# Some combinatorial problems of importance to cryptography

(revised 2/29/96)

René PERALTA & Eiji OKAMOTO

School of Information Science
Japan Advanced Institute of Science and Technology
15 Asahidai, Tatsunokuchi, Nomi, Ishikawa, 923-12 JAPAN
Supported by KOMATSU endowed chair.
Email: peralta@jaist.ac.jp

**ABSTRACT**

We discuss two combinatorial problems for which we have only sub-optimal solutions. We describe known solutions and we explain why better solutions would be of importance to Cryptography. The areas of application are in improving the efficiency of Zero-Knowledge proofs, in relating the quadratic residuosity problem to the problem of integer factorization, and in analyzing some cryptographic schemes based on number-theory.

# 1 Reduction of integer factorization to a combinatorial optimization problem plus a quadratic residuosity oracle.

The "quadratic residuosity assumption" (QRA) states that having any significant advantage in deciding quadratic residuosity is asymptotically infeasible for random elements of $Z_N^*$ whose Jacobi symbol is $+1$, whenever $N$ is an RSA number [4]. This assumption is widely used in cryptography, yet there is no known reduction from integer factorization to deciding quadratic residuosity (the other direction is easy). It is conceivable that factoring is hard but deciding quadratic residuosity is easy. Deciding this matter is a major open problem in computational number theory.

In common parlance, an "oracle" is a person (as a priestess of ancient Greece) through whom a deity is believed to speak. Computer Science has adopted this word to mean an abstract computing device which answers questions you pose to it in constant time. There are many types of oracles. We speak of "faulty" oracles, of "cheating" oracles, of "mostly correct" oracles, etc... In this note, the kind of oracle we need is one which is bound to answer yes-no questions truthfully for $\frac{1}{2}+\epsilon$ of the space of possible questions($\epsilon$ is any constant greater than 0). The reason such a "weak" oracle is enough is that the question "is $x$ a quadratic residue modulo $N$" can be randomized by asking the alternative question "is $r^2x \bmod N$ a quadratic residue modulo $N$", where $r$ is a randomly chosen number. You must not show $r$ to the oracle. If you do, a nasty one would be able to cheat you (one never knows with ancient Greek priestesses).

## 1.1 A quadratic residuosity oracle modulo $N = P\cdot Q$ can be used to decide quadratic residuosity modulo $P$.

Suppose $P$ is a secret large prime. If we could compute $P \bmod q$ for arbitrary small prime $q$, then we could find $P$ via the Chinese Remainder Theorem. Thus it should not be surprising that, for integers of the form $N = PQ$, it is not known how to restrict the possible values of $P \bmod q$ in any way whatsoever (except for the trivial case $q = 2$). We will show that, if we are given a Quadratic Residuosity Oracle, then half the possible values of $P \bmod q$ can be easily ruled out.

Let $N = PQ \quad (P < Q)$ and let $\alpha \in Z_N^*$ be such that $\alpha$ is a quadratic nonresidue modulo $P$ and a quadratic residue modulo $Q$. It is not known how to efficiently decide whether $\alpha$ has these properties. However, the probability that a random $\alpha \in Z_N^*$ with $\left(\frac{\alpha}{N}\right) = -1$ satisfies the properties is $\frac{1}{2}$. Thus we may simply assume we know such an $\alpha$.

Now assume deciding quadratic residuosity modulo $N$ is feasible. To decide whether an arbitrary $x \in Z_N^*$ is a quadratic residue modulo $P$, apply the following rules:

- if $\left(\frac{x}{N}\right) = 1$ then $x$ is a quadratic residue modulo $P$ if and only if $x$ is a quadratic residue modulo $N$.
- if $\left(\frac{x}{N}\right) = -1$ then $x$ is a quadratic residue modulo $P$ if and only if $\alpha x$ is a quadratic nonresidue modulo $N$.

This technique can easily be extended to the case where $N$ has any constant number of factors.

## 1.2 Reduction to a combinatorial problem.

We will now assume that we have a quadratic residue oracle modulo $P$ where $P$ is a fixed factor of $N$. For simplicity of exposition, we will assume that $P$ is 1 modulo 4. This assumption is made so that $\left(\frac{P}{q}\right) = \left(\frac{q}{P}\right)$ for all odd primes $q$.

