \mathsf{C}(\mathsf{m}))$, *where* $(\widetilde{\mathsf{C}}, \{\mathsf{lb}_{i,b}\}) \leftarrow_{\$} \mathsf{Garb}(1^\lambda, \mathsf{C})$.
– ***Anonymous Security*** *[8]: For any* $\mathsf{C} \in \mathcal{C}$, *choosing* $y \leftarrow_{\$} \{0,1\}^m$, *the output of* $\mathsf{Sim}(1^\lambda, y)$ *is pseudorandom.*

**Lemma 4 ([8])** *Anonymous garbled circuits can be built from one-way functions (OWFs).*

*Notation on Hash Encryption.* Throughout this section we assume $\mathsf{Hash}(\mathsf{pp}, \cdot)\colon \{0,1\}^n \mapsto \{0,1\}^\lambda$, where $n = 3\lambda$. We use $\{\mathsf{lb}_{i,b}\}$ to define a sequence of pairs of labels, where (throughout this section) $i \in [n]$ and $b \in \{0,1\}$. For $\boldsymbol{r} := \{r_{i,b}\}$ we let $\mathsf{HEnc}(\mathsf{pp}, h, \{\mathsf{lb}_{i,b}\}; \boldsymbol{r})$ denote the ciphertexts $\{\mathsf{cth}_{i,b}\}$, where $\mathsf{cth}_{i,b} = \mathsf{HEnc}(\mathsf{pp}, h, \mathsf{lb}_{i,b}, (i,b); r_{i,b})$. We further overload the notation as follows. We use $\{\mathsf{lb}_i\}$ to denote a sequence of $3\lambda$ elements. For $\boldsymbol{r} := \{r_{i,b}\}$ we

[18] We note that the CDH construction of [8] satisfies a weaker notion of anonymity, in which only some part of the ciphertext is pseudorandom. But for ease of presentation we keep the notion as is, and remark that our $\ell$PSI construction works also with respect to that weaker notion.

| **Circuit** $\mathsf{F}[\mathsf{id}, r, r'](\mathsf{id}', x, x')$**:** | **Circuit** $\mathsf{V}[\mathsf{pp}, \mathsf{id}, \{\mathsf{lb}_{i,b}\}, \boldsymbol{r}](h_1, h_2, \mathsf{id}')$**:** |
|---|---|
| – **Hardwired:** target identity $\mathsf{id}$ and randomness values $r$ and $r'$.<br>– **Operation:** Return $y := \begin{cases} r & \mathsf{id} = \mathsf{id}' \\ r' & \text{else} \end{cases}$ | – **Hardwired:** Hash public parameter $\mathsf{pp}$, target identity $\mathsf{id}$, labels $\{\mathsf{lb}_{i,b}\}$, randomness $\boldsymbol{r}$.<br>– **Operation:** Return $\mathsf{ct} := \begin{cases} \mathsf{HEnc}(\mathsf{pp}, h_1, \{\mathsf{lb}_{i,b}\}; \boldsymbol{r}) & \mathsf{id} \leq \mathsf{id}' \\ \mathsf{HEnc}(\mathsf{pp}, h_2, \{\mathsf{lb}_{i,b}\}; \boldsymbol{r}) & \text{else} \end{cases}$ |

**Procedure** $\mathsf{DecPath}(\mathsf{pth}, \mathsf{psi}_2)$**:**

– **Input:** A leaf-root Path $\mathsf{pth}$ and ciphertext $\mathsf{psi}_2 := (\widetilde{\mathsf{C}}_0, \ldots, \widetilde{\mathsf{C}}_d, \{\mathsf{cth}^{(d)}_{i,b}\})$.

– **Operation:** Parse $\mathsf{pth} := ((\underbrace{\mathsf{id}, x, x'}_{z_0}), (\underbrace{h_0, h'_0, \mathsf{id}_0}_{z_1}), \ldots, (\underbrace{h_{d-1}, h'_{d-1}, \mathsf{id}_{d-1}}_{z_d}), \mathsf{hr}_{\mathsf{root}})$. For $w \in \{d, \ldots, 1\}$:

```
  1. Let $\{\mathsf{lb}^{(w)}_i\} := \mathsf{HDec}(\mathsf{z}_w, \{\mathsf{cth}^{(w)}_{i,b}\})$.
  2. Set $\{\mathsf{cth}^{(w-1)}_{i,b}\} := \mathsf{Eval}(\widetilde{\mathsf{C}}_w, \{\mathsf{lb}^{(w)}_i\})$.
Let $\{\mathsf{lb}^{(0)}_i\} := \mathsf{HDec}(z_0, \{\mathsf{cth}^{(0)}_{i,b}\})$. Return $\mathsf{Eval}(\widetilde{\mathsf{C}}_0, \{\mathsf{lb}^{(0)}_i\})$.
```

**Table 1.** Circuits $\mathsf{F}, \mathsf{V}$ and procedure $\mathsf{DecPath}$

let $\mathsf{HEnc}(\mathsf{pp}, h, \{\mathsf{lb}_i\}; \boldsymbol{r})$ denote a hash encryption where both plaintext rows are $\{\mathsf{lb}_i\}$; namely, the ciphertexts $\{\mathsf{cth}_{i,b}\}$, where $\mathsf{cth}_{i,b} = \mathsf{HEnc}(\mathsf{pp}, h, \{m_{i,b}\}; r_{i,b})$, where $m_{i,0} = m_{i,1} = \mathsf{lb}_i$, for all $i$.

*Tree Terminology.* Throughout this section we work with full binary trees. The depth of a tree is the length of a root-leaf path. We call the leaf level level 0, the level above it level one, and so on. We order the root-leaf paths from left to right; namely, the path from the root to the leftmost leaf node is the first root-leaf path, and the path from the root to the rightmost leaf node is the $2^d$th root-leaf path, where $d$ is the depth. Each node has an associated hash value, computed based on values associated to its children. Thus, when representing a root-leaf path, we include both children of each branching intermediate node.

*Sender's Set Size is One.* We assume without loss of generality that the sender holds a single element. For the general case where the sender may have multiple elements, we reuse the first message of the receiver for each element in the sender's set. The overall running time of the sender will only scale with its own set size, and not with the receiver's set size.

**Construction 1 ($\ell$PSI Construction)** *We require the following ingredients in our $\ell$PSI Construction.*

1. *A hash encryption scheme* $\mathsf{HE} = (\mathsf{HGen}, \mathsf{Hash}, \mathsf{HEnc}, \mathsf{HDec})$*, where* $\mathsf{Hash}(\mathsf{pp}, \cdot)\colon \{0,1\}^{3\lambda} \mapsto \{0,1\}^{\lambda}$.

2. *A garbling scheme* $\mathsf{GS} = (\mathsf{Garb}, \mathsf{Eval}, \mathsf{Sim})$.
3. *Circuits* $\mathsf{F}$ *and* $\mathsf{V}$*, as well as procedure* $\mathsf{DecPath}$*, defined in Table 1.*

*We assume the elements of the receiver and the sender are strings in* $\{0,1\}^\lambda$*. We refer to each element as an identity. Build* $(\mathsf{GenCRS}, \mathsf{R}_1, \mathsf{S}, \mathsf{R}_2)$ *as follows.*

$\mathsf{GenCRS}(1^\lambda)$*: Return* $\mathsf{crs} \leftarrow_\$ \mathsf{HGen}(1^\lambda, 3\lambda)$.

