A new Protocol for 1-2 Oblivious Transfer

Björn Grohmann

Universität Karlsruhe, Fakultät für Informatik
76128 Karlsruhe, Germany
nn@mhorg.de

Abstract

A new protocol for 1-2 (String) Oblivious Transfer is proposed. The protocol uses 5 rounds of message exchange.

Keywords: Oblivious Transfer, cryptographic Hash-Function, One-Way-Function.

1 Introduction

During a 1-2 (String) Oblivious Transfer protocol, Bob should learn one of two bit strings provided by Alice, but not both, while Alice should not learn anything about Bob's choice.

A protocol fulfilling these constraints would be a powerful cryptographic primitive (cf. [3] for an introduction to the subject).

In this article, we propose a protocol that uses 5 rounds of message exchange. Since most of the computational part of the protocol takes place in the unit group of a finite field, we further investigate the question whether Alice or Bob can gain more information, if it turns out that the computation of discrete logarithms in this group is easy.

2 The Protocol

Initialisation: Before the actual start of the protocol Alice and Bob agree on a positive integer $n \in \mathbb{N}$, a prime p of size $\sim 2^{\sqrt{n \log n}}$, a random matrix $C = (c_{i,j})_{i,j} \in \mathbb{F}_p^{n \times n}$, $i, j = 1, \ldots n$, a cryptographic Hash-Function $h_1 : \mathbb{F}_p \longrightarrow \{0,1\}^q$ and an injective (polynomial-time computable) One-Way-Function $h_2 : \{0,1\}^q \longrightarrow \{0,1\}^{q'}$, for integers q and q'. Here, $\mathbb{F}_p$ denotes the finite field with p elements.

Round 1: Alice starts by choosing n random bits $t_1, \ldots, t_n$, two distinct random elements $a, b \in \mathbb{F}_p$, with $a \neq -c_{i,j} \neq b$ for $i, j = 1, \dots, n$, two distinct random elements $\alpha_a, \alpha_b \in \mathbb{F}_p^\times$ of order $p-1$ (i.e. each of these elements is a generator of the unit group $\mathbb{F}_p^\times := \mathbb{F}_p - \{0\}$) and two random permutations σ_a, σ_b on the set $\{1, \dots, n\}$. She then computes, for $j = 1, \dots, n$,

$$\mu_{j,a} := \alpha_a^{\sigma_a(j)} \prod_{i=1}^{n} (a + c_{i,j})^{t_i} \quad \text{and} \quad \mu_{j,b} := \alpha_b^{\sigma_b(j)} \prod_{i=1}^{n} (b + c_{i,j})^{t_i} \tag{1}$$

and sends $((\mu_{j,a})_j, (\mu_{j,b})_j)$ to Bob.

Round 2: Bob chooses n random bits $s_1, \dots, s_n$. He computes

$$\tau_{A,a} := \prod_{j=1}^{n} \mu_{j,a}^{s_j} \quad \text{and} \quad \tau_{A,b} := \prod_{j=1}^{n} \mu_{j,b}^{s_j} \tag{2}$$

and sends $(\tau_{A,a}, \tau_{A,b})$ to Alice.

Round 3: Alice chooses two (random) bit strings m_a, m_b of size q (the messages) and computes $z_a := h_2(m_a)$ and $z_b := h_2(m_b)$. She then computes, for $k = 1, \dots, \frac{n(n-1)}{2}$,

$$s_{k,a} := h_1\left(\alpha_a^{-k} \tau_{A,a}\right) \oplus m_a \quad \text{and} \quad s_{k,b} := h_1\left(\alpha_b^{-k} \tau_{A,b}\right) \oplus m_b, \tag{3}$$

where $\oplus$ denotes the `XOR`-function, and sends $((s_{k,a})_k, (s_{k,b})_k, a, b, z_a, z_b)$ to Bob.

Round 4: Bob chooses a random element $\beta \in \mathbb{F}_p^\times$ of order $p-1$, a random permutaion ρ on the set $\{1, \dots, n\}$ and an element $d \in \{a, b\}$. He then computes, for $i = 1, \dots, n$,

$$\nu_i := \beta^{\rho(i)} \prod_{j=1}^{n} (d + c_{i,j})^{s_j} \tag{4}$$

and sends $(\nu_i)_i$ to Alice.

Round 5: Alice computes

$$\tau_B := \prod_{i=1}^{n} \nu_i^{t_i} \tag{5}$$

and sends τ_B to Bob.

Finally, Bob computes for $r = 1, \dots, \frac{n(n-1)}{s}$ the list $(\beta^{-r}\tau_B)_r$ until he finds r_0 and k_0 such that $h_2(h_1(\beta^{-r_0}\tau_B) \oplus s_{k_0,d}) = z_d$, which gives him the message $m_d = h_1(\beta^{-r_0}\tau_B) \oplus s_{k_0,d}$.

3 Analysis

The following theorem states the correctness of the protocol and (roughly) counts the computational cost for both sides (for simplicity, we count addition and multiplication in $\mathbb{F}_p$ as one elementary operation and leave aside the randomized selection process).

