# Blazing Fast Actively Secure OT Extension Protocol

Arpita Patra, Ajith S and Pratik Sarkar
Indian Institute Of Science, Bangalore

## Abstract

Oblivious transfer (OT) is one of the most important building blocks in secure computation. Informally it is a two-party protocol between a sender and a receiver, where the sender holds a pair of strings and the receiver holds a selection bit. At the end of the protocol the receiver should learn just the selected string, and the sender should not gain any new information. Moreover, it is required that the above properties hold even if the sender or the receiver maliciously deviate from the protocol. By malicious, we mean that the party can arbitrarily deviate from the protocol. OT extension refers to the process of starting with a small number of base OTs, create many OTs at the expense of symmetric primitives. In this work, we present an OT extension protocol for the setting of malicious adversaries that is more efficient and uses less communication than previous works.

## Introduction

**Oblivious Transfer (OT) :** OT is a protocol between two parties: a sender and a receiver, where the sender holds a pair of strings and the receiver holds a selection bit. At the end of the protocol,

- The receiver should learn just the selected string.
- The sender should not gain any new information.

However, computing a large number of OTs is expensive since all known OTs are based on public-key primitives and such primitives are expensive.

**OT Extension :** A primitive that can generate a large number of OTs using a small number of OTs (henceforth referred as base OTs or seed OTs) and relying on some extra cheap operations. Surprisingly, Beaver (STOC 1996)[2] showed that it is possible to obtain *poly(n)* OTs from *n* OT calls and using one-way functions. In addition, the author showed that it is impossible to extend OTs information theoretically. Another groundbreaking work was proposed by Ishai et al. [3] to show how to practically extend OTs in the random oracle model assuming passive adversary. Since then, there have been many other purposed OT-extension construction based on Ishai et al.'s protocol in various security models and under various assumptions.

- Bob does not know σ
- Alice does not know $x_{1-\sigma}$

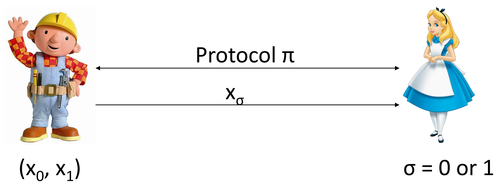


**Figure 1.** Oblivious Transfer

## Results

In this section, we describe our protocol for actively secure OT extension based on the passive KK13[4] protocol. In fact, the OT Extension protocol of KK13 [4] already provides security against a malicious Sender. Thus to deal with malicious Receiver, we need to ensure that Receiver inputs consistent values. In this protocol, we add a consistency check to the existing protocol of KK13 [4] to make it secure in the malicious setting. In most of the previous works, this check is added before the "extension" phase. In our construction, we check the correlation for consistency after the extension step, precisely after the execution of base OTs, actually checking the extended OTs.

| **Sender** | | **Receiver** |
|---|---|---|
| $x_{1,0}, \dots, x_{1,n}$ ..... $x_{m,0}, \dots, x_{m,n}$ | Inputs | $R = (r_1, \dots, r_m)$ $1 \le r_i \le n$ |
| $Q = \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_m \end{bmatrix}_{m \times k}$ $S \in \{0,1\}^k$ | Phase 1 – Base OTs: $s_i, q^i$ → Base OT → $t^i$, $t^i \oplus d^i$; $\mathbf{q}_i = \mathbf{t}_i \oplus (C_{r_i} \odot s)$ | $T = \begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_m \end{bmatrix}_{m \times k}$ $D = \begin{bmatrix} c_{r_1} \\ c_{r_2} \\ \dots \\ c_{r_m} \end{bmatrix}_{m \times k}$ |
| **Consistency Check**<br>• Sender and Receiver together generates randomness for the checks.<br>• Receiver performs a linear combination of the rows of T and D matrices using the randomness generated and sends the result to Sender.<br>• Sender performs the check using properties of Walsh Hadamard Code. | | |
| Phase 2 – Sending Masked Inputs<br>$y_{i,1} = x_{i,1} \oplus H(i, \mathbf{q}_i \oplus (C_1 \odot s))$ ..... $y_{i,r} = x_{i,r} \oplus H(i, \mathbf{q}_i \oplus (C_r \odot s))$ ..... $y_{i,m} = x_{i,m} \oplus H(i, \mathbf{q}_i \oplus (C_m \odot s))$ | $y_{i,1}, \dots, y_{i,m}$ → | $z_i = y_{i,r_i} \oplus H(i, \mathbf{t}_i)$ |
| **Notations :** H - Random Oracle $c_i$ - $i^{th}$ Walsh Hadamard Code Matrix A : $a_i$ - $i^{th}$ row $a^j$ - $j^{th}$ column | | |

