Livre des résumés Eurocode '94
Pascale Charpin Ed. INRIA, 1994.

# Neural Networks and their Cryptographic Applications

David POINTCHEVAL

Laboratoire d'informatique
École Normale Supérieure

**Abstract.** Identification is a useful cryptographic tool. Since the zero-knowledge theory appeared [3], several interactive identification schemes have been proposed (in particular Fiat-Shamir [2] and its variants [4, 6, 5], Schnorr [8]). These identifications are based on number theoretical problems. More recently, new schemes appeared with the particularity that they are more efficient from the computational point of view and their security is based on $\mathcal{NP}$-complete problems: PKP (Permuted Kernels Problem) [9], SD (Syndrome Decoding) [11] and CLE (Constrained Linear Equations) [12].
We present a new linear $\mathcal{NP}$-complete problem which comes from neural networks and learning machines: the Perceptrons Problem. We have some constraints, $m$ vectors $X^i$ of $\{-1,+1\}^n$, and we want to find a vector $V$ of $\{-1,+1\}^n$ such that $X^i \cdot V \geq 0$ for all $i$.
Next, we provide some zero-knowledge interactive identification protocols based on this problem, with an evaluation of its security. Eventually, those protocols are well suited for smart card applications.

## 1 Introduction

An interactive identification protocol involves two persons Alice and Bob, where Alice wants to prove that she is really Alice, interactively. Alice has a public key which everybody knows, and a secret key associated to the public key. She is the only one who knows the secret key, and nobody can compute it. To prove her identity, Alice proves that she knows a secret key associated to her public key. Recently, the zero-knowledge theory showed that we can do this without revealing anything about the secret key. The first efficient zero-knowledge protocols were based on number theoretical problems (Fiat-Shamir [2] and its variants [4, 6, 5], Schnorr [8]). They have two major disadvantages:

- The hardness of the problems used (factorization and discrete logarithm) is not proved. Moreover, more efficient algorithms and computers threaten them.
- Arithmetic operations are very expensive (modular multiplications, modular exponentiations).

However, since 1989, new schemes have appeared, which rely on $\mathcal{NP}$-complete problems, and require only operations over small numbers or even on bits: PKP (Permuted Kernels Problem) [9], SD (Syndrome Decoding) [11] or CLE (Constrained Linear Equations) [12].

This paper introduces another linear scheme based on the Perceptrons Problem, an $\mathcal{NP}$-complete problem, which seems to be well suited for smart card applications.

## 2 The Perceptrons Problem

The following problem appears in physics and the study of the Ising's perceptrons, and in artificial intelligence with neural networks and learning machines. We call it *The Perceptrons Problem*.

**Definition 1.** We call an $\varepsilon$-vector a vector which components are either $-1$ or $+1$.

**Definition 2.** The Perceptrons Problem PP:

Input : $X^1, \ldots, X^m$, $m$ $\varepsilon$-vectors of size $n$.
Question : Is there an $\varepsilon$-vector $Y$ of size $n$ such that for all $j$, $X^j \cdot Y \geq 0$ ?

We will modify the original problem a little bit to make the interactive proof easier, and to make the practical attacks more difficult:

**Definition 3.** Modified Perceptrons Problem PP'

Input : $X^1, \ldots, X^m$, $m$ $\varepsilon$-vectors of size $n$.
$T$, a vector of size $m$ with integer and nonnegative components.
Question : Are there an $\varepsilon$-vector $Y$ of size $n$ and a permutation, $\sigma$, over $\{1, \ldots, m\}$ , such that for all $j$, $X^j \cdot Y = T_{\sigma(j)}$ ?

**Theorem 4.** *Both* PP *and* PP*' are $\mathcal{NP}$-complete.*

Furthermore, we can prove that PP is difficult to approximate in the sense of Papadimitriou and Yannakakis [7].

