数理解析研究所講究録
第 716 巻 1990 年 1-9

Paddable Sets in Number Theory

Mitsunori Ogiwara (荻原 光徳)
Department of Information Science
Tokyo Institute of Technology
2-12-1 Ookayama
Meguro-ku Tokyo, 152 Japan

Abstract

A set A is said to be invertibly paddable if there are two polynomial time computable functions pad and decode such that (i) $x \in A$ if and only if $\mathrm{pad}(x, y) \in A$, and (ii) $\mathrm{decode}(\mathrm{pad}(x, y)) = y$. We consider three number theoretical problems that are used in certain cryptosystems (decision of quadratic residuosity, computation of discrete logarithm and computation of Euler's totient function), and show that the sets that represent these problems are invertibly paddable. These results imply that, if these sets are not in P, then they have complexity cores C such that neither C not the complement of C are sparse.

1 Introduction

There are several problems in number theory that seem to be very difficult to solve. Such problems are used in some public-key cryptosystems, where the security of the systems are based on the intractability of such problems. For example, Goldwasser and Micali's cryptosystem is based on the difficulty of solving "quadratic residuosity problem" concerning large composite numbers ([GM 84]). Rivest, Shamir, and Adleman's system is based on the difficulty of computing the Euler's totient function of composite numbers ([RSA 78]). ElGamal's system is based on the difficulty to compute "discrete logarithms"

([ElG 85]). So, it is important to know the complexity theoretical properties of these sets.

In this paper, the problem we are interested in is how dense the hard instances of these problems are distributed. Technically, this problem can be formulated as whether these sets have polynomial complexity core or not. We show that these sets have polynomial time computable invertible padding functions. This result, together with results by Orponen and Schöning [OS 84], shows that these sets and their complements contain non-sparse pro-per polynomial complexity cores under the assumption that these sets are not in P. Hence, these sets and their complements contains non-sparse sets that consist of hard instances only under the same assumption.

2 Definitions, Notations, and Preliminaries

Throughout this paper, all strings will be over the finite alphabet $\Sigma = \{0,1\}$. λ will denote the null string. For a string s, $|s|$ will denote the length of s. All integers will be nonnegative. For an integer a, $\mathrm{E}(a)$ will denote its unique binary representation and for a string s beginning with 1 or $s = 0$, $\tilde{s}$ will denote the unique integer x such that $\mathrm{E}(x) = s$. For an integer a, $\mathrm{len}(a)$ will denote $|\mathrm{E}(a)|$. $\pi(\cdot,\cdot)$ will denote the standard integer pairing function such that, for any nonnegative integers a and b, $\pi(a,b)= \frac{(a+b)(a+b+1)}{2}+a$. $\pi(x_1, x_2, \cdots, x_n)$ will denote $\pi(\pi(\cdots\pi(\pi(x_1,x_2),x_3),\cdots),x_n)$.

Let A be a set of strings. Then A^c will denote Σ^*-A, the complement of A. $|A|$ will denote the cardinality of A. $A^{\leq n}$ will denote $\{x \in A : |x| \leq n\}$. A set A is sparse if there exists a polynomial p such that for all $n \geq 0$, $|A^{\leq n}| \leq p(n)$. A set A is co-sparse if A^c is sparse.

Our concern is to study the paddability of the following number theoretic problems:

1. Quad-Res(a,m) is a decision problem, where a and m are restricted so that $0 < a < m$ and $\gcd(a,m)= 1$, and, the answer is "yes" if a is quadratic residue modulo m(quad. res. mod. m, for short) and the answer is "no", otherwise.

2. Disc-Log(a,r,m) is a computing problem, where a,r, and m are restricted so that $0 < a < m$, $0 < r < m$, and $\gcd(a,m)= \gcd(r,m)= 1$, and, if

there exists an integer $l > 0$ such that $r^l \equiv a \pmod{m}$, the answer is the smallest such l, and if such l does not exist, the answer is 0.