$\mathsf{R}_1(\mathsf{crs}, S_\mathsf{R})$*: Assume* $|S_\mathsf{R}| = 2^d$*. (With small tweaks the same construction works if* $S_\mathsf{R}$ *is not a power of two.)*

– *Parse* $\mathsf{crs} := \mathsf{pp}$*. Let* $n := 2^d$*, and sort* $S_\mathsf{R} := \{\mathsf{id}_1, \ldots, \mathsf{id}_n\}$*, where* $\mathsf{id}_i < \mathsf{id}_{i+1}$ *for all* $i$*. Populate the leaf node values as follows. For each* $\mathsf{id}_i \in S_\mathsf{R}$*, sample* $x_i, x'_i \leftarrow_\$ \{0,1\}^\lambda$*, and let* $h_i^{(0)} := \mathsf{Hash}(\mathsf{pp}, \mathsf{id}_i, x_i, x'_i)$*. Set* $\mathsf{H}[v_i^{(0)}] := h_i^{(0)}$ *and* $\mathsf{ID}[v_i^{(0)}] := \mathsf{id}_i$*.*
  1. *For* $w \in [d]$*, populate the values for the nodes at level* $w$ *as follows. Informally, the hash value for each node is the hash of the concatenation of its left child, right child, and the largest identity value under its left child. Formally, noting we have* $2^{d-w}$ *nodes on level* $w$*, for* $j \in [2^{d-w}]$*, set* $h_j^{(w)} := \mathsf{Hash}(\mathsf{pp}, (h_{2j-1}^{(w-1)}, h_{2j}^{(w-1)}, \mathsf{id}_{[j,w]}))$*, where* $\mathsf{id}_{[j,w]}$ *denotes the larges leaf identity under the left child of the current node (i.e.,* $\mathsf{id}_{[j,w]} = \mathsf{id}_f$*, where* $f := (2j-1)2^{w-1}$*.) Set* $\mathsf{H}[v_j^{(w)}] = h_j^{(w)}$ *and* $\mathsf{ID}[v_j^{(w)}] = \mathsf{id}_{[j,w]}$*.*
  2. *Set* $\mathsf{psi}_1 := (d, \mathsf{hr}_\mathsf{root})$*, where* $\mathsf{hr}_\mathsf{root} := h_1^{(d)}$ *(i.e., the root hash value). Set* $\mathsf{st} := (S_\mathsf{R}, \{x_i\}, \{x'_i\}, \{v_j^{(w)}\})$ *for all values of* $i \in [n]$*,* $w \in \{0, \ldots, d\}$ *and* $j \in [2^{d-w}]$*.*

$\mathsf{S}(\mathsf{crs}, \mathsf{id}, \mathsf{psi}_1)$*:*

– *Parse* $\mathsf{psi}_1 := (d, \mathsf{hr}_\mathsf{root})$ *and* $\mathsf{crs} := \mathsf{pp}$*. Sample* $r, r' \leftarrow_\$ \{0,1\}^\lambda$ *and let* $\mathsf{C}_0 := \mathsf{F}[\mathsf{id}, r, r']$ *(Table 1). Garble* $(\widetilde{\mathsf{C}}_0, \{\mathsf{lb}_{i,b}^{(0)}\}) \leftarrow_\$ \mathsf{Garb}(\mathsf{C}_0)$*. For* $1 \le w \le d$
  1. *Sample* $\boldsymbol{r}_w$ *at random, and let* $\mathsf{C}_w := \mathsf{V}[\mathsf{pp}, \mathsf{id}, \{\mathsf{lb}_{i,b}^{(w-1)}\}, \boldsymbol{r}_w]$*.*
  2. *Garble* $(\widetilde{\mathsf{C}}_w, \{\mathsf{lb}_{i,b}^{(w)}\}) \leftarrow_\$ \mathsf{Garb}(\mathsf{C}_w)$*.*

– *Let* $\{\mathsf{cth}_{i,b}\} \leftarrow_\$ \mathsf{HEnc}(\mathsf{pp}, \mathsf{hr}_\mathsf{root}, \{\mathsf{lb}_{i,b}^{(d)}\})$*. Return* $\mathsf{psi}_2 := (\widetilde{\mathsf{C}}_0, \ldots, \widetilde{\mathsf{C}}_d, \{\mathsf{cth}_{i,b}\}, r)$*.*

$\mathsf{R}_2(\mathsf{crs}, \mathsf{st}, \mathsf{psi}_2)$*:*

– *Parse* $\mathsf{st} := (S_\mathsf{R}, \{x_i\}, \{x'_i\}, \{v_j^{(w)}\})$*,* $\mathsf{psi}_2 := (\widetilde{\mathsf{C}}_0, \ldots, \widetilde{\mathsf{C}}_d, \{\mathsf{cth}_{i,b}\}, r)$ *and* $S_\mathsf{R} := \{\mathsf{id}_1, \ldots, \mathsf{id}_n\}$*. For* $i \in [n]$ *let* $\mathsf{pth}_i := ((\mathsf{id}_i, x_i, x'_i), \ldots, \mathsf{hr}_\mathsf{root})$ *be the* $i$*'th leaf-root path in the tree, and let*

$$r_i := \mathsf{DecPath}(\mathsf{pth}_i, \widetilde{\mathsf{C}}_0, \ldots, \widetilde{\mathsf{C}}_d, \{\mathsf{cth}_{i,b}\}).$$

*If for a unique index* $i \in [n]$*,* $r_i = r$*, then output* $\mathsf{id}_i$*. Otherwise, output* $\perp$*.*

**Theorem 1.** *Assuming the hash encryption* $\mathsf{HE}$ *is anonymous and robust (robustness defined in Lemma 3), and that the garbling scheme* $\mathsf{GS}$ *is anonymous, the* $\ell$*PSI protocol of Construction 1 is correct and provides statistical security for the receiver and semi-honest security for the sender. As a result, such* $\ell$*PSI protocols can be realized from CDH/LWE.*

*Roadmap for the Proof of Theorem 1.* The fact that the protocol provides statistical security for the receiver follows from the robustness of $\mathsf{HE}$. In particular, robustness implies that $h_i^{(0)}$ values statistically hide $S_{\mathsf{R}}$. We can continue this to argue that all the first-level hash values (i.e., $h_i^{(1)}$) also hide $S_{\mathsf{R}}$, and hence, continuing like this, the root hash value $\mathsf{hr}_{\mathsf{root}}$ statistically hides $S_{\mathsf{R}}$.

We now prove that the protocol provides sender security against semi-honest receivers. Let $\mathsf{id}$ be the sender's input message, and $S_{\mathsf{R}} := \{\mathsf{id}_1, \dots, \mathsf{id}_n\}$ be the receiver's set, where $\mathsf{id}_i < \mathsf{id}_{i+1}$. Assuming $\mathsf{id} \notin S_{\mathsf{R}}$ we will show that the sender's protocol message is pseudorandom in the receiver's point of view. For simplicity suppose $\mathsf{id} < \mathsf{id}_1$; the general case follows via simple changes, which we will explain later. Let

$$\mathsf{pth} := ((\underbrace{\mathsf{id}_1, x_1, x_1'}_{z_0}), (\underbrace{h_0, h_0', \mathsf{id}_0}_{z_1}), \dots, (\underbrace{h_{d-1}, h_{d-1}', \mathsf{id}_{d-1}}_{z_d}), \mathsf{hr}_{\mathsf{root}}) \tag{1}$$

be the leaf-root path from leaf $\mathsf{id}_1$ to the root. Note $\mathsf{hr}_{\mathsf{root}} = \mathsf{Hash}(\mathsf{pp}, z_d)$, and $h_i = \mathsf{Hash}(\mathsf{pp}, z_i)$ for $i \in \{0, \dots, d-1\}$. Noting that $\mathsf{hr}_{\mathsf{root}}$ is the receiver's protocol message produced based on her random coins $\mathsf{st}$, we define the following hybrids for the sender's response message.

$\mathbf{Hyb}_0$*:* The sender's response message $\mathsf{psi}_2$ is formed as in the protocol.