## 3 Cryptographic applications

### 3.1 Finite field

For cryptographic use, we must bound the size of the numbers used in order to store them on a constant number of bits. So we will work in the finite field with $p$ elements, and $m$ vectors $X^i$ of odd size $n$. We bound the value of the dot products by a positive odd integer $t$. We can prove that if $n$, $p$ and $t$ are such that $2p > n + t$ we have:

$$X^i \cdot Y \geq 0 \Longleftrightarrow X^i \cdot Y \bmod p \in \{1, 3, \ldots, t\}$$

### 3.2 Size of the problem

As we will see later, the values of $m$ and $n$ depend on the efficiency of the attacks. However we can already prove a relation between them. Indeed, for the problem PP', we want that only one solution exists. Therefore, we want to know:

- the number of solutions for PP.
  As soon as we know the existence of at least one solution, we can evaluate the number of solutions for an instance of PP by a combinatorial method: $N(m, n)$

– the probability to obtain a given distribution $d$ for $T$: $P_{m,n,d}$

There is just one solution, if we choose $m$ and $n$ such that $N(m,n) \times P_{m,n,d} \approx 1$ for all distribution $d$:

| $m$ | max $n$ | Number of solutions for PP | Least Probability | Number of solutions for PP' |
|---|---|---|---|---|
| 101 | 117 | $9.4 \times 10^{9}$ | $1.0 \times 10^{-10}$ | 0.94 |
| 121 | 137 | $1.6 \times 10^{11}$ | $6.0 \times 10^{-12}$ | 1.06 |
| 141 | 157 | $2.6 \times 10^{12}$ | $3.5 \times 10^{-13}$ | 0.92 |
| 151 | 167 | $7.4 \times 10^{12}$ | $1.4 \times 10^{-13}$ | 1.04 |

## 4 Possible attacks

We tried several attacks against PP and PP' in order to evaluate the security of a possible protocol. But since there is no algebraic structure in those problems, no manipulation of the matrix (the matrix which rows are the given vectors, $X^i$) will leave the problem unchanged (manipulation like Gaussian elimination, used in the past against PKP, CLE or any problem based on error correcting codes will not help here). So, it seems that only (more or less intelligent) exhaustive search or probabilistic attacks would succeed.

### 4.1 The majority vector

The first approximation we want to try is the *majority vector* $M$:

$$\begin{cases} M_j = +1 \text{ if } \#\{i | X^i_j = +1\} > \frac{n}{2} \\ M_j = -1 \text{ otherwise} \end{cases}$$

**Theorem 5.** *For an $m \times n$-instance constructed randomly, with solution $V$ and $m \simeq n$, the average Hamming distance between $V$ and $M$, is roughly $n \cdot (\frac{1}{2} - \frac{1}{\pi})$.*

Then we thought of changing 18% of the components of $M$, and trying the products. But, even if we begin with the most litigious values there are on average $0.8 \times \binom{n}{0.18n}$ possibilities to try.
We can already fix a bound for $n$ (and $m$) to overtake the usual workload of $2^{64}$: $n \geq 100$.

### 4.2 Simulated annealing

Because of the inefficiency of previous attacks, we tried the well-known probabilistic algorithm from artificial intelligence, known as *simulated annealing* [10]. This attack tries to minimize a function, *Energy*, defined on a finite metric space, in a probabilistic way. Simulated annealing algorithms are an improvement of gradient descent algorithms. Whereas gradient descent algorithms can converge to a local minimum and stay there, simulated annealing algorithms try, with

some small random perturbations, to go away. These perturbations, which may be important at the beginning, have to decrease to zero.
It is clear that such an algorithm is not efficient on every energy function. In fact, this function must be roughly "continuous". For this reason, simulated annealing doesn't seem to be suited for PP' but it should be ideal for PP. This algorithm turned out to be the most efficient.
Then, we can evaluate the workload of such an attack with probability of success equal to 0.5:

