

1 Introduction

Oblivious Transfer (OT) is a fundamental cryptographic primitive that enables two parties to transfer information while preserving the privacy of their inputs. In a 1-out-of-2 Oblivious Transfer (1-out-of-2 OT) protocol, a sender holds two messages, and a receiver obtains exactly one message of its choice. The receiver's choice remains hidden from the sender, while the unselected message remains hidden from the receiver. OT is a fundamental building block in secure multiparty computation (MPC), particularly in protocols based on Yao's garbled circuits.

1.1 Scheme Definition

A 1-out-of-2 Oblivious Transfer scheme is an interactive protocol between a sender S and a receiver R , parameterized by a security parameter λ . It is defined by the tuple of probabilistic polynomial-time algorithms

$$\Pi_{\text{OT}} = (\text{Setup}, \text{Send}, \text{Receive}),$$

where the algorithms operate as follows:

- $\text{Setup}(1^\lambda)$: Generates the public parameters pp required by the protocol.
- $\text{Send}(\text{pp}, m_0, m_1)$: The sender takes two messages $m_0, m_1 \in \{0, 1\}^\ell$ as input and participates in the interactive protocol.
- $\text{Receive}(\text{pp}, b)$: The receiver takes a choice bit $b \in \{0, 1\}$ as input and participates in the interactive protocol to obtain the selected message m_b .

The scheme must satisfy the following properties.

Correctness. For every pair of messages $m_0, m_1 \in \{0, 1\}^\ell$ and every choice bit $b \in \{0, 1\}$, honestly executing parties satisfy

$$\Pr[\text{out}_R = m_b] \geq 1 - \text{negl}(\lambda),$$

where out_R denotes the receiver's output and the probability is taken over the randomness of the protocol.

Sender Privacy. A malicious receiver should not learn information about the unselected message m_{1-b} beyond what can be inferred from the selected message m_b . Formally, for any probabilistic polynomial-time receiver adversary $\mathcal{A}$, its views in executions with message pairs (m_0, m_1) and (m'_0, m'_1) must be computationally indistinguishable whenever the messages selected by its choice bit are identical.

Receiver Privacy. A malicious sender should not learn the receiver's choice bit. Formally, for any probabilistic polynomial-time sender adversary $\mathcal{A}$, its views in executions with receiver choices $b = 0$ and $b = 1$ must be computationally indistinguishable for the same sender inputs.

Here, $\text{negl}(\lambda)$ denotes a negligible function in the security parameter λ , and computational indistinguishability is denoted by $\approx_c$.

1.2 1-out-of- n OT

The definition can be generalized to 1-out-of- n Oblivious Transfer (1-out-of- n OT), where the sender holds n messages $(m_1, \dots, m_n)$ and the receiver holds a selection index $b \in \{1, \dots, n\}$. The receiver obtains only the selected message m_b , while the sender learns nothing about the selection index b and the receiver learns no information about the remaining messages $\{m_i\}_{i \neq b}$. The correctness, sender privacy, and receiver privacy requirements are defined analogously to those of 1-out-of-2 OT.

2 Example Constructions

2.1 Diffie–Hellman-Based Oblivious Transfer

Here, we present the 1-out-of-2 OT protocol proposed by Chou and Orlandi [1].

Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order p . Let H be a hash function used for key derivation, and let (Enc, Dec) be a symmetric encryption scheme. The sender S holds messages (M_0, M_1) , and the receiver R holds a choice bit $c \in \{0, 1\}$.

Protocol.

1. **Sender:** Choose $a \xleftarrow{\$} \mathbb{Z}_p$ and compute $A = g^a$. Send A to R .
2. **Receiver:** Choose $b \xleftarrow{\$} \mathbb{Z}_p$ and compute

$$B = \begin{cases} g^b, & c = 0, \\ Ag^b, & c = 1. \end{cases}$$

Send B to S .

3. **Sender:** Compute

$$k_0 = H(B^a), \quad k_1 = H((B/A)^a),$$

and encrypt the messages:

$$e_0 = \text{Enc}_{k_0}(M_0), \quad e_1 = \text{Enc}_{k_1}(M_1).$$

Send (e_0, e_1) to R .

4. **Receiver:** Compute $k_R = H(A^b)$ and output

$$M_c = \text{Dec}_{k_R}(e_c).$$

Correctness. If $c = 0$, then $B = g^b$, and hence $B^a = g^{ab} = A^b$. Thus $k_0 = k_R$. If $c = 1$, then $B = Ag^b$, so $B/A = g^b$ and $(B/A)^a = g^{ab} = A^b$. Thus $k_1 = k_R$. In either case, the receiver derives the encryption key corresponding to its choice and correctly recovers M_c by decryption.

