Improved OT Extension for Transferring Short Secrets

Vladimir Kolesnikov[1] and Ranjit Kumaresan[2]

[1] Bell Labs, Murray Hill, NJ 07974, USA
kolesnikov@research.bell-labs.com
[2] Technion, Haifa, Israel
ranjit@cs.technion.ac.il

Abstract. We propose an optimization and generalization of OT extension of Ishai et al. of Crypto 2003. For computational security parameter k, our OT extension for short secrets offers $O(\log k)$ factor performance improvement in communication and computation, compared to prior work. In concrete terms, for today's security parameters, this means approx. factor 2-3 improvement.
This results in corresponding improvements in applications relying on such OT. In particular, for two-party semi-honest SFE, this results in $O(\log k)$ factor improvement in communication over state of the art Yao Garbled Circuit, and has the same asymptotic complexity as the recent multi-round construction of Kolesnikov and Kumaresan of SCN 2012. For multi-party semi-honest SFE, where their construction is inapplicable, our construction implies $O(\log k)$ factor communication and computation improvement over best previous constructions. As with our OT extension, for today's security parameters, this means approximately factor 2 improvement in semi-honest multi-party SFE.
Our building block of independent interest is a novel IKNP-based framework for 1-out-of-n OT extension, which offers $O(\log n)$ factor performance improvement over previous work (for $n \leq k$), and concrete factor improvement of up to 5 for today's security parameters (n=k=128).
Our protocol is the first practical OT with communication/computation cost sublinear in the security parameter (prior sublinear constructions Ishai et al. [15, 16] are not efficient in concrete terms).

Keywords: OT extension, 1-out-of-2 OT, 1-out-of-n OT.

1 Introduction

Our main contribution is an asymptotic and concrete efficiency improvement of Oblivious Transfer (OT) extension of Ishai et al. [14]. Our improvement applies to OT transfers of short secrets. In this Introduction we first motivate the problem, and then give intuition behind our approach.

Oblivious Transfer (OT) is a fundamental cryptographic primitive that is used as a building block in a variety of cryptographic protocols. It is a critical

piece in general secure computation [29, 10, 18], as well as in a number of tailored solutions to specific problems of interest, such as contract signing [7]. OT performance improvement directly translates into that of secure function evaluation (SFE). In turn, SFE performance is the subject of major research effort in cryptography [14, 22, 20, 6, 12, 25]. Our work can be plugged into several existing candidate solutions, resulting in factor 2-3 performance improvement, which is a major step forward in the state of the art of secure computation.

1.1 Secure Computation

SFE allows two (or more) parties to evaluate any function on their respective inputs x and y, while maintaining privacy of both x and y. SFE is justifiably a subject of an immense amount of research. Efficient SFE algorithms enable a variety of electronic transactions, previously impossible due to mutual mistrust of participants. Examples include auctions, contract signing, set intersection, etc. As computation and communication resources have increased, SFE of many useful functions has become practical for common use. Still, SFE of many of today's functions of interest carries costs sufficient to deter would-be adopters, who instead choose stronger trust models, entice users to give up their privacy with incentives, or use similar crypto-workarounds. We believe that truly practical efficiency is required for SFE to see use in real-life applications.

The current state of the art of SFE research is quite sophisticated. Particularly in the semi-honest model, there have been very few asymptotic/qualitative improvements since the original protocols of Yao [29] and Goldreich et al. [9]. Possibly the most important development in the area of SFE since the 1980's was the very efficient OT extension technique of Ishai et al. [14], which allowed to evaluate an arbitrarily large number of OTs by executing a small (security parameter) number of (possibly inefficient) "bootstrapping" OT instances, and a number of symmetric key primitives. This possibility of cheap OTs made a dramatic difference for securely computing functions with large inputs relative to the size of the function, as well as for GMW-like approaches, where OTs are performed in each level of the circiut.

As secure computation moves from theory to practice, even "small" improvements can have a significant effect. Today, even small factor performance improvements to state-of-the-art algorithms are quite hard to achieve, and are most welcome. This is especially true about the semi-honest model protocols, where the space for improvement appears to be much smaller than in the malicious model.

In this work, we propose an improvement to OT extension of Ishai et al. [14], for the case of OT of short secrets. As we will describe below, this will result in a new multi-party SFE protocol, which is approximately factor 2 (asymptotically factor $O(\log k)$) more efficient than state of the art. Our constructions also improve on standard two-party garbled circuit protocols in asymptotic ($O(\log k)$) and concrete terms, and offer performance in line with the recent work of [19].

1.2 Secure Computation and OT Efficiency Considerations

The efficiency of OT plays a critical role in the overall efficiency of secure computation. It is so to the point that OT performance determines which is the most efficient approach. Until recently, in the semi-honest model, Yao's Garbled Circuit was a clear winner. With the work of [19], which can be seen as a hybrid between GMW and Yao, and our improved OT extension technique, the GMW approach will outperform Yao with a factor of ≈ 2 for today's security parameters. Asymptotically, the performance improvement is logarithmic in the security parameter, as compared to GC-based SFE.

On the cost of SFE rounds. One common consideration in SFE protocol design is the number of rounds. Indeed, in some scenarios the latency associated with the communication rounds can more than double the total execution time. This holds, e.g., when the evaluated circuit is small; with the GMW evaluation, where we need a round of communication per layer of the circuit, the latency may be costly for deep and narrow circuits. This may cause somewhat increased latency of an individual computation – a possible inconvenience to the user of interactive applications.

At the same time, many SFE protocols allow for significant precomputation and streaming, where message transmission may begin (and even a response may be received) before the sender completes the computation and transmission of the message. Thus, round-related latency will usually not be a wasted time and will not cause extra delays. Most importantly, with the speed of the CPU advancing faster than that of communication, the true bottleneck for SFE already is the channel transmission capacity, even for high-speed gigabit LAN.

Thus, we argue that in many scenarios, the number of communication rounds in SFE often plays an insignificant role in practice, and round-related latency either has no impact on performance, or it can be tolerated in exchange of achieving higher throughput.

1.3 Our Contributions

Our main contribution is an asymptotic and concrete efficiency improvement of Oblivious Transfer (OT) extension of Ishai et al. [14]. Our improvement applies to OT transfers of short secrets.

1-out of-2 OT extension. For a security parameter k, our $O(\log k)$ asymptotic improvement results in concrete efficiency improvement of about factor up to 2 for today's security parameters. This yields corresponding asymptotic and concrete improvements in multi-party computation in the semi-honest setting, when applied to state of the art solutions based on GMW protocols.

