Communication Efficient Oblivious Transfer Using Elliptic Curves

Abhishek Parakh

***Abstract*—This paper presents communication efficient protocols for oblivious transfer (OT) using elliptic curve cryptography (ECC). ECC provides better security than RSA for the same key length and is known to have faster implementations. We provide protocols for 1-out-of-2 and 1-out-of-n oblivious transfer in which the latter requires the sender to send only $n+2$ elements over the network and the receiver sends only 1 messages over the network.**

I. Introduction

Oblivious transfer (OT) [1], [2], [3], [4], [5], [6] is an important primitive in cryptography and has numerous applications such as multiparty computations and private information retrieval [7], [8]. Further, elliptic curve cryptography is known to provide same level of security with a 256 bit key as compared to 3072 bit key required by RSA and DSA [9], [10] and is well suited for embedded systems and low power implementations. However, not much attention has been focused on developing oblivious transfer protocols using ECC. To the author's knowledge only two protocols currently exist for OT over ECC [11], [12]. On the other hand oblivious transfer requires numerous rounds of communication between the parties and research has been done on reducing the communication burden on the network [13], [14].

In this paper, we provide implementations for 1-out-of-2 and 1-out-of-n oblivious transfer that requires the sender to send only $n+2$ messages over the network while the receiver sends only 1 messages over the network. In a 1-out-of-2 oblivious transfer, Alice possesses two secrets but is willing to disclose only one of them, whereas Bob does not want Alice to know which of the two secrets he wishes to know. Such a situation may arise, when say Alice possesses answers to two questions of say national security and is willing to disclose it to any one who pays for them. Bob on the other hand is willing to pay for only 1 of the secrets or is interested in only one of them but does not want Alice to know which one. Bob will get only one of the questions answered while not receiving any information about the other, Alice will not know what information Bob received. A 1-out-of-n oblivious transfer is an extension of this idea to n secrets. The starting point of our protocol is a non-interactive oblivious transfer protocol in which Alice sends to Bob two strings s_0 and s_1 as functions of Bobs public keys and her private keys such that Bob is able to extract exactly one of the two strings using only his private key. An early non-interactive oblivious transfer scheme due to Bellare et al [15], [13] used the Diffie-Hellman (DH) protocol that is based on the discrete log problem (DLP) over finite fields.

Further, we use an elliptic curve key exchange protocol [11], [16] to propose a hybrid oblivious transfer (HOT) scheme in which ECC is used to establish a (oblivious) shared secret key between Alice and Bob in an oblivious manner and then the "actual" transfer of messages takes place using conventional (symmetric key) encryption methods that are much faster than public key encryption methods. In the method of hybrid oblivious transfer, Alice will generate two symmetric keys in coalition with Bob; however Bob will only be able to generate one of these symmetric keys. Alice will not know which of the two keys Bob has generated. Alice can then send any number of bits encrypting them using these two symmetric keys, while Bob will only be able to decrypt those bits for which he has the same key as Alice does.

A. Parakh is with the Computer Science Department, Oklahoma State University, Stillwater, OK 74078 USA. Email: {parakh}@cs.okstate.edu

II. Preliminaries

For cryptographic purposes an elliptic curve over finite field $\mathbb{Z}_p$, p a prime, is used [10]. The variables and coefficients all take values between 0 and $p-1$ and all the calculations are performed modulo p. The general form of the equation used is given by $y^2 \bmod p = (x^3 + ax + b) \bmod p$, where a and b are suitably chosen [10] integer constants. The set of (x, y) points that satisfy such an elliptic curve equation is denoted by $E_p(a, b)$. The order r or a point $T = (x_1, y_1)$ on an elliptic curve is defined as the smallest positive integer r such that $rT = 0$. A point G is called a base point in $E_p(a, b)$ and is picked such that its order r is a very large value. The security of an elliptic curve cryptosystem depends on the elliptic curve discrete log problem (ECDLP).