| size | number of solutions | time for a solution (seconds) | time Solution Pr = 1/2 (seconds) | | Workload* $2^n$ elementary operations |
|---|---|---|---|---|---|
| $31 \times 31$ | 40 | 40 | $1.10^3$ | 15 minutes | 36 |
| $51 \times 51$ | 400 | 60 | $17.10^3$ | 4 hours 45 minutes | 40 |
| $71 \times 71$ | 7.500 | 75 | $400.10^3$ | 4 days 15 hours | 44 |
| $101 \times 117$ | $4.7\ 10^9$ | 85 | $399.10^9$ | 12 thousand years | 64 |
| $121 \times 137$ | $8.7\ 10^{10}$ | 130 | $11.10^{12}$ | 350 thousand years | 68 |
| $123 \times 123$ | $9.8\ 10^6$ | 105 | $720.10^6$ | 22 years 10 months | 55 |
| $151 \times 151$ | $306\ 10^6$ | 1800 | $385.10^9$ | 12 thousand years | 64 |
| $151 \times 167$ | $3.7\ 10^{12}$ | 180 | $666.10^{12}$ | 21 million years | 74 |
| $171 \times 171$ | $6.7\ 10^9$ | 7200 | $34.10^{12}$ | 1 million years | 70 |
| $189 \times 189$ | $78\ 10^9$ | 36000 | $2.10^{15}$ | 63 million years | 76 |

* workload estimated using a 60-70 MIPS processor speed

Those various tests allow us to suggest sizes which seem to be good for a secure cryptographic use. The workload usually wanted is about $2^{64}$, so we get $m = 121$ and $n = 137$, or any other greater size.
In addition, whatever the probabilistic attack, it will not be able to differentiate the good solution of PP' from any solution of PP. So, even if we supposed a quick attack for PP (an $\mathcal{NP}$-complete problem) which would need only 1 second to find a solution for a $141 \times 157$-sized instance, the workload would remain above $2^{64}$.

## 5 Protocols

Common data: $p$, $n$, $t$, integers such that $2p > t + n$ and $h$ a collision-free, random hash function.

Let $A$ be a matrix of size $m \times n$ which rows represent a Perceptrons Problem instance with $V$ as a solution. Let $D$ be a random permutation of $AV$.
Public key : ($A$, $D$)
Secret key : $V$

The prover selects
- a random permutation $P$ over $\{0, \ldots, m-1\}$ (to mix the rows of $A$.)
- a random signed permutation $Q$ over $\{0, \ldots, n-1\}$
  (to mix the columns of $A$, and to multiply them randomly by $+1$ or $-1$.)
- a random vector $W$ of $\mathbb{Z}_p^n$

### 5.1 Three pass identification protocol (3p zk)

1. The prover computes $A' = PAQ$, $V' = Q^{-1}V$, $R = W + V'$
   and $h_0 = h(P|Q)$, $h_1 = h(W)$, $h_2 = h(R)$, $h_3 = h(A'W)$, $h_4 = h(A'R)$
   and sends $(h_0, h_1, h_2, h_3, h_4)$ to the verifier.
2. The verifier randomly selects an element $c$ of $\{0, 1, 2, 3\}$
   and sends $c$ to the prover.
3. The prover sends:
   if $c = 0 : (P, Q, W)$
   if $c = 1 : (P, Q, R)$
   if $c = 2 : (A'W, A'V')$
   if $c = 3 : (W, V')$
4. The verifier checks:
   $h_0 = h(P|Q)$, $h_1 = h(W)$ and $h_3 = h(PAQW)$.
   $h_0 = h(P|Q)$, $h_2 = h(R)$ and $h_4 = h(PAQR)$.
   $h_3 = h(A'W)$, $h_4 = h(A'W + A'V')$
   and $\exists\sigma$ such that $\sigma(A'V') = D$.
   $h_1 = h(W)$, $h_2 = h(W + V')$ and $V' \in \{-1, +1\}^n$.