Our new **1-out of-n OT extension** protocol offers $O(\log n)$ factor performance improvement over previous work (for $n < k$ and constant secret length), and concrete factor improvement of up to 5 for today's security parameters.

Further, our protocol is the first OT sublinear in the security parameter other than the non-black-box construction of Ishai et al. [15], and is the only practical OT with this property. Our resulting secure computation protocols can also

be viewed as a significant improvement of the technique of [19], which offered logarithmic in k improvement over state-of-the-art Yao's GC, but, in particular, did not extend to multiparty setting. We work in the (non-programmable) RO model, but, like in [14], we can also use a variant of correlation-robust hash functions.

We also present a new simple trick for OT extension (compatible with ours as well as with[14]), which, in particular, allows to futher cut in half the cost of OT of 1-bit secrets and reduce by 25% the cost OT of k-bit secrets. This optimization is described in Section 6 and Appendix A. To clearly state the performance improvement of our main OT extension protocol, the numbers elsewhere *do not* reflect this optimization.

Applications and Practical Performance Impact. As noted above, our 1-out of-2 OT construction immediately offers approximately factor 2 improvement in nearly all multi-party protocols – GMW and its variants.

In two-party computation, a similar, but more limited in scope, improvement was recently achieved [19]. In particular, [19] didn't work well on very shallow circuits, such as inner product computation. For such circuits, we have $O(\log k)$ improvement over 2PC state of the art, including [19].

More importantly, there is growing evidence that new GMW optimizations will often allow (multiround) GMW-based SFE protocols to outperform (constant round) Yao GC based SFE in practice, despite the round-related latencies. For example, a recent work of Schneider and Zohner [28] introduces and implements several optimizations to mitigate latency impact. It demonstrates performance improvement *of factor up to 100* of GMW over a recent Yao-based implementation of secure face matching even in high-latency (100ms round-trip, intercontinental) network. We expect that future SFE research and CPU-vs-network evolution will further improve GMW relative to Yao.

In sum, our work improves state of the art of 2PC computation for a significant class of problems where GMW protocols outperform Yao.

As noted, our 1-out of-n OT gives logarithmic performance improvement in transferring one in n random secret keys. However, in some cases, where the OT of specific secrets is required, the improvement factor may be smaller due to the fact that all n secrets encrypted with the n keys need to be transferred. In this case, logarithmic improvement applies only to the offline phase, where the secrets are not available.

Another application which immediately benefits from this work is string-selection OT (SOT), a variant of 1-out of-n OT and a building block of [19]. In SOT, the receiver selects one of the sender's two secrets based on his $\log n$-bit selection string.

1.4 Related Work

OT is a critical and heavily used component in much of cryptography, and in particular in secure computation protocols. Naturally, a lot of effort went into

optimizing its performance. Unfortunately, there are fundamental limits to OT efficiency. Impagliazzo and Rudich [13] showed that a black-box reduction from oblivious transfer to a one-way function or a one-way permutation would imply $\mathbf{P} \neq \mathbf{NP}$. It is further not known whether such non-black-box reductions exist.

Beaver [3] was the first to propose OT extension, a non-black-box scheme where a large number of OTs can be obtained from a small number of OTs (possibly executed by using public-key primitives) and one-way functions. Lindell and Zarosim [21] recently showed that one-way functions are in fact needed for OT extension.

Ishai, Kilian, Nissim, and Petrank [14], in their breakthrough work showed a truly practical black-box OT extension. Its cost, in addition to the security parameter number of base OTs, is only two random oracle (RO) evaluations and output transfers. By dramatically changing the cost structure of two-party SFE, especially in the semi-honest model, this work enabled greatly improved SFE for functions with large inputs, previously considered too costly due to the need of a large number of public key operations. It also started a rise in the study of GMW-based SFE protocols, where an OT is needed per multiplicative node. Indeed, recent (yet unoptimized) GMW-based and multiple-round protocols began to outperform traditional GC protocols. In particular, [25] outperforms state-of-the-art GC protocols in the malicious model, and [19] outperforms state-of-the-art GC protocols in the semi-honest model. In addition to considering the semi-honest model, [14] presents a construction secure against malicious participants. In a few follow-up works [24, 11, 17], the performance of the malicious setting of the IKNP OT extension was substantially improved. We present the high-level idea of the basic IKNP construction in Section 3.2.

By employing a more efficient pseudorandom generator in Beaver's non-black-box OT extension protocol, Ishai, Kushilevitz, Ostrovsky, and Sahai [15] obtained an asymptotically more efficient (but expensive in concrete terms) construction for oblivious transfer extension, and consequently for secure computation. In fact, their protocol enjoys a *constant computational/communication overhead* over an insecure evaluation of the function to be evaluated. In order to obtain these strong efficiency results, Ishai et al. [15] make strong complexity-theoretic assumptions on pseudorandom generators. Specifically, they assume that there exists an (arbitrary stretch) pseudorandom generator in $\mathbf{NC}^0$ [2, 1].

In this work, we show logarithmic in the security parameter improvement for black-box OT extension transfer of short secrets. In other words, we improve efficiency of the black-box OT extension protocol of Ishai et al. [14] asymptotically by a $\log(k/\ell)$ factor when the length of the transferred secrets is ℓ. This has important practical applications for secure computation solutions in the semihonest model, such as GMW, that require precisely 1-out-of-2 OT of 1-bit secrets. We calculate both asymptotic and concrete performance of the resulting protocols. Our constructions are presented in the semi-honest model.

We stress that in contrast to the non-black-box techniques of Ishai et al. [15], our extension protocol makes only black-box use of a (non-programmable) random oracle. Also, unlike [15] who mainly focus on asymptotic complexity, we

calculate also the concrete efficiency of our construction, and demonstrate a factor of approximately 2 improvement over state-of-the-art protocols [14, 6].

Finally, we mention PIR work (e.g., [16]) that construct communication efficient 1-out of-n OT protocols but perform $O(n)$ computationally intensive (e.g., public-key operations) per instance. In contrast, we perform a fixed number of public-key operations independent of the number of OT instances.

2 Overview of Our Approach

We give a high-level overview of our solution prior to presenting its technical details in Section 4. We aim that the reader somewhat familiar with the IKNP construction [14] should understand the main idea of our construction from this overview.