3. Euler(m) is a computing problem, where m is greater than 1, and the answer is $\varphi(m)$, where $\varphi(m)$ is the Euler's totient function.

Since the last two are not decision problems, we introduce corresponding decision problems for them, to which these computing problems are polynomial-time reducible.

2$'$. LB-Disc-Log(a,r,m,k): The answer is "yes" if Disc-Log(a,r,m) $\geq k$ and the answer is "no", otherwise.

3$'$. LB-Euler(m,k): The answer is "yes", if $\varphi(m) \geq k$ and the answer is "no", otherwise.

Now we define the sets corresponding to these decision problems. They are

$$\begin{aligned}
\mathrm{QR} &= \{\mathrm{E}(\pi(a,m)) : \text{Quad-Res}(a,m) = \text{"yes"}\}, \\
\text{LB-DL} &= \{\mathrm{E}(\pi(a,r,m,k)) : \text{LB-Disc-Log}(a,r,m,k) \\
&\quad = \text{"yes"}\}, \text{ and} \\
\text{LB-EU} &= \{\mathrm{E}(\pi(m,k)) : \text{LB-Euler}(m,k) = \text{"yes"}\}.
\end{aligned}$$

Next we define "paddable" sets. The following definition is due to [Sch 85]. We say that a set A is (polynomially) paddable if there is a polynomial-time computable function pad:$\Sigma^* \times \Sigma^* \rightarrow \Sigma^*$, such that

(i) for all x,$y \in \Sigma^*$, $\mathrm{pad}(x,y) \in A \iff x \in A$, and

(ii) for all x,y,and $y' \in \Sigma^*$, $y = y' \iff \mathrm{pad}(x,y) = \mathrm{pad}(x,y')$.

We say that a set A is invertibly paddable if A is polynomially paddable and if there is a poly-nomial-time computable function decode:$\Sigma^* \rightarrow \Sigma^*$ such that

(iii) for all x,$y \in \Sigma^*$, $\mathrm{decode}(\mathrm{pad}(x,y)) = y$.

Berman and Hartmanis have conjectured that all $\leq_m^{\mathcal{P}}$-complete sets in NP are invertibly padd-able[BH 77]. They also showed that this conjecture is equivalent to all $\leq_m^{\mathcal{P}}$-complete sets in NP being polynomially isomorphic. Later, Joseph and Young showed that the conjecture that all $\leq_m^{\mathcal{P}}$-complete sets in NP are polynomially paddable is equivalent to all $\leq_m^{\mathcal{P}}$-complete sets in NP being polynomailly one-one reducible to each other[JY 85].

Finally, we state some well-known results in number theory.

Fact 1 *Let $m = m_1 m_2$, $m_1 > 0$, $m_2 > 0$, $\gcd(m_1, m_2) = 1$, and let a, a_1, and a_2 be integers such that $a \equiv a_i \pmod{m_i}$, for $i = 1, 2$. Then, a is quadratic residue modulo m if and only if a_i is quadratic residue modulo m_i for $i = 1, 2$.*

Fact 2 *Let $m = m_1 m_2$, $m_1 > 0$, $m_2 > 0$, $\gcd(m_1, m_2) = 1$, and let a,a_1,a_2, r,r_1, and r_2 be integers such that $a \equiv a_i \pmod{m_i}$ for $i = 1, 2$ and $r \equiv r_i \not\equiv 0 \pmod{m_i}$ for $i = 1, 2$. Furthermore, let l be a nonnegative integer. Then,*

$$r^l \equiv a \pmod{m} \iff r_i^l \equiv a_i \pmod{m_i} \text{ for } i = 1, 2.$$

3 The Polynomial Paddability

In this section, we prove that each of three sets defined in the previous section is polynomially paddable. We begin with QR.