$\mathbf{Hyb}_1$*:* Sample $r, r' \leftarrow_\$ \{0,1\}^\lambda$. Let $(\widetilde{\mathsf{C}}_0, \{\mathsf{lb}_i^{(0)}\} \leftarrow_\$ \mathsf{Sim}(\mathsf{F}, r')$. For $1 \le w \le d$

1. Sample $\{\mathsf{cth}_{i,b}^{(w-1)}\} \leftarrow_\$ \mathsf{HEnc}(\mathsf{pp}, h_{w-1}, \{\mathsf{lb}_i^{(w-1)}\})$.
2. Let $(\widetilde{\mathsf{C}}_w, \{\mathsf{lb}_i^{(w)}\}) \leftarrow_\$ \mathsf{Sim}(\mathsf{V}, \{\mathsf{cth}_{i,b}^{(w-1)}\})$.

Let $\{\mathsf{cth}_{i,b}\} \leftarrow_\$ \mathsf{HEnc}(\mathsf{pp}, \mathsf{hr}_{\mathsf{root}}, \{\mathsf{lb}_i^{(d)}\})$. Return $\mathsf{psi}_2 := (\widetilde{\mathsf{C}}_0, \dots, \widetilde{\mathsf{C}}_d, \{\mathsf{cth}_{i,b}\}, r)$.

**Lemma 5** *Given* $\mathsf{R}$*'s random coins,* $\mathbf{Hyb}_0$ *and* $\mathbf{Hyb}_1$ *are indistinguishable.*

$\mathbf{Hyb}_2$*:* Sample $\mathsf{psi}_2$ at random.

**Lemma 6** *Given* $\mathsf{R}$*'s random coins,* $\mathbf{Hyb}_1$ *and* $\mathbf{Hyb}_2$ *are indistinguishable.*

The above two lemmas establish sender's security; namely — if $\mathsf{id} \notin S_{\mathsf{R}}$, then the sender's message $\mathsf{psi}_2$ is pseudorandom for the receiver. We prove Lemma 5, Lemma 6 and correctness in the full paper [1].

## 5 Reusable DV-NIZK Range Proofs for DJ Ciphertexts

In this section, we construct a DV-NIZK scheme for ranges of DJ ciphertexts. The main idea of our construction is the following: the prover proves that a BGN ciphertext [5] is within a certain range (this can be done via the protocol of [29]). Then it proves that the DJ and BGN ciphertexts encrypt the same value.

We first recall the required cryptosystems used in this section.

*BGN cryptosystem.* Recall that the BGN cryptosystem [5] is defined over a group $\mathbb{G}$ of order $n = pq$ for primes $p, q$. The public key is composed by $(\mathbb{G}, n, G, H)$ where $G$ is a generator of $\mathbb{G}$ and $H$ is an element of order $p$ (let $p\mathbb{G}$ be the subgroup of order $p$). The public key is composed by $(\mathbb{G}, n, G, H)$ and a ciphertext for a message $m \in \{0, 1\}$ is of the form $C = G^m H^t$ for $t \leftarrow_\$ \mathbb{Z}_n$.

*Damgård-Jurik cryptosystem.* The Damgård-Jurik (DJ) cryptosystem[19] [15] is defined over $\mathbb{Z}^*_{N^{\xi+1}}$ where $N \leftarrow_\$ \mathsf{RSA}(\lambda)$. The public key is formed by $(N, \xi, g, h)$ where $g \leftarrow_\$ \mathbb{T}_N$ and $h = g^x$ for $x \leftarrow_\$ [N]$. A ciphertext has the form $(c_1, c_2)$ where $c_1 = g^t \bmod N^{\xi+1}$ and $c_2 = h^t(1 + N)^m \bmod N^{\xi+1}$ for $t \leftarrow_\$ [N]$ and $m \in [N^\xi]$.

### 5.1 Equality of Plaintexts in DJ and BGN ciphertexts.

We now show how to prove that a BGN and a DJ ciphertexts encrypt the same value. Consider the following language

$$\mathcal{EQ}_\Delta = \left\{ D_0, h_0, \{D_i, c_{1,i}, c_{2,i}\}_{i\in[\ell]} : \exists (r, t, \{m_i\}) \text{ s.t. } \begin{array}{c} m_i \in \{0,1\} \\ D_0 = H_0^r \in \mathbb{G} \\ D_i = G^m H_i^r \in \mathbb{G} \\ c_0 = h_0^t \in \mathbb{Z}_{N^{\xi+1}} \\ c_1 = g^t \in \mathbb{Z}_{N^{\xi+1}} \\ c_{2,i} = h_i^t (1+N)^{m_i} \in \mathbb{Z}_{N^{\xi+1}} \end{array} \right\}$$

where $\Delta = (\mathbb{G}, n, G, H_0, \{H_i\}_{i\in[\ell]}, N, \xi, g, h_0, \{h_i\}_{i\in[\ell]})$, $G, H_0, \{H_i\}_{i\in[\ell]} \in \mathbb{G}$ and $g, h_0, \{h_i\}_{i\in[\ell]} \in \mathbb{Z}_{N^{\xi+1}}$.

The DV-NIZK construction for the language above is outlined in Section 2. We defer the full construction and its analysis to [1].

### 5.2 DV-NIZK for Range Proofs of DJ Ciphertexts with Equal Discrete Log

Let $N \leftarrow \mathsf{RSA}(\lambda)$ and $\xi \geq 0$ be a fixed integer. Consider the following language of ranges:

$$\mathcal{REDJ}_\Delta = \left\{ c_1 \in \{\mathbb{Z}^*_{N^{\xi+1}}\}^2 : \exists t \in \{\lceil -N^\xi/2, \ldots, N^\xi/2 \rceil\} \text{ s.t. } \begin{array}{c} t \in [-B, B] \\ c_1 = g^t \bmod N^{\xi+1} \end{array} \right\}$$

which is parametrized by $\Delta = (g, B, N, \xi)$ where $g \in \mathbb{T}_N$, $B \in \mathbb{Z}$, $N$ and $\xi$.

In the following, we present a DV-NIZK scheme for the language above. The main idea is quite simple: The prover outputs BGN ciphertexts $D_i$ encrypting bits $m_i$ and DJ ciphertexts $(c_{1,i}, c_{2,i}$ that encrypt the same values as $D_i$ (we can prove this using the scheme from the previous section). Then, the prover proves that $(c_1, c_2)$ encrypts the same value as $\left(\prod_{i=0}^{\ell} c_{1,i}^{2^i}, \prod_{i=0}^{\ell} c_{2,i}^{2^i}\right)$. Since DJ is linearly-homomorphic, we conclude that $(c_1, c_2)$ encrypts $m = \sum_{i=0}^{\ell} 2^i m_i \leq 2^{\ell-1}$.

Due to space restrictions, the full construction is presented in [1].

[19] Here, we present a slightly different variant of the scheme in [15].

## 6 Reusable Laconic Private Set Intersection

In this section, we present a protocol that implements $\ell$PSI in a black-box fashion. We then prove that the protocol guarantees security against a semi-honest receiver and against a malicious sender. The input sets are subsets of a universe $\mathcal{U}$ of exponential size.

*Protocol.* We now present the construction for reusable PSI.

**Construction 2** *Let $\mathcal{U}$ be a universe which contains the input sets of the parties. Let $\kappa \in \mathbb{Z}$ such that $5\kappa \leq \lambda$ and $\xi \in \mathbb{N}$. We require the following ingredients in this construction:*

1. *A PPRF* $\mathsf{PPRF} : \mathcal{K} \times \mathcal{U} \to \mathsf{Primes}(\kappa)$ *which outputs a prime number.*[20]
2. *A DV-NIZK*

   $$\mathsf{NIZK}_{\mathcal{REDJ}_\Delta} = (\mathsf{NIZK.GenCRS}_{\mathcal{REDJ}_\Delta}, \mathsf{NIZK.Prove}_{\mathcal{REDJ}_\Delta}, \mathsf{NIZKVerify}_{\mathcal{REDJ}_\Delta})$$

   *for the language $\mathcal{REDJ}_\Delta$ which is defined in Section 5, for some $\Delta = (g_0, B, N, \xi)$.*
3. *An IND-CPA PKE scheme* $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$
4. *A $(\kappa - 1, \mathsf{negl}(\lambda))$-strong extractor* $\mathsf{Ext} : \mathcal{S} \times \mathbb{Z}_{N^{\xi+1}} \to \{0,1\}^\lambda$.