Consider the random $m \times k$ matrix designed by [14], which is transferred column-wise via k 1-out-of-2 base OTs from the receiver $\mathcal{R}$ to the sender $\mathcal{S}$. In [14], each row of this matrix is used to implement a 1-out-of-2 OT, as it has the randomness from which a random OT can be constructed.

Our main observation is that, *for the same communication cost*, each row of this matrix can be instead used to perform a 1-out-of-n OT, but of shorter secrets. Further, a 1-out-of-n OT of $\log n$-bit long secrets can be trivially used to construct $\log n$ instances of 1-out-of-2 1-bit OTs, which is precisely the kind of OT needed in the GMW protocol and its variants. Thus, effectively, we trade the length of the OT-transferred secrets for the number of OTs, which results in significant gain for MPC applications.

The intuition for our 1-out-of-n OT is as follows. First, recall that in IKNP, for each column of the $m \times k$ matrix, $\mathcal{S}$ randomly selects (via OT), whether he receives the random column, or the random column XORed with the m-bit long input of $\mathcal{R}$. Viewed row-wise, this effectively means that for each row j, $\mathcal{S}$ either receives (via OT) the j-th row of the randomly chosen $m \times k$ matrix (if $\mathcal{R}$'s j-th selection bit is 0), or that row XORed with his k-bit selection vector to the OT (if $\mathcal{R}$'s j-th selection bit is 1). Then $\mathcal{S}$ masks each of his two j-th input secrets with (RO hashes of) vector received as output from OT and the same vector XORed with its k-bit selection vector respectively and sends both to $\mathcal{R}$, who is able to take the mask off exactly one of the two messages. The other masked message remains hidden since $\mathcal{R}$ does not learn the selection vector provided by $\mathcal{S}$.

In the following, let $\mathcal{C}$ denote a binary code, and let r_j denote the input of $\mathcal{R}$ to the j-th instance of 1-out-of-n OT. In our 1-out-of-n OT, we modify the scheme presented above such that for each row j, $\mathcal{S}$ receives (via OT) the actual j-th row of the $m \times k$ matrix XORed with a vector that is the result of the r_j-th codeword in $\mathcal{C}$ bitwise-ANDed with the k-bit selection vector. This allows $\mathcal{S}$ to generate n random pads from each row of the matrix—the i-th such pad being the j-th row it received (via OT) XORed with a vector that is the result of the i-th codeword in $\mathcal{C}$ bitwise-ANDed with the k-bit selection vector. These n random pads may then be used by $\mathcal{S}$ to carry out a 1-out-of-n OT with $\mathcal{R}$.

The security of this construction naturally depends on the underlying code. The exact property that we need is that $\mathcal{C}$ must contain at least n codewords, each of length at most k, such that the codewords in $\mathcal{C}$ are spaced as far apart as possible from each other. This, combined with the fact that $\mathcal{R}$ does not learn the selection vector provided by $\mathcal{S}$, will ensure that $\mathcal{R}$ can efficiently recover only one of the n pads used by $\mathcal{S}$. The above is presented in detail in Section 4.

Using Walsh-Hadamard code for $\mathcal{C}$ gives a 1-out-of-n OT for n equal to the security parameter k. This OT is suitable for generation of $\log n$ instances of 1-out-of-2 OTs (Section 5.1). Using a higher-rate code with high distance results in 1-out-of-n OT for any n polynomial in k (Section 5.3).

3 Preliminaries and Notation

3.1 Notation

We use the notation OT_ℓ^m to denote m instances of 1-out-of-2 string-OT where the string is ℓ bits long. Let $\mathcal{S}$ denote the sender, and let $\mathcal{R}$ denote the receiver. In 1-out-of-2 OT, the sender's input is $\{(x_{j,0}, x_{j,1})\}_{j\in[m]}$, i.e., m pairs of strings, each of length ℓ, and the receiver holds input $\{r_j\}_{j\in[m]}$, where each r_j is an integer which is either 0 or 1. Note that if $\mathcal{S}$ provides input $\{(x_{j,0}, x_{j,1})\}_{j\in[m]}$ to OT_ℓ^m, and if $\mathcal{R}$ provides input $\{r_j\}_{j\in[m]}$ to OT_ℓ^m, then $\mathcal{R}$ receives back $\{x_{j,r_j}\}_{j\in[m]}$, while $\mathcal{S}$ receives nothing.

In Section 4, we construct protocols for 1-out-of-n OT, which is a straightforward generalization of 1-out-of-2 OT. We explain this further. We use the notation $\binom{n}{1}\text{-OT}_\ell^m$ to denote m instances of 1-out-of-n string-OT where the string is ℓ bits long. In 1-out-of-n OT, the sender's input is $\{(x_{j,0}, \ldots, x_{j,n-1})\}_{j\in[m]}$, and the receiver holds input $\{r_j\}_{j\in[m]}$, where each r_j is an integer which between 0 and $n-1$. Note that if $\mathcal{S}$ provides input $\{(x_{j,0}, \ldots, x_{j,n-1})\}_{j\in[m]}$ to $\binom{n}{1}\text{-OT}_\ell^m$, and if $\mathcal{R}$ provides input $\{r_j\}_{j\in[m]}$ to $\binom{n}{1}\text{-OT}_\ell^m$, then $\mathcal{R}$ receives back $\{x_{j,r_j}\}_{j\in[m]}$, while $\mathcal{S}$ receives nothing.

Following the convention in IKNP, we denote vectors in bold, and matrices in capitals. For a matrix A, we let $\mathbf{a}_j$ denote the j-th row of A, and $\mathbf{a}^i$ denote the i-th column of A. If $\mathbf{a} = a_1 \| \cdots \| a_p$ and $\mathbf{b} = b_1 \| \cdots \| b_p$ are two vectors, then we define $\oplus$ and $\odot$ operations as follows. We use the notation $\mathbf{a} \oplus \mathbf{b}$ to denote the vector $(a_1 \oplus b_1) \| \cdots \| (a_p \oplus b_p)$. Similarly, the notation $\mathbf{a} \odot \mathbf{b}$ denotes the vector $(a_1 \cdot b_1) \| \cdots \| (a_p \cdot b_p)$. Finally, suppose $c \in \{0, 1\}$, then $c \cdot \mathbf{a}$ denotes the vector $(c \cdot a_1) \| \cdots \| (c \cdot a_p)$.

Our constructions assume the existence of a random oracle H. We denote the security parameter by k, and assume (without loss of generality) that it is a power of 2.