### 5.2 Five pass identification protocol (5p zk)

1. The prover computes $A' = PAQ$, $V' = Q^{-1}V$
   and $h_0 = h(P, Q)$, $h_1 = h(W|V')$, $h_2 = h(A'W|A'V')$
   and sends $(h_0, h_1, h_2)$ to the verifier.
2. The verifier randomly selects an element $k$ of $Z^\star(p)$
   and sends $k$ to the prover.
3. The prover computes $R = kW + V'$ and $h_3 = h(R)$, $h_4 = h(A'R)$
   and sends $(h_3, h_4)$ to the verifier.
4. The verifier randomly selects an element $c$ of $\{0, 1, 2\}$
   and sends $c$ to the prover.
5. The prover sends:
   if $c = 0 : (P, Q, R)$
   if $c = 1 : (A'W, A'V')$
   if $c = 2 : (W, V')$
6. The verifier checks:
   $h_0 = h(P|Q)$, $h_3 = h(R)$ and $h_4 = h(PAQR)$
   $h_2 = h(A'W|A'V')$, $h_4 = h(kA'W + A'V')$
   and $\exists\sigma$ such that $\sigma(A'V') = D$.
   $h_1 = h(W|V')$, $h_3 = h(kW + V')$
   and $V' \in \{-1, +1\}^n$

### 5.3 Properties

**Theorem 6.** *Both **3p zk** and **5p zk** protocols are Interactive Proof Systems for PP'.*

**Lemma 7.** *Assume that some probabilistic polynomial-time adversary is accepted with probability greater than* $\left(\frac{3}{4}\right)^r + \epsilon$ *after r rounds, then there exists a polynomial-time probabilistic machine which extracts the secret key S from the public data or outputs collisions for the commitment function, with overwhelming probability.*

**Lemma 8.** *Assume that some probabilistic polynomial-time adversary is accepted with probability greater than* $\left(\frac{2p-1}{3(p-1)}\right)^r + \epsilon$ *after r rounds, then there exists a polynomial-time probabilistic machine which extracts the secret key S*

*from the public data or outputs collisions for the commitment function, with overwhelming probability.*

Using the idea of resettable simulation [3], in the *random oracle model* [1], it can be shown that both protocols are zero-knowledge. Alternatively, one has to assume specific statistical independance properties for the hash function. A light version (**3p light** and **5p light**) of those protocols, which reduces the number of required rounds, can be made but it is no more zero-knowledge. However, the given information is quite small and seems to be unusable.

## 6 Performances

The performances of this scheme are similar to those of the already existing linear ones:

- As we can see in the following figure, to obtain the standard security level of $10^{-6}$, with a very secure secret key, an identification requires between 4 and 6 kbytes of communication between the prover and the verifier (to be compared to the 3kbytes for PKP and the 5kbytes for SD). And we can improve them by the use of hash trees.
- Moreover, all the operations are additions and subtractions between small integers (less than one byte). They are well suited to a very minimal environment of 8-bit processors.
- They require very little RAM.
- We notice that public and secret keys are very small (less than 20 bytes for the secret key, and about 32 for the public one). But we should not forget that as for PKP, SD and CLE, this scheme is not identity based. It means that public keys have to be certified by an authority.

## 7 Conclusion

We have defined a new identification scheme which is very easy to implement on every kind of smart card because of its very simple operations, and the small size of the data. We welcome attacks from readers.