**Algorithm 1.** Our Actively Secure Protocol

| | KK13 | | Our Protocol | |
|---|---|---|---|---|
| Run Time (In milliseconds) | LAN | WAN | LAN | WAN |
| | 3258.53 | 23218.17 | 3424.22 | 25742.42 |
| Communication (In Bytes) | 24005671 | | 24007082 | |

**Table 1.** Comparison of our protocol (PSS) with KK13.

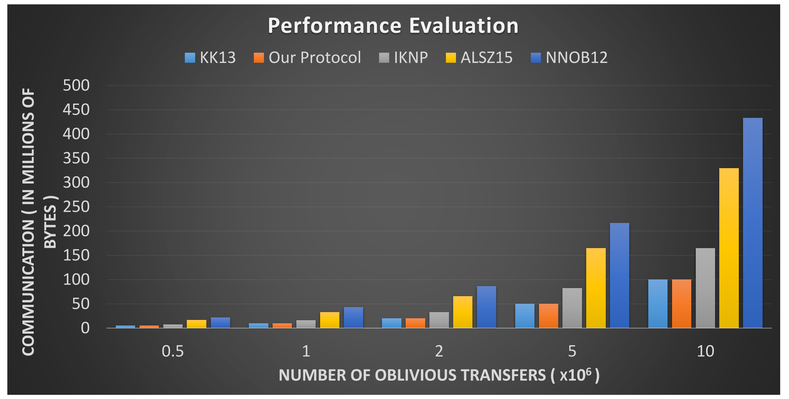


**Chart 1.** Performance Evaluation of OT Extension protocols.

## Discussion

The protocol of KK13[4] provides a O(log(k)) factor improvement over IKNP[3] in both communication and computation for bit inputs . As you can see from **Table 1**, our protocol adds only 5% computation overhead to KK13[4] in the LAN settings, and achieves active security.

In terms of communication overhead, our protocol is as efficient as the passive KK13 protocol, adding an overhead of 0.01%, which is negligible. For the results in the local setting we can observe that our active-secure OT extension protocol outperforms the ALSZ15[1] protocol for all OTs tested on and scales better with increasing number of OTs (See **Chart 1**). ALSZ15[1] has an overhead of around 220% in comparison with our protocol. In fact our active secure protocol outperforms the passive IKNP[3] protocol itself, reducing the overall communication by 62%.

## Conclusions

In this work, we present an actively secure OT extension protocol with efficiency very close to the passive OT extension protocol of KK13. Our protocol outperforms all existing actively secure OT extension protocols in terms of communication.

For communication and computation costs, the overhead on top of KK13 is negligible: our protocol requires 2 finite field operations per extended OT, plus a small communication overhead of O(k) bits in a constant number of rounds, independent of the number of OTs being performed, which amortizes away when creating many OTs.

We present implementation results that show our protocol takes no more than 5% more time than the passively secure KK13[4] extension and thus is essentially optimal with respect to the passive protocol.

## Contact

Ajith S
Indian Institute Of Science, Bangalore
Email : ajithskoppara@gmail.com
Phone : +918762049224

## References

1. [ALSZ15] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. *More efficient oblivious transfer extensions with security for malicious adversaries*. In Elisabeth Oswald and Marc Fischlin, editors, EUROCRYPT 2015, Part I, volume 9056 of LNCS, pages 673-701, Sofia, Bulgaria, April 26-30, 2015. Springer, Berlin, Germany.
2. [Bea96] Donald Beaver. *Correlated pseudo randomness and the complexity of private computations*. In STOC, pages 479-488, 1996.
3. [IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. *Extending oblivious transfers efficiently*. In Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, pages 145–161, 2003.
4. [KK13] Vladimir Kolesnikov and Ranjit Kumaresan. *Improved OT extension for transferring short secrets*. In Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II, pages 54–70, 2013.
5. [NNOB12] 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 2012, volume 7417 of LNCS, pages 681-700, Santa Barbara, CA, USA, August 19-23, 2012. Springer, Berlin, Germany