3.2 IKNP OT Extension

In this section, we present the OT extension protocol of Ishai, Kilian, Nissim, and Petrank [14]. The protocol will reduce OT_ℓ^m to OT_m^k. This implies a reduction

(via use of a PRG) to OT_k^k with some additional cost. The security of the protocol holds as long as the receiver is semi-honest. (Note: the sender may be malicious.)

We now describe the protocol that realizes OT_ℓ^m given ideal access to OT_m^k.

INPUT OF $\mathcal{S}$: m pairs $(x_{j,0}, x_{j,1})$ of ℓ-bit strings, $1 \le j \le m$.
INPUT OF $\mathcal{R}$: m selection bits $\mathbf{r} = (r_1, \ldots, r_m)$.
COMMON INPUT: a security parameter k.
ORACLE: a random oracle $H : [m] \times \{0,1\}^k \to \{0,1\}^\ell$.
CRYPTOGRAPHIC PRIMITIVE: an ideal OT_m^k primitive.

1. $\mathcal{S}$ chooses $\mathbf{s} \leftarrow \{0,1\}^k$ at random. Let s_i denote the i-th bit of $\mathbf{s}$.
2. $\mathcal{R}$ forms $m \times k$ matrices T_0, T_1 in the following way:
   - Choose $\mathbf{t}_{j,0}, \mathbf{t}_{j,1} \leftarrow \{0,1\}^k$ at random such that $\mathbf{t}_{j,0} \oplus \mathbf{t}_{j,1} = (r_j \| \cdots \| r_j)$.

   Let $\mathbf{t}_0^i, \mathbf{t}_1^i$ denote the i-th column of matrices T_0, T_1 respectively.
3. $\mathcal{S}$ and $\mathcal{R}$ interact with OT_m^k in the following way:
   - $\mathcal{S}$ acts as *receiver* with input $\{s_i\}_{i \in [k]}$.
   - $\mathcal{R}$ acts as *sender* with input $\{\mathbf{t}_0^i, \mathbf{t}_1^i\}_{i \in [k]}$.
   - $\mathcal{S}$ receives output $\{\mathbf{q}^i\}_{i \in [k]}$.

   $\mathcal{S}$ forms $m \times k$ matrix Q such that the i-th column of Q is the vector $\mathbf{q}^i$. (Note $\mathbf{q}^i = t_{s_i}^i$.) Let $\mathbf{q}_j$ denote the j-th row of Q. (Note $\mathbf{q}_j = ((\mathbf{t}_{j,0} \oplus \mathbf{t}_{j,1}) \odot \mathbf{s}) \oplus \mathbf{t}_{j,0}$. Simplifying, $\mathbf{q}_j \oplus \mathbf{t}_{j,0} = r_j \cdot \mathbf{s}$.)
4. For $j \in [m]$, $\mathcal{S}$ sends $y_{j,0} = x_{j,0} \oplus H(j, \mathbf{q}_j)$ and $y_{j,1} = x_{j,1} \oplus H(j, \mathbf{q}_j \oplus \mathbf{s})$.
5. For $j \in [m]$, $\mathcal{R}$ recovers $z_j = y_{j,r_j} \oplus H(j, \mathbf{t}_{j,0})$.

EFFICIENCY. The protocol makes a single call to OT_m^k. The cost of OT_m^k is the cost of OT_k^k (which is independent of m) plus a generation of $2k$ pseudorandom strings each of length m. Other than this call to OT_m^k, each party evaluates at most $2m$ times (an implementation of) a random oracle. It is easy to see that the total communication cost of OT_ℓ^m is the communication cost of implementing OT_m^k plus $2m\ell$ bits transferred between $\mathcal{S}$ and $\mathcal{R}$ in Step 4. Thus we conclude that the communication cost of OT_ℓ^m is $2mk + 2m\ell$ bits. Note that the total computational cost of the protocol is proportional to its communication cost.

3.3 Walsh-Hadamard (WH) Codes

For $\alpha \in \{0,1\}^q$, let $\mathrm{WH}(\alpha) = (\langle \alpha, x \rangle)_{x \in \{0,1\}^q}$, where the inner product between the two vectors is taken modulo 2. That is, $\mathrm{WH}(\alpha)$, also known as the Walsh-Hadamard encoding of α, is the 2^q-bit string consisting of inner products of each q-bit string with α. For each k, Walsh-Hadamard codes, denoted by $\mathcal{C}_{\mathrm{WH}}^k$, are simply defined as the set $\{\mathrm{WH}(\alpha)\}_{\alpha \in \{0,1\}^{\log k}}$. Note that $\mathcal{C}_{\mathrm{WH}}^k$contains k strings (or, codewords) each of length k bits. In our constructions, we will use the well-known fact that the relative distance of $\mathcal{C}_{\mathrm{WH}}^k$ is $1/2$ when k is a power of 2.

4 Extending 1-out-of-n OT

Recall, k is a security parameter. We present a natural generalization of 1-out-of-2 OT extension protocol given in [14]. We consider 1-out-of-n OT for any $n \leq k$.[3] First, recall that it is easy to construct a 1-out-of-n OT protocol from $O(\log n)$ instances of a 1-out-of-2 OT protocol in the semi-honest setting. The communication cost of m instances of 1-out-of-n OT on ℓ-bit strings would be the cost of $\mathrm{OT}_k^{m \log n}$ plus the cost required to transmit at most mn masked secrets each of length ℓ. Thus, the communication cost of obtaining m instances of 1-out-of-n OT on ℓ-bit strings is at most $O(m(k \log n + n\ell))$ bits. Further, its computational cost is proportional to the communication cost.

Our main contribution, formally presented in this section, is showing how to generalize IKNP's technique to directly obtain (i.e., without going via a construction for 1-out-of-2 OT) an extension protocol for 1-out-of-n OT when $n \leq k$. For the same security parameter and the same size of setup matrix at IKNP, the concrete security of our construction corresponds to that provided by security parameter $k_{\mathrm{IKNP}} \approx k/2$. If exactly same concrete security as IKNP is desired, this can be achieved by setting our security parameter $k \approx 2k_{\mathrm{IKNP}}$, which results in a multiplicative factor 2 overhead compared to IKNP. However, because we do 1-out-of-n OT at this cost, our construction will still result in asymptotic and concrete performance improvement of 1-out-of-n OT.

Let $\binom{n}{1}$-OT_ℓ^m denote m instances of 1-out-of-n OT on ℓ-bit strings. As in [14], we will reduce $\binom{n}{1}$-OT_ℓ^m to OT_m^k(which can be trivially efficiently reduced to OT_k^k). As the [14] basic protocol, our protocol is secure against a malicious sender and semi-honest receiver. Our protocol will use Walsh-Hadamard codes, denoted by $\mathcal{C}_{\mathrm{WH}}^k = (\mathbf{c}_0, \ldots, \mathbf{c}_{k-1})$.