## References

1. M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, U.S.A., 1993. ACM press.
2. A. Fiat and A. Shamir. How to Prove Yourself: practical solutions of identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology – Proceedings of CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa-Barbara, California, 1987. Springer-Verlag.
3. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proceedings of the 17th ACM Symposium on the Theory of Computing STOC*, pages 291–304, Providence, Rhode Island, U.S.A., 1985. ACM Press.

| Protocol | $m$ | $n$ | $p$ | rounds | $t$ | Commitments size (bytes) | Communication bytes | kbytes |
|---|---|---|---|---|---|---|---|---|
| (3p zk) | 101 | 117 | 127 | 48 | 33 | 8 | 5386 | 5.26 |
| | | | | | | 16 | 7306 | 7.14 |
| | 121 | 137 | 127 | 48 | 33 | 8 | 5956 | 5.82 |
| | | | | | | 16 | 7876 | 7.69 |
| | 151 | 151 | 127 | 48 | 65 | 8 | 6846 | 6.68 |
| | | | | | | 16 | 8766 | 8.56 |
| | 151 | 167 | 127 | 48 | 65 | 8 | 7038 | 6.87 |
| | | | | | | 16 | 8958 | 8.75 |
| (5p zk) | 101 | 117 | 127 | 35 | 33 | 8 | 4611 | 4.50 |
| | | | | | | 16 | 6011 | 5.87 |
| | 121 | 137 | 127 | 35 | 33 | 8 | 5165 | 5.04 |
| | | | | | | 16 | 6565 | 6.41 |
| | 151 | 151 | 127 | 35 | 65 | 8 | 6030 | 5.89 |
| | | | | | | 16 | 7430 | 7.26 |
| | 151 | 167 | 127 | 35 | 65 | 8 | 6217 | 6.07 |
| | | | | | | 16 | 7617 | 7.44 |
| (5p light) | 101 | 117 | 127 | 20 | 33 | 8 | 3379 | 3.30 |
| | | | | | | 16 | 3859 | 3.77 |
| | 121 | 137 | 127 | 20 | 33 | 8 | 3854 | 3.76 |
| | | | | | | 16 | 4334 | 4.23 |
| | 151 | 151 | 127 | 35 | 65 | 8 | 4595 | 4.49 |
| | | | | | | 16 | 5075 | 4.96 |
| | 151 | 167 | 127 | 35 | 65 | 8 | 4755 | 4.64 |
| | | | | | | 16 | 5235 | 5.11 |


4. L. C. Guillou and J.-J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In C. G. Günter, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '88*, volume 330 of *Lecture Notes in Computer Science*, pages 123–128, Davos, Switzerland, 1988. Springer-Verlag.
5. K. Ohta and T. Okamoto. A Modification of the Fiat-Shamir Scheme. In S. Goldwasser, editor, *Advances in Cryptology – Proceedings of CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 232–243, Santa-Barbara, California, 1989. Springer-Verlag.
6. H. Ong and C.P. Schnorr. Fast Signature Generation with a Fiat-Shamir-Like Scheme. In I. B. Damgard, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '90*, volume 473 of *Lecture Notes in Computer Science*, pages 432–440, Aarhus, Denmark, 1991. Springer-Verlag.
7. C. Papadimitriou and M. Yannakakis. Optimization, Approximation, and Complexity Classes. *Journal of Computer and Systems Sciences*, 43:425–440, 1991.
8. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In G. Brassard, editor, *Advances in Cryptology – Proceedings of CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 235–251, Santa-Barbara, California, 1990. Springer-Verlag.
9. A. Shamir. An Efficient Identification Scheme Based on Permuted Kernels. In G. Brassard, editor, *Advances in Cryptology – Proceedings of CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 606–609, Santa-Barbara, California, 1990. Springer-Verlag.
10. M. Skubiszewski. *Optimisation par Recuit Simulé : mise en œuvre matérielle de la machine de Boltzmann, application à l'étude des suites synchronisantes*. PhD thesis, Université d'Orsay, June 1993.
11. J. Stern. A New Identification Scheme Based on Syndrome Decoding. In D. R. Stinson, editor, *Advances in Cryptology – proceedings of CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21, Santa-Barbara, California, 1994. Springer-Verlag.
12. J. Stern. Designing Identification Schemes with Keys of Short Size. In Y. G. Desmedt, editor, *Advances in Cryptology – proceedings of CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 164–173, Santa-Barbara, California, 1994. Springer-Verlag.