Let $q_1, \ldots, q_k$ be a set of small odd primes, where $k = \lfloor \lg N \rfloor$. Let $R = \prod_{i=1}^{k} q_i$ and note that $R > N$. Define

$$S_i = \left\{ a \in Z_{q_i} \;\middle|\; \left(\frac{a}{q_i}\right) = \left(\frac{q_i}{P}\right) \right\} \quad \text{for } i = 1, \ldots, k.$$

and note that $a \in S_i$ if and only if $\left(\frac{a}{q_i}\right) = \left(\frac{P}{q_i}\right)$. This implies that computing the set $S_i$ allows one to discard half the possible values of $P \bmod q_i$.

For any integer $a$, define $\mu_i(a)$ as the unique integer in the range $\{0, \ldots, R-1\}$ which satisfies

$$\mu_i(a) \bmod q_j \equiv \begin{cases} 0 & \text{if } j \neq i \\ a & \text{otherwise} \end{cases}$$

The formula

$$\mu_i(a) = \left( a \cdot \left(\frac{R}{q_i}\right) \cdot \left[ \left(\frac{R}{q_i}\right)^{-1} \bmod q_i \right] \right) \bmod R$$

gives an efficient way to calculate $\mu_i$. Note that division is integer division and the inverse is modulo $q_i$.

Let

$$T_i = \{\mu_i(a) \mid a \in S_i\} \quad \text{for } i = 1, \ldots, k.$$

Then the Chinese Remainder Theorem implies that $P$ is of the form $\sum_{i=1}^{k} t_i - lR$, where $t_i \in T_i$ for each $i$. Since $l \in \{0, \ldots, k\}$, and $k$ is logarithmic in $N$, we may assume we know $l$. Therefore $P$ can be computed if we can find the $t_i$'s.

Note that the set $T_i$ can be computed in polynomial time. The set $S_i$ can be computed in polynomial time with the aid of a quadratic residue oracle modulo $P$.

Now consider what happens if we choose $s_i \in T_i$ for each $i$ and compute $g = (\sum_{i=1}^{k} s_i) \bmod R$. Let $G$ be the set of $g$'s computable in this way. We expect $G$ to look like a set of random numbers modulo $R$. Since $G$ contains about $\prod_{i=1}^{k}(q_i/2) \approx R/N$ elements, we expect the smallest $g \in G$ to be about $R(R/N)^{-1} = N$. But one of the $g$'s is $P$, which is $O(N^{1/3})$. Therefore $P$ is an unusually small number constructible by choosing one element from each $T_i$, adding the elements, and subtracting $lR$.

> We conclude that we might find $P$ if we can choose exactly one $t_i$ from each $T_i$ in such a way that the distance $\mid lR - \sum_{i=1}^{k} t_i \mid$ is minimized.

This is a variation on the well known "subset sum" problem. Although this problem is NP-complete, there is no reason to believe these instances to be uniformly hard. Problems such as these have been successfully attacked using the Lenstra,Lenstra,Lovasz lattice reduction algorithm. The right lattice might provide a fast enough solution, on average, for this problem. In attempting to construct such a lattice, the main problem that arises is how to enforce the rule that exactly one of the $t_i$'s be chosen for each $i$.

It may well be that a heuristic solution can be found which does not use LLL. To date, our best solution takes time $O(\log^2(G))$, which is not nearly fast enough to solve the combinatorial problem for RSA modulus of size 512 bits (much less the 1024 bits which will surely become the standard). But not a significant amount of time has been dedicated to this problem.

What is of importance to cryptography here, is that it is the *concrete complexity* of the combinatorial problem which matters. Ten years ago it was thought that the concrete complexity of factoring 512-bit numbers was beyond anything achievable in the near future. A decade of work by many people is about to force RSA to change from 512 bits to 1024 bits. It is not that factoring has been conquered. It is that the *concrete complexity* of factoring 512 digit numbers is now known to be within reach. There is no way of predicting what a decade of work on the above combinatoric problem would yield as a "safe modulus even when QR fails". Furthermore, there are cryptographic schemes being used for which quadratic residuosity modulo $P$ is efficiently computable. Whenever the modulus $N$ is of the form $P \cdot Q^2$, we have $\left(\frac{x}{N}\right) = \left(\frac{x}{P}\right)$. This type of modulus is present in some very useful cryptographic schemes (see [3, 6]) which would be a pitty to have to do without.