Solving the elliptic curve discrete log of a point Q is equivalent of solving for an integer k such that $Q = kP$, where only Q and P are known. It is assumed that there does not exist any polynomial time solution to such an equation. In other words, given P and k, it is easy to compute $Q = kP$; however, given Q and P, such that $Q = kP$, it is relatively hard to determine the value of k. It is taken that all points in a given computation lie on the same elliptic curve.

III. 1-out-of-2 Oblivious Transfer

Alice and Bob agree upon an elliptic curve to use and a base point G. Alice randomly and uniformly chooses from the field k_0 and k_1 and generates two points P_0 and P_1 such that $P_0 = k_0G$ and $P_1 = k_1G$. The values k_0 and k_1 must not be known to Bob. In other words, computing values of k_0 and k_1, given P_0 and P_1, is equivalent finding elliptic curve discrete log (ECDL) of points P_0 and P_1, respectively. Such

agreements can be made long before the actual transfer of messages.

A. Proposed Protocol

Given: Base point G and points P_0 and P_1. Let $P = P_0 + P_1$ and $i \in 0, 1$.

1) Bob creates public keys (U_0, U_1) by choosing a random integer m and setting $U_i = mG$. Let $U_b = mG - P_i$ and $U_{1-i} = P_{1-i} - U_b$. Bobs private key is m.
2) Alice chooses two random integers a_0 and a_1 and computes $V_0 = a_0 G$ and $V_1 = a_1 G$. She then computes $W_0 = a_0 U_0$ and $W_1 = a_1 U_1$ and sends to Bob V_0, V_1, $X_0 = s_0 \oplus W_0$ and $X_1 = s_1 \oplus W_1$.
3) Bob first computes $mV_i = ma_i G = W_i$ and then extracts s_i using $s_i = W_i \oplus X_i$.

Bob cannot extract s_{1-i}. This is because he cannot compute n, where $U_{1-i} = nG$.

Alice does not know which of the two U_is is equal to mG such that Bob knows the value of m; hence she does not know which of the two secrets Bob receives. Alice can check the proper formation of Bobs public keys using the following steps,

1) Check if $U_i + U_{1-i} = P$.
2) Compute $U_b = U_i - P_i$ and check if it is negative of $U_{1-i} - P_{1-i}$.

B. Security of the Protocol

Claim 1: Alice cannot deduce which secret Bob has retrieved.

This is because she does not know which of the two U_is is equal to mG such that Bob knows the value of m.

She can further check the proper formation of Bobs public keys using the steps described above.

Claim 2: Security of our protocol, against Bob, is based on the assertion that he only knows the elliptic curve discrete log of one of the U_is.

We know that for any randomly chosen integer m and point $P_b = mG - U$, where point $P = kG$, and point $U_b = lG$ then computing l is equivalent to computing the elliptic curve discrete log of point P and vice versa.

This is because $P_b = mG - U$, $U_b = mG - P$, i.e. $U_b = mG - kG = (m-k)G$. Hence, $lG = (m-k)G$ and $l = m-k$. Since m is known, computing k will reveal l, which is the ECDL of point U_b, and vice versa.

For any randomly chosen integer m, $i \in \{0, 1\}$, point $U_i = mG = P_i + U_b$, where point $P_i = k_i G$, and point $U_b = lG$ and the point $U_{1-i} = P_{1-i} - U_b$, where $P_{1-i} = k_{1-i} G$, then computing ECDL of point U_{1-i} is as hard as computing the ECDL of both P_i and P_{1-i}.

Above we have estabilised that determining ECDL of U_b is at least as hard as determining the ECDL of point P_i and it follows that in order to determine the ECDL of point U_{1-i}, we need to calculate the ECDL of U_b and P_{1-i}.

Hence, determining the ECDL of U_{1-i} is equivalent to computing the ECDL of both P_i and P_{1-i}.

Consequently, the security of our protocol is equivalent to the problem of computing the elliptic curve discrete log of both the points P_0 and P_1. However, it is assumed that this problem does not have a polynomial time solution for "properly" chosen points P_0 and P_1. Therefore, Bob only knows the discrete log of one of the U_is in all practical cases.

