Available online at www.sciencedirect.com

ScienceDirect

Mathematical and Computer Modelling 46 (2007) 174–179

MATHEMATICAL AND COMPUTER MODELLING

www.elsevier.com/locate/mcm

# Assessing the effectiveness of artificial neural networks on problems related to elliptic curve cryptography

E.C. Laskari[a,d], G.C. Meletiou[b,d], Y.C. Stamatiou[c,d], D.K. Tasoulis[a,d], M.N. Vrahatis[a,d,*]

[a] *Computational Intelligence Laboratory, Department of Mathematics, University of Patras, GR–26110 Patras, Greece*
[b] *A.T.E.I. of Epirus, P.O. Box 110, GR–47100 Arta, Greece*
[c] *Department of Mathematics, University of Ioannina, GR–45110 Ioannina, Greece*
[d] *University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR–26110 Patras, Greece*

Received 18 May 2006; accepted 15 December 2006

**Abstract**

Cryptographic systems based on elliptic curves have been introduced as an alternative to conventional public key cryptosystems. The security of both kinds of cryptosystems relies on the hypothesis that the underlying mathematical problems are computationally intractable, in the sense that they cannot be solved in polynomial time. In this paper, we study the performance of artificial neural networks on the computation of a Boolean function derived from the use of elliptic curves in cryptographic applications.

*Keywords:* Artificial neural networks; Boolean functions; Elliptic curves; Discrete logarithm

## 1. Introduction

Cryptographic systems based on elliptic curves (ECG) have been proposed in [1,2] as an alternative to conventional public key cryptosystems. Their main advantage is that they use smaller parameters compared to the conventional cryptosystems (e.g. RSA). This is due to the apparently increased difficulty of the underlying mathematical problem, the *Elliptic Curve Discrete Logarithm Problem* (ECDLP). This problem is believed to require more time for its solution than the time required for the solution of its finite field analogue, the *Discrete Logarithm Problem* (DLP), that ensures the security of a number of cryptosystems (e.g. El Gamal). The security of cryptosystems that rely on discrete logarithms is based on the hypothesis that the underlying mathematical problems are computationally intractable, in the sense that they cannot be solved in polynomial time. Numerous techniques have been proposed to speed up the solution of these two types of the discrete logarithm problem, relying on both algebraic and number theoretic methods, software oriented methods, as well as, approximation and interpolation techniques [3–7].

Artificial neural networks (ANNs) have the inherent capability of storing experiential knowledge and rendering it available for use. These characteristics give ANNs the ability of solving complex real world problems. In this paper,

* Corresponding author at: Computational Intelligence Laboratory, Department of Mathematics, University of Patras, GR–26110 Patras, Greece.
*E-mail addresses:* elena@math.upatras.gr (E.C. Laskari), gmelet@teiep.gr (G.C. Meletiou), istamat@cc.uoi.gr (Y.C. Stamatiou), dtas@math.upatras.gr (D.K. Tasoulis), vrahatis@math.upatras.gr (M.N. Vrahatis).

0895-7177/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2006.12.017

we study the performance of ANNs on the computation of bits of discrete logarithms over elliptic curves. In particular, we consider a Boolean function that represents the problem at hand and study the ability of ANNs on the computation of the considered Boolean function.

The rest of the paper is organized as follows: In Section 2 an introduction to elliptic curve basics is given and the problem is formulated. In Section 3 basic notions relating to artificial neural networks are described. The experimental setup and the obtained results are reported in Section 4. The paper concludes in Section 5.

## 2. Introduction to elliptic curves and problem formulation

An *elliptic curve* over a prime finite field $\mathbb{F}_p$, where $p > 3$ and prime, is denoted by $E(\mathbb{F}_p)$ and is defined as the set of all pairs $(x, y) \in \mathbb{F}_p$ (points in affine coordinates) that satisfy the equation $y^2 = x^3 + ax + b$ where $a, b \in \mathbb{F}_p$, with the restriction $4a^3 + 27b^2 \neq 0$. These points, together with a special point denoted by $\mathcal{O}$, called the *point at infinity*, and an appropriately defined point addition operation form an Abelian group. This is the *elliptic curve group* and the point $\mathcal{O}$ is its identity element (see [8,9] for more details on this group).