Theorem 1 QR *is polynomially paddable.*

Proof Assume $x = \mathrm{E}(\pi(a, m))$, $0 < a < m$, and $\gcd(a, m) = 1$. Let $f : N \times \Sigma^* \rightarrow N$ be any function computable in polynomial-time satisfying

(i) for any $m > 1$ and $y \in \Sigma^*$, $\gcd(m, f(m, y)) = 1$, and

(ii) for any $m > 1$ and $y, y' \in \Sigma^*$, $y = y' \iff f(m, y) = f(m, y')$.

(For example, $f(m, y) = m\widetilde{1y} + 1$ satisfies these requirements.)

Moreover, let $M = f(m, y)$ and let μ and ν be any integers such that $\mu m \equiv 1 \pmod{M}$ and $\nu M \equiv 1 \pmod{m}$, and define

$$\mathrm{pad}(x, y) = \mathrm{E}(\pi(a', m')),$$

where $m' = mM = mf(m,y)$ and $a' = (\mu m + a\nu M) \bmod m'$.

Then this function satisfies the requirements for the polynomial paddability. For a given m and M such that $\gcd(m, M) = 1$, the inverse elements μ and ν are computed in polynomial-time using the Euclid's g.c.d. algorithm. Furthermore, $M = f(m, y)$ is computed in polynomial-time. Therefore, pad is computed in polynomial-time.

On the other hand, since $f(m,y)$ is one-to-one on the second component, $\mathrm{pad}(x, y)$ is also one-to-one on the second component.

Finally, from the definition, we have $a' \equiv a \pmod m$ and $a' \equiv 1 \pmod M$. Then, from Fact 1, we have a' is quad. res. mod. $m' \iff a$ is quad. res. mod. m and 1 is quad. res. mod. M. Since $1 \equiv 1^2 \pmod M$, we have a' is quad. res. mod. $m' \iff a$ is quad. res. mod. m, namely $x \in \mathrm{QR} \iff \mathrm{pad}(x,y) \in \mathrm{QR}$. This proves the theorem. Q.E.D.

Remark In the above proof, we did not explain how to define the mapping $\mathrm{pad}(x, y)$ for x's that do not satisfy the restricting conditions. We can easily complete the proof by defining $\mathrm{pad}(x,y) = 0^{|x|+1}1^{|y|+1}xy$ for such x's. In all the remaining proofs of paddability of functions, this mapping will be applied for incorrect x's.

Theorem 2 LB-DL *is polynomially paddable.*

Proof Assume $x = \mathrm{E}(\pi(a, r, m, k))$, $0 < a < m$, $0 < r < m$, and $\gcd(a, m) = \gcd(r, m) = 1$. Let f, M, μ, and ν be as in the proof of Theorem 1, and define

$$\mathrm{pad}(x, y) = \mathrm{E}(\pi(a', r', m', k)),$$

where $m' = mM = mf(x, y)$, $a' = (\mu m + a\nu M) \bmod m'$, and $r' = (\mu m + r\nu M) \bmod m'$.

Similar to the proof in Theorem 1, it is easily seen that pad is computable in polynomial-time and, for any $x, y, y' \in \Sigma^*$, $y = y' \iff \mathrm{pad}(x, y) = \mathrm{pad}(x, y')$.

Moreover, we have $a' \equiv a \pmod m$, $a' \equiv 1 \pmod M$, $r' \equiv r \pmod m$, and $r' \equiv 1 \pmod M$ by definition. Since $1^l \equiv 1 \pmod M$ for all $l > 0$, applying Fact 2, we have $x \in \text{LB-DL} \iff \mathrm{pad}(x,y) \in \text{LB-DL}$.

This proves the theorem. Q.E.D.

Theorem 3 LB-EU *is polynomially paddable.*

Proof Assume $x = \mathrm{E}(\pi(m,k))$ and $m > 1$. Let $L = |y|$ and y_i be the i-th symbol in y $(1 \leq y \leq L)$. Furthermore, let $M = \prod_{i=1}^{L+1} p_i^{e_i}$, where $p_1,\cdots,p_{L+1}$ are the smallest $L+1$ primes not dividing m in increasing order and e_i be integers satisfying, $e_i = 1$ if $y_i = 1$ and $e_i = 0$ if $y_i = 0$ $(1 \leq i \leq L)$ and $e_{L+1} = 1$. Then define

$$\mathrm{pad}(x,y) = \mathrm{E}(\pi(m',k')),$$

where $m' = mM$ and $k' = k\varphi(M)$.