We now describe our protocol that realizes $\binom{n}{1}$-OT_ℓ^m given ideal access to OT_m^k.

Construction 1 (1-out-of-n OT Extension)
Input of $\mathcal{S}$: *m tuples $(x_{j,0}, \ldots, x_{j,n-1})$ of ℓ-bit strings, $1 \leq j \leq m$.*
Input of $\mathcal{R}$: *m selection integers $\mathbf{r} = (r_1, \ldots, r_m)$ such that $0 \leq r_j < n$ for $1 \leq j \leq m$.*
Common Input: *a security parameter k such that $k \geq n$, and Walsh-Hadamard codes $\mathcal{C}_{\mathrm{WH}}^k = (\mathbf{c}_0, \ldots, \mathbf{c}_{k-1})$.*
Oracle: *a random oracle $H : [m] \times \{0,1\}^k \to \{0,1\}^\ell$.*
Cryptographic Primitive: *an ideal OT_m^k primitive.*

1. *$\mathcal{S}$ chooses $\mathbf{s} \leftarrow \{0,1\}^k$ at random. Let s_i denote the i-th bit of $\mathbf{s}$.*
2. *$\mathcal{R}$ forms $m \times k$ matrices T_0, T_1 in the following way:*
   - *Choose $\mathbf{t}_{j,0}, \mathbf{t}_{j,1} \leftarrow \{0,1\}^k$ at random such that $\mathbf{t}_{j,0} \oplus \mathbf{t}_{j,1} = \mathbf{c}_{r_j}$.*

   Let $\mathbf{t}_0^i, \mathbf{t}_1^i$ denote the i-th column of matrices T_0, T_1 respectively.
3. *$\mathcal{S}$ and $\mathcal{R}$ interact with OT_m^k in the following way:*
   - *$\mathcal{S}$ acts as* receiver *with input $\{s_i\}_{i \in [k]}$.*

[3] We discuss how to extend 1-out-of-n OT for $n = \mathrm{poly}(k)$ in Section 5.3.

– $\mathcal{R}$ *acts as* sender *with input* $\{\mathbf{t}_0^i, \mathbf{t}_1^i\}_{i\in[k]}$.
– $\mathcal{S}$ *receives output* $\{\mathbf{q}^i\}_{i\in[k]}$.

$\mathcal{S}$ forms $m\times k$ matrix Q such that the i-th column of Q is the vector $\mathbf{q}^i$. (Note $\mathbf{q}^i = \mathbf{t}_{s_i}^i$.) Let $\mathbf{q}_j$ denote the j-th row of Q. (Note $\mathbf{q}_j = ((\mathbf{t}_{j,0}\oplus\mathbf{t}_{j,1})\odot\mathbf{s})\oplus\mathbf{t}_{j,0}$. Simplifying, $\mathbf{q}_j\oplus\mathbf{t}_{j,0} = \mathbf{c}_{r_j}\odot\mathbf{s}$.)

4. *For $j\in[m]$ and for every $0\le r<n$, $\mathcal{S}$ sends $y_{j,r} = x_{j,r}\oplus H(j, \mathbf{q}_j\oplus(\mathbf{c}_r\odot\mathbf{s}))$.*
5. *For $j\in[m]$, $\mathcal{R}$ recovers $z_j = y_{j,r_j}\oplus H(j,\mathbf{t}_{j,0})$.*

This concludes the description of the protocol. It is easy to verify that the protocol's outputs are correct (i.e., $z_j = x_{j,r_j}$) when both parties follow the protocol.

Efficiency. The protocol makes a single call to OT_m^k. The cost of OT_m^k is the cost of OT_k^k (which is independent of m) plus a generation of $2k$ pseudorandom strings each of length m. Other than this call to OT_m^k, each party evaluates at most mn times (an implementation of) a random oracle. It is easy to see that the total communication cost of OT_ℓ^m is the communication cost of implementing OT_m^k plus $mn\ell$ bits transferred between $\mathcal{S}$ and $\mathcal{R}$ in Step 4. Thus we conclude that the communication cost of OT_ℓ^m is $O(m(k+n\ell))$ bits. Note that the total computational cost of the protocol is proportional to its communication cost. Recall that $n\le k$, and thus when $\ell = 1$, the asymptotic cost of our $\binom{n}{1}$-OT_ℓ^m protocol is $O(mk)$ which is the same as the asymptotic cost of Ishai et al.'s OT_ℓ^m protocol described in Section 3.2. In terms of concrete performance, as mentioned above, we need to use a security parameter $k\approx 2k_{\mathrm{IKNP}}$, resulting in a factor 2 overhead compared to IKNP's OT_ℓ^m execution. Because we are performing the more powerful $\binom{n}{1}$-OT_ℓ^m, this corresponds to asymptotic (and concrete!) performance improvement.

Theorem 1. *Construction 1 is a secure protocol for evaluating $\binom{n}{1}$-OT_ℓ^m in the semi-honest model.*

The proof of security of Theorem 1 appears in the full version.

Remarks. In Construction 1, one can replace $\mathcal{C}_{\mathrm{WH}}^k$ with an encoding map $\mathsf{enc} : \{0,1\}^{\log n}\to\{0,1\}^k$ that has the property that for $r, r'\in\{0,1\}^{\log n}$ with $r\neq r'$, the Hamming distance between $\mathsf{enc}(r)$ and $\mathsf{enc}(r')$ is at least $\Omega(k)$. It is instructive to see that when $n=2$ and when enc is the k-bit *repetition* encoding of the input bit, i.e., $\mathsf{enc}(r) = (r,\ldots,r)\in\{0,1\}^k$, then we get exactly the IKNP construction. Note that for $r\neq r'$, the Hamming distance between $\mathsf{enc}(r)$ and $\mathsf{enc}(r')$ is exactly k. As we saw in Construction 1, using the encoding map $\mathsf{enc}(r) = \mathbf{c}_r$, where $\mathbf{c}_r$ is the r-th Walsh-Hadamard codeword, gives us an $\log k$ efficiency improvement. Since the Walsh-Hadamard code is a low-rate code, the maximum value of n is restricted to be less than or equal to k. A natural question that arises is whether a code with a better rate enables us to remove this restriction. Indeed, in Section 5.3, by using more sophisticated codes (cf. Claim 5.3) we show an improvement in the (offline) communication complexity of 1-out-of-n OT extension for arbitrary $n = \mathrm{poly}(k)$.