The *order m of an elliptic curve* is defined as the number of points in $E(\mathbb{F}_p)$. According to Hasse's theorem (see e.g., [8,9]) it holds that:

$$p + 1 - 2\sqrt{p} \leqslant m \leqslant p + 1 + 2\sqrt{p}.$$

The *order of a point* $P \in E(\mathbb{F}_p)$ is the smallest positive integer, $n$, for which $nP = \mathcal{O}$. From Lagrange's theorem, it holds that the order of a point is a divisor of the order of the elliptic curve.

The discrete logarithm problem can be described as follows. Let $G$ be any cyclic group and $y$ one of its elements. The discrete logarithm problem for $G$ to the base $g \in G$ consists of determining an integer, $x$, such that $g^x = y$, when the group operation is written as multiplication, or $xg = y$ when the group operation is written as addition. In groups formed by elliptic curve points the group operation is addition. Therefore, the definition of the discrete logarithm problem over elliptic curves is as follows. Let $E$ be an elliptic curve over a finite field $\mathbb{F}_q$, $P$ a point on $E(\mathbb{F}_q)$ of order $n$ and $Q$ a point on a $E(\mathbb{F}_q)$, such that $Q = tP$, with $0 \leqslant t \leqslant (n - 1)$. The ECDLP consists of determining the value of $t$. Groups defined on elliptic curves are special since the best algorithms that solve the discrete logarithm problem over them require an exponential number of expected steps. In contrast, the best algorithms known today for the discrete logarithm problem defined over the multiplicative group of $\mathbb{F}_q$, are sub-exponential.

Regarding the discrete logarithm problem over elliptic curves, the following proposition is derived from the bit security of discrete logarithms over any cyclic group [10,11].

**Proposition 1.** *Given an elliptic curve, $E$, over a finite field, $\mathbb{F}_q$, with known order $n$, and an oracle for a bit of the discrete logarithm that does not correspond to any power of* 2 *that divides the order $n$, then all the bits of the discrete logarithm can be computed in polynomial time.*

**Remark 2.** Currently, there is no polynomial algorithm for finding the order of an elliptic curve. Furthermore, the complexity for the computation of the discrete logarithm problem over elliptic curves with no knowledge of its order is exponential, and hence it remains a difficult task.

From Proposition 1 it follows that in the case of an elliptic curve with odd order, $n$, an oracle that gives the least significant bit of the discrete logarithm of a point over the elliptic curve leads to the computation of all bits of the discrete logarithm in polynomial time. Moreover, prime order elliptic curves are considered more secure. Thus, the focus of this paper is on the computation of the least significant bit of the discrete logarithm of a point over elliptic curves of odd order. Complexity boundaries for the computation of bits of the discrete logarithm over different fields can be found in [3,12].

In relation to our problem, the considered Boolean function is defined as follows. Assume an elliptic curve $E(\mathbb{F}_p)$ and let $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ be two points of $E(\mathbb{F}_p)$, such that $Q = tP$, with $0 \leqslant t \leqslant (n - 1)$. Define the Boolean function $f : \{0, 1\}^{4\lceil \log p \rceil} \mapsto \{0, 1\}$, with

$$f(x_P, y_P, x_Q, y_Q) = \text{lsb}(t) \tag{1}$$

which has inputs the coordinates $x_P$, $y_P$, $x_Q$, $y_Q$, in binary representation, and outputs the least significant bit (lsb) of $t$, i.e., 1 if the least significant bit of $t$ is 1, and 0 otherwise.

In general, a Boolean circuit that computes this function can be exponentially large in $\lceil \log p \rceil$ [3]. For the computation of this Boolean function we employ Artificial Neural Networks.

## 3. Computation through artificial neural networks

The complex and parallel functionality of the human brain has motivated the design of Artificial Neural Networks (ANNs). An ANN can be considered as a massively parallel distributed processor, comprised of simple units, (*neurons*), and characterized by the inherent ability to acquire knowledge from data through a learning process. Knowledge is stored at the interneuron connection strengths, called *weights*, making it thus available for use [13].