## Remarks.

1. The set of $q_i$'s need not be the smallest $k$ odd primes. The analysis holds for any set of small primes. This is important because it means the reduction can be made into a randomized one: if one random set of primes fails, we can try a different one.

2. The analysis predicting that $P$ is the smallest element of $G$ holds for $k$ as small as $\lceil (\lg N)/3 \rceil$.

3. We realize some of the readers may not care much for the number theoretic aspects of all this. So that he/she can choose to ignore all the number theory above and still understand the combinatorial problem, we provide a toy example: you must choose one number from each column. If you choose the right numbers and add modulo $R = 37182145$, you will obtain a secret prime less than 50.

| | | | | | | |
|---|---|---|---|---|---|---|
| 22309287 | 21246940 | 30421755 | 17160990 | 30620590 | 31311280 | 3233230 |
| 14872858 | 5311735 | 16900975 | 5720330 | 26246220 | 9784775 | 19399380 |
| | 10623470 | 10140585 | 14300825 | 24059035 | 13698685 | 6466460 |
| | | 3380195 | 22881320 | 21871850 | 7827820 | 14549535 |
| | | 13520780 | 31461815 | 15310295 | 1956955 | 1616615 |
| | | | 20021155 | 13123110 | 17612595 | 9699690 |
| | | | | 10935925 | 33268235 | 25865840 |
| | | | | 6561555 | 11741730 | 4849845 |
| | | | | | 21526505 | 12932920 |
| | | | | | | 21015995 |
| | | | | | | 29099070 |

## 2 The matrices problem

The problem is to construct two very special $n \times n$ matrices $A, B$. Matrix $A$ will contain entries $a_i$ and 0's. Matrix $B$ will contain entries $b_i$ and 0's. We define $C$ to be the ordinary matrix product $A \cdot B$. Matrix $C$ will contain entries $c_i$. When $c_i = a_i \cdot b_i$ we call that a "good" entry in $C$, otherwise we call $c_i$ a "garbage" entry. The challenge is to place the $a_i$'s and $b_i$'s in $A$ and $B$ such that the product matrix $AB$ contains as many good entries as possible. If we put the $a_i$'s and the $b_i$'s on the diagonals of their respective matrices, then we obtain $n$ good entries. Let $\tau(n)$ denote the maximum number of good entries obtainable in $n \times n$ matrices. We know $n \leq \tau(n) \leq n^2$. On first impression one might be tempted to conclude that $\tau(n)$ is simply $n$. But consider the following $3 \times 3$ matrices.

$$\begin{pmatrix} a_1 & a_2 & 0 \\ a_3 & 0 & a_4 \\ 0 & a_5 & a_6 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} b_3 & b_1 & 0 \\ b_5 & 0 & b_2 \\ 0 & b_6 & b_4 \end{pmatrix}.$$

Multiplying the two matrices one obtains

$$\begin{pmatrix} a_1 b_3 + a_2 b_5 & a_1 b_1 & a_2 b_2 \\ a_3 b_3 & a_3 b_1 + a_4 b_6 & a_4 b_4 \\ a_5 b_5 & a_6 b_6 & a_5 b_2 + a_6 b_4 \end{pmatrix}.$$

The product matrix contains 6 good entries and 3 garbage entries. We may use these matrices as "templates" for the recursive construction of $3^k \times 3^k$ matrices. For example, taking $k = 2$ we have

$$\begin{pmatrix} A_1 & A_2 & 0 \\ A_3 & 0 & A_4 \\ 0 & A_5 & A_6 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} B_3 & B_1 & 0 \\ B_5 & 0 & B_2 \\ 0 & B_6 & B_4 \end{pmatrix}.$$