Theorem 1 *At the end of the protocol, Bob is in possession of the message he asked for. The computational cost for Alice equals* $\mathbf{O}(n^2 \cdot (\text{cost of } h_1))$ *elementary operations, while on Bob's side it sums up to* $\mathbf{O}(n^2 \cdot (\text{cost of } h_1) + n^4 \cdot (\text{cost of } h_2))$.

Proof. The first statement of the theorem is easily seen to be true, since

$$\tau_{A,d} = \alpha_d^{k'} \prod_{i,j=1}^{n} (d + c_{i,j})^{t_i s_j} \tag{6}$$

and respectively

$$\tau_B = \beta^{r'} \prod_{i,j=1}^{n} (d + c_{i,j})^{t_i s_j}, \tag{7}$$

with $d \in \{a, b\}$ and $1 \leqslant k', r' \leqslant n(n-1)/2$. The calculation of the computational cost is straightforward. $\square$

We now turn to the two fundamental questions for this protocol. For this, we define the function $f(y) := \prod_{i,j}(y + c_{i,j})^{t_i s_j}$. It is clear that, for $d \in \{a, b\}$, the knowledge of $f(d)$ leads to the knowledge of the message m_d.

Q1: Can Alice efficiently decide whether Bob chose $d = a$?

Q2: Can Bob, who knows $f(d)$, efficiently compute $f(a + b - d)$?

So far, the author of this article is not aware of any polynomial time algorithm that would answer one of these questions with "yes".

In the following we shall see that even the ability to efficiently compute discrete logarithms in $\mathbb{F}_p^\times$ does not seem to help much.

So, from now on we will assume that Alice and Bob can compute discrete logarithms in $\mathbb{F}_p^\times$ efficiently. To start with Bob (i.e. **Q2**) it is easily seen that the knowledge of Alice's secret bits $t_1, \ldots, t_n$ immediately gives him both messages m_a and m_b (he can compute $f(a)$ and $f(b)$). To get these bits, Bob can choose a generator g of the group $\mathbb{F}_p^\times$ and try to solve the equation (cf. (5))

$$x_1 \delta_g(\nu_1) + \cdots + x_n \delta_g(\nu_n) \equiv \delta_g(\tau_B) \bmod p - 1, \tag{8}$$

where $\delta_g(\cdot)$ denotes the discrete logarithm function with respect to g. Since there are 2^n ways to select the values of the x_i's, there are, heuristcally speaking, approximately $2^{n-\log p} \sim 2^{n(1-\sqrt{\log n/n})}$ solutions to equation (8). Now suppose that Bob knows $f(a)$. He then can compute $\alpha_a^{k'}$, with an unknown positive integer $k' \leqslant n(n-1)/2$. Suppose further that he somehow manages to determine α_a (or at least a list of possible candidates for α_a). Since $\gcd(\delta_q(\alpha_a), p-1) = 1$ this leads (cf. (1)) in general to the following

Challenge 1 *Given* $n \in \mathbb{N}$, *a prime* p *of size* $\sim 2^{\sqrt{n \log n}}$, *a matrix* $(e_{i,j})_{i,j=1,\ldots,n}$ *with integer coefficients and a list of integers* $(f_j)_{j=1,\ldots,n}$, *compute* $x_1, \ldots, x_n$, *with* $x_i \in \{0,1\}$, *and a permutation* π *on the set* $\{1, \ldots, n\}$ *such that*

$$\begin{aligned}
x_1 e_{1,1} + \ldots + x_n e_{1,n} + \pi(1) &\equiv f_1 \bmod p-1 \\
x_1 e_{2,1} + \ldots + x_n e_{2,n} + \pi(2) &\equiv f_2 \bmod p-1 \\
&\vdots \\
x_1 e_{n,1} + \ldots + x_n e_{n,n} + \pi(n) &\equiv f_n \bmod p-1.
\end{aligned}$$

Again, the author of these lines is not aware of any efficient method that solves this challenge.

Now, Alice's story (**Q1**) is pretty much the same. In the end, Alice finds herself confronted with a decision version of Challenge 1, but as is easily seen, an algorithm that can decide in polynomial time whether a solution exists can also be used to efficiently compute a solution.

References

[1] Goldreich, O., Micali, S., Wigderson, A.: How to Play Any Mental Game, or: A completness theorem for protocols with honest majority. In: Proc., 19th Annual ACM Symposium on the Theory of Computation (STOC), pp. 218-229, 1987

[2] Grohmann, B.: A New Key-Agreement-Protocol. arXiv:0904.1186 [cs.CR], 2009

[3] Killian, J.: Founding Cryptography on Oblivious Transfer. In: Proc., 20th Annual ACM Symposium on the Theory of Computation (STOC), pp. 20-31, 1988

[4] Rabin, M.O.: How to exchange secrets by oblivious transfer. In: Technical Report TR-81, Akien Computation Laboratory, Havard University, 1981

[5] Shamir, A.: An efficient identification scheme based on permutation kernels. In: Proc of Crypto 89, Vol. 435 LNCS, pp. 606-609, Springer, 1990