This function satisfies the requirements for the polynomial paddability. Since $\mathrm{len}(m) \geq \log_2 m$, there are only at most $\mathrm{len}(m)$ distinct primes dividing m. So p_{L+1} is not exceeding $[L + 1 + \mathrm{len}(m)]$-th prime number. Since it is known that there exists some constant $c > 0$ such that n-th prime number is less than cn^2(see [?]), p_{L+1} is less than $c'(|x|+|y|)^2$ for some constant $c' > 0$. Therefore, $p_1,\ldots,p_{L+1}$ are computed in polynomial-time. Moreover, since the prime factorization of M is known, $\varphi(M)$ is computed in polynomial-time, and hence pad is computed in polynomial-time.

Furthermore, it is well-known that if $m_1 > 1$, $m_2 > 1$, and $\gcd(m_1, m_2) = 1$, $\varphi(m_1 m_2) = \varphi(m_1)\ \varphi(m_2)$. Since $\gcd(m, M) = 1$ by definition, we have $\varphi(m') = \varphi(m)\varphi(M)$. Therefore, $x \in$ LB-EU $\iff$ $\mathrm{pad}(x,y) \in$ LB-EU.

Finally, it is easily seen that pad is one-to-one on the second component. Therefore, LB-EU is polynomially paddable. Q.E.D.

4 The Invertible Paddability

In this section, we show that all of three sets are invertibly paddable. The invertible paddability is obtained by redefining M in each of three padding functions.

Redefinition of M:

> Let $p_1,\ldots,p_{L+1}$ be the smallest $L+1$ primes not diving m in increasing order, where $L = |y|$. Let $\alpha = \mathrm{len}(m)$ and define e_i to be integers such that, $e_i = \alpha + 1$ if $y_i = 1$ and $e_i = \alpha$ if $y_i = 0$ $(1 \leq i \leq L)$ and $e_{L+1} = \alpha + 2$. Then define $M = \prod_{i=1}^{L+1} p_i^{e_i}$.

It is easy to see that, for any of the three sets, the polynomial paddability is preserved when M is replaced by the above defined value.

On the other hand, decoding function for this version is defined as follows.

Decoding y from the value m':

Let $q_1,\ldots,q_K$ be the smallest K primes in increasing order, where $K = \mathrm{len}(m')$. Let $d_1,\ldots,d_K$ be integers such that $q_j^{d_j}$ divides m' and $q_j^{d_j+1}$ does not divide m' $(1 \le j \le K)$ and define $\beta = \max_{1\le j\le K}\{d_j\} - 2$.

Furthermore, Let $r_1,\ldots,r_{K'}$ be the enumeration of all q_j's such that $d_j = \beta$ or $\beta + 1$ in increasing order and define c_i $(1 \le i \le K')$ to be corresponding d_j's for r_i.

Finally, define y_i $(1 \le i \le K')$ to be symbols such that $y_i = 1$ if $c_i = \beta + 1$ and $y_i = 0$ if $c_i = \beta$, and $y = y_1 \cdots y_{K'}$.

This function satisfies the third requirement for the invertible paddability. For, since m' satisfies

$$m' = mM \ge m \prod_{i=1}^{L+1} p_i^{e_i} \ge m \prod_{i=1}^{L+1} 2^{e_i} \ge m \prod_{i=1}^{L+1} 2,$$

we have $\mathrm{len}(m') \ge L + 1 + \mathrm{len}(m)$. Since p_{L+1} does not exceed the $[L + 1 + \mathrm{len}(m)]$-th prime number, $p_1,\ldots,p_{L+1}$ are in $q_1,\ldots,q_K$. On the other hand, since $\alpha = \mathrm{len}(m) > \log_2 m$, for each prime p dividing m, p^α does not divide m. So $\max d_j$ must be $\alpha + 2$. Hence we have $\beta = \alpha$, $L = K'$, and, $p_1,\ldots,p_L$ are exactly $r_1,\ldots,r_{K'}$ and $e_1,\ldots,e_L$ are exactly $c_1,\ldots,c_{K'}$. Thus y_i's are correctly computed and hence, y is correctly decoded.