Multiplying the two $9 \times 9$ matrices one obtains

$$\begin{pmatrix} A_1 B_3 + A_2 B_5 & A_1 B_1 & A_2 B_2 \\ A_3 B_3 & A_3 B_1 + A_4 B_6 & A_4 B_4 \\ A_5 B_5 & A_6 B_6 & A_5 B_2 + a_6 B_4 \end{pmatrix}.$$

where now each $A_i$ and $B_i$ are $3 \times 3$ matrices. Notice that each $3 \times 3$ submatrix of the form $A_i B_i$ yields 6 good entries. The total number of good entries is thus 36. So we know $\tau(9) \geq 36$. If you attempt to beat this construction for $n = 9$ you will quickly get a feeling for the combinatorial explosion we are dealing with.

Letting $n = 3^k$, we now know that we can construct $n \times n$ matrices which yield $n^\alpha$ good entries, where $\alpha$ is $\log_3 6 > 1.63$. Thus, at least for $n$ a power of 3, we have $\tau(n) > n^{1.63}$.

After much searching Joan Boyar, was able to beat this with a paper-and-pencil construction [1]. Here is her construction, an $8 \times 8$ matrix product where the asterisk is used to denote garbage terms.

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & 0 & 0 & 0 & 0 \\ a_5 & 0 & a_6 & 0 & a_7 & 0 & a_8 & 0 \\ a_9 & a_{10} & 0 & 0 & a_{11} & a_{12} & 0 & 0 \\ a_{13} & 0 & 0 & a_{14} & 0 & a_{15} & a_{16} & 0 \\ 0 & a_{17} & a_{18} & 0 & a_{19} & 0 & 0 & a_{20} \\ 0 & 0 & a_{21} & a_{22} & 0 & 0 & a_{23} & a_{24} \\ 0 & a_{25} & 0 & a_{26} & 0 & a_{27} & 0 & a_{28} \\ 0 & 0 & 0 & 0 & a_{29} & a_{30} & a_{31} & a_{32} \end{pmatrix} \times \begin{pmatrix} b_1 & 0 & 0 & 0 & b_{13} & b_5 & b_9 & 0 \\ 0 & b_2 & 0 & 0 & b_{25} & b_{17} & 0 & b_{10} \\ 0 & 0 & b_3 & 0 & b_{21} & 0 & b_{18} & b_6 \\ 0 & 0 & 0 & b_4 & 0 & b_{22} & b_{26} & b_{14} \\ b_{19} & b_7 & b_{11} & 0 & b_{29} & 0 & 0 & 0 \\ b_{27} & b_{15} & 0 & b_{12} & 0 & b_{30} & 0 & 0 \\ b_{23} & 0 & b_{16} & b_8 & 0 & 0 & b_{31} & 0 \\ 0 & b_{24} & b_{28} & b_{20} & 0 & 0 & 0 & b_{32} \end{pmatrix}$$

$$
= \begin{pmatrix}
c_1 & c_2 & c_3 & c_4 & * & * & * & * \\
* & c_7 & * & c_8 & * & c_5 & * & c_6 \\
* & * & c_{11} & c_{12} & * & * & c_9 & c_{10} \\
* & c_{15} & c_{16} & * & c_{13} & * & * & c_{14} \\
c_{19} & * & * & c_{20} & * & c_{17} & c_{18} & * \\
c_{23} & c_{24} & * & * & c_{21} & c_{22} & * & * \\
c_{27} & * & c_{28} & * & c_{25} & * & c_{26} & * \\
* & * & * & * & c_{29} & c_{30} & c_{31} & c_{32}
\end{pmatrix}
$$

Thus we now know that one can put 32 entries in $8 \times 8$ matrices. The recursive construction then yields, for $n = 8^k$, matrices with $n^{5/3}$ good entries. Therefore $\tau(n) \geq n^{5/3}$.

This is as far as could be achieved with paper and pencil. We then tried to use computers to improve on this. The result was disastrous. In fact, it took several months and a very complicated and time-consuming program to find the $8 \times 8$ construction. Most of the time the computer would answer with a lousy construction after having spent days searching.