Each artificial neuron implements a local computation. The output of this computation is determined by the neuron's input and its activation function. The overall functionality of a network is determined by its topology, i.e. the number of neurons and their interconnection pattern, the training algorithm that is applied to it, and its neuron characteristics [14,15].

A Feedforward Neural Network (FNN) is a network with acyclic and one-way directed interneuron connections, where all neurons can be grouped into layers. Thus, the network's topology can be described by a series of integers each representing the number of units that belong to the corresponding layer.

The goal of training an ANN is to assign to the weights (free parameters) of the network, $W$, values such that the difference between the desired output (target) and the actual output of the network is minimized. The adaptation process starts by presenting to the network a series of patterns for which the desired outputs are *a priori* known, and computing a total error function $E = \sum_{k=1}^{P} E_k$. In this equation, $P$ is the number of patterns and $E_k$ is the partial network error with respect to the $k$th pattern, computed by summing the squared difference between the actual network outputs and the desired outputs for this pattern. The training patterns can be presented numerous times to the network. Each pass of all the patterns that belong to the training set, $T$, is called a *training epoch*. The total number of epochs required can be considered as the speed of the training method. Several training techniques can be found in [13,16–20].

In this paper, we focus on FNNs for the computation of the Boolean function derived by elliptic curve cryptography defined in Eq. (1). For the general problem of the computation of a Boolean function by FNNs, the following theorem proved in [21], supports the effectiveness of the proposed approach.

**Theorem 3.** *There is a threshold network with one hidden layer capable of computing any Boolean function.*

## 4. Experimental setup and results

Training ANNs with threshold units requires the use of training methods that do not employ information by the derivatives of the error function. Furthermore, as shown in [22], analog neural networks can be more powerful than neural networks using thresholds, even for the computation of Boolean functions. Thus, we study the performance of ANNs using the hyperbolic tangent activation function:

$$F = \tanh \frac{\lambda x}{2} = \frac{2}{1 + \exp^{-\lambda x}} - 1,$$

which approximates a threshold function as $\lambda$ tends to infinity. In all experiments for addressing of the problem at hand, the output layer consists of two neurons, and the neuron with the highest output value determines the class in which the computed bit is classified. Thus, if the first neuron's output value is smaller than the value of the second neuron, the bit is considered to belong to Class 0, which corresponds to a "0" value of the bit, and vice versa. The prescribed setting enables us to use training methods that employ derivatives of the error function. In particular, we have studied the performance of three training algorithms each from a different category of training algorithms, namely the Resilient Back Propagation method (RPROP) [19], the Adaptive On-line Back Propagation method (AOBP) [16] and the Differential Evolution algorithm (DE) [23].

Regarding the topology of the networks studied, the "optimal" network topology for any particular problem is quite difficult and remains an open problem. To this end, we tested a variety of topologies with various numbers of neurons at each layer. We report only the best results obtained for each problem.

For the construction of the datasets the ECC_LIB library for elliptic curve cryptography [24] was used. The performance of ANNs was tested for three different datasets of the considered Boolean function that correspond

Table 1
Results for $p$ of bit length 14, using 56-3-2 topology

| Epochs | | Train | | | Test | | |
|---|---|---|---|---|---|---|---|
| | | Class 0 | Class 1 | Accuracy (%) | Class 0 | Class 1 | Accuracy (%) |
| 500 | Class 0 | 168 | 33 | 83.58 | 30 | 24 | 55.56 |
| | Class 1 | 48 | 151 | 75.88 | 23 | 23 | 50.00 |
| 650 | Class 0 | 184 | 17 | 91.54 | 33 | 21 | 61.11 |
| | Class 1 | 32 | 167 | 83.92 | 23 | 23 | 50.00 |
| 700 | Class 0 | 183 | 18 | 91.04 | 33 | 21 | 61.11 |
| | Class 1 | 30 | 169 | 84.92 | 21 | 25 | 54.35 |
| 1000 | Class 0 | 186 | 15 | 92.54 | 33 | 21 | 61.11 |
| | Class 1 | 25 | 174 | 87.44 | 18 | 28 | 60.87 |