Claim 3: Assuming Alice follows the protocol correctly, Bob can deduce only one of the secrets.

This result is obvious from the fact that Bob knows the ECDL of only one of his public keys.

IV. Communication efficient 1-out-of-n oblivious transfer

Assume an agreement on the elliptic curve to be used (as discussed above). Then the protocol for 1-out-of-n oblivious transfer proceeds as follows,

Inputs: Alice's input are secrets s_0, s_1, ..., s_{n-1}. Bob's input is his choice $\sigma \in \{0, 1, \ldots, n-1\}$.

1) Alice selects a point $C = P$ on the elliptic curve. (It is important that Bob does not know the ECDL of P. It does not matter, if Alice knows it or not.)
2) Alice selects randomly and uniformly a number r from the field and generates rG.
3) Alice sends C and rG to Bob.
4) Bob selects a random number k randomly and uniformly from the field and sets $PK_\sigma = kG$.
5) Bob computes the decryption key $krG = rPK_\sigma$.
6) If $\sigma \neq 0$, Bob computes $PK_0 = \sigma C - PK_\sigma$.
7) Bob sends PK_0 to Alice.
8) Alice computes rPK_0 and for all $1 \leq i \leq n-1$, she computes $rPK_i = riC - rPK_0$.
9) Alice uses rPK_i as encryption key to encrypt s_i: $E(s_i)$ and sends them to Bob.
10) Bob chooses $E(s_\sigma)$ and decrypts it using rPK_σ to extract s_σ.

Note that when $\sigma = i$ in the above protocol, $rPK_i = riC - rPK_0 = r\sigma C - r\sigma C - rPK_\sigma = rPK_\sigma$. When $\sigma \neq i$, Bob does not get any information.

The protocol has a communication overhead of $n + 2$ messages for the sender and just 1 message for the receiver.

V. Oblivious Transfer Channel Based on Elliptic Curve Key Exchange

An oblivious transfer channel [7], [1] is a pair $C = (C_0, C_1)$ of channels such that Alice can send any number of bits, to Bob, on either C_0 or C_1. However, only one of the channels is clear to Bob while the other is opaque, and Alice does not know which of the two channels is clear. Alice wishes to send encrypted strings, s_0 and s_1, to Bob such that Bob is able to decrypt only one of them and she does not know which one of the two strings did Bob decrypt, however, the choice lies with Bob.

In this section we modify the elliptic curve key exchange protocol to implement oblivious transfer channel. Besides having the advantage of being adaptable to any other public key cryptosystem, such a scheme represents the first implementation of a hybrid scheme for oblivious transfer. In a hybrid scheme, the initial oblivious key exchange takes place using a public key cryptosystem however subsequent transfer

of information occurs using a symmetric key cryptosystem. Hybrid oblivious transfer protocol is very useful in setting up an oblivious transfer channel as described below.

VI. The Proposed Protocol

Suppose Alice is the sender and Bob is the recipient. Alice will send many strings of data on two channels, however, only one of the channels will be clear to Bob and he will receive only one of the two streams of data. The data will be encrypted using a symmetric key; hence encryption is fast and efficient.

Alice and Bob agree upon an elliptic curve to be used. Alice secretly chooses a base point G and chooses two random numbers a_0 and a_1. She then discloses $U_0 = a_0G$ and $U_1 = a_1G$ as her public keys, where a_0 and a_1 are her secret keys. She also declares $U_A = a_0a_1G$. The protocol proceeds as follows:

1) Bob chooses a secret random number b and sends to Alice either $U = bU_i$ or $U = bU_{1-i}$.
2) Alice generates two secret key $K_0 = a_0U$ and $K_1 = a_1U$. Key K_i will be used to encrypt messages sent on channel C_i, where $i \in 0, 1$.
3) Bob generates a secret key $K = bU_A$.