*We assume that the receiver's set is of size $M$ and the sender's set is of size $m$, where $M > m$. The protocol is composed by the following algorithms:*

$\mathsf{GenCRS}(1^\lambda)$ :

- *Sample $N \leftarrow_\$ \mathsf{RSA}(\lambda)$, that is, $N = PQ$ where $P, Q$ are safe prime numbers. Choose $B$ such that $N^{\xi-1}/2 \geq B > N2^\kappa$.*
- *Sample a pair of public and secret keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$. *Additionally, sample a PPRF key $k \leftarrow_\$ \mathcal{K}$. Set $\Delta = (g_0, B, N, \xi)$ where $g_0 \leftarrow_\$ \mathbb{T}_N$.*
- *Output* $\mathsf{crs} = (N, \mathsf{pk}, g_0, B, k, \Delta)$.

$\mathsf{R}_1(\mathsf{crs}, S_\mathsf{R})$ :

- *Parse* $\mathsf{crs} := (N, \mathsf{pk}, g_0, B, k, \Delta)$, *and* $S_\mathsf{R} := \{\mathsf{id}_i\}_{i \in [M]} \subseteq \mathcal{U}$
- *Compute the prime numbers $p_i \leftarrow \mathsf{PPRF}(k, \mathsf{id}_i)$, for all $i \in [M]$.*
- *Sample $r \leftarrow_\$ [N/4]$ and compute $h = g_0^{r \prod_{i \in [M]} p_i} \bmod N^{\xi+1}$.*
- *Run* $(\mathsf{crs}_1, \mathsf{td}_1) \leftarrow \mathsf{NIZK.GenCRS}_{\mathcal{REDJ}_\Delta}(1^\lambda)$.
- *Output* $\mathsf{st} = (r, \mathsf{td}_1)$ *and* $\mathsf{psi}_1 = (h, \mathsf{crs}_1)$.

[20] We remark that we use a PPRF, not because we want uniform outputs, but to implicitly define the set of primes. A similar trick was used in [6].

$\mathsf{S}(\mathsf{crs}, S_\mathsf{S}, \mathsf{psi}_1)$ :

– *Parse* $\mathsf{crs} := (N, \mathsf{pk}, g_0, B, k, \Delta)$, $\mathsf{psi}_1 := (h, \mathsf{crs}_1)$ *and* $S_\mathsf{S} := \{\mathsf{id}'_i\}_{i \in [m]} \subseteq \mathcal{U}$.
– *For* $i \in [m]$ *do the following:*
  - *Sample* $\rho_i \leftarrow_\$ [N/4]$. *Compute the prime numbers* $p_i \leftarrow \mathsf{PPRF}(k, \mathsf{id}'_i)$.
  - *Sample an extractor seed* $s_i \leftarrow_\$ \mathcal{S}$ *and compute* $R_i \leftarrow \mathsf{Ext}(s_i, h^{\rho_i} \bmod N^{\xi+1})$
  - *Compute* $f_i = g_0^{\rho_i p_i} \bmod N^{\xi+1}$ *and* $\mathsf{ct}_i \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{id}'_i; R_i)$.
  - *Compute* $\pi_i \leftarrow \mathsf{NIZK.Prove}_{\mathcal{REDJ}_\Delta}(\mathsf{crs}_1, x_i, w_i)$ *where* $x_i = f_i$ *and* $w_i = \rho_i p_i$.
– *Output* $\mathsf{psi}_2 = \{f_i, \mathsf{ct}_i, s_i, \pi_i\}_{i \in [m]}$.

$\mathsf{R}_2(\mathsf{crs}, \mathsf{st}, \mathsf{psi}_2)$ :

– *Parse* $\mathsf{st} := (r, \mathsf{td}_1)$ *and* $\mathsf{psi}_2 := \{f_i, \mathsf{ct}_i, s_i, \pi_i\}_{i \in [m]}$. *Set* $\mathcal{I} = \emptyset$
– *For all* $j \in [m]$ *do the following:*
  - *If* $0 \leftarrow \mathsf{NIZK.Verify}_{\mathcal{REDJ}_\Delta}(\mathsf{td}_1, x_j, \pi_j)$ *where* $x_j = f_j$, *abort the protocol.*
  - *If there is a* $i \in [M]$ *such that*

$$\mathsf{ct}_j = \mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{id}_i; R'_i)$$

    *where* $R'_i \leftarrow \mathsf{Ext}(s_j, f_j^{r_i} \bmod N^{\xi+1})$ *and* $r_i = r \prod_{\ell=1, \ell \neq i}^{M} p_\ell$, *then add the element* $\mathsf{id}_i$ *to* $\mathcal{I}$.
– *Output* $\mathcal{I}$.

*Communication cost.* Here, we analyze the communication cost of the protocol as a function of the input set sizes $|S_\mathsf{S}| = m$ and $|S_\mathsf{R}| = M$ and we omit polynomial factors in the security parameter $\lambda$. The first message outputted by $\mathsf{R}_1$ has size $\mathcal{O}(1)$. The second message outputted by $\mathsf{S}$ has size $\mathcal{O}(m)$. The overall communication cost is $\mathcal{O}(m)$, that is, it is independent of $M$.

*Analysis.* We now analyze the correctness and security of the protocol.

**Theorem 2.** *The protocol presented in Construction 2 is correct given that* $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$ *is complete and* $\mathsf{PKE}$ *is correct.*

The proof is presented in the full paper [1].

**Theorem 3.** *The protocol presented in Construction 2 securely UC-realizes functionality* $\mathcal{F}_{\mathsf{rPSI}}$ *in the* $\mathcal{G}_{\mathsf{CRS}}$*-hybrid model against:*

– *a semi-honest receiver given that the* $\phi$*-hiding assumption hold and* $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$ *is zero-knowledge;*
– *a malicious sender, given that the DCR assumption holds and* $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$ *is reusable sound.*

*Proof.* We start by proving that the protocol is secure against semi-honest adversaries corrupting the receiver.

**Lemma 7** *The protocol is secure against a semi-honest receiver.*

We first show how the simulator $\mathsf{Sim}_\mathsf{R}$ works. In the following, let $\mathsf{Sim}_\mathsf{NIZK}$ be the zero-knowledge simulator from Lemma **??** for the $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$ scheme.

1. $\mathsf{Sim}_\mathsf{R}$ takes the input $S_\mathsf{R}$ of $\mathsf{R}$ and sends it to the ideal functionality $\mathcal{F}_\mathsf{rPSI}$.
2. **CRS generation.** To generate the CRS, $\mathsf{Sim}$ behaves as the honest algorithm would do.
3. The simulator creates the semi-honest receiver's view exactly as in the real protocol and keeps $\mathsf{st} = (r, \mathsf{td}_1)$ to itself.
4. Upon receiving a message $\mathsf{psi}_1 = (h, \mathsf{crs}_1)$ from $\mathsf{R}$ and a message $\mathcal{I}$ (of size $m'$, that is, $|\mathcal{I}| = m'$) from the ideal functionality $\mathcal{F}_\mathsf{rPSI}$, the simulator does the following:
   - Sample a subset $\mathcal{X}$ of size $m-m'$ from the universe $\mathcal{U}$ and sets $S_\mathcal{S} = \mathcal{I} \cup \mathcal{X}$.
   - For all $i \in \mathcal{I}$, $\mathsf{Sim}_\mathsf{R}$ computes $(f_i, \mathsf{ct}_i, s_i, \pi_i)$ as in the real protocol.
   - For all $i \in S_\mathsf{S} \setminus \mathcal{I}$, $\mathsf{Sim}_\mathsf{R}$ simulates proofs $\pi_i \leftarrow \mathsf{Sim}_\mathsf{NIZK}(\mathsf{td}_1, x)$ for $x = f_i$ where $f_i \leftarrow\!\!\$\, \mathbb{T}_N$. Then, it encrypts $\mathsf{ct}_i \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, 0; R_i)$ where $R_i \leftarrow \{0,1\}^\lambda$.