Enter Szemerédi, a powerful combinatorist. While working on a completely different problem, he noticed that an old result from combinatorics could be used to prove that $\tau(n)$ is asymptotically $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$. Having been pointed in the right direction, cite[subquadratic] found (and slightly improved on) the construction which yields this bound. However, in order to beat the $n^{5/3}$ construction, the smallest matrix they could construct is $2^{81} \times 2^{81}$. The yield for these enormous matrices is $n^{1.6875}$, a very slight improvement over the yield of the $8 \times 8$ matrix.. We believe that somewhere between $n = 8$ and $n = 2^{81}$ there is a pair of matrices of size $n \times n$ which has a yield better than $n^{1.6875}$. Perhaps $n$ as low as 10 would work. So far, even for very small values of $n$, we have been unable to prove that the $n^{1.6875}$ cannot be beaten.

## 2.1 Application to cryptography

The matrices problem is of interest on its own right. We have found connections to graph theory, number theory, algebra, etc... However, this being a cryptography paper, we briefly discuss the importance to cryptography of improving on the construction of matrices of size in the practical range.

We shall call a proof of a statement "discreet" if it is impossible for the recipient of the proof to obtain any information beyond the fact that the statement is true.[1] The statement may be a mathematical equation or a statement of the form "I know X".

We illustrate with an example:
consider the 200-digit integer $N$

$$13811004964124334337215332737021459316473868369125360846776305757758$$

$$18616370580095435584782640820169799510659599688388592567994091711118$$

$$0892619657602622485626692180772832155444282672997684087214517747789$$

This integer has exactly two prime factors $P$ and $Q$. We know so because we created the integer in that way. Currently there is no practical computer algorithm capable of

[1]We are deliberately avoiding the term "Zero-Knowledge" here [5], as this is a very complex area with very precise definitions.

finding $P$ and $Q$ without our help. Recently there was a world-wide effort to find the prime factors of a 129-digit number. When the network of several thousand computers succeeded, the results made the New York Times, Scientific American, and other major media. The same coalition has set their current goal at 139-digits. This is estimated to be about 10 times more difficult than 129 digits. Nobody who knows what he/she is doing would currently attempt to get $P$ and $Q$ from this 200 digit number without our help.

Since we are the only one that knows $P$ and $Q$, such knowledge serves to **identify us** to a distrusting party, say a bank clerk. We could give the bank clerk $P$ and he could easily verify that $P$ indeed divides $N$ (integer division is easy and fast). After that, he would let us make a withdrawal from our savings account. But now we have a problem: the clerk knows $P$. Tonight he may get drunk and tell $P$ to all the patrons in the local bar. Our hard-earned money might be gone by tomorrow.

It is a remarkable scientific achievement, of the last fifteen years or so, that ways have been found to convince the teller that we know $P$ without giving him **any** other useful information about $P$. In fact, cryptographers can now do this for any $NP$ statement.

However, the techniques so far developed are somewhat impractical. The paper [1], shows how to make these proofs shorter by using the above matrices. The more good entries you can put into the matrix $C$, the faster their cryptographic protocol goes. The paper [2] gives upper bounds on how far we might be able to push the techniques of [1].

# References


[1] Boyar, J., Brassard, G., Peralta, R.: Subquadratic zero-knowledge. Journal of the Association for Computing Machinery **42** (1995) 1169–1193.

[2] Boyar, J., Fich, F., Larsen, K.: Bounds on certain multiplications of affine combinations. Discrete Applied Mathematics **52** (1994) 155–167.

[3] Fujioka, A., Okamoto, T., Miyaguchi, S.: ESIGN: An efficient digital signature implementation for smart cards. In Advances in Cryptology - Proceedings of EUROCRYPT 91 (1991) pp. 446–457.

[4] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28** (1984) 270–299.

[5] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. SIAM Journal on Computing **18** (1989) 186–208.

[6] Okamoto, T., Fujioka, A., Fujisaki, E.: An efficient digital signature scheme based on an elliptic curve over the ring $z_n$. In Advances in Cryptology - Proceedings of CRYPTO 92 (1993) vol. 740 of Lecture Notes in Computer Science, pp. 54–65.