Sender Privacy (Intuition). The receiver knows $A = g^a$, B , and b , and can compute the selected key from $A^b = g^{ab}$. Recovering the unselected message requires the corresponding unselected key, which depends on the additional group element g^{a^2} . Computing this element from g and $A = g^a$ is an instance of the Computational Diffie–Hellman problem. Under the relevant computational hardness assumption and appropriate encryption and hash-function security, the unselected message remains hidden.

Receiver Privacy. For either choice bit, B is uniformly distributed in $\mathbb{G}$ because b is uniformly random. Therefore, the distribution of B does not reveal c to the sender.

2.2 Oblivious Transfer Protocol

We describe a 1-out-of-2 Oblivious Transfer (OT) protocol between Alice and Bob proposed by Barreto et al. [2]. Alice holds two messages (m_0, m_1) , and Bob holds a choice bit $c \in \{0, 1\}$. The goal is to allow Bob to obtain m_c without learning anything about m_{1-c} , while Alice learns nothing about Bob's choice bit c .

Let KG, Enc, and Dec denote the key generation, encryption, and decryption algorithms of a public-key encryption scheme. Let H_1 and H_2 be suitable hash functions.

The OT protocol proceeds as follows.

Step 1: Bob's key generation. Bob chooses a random bit $c \in \{0, 1\}$ and generates a key pair $(pk_c, sk_c) \leftarrow \text{KG}(1^\lambda)$. He samples $s \leftarrow \{0, 1\}^\kappa$ and computes $q = H_1(s)$.

Bob constructs the other public key as

$$pk_{1-c} = q \cdot pk_c^{-1}.$$

He sends (s, pk_0) to Alice. Bob retains sk_c .

Step 2: Alice's message encryption. Upon receiving (s, pk_0) , Alice computes

$$q = H_1(s), \quad pk_1 = q \cdot pk_0^{-1}.$$

Thus, Alice obtains both public keys pk_0 and pk_1 .

Alice independently samples $b_0, b_1 \leftarrow \{0, 1\}^\kappa$ and computes

$$\begin{aligned} p'_0 &= H_2(b_0), & p'_1 &= H_2(b_1), \\ m'_0 &= m_0 \oplus p'_0, & m'_1 &= m_1 \oplus p'_1, \\ ct_0 &= \text{Enc}(pk_0, b_0), & ct_1 &= \text{Enc}(pk_1, b_1). \end{aligned}$$

Alice sends the tuple

$$(ct_0, ct_1, m'_0, m'_1)$$

to Bob.

Step 3: Bob's message recovery. Using his secret key sk_c , Bob decrypts the corresponding ciphertext:

$$b_c = \text{Dec}(sk_c, ct_c).$$

He then computes

$$m_c = H_2(b_c) \oplus m'_c.$$

Bob obtains his chosen message m_c .

The message flow between the parties in protocol is shown in the Figure 1.

Correctness Assuming correct public-key encryption and successful decryption, Bob recovers b_c from ct_c . Therefore,

$$\begin{aligned} m_c &= H_2(b_c) \oplus m'_c \\ &= H_2(b_c) \oplus m_c \oplus H_2(b_c) \\ &= m_c. \end{aligned}$$

Hence, Bob correctly recovers the message corresponding to his choice bit c . A in

Security Intuition

- **Receiver privacy:** Alice receives only (s, pk_0) . Under the appropriate assumptions on the public-key construction, the public keys do not reveal which one Bob generated together with the corresponding secret key. Thus, Alice cannot determine Bob's choice bit.
- **Sender privacy:** Bob possesses only sk_c , so he can decrypt ct_c and recover m_c . Recovering the other message requires obtaining the corresponding encryption key or breaking the underlying encryption scheme.

The security of the protocol depends on the precise algebraic structure of the public-key space, the key-generation algorithm, and the security assumptions of the underlying encryption scheme.