To prove indistinguishability between the real protocol and the simulated one, we consider the following sequence of hybrids:

$\mathbf{Hyb}_0$*:* The is the real protocol.

$\mathbf{Hyb}_1$*:* This hybrid is identical to the previous one, except that, for $i \in S_\mathsf{S} \setminus \mathcal{I}$, $\mathsf{Sim}_\mathsf{R}$ simulates the proofs $\pi_i \leftarrow \mathsf{Sim}_\mathsf{NIZK}(\mathsf{td}_1, x)$ for $x_i = f_i$.

**Claim 1** *Hybrids* $\mathbf{Hyb}_0$ *and* $\mathbf{Hyb}_1$ *are statistically indistinguishable.*

The claim above follows directly from the statistical zero-knowledge of the scheme $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$.

$\mathbf{Hyb}_{2,\ell}$*:* This hybrid is identical to the previous one, except that the simulator samples $f_{u_\ell} \leftarrow\!\!\$\, \mathbb{T}_N$ and computes

$$R_{u_\ell} \leftarrow \mathsf{Ext}\left(s, f_{u_\ell}^{r q_{u_\ell}^{-1} \Pi_j^M p_j} \bmod N^{\xi+1}\right)$$

where $q_{u_\ell} \leftarrow \mathsf{PPRF}(k, x_{u_\ell})$ for all $u_\ell \in \{i : x_i \in S_\mathsf{S} \setminus \mathcal{I}\}$ and $p_j \leftarrow \mathsf{PPRF}(k, y_j)$ for all $y_j \in S_\mathsf{R}$. The hybrid is defined for $\ell = 1, \ldots, m - m'$.

**Claim 2** *Hybrids* $\mathbf{Hyb}_1$ *and* $\mathbf{Hyb}_{2,m-m'}$ *are indistinguishable.*

The proof of the claim is deferred to the full paper [1].

$\mathbf{Hyb}_{3,\ell}$*:* This hybrid is identical to the previous one except that $\mathsf{Sim}_\mathsf{R}$ computes $R_{u_\ell} \leftarrow\!\!\$\, \{0,1\}^\lambda$ for all $u_\ell \in \{i : x_i \in S_\mathsf{S} \setminus \mathcal{I}\}$. The hybrid is defined for $\ell = 1, \ldots, m - m'$.

**Claim 3** *Assume that* $\mathsf{Ext}$ *is a* $(\kappa - 1, \mathsf{negl}(\lambda))$*-strong extractor and that the* $\phi$*-hiding assumption holds. Then hybrids* $\mathbf{Hyb}_{2,m-m'}$ *and* $\mathbf{Hyb}_{3,m-m'}$ *are indistinguishable.*

The proof of the claim is in [1].

**Hyb**$_{4,\ell}$*:* This hybrid is identical to the previous one except that $\mathsf{Sim}_{\mathsf{R}}$ encrypts $\mathsf{ct}_{u\ell} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, 0; R_{u_\ell})$ for all for all $u_\ell \in \{i : x_i \in S_{\mathsf{S}} \setminus \mathcal{I}\}$. The hybrid is defined for $\ell = 1, \ldots, m - m'$. Hybrid $\mathbf{Hyb}_{4,m-m'}$ is identical to the simulation.

**Claim 4** *Assume that* PKE *is an IND-CPA PKE. Then hybrids* $\mathbf{Hyb}_{3,m-m'}$ *and* $\mathbf{Hyb}_{4,m-m'}$ *are indistinguishable.*

The claim follows directly from the IND-CPA property of the underlying PKE. That is, given an adversary $\mathcal{A}$ that distinguishes both hybrids, we can easily build an adversary $\mathcal{B}$ against the IND-CPA property of PKE. This adversary $\mathcal{B}$ simply chooses as messages $m_0 = x_{u_\ell}$ (where $x_{u_\ell} \in S_{\mathsf{S}} \setminus \mathcal{I}$) and $m_1 = 0$. It outputs whatever $\mathcal{A}$ outputs.

We first show how the simulator $\mathsf{Sim}_{\mathsf{S}}$ extracts the sender's input:

1. **CRS generation.** $\mathsf{Sim}_{\mathsf{S}}$ generates the $\mathsf{crs}$ following the algorithm $\mathsf{GenCRS}$, except that it sets $g_0 = g_0'(1+N)$ for $g_0' \leftarrow_{\$} \mathbb{T}_N$. It keeps $\phi(N)$ to itself (which can be computed using the prime numbers $P, Q$) and the secret key $\mathsf{sk}$ corresponding to $\mathsf{pk}$. It outputs $\mathsf{crs} = (\mathsf{pk}, g_0, B, k, \Delta)$
2. $\mathsf{Sim}_{\mathsf{S}}$ samples $h \leftarrow_{\$} \mathbb{T}_N$ and computes $(\mathsf{crs}_1, \mathsf{td}_1) \leftarrow \mathsf{NIZK.GenCRS}_{\mathcal{REDJ}_\Delta}(1^\lambda)$. It sends $\mathsf{psi}_1 = (h, \mathsf{crs}_1)$ to the malicious sender.
3. Whenever $\mathsf{Sim}_{\mathsf{S}}$ receives a message $\mathsf{psi}_2 = \{f_i, \mathsf{ct}, s_i, \pi_i\}_{i\in[m]}$ from the sender, the simulator initially sets $S_{\mathsf{S}}$ and does the following for all $i \in [m]$:
   - It checks if $1 \leftarrow \mathsf{NIZK.Verify}_{\mathcal{REDJ}_\Delta}(\mathsf{td}_1, x_j, \pi_j)$ where $x_j = f_j$, and aborts otherwise.
   - It computes $\mathsf{id}_i' \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct}_i)$ and $p_i \leftarrow \mathsf{PPRF}(k, \mathsf{id}_i')$. Additionally, it extracts $\zeta_i$ by recovering $\zeta_i'$ from $(1+N)^{\zeta_i'} = f_i^{\phi(N)}$ and computing $\zeta = \zeta'/\phi(N)$ over the integers. It computes $\rho_i' = \zeta_i/p_i$ over the integers. If $\mathsf{ct}_i = \mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{id}_i'; R_i)$ where $R_i = \mathsf{Ext}(s_i, h^{\rho_i'} \bmod N^{\xi+1})$, then it adds $\mathsf{id}_i'$ to $S_{\mathsf{S}}$.
4. It sends $S_{\mathsf{S}}$ to $\mathcal{F}_{\mathsf{PSI}}$ and halts.

We now show that the simulation is indistinguishable from the real protocol via the following sequence of hybrids.

**Hyb**$_0$*:* This hybrid is the real protocol.

**Hyb**$_1$*:* This hybrid is identical to the previous one except that the simulator computes the first message (sent by the receiver) as $h \leftarrow_{\$} \mathbb{T}_N$.

**Claim 5** *Hybrids* $\mathbf{Hyb}_0$ *and* $\mathbf{Hyb}_1$ *are statistically indistinguishable.*

Since $g_0$ is a generator of $\mathbb{T}_N$, the distributions of $g^x$ and $h \leftarrow_{\$} \mathbb{T}_N$ are identical. It follows that the hybrids are indistinguishable.