From the above series of steps either $K_0 = K$ or $K_1 = K$. This happens with a probability of one-half. Alice does not know which of the two possibilities has taken place. Bob always generates $K = bU_A = ba_0a_1G$. However, Alice generates key pairs $K_0 = a_0U = a_0bU_0 = a_0^2bG$ and $K_1 = a_1U = a_1bU_0 = a_1a_0bG$, in which case channel C_1 is clear to Bob. Else she generates key pairs $K_0 = a_0U = a_0bU_1 = a_0a_1bG$ and $K_1 = a_1U = a_1bU_1 = a_1^2bG$ in which case channel C_0 is clear to Bob. The choice of which channel Bob wishes to read lies with Bob. Note that keys K_0 and K_1 are used as inputs to a symmetric key encryption algorithm.

Security. The security of our protocol, against Bob, is based on the elliptic curve assumption which may be defined as given a point $U = aG$, it is not possible to compute $V = a^2G$, without knowing the value of a. Further, computing the value of a is equivalent to solving the ECDLP for U, which is assumed to be "hard".

The security, against Alice, is based on the assumption that given two points $V_0 = ba_0G$ and $V_1 = ba_1G$, where a_0 and a_1 are known but b is not known, then it is not possible to distinguish V_0 and V_1. This is an extension of the ECDLP problem.

VII. Conclusions

We have presented protocols for 1-out-of-2 oblivious transfer and a communication efficient 1-out-of-n oblivious transfer protocol using elliptic curve cryptography. Our protocol, not only reduces burden on communication over the network, but due to properties ECC, reduces computational burdens as well, for both sender and receiver. We have also introduced a hybrid oblivious transfer scheme in which initial oblivious key exchange takes place using a public key cryptosystem (here ECC) and uses symmetric key encryption for subsequent (possibly larger) message transfers. The security of these schemes is predicated on the hardness of elliptic curve discrete log problem.

The idea of hybrid oblivious transfer may further be used to establish oblivious transfer sessions.

References

[1] M. O. Rabin, "How to exchange secrets with oblivious transfer," Cryptology ePrint Archive, Report 2005/187, 2005, http://eprint.iacr.org/.

[2] C. Crpeau, "Equivalence between two flavours of oblivious transfers," pp. 350– 354, 1988.

[3] G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Proceedings on Advances in cryptology—CRYPTO '86*. London, UK: Springer-Verlag, 1987, pp. 234–238.

[4] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proc. CRYPTO, Springer LNCS*. Springer-Verlag, 1999, pp. 573–590.

[5] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," pp. 554– 571, 2008.

[6] Y. Ishai, M. Prabhakaran, and A. Sahai, "Founding cryptography on oblivious transfer — efficiently," in *CRYPTO 2008: Proceedings of the 28th Annual conference on Cryptology*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 572–591.

[7] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 13, no. 1, pp. 73–78, January 1989.

[8] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," in *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1999, pp. 245–254.

[9] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Nist sp800-57: Recommendation for key management part 1: General(revised)," Tech. Rep., March 2007.

[10] A. Enge, *Elliptic curves and their applications to cryptography: an introduction*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[11] A. Parakh, "Oblivious transfer using elliptic curves," *Cryptologia*, vol. 31, no. 2, pp. 125–132, 2007.

[12] H. Huang and C. Chang, "A new t-out-n oblivious transfer with low bandwidth," *Applied Mathematical Sciences*, vol. 1, no. 7, pp. 311–320, 2007.

[13] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, 2001, pp. 448 – 457.

[14] H. Lipmaa, "New communication-efficient oblivious transfer protocols based on pairings," pp. 441–454, 2008.

[15] M. Bellare and S. Micali, "Non-interactive oblivious transfer and applications," in *CRYPTO '89: Proceedings on Advances in cryptology*. New York, NY, USA: Springer-Verlag New York, Inc., 1989, pp. 547–557.

[16] A. Parakh, "Oblivious transfer based on key exchange," *Cryptologia*, vol. 32, no. 1, pp. 37–44, 2008.