5 Resulting Efficiency Improvements

We evaluate performance improvements of Construction 1, and corresponding two- and multi-party SFE improvements. Recall that in the semi-honest model, a single instance of 1-out-of-n OT may be used to generate $\log n$ instances of 1-out-of-2 OT over slightly shorter strings with no additional cost. More precisely, the cost of OT_ℓ^m is exactly equal to the cost of $\binom{n}{1}$-$\mathrm{OT}_{\ell \log n}^{m/\log n}$. This observation will allow us to leverage our efficient construction of $\binom{n}{1}$-OT_ℓ^m to obtain improved efficiency for 1-out-of-2 OT, and consequently for secure computation.

5.1 Efficiency Improvements for 1-out-of-2 OT

In this section, we demonstrate a $\log k$ asymptotic improvement in the efficiency of 1-out-of-2 OT when sender's secrets are just bits (i.e., length of sender's secrets, $\ell = 1$). As observed previously, we do this by constructing 1-out-of-2 OTs via 1-out-of-n OTs.

Recall that the cost of our $\binom{n}{1}$-OT_ℓ^m protocol described in Section 4 is $O(m(k + n\ell))$. Using the fact that the cost of OT_ℓ^m is exactly equal to the cost of $\binom{n}{1}$-$\mathrm{OT}_{\ell \log n}^{m/\log n}$, we conclude that OT_ℓ^m may be reduced to OT_k^k while incurring an additional cost at most $O((m/\log n) \cdot (k + n\ell \log n))$. By choosing n such that $n \log n = k/\ell$, we see that this additional cost is asymptotically $O(mk/\log(k/\ell))$. In summary, we have shown a reduction from OT_ℓ^m to OT_k^k with cost $O(mk/\log(k/\ell))$.

Contrast our result above with the result of [14], where the cost of the reduction from OT_ℓ^m to OT_k^k was $O(m(k + \ell))$. Observe that for the important case when $\ell = 1$, our construction offers a logarithmic factor improvement in the efficiency of the reduction.

As noted in Section 4, to achieve concrete security equal to that of IKNP, we need a security parameter approximately twice theirs, which results in a factor 2 overhead of our protocol. Even with this efficiency loss we have both asymptotic and concrete performance advantage over IKNP.

Concrete Efficiency. We begin with a concrete cost analysis of $\binom{n}{1}$-OT_ℓ^m. Recall that the exact cost of reduction from OT_m^k to OT_k^k involves sending $2mk$ bits. Then, in Step 4 of Construction 1, $\mathcal{S}$ transmits $mn\ell$ bits to $\mathcal{R}$. Thus, the concrete cost of $\binom{n}{1}$-OT_ℓ^m is $m(2k+n\ell)$. Using the fact that the cost of our OT_ℓ^m is exactly equal to the cost of $\binom{n}{1}$-$\mathrm{OT}_{\ell \log n}^{m/\log n}$, we conclude that OT_ℓ^m may be reduced to OT_k^k with cost $(m/\log n) \cdot (2k + n\ell \log n)$ bits. The minimum cost can then be obtained by choosing a suitable value of n.

In contrast, the concrete communication cost of IKNP's construction of OT_ℓ^m is $2m(k + \ell)$ bits. As described earlier, there's a small gap between the security guarantees between our consruction and IKNP's. We take that into account in our cost calculation, and present the results in Table 1.

level of security	our cost	IKNP cost
50	74	102
112	130	226
238	227	478

Table 1. Comparison of (amortized) communication cost (measured in bits) of 1-out-of-2 bit OT for a given security level. The costs are computed assuming parties are semi-honest. The performance improvement ratio betwen our work and IKNP represents the resulting improvement factor for MPC protocols based on the GMW approach.

5.2 Efficiency Improvements for Secure Computation

In this section, we will discuss applications of our OT_ℓ^m protocol to secure two-party and multi-party computation. As pointed out in the Introduction, efficient OT forms a criticial component of secure computation protocols, and improvements in the efficiency of OT translates to an improvement in the efficiency of secure computation protocols built on top of OT.

In the previous section, we saw how our construction asymptotically outperforms the extension protocol of [14] by a factor of $O(\log(k/\ell))$. Clearly, this improvement factor is maximized when $\ell = 1$, i.e., for 1-bit OT. Thus, our construction has maximum benefit for secure computation protocols that extensively rely on 1-bit OTs. One such example is the well known GMW protocol [9] where each AND gate of the circuit is evaluated using (two invocations of) 1-bit OTs (and negligible additional cost). Until now, efficient implementations of the GMW protocol in the semi-honest setting (e.g., [6]) relied on the OT extension protocol of [14]. Because OT costs dominate the protocol costs, simply by using our extension protocol (instead of [14]), the semi-honest GMW protocol will enjoy an asymptotic $\log k$ efficiency improvement (and improvement in concrete terms as well).

Secure Two-Party Computation. As discussed in Section 1.3, a large class of 2PC problems is solved more efficiently with GMW than Yao. For problems in this class, our OT extension improvement results in corresponding 2PC improvement. For other problems, where Yao is faster, the relative performance of the approaches is discussed next.

The concrete improvements for the specific case of two-party computation are shown in Table 2. From the table, it is evident that our protocol begins to outperform state-of-the-art constant round protocols (e.g., [26]) for reasonable levels of security. However, for practical values of the security parameter, it performs worse, in concrete terms, when compared to the best-case performance of [19], a non-constant round protocol that generalizes both Yao garbled circuits and GMW. (We note that the communication cost of our protocol is asymptotically the same as the communication cost of [19].) In more detail, the performance of [19] is highly sensitive to the topology of the circuit. Their best performance, as noted in Table 2, is for the case of constant width circuits. In contrast, our improvements are independent of the topology of the circuit being evaluated. We

point out that the approach of [19] can be viewed as somewhat related to ours (but more narrow; in particular, it is not applicable to multiparty comptuation). Furthermore, our OT extension protocol can improve the performance of [19] for circuits with low depth.

level of security	our cost per gate	[26] cost per gate	[19] cost per gate
50	148	100	66
112	260	224	112
238	454	476	196

Table 2. Comparison of (amortized) communication cost (measured in bits) per gate of the circuit for various semi-honest secure two-party protocols. We note that protocols of [19] do not extend to multi-party setting, while ours do.