Table 2
Results for $p$ of bit length 20, using 80-3-2 topology

| Epochs | | Train | | | Test | | |
|---|---|---|---|---|---|---|---|
| | | Class 0 | Class 1 | Accuracy (%) | Class 0 | Class 1 | Accuracy (%) |
| 2000 | Class 0 | 186 | 14 | 93.0 | 32 | 26 | 55.17 |
| | Class 1 | 23 | 177 | 88.5 | 17 | 25 | 59.52 |
| 3000 | Class 0 | 191 | 9 | 95.5 | 30 | 28 | 51.72 |
| | Class 1 | 19 | 181 | 90.5 | 21 | 21 | 50.00 |
| 4000 | Class 0 | 194 | 6 | 98.0 | 32 | 26 | 55.17 |
| | Class 1 | 18 | 182 | 91.0 | 19 | 23 | 54.76 |
| 6000 | Class 0 | 196 | 4 | 98.0 | 33 | 25 | 56.90 |
| | Class 1 | 17 | 183 | 91.5 | 20 | 22 | 52.38 |

to randomly chosen $p$'s of bit length 14, 20, and 32, respectively, where $\mathbb{F}_p$ is the finite field over which the elliptic curve is constructed.

At each experiment the dataset was randomly partitioned into a training set and a test set. Four fifths of the dataset were assigned to the training set and the remaining one fifth comprised the test set.

To evaluate the network performance, first we measure the average of the percentage of the training set over all 10 experiments, for which the network was able to correctly predict the least significant bit. Then, the network's performance is evaluated by measuring the average percentage of the test set over all experiments.

The best results, for the prescribed setting and $\lambda = 1$, were obtained using the AOBP method and are reported in Tables 1–3, respectively.

The results indicate that for three bit lengths, ANNs are able to adapt to the training data with an average accuracy of 90%. With respect to the test sets, ANNs achieved for all three bit lengths an average accuracy of 57%, i.e. a slightly higher than random selection. Regarding the training epochs required in each case, as the bit length of $p$ increases, more epochs are needed to achieve the same accuracy.

Another interesting finding regarding the training set, is that the network is able to learn the training patterns and respond correctly about the least significant bit of the discrete logarithm, using less storage than that required by the corresponding dataset. The results for the data compression are reported in Table 4. Regarding the notation of Table 4, "BL($p$)" denotes the bit length of $p$, "Data Stor." denotes the storage bits required for the dataset, "ANN Stor." denotes the storage bits required for the network weights and "Accuracy" corresponds to the accuracy of the network to identify the desired value for both classes.

## 5. Conclusions

In this paper we study the performance of Artificial Neural Networks on the problem of computing the least significant bit of the discrete logarithm of a point over elliptic curves. The computation of the least significant bit of the discrete logarithm over elliptic curves with known odd order is important for cryptographic applications as it leads

Table 3
Results for $p$ of bit length 32, using 128-3-2 topology

| Epochs | | Train | | | Test | | |
|---|---|---|---|---|---|---|---|
| | | Class 0 | Class 1 | Accuracy (%) | Class 0 | Class 1 | Accuracy (%) |
| 4000 | Class 0 | 193 | 5 | 97.47 | 36 | 21 | 63.16 |
| | Class 1 | 16 | 186 | 92.08 | 20 | 23 | 53.49 |
| 5000 | Class 0 | 193 | 5 | 97.47 | 36 | 21 | 63.16 |
| | Class 1 | 15 | 187 | 92.57 | 19 | 24 | 55.81 |
| 8000 | Class 0 | 193 | 5 | 97.47 | 35 | 22 | 61.40 |
| | Class 1 | 14 | 188 | 93.07 | 18 | 25 | 58.14 |
| 9000 | Class 0 | 193 | 5 | 97.47 | 35 | 22 | 61.40 |
| | Class 1 | 14 | 188 | 93.07 | 16 | 27 | 62.79 |

Table 4
Data compression results

| BL($p$) | Data stor. | ANN stor. | Accuracy (%) |
|---|---|---|---|
| 14 | 23 200 | 8400 | 89.99 |
| 20 | 32 800 | 11 856 | 94.75 |
| 32 | 52 000 | 18 768 | 95.27 |

to the computation of all bits of the discrete logarithm. The results of this first attempt to address the specific problem using ANNs, indicate that they are able to adapt to the data presented with high accuracy, while the response of ANNs to unknown data is slightly higher than a random selection. Another important finding is that ANNs require a small amount of storage for the known patterns in contrast to the storage needed for the dataset itself.