**Hyb**$_2$*:* This hybrid is identical to the previous one, except that $g_0 = g_0'(1+N)$ for $g_0' \leftarrow_{\$} \mathbb{T}_N$ (instead of choosing $g_0 \leftarrow_{\$} \mathbb{T}_N$). Additionally, $\mathsf{Sim}_{\mathsf{S}}$ keeps $(\phi(N), \mathsf{sk})$ while creating $\mathsf{crs}$.

**Claim 6** *Assume that the DCR assumption holds. Then hybrids* $\mathbf{Hyb}_1$ *and* $\mathbf{Hyb}_2$ *are indistinguishable.*

**Hyb**$_3$*:* This hybrid is identical to the previous one except that the simulator, instead of checking if there is an index $i$ for which

$$\mathsf{ct}_j = \mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{id}_i; R'_i)$$

where $R'_i = \mathsf{Ext}(s_j, f_j^{r_i})$ and $r_i = r \prod_{\ell=1,\ell\neq i}^{M} p_\ell$ (as in the real protocol), it does the checks as in the simulation. That is, it computes $\mathsf{id}'_i \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct}_i)$ and $p_i \leftarrow \mathsf{PPRF}(k, \mathsf{id}'_i)$. Additionally, it extracts $\zeta_i$ by recovering $\zeta'_i$ from $(1+N)^{\zeta'_i} = f_i^{\phi(N)}$ and computing $\zeta = \zeta'/\phi(N)$. It computes $\rho'_i = \zeta_i/p_i$ over the integers. Then, it checks if $\mathsf{ct}_i = \mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{id}'_i; R_i)$ where $R_i = \mathsf{Ext}(s_i, h^{\rho'_i})$.

**Claim 7** *Hybrids* **Hyb**$_2$ *and* **Hyb**$_3$ *are indistinguishable given that* PKE *is correct and* $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$ *is simulation sound.*

By the simulation soundness of $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$, $\zeta_i < N^{\xi-1}/2$. Hence, $\zeta'_i < N^\xi/2$ and thus $\zeta'_i \bmod N^\xi$ is equal to $\zeta'_i$ as an integer. Computing $\zeta = \zeta'_i/\phi(N)$ yields $\rho_i p_i$ over $\mathbb{Z}$. Thus $\rho_i = \zeta_i/p_i$ over $\mathbb{Z}$.

Thus, performing the checks in this hybrid has the same outcome as in the real protocol. □

*Setting the parameters.* The value $B$ is such that $N^{\xi-1}/2 \geq B > N2^\kappa$ for $5\kappa \leq \lambda$. Then, it is enough to set $\xi = 3$, so that we can find a $B$ that fulfills the condition.

*Achieving statistical security against the sender.* The protocol presented in Construction 2 achieves computational security against a malicious sender given that the DCR assumption holds (recall that $\mathsf{NIZK}_{\mathcal{REDJ}_\Delta}$ achieves statistical reusable soundness).

The only place where we use the DCR assumption in the proof of security against a malicious sender is when we replace $g_0 \leftarrow_\$ \mathbb{T}_N$ by $g_0 = g'_0(1+N)$. Hence, consider the following modification of the protocol presented in Construction 2: In GenCRS, the element $g_0$ is chosen as $g'_0(1+N)$ for $g'_0$. This simple modification of the protocol yields a new one which is *statistically secure against a malicious sender*. On the other hand, security against a semi-honest receiver now relies on the hardness of $\phi$-hiding (as before) *and* the DCR assumption.

## 7 Self-Detecting Encryption

In this section we define self-detecting encryption, and show how to build it from laconic PSI. We first give a semi-honest definition, and will present the malicious definition in the full paper [1].

**Definition 7** *A Self-Detecting Encryption (SDE) scheme is tuple of (randomized) algorithms* $\mathsf{SDE} = (\mathsf{Prm}, \mathsf{Gen}, \mathsf{Hash}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Detect})$ *such that:*

- $\mathsf{Prm}(1^\lambda)$*: Takes as input a security parameter* $1^\lambda$*, and outputs a public parameter* $\mathsf{pp}$.
- $\mathsf{Gen}(\mathsf{pp})$*: Takes as input a public parameter* $\mathsf{pp}$*, and outputs a pair of keys* $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Hash}(\mathsf{pp}, \mathsf{DB})$*: Takes as input a public parameter* $\mathsf{pp}$ *and a database* $\mathsf{DB}$*, and outputs a hash value* $\mathsf{h}$ *and a private state* $\mathsf{st}$*. We require* $|\mathsf{h}| \leq \mathsf{poly}(\lambda)$*, for a fixed polynomial* $\mathsf{poly}$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{h}, \mathsf{m})$*: Takes as input a public key* $\mathsf{pk}$*, a hash value* $\mathsf{h}$*, and a message* $\mathsf{m}$*, and outputs a ciphertext* $\mathsf{ct}$.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$*: Takes as input a secret key* $\mathsf{sk}$ *and a ciphertext* $\mathsf{ct}$*, and outputs a message* $\mathsf{m}$ *or* $\perp$.
- $\mathsf{Detect}(\mathsf{st}, \mathsf{ct})$*: Takes as input a private state* $\mathsf{st}$ *and a ciphertext* $\mathsf{ct}$*, and outputs a message* $\mathsf{m}$ *or* $\perp$.

*We require the following properties:*

- ***Correctness.*** *For any message* $\mathsf{m}$*, letting* $\mathsf{pp} \leftarrow_\$ \mathsf{Prm}(1^\lambda)$ *and* $(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{Gen}(\mathsf{pp})$*:* $\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) \neq \mathsf{m}] \leq \mathsf{negl}(\lambda)$.
- ***Detection.*** *For any* $\mathsf{pp} \in \mathsf{Prm}$*, any* $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$*, any database of strings* $\mathsf{DB}$*, and any message* $\mathsf{m}$*, letting* $(\mathsf{h}, \mathsf{st}) \leftarrow_\$ \mathsf{Hash}(\mathsf{pp}, \mathsf{DB})$ *and* $\mathsf{ct} \leftarrow_\$ \mathsf{Enc}(\mathsf{pk}, \mathsf{h}, \mathsf{m})$*, if* $\mathsf{m} \in \mathsf{DB}$ *then* $\mathsf{Detect}(\mathsf{st}, \mathsf{ct}) = \mathsf{m}$.
- ***Efficiency.*** *The size of $h$ and running time of* $\mathsf{Enc}$ *are independent of the database size. There exists a polynomial* $\mathsf{poly}$ *s.t. for all $n := n(\lambda)$, any* $\mathsf{DB} \in \{0,1\}^n$*, letting* $h \leftarrow_\$ \mathsf{Hash}(\mathsf{pp}, \mathsf{DB})$ *and* $\mathsf{pp}, \mathsf{pk}$ *be as above, then* $|h| \leq \mathsf{poly}(\lambda)$ *and also the running time of* $\mathsf{Enc}(\mathsf{pk}, \mathsf{h}, \mathsf{m})$ *is upper bounded by* $\mathsf{poly}(|\mathsf{m}|, \lambda)$.
- ***Database Hiding.*** *For any two databases* $(\mathsf{DB}_0, \mathsf{DB}_1)$ *of equal size, if* $(\mathsf{h}_0, *) \leftarrow_\$ \mathsf{Hash}(\mathsf{pp}, \mathsf{DB}_0)$ *and* $(\mathsf{h}_1, *) \leftarrow_\$ \mathsf{Hash}(\mathsf{pp}, \mathsf{DB}_1)$ *then* $\mathsf{h}_0$ *and* $\mathsf{h}_1$ *are indistinguishable where* $\mathsf{pp} \leftarrow_\$ \mathsf{Gen}(1^\lambda)$.
- ***Semantic Security.*** *For any database of strings* $\mathsf{DB}$ *and any two messages* $(\mathsf{m}_0, \mathsf{m}_1)$*:* $(\mathsf{pk}, \mathsf{h}, \mathsf{Enc}(\mathsf{pk}, \mathsf{h}, m_0)) \overset{c}{\equiv} (\mathsf{pk}, \mathsf{h}, \mathsf{Enc}(\mathsf{pk}, \mathsf{h}, m_1))$*, where all the variables are sampled as above.*
- ***Security Against the Authority.*** *For any two messages* $(\mathsf{m}_0, \mathsf{m}_1)$*, if* $\mathsf{m}_0 \notin \mathsf{DB}$ *and* $\mathsf{m}_1 \notin \mathsf{DB}$ *then*