Secure Multi-Party Computation. Today, practical protocols for secure *multi-party* computation are based on the GMW approach (e.g., [25, 6]).[4] GMW-based secure computation protocols for t parties, in the semi-honest setting, operate in almost the same way as in the two-party case except that now parties compute pairwise OTs (more precisely, a total of $2t^2$ OTs) to securely evaluate each AND gate. That is, for each AND gate of the circuit parties evaluate a total of $2t^2$ 1-bit OTs (with negligible additional cost). Therefore, simply by using our extension protocol (instead of [14]), we will improve the asymptotic complexity by a $\log k$ factor. Concrete improvements in this setting are the same as those found in Table 1. Specifically, for "50-bit security" we obtain an improvement of $102/74 = 1.378$ in the communication cost. Similarly, we obtain an improvement factor of > 2 for "238-bit security".

5.3 Efficiency Improvements for 1-out-of-n OT

Recall that the cost of extending 1-out-of-n OT from [14] is $O(m(k \log n + n\ell))$ bits. Our main construction of 1-out-of-n OT described in Section 4 reduces the cost of 1-out-of-n OT extension to $m(2k + n\ell)$ bits. As described in Section 4, for the same guarantee as in IKNP, our security parameter should be set as $k \approx 2k_{\text{IKNP}}$. Previous solutions [23, 14] cost $(4mk_{\text{IKNP}} \log n + mn\ell)$ bits. Hence for $k_{\text{IKNP}} = 128$, with $n = k$ and $\ell = 1$, our solution improves upon existing solutions by a factor ≈ 5.39.

Note that the above improvement holds only when $n \leq k$. In this section, we show how to modify Construction 1 to support $n = \text{poly}(k)$. In the resulting protocol, the (offline) communication cost of the generating 1-out-of-n OT correlations will be $O(mk)$ bits, i.e., completely independent of n. This improves

[4] Yao GC-based approach does not seem to map naturally into the multiparty setting. This is true even for the three party semi-honest setting. A more complicated solution is possible [4], but much less practical than GMW-based approaches [6].

over the best known offline communication complexity (which was $O(mk \log n)$ bits).

The total complexity (i.e., both online and offline) of our construction will asymptotically outperform existing constructions only for $n \leq ck$ where c is an arbitrary constant. For $n = \omega(k)$, the online cost of our protocol $O(mn\ell)$ dominates the total cost, but is still as efficient as existing constructions.

The main idea of our construction is to replace $\mathcal{C}^k_{\mathrm{WH}}$ with a code from a family of linear error correcting codes with the following special properties. (Our claim below is taken verbatim from [16].)

Claim ([16, 5, 8]). There exists a finite field $\mathbb{F}$ of characteristic 2 and an efficiently constructible family of linear error-correcting codes $C_K : \mathbb{F}^K \to \mathbb{F}^{N_K}$ with the following properties: (1) $N_K = O(K)$; (2) The dual distance of C_K is $\delta_K = \Omega(K)$; (3) The linear code C'_K spanned by all pointwise-products of pairs of codewords in C_K has minimal distance $\Delta_K = \Omega(K)$ and supports efficient decoding of up to $\mu_K = \Omega(K)$ errors. (The pointwise product of $(c_1, \ldots, c_N)$ and $(c'_1, \ldots, c'_N)$ is $(c_1 c'_1, \ldots, c_N c'_N)$.)

The last property implies that C_K also has minimal distance $d_K = \Omega(K)$.

Setting $N_K = k$ and $K \geq \log n$ is enough to provide the desired improvements stated above. The security level provided by this construction will be $\log(2^{d_K}/n^2) = \Omega(k)$ for n polynomial in k.

6 Optimizing the Reduction from $\binom{n}{1}$-OT^m_ℓ to OT^k_k

In our OT extension protocol, the OT^k_m primitive is reduced to OT^k_k. Further, the roles of $\mathcal{R}$ and $\mathcal{S}$ are reversed in our application of the reduction in our protocol. We provide an optimization that exploits this fact. This optimization was independently discovered by us and by Schneider and Zohner [27].

The main idea of the optimization is that inside the OT extension protocol of IKNP (as well as our protocol) 1-out of-2 OT of very long (m-bit long) random-looking correlated strings is executed. We cut the communication almost in half by OT-sending a PRG seed used to generate the strings. In other words, we obtain efficiency improvements by employing pseudorandom additive sharing instead of a completely random additive sharing. Because the strings need to be correlated in a specific way, a "correction" string needs to be sent so that exactly the right secret is recovered.

Note that this technique can also be applied to the IKNP construction. Such an application would reduce the IKNP cost of OT^m_ℓ from $m(2k+2\ell)$ to $m(k+2\ell)$. Observe that the reduced costs also have an impact on the oblivious key transfer phase (by constant factor 4/3) of Yao-based constructions where $\ell = k$.

For the case of 1-out-of-2 1-bit OT extension with 160-bit security, we get an improvement factor of ≈ 3.15 over the protocol of [14], and an improvement factor of ≈ 1.5 over the optimized IKNP protocol. See Appendix A for a detailed description of the protocol. We stress that Tables 1 and 2 do not take into account the optimizations described in this section.

Acknowledgments. We thank Yuval Ishai for many useful discussions, and the anonymous referees for their comments. The second author would like to thank Rajsekar Manokaran for useful discussions.

The first author was supported in part by the Intelligence Advanced Research Project Activity (IARPA) via Department of Interior National Business Center (DoI / NBC) contract Number D11PC20194. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

The second author was supported by ERC-CaC, grant number 259426.

References

1. Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *44th Annual ACM Symposium on Theory of Computing*, pages 805–816. ACM Press, 2012.
2. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th Annual Symposium on Foundations of Computer Science*, pages 166–175. IEEE Computer Society Press, October 2004.
3. Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *28th Annual ACM Symposium on Theory of Computing*, pages 479–488. ACM Press, May 1996.
4. Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols. In *22nd Annual ACM Symposium on Theory of Computing*, pages 503–513. ACM Press, May 1990.
5. Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 521–536. Springer, August 2006.
6. Seung Geol Choi, Kyung-Wook Hwang, Jonathan Katz, Tal Malkin, and Dan Rubenstein. Secure multi-party computation of boolean circuits with applications to privacy in on-line marketplaces. In *Topics in Cryptology – CT-RSA 2012*, Lecture Notes in Computer Science, pages 416–432. Springer, 2012.
7. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. C. ACM, 28:637-647, 1985.
8. A. Garcia and H. Stichtenoth. On the asymptotic behavior of some towers of function fields over finite fields. Journal of Number Theory, 61(2):248-273, 1996.
9. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 218–229. ACM Press, May 1987.
10. Oded Goldreich and Ronen Vainish. How to solve any protocol problem - an efficiency improvement. In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, pages 73–86. Springer, August 1988.

11. Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. OT-combiners via secure computation. In *TCC 2008*.
12. Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure two-party computation using garbled circuits. USENIX Security Symposium, 2011.
13. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing*, pages 44–61. ACM Press, May 1989.
14. Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161. Springer, August 2003.
15. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 433–442. ACM Press, May 2008.
16. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Extracting correlations. In *50th Annual Symposium on Foundations of Computer Science*, pages 261–270. IEEE Computer Society Press, 2009.
17. Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer—efficiently. In CRYPTO 2008.
18. Joe Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31. ACM, 1988.
19. Vladimir Kolesnikov and Ranjit Kumaresan. Improved secure two-party computation via information-theoretic garbled circuits. In *8th Conference on Security and Cryptography for Networks*, pages 205–221, 2012.
20. Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In ICALP 2008, pages 486–498. Springer, July 2008.
21. Yehuda Lindell and Hila Zarosim. On the feasibility of extending oblivious transfer. In TCC, 2013.
22. Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. Fairplay - secure two-party computation system. USENIX Security Symposium, 2004.
23. Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18(1):1–35, January 2005.
24. Jesper Buus Nielsen. Extending oblivious transfers efficiently - how to get robustness almost for free. *IACR Cryptology ePrint Archive*, 2007:215, 2007.
25. Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 681–700. Springer, 2012.
26. Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams. Secure two-party computation is practical. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 250–267. Springer, December 2009.
27. Thomas Schneider and Michael Zohner. Private communication. 2012.
28. Thomas Schneider and Michael Zohner. GMW vs. Yao? efficient secure two-party computation with low depth circuits. In *FC 2013*.
29. Andrew Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science*, pages 162–167. IEEE Computer Society Press, October 1986.

A Optimizing the Reduction from $\binom{n}{1}$-OT_{ℓ}^{m} to OT_{k}^{k}

Construction 2 (Optimized 1-out-of-n OT Extension)
INPUT OF $\mathcal{S}$: *m tuples $(x_{j,0}, \ldots, x_{j,n-1})$ of ℓ-bit strings, $1 \leq j \leq m$.*
INPUT OF $\mathcal{R}$: *m selection integers $\mathbf{r} = (r_1, \ldots, r_m)$ such that $0 \leq r_j < n$ for $1 \leq j \leq m$.*
COMMON INPUT: *a security parameter k such that $k \geq n$, and Walsh-Hadamard codes $\mathcal{C}_{\mathrm{WH}}^{k} = (\mathbf{c}_0, \ldots, \mathbf{c}_{k-1})$.*
ORACLE: *random oracles $H : [m] \times \{0,1\}^k \to \{0,1\}^\ell$, and $G : \{0,1\}^k \to \{0,1\}^m$.*
CRYPTOGRAPHIC PRIMITIVE: *an ideal OT_{m}^{k} primitive.*

1. *$\mathcal{S}$ chooses $\mathbf{s} \leftarrow \{0,1\}^k$ at random. Let s_i denote the i-th bit of $\mathbf{s}$.*
2. *$\mathcal{R}$ forms a $(m \times k)$ matrix D by setting $\mathbf{d}_j = \mathbf{c}_{r_j}$. $\mathcal{R}$ then forms $m \times k$ matrices T_0, T_1 in the following way:*
   - *Set $\mathbf{t}_1^i = G(v_i)$ for a randomly chosen $v_i \leftarrow \{0,1\}^k$.*
   - *Set $\mathbf{t}_0^i = \mathbf{d}^i \oplus \mathbf{t}_1^i$.*

   In the above, $\mathbf{t}_0^i, \mathbf{t}_1^i$ denotes the i-th column of matrices T_0, T_1 respectively. (Note that T_0, T_1 form a pseudorandom sharing of the matrix D.)
3. *$\mathcal{S}$ and $\mathcal{R}$ interact with OT_{k}^{k} in the following way:*
   - *$\mathcal{S}$ acts as* receiver *with input $\{s_i\}_{i \in [k]}$.*
   - *$\mathcal{R}$ acts as* sender *with inputs $\{u_i, v_i\}_{i \in [k]}$, where each u_i is chosen uniformly at random from $\{0,1\}^k$. (Note v_i was already chosen by $\mathcal{R}$ in Step 2.)*
   - *$\mathcal{S}$ receives output $\{\mathbf{a}^i\}_{i \in [k]}$.*

   $\mathcal{S}$ forms $k \times k$ matrix A such that the i-th column of A is the vector $\mathbf{a}^i$.
4. *For each $i \in [k]$, $\mathcal{R}$ sends $w_i = G(u_i) \oplus \mathbf{t}_0^i$.*
5. *$\mathcal{S}$ forms $m \times k$ matrix Q such that*
   - *if $s_i = 0$, then $\mathbf{q}^i = w_i \oplus G(\mathbf{a}^i)$,*
   - *else if $s_i = 1$, then $\mathbf{q}^i = G(\mathbf{a}^i)$.*

   Let $\mathbf{q}_j$ denote the j-th row of Q. (Note $\mathbf{q}^i = \mathbf{t}_{s_i}^i$. Note $\mathbf{q}_j = ((\mathbf{t}_{j,0} \oplus \mathbf{t}_{j,1}) \odot \mathbf{s}) \oplus \mathbf{t}_{j,0}$. Simplifying, $\mathbf{q}_j \oplus \mathbf{t}_{j,0} = \mathbf{c}_{r_j} \odot \mathbf{s}$.)
6. *For $j \in [m]$ and for every $0 \leq r < n$, $\mathcal{S}$ sends $y_{j,r} = x_{j,r} \oplus H(j, \mathbf{q}_j \oplus (\mathbf{c}_r \odot \mathbf{s}))$.*
7. *For $j \in [m]$, $\mathcal{R}$ recovers $z_j = y_{j,r_j} \oplus H(j, \mathbf{t}_{j,0})$.*

The amortized cost per instance of the $\binom{n}{1}$-OT_{ℓ}^{m} protocol above is $(k + n\ell)$. This yields a OT_{ℓ}^{m} protocol whose amortized concrete cost per instance is $n\ell + (k/\log n)$ bits.