And furthermore, it is easily seen that y is computed in polynomial-time in $\mathrm{len}(m')$. Therefore, redefining M gives the invertible paddability.

From the above considerations we have the following theorems.

Theorem 4 QR *is invertibly paddable.*

Theorem 5 LB-DL *is invertibly paddable.*

Theorem 6 LB-EU *is invertibly paddable.*

5 The Paddability and Complexity Cores

In this section, we consider the intractablility of the problems.

For any deterministic Turing machine M and any input x to M, let $t_M(x)$ denote the number of steps that M takes on the input x. If M does not halt on x, $t_M(x) = \infty$.

Let A be any set. We say that a set C is a polynomial complexity core for A if for any deterministic Turing machine M accepting A, and for any polynomial p, $t_M(x)$ is greater than $p(|x|)$, for all but finitely many $x \in C$. We say that a complexity core C for A is proper if $C \subset A$.

The concept of core was introduced by Lynch [Lyn 75]. A set C being a proper polynomial complexity core implies that C is the set of "hardset" elements in A. It is shown by Lynch that A is not in P if and only if A has an infinite polynomial complexity core.

The following propositions are by Orponen and Schöning[OS 84].

Proposition 1 *If a set A is not in P and polynomially paddable, A has a non-sparse proper polynomial complexity core.*

Proposition 2 *If a set A is invertibly paddable, A does not have a co-sparse proper polynomial complexity core.*

Combining these results and the theorems in the previous section, we have the following corollaries.

Corollary 1 *If* QR $\notin P$*, then* QR *has a proper polynomial complexity core C such that both C and C^c are non-sparse.*

Corollary 2 *If* LB-DL $\notin P$*, then* LB-DL *has a proper polynomial complexity core C such that both C and C^c are non-sparse.*

Corollary 3 *If* LB-EU $\notin P$*, then* LB-EU *has a proper polynomial complexity core C such that both C and C^c are non-sparse.*

References

[BH 77] L. Berman and J. Hartmanis, "On Isomorphism and Density of NP and Other Sets," SIAM. J. Comput., 6, pp. 305-322 (1977)

[ElG 85] T. ElGamal, "A Public Key Cryptosystems and a Signature Scheme Based on Discrete Logarithms." IEEE Trans. Inf. Theory, **IT-31**, pp.469-472 (1985)

[GM 84] S. Goldwasser and S. Micali, "Probabilistic Encryption," J. Comput. & Syst. Sci., **28**, pp. 270-299 (1984)

[JY 85] D. Joseph and P. Young, "Some Remarks on Witness Functions for Nonpolynomial and Noncomplete Sets in NP," Theoret. Comput. Sci., **39** pp.225-237 (1985)

[Lyn 75] N. Lynch, "On Reducibility to Complex or Sparse Sets," J. Assoc. Comput. Mach., **22**, pp.341-345 (1975)

[OS 84] P. Orponen and U. Schöning, "The Structure of Polynomial Complexity Core," Proc. 25-th Ann. Symp. Found. Comp. Sci., Lecture Notes in Computer Science **176**, pp.452-458 (1984)

[RSA 78] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." Commun. ACM, **32**, pp.120-126 (1978)

[RS 62] J. B. Rosser and L. Schoenfeld, "Approximate Formulas for Some Functions of Prime Numbers," Illinois J. Math. **6** pp. 64-94 (1962)

[Sch 85] U. Schöning, "Complexity and Structure," Lecture Notes in Computer Science **211**, Springer-Verlag (1985)