$$\big(\mathsf{pk}, (\mathsf{h}, \mathsf{st}), \mathsf{Enc}(\mathsf{pk}, \mathsf{h}, m_0)\big) \overset{c}{\equiv} \big(\mathsf{pk}, (\mathsf{h}, \mathsf{st}), \mathsf{Enc}(\mathsf{pk}, \mathsf{h}, m_1)\big),$$

*where* $\mathsf{pp} \leftarrow_\$ \mathsf{Prm}(1^\lambda)$*,* $(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{Gen}(\mathsf{pp})$*, and* $(\mathsf{h}, \mathsf{st}) \leftarrow_\$ \mathsf{Hash}(\mathsf{pp}, \mathsf{DB})$.

We now show how to realize self-detecting encryption from semi-honest laconic PSI. Informally, the SDE hash is the receiver's first-round laconic PSI message, and the encryption of a message $m$ consists of a PKE encryption of $m$ as well as a second-round PSI message based on $m$.

**Construction 3** *Let* $\mathsf{PKE} = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ *be a CPA-secure PKE scheme*[21] *and* $\mathsf{LPSI} = (\mathsf{GenCRS}, \mathsf{R}_1, \mathsf{S}, \mathsf{R}_2)$ *a laconic PSI.*

[21] We proceed with an independent PKE scheme for the sake of simplicity.

- $\mathsf{Prm}(1^\lambda)$*: Sample* $\mathsf{crs} \leftarrow_\$ \mathsf{LPSI.GenCRS}(1^\lambda)$*, and let* $\mathsf{pp} := \mathsf{crs}$.
- $\mathsf{Gen}(\mathsf{pp})$*: Run* $\mathsf{PKE.Gen}'(1^\lambda)$ *to generate a pair of keys* $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Hash}(\mathsf{pp}, \mathsf{DB})$*: Let* $\mathsf{h}$ *be the output of the receiver on* $\mathsf{DB}$ *and* $\mathsf{pp}$*, i.e.,* $\mathsf{h} \leftarrow_\$ \mathsf{LPSI.R}_1(\mathsf{pp}, \mathsf{DB})$*. In addition, let* $\mathsf{st}$ *be the private state of the receiver.*
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{h}, \mathsf{m})$*: Output* $(\mathsf{ct}_1, \mathsf{ct}_2)$*, where* $\mathsf{ct}_1 \leftarrow_\$ \mathsf{PKE.Enc}'(\mathsf{pk}, \mathsf{m})$ *and* $\mathsf{ct}_2 \leftarrow_\$ \mathsf{LPSI.S}(\mathsf{pp}, \{\mathsf{m}\}, \mathsf{h})$.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2))$*: Output* $\mathsf{PKE.Dec}'(\mathsf{sk}, \mathsf{ct}_1)$.
- $\mathsf{Detect}(\mathsf{st}, \mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2))$*: Output* $\mathsf{R}_2(\mathsf{st}, \mathsf{ct}_2)$.

Correctness and efficiency follow immediately.

- **Statistical** database hiding follows from PSI-receiver statistical security.
- Semantic security and security against the authority property of the scheme follows from the CPA security of PKE scheme $\Pi$ and the sender's security. Observe that if $\mathsf{m} \notin \mathsf{DB}$ then both $\mathsf{ct}_1$ and $\mathsf{ct}_2$ computationally hide the message even in the presence of the private state $\mathsf{st}$ of PSI. Specifically, one can argue that $\mathsf{ct}_1$ computationally hides $\mathsf{m}$ because of the CPA security of PKE scheme $\Pi$, and $\mathsf{ct}_2$ computationally hides $\mathsf{m}$ because of the sender's security of laconic PSI. The arguments above can be made formal via a routine hybrid argument, and we omit the details.

## Acknowledgment

Pedro Branco thanks the support from DP-PMI and FCT (Portugal) through the grant PD/BD/135181/2017. This work is supported by Security and Quantum Information Group of Instituto de Telecomunicações, by the Fundação para a Ciência e a Tecnologia (FCT) through national funds, by FEDER, COMPETE 2020, and by Regional Operational Program of Lisbon, under UIDB/50008/2020.

Nico Döttling: This work is partially funded by the Helmholtz Association within the project "Trustworthy Federated Data Analytics" (TFDA) (funding number ZT-I-OO1 4).

Sanjam Garg is supported in part by DARPA under Agreement No. HR00112020026, AFOSR Award FA9550-19-1-0200, NSF CNS Award 1936826, and research grants by the Sloan Foundation, Visa Inc., and Center for Long-Term Cybersecurity (CLTC, UC Berkeley). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

Mohammad Hajiabadi is supported in part by NSF CNS Award 2055564.

## References

1. Alamati, N., Branco, P., Döttling, N., Garg, S., Hajiabadi, M., Pu, S.: Laconic private set intersection and applications. Cryptology ePrint Archive, Report 2021/728 (2021), `https://ia.cr/2021/728`

2. Ateniese, G., De Cristofaro, E., Tsudik, G.: (If) size matters: Size-hiding private set intersection. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography. Lecture Notes in Computer Science, vol. 6571, pp. 156–173. Springer, Heidelberg, Germany, Taormina, Italy (Mar 6–9, 2011)
3. Baum, C., Bootle, J., Cerulli, A., del Pino, R., Groth, J., Lyubashevsky, V.: Sublinear lattice-based zero-knowledge arguments for arithmetic circuits. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part II. Lecture Notes in Computer Science, vol. 10992, pp. 669–699. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018)
4. Benhamouda, F., Lin, H.: k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part II. Lecture Notes in Computer Science, vol. 10821, pp. 500–532. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018)
5. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005: 2nd Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 3378, pp. 325–341. Springer, Heidelberg, Germany, Cambridge, MA, USA (Feb 10–12, 2005)
6. Boyle, E., Gilboa, N., Ishai, Y.: Breaking the circuit size barrier for secure computation under DDH. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 509–539. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016)
7. Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 1–20. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 2010)
8. Brakerski, Z., Lombardi, A., Segev, G., Vaikuntanathan, V.: Anonymous IBE, leakage resilience and circular security from new assumptions. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part I. Lecture Notes in Computer Science, vol. 10820, pp. 535–564. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018)
9. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy. pp. 315–334. IEEE Computer Society Press, San Francisco, CA, USA (May 21–23, 2018)
10. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science. pp. 136–145. IEEE Computer Society Press, Las Vegas, NV, USA (Oct 14–17, 2001)
11. Chen, H., Huang, Z., Laine, K., Rindal, P.: Labeled PSI from fully homomorphic encryption with malicious security. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018: 25th Conference on Computer and Communications Security. pp. 1223–1237. ACM Press, Toronto, ON, Canada (Oct 15–19, 2018)
12. Chen, H., Laine, K., Rindal, P.: Fast private set intersection from homomorphic encryption. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017: 24th Conference on Computer and Communications Security. pp. 1243–1255. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017)
13. Cho, C., Döttling, N., Garg, S., Gupta, D., Miao, P., Polychroniadou, A.: Laconic oblivious transfer and its applications. In: Katz, J., Shacham, H. (eds.) Advances

in Cryptology – CRYPTO 2017, Part II. Lecture Notes in Computer Science, vol. 10402, pp. 33–65. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017)

14. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) Advances in Cryptology – EUROCRYPT 2002. Lecture Notes in Computer Science, vol. 2332, pp. 45–64. Springer, Heidelberg, Germany, Amsterdam, The Netherlands (Apr 28 – May 2, 2002)
15. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001: 4th International Workshop on Theory and Practice in Public Key Cryptography. Lecture Notes in Computer Science, vol. 1992, pp. 119–136. Springer, Heidelberg, Germany, Cheju Island, South Korea (Feb 13–15, 2001)
16. Döttling, N., Garg, S.: From selective IBE to full IBE and selective HIBE. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017: 15th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 372–408. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017)
17. Döttling, N., Garg, S.: Identity-based encryption from the Diffie-Hellman assumption. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017, Part I. Lecture Notes in Computer Science, vol. 10401, pp. 537–569. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017)
18. Döttling, N., Garg, S., Goyal, V., Malavolta, G.: Laconic conditional disclosure of secrets and applications. In: Zuckerman, D. (ed.) 60th Annual Symposium on Foundations of Computer Science. pp. 661–685. IEEE Computer Society Press, Baltimore, MD, USA (Nov 9–12, 2019)
19. Döttling, N., Garg, S., Hajiabadi, M., Masny, D.: New constructions of identity-based and key-dependent message secure encryption schemes. In: Abdalla, M., Dahab, R. (eds.) PKC 2018: 21st International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 10769, pp. 3–31. Springer, Heidelberg, Germany, Rio de Janeiro, Brazil (Mar 25–29, 2018)
20. Döttling, N., Garg, S., Ishai, Y., Malavolta, G., Mour, T., Ostrovsky, R.: Trapdoor hash functions and their applications. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019, Part III. Lecture Notes in Computer Science, vol. 11694, pp. 3–32. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2019)
21. Garg, S., Gay, R., Hajiabadi, M.: New techniques for efficient trapdoor functions and applications. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, Part III. Lecture Notes in Computer Science, vol. 11478, pp. 33–63. Springer, Heidelberg, Germany, Darmstadt, Germany (May 19–23, 2019)
22. Garg, S., Hajiabadi, M., Mahmoody, M., Rahimi, A.: Registration-based encryption: Removing private-key generator from IBE. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018: 16th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 11239, pp. 689–718. Springer, Heidelberg, Germany, Panaji, India (Nov 11–14, 2018)
23. Garg, S., Hajiabadi, M., Mahmoody, M., Rahimi, A., Sekar, S.: Registration-based encryption from standard assumptions. In: Lin, D., Sako, K. (eds.) PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 11443, pp. 63–93. Springer, Heidelberg, Germany, Beijing, China (Apr 14–17, 2019)

24. Garg, S., Srinivasan, A.: Garbled protocols and two-round MPC from bilinear maps. In: Umans, C. (ed.) 58th Annual Symposium on Foundations of Computer Science. pp. 588–599. IEEE Computer Society Press, Berkeley, CA, USA (Oct 15–17, 2017)
25. Garg, S., Srinivasan, A.: Two-round multiparty secure computation from minimal assumptions. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part II. Lecture Notes in Computer Science, vol. 10821, pp. 468–499. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018)
26. Goyal, R., Vusirikala, S.: Verifiable registration-based encryption. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2020, Part I. pp. 621–651. Lecture Notes in Computer Science, Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2020)
27. Goyal, R., Vusirikala, S., Waters, B.: New constructions of hinting PRGs, OWFs with encryption, and more. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2020, Part I. pp. 527–558. Lecture Notes in Computer Science, Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2020)
28. Green, M.: (2019), https://blog.cryptographyengineering.com/2019/12/08/on-client-side-media-scanning/
29. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 339–358. Springer, Heidelberg, Germany, St. Petersburg, Russia (May 28 – Jun 1, 2006)
30. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) Advances in Cryptology – EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 415–432. Springer, Heidelberg, Germany, Istanbul, Turkey (Apr 13–17, 2008)
31. Hazay, C., Venkitasubramaniam, M.: Scalable multi-party private set-intersection. In: Fehr, S. (ed.) PKC 2017: 20th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 10174, pp. 175–203. Springer, Heidelberg, Germany, Amsterdam, The Netherlands (Mar 28–31, 2017)
32. Hubacek, P., Wichs, D.: On the communication complexity of secure function evaluation with long output. In: Roughgarden, T. (ed.) ITCS 2015: 6th Conference on Innovations in Theoretical Computer Science. pp. 163–172. Association for Computing Machinery, Rehovot, Israel (Jan 11–13, 2015)
33. Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007: 4th Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 4392, pp. 575–594. Springer, Heidelberg, Germany, Amsterdam, The Netherlands (Feb 21–24, 2007)
34. Jarecki, S., Liu, X.: Fast secure computation of set intersection. In: Garay, J.A., Prisco, R.D. (eds.) SCN 10: 7th International Conference on Security in Communication Networks. Lecture Notes in Computer Science, vol. 6280, pp. 418–435. Springer, Heidelberg, Germany, Amalfi, Italy (Sep 13–15, 2010)
35. Kissner, L., Song, D.X.: Privacy-preserving set operations. In: Shoup, V. (ed.) Advances in Cryptology – CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 241–257. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2005)
36. Kolesnikov, V., Matania, N., Pinkas, B., Rosulek, M., Trieu, N.: Practical multi-party private set intersection from symmetric-key techniques. In: Thuraisingham,

B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017: 24th Conference on Computer and Communications Security. pp. 1257–1272. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017)

37. Lindell, Y., Nissim, K., Orlandi, C.: Hiding the input-size in secure two-party computation. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology – ASIACRYPT 2013, Part II. Lecture Notes in Computer Science, vol. 8270, pp. 421–440. Springer, Heidelberg, Germany, Bengalore, India (Dec 1–5, 2013)
38. Pinkas, B., Schneider, T., Weinert, C., Wieder, U.: Efficient circuit-based PSI via cuckoo hashing. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part III. Lecture Notes in Computer Science, vol. 10822, pp. 125–157. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018)
39. Pinkas, B., Schneider, T., Zohner, M.: Faster private set intersection based on OT extension. In: Fu, K., Jung, J. (eds.) USENIX Security 2014: 23rd USENIX Security Symposium. pp. 797–812. USENIX Association, San Diego, CA, USA (Aug 20–22, 2014)
40. Quach, W., Wee, H., Wichs, D.: Laconic function evaluation and applications. In: Thorup, M. (ed.) 59th Annual Symposium on Foundations of Computer Science. pp. 859–870. IEEE Computer Society Press, Paris, France (Oct 7–9, 2018)
41. Resende, A.C.D., Aranha, D.F.: Faster unbalanced private set intersection. In: Meiklejohn, S., Sako, K. (eds.) FC 2018: 22nd International Conference on Financial Cryptography and Data Security. Lecture Notes in Computer Science, vol. 10957, pp. 203–221. Springer, Heidelberg, Germany, Nieuwpoort, Curaçao (Feb 26 – Mar 2, 2018)
42. Rindal, P., Rosulek, M.: Malicious-secure private set intersection via dual execution. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017: 24th Conference on Computer and Communications Security. pp. 1229–1242. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017)
43. Thyagarajan, S.A.K., Bhat, A., Malavolta, G., Döttling, N., Kate, A., Schröder, D.: Verifiable timed signatures made practical. In: ACM CCS 20: 27th Conference on Computer and Communications Security. pp. 1733–1750. ACM Press (2020)