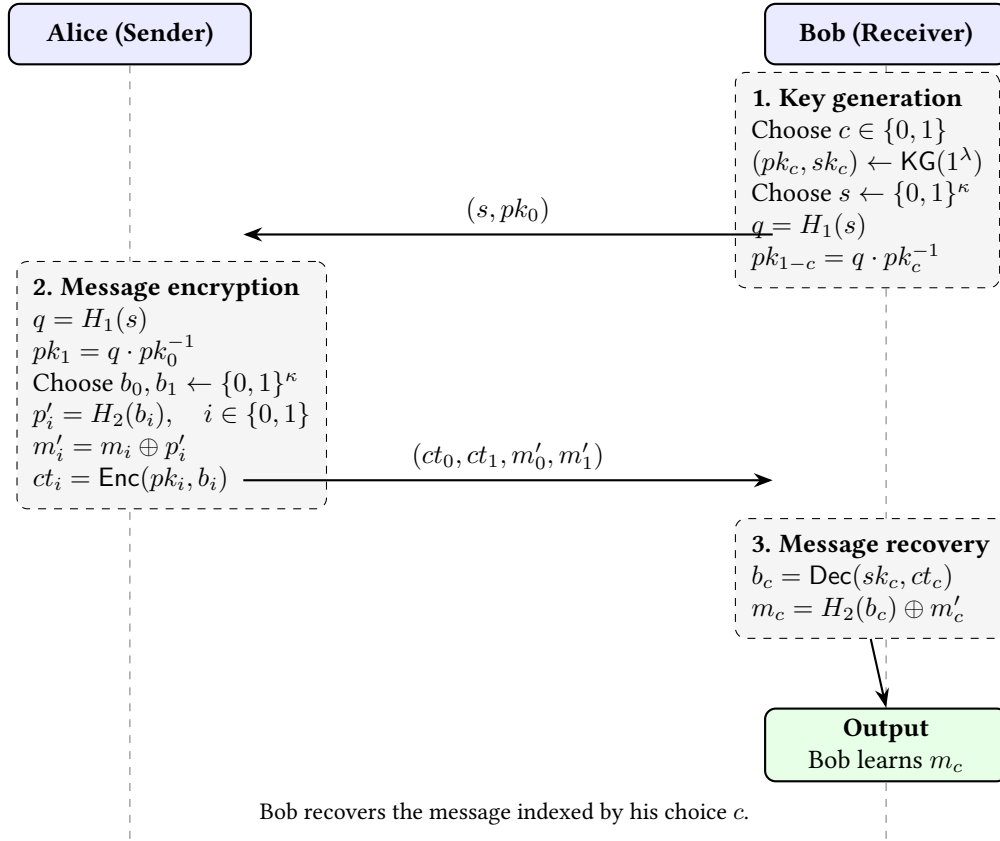


Figure 1: Message flow of the post-quantum oblivious transfer protocol.

3 Fundamental Idea of PQ-Safe OTs

Post-quantum oblivious transfer (PQ-OT) protocols follow two main approaches. The first uses quantum communication, such as quantum oblivious key (QOK) distribution, to achieve security based on quantum mechanics, but requires specialized hardware and infrastructure [5]. The second, more practical approach constructs OT from post-quantum cryptographic primitives based on computational problems believed to resist both classical and quantum attacks. These include lattice-based assumptions, such as Learning With Errors (LWE) and Learning With Rounding (LWR), and code-based assumptions relying on the hardness of decoding random linear codes. By avoiding classical assumptions based on integer factorization and discrete logarithms, which are vulnerable to Shor's algorithm, these constructions aim to provide quantum-resistant security with practical computational and communication costs.

References

- [1] T. Chou and C. Orlandi, "The Simplest Protocol for Oblivious Transfer," in *Progress in Cryptology – LATINCRYPT 2015*, Lecture Notes in Computer Science, vol. 9230, pp. 40–58, Springer, 2015.
- [2] P. S. L. M. Barreto, B. David, R. Dowsley, K. Morozov, and A. C. A. Nascimento, "A Framework for Efficient Adaptively Secure Composable Oblivious Transfer in the ROM," Cryptology ePrint Archive, Report 2017/993, 2017. <https://eprint.iacr.org/2017/993>
- [3] D. Masny and P. Rindal, "Endemic Oblivious Transfer," Cryptology ePrint Archive, Report 2019/706, 2019.
- [4] S. Dong, H. Cui, K. Zhang, K. Yang, and Y. Yu, "A Simple Post-Quantum Oblivious Transfer Protocol from Mod-LWR," Cryptology ePrint Archive, Paper 2024/1116, 2024. <https://eprint.iacr.org/2024/1116>
- [5] A. Khutsaeva, A. Leevik, and S. Bezzateev, "A Survey of Post-Quantum Oblivious Protocols," *Cryptography*, vol. 9, no. 4, p. 62, 2025. <https://doi.org/10.3390/cryptography9040062>