In a future correspondence we intend to further study the performance of ANNs for larger values of $p$ and elliptic curves of different order, as well as other related problems, such as the computation of the order of elliptic curves.

## Acknowledgements

The authors would like to acknowledge partial support by the “Archimedes” research programme awarded by the Greek Ministry of Education and Religious Affairs and the European Union.

## References

[1] N. Koblitz, Elliptic curve cryptosystems, Math. Comp. 48 (1987) 203–209.
[2] V. Miller, Uses of elliptic curves in cryptography, LNCS 218 (1986) 417–426.
[3] D. Coppersmith, I. Shparlinski, On polynomial approximation of the discrete logarithm and the Diffie–Hellman mapping, J. Cryptology 13 (2000) 339–360.
[4] E.C. Laskari, G.C. Meletiou, M.N. Vrahatis, Aitken and neville inverse interpolation methods over finite fields, Appl. Numer. Anal. Comput. Math. 2 (1) (2005) 100–107.
[5] U. Maurer, S. Wolf, The relationship between breaking the Diffie–Hellman protocol and computing discrete logarithms, SIAM J. Comput. 28 (1999) 1689–1721.
[6] G.C. Meletiou, G.L. Mullen, A note on discrete logarithms in finite fields, Appl. Algebra Engrg. Comm. Comput. 3 (1) (1992) 75–79.
[7] A. Winterhof, Polynomial interpolation of the discrete logarithm, Des. Codes Cryptogr. 25 (1) (2002) 63–72.
[8] I. Blake, G. Seroussi, N. Smart, Elliptic Curves in Cryptography, in: London Mathematical Society Lecture Notes Series, vol. 265, Cambridge University Press, 1999.
[9] J.H. Silverman, The Arithmetic of Elliptic Curves, Springer-Verlag, 1986.
[10] R. Peralta, Simultaneous security of bits in the discrete log, in: LNCS, vol. 219, 1986, pp. 62–72.
[11] D. Stinson, Cryptography: Theory and Practice (Discrete Mathematics and Its Applications), Second ed., CRC Press, 2002.
[12] I. Shparlinski, Cryptographic Applications of Analytic Number Theory, Birkhauser, 2003.
[13] S. Haykin, Neural Networks, Macmillan College Publishing Company, 1999.
[14] K. Hornik, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.
[15] A. Pincus, Approximation theory of the MLP model in neural networks, in: Acta Numerica, Cambridge University Press, 1999, pp. 143–195.
[16] G.D. Magoulas, V.P. Plagianakos, M.N. Vrahatis, Adaptive stepsize algorithms for on-line training of neural networks, Nonlinear Anal. TMA 47 (5) (2001) 3425–3430.

[17] G.D. Magoulas, M.N. Vrahatis, G.S. Androulakis, Effective backpropagation training with variable stepsize, Neural Netw. 10 (1) (1997) 69–82.

[18] G.D. Magoulas, M.N. Vrahatis, G.S. Androulakis, Increasing the convergence rate of the error backpropagation algorithm by learning rate adaptation methods, Neural Comput. 11 (7) (1999) 1769–1796.

[19] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, in: Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, 1993, pp. 586–591.

[20] M.N. Vrahatis, G.S. Androulakis, J.N. Lambrinos, G.D. Magoulas, A class of gradient unconstrained minimization algorithms with adaptive stepsize, J. Comput. Appl. Math. 114 (2) (2000) 367–386.

[21] M. Anthony, Boolean functions and artificial neural networks, CDAM Research Report LSE-CDAM-2003-01, London WC2A 2AE, UK.

[22] B. DasGupta, G. Schnitger, Analog versus discrete neural networks, Neural Comput. 8 (4) (1996) 805–818.

[23] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.

[24] E. Konstantinou, Y. Stamatiou, C. Zaroliagis, A software library for elliptic curve cryptography, LNCS 2461 (2002) 625–637.